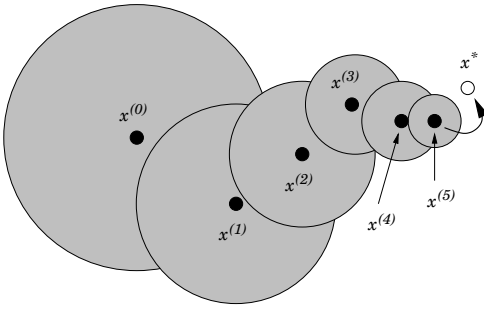
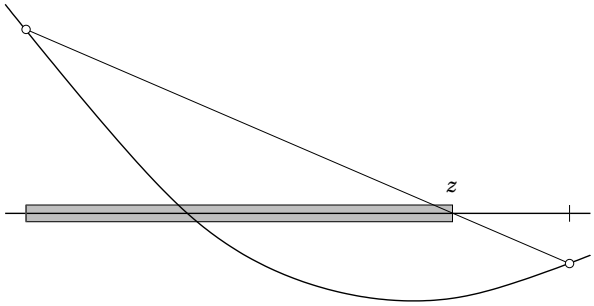


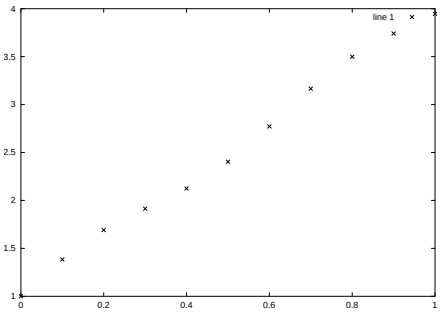


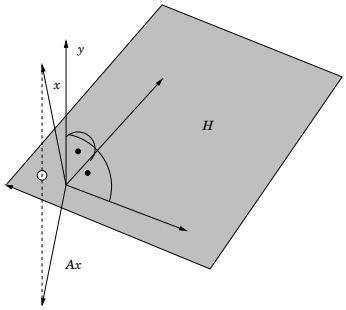
Einführung in die Numerische Mathematik für Hörer aller Fachbereiche

Vorlesung im Wintersemester 2000/2001

Tomas Sauer
Mathematisches Institut
Justus–Liebig–Universität Gießen
Heinrich–Buff–Ring 44
D–35392 Gießen

Version: 14.4.11





Statt einer Leerseite . . .

Es gibt sogenannte Mathematiker, die sich gerne ebenso für Gesandte der Weisheit gehalten wissen wissen möchten als manche Theologen für Gesandte Gottes und ebenso das Volk mit algebraischem Geschwätz, das sie Mathematik nennen, als jene mit einem Kauderwelsch hintergehen, dem sie den Namen biblisch beilegen.
[. . .]
Dieses ist so gewiß, als $(a - x) \cdot (a + x) = a^2 - x^2$ ist.

Georg Christoph Lichtenberg

Inhaltsverzeichnis

To solve a well–conditioned problem with an ill–conditioned method is a numerical crime.

W. Gautschi

1 Was will Numerische Mathematik?

Viele Probleme aus der "realen" Welt lassen sich mathematisch formulieren und dann entweder von Hand oder unter Zuhilfenahme eines Computers lösen. Dabei stellt man normalerweise die folgenden Forderungen an ein Lösungsverfahren:

Interpretierbarkeit: Ein potentieller Anwender will die Lösung interpretieren können. So ist beispielsweise π oder `RootOf( x^4 - 12 x^3 + 9 x^3 - 1 )` in vielen Fällen nicht ausreichend, die Werte 3.141 oder 1.515 dagegen schon.

Effizienz: Die Lösung soll innerhalb eine bestimmten problemabhängigen Zeitspanne (Echtzeit?) geliefert werden.

Genauigkeit: Die berechnete Lösung soll innerhalb einer bestimmten Toleranzschranke um die gewünschte Lösung liegen.

Um die ersten beiden Forderungen, also Interpretierbarkeit und Verfügbarkeit gewährleisten zu können, muß man sich meistens mit einer *Näherung* an die exakte Lösung zufriedengeben. Numerische Mathematik beschäftigt sich mit

- der Konstruktion von Verfahren zum Auffinden von "Lösungen".
- der Analyse dieser Verfahren bezüglich *Effizienz* und *Störungsanfälligkeit*.

Oftmals gibt es natürlich mehrere Lösungsverfahren für ein Problem, die zudem für einen Problemtyp sehr gut, für einen anderen sehr schlecht sein können. Natürlich gibt es auch Verfahren, die immer schlecht sind.

Außerdem sind viele "Verfahren", die man so in "reinen" Mathematikvorlesungen kennenlernt, in der Praxis mit einiger Vorsicht zu genießen. Wir werden uns mal ein paar Beispiele ansehen.

1.1 Berechnung der Varianz

Die Berechnung von Mittelwerten und Varianzen ist eine häufig auftretende "Arbeit" in der Statistik, die so monoton und stupide ist, daß man sie besser dem Computer überläßt. Dabei kann man aber durchaus einiges an Fehlern einbauen, wenn man sich nicht auch um die Numerik kümmert.

Definition 1.1 *Es sei $X = (x_1, \ldots, x_N) \in \mathbb{R}^N$ ein Vektor von Zahlen (Meßwerten ...). Der* Erwartungswert $E(X)$ *ist definiert als*

$$E(X) = \frac{1}{N} \sum_{j=1}^{N} x_j, \tag{1.1}$$

und die Varianz $V(X)$ *sowie die* empirische Standardabweichung $\sigma(x)$ *als*

$$V(X) = \frac{1}{N-1} \sum_{j=1}^{N} \left(x_j - E(X)\right)^2, \qquad \sigma(X) = \sqrt{V(X)}. \tag{1.2}$$

Die Varianz ist nichts anderes als die *mittlerer quadratische Abweichung* vom Erwartungswert, also ein Maß für die *Streuung* der Daten. Die Normierung $\frac{1}{N-1}$ kommt daher, daß man immer annehmen kann, daß $x_1 = E(X)$ ist[1], man also eigentlich nur über $N-1$ Werte summiert, weil man z.B. x_N von allen anderen Werten abziehen kann und so nur die $N-1$ Werte $x_1, \ldots, x_{N-1}$ vorliegen.

Das Problem bei der naiven Berechnung der Varianz besteht nun darin, daß man die Meßwerte *zweimal* durchlaufen müsste, zuerst bei der Berechnung des Erwartungswerts und dann bei der Berechnung der Abweichung vom Erwartungswert. Dies würde es nötig machen, die Meßwerte zwischenzuspeichern, was bei vielen, vielen Meßwerten ein zusätzlicher Programm- und Speicheraufwand wäre. Daher sucht man nach Methoden, die mit nur *einem* Durchlauf durch die Daten auskommen.

In vielen Statistikbüchern findet man die folgende Formel:

$$\sigma(X)^2 = \frac{1}{N-1} \left(\sum_{j=1}^{N} x_j^2 - \frac{1}{N} \left(\sum_{j=1}^{N} x_j \right)^2 \right). \tag{1.3}$$

Damit ergibt sich die Berechnungsvorschrift

$$S_0 = Q_0 = 0 \tag{1.4}$$

$$S_j = S_{j-1} + x_j, \quad Q_j = Q_{j-1} + x_j^2, \qquad j = 1, \ldots, n \tag{1.5}$$

und

$$\sigma(X) = \sqrt{\frac{1}{N-1} \left(Q_N - \frac{S_N^2}{N} \right)}, \tag{1.6}$$

So schön diese Formel ist, so schön kann man damit auf die Nase fallen.

Beispiel 1.2 *Bei Rechnung mit einfacher Genauigkeit*[2] *ergibt sich folgendes:*

[1] Für Mathe–Freaks: Die Varianz ist *invariant* unter Verschiebung des Vektors, das heißt, wenn man von jeder Beobachtung denselben Wert abzieht, interessiert das die Varianz nicht im geringsten

[2] Das kann man beispielsweise über ein `C`–Programm realisieren, wobei man aufpassen muß, denn "intern" rechnen heute eigentlich alle handelsüblichen Prozessoren mit mindestens doppelter Genauigkeit.

```
%%  Varianz1.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Berechnung der Varianz, naiv
%%  Eingabe:
%%    X    Datenvektor

function [V,E] = Varianz1( X )
  N = length( X );
  S = Q = 0;

  for j = 1:N
    S = S + X(j);
    Q = Q + X(j)^2;
  end

  E = S/N;
  V = sqrt( ( Q - S^2 / N ) / (N-1));
```

Programm 1.1 `Varianz1.m`: Varianzberechnung nach (1.4), (1.5) und (1.6).

1. *Wir betrachten* $X = (5000, 5001, 5002)$. *Man sieht leicht, daß* $E(X) = 5001$ *und* $V(X) = \sigma(X) = 1$. *Das Programm liefert aber*[3] *den ziemlich falschen Wert* $\sigma(X) = 0.707107$.
2. *Noch besser wird es für* $X = (10000, 10001, 10002)$*: Das Ergebnis wird dann* `NaN` *(not a number), d.h., es gibt keine Variation mehr ...*

Dieses Verhalten werden wir im nächsten Kapitel erklären. Der Vollständigkeit halber muß man aber auch sagen, daß es auch ein stabiles Verfahren gibt, um die Varianz in *einem* Durchlauf zu berechnen, nämlich mittels der Berechnungsregeln $E_0 = Q_0 = 0$ und

$$Q_j = Q_{j-1} + \frac{(j-1)\,(x_j - E_{j-1})^2}{j}, \qquad j = 1, \ldots, n, \tag{1.7}$$

$$E_j = E_{j-1} + \frac{x_j - E_{j-1}}{j}, \qquad j = 1, \ldots, n. \tag{1.8}$$

Das Ergebnis ist dann

$$\sigma(X) = \sqrt{\frac{Q_N}{N-1}}. \tag{1.9}$$

Dieses Schema berechnet auch in einfacher Genauigkeit die Variation aus Beispiel 1.2 korrekt zu $V(X) = 1.00000$. Übrigens ist M_j immer gerade der Erwartungswert der ersten j Komponenten von X.

[3] Auf (m)einem Pentium II unter Linux, SuSE Version 6.0

```
%%  Varianz2.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Berechnung der Varianz, clever
%%  Eingabe:
%%    X    Datenvektor

function [V,E] = Varianz1( X )
  N = length( X );
  M = Q = 0;

  for j = 1:N
    Q = Q + (j-1) / j * ( X(j) - M )^2;
    M = M + ( X(j) - M )/j;
  end

  E = M;
  V = sqrt( Q / (N-1) );
```

Programm 1.2 `Varianz2.m`: Die “clevere”, das heißt numerisch stabile Art der Varianzberechnung.

```
%%  Heron1.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Berechnung des Flaecheinhalts eines Dreiecks
%%  Eingabe:
%%    a,b,c   Seitenlaengen

function A = Heron1( a,b,c )
  s = (a+b+c) / 2;
  A = sqrt( s*(s-a)*(s-b)*(s-c) );
```

Programm 1.3 `Heron1.m`: Dreiecksfläche á la Heron.

1.2 Flächen von Dreiecken

Ein anderes "klassisches" Problem ist die Berechnung von Dreiecksflächen aus den drei Seitenlängen a, b, c. Die "normale" Formel

$$A_\Delta = \sqrt{s\ (s-a)\ (s-b)\ (s-c)}, \qquad s = \frac{a+b+c}{2},$$

geht bereits auf den griechischen Naturphilosophen *Heron von Alexandria*[4] zurück, siehe [2]. Diese Formel ist gut, solange alle drei Seiten von vergleichbarer Größe sind. Wird jedoch eine der Seiten sehr klein, wird das Dreieck also sehr flach, dann treten Probleme auf.

Beispiel 1.3 *Die Fläche eines rechtwinkligen Dreiecks mit den Seitenlängen* $1, c, \sqrt{1+c^2}$ *ist natürlich* $\frac{1}{2}c$ *– lernt man*[5] *ja schon in der Schule. Berechnen wir nun für kleine Werte von* c *diese Fläche mit Programm 1.3, dann erhalten wir die folgenden Ergebnisse.*

c	Berechnet	Exakt	Fehler in %
1	0.5	0.5	3.33×10^{-14}
10^{-5}	0.5×10^{-5}	0.5×10^{-5}	6.55×10^{-10}
10^{-10}	0.5×10^{-10}	0.5×10^{-10}	6.55×10^{-10}
10^{-15}	0.444×10^{-15}	0.5×10^{-15}	11.12
10^{-16}	0	0.5×10^{-16}	100
$\vdots$	$\vdots$	$\vdots$	$\vdots$
10^{-20}	0	0.5×10^{-20}	100

[4] Lebte etwa von 10 n. Chr. bis 75 n. Chr., Mathematiker, Physiker und Erfinder. Neben der Formel für die Dreiecksfläche bringt man seinen Namen auch mit einem numerischen Verfahren zur Berechnung von Quadratwurzeln in Verbindung, das aber bereits in babylonischen Keilschrifttexten nachgewiesen ist, doch dazu später mehr. Außerdem konstruierte er Automaten, darunter selbsttätig zwitschernde Vögel und eine *Dampfmaschine*.

[5] "Einhalb mal Grundlinie mal Höhe ..."

Daß es bei 10^{-16} *so richtig wild wird, ist kein Zufall. Wie wir bald sehen werden, ist* 10^{-16} *so eine Art "Naturkonstante" bei heutigen PCs (hat etwas mit den* IEEE*–Standards zu tun).*

Auch für die Flächenberechnung von Dreiecken gibt es aber eine stabile[6] Formel, die auf *Kahan* zurückgeht, siehe z.B. [4], nämlich

$$A = \frac{1}{4}\sqrt{(a+(b+c))\,(c-(a-b))\,(c+(a-b))\,(a+(b-c))}.$$

Die Klammer sind dabei übrigens wichtig!

1.3 Aber sowas passiert doch nicht wirklich . . .

Beispiel 1.4 *(Börse von Vancouver)*
Der Aktienindex der Börse von Vancouver wurde in den frühen 80ern mit Computern berechnet. Im Januar 1982 wurde der Aktienindex mit dem Wert 1000 *gestartet und landete im November 1983 bei einem Stand von* 520*, obwohl allem Anschein die individuellen Aktien eigentlich ganz gut standen. Grund: Der Index wurde auf drei Dezimalstellen berechnet und dann wurde* abgeschnitten *anstatt zu runden. Dieser Index wurde (wohl aus dem "alten" Index und den individuellen Differenzen der Aktien) mehrere tausend Mal pro Tag upgedated und jedes Mal wurde abgeschnitten anstatt zu runden. Nach einer Neuberechnung mit Rundung* verdoppelte *sich der Aktienindex beinahe.*

[6] Unter gewissen Voraussetzungen an die Rechnerarithmetik!

Der hundertjährige Krieg währte 113 Jahre

Hanswilhelm Haefs, *Handbuch des nutzlosen Wissens*

2 Fehler und Zahlen

Bei der numerischen Behandlung eines Problems aus der "Realität" können verschiedene Arten von Fehlern auftreten:

Modellierungsfehler: Das mathematische Modell des Problems enthält nicht alle relevanten Parameter. Das ist in vielen Fällen aus Komplexitätsgründen notwendig, manchmal auch schlicht unvermeidbar, da die Parameter ohnehin nicht ermittelt werden können.

Meßfehler: Die Eingabedaten sind nicht zu 100% exakt, sondern mit Meßfehlern behaftet (was in der Realität fast immer passiert).

Verfahrensfehler: Viele numerische Verfahren ermitteln gar keine "exakte" Lösung des Problems[7] sondern "nur" (hoffentlich) immer bessere Näherungen an eine Lösung.

Rechenfehler: Beim Rechnen mit *endlicher* Genauigkeit sind (Rundungs-) Fehler unvermeidbar, die sich natürlich auch "hochschaukeln" können.

Um Fehler als Größen beschreiben zu können verwendet man entweder den *absoluten* oder den *relativen* Fehler.

Definition 2.1 *Es sei $\widehat{x} \in \mathbb{R}$ der gemessene oder berechnete Wert zu $x \in \mathbb{R}$. Dann heißt $x - \widehat{x}$ der* absolute *Fehler von $\widehat{x}$ und $\frac{x-\widehat{x}}{x}$ der* relative *Fehler von $\widehat{x}$.*

Ein paar Bemerkungen zu absoluten und relativen Fehlern:

- Relative Fehler machen natürlich nur dann Sinn, wenn der Ausgangswert x nicht den Wert 0 hat.
- Absolute Fehler sind, wenn es sich um Meßgrößen dreht, mit einer Einheit versehen: Ist $\widehat{x} = 1.39\,\mathrm{m}$ der gemessene Wert zu $x = 1.40\,\mathrm{m}$, dann ist der absolute Fehler $0.01\,\mathrm{m}$. Der relative Fehler ist hingegen *unabhängig* von der Maßeinheit.
- Welcher Fehler nun der wichtigere oder "richtigere" ist, hängt ganz klar vom Problem ab! In manchen Fällen ist durchaus eine *absolute* Paßgenauigkeit vorgegeben, zumeist ist aber der relative Fehler sinnvoller, da er sich "automatisch" an die Problemdimensionen anpaßt.

[7] Das ist eine Lösung, die exakt wäre, wenn man mit exakten Eingangsdaten exakt rechnen könnte.

- Normalerweise werden Fehler als Absolutbeträge definiert – und so werden wir sie im Verlauf dieser Vorlesung auch zumeist abschtzen. Dies ist aber in vielen Fällen unrealistisch: Hat man z.B. etwas zu wenig von einem Werkstück abgefräst, so ist das im allgemeinen leichter zu korrigieren als wenn zuviel abgefräst wurde.

Übung 2.1 Ändert sich der relative Fehler, wenn man die Rollen von x und $\widehat{x}$ vertauscht? Anders gefragt: Spielt es eine Rolle, welcher der beiden Werte der Soll–Wert und welcher der Ist–Wert ist?

2.1 Zahlendarstellung

Das *Stellenwertsystem* stellt eine der großen kulturellen Leistungen der Menschheit dar[8]. Zu einer beliebigen Basis $B \geq 2$ kann man nämlich jede natürliche Zahl $N \in \mathbb{N}$ als

$$N = z_0 + z_1\ B + z_2\ B^2 + \cdots + z_n\ B^n = \sum_{j=0}^{n} z_j\ B^j, \qquad z_j \in \{0, \ldots, B-1\},$$

darstellen. Die Zahlen z_j sind die *Ziffern* von N.

Beispiel 2.2 *Die Zahl* 132 *steht im gebräuchlichen Dezimalsystem (*$B = 10$*) für*

$$1 \times 10^2 + 3 \times 10 + 2 \times 10^0,$$

könnte aber auch im Binärsystem ($B = 2$*) als*

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 12^2 + 0 \times 2^1 + 0 \times 2^0 = 1000100$$

geschrieben werden, im Ternärsystem ($B = 3$*) erhalten wir hingegen*

$$1 \times 3^4 + 1 \times 3^3 + 2 \times 3^2 + 2 \times 3^1 + 0 \times 3^0 = 11220.$$

Auf Digitalcomputern[9] spielt natürlich die Basis $B = 2$ eine entscheidende Rolle, da dann jede Ziffer durch ein *Bit*[10], also durch Strom an/aus dargestellt werden kann.

Reelle Zahlen kann man dann durch eine *unendliche* B–adische Entwicklung darstellen, nämlich als

$$x = z_n \ldots z_1 z_0\ .\ z_{-1} z_{-2} \ldots = \sum_{j=-\infty}^{n} z_j\ B^j.$$

Mathematisch ist diese Darstellung allerdings etwas ungenau: Wenn wir präzise sein wollen, müssen wir diese Zahlen als (konvergente) Reihen auffassen und zwei Darstellungen für gleich

[8] Die Babylonier hatten es, die Römer bekanntlich nicht. In Europa wurde es (via Spanien) von den Arabern übernommen.

[9] Im Gegensatz zu den heute nicht mehr üblichen *Analogrechnern*, die angeblich beliebig exakt rechnen konnten.

[10] "Binary digit"

erklären, wenn sie denselben *Grenzwert* haben. Ansonsten kommt es leicht zum berüchtigten "Laienfehler" $0.99999\ldots \neq 1$, siehe [12], der sich in gewissem Sinne auch hinter dem berühmten *Zenoschen*[11] *Paradoxon* verbirgt [1].

Übung 2.2 Zeigen Sie: Eine Zahl x ist genau dann *rational*, das heißt, als Bruch p/q, $p, q \in \mathbb{N}$, darstellbar, wenn ihre B–adische Entwicklung entweder *endlich* (d.h. $z_{-j} = 0$ für alle hinreichend großen Werte von j) oder *periodisch* (d.h. $z_{-j-M} = z_{-j}$ für ein passendes M und alle hinreichend großen j) ist.

Da ein Computer nur endlich viele (Mega-) Bytes und damit Bits besitzt, ist natürlich auch die Größe der darstellbaren Zahlen beschränkt, sagen wir auf M Ziffern. Damit können wir also *reelle*[12] Zahlen der in der Form

$$x = z_n \ldots z_0 \,.\, z_{-1} \ldots z_{n-M-1}, \qquad n \leq M,$$

darstellen, oder, da eine Multiplikation mit B^n eine *Linksverschiebung* des Punktes um n Stellen bedeutet, wir können nach Umnumerierung der Ziffern x als

$$x = \pm \quad .z_1 \ldots z_M \quad \times \quad B^e, \qquad z_1, \ldots, z_M \in \{0, \ldots, B-1\}, \quad e \in E \subset \mathbb{Z}, \tag{2.1}$$

schreiben. Das Format (2.1) bezeichnet man als *technisch–wissenschaftliches Format* oder auch als *Fließpunkt-* bzw. *Gleitkommadarstellung*[13].

Definition 2.3 *(Fließpunktzahlen)*

1. *Die Menge aller Fließpunktzahlen bezeichnen wir mit*

$$\mathbb{F} = \mathbb{F}(M, E) = \{\pm .z_1 \ldots z_M \times B^e \;:\; 0 \leq z_j < B,\, e \in E\}.$$

2. *Die Zahl* $.z_1 \ldots z_M \in (0, 1)$ *bezeichnet man als* Mantisse *und* M *als* Mantissenlänge*. Die Zahl* e *ist der* Exponent*.*

3. *Die Zahl* $u = \frac{1}{2} B^{1-M}$ *bezeichnet man als* Rundungsfehlereinheit *oder* unit roundoff*.*

4. *Normalerweise sind Fließpunktzahlen* normiert*, das heißt, es ist* $z_1 \neq 0$*, andernfalls heißt die Zahl* subnormal*.*

5. *Der Exponentenbereich* E *ist eine zusammenhängende Menge von ganzen Zahlen, die zumindest* 0 *und* 1 *enthalten sollte.*

[11] Zeno von Elea, 5. Jhdt. vor Christus, Verfasser von Paradoxien, die nachzuweisen suchten, daß es Bewegung eigentlich nicht gibt.

[12] Ja und nein! Es sind ja nur endlich viele Ziffern und damit ist die Zahl streng genommen "nur" rational, das heißt, ein Bruch. Andererseits sind alle rationalen Zahlen natürlich auch reelle Zahlen, wenn auch die eher einfachen Vertreter dieser Zunft.

[13] In der deutschsprachigen Literatur kann man alle vier Kombinationen von Fließ-/Gleit- und -komma/-punkt finden.

6. *Der* darstellbare Bereich $D \subset \mathbb{R}$ *ist definiert als*

$$\pm \left[B^{\min E-1}, \left(1 - B^{-M}\right)\ B^{\max E} \right] \cup \{0\}.$$

Mal wieder ein paar Bemerkungen:

- Die Mantissenlänge M ist verantwortlich für die *Rechengenauigkeit*, die Größe des Exponentenbereichs für die Größe des *darstellbaren Zahlbereichs*.
- Subnormale Zahlen sind ein Problem, da sie eigentlich Rechengenauigkeit verschenken. Sie treten auf, wenn der Exponent e bereits der in E zulässige (negative!) Minimalwert ist und daher nicht weiter normiert werden kann. Viele Numerik–Koprozessoren geben in diesem Fall eine Warnung ab.
- Der Exponentenbereich sollte positive und negative Zahlen enthalten, ist aber im allgemeinen nicht symmetrisch.

Übung 2.3 Wie wird die Dezimalzahl 0.1 in der Basis $B = 2$ dargestellt. Welche Konsequenzen hat das für die zugehörige Fließpunktdarstellung.

Die entscheidende Eigenschaft von Fließpunktzahlen ist, daß sie bezüglich des *relativen* Fehlers ziemlich dicht in D liegen. Mathematisch läßt sich das wie folgt formulieren.

Satz 2.4 *Zu jeder Zahl $x \in D$ gibt es eine Fließpunktzahl $\widehat{x} \in \mathbb{F}$ so daß*

$$\frac{|x - \widehat{x}|}{|x|} \leq u.$$

Übung 2.4 Bestimmen Sie die Fließpunktzahlen mit Basis $B = 10$, Mantissenlänge $M = 3$ und Exponentenbereich $\{-4, \ldots, 3\}$ und plotten Sie deren Verteilung.

Beispiel 2.5 *(IEEE 754) Die in heutigen PC–Prozessoren verwendete Fließpunktarithmetik entspricht dem* IEEE[14] 754*–Standard. Die dabei verwendete Basis ist* 2 *und die arithmetischen Datentypen sind als*

Typ	*Mantisse*	*Exponent*	*Roundoff*	*Größenordnung*
`float`	*23+1*	*8*	$2^{-24} = 5.96 \times 10^{-8}$	$10^{\pm 38}$
`double`	*52+1*	*11*	$2^{-53} = 1.11 \times 10^{-16}$	$10^{\pm 308}$

festgelegt. Das eigentlich interessante am Standard ist aber die Festlegung des Rundungsverhaltens und die Existenz und Weiterverarbeitung von `NaN`*s*[15].
Der Standard IEEE 854 läßt übrigens als Basis die Werte 2 *und* 10 *zu!*

[14] IEEE = Institute of Electrical and Electronical Engineers
[15] `N`ot `a N`umber, Ergebnisse von unzulässigen Operationen wie Wurzeln aus negativen Zahlen

2.2 Rundungsfehler bei Fließpunktrechnung

Beim Rechnen mit Zahlen endlicher Genauigkeit wird im Normalfall der darstellbare Zahlenbereich überschritten. Um also wieder eine darstellbare Zahl erhalten zu können, muß man irgendwas machen.

Beispiel 2.6 *($B = 10$, $M = 2$)*
Das Ergebnis der Addition $.12 + .34 \times 10^{-2}$ *ist* $.1234$*. Bei Addition und Subtraktion* können *Bereichsüberschreitungen auftreten, wenn die Zahlen verschiedene Exponenten haben.*
Das Ergebnis der Multiplikation[16] $.12 \times .34$ *hingegen ist* $.0408$ *– hier* verdoppelt *sich die Anzahl der Stellen.*
Das Ergebnis der Division $.12 \div .34$ *hingegen hat sogar* unendlich viele *Stellen, da es ein periodischer Dezimalbruch ist.*

Das Ziel beim Runden ist klar: Man versucht eine *darstellbare* Zahl zu finden, die nah bei der "Ausgangszahl" liegt. "Nah" heißt hier wieder, daß der *relative* Fehler klein werden soll. Schenken wir uns also den Exponenten und betrachten die normalisierte Zahl[17] $x = .z_1 z_2 \ldots$, $z_1 \neq 0$, dann wählen wir als Näherung

$$\widehat{x} = .z_1 \ldots z_M + \begin{cases} 0, & 0 \leq z_{M+1} < \frac{B}{2}, \\ B^{-M}, & \frac{B}{2} \leq z_{M+1} < B. \end{cases} \tag{2.2}$$

Diese Regel ist nichts anderes als das allseits beliebte "kaufmännische" Runden: Man rundet "ab", wenn die $(M+1)$te Stelle kleiner als $\frac{B}{2}$ ist und man rundet "auf", d.h., man zählt 1 zur letzten, M–ten, Stelle dazu, wenn die $(M+1)$–te Stelle größer oder gleich $\frac{B}{2}$ ist.

Beispiel 2.7 *Bei* $B = 10$ *wird abgerundet, wenn die* $(M+1)$*–te Ziffer* $0, 1, 2, 3, 4$ *ist und aufgerundet, wenn sie den Wert* $5, 6, 7, 8, 9$ *hat. Im Falle* $B = 2$ *ist die Rundungsvorschrift noch viel einfacher: Abrunden falls* $z_{M+1} = 0$ *und Aufrunden wenn* $z_{M+1} = 1$ *– mehr Möglichkeiten gibt es ja nicht.*

Uns Mathematiker interessiert natürlich die Frage, wie genau dieses Rundungsverfahren ist und in der Tat kann man zeigen, daß es für die Hälfte aller Zahlensysteme tatsächlich die "optimale", das heißt nächstliegende Fließpunktzahl liefert[18].

Satz 2.8 *Ist* B *gerade und ist* x *eine normalisierte Fließpunktzahl, dann ergibt die Rundungsvorschrift (2.2), daß*

$$\frac{|\widehat{x} - x|}{|x|} \leq \frac{1}{2} B^{1-M} = u, \qquad x \neq 0. \tag{2.3}$$

[16] Bei der Multiplikation und Division sind die Exponenten (weitestgehend) irrelevant: $(a \times B^x) \cdot (b \times B^y) = (a \cdot b) \times B^{x+y}$.

[17] Wie gesagt: x kann durchaus *unendlich viele* Stellen haben, beispielsweise als Ergebnis einer Division

[18] Der Beweis ist übrigens gar nicht so banal, siehe [11].

Witzig an der Sache ist, daß B gerade sein muß[19]. Das wird aber schnell klar, wenn man sich für ein *ungerades* B (z.B. $B = 5$) einmal die Frage der "Auf-" und "Abrundung" ansieht: Man hat jetzt auch eine *ungerade* Anzahl von Ziffern (z.B. $\{0, 1, 2, 3, 4\}$) und es ist auch nur vernünftig, daß man für $0, \ldots, \frac{B-1}{2} - 1$ (z.B. $0, 1$) abrundet und für $\frac{B-1}{2} + 1, \ldots, B - 1$ (z.B. $3, 4$) aufrundet. Was aber soll man mit der Ziffer $\frac{B-1}{2}$ machen? Auf- oder Abrunden zerstört die Symmetrie und damit macht man dann entweder beim Ab- oder beim Aufrunden[20] einen etwas zu großen Fehler. Bleibt also nur die "Vertagung" auf die nächste Ziffer, aber dann reichen endlich viele Ziffern nicht mehr aus, denn

$$x = .z_1 \ldots z_k \, \frac{B-1}{2} \ldots \frac{B-1}{2} \, z_{N+1}$$

entscheidet sich halt leider erst an Stelle $N + 1$, ob es auf- oder abgerundet werden möchte.

Wie aber wird aber nun "praktisch" gerechnet? Hat man zwei Zahlen, $x = \pm.z_1 \ldots z_M \times B^e$ und $x' = \pm.z'_1 \ldots z'_M \times B^{e'}$ gegeben, dann verfährt man wie folgt:

Addition/Subtraktion: Schiebe die Zahl mit dem *kleineren* Exponenten, sagen wir x', durch Einfügen von $e-e'$ Nullen um entsprechend viele Dezimalstellen nach rechts und berechne[21]

$$\begin{array}{rccccccccccc} & \pm. & z_1 & \ldots & z_M & 0 & \ldots & 0 & 0 & \ldots & 0 & \times B^e \\ + & \pm. & 0 & \ldots & 0 & 0 & \ldots & 0 & z'_1 & \ldots & z'_M & \times B^e \end{array}$$

Multiplikation/Division: Addiere die Exponenten und multipliziere/dividiere die Mantissen.

Für diese Rechenoperationen gilt:

> *Die aufwendigste[22] Rechenoperation ist die Division, die "gefährlichste" die Subtraktion.*

Wir werden uns jetzt mit dem zweiten Aspekt dieser Aussage befassen, denn die meisten numerischen Fehlleistungen stammen tatsächlich daher, daß man die Fallstricke der Subtraktion nicht erkannt hat. Das Hauptproblem hier ist die sogenannte *Auslöschung*.

Beispiel 2.9 *Betrachten wir $B = 10$, $M = 3$ und $x - y$ für $x = .100 \times 10$ sowie $y = .999$. Das Ergebnis ist $.001$ und wird zu $.100 \times 10^{-2}$ normalisiert, also eigentlich doch gar nichts schlimmes. Aber: Die beiden Nullen am Ende des Ergebnisses sind reine* Phantasieziffern*! Um das einzusehen, erinnern wir uns daran, daß vermittels Rundung, ja eigentlich x für "irgendeine Zahl zwischen 0.9995 und 1.005 steht[23] und genauso y für eine Zahl zwischen 0.9985 und 0.9995, das "wahre" Ergebnis liegt also irgendwo zwischen 0 und 0.002. Damit kann aber der relative Fehler dieser Operation sogar 1 (=100%!) werden.*

[19] Was unsere beiden "Standard-Zahlensysteme" $B = 2$ und $B = 10$ ja zufällig (?) sind.

[20] Also bei der Aktion, die "seltener" passiert.

[21] Im Falle der Addition – das reicht natürlich, da $x - x' = x + (-x')$.

[22] Das Wort kommt schließlich von "aufwenden", nicht von "aufwanden" – letzteres klingt ziemlich nach "mauern".

[23] Um ganz mathematisch exakt zu sein: $x \in [0.9995, 1.005)$, wobei der rechte Endpunkt des *halboffenen* Intervalls nicht mehr dazugehört.

Übung 2.5 Können die extremalen Werte 0 und 0.002 angenommen werden?

In vielen Numerikbüchern wird die Annahme gemacht, die Fließpunktrechnung würde nach dem Motto

rechne exakt, normalisiere und runde dann

durchgeführt, was nicht ganz realitätsnah, aber handlich ist – nach Satz 2.4 haben dann alle Operationen einen relativen Fehler von höchstens u. In Wirklichkeit hat aber die Recheneinheit, der sogenannte *Akkumulator*, nur eine endliche Länge, die natürlich nicht kleiner als M sein sollte. Tatsächlich sollte er aber ein kleines bißchen größer sein.

Satz 2.10 *(Länge des Akkumulators)*

1. *Bei Rechnung mit M–stelligem Akkumulator kann bei der Subtraktion ein relativer Fehler von bis zu $B-1$ auftreten*[24].

2. *Bei Rechnung mit $(M+1)$–stelligem Akkumulator ("Guard digit") beträgt der relative Fehler bei allen Operationen höchstens $2u$.*

Beispiel 2.11 *Nochmals zurück zu $1.00 - .999$ bei dreistelligem Akkumulator. In diesem Fall erhalten wir*

$$\begin{array}{rll} & .100 & \times 10^1 \\ - & .099 & \times 10^1 \\ \hline & .001 & \times 10^1 \end{array} \qquad = .100 \times 10^{-1},$$

also 0.01 anstelle von 0.001 – der vorhergesagte Fehler von 900%.

Übung 2.6 Beweisen Sie Teil 1) von Satz 2.10 für beliebiges $B \geq 2$.

2.3 Fehlerfortpflanzung

Eigentlich sind wir ja jetzt in einer ganz guten Situation: Wenn unsere Rechnerarithmetik über einen mindestens $(M+1)$–stelligen Akkumulator, also eine "Guard digit" verfügt[25], dann können alle Fließpunktoperationen $\oplus, \ominus, \otimes, \oslash$ mit einer relativen Genauigkeit von $\widehat{u} := 2u$ durchgeführt werden, das heißt

$$\frac{|x \odot y - x \cdot y|}{|x \cdot y|} \leq \hat{u}, \qquad \cdot = +, -, \times, /. \tag{2.4}$$

[24] "Glücklicherweise" rechnen die meisten Computer mit der Basis $B = 2$, so daß man maximal um 100% danebenliegen kann – was natürlich schlimm genug ist.

[25] Beispielsweise alle PCs mit "normalen" Fließkommaprozessoren haben so was, wer es peinlicherweise nicht hatte, waren numerische (!) Supercomputer der Firma Cray, was auch prompt zu katastrophal falschen Ergebnissen führte, auch und gerade im Zusammenhang mit der Formel für Dreiecksflächen von Kahan. Dies wird (angeblich) in [7] beschrieben.

Die Forderung (2.4) wird oft auch als *Standardmodell der Fließpunktrechnung* bezeichnet – bis auf den Faktor 2 bei $\hat{u}$ entspricht sie auch der Philosophie "Rechne exakt und runde".

Übung 2.7 Zeigen Sie, daß man (2.4) auch als

$$x \odot y = (1+\delta)\,(x \cdot y)\,, \qquad \delta \in [-\hat{u}, \hat{u}]\,, \qquad \cdot = +, -, \times, /, \tag{2.5}$$

schreiben kann.

Leider haben aber die Rundungsfehler, selbst wenn jede einzelne Operation (2.4) erfüllt, eine starke Tendenz, sich über mehrere Operationen fortzupflanzen und dabei extrem zu verstärken. Und, um das Ganze noch interessanter zu machen, es kommt dabei oftmals noch darauf an, wie man die Operationen anordnet.

Beispiel 2.12 *Wir, wollen für $M = 3$ und $B = 10$, den Wert $x^2 - y^2$ berechnen, wobei $x = .334$ und $y = .333$ sein sollen*[26]*. Diese Berechnung können wir auf zwei Arten durchführen.*

$(x \otimes x) \oplus (y \otimes y)$**:** *Hier ist*[27] $x \otimes x = .112$ *und* $y \otimes y = .111$ *und damit erhalten wir* $.100 \times 10^{-2}$ *als Ergebnis.*

$(x \oplus y) \otimes (x \ominus y)$**:** *Es ist* $x \oplus y = .667$ *und* $x \ominus y = .100 \times 10^{-2}$*, also erhalten wir* $.667 \times 10^{-3}$ *als Ergebnis.*

Welches der beiden Ergebnisse ist nun richtig? Wenn man genau hinsieht, zeigt sich, daß wir im zweiten Fall immer exakt *gerechnet haben, also muß das auch der richtige Wert sein. Das heißt aber, wir machen im ersten Fall einen relativen Fehler von*[28]

$$\frac{.100 \times 10^{-2} - .667 \times 10^{-3}}{.667 \times 10^{-3}} = \frac{.333 \times 10^{-3}}{.667 \times 10^{-3}} \sim .5,$$

also von der Größenordnung $50\hat{u}$ (zur Erinnerung: $\hat{u} = 2\frac{1}{2}B^{-M} = B^{1-M} = 10^{-2}$)!

Was uns dieses Beispiel zeigt, ist, daß es für das Problem "Berechne $x^2 - y^2$" zwei *mathematisch* äquivalente numerische Verfahren gibt, nämlich

1. Berechne $a = x \otimes x$ sowie $b = y \otimes y$ und dann $a \ominus b$
2. Berechne $a = x \oplus y$ sowie $b = x \ominus y$ und dann $a \otimes b$

die sich in ihrer *numerischen* Qualität ziemlich dramatisch unterscheiden können. Was wir also brauchen, ist ein Konzept, um "üble" Probleme[29] von schlechten Verfahren trennen zu können.

[26] Dies verwundert nun nicht mehr – schließlich wissen wir ja schon, daß die "interessanten" Effekte gerne im Zusammenspiel mit Auslöschung auftreten.

[27] Exakt gerechnet und gerundet, also im Einklang mit (2.4)

[28] Exakt gerechnet ...

[29] Sowas gibt's auch.

2.4 Rückwärtsfehler und Konditionszahl

Jetzt wird's mal kurzzeitig ein bißchen mathematischer und abstrakter, denn das Konzept, das wir entwickeln wollen, soll ja auf eine möglichst große Anzahl von Problemen anwendbar sein und nicht nur auf die Addition von zwei ganzen Zahlen.

Beginnen wir mit dem Rückwärtsfehler und einem ganz einfachen Problem: Zu einem *fest* vorgegebenen Wert $y \in \mathbb{F}$ soll *numerisch* der Wert $f(x) = x \cdot y$ berechnet werden. Nach (2.5) ist dann, für ein $\delta \in [-\hat{u}, \hat{u}]$,

$$\widehat{f}(x) = x \otimes y = (1+\delta)xy = \underbrace{((1+\delta)x)}_{=:\hat{x}} \cdot y = \hat{x} \cdot y = f(\hat{x}),$$

wobei

$$|\hat{x} - x| = |(1+\delta)x - x| = |\delta|\,|x| \le \hat{u}|x| \qquad \Longrightarrow \qquad \frac{|\hat{x}-x|}{|x|} \le \hat{u}.$$

Eigentlich haben wir eben etwas sehr seltsames gemacht, nämlich den *Rechenfehler*, der während des numerischen Verfahrens aufgetreten ist, auf die *Eingabedaten*, nämlich x geschoben und uns dann angesehen, inwieweit dieser Fehler absolut und relativ beschränkt ist. Das ist auch schon die Idee des *Rückwärtsfehlers*, die auf Wilkinson[30] zurückgeht. Aber bevor wir ihn formal definieren, erst noch ein etwas komplizierteres Beispiel. Jetzt sei $f(x) = x \cdot y + c$ und unser numerisches Verfahren ist[31]

$$\begin{aligned} \widehat{f}(x) &= (x \otimes y) \oplus c = ((1+\delta)x \cdot y) \oplus c = ((1+\delta)x \cdot y + c)(1+\epsilon) \\ &= (1+\delta)(1+\epsilon)x \cdot y + c + \epsilon c. \end{aligned}$$

Das unschuldige $\epsilon \cdot c$ am Ende macht nun etwas Schwierigkeiten, denn diesen Teil des Fehler können wir ja anscheinend nicht auf x abschieben. Irrtum! Unter der Annahme $xy \neq 0$ schreiben wir den Ausdruck einfach wie folgt um:

$$\widehat{f}(x) = (1+\delta)(1+\epsilon)x \cdot y + c + \epsilon\frac{x \cdot y}{x \cdot y}c = \underbrace{\left((1+\delta)(1+\epsilon) + \epsilon\frac{c}{xy}\right)x}_{=:\hat{x}}\, y + c = f(\hat{x})$$

und dann ist

$$\frac{|\hat{x}-x|}{|x|} = \left|1 + \epsilon + \delta + \epsilon\delta + \epsilon\frac{c}{xy} - 1\right| = \left|\delta + \epsilon\left(1 + \delta + \frac{c}{xy}\right)\right| \le \left(2 + \frac{c}{xy}\right)\hat{u} + \hat{u}^2.$$

Die Qualität des Verfahrens steht und fällt also mit der Frage, ob c groß ist im Vergleich zu xy oder nicht.

[30] J. H. Wilkinson, einer der (britischen) Pioniere der Numerischen Mathematik und der theoretischen Untersuchung des Rechnens auf Computern.

[31] Achtung: Reihenfolge und Klammern können entscheidend sein. Es ist nicht unbedingt $(x \oplus y) \oplus z = x \oplus (y \oplus z)$.

Definition 2.13 *Sei $f : \mathbb{R} \to \mathbb{R}$ eine beliebige Funktion und $\widehat{f}$ ein numerisches Verfahren dafür. Wir sagen, f hat einen (relativen)* Rückwärtsfehler *von höchstens $\rho > 0$, wenn es zu jedem $x \neq 0$ ein $\hat{x}$ gibt, so daß*

$$\hat{f}(x) = f(\hat{x}) \qquad \textit{und} \qquad \frac{|\hat{x} - x|}{|x|} \leq \rho.$$

Hat $f = f(x_1, \ldots, x_n)$ hingegen mehrere Eingabeparameter, so betrachten wir zuerst die komponentenweisen Rückwärtsfehler $\rho_j > 0$

$$\hat{f}(x_1, \ldots, x_n) = f(\hat{x}_1, \ldots, \hat{x}_n) \qquad \textit{und} \qquad \frac{|\hat{x}_j - x_j|}{|x_j|} \leq \rho_j, \quad j = 1, \ldots, n,$$

und sprechen vom normweisen Rückwärtsfehler

$$\rho = \max_{j=1,\ldots,n} \rho_j.$$

Was ist nun so wichtig am Rückwärtsfehler, daß man ihn sogar in einer Vorlesung für Hörer aller Fachbereiche einbauen muß?

1. Dadurch, daß Verfahrensfehler auf die Eingabewerte geschoben werden, werden sie genauso behandelt wie Eingabefehler oder Rundung inexakter Eingabedaten (hat hier jemand `0.1` eingegeben?).

2. Der Rückwärtsfehler liefert wirklich *verfahrensabhängige* Abschätzungen, wie wir gleich in Beispiel 2.14 sehen werden. Aber: Die Fehlerabschätzungen sind nur *obere* Abschätzungen, die zwar korrekt, aber nicht *scharf* sind[32]

3. Diese Abschätzungen sind zwar im allgemeinen ziemlich grob, aber sie zeigen deutlich auf, für welche Konfigurationen das Verfahren schlecht werden kann.

Beispiel 2.14 *(Rückwärtsfehler für die Summation)*
Für $x = (x_1, \ldots, x_n)$ betrachten wir die $S(x) = x_1 + \cdots + x_n$, also die Summe der Komponenten von x.

1. *Für das "naive" Summationsverfahren $S_0 = 0$, $S_j = S_{j-1} + x_j$, $j = 1, \ldots, n$, kann man ziemlich einfach ermitteln, daß*[33]

$$\widehat{S}_n(x) = S_n(\hat{x}), \qquad \frac{|\hat{x}_j - x_j|}{|x_j|} \preceq (n + 1 - j)\hat{u}, \quad j = 1, \ldots, n.$$

[32] Etwa: "Die Flügelspannweite einer Stubenfliege ist kleiner als ein Meter" ist (hoffentlich) korrekt, aber nicht sonderlich aussagekräftig.

[33] In erster Näherung ...

Das kann man sich noch relativ leicht intuitiv vorstellen: Die Komponente x_j ist an $n-j$ Additionen beteiligt und hat so $n-j$ Möglichkeiten, einen Rundungsfehler "abzubekommen"[34].

2. *Es gibt aber tatsächlich etwas besseres, nämlich das Summationsverfahren von Kahan, bei dem*

$$\frac{|\hat{x}_j - x_j|}{|x_j|} \preceq 2\hat{u}, \qquad j = 1, \ldots, n,$$

unabhängig *von j! Allerdings ist sowohl das Verfahren wie auch sein Beweis ziemlich trickreich, siehe [11] oder [3], von wo auch der Beweis abgeschrieben ist. Die Summationsregel lautet übrigens*

$$\begin{aligned} s_1 &= x_1, \quad c_1 = 0 \\ Y &= x_j \ominus c_{j-1}, \quad T = s_{j-1} \oplus Y, \quad c_j = (T \ominus s_{j-1}) \ominus Y, \quad s_j = T, \qquad j = 2, \ldots, n, \end{aligned}$$

und das Ergebnis ist die Zahl s_n.

Die Rückwärtsfehleranalyse ermittelt also, zu welchen Veränderungen der Eingabedaten die Verwendung eines bestimmten numerischen Verfahrens führt. Wenn wir jetzt also noch wissen, wie empfindlich die Funktion f auf Veränderungen reagiert, haben wir eine Möglichkeit, Aussagen über den relativen Fehler zu machen. Und genau diese "Empfindlichkeit" ist durch die *Konditionszahl* des Problems beschrieben.

Definition 2.15 *Die* relative Konditionszahl $\kappa_f(x)$ *einer Abbildung*[35] *ist eine (möglichst kleine) Schranke, so daß*

$$|f(x) - f(x+\delta)| \leq |f(x)| \ \kappa_f(x) \ |\delta|, \qquad f(x) \neq 0, \tag{2.6}$$

solange δ hinreichend klein ist.

Ein paar Bemerkungen zu diesem Begriff:

1. Die Konditionszahl ist eine sogenannte *Linearisierung* des Fehlers.
2. Die Konditionszahl hängt nur von f und x, nicht aber vom verwendeten Verfahren ab und ist so ein Maßstab wie gut das Problem generell lösbar ist.

[34] Wen es interessiert: Jeder dieser Rundungsfehler sorgt für einen multiplikativen Faktor $(1+\delta)$, wobei $|\delta| \leq \hat{u}$ ist, der relative Gesamtfehler ist also von der Größenordnung

$$(1+\hat{u})^{n-j} - 1 = \sum_{k=1}^{n-j} \binom{n-j}{k} \hat{u}^k = (n-j)\hat{u} + \ldots$$

Da bei einer vernünftigen Arithmetik $\hat{u}$ sehr klein sein sollte (also $\hat{u}^2$ noch viel, viel kleiner, ganz zu schweigen von höheren Potenzen von $\hat{u}$), vernachlässigt man einfach diese Terme.

[35] Das ist das Problem, das wir numerisch angehen wollen.

3. Zusammen mit dem Rückwärtsfehler ergibt sich die "berühmte" Abschätzung

$$\text{Relativer Fehler} \leq \text{Konditionszahl} \times \text{Rückwärtsfehler},$$

   wobei auf der rechten Seiten die Fehlerquellen "entkoppelt" sind.

4. Die (ziemlich schwammige) Forderung "δ klein" ist einerseits vernünftig[36] und andererseits notwendig, da auch bei "braven" Problemen die *globalen* Konditionszahlen, die sich für $\delta \in \mathbb{R}$ ergeben, sinnlos sind, siehe Übung 2.8.

5. Formt man um, so ist

$$\kappa_f(x) \geq \frac{|f(x+\delta) - f(x)|}{|\delta|} \frac{1}{|f(x)|}$$

   und ist f differenzierbar und wird $|\delta|$ ganz klein ($|\delta| \to 0$), dann ist

$$\kappa_f(x) \sim \left| \frac{f'(x)}{f(x)} \right|.$$

Übung 2.8 Zeigen Sie, daß für $f(x) = x^2$ und jedes $x \in \mathbb{R}$ man $\kappa_f(x) = \infty$ erhält, wenn δ in (2.6) alle reellen Zahlen durchlaufen darf.

Das Prinzip von Rückwärtsfehler und Konditionszahl ist in Abb. 2.1 auch noch einmal grafisch dargestellt.

[36] Man geht eigentlich immer von Verfahren aus, die einen moderaten (etwa 10^{-3}) Rückwärtsfehler haben, ansonsten läßt man es ohnehin besser bleiben.

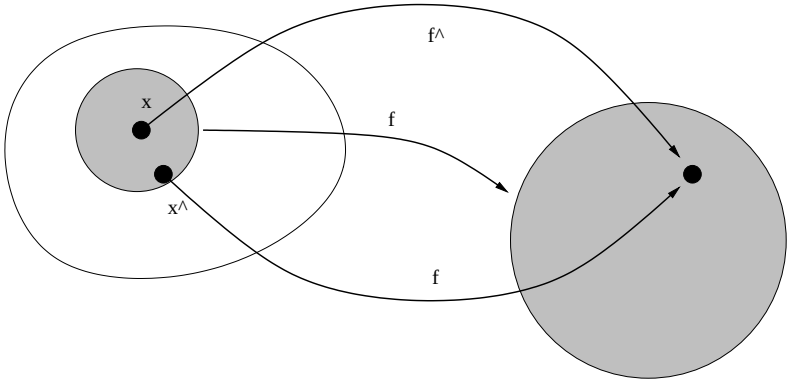


Abbildung 2.1: Rückwärtsfehler und Konditionszahl. Zuerst findet man ein (natürlich möglichst gutes) $\widehat{x}$ so daß $\hat{f}(x) = f(\widehat{x})$, der Rückwärtsfehler ist dann der Radius eines (möglichst kleinen) Kreises um x, der auch $\widehat{x}$ enthält. Dieser Kreis wird nun durch f auf ein anderes Gebilde (hier der Einfachheit halber ebenfalls ein Kreis) abgebildet und das Verhältnis aus dem Radius dieses Gebildes und des Ausgangskreises ist schließlich die Konditionszahl.

> $\frac{1}{4}$ *Breite und Länge zusammen sind* 7 *Handbreiten, Länge und Breite zusammen sind* 10 *Handbreiten.*
>
> Babylonisches Gleichungssystem in zwei Unbekannten, Susa, 2. Jahrtausend v. Chr

3 Lineare Gleichungssysteme

Lineare Gleichungssysteme zählen zu den ältesten (siehe Zitat) aber immer noch aktuellen Problemen der Numerischen Mathematik und es ist kein Zufall, daß große Namen wir Gauß[37] mit Lösungsverfahren für lineare Gleichungssysteme in Verbindung zu bringen sind.

Die einfachsten linearen Gleichungssysteme sind natürlich die, die aus einer Gleichung in einer Variablen bestehen, also von der Form

$$ax = b, \qquad a, b \in \mathbb{R}, \tag{3.1}$$

sind. Hier können wir sofort eine vollständige "Lösungstheorie" angeben:

1. Ist $a \neq 0$, dann gibt es zu jedem $b \in \mathbb{R}$ genau eine Lösung x von (3.1), nämlich $x = \frac{b}{a}$.
2. Ist hingegen $a = 0$, dann gibt es nur dann eine Lösung x, wenn auch $b = 0$ ist – dann ist aber sogar *jedes* $x \in \mathbb{R}$ eine Lösung des Gleichungs"systems" (3.1).

Wir sehen also schon: Die Lösbarkeit von solchen Systemen wird wesentlich von dem Faktor a abhängen.

Machen wir's nun etwas komplizierter und behandeln wir das babylonische Gleichungssystem von oben, dann erhalten wir in "moderner" algebraischer[38] Notation unter Verwendung von ℓ für die Länge und b für die Breite die *beiden* Gleichungen

$$\begin{aligned} \frac{1}{4}b + \ell &= 7 \\ \ell + b &= 10. \end{aligned}$$

Wie löst man nun sowas auf naive Art und Weise[39]? Nun, wir könnten beispielsweise über die zweite Gleichung ℓ in Abhängigkeit von b als $\ell = 10 - b$ schreiben und das in die erste Gleichung einsetzen, was uns

$$\frac{1}{4}b + 10 - b = 7 \qquad \Longleftrightarrow \qquad \frac{3}{4}b = 3,$$

[37] Carl Friedrich Gauß, 1777–1855, der Mann auf dem Zehnmarkschein, weit mehr als Erfinder der allseits beliebten "Glockenkurve".

[38] Der Name "Algebra" stammt von der *Al'Gabr* des *Al'Chwarizmi*, der sich seinerseits als Namensgeber für das Wort "Algorithmus" verewigt hat. Das zeigt schon, daß unser heutiges "Buchstabenrechnen" arabischen Ursprungs ist und über das mittelalterliche Spanien nach Europa kam.

[39] Aber nicht "nach Adam Riese"! Der hat's nämlich ganz anders gemacht, aber dazu später mehr.

also $b = 4$ und somit $\ell = 6$ liefert. Dieses Verfahren ist ganz nett, aber für mehr Gleichungen in einer größeren Anzahl von Variablen nur noch etwas für Masochisten.

Für eine vernünftige Beschreibung unseres Gleichungssystems in einer ähnlichen Form wie (3.1) brauchen wir nun erst mal etwas Terminologie.

3.1 Matrizen und Vektoren

Für uns hier ist ein *Vektor*, genauer, ein *Spaltenvektor* $x \in \mathbb{R}^n$ ein n*–Tupel*

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = [x_j \ : \ j = 1, \ldots, n]$$

von reellen Zahlen. Mathematisch ist dies übrigens inkorrekt! Ein Vektor ist eigentlich ein Element eines *Vektorraums*, also eines Gebildes, das bestimmten Axiomen[40] genügt – allerdings ist jeder n–dimensionale Vektorraum *isomorph* zum $\mathbb{R}^n$. Aber Spaß beiseite – analog ist ein *Zeilenvektor* von der Form

$$x^T = [x_1 \ \cdots \ x_n] \, .$$

So künstlich der Unterschied auch aussieht, man muß leider darauf achten. Vektoren kann man nun

1. *addieren:* Für $x, y \in \mathbb{R}^n$ ist

$$x + y = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix} = [x_j + y_j \ : \ j = 1, \ldots, n] \, , \tag{3.2}$$

2. *mit einer Zahl multiplizieren:* Für $x \in \mathbb{R}^n$ und $c \in \mathbb{R}$ ist

$$cx = xc = \begin{bmatrix} cx_1 \\ \vdots \\ cx_n \end{bmatrix} = [c\, x_j \ : \ j = 1, \ldots, n] \, , \tag{3.3}$$

3. *skalar multiplizieren:* Für $x, y \in \mathbb{R}^n$ ist

$$x^T y = y^T x = \sum_{j=1}^{n} x_j \, y_j . \tag{3.4}$$

Übung 3.1 Wie lauten die Regeln für Addition und Multiplikation mit einer Zahl bei Zeilenvektoren?

[40] Mathematische Konkretisierung des Intuitiven

Besonders interessant ist (3.4). Man sieht leicht, daß diese Multiplikation in x und y *linear* ist, das heißt, daß für $x, x', y, y' \in \mathbb{R}^n$

$$x^T\,(c\,y + c'\,y') = c\,x^T y + c'\,x^T y' \quad \text{und} \quad (c\,x + c'\,x')^T\,y = c\,x^T y + c'\,x'^T y, \qquad c, c' \in \mathbb{R},$$

aber viel reizvoller wird für $x, y, z \in \mathbb{R}^n$ der Ausdruck

$$x\,\underbrace{\left(y^T z\right)}_{\in\mathbb{R}} = \left(xy^T\right) z,$$

der ja nach (3.3) ein Vektor sein muß. Also ist xy^T ein Gebilde, das Vektoren in Vektoren überführt und das bezüglich z linear ist, das heißt

$$\left(xy^T\right)(c\,z + c'\,z') = c\,\left(xy^T\right) z + c'\,\left(xy^T\right) z'.$$

Solche Gebilde bezeichnet, also lineare Abbildungen von $\mathbb{R}^n$ nach $\mathbb{R}^m$ bezeichnet man als *Matrizen*[41], die man pratischerweise als rechteckiges Schema

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

darstellt. Dabei ist also a_{jk} der Eintrag in *Zeile* j und *Spalte* k, $j = 1, \dots, m$, $k = 1, \dots, n$. Wenn wir nun genau hinsehen, könnten wir sagen, daß

1. Spaltenvektoren $n \times 1$–Matrizen,
2. Zeilenvektoren $1 \times n$–Matrizen

sind – und genauso sehen das `Matlab` und `Octave` auch.
Nun können wir die Matrix aber auch anders auffassen, nämlich entweder als einen *Spaltenvektor von Zeilenvektoren*

$$A = \begin{bmatrix} \boxed{\begin{matrix} a_{11} & \dots & a_{1n} \end{matrix}} \\ \vdots \\ \boxed{\begin{matrix} a_{m1} & \dots & a_{mn} \end{matrix}} \end{bmatrix} = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}, \qquad a_j = \begin{bmatrix} a_{j1} \\ \vdots \\ a_{jn} \end{bmatrix} \in \mathbb{R}^n, \quad j = 1, \dots, m, \tag{3.5}$$

oder aber als einen *Zeilenvektor von Spaltenvektoren*

$$A = \begin{bmatrix} \boxed{\begin{matrix} a_{11} \\ \vdots \\ a_{m1} \end{matrix}} & \dots & \boxed{\begin{matrix} a_{1n} \\ \vdots \\ a_{mn} \end{matrix}} \end{bmatrix} = \left[\bar{a}_1 \cdots \bar{a}_n\right], \qquad \bar{a}_j = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix} \in \mathbb{R}^m, \quad j = 1, \dots, n. \tag{3.6}$$

Und tatsächlich sagt uns nun (3.5) und (3.6), wie Matrizen und Vektoren miteinander zu multiplizieren sind:

[41] Wieder mal mathematisch ungenau: Eigentlich sind Matrizen *Darstellungen* linearer Abbildungen von zwischen endlichdimensionalen Vektorräumen bezüglich einer Basis. Aber wen kümmert's?

Für $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ und $y \in \mathbb{R}^m$ ist

$$Ax = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} x = \begin{bmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{bmatrix} = \left[\sum_{k=1}^{n} a_{jk}x_k \;:\; j = 1,\ldots,m\right], \tag{3.7}$$

sowie

$$y^T A = y\,[\bar{a}_1 \cdots \bar{a}_m] = \left[\sum_{k=1}^{m} a_{kj}y_k \;:\; j = 1,\ldots,n\right]. \tag{3.8}$$

Aus (3.7) erhalten wir als erstes eine Formel für die Matrix $A = xy^T$, $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$. Für jedes $z \in \mathbb{R}^n$ ist ja

$$\left[\sum_{k=1}^{n} a_{jk}z_k \;:\; j = 1,\ldots,m\right] = Az = \left(xy^T\right)z = x\left(y^T z\right) = \left[\sum_{k=1}^{n} (x_j y_k)\,z_k \;:\; j = 1,\ldots,m\right],$$

also

$$xy^T = \begin{bmatrix} x_1y_1 & \ldots & x_1y_n \\ \vdots & \ddots & \vdots \\ x_my_1 & \ldots & x_my_n \end{bmatrix} = \left[x_j y_k \;:\; \begin{matrix} j = 1,\ldots,m \\ k = 1,\ldots,n \end{matrix}\right]. \tag{3.9}$$

Und eigentlich sagt uns (3.7) sogar noch mehr: Wir haben ja die Matrix $A \in \mathbb{R}^{m\times n}$ mit der "Matrix" $x \in \mathbb{R}^{n\times 1}$ multipliziert und eine "Matrix" in $\mathbb{R}^{m\times 1}$, den Spaltenvektor erhalten. Nun, dann können wir aber auch eine Matrix $A \in \mathbb{R}^{m\times r}$ mit einer Matrix $B \in \mathbb{R}^{r\times n}$ multiplizieren[42], indem wir (3.9) ausnutzen, um

$$\begin{aligned} AB &= \begin{bmatrix} \boxed{a_{11} \;\ldots\; a_{1r}} \\ \vdots \\ \boxed{a_{m1} \;\ldots\; a_{mr}} \end{bmatrix} \begin{bmatrix} \boxed{b_{11}} & \cdots & \boxed{b_{1n}} \\ \boxed{\vdots} & & \boxed{\vdots} \\ \boxed{b_{r1}} & & \boxed{b_{rn}} \end{bmatrix} \\ &= \left[A\bar{b}_1 \;\cdots A\bar{b}_n\right] = \begin{bmatrix} a_1^T B \\ \vdots \\ a_m^T B \end{bmatrix} = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \left[\bar{b}_1 \cdots \bar{b}_n\right] = \begin{bmatrix} a_1^T\bar{b}_1 & \ldots & a_1^T\bar{b}_n \\ \vdots & \ddots & \vdots \\ a_m^T\bar{b}_1 & \ldots & a_m^T\bar{b}_n \end{bmatrix} \\ &= \left[\sum_{\ell=1}^{r} a_{j\ell}b_{\ell k} \;:\; \begin{matrix} j = 1,\ldots,m \\ k = 1,\ldots,n \end{matrix}\right], \end{aligned}$$

zu erhalten – und damit haben auch schon wir die Multiplikationsformel für Matrizen hergeleitet:

$$AB = \left[\sum_{\ell=1}^{r} a_{j\ell}b_{\ell k} \;:\; \begin{matrix} j = 1,\ldots,m \\ k = 1,\ldots,n \end{matrix}\right]. \tag{3.10}$$

[42] Entscheidend ist ja nur, daß beide Matrizen an der "Berührseite" dieselbe Größe haben!

Die angenehmste Matrix bezüglich der Multiplikation ist die *Einheitsmatrix*

$$I = I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

die als *"multiplikative Einheit im Ring der $m \times n$–Matrizen[43]"* fungiert, was heißt, daß für jede Matrix $A \in \mathbb{R}^{m\times n}$

$$A = I_m A = A I_n$$

gilt.

Übung 3.2 Geben Sie die Additionsregel für $m \times n$–Matrizen sowie für die Multiplikation mit einer Zahl $c \in \mathbb{R}$ an.

3.2 Lineare Gleichungssysteme als Matrizen

Nehmen wir also an, wir stünden vor einem linearen Gleichungssystem, in dem m Gleichungen mit n Unbekannten zu lösen sind, das also die Form

$$\begin{array}{ccccc} a_{11}x_1+ & \cdots & +a_{1n}x_n & = & b_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{m1}x_1+ & \cdots & +a_{mn}x_n & = & b_m \end{array}$$

hat, oder, in unserer schönen neuen Matrixform

$$Ax = b, \qquad A \in \mathbb{R}^{m\times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \tag{3.11}$$

was nun schon ziemlich nach (3.1) aussieht. Wenn wir nun so ein Gleichungssystem lösen, dann werden ja durch die Lösung die Werte x_j, $j = 1, \dots, n$, *bestimmt* und zwar "kann man", wie die Beispiele am Anfang gezeigt haben, ja anscheinend mit jeder Gleichung eine Variable bestimmen. Aus diesem Grund nennt man ein Gleichungssystem

1. *unterbestimmt*, wenn $m < n$,
2. *quadratisch*, wenn $m = n$,
3. *überbestimmt*, wenn $m > n$.

Es wird sich herausstellen, daß im "generischen" Fall, das ist die Situation die "praktisch immer", aber leider nicht immer in der Praxis, auftritt, unterbestimmte Gleichungssysteme *jede Menge* Lösungen[44], quadratische Gleichungssysteme *genau eine* Lösung und überbestimmte Gleichungssysteme *gar keine* Lösung.

[43] Was auch immer das ist.

[44] Genauer: Man kann im generischen Fall $n - m$ Parameter der Lösung frei wählen, die übrigen m Parameter hängen dann von diesen ab

Zuerst wollen wir uns nun einmal mit quadratischen Gleichungssystemen beschäftigen, wie es ja auch unsere beiden Beispiele am Anfang waren. Dazu ein wichtige Begriff aus der Linearen Algebra[45], nämlich den der *Invertierbarkeit* einer Matrix.

Definition 3.1 *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt* invertierbar*, wenn es eine Matrix $A^{-1} \in \mathbb{R}^{n\times n}$ gibt, so daß*[46]

$$A^{-1}A = AA^{-1} = I$$

ist.

Übung 3.3 Zeigen Sie:

1. Die Inverse einer Matrix A ist eindeutig.

2. Sind $A, B \in \mathbb{R}^{n\times n}$ invertierbar, dann ist

$$(AB)^{-1} = B^{-1}A^{-1}.$$

Wenn man einen Begriff definiert, dann sollte man immer zeigen, daß der Begriff auch sinnvoll ist, daß es also mindestens ein Objekt gibt, das die Definition erfüllt[47] und daß es auch ein Objekt gibt, daß die Definition *nicht* erfüllt[48]. Also, schau'n mer mal ...

Beispiel 3.2 *(Invertierbare Matrizen)*

1. *Die Matrix I ist offenbar invertierbar und sogar ihre eigene Inverse (aber nicht die einzige Matrix mit dieser Eigenschaft). In der Tat sind sogar fast alle Matrizen invertierbar, genauer, die invertierbaren Matrizen bilden eine* offene und dichte *Teilmenge*[49]*, das heißt:*

   (a) *Ist A invertierbar und $B \in \mathbb{R}^{n\times n}$ beliebig, dann ist $A + tB$ invertierbar, solange nur $|t|$ klein genug ist ("offen").*

   (b) *Ist $A \in \mathbb{R}^{n\times n}$ beliebig, dann ist $A + tI$ invertierbar, solange nur $|t|$ hinreichend klein und $t \neq 0$ ist ("dicht").*

[45] Auch wenn Lineare Algebra heute eine Anfängervorlesung des ersten Semesters ist und in banalisierter Form sogar an Schulen gelehrt wird, ist sie eine eher junge Teildisziplin der Mathematik, die erst etwa um 1850 das Licht der Welt erblickte. *Gelöst* hingegen wurden Gleichungssysteme schon beinahe 4000 Jahre vorher.

[46] Folgt die Größe einer Einheitsmatrix direkt aus dem Kontext, dann werde ich den Subskript, der ihre Größe angibt, weglassen. Hier steht also I für I_n.

[47] Man könnte ja auch über die Menge der "schwarzen Schimmel" reden. Schlimmer noch: In der formalen Logik sind für solche leeren Menge *per definitionem* **alle** Aussagen richtig.

[48] Über "weiße Schimmel" zu reden ist eine sogenannte *Tautologie*, zwar kein logisches Desaster, aber halt einfach eine überflüssige Begriffsbildung.

[49] Ja, schon wieder fachchinesisches (mathesinologisches?) Kauderwelsch.

2. *Die* Nullmatrix $0 = 0_n$ *ist nicht invertierbar, denn es ist ja* $A0 = 0A = 0$ *für alle* $n \times n$*–Matrizen* A*, also kann es keine Matrix* $A = 0^{-1}$ *geben, so daß* $A0 = 0A = I$*. Zugegeben, kein besonders aufregendes Beispiel, aber jetzt wird es interessanter: Hat* A *eine* Nullzeile*, das heißt, ist* $a_j = 0$ *für mindestens ein* $1 \leq j \leq n$*, dann ist*

$$AB = \begin{bmatrix} a_1^T \\ \vdots \\ a_j^T \\ \vdots \\ a_n^T \end{bmatrix} \begin{bmatrix} \bar{b}_1 \cdots \bar{b}_n \end{bmatrix} = \begin{bmatrix} a_1^T\bar{b}_1 & \dots & a_1^T\bar{b}_n \\ \vdots & \ddots & \vdots \\ a_j^T\bar{b}_1 & \dots & a_j^T\bar{b}_n \\ \vdots & \ddots & \vdots \\ a_n^T\bar{b}_1 & \dots & a_n^T\bar{b}_n \end{bmatrix} = \begin{bmatrix} a_1^T\bar{b}_1 & \dots & a_1^T\bar{b}_n \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ a_n^T\bar{b}_1 & \dots & a_n^T\bar{b}_n \end{bmatrix}$$

und AB *kann für kein* B *der Welt eine Einheitsmatrix werden, weil die* j*–te Zeile nur aus Nullen besteht. Etwas entsprechendes passiert, wenn* B *eine Nullspalte hat – dann hat auch das Ergebnis eine Nullspalte.*

Übung 3.4 Folgern Sie die Dichtheit der (Menge der) invertierbaren Matrizen aus deren Offenheit.

So, was haben nun invertierbare Matrizen mit unserem Problem, lineare Gleichungssysteme zu lösen, zu tun? Die Antwort ist: Alles, sie sind das Gegenstück zur Unterscheidung $a = 0$ bzw. $a \neq 0$ in (3.1) und der theoretische Unterbau der gesamten Lösbarkeitsfrage.

Satz 3.3 *Für eine Matrix* $A \in \mathbb{R}^{n\times n}$ *sind äquivalent.*

1. A *ist invertierbar.*
2. *Für alle* $b \in \mathbb{R}^n$ *hat das lineare Gleichungssystem* $Ax = b$ *genau eine Lösung.*
3. *Für alle* $0 \neq x \in \mathbb{R}^n$ *ist* $Ax \neq 0$*.*

Ausnahmsweise wollen wir diesen Satz größtenteils beweisen – der Beweis ist einfach, elementar und gibt hoffentlich ein bißchen Einsicht in die Struktur der Problems.
Beweis: Wir beweisen zunächst, daß 1 und 2 äquivalent sind. Und wirklich: Ist A invertierbar, so multiplizieren wir beide Seiten von $Ax = b$ mit A^{-1} und erhalten

$$x = A^{-1}\underbrace{Ax}_{=b} = A^{-1}b.$$

Umgekehrt verwenden wir die *Einheitsvektoren* $e^j \in \mathbb{R}^n$, die an der j–ten Stelle den Wert 1 und sonst den Wert 0 haben, definieren x^j als die Lösung von $Ax^j = e^j$ und setzen $B = [x^1 \cdots x^n]$. Dann ist

$$AB = \begin{bmatrix} Ax^1 \cdots Ax^n \end{bmatrix} = \begin{bmatrix} e^1 \cdots e^n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} = I,$$

also[50] $B = A^{-1}$ und damit ist A invertierbar.
Für die Äquivalenz von 2 und 3 zeigen wir die "Äquivalenz der Negation"[51] beider Aussagen – was genau dasselbe ist. Gibt es nämlich eine rechte Seite b, für die es mindestens *zwei* Lösungen gibt[52], als ein $b \in \mathbb{R}^n$ so daß $Ax = Ax' = b$ für $x \neq x' \in \mathbb{R}^n$, dann ist

$$A(x - x') = Ax - Ax' = b - b = 0 \qquad \text{und} \qquad x - x' \neq 0,$$

und ist, umgekehrt $0 \neq x'$ so, daß $Ax' = 0$, dann ist

$$A(x \pm x') = Ax \pm Ax' = b \pm 0 = b \qquad \text{und} \qquad x \pm x' \neq x.$$

□

Aus diesem Beweis können wir nun zwei wichtige Schlußfolgerungen ziehen:

1. Kennt man die Inverse einer Matrix, so kann man das zugehörige lineare Gleichungssystem sofort lösen, kann man hingegen das lineare Gleichungssystem für alle rechten Seiten b lösen, so läßt sich damit relativ einfach die Inverse der Matrix A bestimmen. In diesem Sinne sind die beiden Probleme also etwa "gleichschwer".

2. Genauso sind die Nulllösungen und die Mehrdeutigkeiten miteinander verwandt: Jede Nulllösung liefert eine "Familie" von Mehrdeutigkeiten und umgekehrt.

3.3 Einfache lineare Gleichungssysteme – Dreiecksmatrizen

Jetzt aber erst einmal Schluß mit der Theorie und ran an die *Berechnung* der Lösungen des linearen Gleichungssystems. Unsere Strategie wird "zweiteilig" sein: Zuerst suchen wir nach Typen von Gleichungssystemen, die wir einfach lösen können, das heißt nach speziellen Matrizen, und dann werden wir Verfahren kennenlernen, die "beliebige" Matrizen in diese einfache Form zerlegen.
Der einfachste Fall eines eindeutig lösbaren Gleichungssystems wäre natürlich $A = I$, aber das können wir uns schenken. Auch *Diagonalmatrizen*, das heißt Matrizen der Form

$$D = \begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & d_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

sind nicht viel schwerer, da sie *entkoppelte* Gleichungen der Form $d_{jj}x_j = b_j$, $j = 1, \ldots, n$, liefern, die genau dann lösbar sind, wenn alle $d_{jj} \neq 0$ sind.

Übung 3.5 Bestimmen Sie die Inverse einer invertierbaren Diagonalmatrix.

Noch relativ einfach sind Matrizen, die sozusagen "halbe" Diagonalmatrizen sind, die sogenannten *Dreiecksmatrizen*.

[50] Der Beweis ist nicht 100% vollständig. Man müßte noch zeigen, daß die *Rechtsinverse* B auch eine *Linksinverse* ist, also $BA = I$, aber das geht.

[51] Mathesinologie

[52] Man kann zeigen, daß es für jede "unlösbare" rechte Seite auch eine rechte Seite mit mindestens zwei Lösungen geben muß, das hat mit der *Dimension* von Vektorräumen und dem *Rang* einer Matrix zu tun ...

Definition 3.4 *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt*

1. Rechtsdreiecksmatrix, *wenn $a_{jk} = 0$ wann immer $k < j$ ist, das heißt, wenn sie die Form*

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & \dots & 0 & a_{nn} \end{bmatrix} \simeq \begin{bmatrix} * & * & \dots & * \\ 0 & * & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \dots & 0 & * \end{bmatrix}$$

   hat[53].

2. Linksdreiecksmatrix, *wenn $a_{jk} = 0$ wann immer $k > j$ ist, das heißt, wenn*

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

   ist.

Im weiteren wird der Buchstabe L immer eine Linksdreiecksmatrix, der Buchstabe R immer eine Rechtsdreiecksmatrix bezeichnen.

Die Gleichungssysteme, die sich aus Rechts- und Linksdreiecksmatrizen ergeben, also $Rx = b$ bzw. $Lx = b$ haben nun die Form

$$\begin{array}{ccccccccccc} r_{11}x_1 & + & r_{12}x_2 & +\cdots+ & r_{1,n-1}x_{n-1} & + & r_{1n}x_n & = & b_1 \\ & & r_{22}x_2 & +\cdots+ & r_{2,n-1}x_{n-1} & + & r_{2n}x_n & = & b_2 \\ & & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & r_{n-1,n-1}x_{n-1} & + & r_{n-1,n}x_n & = & b_{n-1} \\ & & & & & & r_{nn}x_n & = & b_n \end{array}$$

beziehungsweise

$$\begin{array}{ccccccccccc} \ell_{11}x_1 & & & & & & & & & = & b_1 \\ \ell_{21}x_1 & + & \ell_{22}x_2 & & & & & & & = & b_2 \\ \vdots & \vdots & \vdots & & \ddots & & & & & \vdots & \vdots \\ \ell_{n-1,1}x_1 & + & \ell_{n-1,2}x_2 & +\cdots+ & \ell_{n-1,n-1}x_{n-1} & & & & & = & b_{n-1} \\ \ell_{n1}x_1 & + & \ell_{n2}x_2 & +\cdots+ & \ell_{n,n-1}x_{n-1} & + & \ell_{nn}x_n & = & b_n \end{array}$$

und sind besonders einfach zu lösen. Das sehen wir uns mal am Beispiel von $Lx = b$ an: Zuerst lösen wir die erste Gleichung, das heißt $\ell_{11}x_1 = b_1$ und erhalten

$$x_1 = \frac{b_1}{\ell_{11}}, \qquad \ell_{11} \neq 0. \tag{3.12}$$

[53] Hier und im Folgenden steht das Symbol "$*$" für eine *beliebige* reelle Zahl und es ist nicht verboten, daß diese Zahl den Wert 0 annimmt.

```
%%  VorElim.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Vorwaertselimination, ueberschreibt b.
%%  Eingabe:
%%    L    Linksdreiecksmatrix
%%    b    rechte Seite

function x = VorElim( L,b )
  n = length( b );

  for j = 1:n
    for k = 1:j-1
      b(j) = b(j) - L(j,k) * b(k);
    end
    b(j) = b(j) / L(j,j);
  end

  x = b;
%endfunction
```

Programm 3.1 `VorElim.m`: Die Vorwärtselimination nach (3.14).

Dann greifen wir uns die zweite Gleichung, schreiben sie in $\ell_{22}x_2 = b_2 - \ell_{21}x_1$ um, setzen (3.12) ein und erhalten

$$x_2 = \frac{1}{\ell_{22}}\left(b_2 - \ell_{21}x_1\right) = \frac{1}{\ell_{22}}\left(b_2 - \frac{\ell_{21}}{\ell_{11}}b_1\right), \qquad \ell_{22} \neq 0. \tag{3.13}$$

Generell erhalten wir so für $j = 1, \ldots, n$ die Regel

$$x_j = \frac{1}{\ell_{jj}}\left(b_j - \sum_{k=1}^{j-1} \ell_{jk}x_k\right), \qquad \ell_{jj} \neq 0, \tag{3.14}$$

die man als *Vorwärtselimination* bezeichnet. Eine analoge Regel erhält man auch für Rechtsdreiecksmatrizen, diese bezeichnet man dann als *Rücksubstitution*.

Übung 3.6 Geben Sie das Gegenstück zu (3.14) für die Rücksubstitution an.

3.4 Einfache lineare Gleichungssysteme – orthogonale Matrizen

Es gibt aber noch andere Matrizen, die ganz einfach zu invertieren (aber wohl schwerer zu erhalten oder auch nur zu identifizieren) sind. Dazu brauchen wir zuerst den Begriff der *Transponierten*[54]

[54] Wenn man's genau nimmt, haben wir diesen Begriff heimlich, stille, leise und implizit bereits bei den Zeilenvektoren verwendet.

```
%%  VorElim.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Vorwaertselimination, ueberschreibt b.
%%  "Richtiges Matlab" (mit Skalarprodukt)
%%  Eingabe:
%%    L    Linsdreiecksmatrix
%%    b    rechte Seite

function x = VorElim2( L,b )
  n = length( b );

  b(1) = b(1) / L(1,1);
  for j = 2:n
    b(j) = ( b(j) - L(j,1:j-1) * b(1:j-1) ) / L(j,j);
  end

  x = b;
%endfunction
```

Programm 3.2 `VorElim2.m`: Effizientere Realisierung der Vorwärtselimination. Grund: `Matlab` kann (Stichwort: “Vektorisierung”) Skalarprodukte sehr viel schneller berechnen als Schleifen.

```
%%  RueckSubs.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Ruecksubstitution, ueberschreibt b
%%  Eingabe:
%%    U    Rechtsreiecksmatrix
%%    b    rechte Seite

function x = RueckSubs( U,b )
  n = length( b );

  for j = n : -1 : 1
    for k = n : -1 : j+1
      b(j) = b(j) - U(j,k) * b(k);
    end
    b(j) = b(j) / U(j,j);
  end

  x = b;
%endfunction
```

Programm 3.3 `RueckSubs.m`: Rücksubstitution, eine entsprechende beschleunigte Version ist das `m`–File `RueckSubs2.m`.

A^T einer Matrix A, das ist diejenige Matrix, bei der die Rolle von Zeilen und Spalten vertauscht wird.

Definition 3.5 *Sei* $A = \left[a_{jk} \ : \ \begin{smallmatrix} j=1,\ldots,m \\ k=1,\ldots,n \end{smallmatrix}\right] \in \mathbb{R}^{m\times n}$. *Die* Transponierte *zu* A *ist die Matrix*

$$A^T = \left[a_{kj} \ : \ \begin{matrix} k = 1,\ldots,n \\ j = 1,\ldots m \end{matrix}\right] \in \mathbb{R}^{n\times m}.$$

Beispiel 3.6 *(Transponierte)*

1. *Die Transponierte eines Spaltenvektors (als* $n\times 1$*–Matrix) ist ein Zeilenvektor (eine* $1\times n$*–Matrix).*

2. *Schreiben wir* A *wieder als den Spaltenvektor der Zeilenvektoren, also*

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} = \begin{bmatrix} \boxed{a_{11} \quad \ldots \quad a_{1r}} \\ \vdots \\ \boxed{a_{m1} \quad \ldots \quad a_{mr}} \end{bmatrix},$$

   dann ist die Transponierte davon der Zeilenvektor der aus den transponierten Zeilenvektoren von A *– das sind dann Spaltenvektoren – gebildet wird*[55]*, also*

$$A^T = [a_1 \cdots a_m] = \begin{bmatrix} \boxed{\begin{matrix} a_{11} \\ \vdots \\ a_{1n} \end{matrix}} & \cdots & \boxed{\begin{matrix} a_{m1} \\ \vdots \\ a_{mn} \end{matrix}} \end{bmatrix}.$$

3. *Eine Matrix* $A \in \mathbb{R}^{n\times n}$ *heißt* symmetrisch*, wenn sie ihre eigene Transponierte ist, das heißt, wenn* $A^T = A$ *ist. Die Einheitsmatrix ist beispielsweise symmetrisch.*

Übung 3.7 Zeigen Sie: Für $A \in \mathbb{R}^{n\times n}$ sind die Matrizen A^TA und AA^T symmetrisch.

Definition 3.7 *Eine Matrix* $Q \in \mathbb{R}^{n\times n}$ *heißt* orthogonal*, wenn* $Q^TQ = QQ^T = I$ *ist.*

Was das nun schon wieder heißt und vor allem wo der Name herkommt erfahren wir, wenn wir uns mal ansehen, wie die Orthogonalität von Matrizen aussieht, wenn wir Q als Spaltenvektor von Zeilenvektoren bzw. als Zeilenvektor von Spaltenvektoren schreiben, denn dann ist

$$\begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 0 & \ldots & 0 & 1 \end{bmatrix} = I = QQ^T = \begin{bmatrix} q_1^T \\ \vdots \\ q_n^T \end{bmatrix} [q_1 \cdots q_n] = \begin{bmatrix} q_1^Tq_1 & \ldots & q_1^Tq_n \\ \vdots & \ddots & \vdots \\ q_n^Tq_1 & \ldots & q_n^Tq_n \end{bmatrix}$$

[55] Das klingt komplizierter als es ist. Man kann ja mal versuchen, diesen Satz zehnmal hintereinander schnell aufzusagen.

beziehungsweise

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} = I = Q^T Q = \begin{bmatrix} \bar{q}_1^T \\ \vdots \\ \bar{q}_n^T \end{bmatrix} [q_1 \cdots q_n] = \begin{bmatrix} \bar{q}_1^T \bar{q}_1 & \dots & \bar{q}_1^T \bar{q}_n \\ \vdots & \ddots & \vdots \\ \bar{q}_n^T \bar{q}_1 & \dots & \bar{q}_n^T \bar{q}_n \end{bmatrix},$$

das heißt, die Zeilen- und Spaltenvektoren von Q haben die Eigenschaft, daß

$$q_j^T q_k = \bar{q}_j^T \bar{q}_k = \begin{cases} 1 & j = k, \\ 0 & j \neq k. \end{cases}$$

In diesem Fall sagt man auch, die Vektoren seien *orthogonal* – sie stehen aufeinander senkrecht.

Beispiel 3.8 *Die "einfachsten" orthogonalen Matrizen sind, im* 2×2*–Fall, die* Rotationsmatrizen

$$R_\phi = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}, \qquad \phi \in [0, 2\pi].$$

Geometrisch entsprechen Sie einer Drehung um den Winkel ϕ *im Uhrzeigersinn, siehe Abb. 3.1. Diese Matrizen haben nun schöne Eigenschaften:*

1. *Sie sind orthogonal:*

$$\begin{aligned} R_\phi R_\phi^T &= \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \\ &= \begin{bmatrix} \cos^2\phi + \sin^2\phi & -\cos\phi\sin\phi + \cos\phi\sin\phi \\ -\cos\phi\sin\phi + \cos\phi\sin\phi & \cos^2\phi + \sin^2\phi \end{bmatrix} \end{aligned}$$

   und da bekanntlich[56] *für jeden Winkel* ϕ *die Identität* $\sin^2\phi + \cos^2\phi = 1$ *gilt, liefert das die Einheitsmatrix.*

2. *Für jeden Drehwinkel* ϕ *ist*

$$R_{-\phi} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} = R_\phi^T.$$

Beispiel 3.9 *Ein anderer wichtiger Typ von orthogonalen Matrizen sind* Spiegelungen. *Formal ist zu einem Vektor* $y \in \mathbb{R}^n$ *die zugehörige Spiegelung als*

$$H_y = I - 2\,\frac{yy^T}{y^T y}$$

definiert[57]*. Anschaulich heißt das, daß* $H_y x$ *einen Vektor* $x \in \mathbb{R}^n$ *nimmt und an der Ebene*

$$H = \left\{ h \in \mathbb{R}^n \ : \ h^T y = 0 \right\}$$

spiegelt, siehe Abb. 3.2.

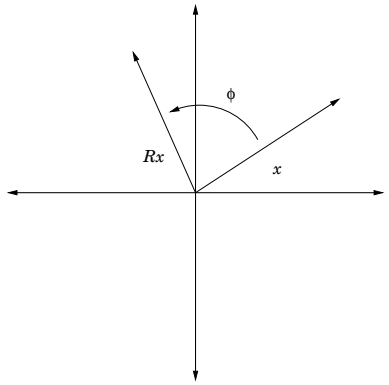


Abbildung 3.1: Die Drehung eines Vektors (dargestellt als Pfeil mit Anfang im Ursprung) um den Fußpunkt im Ursprung wird realisiert als Multiplikation mit einer Rotationsmatrix.

Nun aber zu der langen Rede kurzem Sinn. Warum sind orthogonale Matrizen so schön einfach. Ganz einfach, ist nämlich $Q \in \mathbb{R}^{n\times n}$ orthogonal, dann ist

$$Qx = b \qquad \Longleftrightarrow \qquad x = Q^T b. \tag{3.15}$$

3.5 Direkte Verfahren – Faktorisierungsmethoden

Die Strategie bei Faktorisierungsmethoden besteht darin, eine Matrix in ein Produkt zweier Matrizen zu zerlegen (sie zu "faktorisieren") und dann die beiden resultierenden Gleichungssysteme separat zu lösen. Natürlich wählt man die Zerlegung so, daß diese beiden Systeme *einfach* sind[58], das heißt, wir interessieren uns nun dafür

1. ob und wann wir eine beliebige Matrix $A \in \mathbb{R}^{n\times n}$ als Produkt einer orthogonalen Matrix mit einer Rechtsdreiecksmatrix ("QR–Zerlegung") oder als Produkt einer Linksdreiecksmatrix mit einer Rechtsdreiecksmatrix ("LR–Zerlegung") darstellen können.

2. wie wir diese Faktoren *algorithmisch*[59] bestimmen können.

Bevor wir beginnen, uns mit diesen Fragen zu beschäftigen, erst noch eine allgemeine Vorbemerkung. Nehmen wir einfach mal an, wir hätten eine Darstellung

$$A = QR \qquad \text{oder} \qquad A = LR, \tag{3.16}$$

[56] Fällt unter Schulbildung.

[57] Hier sieht man wieder mal sehr schön, daß yy^T und $y^T y$ zwei ganz verschiedene Paar Schuhe sind.

[58] Denn die "Strategie", das Lösen *eines* Gleichungssystems "nur so" auf das Lösen *zweier* Gleichungssysteme zurückzuführen wäre so idiotisch wie sie klingt, wenn diese Gleichungssysteme nicht entscheidend einfacher zu lösen wären.

[59] Und wenn's geht auch noch effizient.

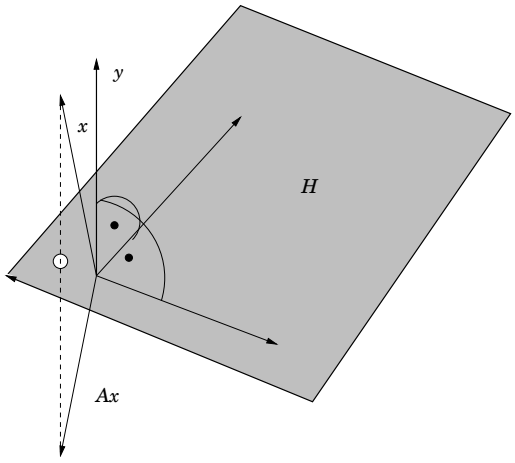


Abbildung 3.2: Die Geometrie von $Ax = H_y x$. Spiegelung heißt hierbei nicht "Reflektion" (wie bei einem Spiegel) sondern entspricht physikalisch der *Brechung* an einer planen Fläche, beispielsweise einer unendlich dünnen Glasscheibe. Beim Durchgang durch eine "richtige" Glasscheibe mit endlicher Ausdehnung wird der Lichtstrahl *zweimal* gebrochen – an jeder Grenzfläche einmal.

wobei Q für eine orthogonale Matrix steht, L für eine Linksdreiecksmatrix und R für eine Rechtsdreiecksmatrix. Achtung: Die beiden Matrizen "R" in der QR- und der LR–Zerlegung in (3.16) sind normalerweise *unterschiedliche* Matrizen!
Nun nehmen wir außerdem an, daß A invertierbar wäre – eine vernünftige Minimalvoraussetzung, wenn wir eine eindeutige Lösung des Gleichungssystems $Ax = b$ für alle rechten Seiten b erhalten wollen. Ist aber eine *invertierbare* Matrix als Produkt zweier Matrizen darstellbar, dann müssen aber auch diese beiden Faktoren invertierbar sein[60] und wir können also (3.16) in

$$Q^T A = R \qquad \text{oder} \qquad L^{-1} A = R \tag{3.17}$$

umschreiben. Nun ist natürlich $Q^T = Q^{-1}$ wieder eine orthogonale Matrix[61], aber man kann auch zeigen, daß die Inverse einer unteren Dreiecksmatrix wieder eine untere Dreiecksmatrix ist, und zwar indem man sich ansieht, was die Vorwärtselimination mit *Einheitsvektoren*[62] anstellt, denn das liefert ja[63] die jeweiligen *Spalten* der Inversen von L.

Übung 3.8 Zeigen Sie, daß die Inverse einer Linksdreiecksmatrix wieder eine Linksdreiecksmatrix ist.

[60] Dies ist das "Matrix–Gegenstück" zu der Tatsache, daß wenn das Produkt zweier Zahlen nicht Null ist, dann auch diese beiden Zahlen von Null verschieden sein müssen. Die Umkehrung gilt übrigen für Matrizen (im Gegensatz zu Zahlen) *nicht* mehr.

[61] Denn die Transponierte $\left(Q^T\right)^T = Q$ von Q^T erfüllt ja $Q^T Q = Q Q^T = I$ und ist damit auch die Inverse von Q^T.

[62] Das sind Vektoren von der Form $e_j = [0, \ldots, 0, 1, 0, \ldots, 0]$, wobei die 1 an der Stelle j steht, $j = 1, \ldots, n$.

[63] Siehe Beweis von Satz 3.3!

Damit haben wir aber eine Idee, wie wir an unser Zerlegungsproblem herangehen können, nämlich:

> *Bestimme eine orthogonale Matrix $\widetilde{Q}$ bzw. eine (invertierbare) Linksdreiecksmatrix $\widetilde{L}$, so daß $\widetilde{Q}A = R$ bzw. $\widetilde{L}A = R$ jeweils eine Rechtsdreiecksmatrix liefert.*

Soviel hätten wir damit aber noch nicht gewonnen, wenn uns nicht die folgende Aussage zeigen würde, daß wir auch "Schritt für Schritt" vorgehen können:

1. *Das Produkt zweier orthogonaler Matrizen ist wieder eine orthogonale Matrix.*
2. *Das Produkt zweier Linksdreiecksmatrizen ist wieder eine Linksdreiecksmatrix.*

Also – der langen Rede kurzer Sinn – wird unsere Vorgehensweise nun darin bestehen, Folgen $Q_1, Q_2, \ldots$ bzw. $L_1, L_2, \ldots$ von orthogonalen bzw. invertierbaren Linksdreiecksmatrizen dergestalt zu konstruieren, daß die Matrizen

$$\underbrace{Q_1A}_{=:A_1},\ \underbrace{Q_2Q_1A}_{=:A_2=Q_2A_1},\ \underbrace{Q_3Q_2Q_1A}_{=:A_3=Q_3A_2},\cdots \qquad \text{bzw.} \qquad \underbrace{L_1A}_{=:A_1},\ \underbrace{L_2L_1A}_{=:A_2=L_2A_1},\ \underbrace{L_3L_2L_1A}_{=:A_3=L_3A_2},\cdots$$

immer "rechtsdreieckiger" werden und daß wir nach endlich vielen Schritten auch wirklich bei einer Rechtsdreiecksmatrix landen. Ist dann nämlich, nach sagen wir m Schritten,

$$R = A_m = \underbrace{Q_m \cdots Q_1}_{=\widetilde{Q}} A \qquad \text{bzw.} \qquad R = A_m = \underbrace{L_m \cdots L_1}_{=\widetilde{L}} A,$$

dann ist

$$A = \widetilde{Q}^T R =: QR \qquad \text{bzw.} \qquad A = \widetilde{L}^{-1} R =: LR$$

die gesuchte Zerlegung. In den nächsten beiden Abschnitten werden wir uns zwei Verfahren herleiten, die genau dieser Strategie folgen.

3.6 Direkte Verfahren I – QR–Zerlegung durch Rotationen

Anstatt eine allgemeine Theorie herzuleiten, beginnen wir mit einem einfachen Beispiel, nämlich dem einfachsten Matrixtypen, den es so gibt, einer 2×2–Matrix.

Beispiel 3.10 *Wir wollen*

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

durch Drehungen auf Rechtsdreiecksgestalt bringen. Wie Abb. 3.3 zeigt, sollte dies mit der

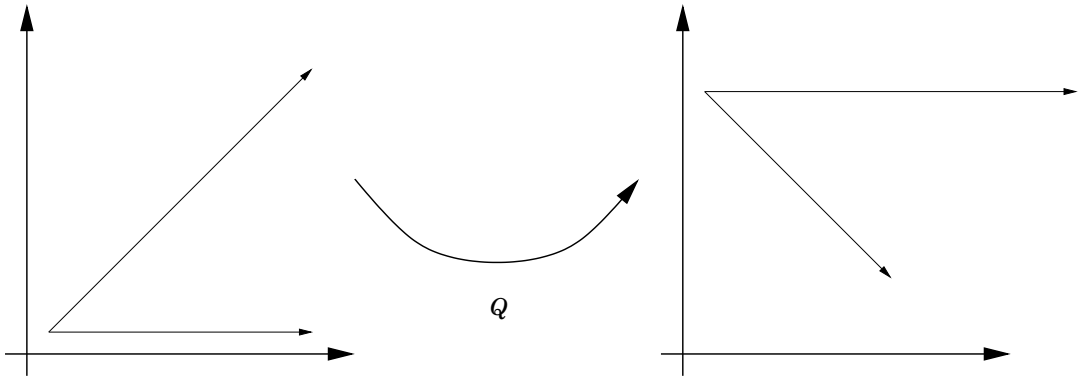


Abbildung 3.3: Die beiden Spaltenvektoren aus Beispiel 3.10. Offensichtlich bringt eine Drehung um $45^\circ = \pi/4$ den ersten Spaltenvektor $[1,1]^T$ auf ein Vielfaches des ersten Einheitsvektors $[1,0]^T$.

Matrix $Q = R_{\pi/4}$ möglich sein. Da $\sin\frac{\pi}{4} = \cos\frac{\pi}{4} = \sqrt{\frac{1}{2}}$ ist, können wir die geometrische Beobachtung nun auch arithmetisch nachprüfen und in der Tat ist

$$R_{\pi/4}A = \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \sqrt{2} & \sqrt{\frac{1}{2}} \\ 0 & -\sqrt{\frac{1}{2}} \end{bmatrix}}_{=:R},$$

also hat A die QR–Zerlegung

$$A = \underbrace{\begin{bmatrix} \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}}_{=:Q} \underbrace{\begin{bmatrix} \sqrt{2} & \sqrt{\frac{1}{2}} \\ 0 & -\sqrt{\frac{1}{2}} \end{bmatrix}}_{=:R}$$

Als nächstes machen wir uns das Leben ein bißchen schwerer und daher auch die Matrix ein wenig größer, denn der 3×3–Fall wird uns dann schon genug Ideen für das allgemeine Vorgehen liefern. Vorher aber müssen wir uns klar werden, welche Drehungen wir im $\mathbb{R}^3$ oder allgemein im $\mathbb{R}^n$ verwenden wollen, denn einen "Raumwinkel" kann man ja normalerweise nicht mehr mit nur einer Zahl beschreiben. Ist aber auch nicht schlimm: Anstatt "voll" im $\mathbb{R}^3$ zu rotieren, betrachten wir Rotationen in den *Koordinatenebenen*, die von den Einheitsvektoren e_1, e_2 bzw. e_1, e_3 oder e_2, e_3 aufgespannt werden, siehe Abb. 3.4. Allgemein, im $\mathbb{R}^n$, betrachtet man Rotationen in den Ebenen, die von *zwei*[64] Koordinatenvektoren, sagen wir e_j und e_k, $1 \leq$

[64]Hier sind die zwei- und die dreidimensionale Situation manchmal irreführend: Im $\mathbb{R}^2$ sind Ebenen und Vektoren ja im wesentlichen dasselbe, im $\mathbb{R}^3$ fallen immerhin noch "echte" Ebenen (die von *zwei* Vektoren aufgespannt werden) und *Hyperebenen* (die auf einen Vektor senkrecht stehen) zusammen. In mehr als drei Dimensionen gehen diese beiden Dinge aber auseinander, da eine Hyperebene im $\mathbb{R}^n$ von $n-1$ Vektoren aufgespannt wird.

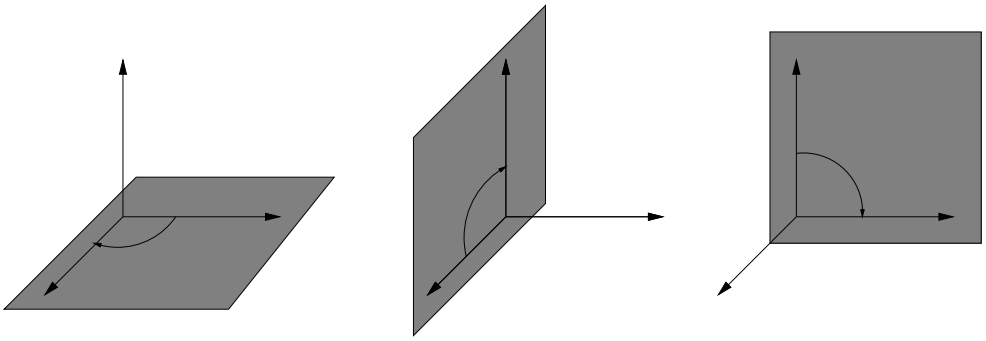


Abbildung 3.4: Rotationen in den Koordinatenebenen. Der Wert bezüglich der verbleibenden Koordinate wird, wie man sieht, einfach beibehalten.

$j < k \leq n$, aufgespannt werden. Die zugehörige Matrix hat dann die Gestalt

$$R_\phi[j,k] = \begin{array}{c|ccccccccc} & & & j & & & k & & & \\ \hline & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & 1 & & & & & & \\ j & & & & \cos\phi & & & \sin\phi & & \\ & & & & & 1 & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & 1 & & \\ k & & & & -\sin\phi & & & \cos\phi & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{array}$$

Beispiel 3.11 *Diesmal soll die Matrix*

$$A = \begin{bmatrix} \sqrt{\frac{1}{2}} & 1 & 0 \\ \sqrt{\frac{1}{2}} & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

auf Vordermann, das heißt auf Rechtsdreiecksgestalt gebracht werden, und zwar jetzt "Schritt für Schritt". Dafür beginnen wir mit der 2×2–Matrix oben links in A und drehen deren ersten Spaltenvektor

$$\begin{bmatrix} \boxed{\sqrt{\frac{1}{2}}} & 1 & 0 \\ \boxed{\sqrt{\frac{1}{2}}} & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} \end{bmatrix}$$

auf ein Vielfaches des ersten Einheitsvektors. Das kennen wir aber bereits aus Beispiel 3.10, denn das ist die Drehung $R_{\pi/4}$, jetzt aber in der (e_1, e_2)–Ebene, also die Drehung $R_{\pi/4}[1,2]$. Und in der Tat ist

$$A_1 = R_{\pi/4}\,[1,2]\; A = \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & 0 \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{2}} & 1 & 0 \\ \sqrt{\frac{1}{2}} & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 1 & 0 & 0 \end{bmatrix}$$

Na also! Eine Null haben wir schon in der ersten Spalte. Schauen wir also, daß wir die 1 unten links auch noch loswerden können, indem wir den Vektor

$$\begin{bmatrix} \boxed{1} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \boxed{1} & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

ebenfalls auf ein Vielfaches von e_1 bringen – nun kein Problem mehr für uns, denn das ist wieder eine Drehung um $\pi/4$, aber natürlich in der (e_1, e_3)–Ebene. Rechnen wir's nach:

$$A_2 = R_{\pi/4}[1,3]\, A_1 = \begin{bmatrix} \sqrt{\frac{1}{2}} & 0 & \sqrt{\frac{1}{2}} \\ 0 & 1 & 0 \\ -\sqrt{\frac{1}{2}} & 0 & \sqrt{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 1 & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix},$$

und "schon" hat die erste Spalte Rechtsdreiecksgestalt. Schaffen wir es jetzt also noch, einen Winkel ϕ zu finden, der den Vektor

$$\begin{bmatrix} \sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \boxed{-\sqrt{\frac{1}{2}}} & \sqrt{\frac{1}{2}} \\ 0 & \boxed{-\frac{1}{2}} & -\frac{1}{2} \end{bmatrix} \to \begin{bmatrix} -\sqrt{\frac{1}{2}} \\ -\frac{1}{2} \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix}$$

auf den ersten Einheitsvektor dreht, dann ist

$$R_\phi[2,3]\, A_2 = R_\phi[2,3]\, R_{\pi/4}[1,3]\, R_{\pi/4}[1,2]\, A = R$$

eine Rechtsdreiecksmatrix und damit

$$A = \underbrace{R_{-\pi/4}[1,2]\, R_{\pi/4}[1,3]\, R_\phi[2,3]}_{=:Q}\, R = QR.$$

Was aber jetzt natürlich noch offen ist, ist die Bestimmung des "magischen" Drehwinkels ϕ für den Vektor $-\frac{1}{2}\left[\sqrt{2}, 1\right]^T$. Wenn wir aber mal kurz nachdenken[65], dann stellen wir fest, daß wir für unsere Drehung gar nicht den *Winkel* ϕ benötigen, sondern nur die beiden Zahlen $c = \cos\phi$ und $s = \sin\phi$ und diese bekommen wir aber relativ billig, wie wir gleich sehen

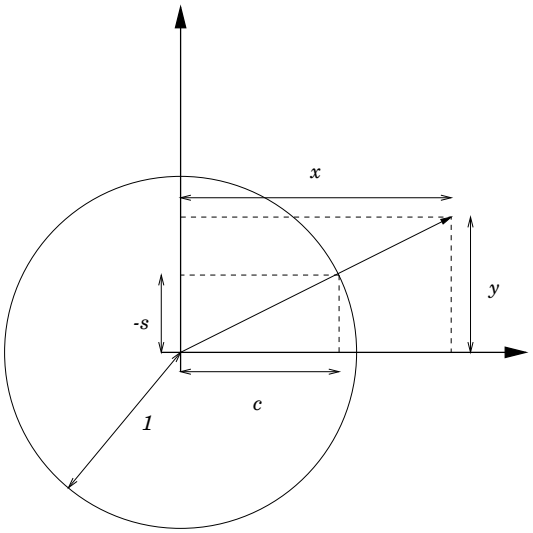


Abbildung 3.5: Bestimmung des Drehwinkels. Anstatt den Vektor $[x, y]^T$ *im Uhrzeigersinn* auf ein Vielfaches des Einheitsvektors zu rotieren, können wir genauso den Einheitsvektor $[1, 0]^T$ *gegen den Uhrzeigersinn* auf ein Vielfaches dieses Vektors rotieren – den Sinus und den Cosinus können wir dann ganz einfach von der normierten Version von x ablesen.

werden. Und um das Problem ein für alle Male zu lösen, sehen wir uns das gleich für einen *beliebigen* Vektor $[x, y]^T \in \mathbb{R}^2$ an. Betrachten wir nämlich Abb. 3.5, dann sehen wir daß unsere "gesuchte" Drehung R_ϕ mit der Eigenschaft

$$R_\phi \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \lambda = \sqrt{x^2 + y^2},$$

die Beziehung

$$\frac{1}{\lambda} \begin{bmatrix} x \\ y \end{bmatrix} = R_{-\phi} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix}$$

erfüllt. Also erhalten wir die Bestimmungsformeln

$$c = \cos\phi = \frac{x}{\sqrt{x^2 + y^2}} \qquad \text{und} \qquad s = \sin\phi = \frac{y}{\sqrt{x^2 + y^2}}. \tag{3.18}$$

Das wenden wir gleich einmal auf den Vektor $-\frac{1}{2}\left[\sqrt{2}, 1\right]$, für den

$$\sqrt{x^2 + y^2} = \frac{1}{2}\sqrt{2 + 1} = \frac{\sqrt{3}}{2}$$

[65] Dies ist in der Mathematik nicht verboten, noch nicht einmal unerwünscht.

ist und deswegen

$$c = \frac{2}{\sqrt{3}}\,\frac{-\sqrt{2}}{2} = -\sqrt{\frac{2}{3}} \qquad \text{und} \qquad s = \frac{2}{\sqrt{3}}\,\frac{-1}{2} = -\sqrt{\frac{1}{3}}.$$

Um zu sehen ob und daß das auch stimmt bekommen wir jetzt also

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{3}} \\ 0 & \sqrt{\frac{1}{3}} & -\sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \sqrt{\frac{3}{4}} & -\sqrt{\frac{1}{12}} \\ 0 & 0 & \sqrt{\frac{2}{3}} \end{bmatrix},$$

und siehe da – die Rechtsdreiecksmatrix ist erfolgreich konstruiert.
Damit ist aber die "Strategie", die Matrix A durch Drehungen auf Rechtsdreiecksgestalt zu bringen, komplett:

Die sukzessive Bestimmung von Drehwinkeln

$$\begin{array}{ccccc} \phi_{12} & \phi_{13} & \cdots & \phi_{1,n-1} & \phi_{1n} \\ & \phi_{23} & \cdots & \phi_{2,n-1} & \phi_{2n} \\ & & \ddots & \vdots & \vdots \\ & & & \phi_{n-2,n-1} & \phi_{n-2,n} \\ & & & & \phi_{n-2,n} \end{array}$$

nach (3.18) liefert uns Rotationen

$$R_{\phi_{jk}}[j,k], \qquad k = j+1, \ldots, n, \quad j = 1, \ldots, n-1,$$

die Spalte für Spalte und innerhalb jeder Spalte Zeile für Zeile die "störenden" Elemente[66] *der Matrix A eliminieren und am Schluß bleibt also eine Rechtsdreiecksmatrix*

$$R = R_{\phi_{n-1,n}}[n-1,n] \cdots R_{\phi_{1n}}[1,n] \cdots R_{\phi_{12}}[1,n]\ A$$

stehen, so daß wir die QR–Zerlegung

$$A = \underbrace{R_{-\phi_{12}}[1,2] \cdots R_{-\phi_{1n}}[1,n] \cdots R_{-\phi_{n-1,n}}[n-1,n]}_{=:Q} R$$

erhalten.

Ein kleiner Punkt am Rande bleibt noch: Was machen wir, wenn in (3.18) durch Null geteilt werden würde, wenn also $\sqrt{x^2+y^2} = 0$, also $x = y = 0$ ist? Antwort: Gar nichts! Denn schließlich besteht unser Ziel bei der Elimination ja darin, y auf Null zu drehen und wenn y schon den Wert Null hat, umso besser.

Dieses Verfahren zur Bestimmung der QR–Zerlegung einer Matrix A wird *Jacobi–Verfahren* oder *QR–Zerlegung mittels Jacobi–Rotationen* genannt. Es geht tatsächlich auf Jacobi[67] [5]

```
%%  JacobiQR.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Berechne QR-Zerlegung mittels Jacobirotationen
%%  Ueberschreiben der Matrix A
%%  Eingabe:
%%    A    Matrix

function [Q,R] = JacobiQR( A )
  n = length( A );
  Q = eye( n );

  for j = 1:n-1
    for k = j+1:n
      Qjk = RotMat( [ A( j,j ), A( k,j ) ],j,k,n );
      A = Qjk * A;
      Q = Q * Qjk';
    end
  end

  R = A;
```

Programm 3.4 `JacobiQR.m`: Berechnung der QR–Zerlegung mit Jacobi–Rotationen. Die drei Schritte in der Schleife sind: Berechnung der Rotationsmatrix, Elimination in A durch Multiplikation von links und schließlich "Update" des Kandidaten für Q. Diese Implementation ist übrigens zwar einfach aber ineffizient.

```
%%  RotMat.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Generiere Rotationsmatrix
%%  Eingabe:
%%    x    2x2 Vektor
%%    j,k  Indizes
%%    n    Groesse

function R = RotMat( x,j,k,n )
  R = eye( n );
  l = sqrt( x( 1 )^2 + x( 2 )^2 );

  %% Pruefe ob ungleich Null
  if ( abs(l) > eps )
    R( j,j ) = R( k,k ) = x( 1 ) / l;
    R( j,k ) = x( 2 ) / l;
    R( k,j ) = -x( 2 ) / l;
  end
```

Programm 3.5 `RotMat.m`: Bestimmung der Rotationsmatrix, Hilfsprogramm.

zurück und wird von ihm Titel der Arbeit ausdrücklich als ein *"leichtes Verfahren"* bezeichnet. In Programm 3.4 ist diese Berechnung der QR–Zerlegung dargestellt (eigentlich wirklich ein leichtes Verfahren), wobei im Programm 3.5, `RotMat.m` die Aufgabe erledigt wird, eine passende Rotationsmatrix (nicht aber der Winkel) zu bestimmen.

3.7 Direkte Verfahren II – LR–Zerlegung durch Elimination

In diesem Abschnitt werden wir uns nun mit dem Problem herumschlagen, wie man eine Matrix mit Hilfe von Linksdreiecksmatrizen auf Rechtsdreiecksgestalt bringen kann. Dabei wird die Bestimmung der Linksdreiecksmatrix auch noch sehr einfach sein!

Nun, was sind, nach den Diagonalmatrizen, besonders einfache untere Dreiecksmatrizen? Beispielsweise doch die, die nur in einer einzigen Spalte, sagen wir in Spalte j, einen von Null

[66] Das sind also die Elemente, die unterhalb der Diagonalen stehen.

[67] Carl Gustav Jacob Jacobi, 10.10.1804–18.2.1851, Zeitgenosse von Gauß und sicherlich einer der bedeutendsten Mathematiker seiner Zeit, sowohl in theoretischer als auch in angewandter Hinsicht.

verschiedenen Eintrag haben, also Matrizen der Form

$$M_j[y] = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & y_{j+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & y_n & & & 1 \end{bmatrix} = I + ye_j^T, \qquad y_1 = \cdots = y_j = 0. \tag{3.19}$$

Dabei erinnern wir uns, daß für $x, y \in \mathbb{R}^n$ mit xy^T die Matrix $[y_j x_k \ : \ j,k = 1,\ldots,n]$ bezeichnet wird, also ist insbesondere für $x = e_j$

$$ye_j^T = \begin{bmatrix} y_1 (e_j)_1 & \ldots & y_1 (e_j)_j & \ldots & y_1 (e_j)_n \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_n (e_j)_1 & \ldots & y_n (e_j)_j & \ldots & y_n (e_j)_n \end{bmatrix} = \begin{bmatrix} 0 & \ldots & 0 & y_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & y_n & 0 & \ldots & 0 \end{bmatrix}.$$

Ist nun $A = [a_1,\ldots,a_n] \in \mathbb{R}^{n\times n}$ eine Matrix, geschrieben als Zeilenvektor der Spaltenvektoren $a_1,\ldots,a_n \in \mathbb{R}^n$, dann ist

$$e_j^T A = \left[e_j^T a_1, \ldots, e_j^T a_n\right] = [a_{j1}, a_{j2}, \ldots, a_{jn}] = \bar{a}_j,$$

wobei nun $\bar{a}_j$ wieder den $j$ten *Zeilenvektor* von A bezeichnet. Damit ist aber für $y \in \mathbb{R}^n$

$$ye_j^T A = \begin{bmatrix} y_1 a_{j1} & \ldots & y_1 a_{jn} \\ \vdots & \ddots & \vdots \\ y_n a_{j1} & \ldots & y_n a_{j1} \end{bmatrix} = \begin{bmatrix} y_1\, \bar{a}_j \\ \vdots \\ y_n\, \bar{a}_j \end{bmatrix}$$

was für die Matrix $M_j[y]$ aus (3.19)[68]

$$M_j[y]\, A = \left(I + ye_j^T\right) A = A + ye_j^T A = \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_j \\ \bar{a}_{j+1} + y_{j+1}\bar{a}_j \\ \vdots \\ \bar{a}_n + y_n \bar{a}_j \end{bmatrix}$$

liefert.

Definition 3.12 *Die Matrizen $M_j[y]$ aus (3.19) heißen* Gauß–Transformationen.

[68] Nicht vergessen: der Index j bedeutet insbesondere, daß die ersten j Komponenten von y den Wert Null haben müssen!

Als nächstes ein paar Bemerkungen zu den Gauß–Transformationen:

1. Die Multiplikation einer Gauß–Transformation $M_j[y]$ von links an eine Matrix A bedeutet, daß man das y_k–fache der Zeile $\bar{a}_j$ zur k–ten Zeile, $\bar{a}_k$, addiert, und das für alle $k = j+1, \ldots, n$. Eine Zeile wird also nur zu Zeilen addiert, die *später* kommen.

2. Dieses Verfahren kann man auch so interpretieren, daß man sagt, daß sich die Gültigkeit von Gleichungen nicht ändert, wenn man ein Vielfaches einer Gleichung zu einer anderen addiert ("Gleich plus gleich ist gleich"), solange man dies nur auch mit der rechten Seite macht.

3. Die Inverse einer Gauß–Transformation ist wieder eine Gauß–Transformation und zwar ist $M_j^{-1}[y] = M_j[-y]$, denn

$$M_j[y]\, M_j[-y] = \left(I + ye_j^T\right)\left(I - ye_j^T\right) = I - \underbrace{I\, ye_j^T}_{=ye_j^T} + \underbrace{ye_j^T\, I}_{=ye_j^T} + y\, \underbrace{e_j^T y}_{=y_j=0}\, e_j^T = I.$$

Wie aber können wir die Gauß–Transformationen nun verwenden, um eine Zerlegung, in diesem Fall also eine LR–Zerlegung unserer Matrix A zu bestimmen? Nun, indem wir zuerst mal einen Vektor y^1 so bestimmen, daß

$$A_1 = M_1\left[y^1\right]\ A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ 0 & * & \ldots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \ldots & * \end{bmatrix}$$

ist, denn dann sieht ja die erste Spalte von A_1 schon so aus, wie wir es gerne hätten. Außerdem sind die Werte $y_2^1, \ldots, y_n^1$ ja einfach zu bestimmen[69], nämlich über

$$0 = (A_1)_{j1} = a_{j1} + y_j^1 a_{11} \qquad \Longrightarrow \qquad y_j^1 = -\frac{a_{j1}}{a_{11}}, \qquad j = 2, \ldots, n. \tag{3.20}$$

Bleibt nur ein Problem: Was tun, wenn $a_{11} = 0$? Nun, auch nicht so wild, dann suchen wir uns eben eine Zeile j, für die $a_{1j} \neq 0$ ist und vertauschen die Zeilen 1 und j. Und wenn das auch nicht geht? Nun, dann ist die erste Spalte von A der Nullvektor und dann ist die Matrix A ohnehin nicht invertierbar und unser Gleichungssystem nicht eindeutig lösbar – warum sich also weiter damit beschäftigen. Und wenn wir schon dabei sind: Wir erinnern uns, daß Division durch kleine Zahlen normalerweise nicht so erwünscht ist, weil dies ja auch kleine Fehler verstärken kann, deswegen durchsucht man am besten gleich die gesamte erste Spalte und vertauscht die erste Zeile mit derjenigen Zeile, in der das *betragsgrößte* Element der Spalte steht, so daß also dann, nach der Vertauschung,

$$|a_{11}| = \max_{k=1,\ldots,n} |a_{1k}| \qquad \Longrightarrow \qquad \left|y_k^1\right| = \frac{|a_{1j}|}{|a_{11}|} \leq 1$$

[69] Und außerdem is ja immer noch $y_1^1 = 0$!

gilt. Diese Vorgehensweise bezeichnet man als *Spaltenpivotsuche*. Natürlich muß man über diese Vertauschungsoperationen ordentlich Buch führen, denn am Ende bestimmt man ja nicht die LR–Zerlegung der Matrix A sondern einer Matrix $\widetilde{A}$, die zwar dieselben Zeilen wie A enthält, möglicherweise aber in vertauschter Reihenfolge und diese Vertauschung müssen wir selbstverständlich auch auf die rechte Seite unseres Gleichungssystems anwenden.

Aber zurück zu unserer eigentlichen Gauß–Elimination. Der nächste Schritt ist unschwer zu erraten: Nun bestimmen wir einen Vektor y^2, so daß

$$M_2\left[y^2\right]\ A_1 = \begin{bmatrix} * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \dots & * \end{bmatrix}$$

wird und erhalten jetzt natürlich[70], daß

$$y_1^2 = y_2^2 = 0, \qquad y_j^2 = -\frac{(A_1)_{j2}}{(A_1)_{22}}, \quad j = 3, \dots, n,$$

möglicherweise wieder nach einer Zeilenvertauschung. Und so weiter, und so fort. Nach $n-1$ Eliminationsschritten erhalten wir auf diese Art und Weise schließlich eine Rechtsdreiecksmatrix $R = A_{n-1}$ und die Zerlegung $A = LR$, wobei

$$\begin{aligned} L &= \left(M_{n-1}\left[y^{n-1}\right] \cdots M_1\left[y^1\right]\right)^{-1} = M_1^{-1}\left[y^1\right] \cdots M_{n-1}^{-1}\left[y^{n-1}\right] \\ &= \left(I - y^1 e_1^T\right)\left(I - y^2 e_2^T\right) \cdots \left(I - y^{n-1} e_{n-1}^T\right) \\ &= \Big(I - y^1 e_1^T - y^2 e_2^T + \underbrace{y^1 e_1^T y^2 e_2^T}_{=y_1^2=0}\Big)\left(I - y^3 e_3^T\right) \cdots \left(I - y^{n-1} e_{n-1}^T\right) \\ &= I - y^1 e_1^T - \cdots - y^{n-1} e_{n-1}^T = \begin{bmatrix} 1 & & & \\ -y_2^1 & 1 & & \\ \vdots & \ddots & \ddots & \\ -y_n^1 & \dots & -y_n^{n-1} & 1 \end{bmatrix}. \end{aligned}$$

Anders gesagt: Für die Linksdreiecksmatrix L müssen wir noch nicht einmal etwas rechnen, denn wir erhalten sie ganz einfach dadurch, daß wir die "echten" Koeffizienten der Vektoren $y^1, \dots, y^{n-1}$ aufsammeln.

Beispiel 3.13 *Es sei*

$$A = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 1 & 3 & 3 & 2 \\ 2 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

[70] Lernzielkontrolle ...

Ohne Pivotsuche erhalten wir, daß

$$A = \begin{bmatrix} \boxed{\boxed{1}} & 2 & 2 & 1 \\ \boxed{1} & 3 & 3 & 2 \\ \boxed{2} & -1 & 0 & 1 \\ \boxed{0} & 1 & 0 & 1 \end{bmatrix} \qquad \Longrightarrow \qquad y^1 = -\frac{1}{\boxed{\boxed{1}}} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -2 \\ 0 \end{bmatrix}$$

und damit ist

$$A_1 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 0 & \boxed{\boxed{1}} & 1 & 1 \\ 0 & \boxed{-5} & -4 & -1 \\ 0 & \boxed{1} & 0 & 1 \end{bmatrix} \qquad \Longrightarrow \qquad y^2 = -\frac{1}{1} \begin{bmatrix} 0 \\ 0 \\ -5 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 5 \\ -1 \end{bmatrix},$$

also

$$A_2 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & \boxed{\boxed{1}} & 4 \\ 0 & 0 & \boxed{-1} & 0 \end{bmatrix} \qquad \Longrightarrow \qquad y^3 = -\frac{1}{1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Das liefert uns dann auch schon unsere Zerlegung

$$A = LR = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 2 & -5 & 1 & \\ 0 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 & 1 \\ & 1 & 1 & 1 \\ & & 1 & 4 \\ & & & 4 \end{bmatrix},$$

wobei die "strukturellen" Nulleinträge weggelassen wurden.

Und weil's so schön war, machen wir das Ganze nochmal, diesmal mit *Pivotsuche. In der Tat erhalten wir nun aus*

$$A = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 1 & 3 & 3 & 2 \\ \boxed{2} & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} \boxed{\boxed{2}} & -1 & 0 & 1 \\ \boxed{1} & 3 & 3 & 2 \\ \boxed{1} & 2 & 2 & 1 \\ \boxed{0} & 1 & 0 & 1 \end{bmatrix} \quad \Longrightarrow \quad y^1 = -\frac{1}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{2} \\ -\frac{1}{2} \\ 0 \end{bmatrix},$$

daß

$$A_1 = \begin{bmatrix} 2 & -1 & 0 & 1 \\ 0 & \boxed{\boxed{\frac{7}{2}}} & 3 & \frac{3}{2} \\ 0 & \boxed{\frac{5}{2}} & 2 & \frac{1}{2} \\ 0 & \boxed{1} & 0 & 1 \end{bmatrix} \quad \Longrightarrow \quad y^2 = -\frac{2}{7} \begin{bmatrix} 0 \\ 0 \\ \frac{5}{2} \\ \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\frac{5}{7} \\ -\frac{3}{7} \end{bmatrix}$$

und damit

$$A_2 = \begin{bmatrix} 2 & -1 & 0 & 1 \\ 0 & \frac{7}{2} & 3 & \frac{3}{2} \\ 0 & 0 & -\frac{1}{7} & -\frac{4}{7} \\ 0 & 0 & \boxed{-\frac{6}{7}} & \frac{4}{7} \end{bmatrix} \to \begin{bmatrix} 2 & -1 & 0 & 1 \\ 0 & \frac{7}{2} & 3 & \frac{3}{2} \\ 0 & 0 & \boxed{\boxed{-\frac{6}{7}}} & \frac{4}{7} \\ 0 & 0 & \boxed{-\frac{1}{7}} & -\frac{4}{7} \end{bmatrix} \quad \Longrightarrow \quad y^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{6} \end{bmatrix},$$

also

$$L = \begin{bmatrix} 1 & & & \\ \frac{1}{2} & 1 & & \\ \frac{1}{2} & \frac{5}{7} & 1 & \\ 0 & \frac{3}{7} & \frac{1}{6} & 1 \end{bmatrix} \qquad \textit{und} \qquad R = \begin{bmatrix} 2 & -1 & 0 & 1 \\ & \frac{7}{2} & 3 & \frac{3}{2} \\ & & -\frac{6}{7} & \frac{4}{7} \\ & & & -\frac{2}{3} \end{bmatrix},$$

wobei nun

$$LR = \widetilde{A} = \begin{bmatrix} 2 & -1 & 0 & 1 \\ 1 & 3 & 3 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix}.$$

Bemerkung 3.14 *Eigentlich wird Spaltenpivotsuche gerne dadurch motiviert, daß sie es schafft, die auftretenden Rundungsfehler (Rückwärtsfehler) geringer zu halten, indem die Matrix L "kleingehalten" wird: Wie bereits erwähnt, führt ja Spaltenpivotsuche dazu, daß alle Einträge von L betragsmäßig ≤ 1 sind. Trotzdem ist im obigen Beispiel die Gauß–Elimination* ohne *Pivotsuche genauer, da dort alle Operationen in Ganzzahlarithmetik, also* exakt *ausgeführt werden.*

Trotzdem, jetzt erst einmal der "theoretische Unterbau", der aus unserer Gauß–Elimination folgt.

Satz 3.15 *Ist $A \in \mathbb{R}^{n\times n}$ eine invertierbare Matrix, dann gibt es eine Matrix $\widetilde{A} \in \mathbb{R}^{n\times n}$, die aus A durch Zeilenvertauschungen hervorgeht sowie eine Linksdreiecksmatrix $L \in \mathbb{R}^{n\times n}$ und eine Rechtsdreiecksmatrix $R \in \mathbb{R}^{n\times n}$, so daß*

$$\widetilde{A} = LR \qquad \textit{und} \qquad \ell_{jj} = 1, \quad j = 1, \ldots, n.$$

Übung 3.9 Implementieren Sie, unter Verwendung von Programm 3.6 die Gauß–Elimination mit Spaltenpivotsuche.

3.8 Größe von Vektoren und Matrizen und Fehleraussagen

In diesem Abschnitt wollen wir uns kurz ansehen, welche Aussagen man über den (relativen) Fehler beim Lösen von Gleichungssystemen via Gauß–Elimination machen kann und vor allem, von welchen Parametern der relative Fehler abhängt. Dafür brauchen wir aber zuerst einen Begriff der "Größe" von Vektoren und das führt uns zu *Normen*.

```
%%  Gauss.m (Numerik HaF)
%%  ---------------------------------------------------------
%%  Berechne LR-Zerlegung mittels Gauss-Elimination
%%  ohne Pivotsuche, "Hardcore Matlab"
%%  Ueberschreiben der Matrix A
%%  Eingabe:
%%    A    Matrix

function [L,R] = Gauss( A )
  n = length( A );
  L = eye( n );

  for j = 1:n-1
    y = -A( j+1:n,j ) / A( j,j );
    A( j+1:n,j ) = 0;
    A( j+1:n,j+1:n ) = A( j+1:n,j+1:n ) + y * A( j,j+1:n );
    L( j+1:n, j ) = -y;
  end
  R = A;
```

Programm 3.6 `Gauss.m`: Gauß–Elimination *ohne* Pivotsuche, in beschleunigtem "Hardcore–`Octave`" ausgeführt, das heißt, alle Operationen sind *vektorisiert* und enthalten so wenige Schleifen wie möglich.

Beispiel 3.16 *Die* euklidische Länge *eines Vektors $x \in \mathbb{R}^n$ bestimmt sich als*

$$\|x\|_2 := \sqrt{\sum_{j=1}^{n} x_j^2},$$

was nichts anderes als eine mehrfache Anwendung des Satzes von Pythagoras ist. Sehen wir uns das mal im $\mathbb{R}^3$ mit einem Vektor der Form $[x, y, z]^T$ an. Die Länge dieses Vektors ist die

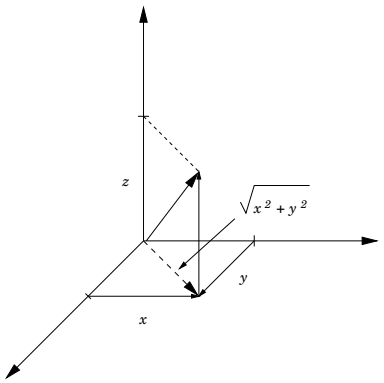


Abbildung 3.6: Die Bestimmung der euklidischen Norm des Vektors $[x, y, z]^T$, indem zuerst die Länge des Vektors in der (x, y)–Ebene bestimmt wird und dann daraus die Länge des Vektors $[(x, y), z]$.

Länge der Hypothenuse des rechtwinkligen Dreiecks, das von $[0, x]^T$ und $[0, y]^T$ gebildet wird und die Länge des Gesamtvektors ist dann

$$\sqrt{\left(\sqrt{x^2+y^2}\right)^2 + z^2} = \sqrt{x^2+y^2+z^2}.$$

Eine Verallgemeinerung dieses Längenbegriffs führt zur *axiomatischen* Definition des Begriffs "Norm".

Definition 3.17 *Eine Abbildung $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ heißt* Norm*, wenn sie die folgenden Bedingungen erfüllt:*

1. *Alle Vektoren $\neq 0$ haben positive Länge*[71]*:*

$$\|x\| \geq 0 \qquad \textit{und} \qquad \|x\| = 0 \iff x = 0.$$

2. *Streckt man einen Vektor, so streckt man die Länge*[72]*:*

$$\|cx\| = |c|\,\|x\|, \qquad c \in \mathbb{R}.$$

[71] Fehlt die Eigenschaft, daß nur der Nullvektor Länge 0 haben darf, dann spricht man von einer *Halbnorm*.
[72] Der hochtrabende Begriff hier ist *positive Homogenität*.

3. *Der kürzeste Weg ist immer der direkte Weg*[73]*:*

$$\|x + y\| \leq \|x\| + \|y\|.$$

Übung 3.10 Zeigen Sie, daß die euklidische Norm alle drei Axiome aus Definition 3.17 erfüllt.

Beispiel 3.18 *Außer der (geometrisch einleuchtenden) euklidischen Norm spielen noch die "Manhattan–Norm"*

$$\|x\|_1 = \sum_{j=1}^{n} |x_j|$$

und die "Maximumsnorm"

$$\|x\|_\infty = \max_{j=1,\ldots,n} |x_j|$$

eine wichtige Rolle. Sie sind, wie auch die euklidische Norm Spezialfälle der sogenannten p–Normen

$$\|x\|_p = \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p}, \qquad 1 < p < \infty.$$

Definition 3.19 *Die* p–Operatornorm *einer Matrix* $A \in \mathbb{R}^{n\times n}$ *ist definiert als "maximaler Expansionsfaktor"*

$$\|A\|_p = \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \tag{3.21}$$

und die p–Konditionszahl *für eine* invertierbare *Matrix* A *als*

$$\kappa_p(A) = \|A\|_p \left\|A^{-1}\right\|_p. \tag{3.22}$$

Die Konditionszahlen $\kappa_p(A)$ sind tatsächlich Konditionszahlen im Sinne von Definition 2.15, das heißt ein Maß für das linearisierte Verhalten der Abbildung $A \mapsto A^{-1}$ bei kleinen Störungen von A. Nun ist die Konditionszahl in (3.22) ja nur für *invertierbare* Matrizen definiert! Erinnern wir uns aber[74], daß wir jede nichtinvertierbare Matrix mit einer beliebig kleinen Störung in eine invertierbare Matrix verwandeln können, dann kann man auch die Konditionszahl nichtinvertierbarer ("singulärer") Matrizen über einen solchen Grenzprozess bestimmen[75] und erhält:

> *Eine Matrix ist genau dann invertierbar, wenn* $\kappa_p(A) < \infty$ *für eine beliebiges* $1 \leq p \leq \infty$.

[73] Diese Ungleichung bezeichnet man auch als *Dreiecksungleichung*.

[74] Siehe Beispiel 3.2.

[75] Was man dabei natürlich auch nachweisen muß ist, daß *jeder* derartige Grenzprozess *denselben* Grenzwert liefert!

In gewissem Sinne ist die Konditionszahl also auch ein Maßstab für die *Invertierbarkeit* einer Matrix und zwar ein wichtiger, vor allem in der Welt der numerischen Mathematik wo durch die allgegenwärtigen Rundungsfehler fast alle Matrizen "ein bißchen invertierbar" werden, aber dafür eben eine dramatische Konditionszahl haben.

Eine typische Fehleraussage für die Lösung von linearen Gleichungssystemen mittels Gauß–Elimination, Vorwärtselimination und Rücksubstitution[76] hat, für die $\|\cdot\|_\infty$–Norm die folgende Gestalt.

Satz 3.20 *Ist $A \in \mathbb{R}^{n\times n}$ invertierbar und ist*

$$\hat{u} < \frac{1}{\kappa_\infty(A)}, \tag{3.23}$$

dann erfüllt die mit Gauß–Elimination[77], Vorwärtselimination und Rücksubstitution[78] berechnete Lösung $\hat{x}$ die Ungleichung

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leq 6n^3\hat{u}\,\frac{\gamma(A)}{1-\hat{u}\,\kappa_\infty(A)}\,\kappa_\infty(A), \tag{3.24}$$

wobei der Wachstumsfaktor $\gamma(A)$ *der Quotient aus dem betragsgrößten Eintrag in einer der Eliminationsmatrizen $A, A_1, \ldots$ und dem betragsgrößten Eintrag von A selbst ist.*

Keine Angst – weder werden wir diesen Satz beweisen, was nun doch etwas zu weit führen würde, noch muß man derartige Formeln auswendig können. Vielmehr sollte man versuchen, zu verstehen, was einem derartige Formeln "sagen" wollen:

1. Die Bedingung (3.23) verknüpft die Rechengenauigkeit mit der Konditionszahl der Matrix und erscheint zuerst einmal wie eine "technische" Bedingung, das heißt, eine Voraussetzung, die man an einer Stelle im Beweis benötigt, weil man sonst nicht weiterkommt. Das wäre aber wesentlich weniger als die halbe Wahrheit, denn (3.23) sagt uns, wann ein Problem mit der vorhandenen Rechengenauigkeit nicht mehr angegangen werden sollte. Wie wichtig das in der Praxis ist, erkennt man schon daran, daß `Matlab` und `Octave` *automatisch* beim Lösen eines Gleichungssystems die Konditionszahl der Matrix abschätzen[79] und eine Warnung ausgeben, wenn (3.23) verletzt ist.

2. Ist darüberhinaus $\hat{u}\,\kappa_\infty(A) \sim 1$, kann schon der Bruch in (3.24) beliebig groß werden und wir erhalten eine beliebig unsinnige Abschätzung. Es ist zwar nur eine obere Abschätzung, aber leider nicht immer eine Überschätzung.

[76] Nicht vergessen: Der Rückwärtsfehler ist ein Verfahrensfehler und hängt somit von der verwendeten Lösungsmethode ab.

[77] Zerlegung in Dreiecksmatrizen

[78] Lösung der Dreieckssysteme

[79] Eine "exakte" Berechnung wäre genauso aufwendig wie das Lösen des Gleichungssystems, der Aufwand bei den Abschätzungen hingegen ist für große Matrizen vernachlässigbar gegenüber dem Aufwand für das Lösungsverfahren.

3. Das Auftreten der Konditionszahl als Faktor in (3.24) sollte uns nicht überraschen, wenn wir uns an die "Theorie" von Rückwärtsfehler und Konditionszahl erinnern.

4. Der Wachstumsfaktor $\gamma(A)$ ist eine *a posteriori*–Größe, die theoretisch kompliziert, praktisch aber einfach zu ermitteln ist – man sucht "lediglich" nach jedem Eliminationsschritt den betragsgrößten Eintrag in der erhaltenen Matrix. Auch wenn der Wachstumsfaktor *dramatisch* groß werden kann, nämlich bis zu 2^{n-1} für (ziemlich spezielle) $n \times n$–Matrizen, so gibt es doch praxisrelevante Matrizen, bei denen er "nur" wie n wächst oder sogar von n unabhängig beschränkt werden kann.

Also daß Rechnen ein fundament und grundt aller Künste ist / Dann ohne zal mag kein Musicus seinen Gesang / kein Geometer sein Mensur vollbringe / auch kein Astronomus den lauff des Himmels erkennen.

Adam Riese [10]

4 Iterative Lösungsverfahren für Gleichungen

Jetzt beschäftigen wir uns mit einem anderen Typ von Lösungsmethoden, bei denen nicht in einer von vornherein absehbaren endlichen Anzahl von Rechenschritten[80] ein Ergebnis bestimmt wird, das, könnte man exakt rechnen, auch exakt wäre, sondern wir betrachten jetzt Verfahren, bei denen man immer neue "geratene" Werte bestimmt, die sich hoffentlich der Lösung mehr und mehr annähern. Als *"iterativ"* bezeichnet man diese Verfahren, wenn man dieselbe Operation stur immer und immer wieder durchführt ("iteriert"), bis die gewünschte Genauigkeit der Lösung erreicht ist[81], oder aber der Abschreibungstermin des Computers erreicht wird.

Da wir aber auch bei Algorithmen ohnehin nicht exakt rechnen[82], sondern alle Rechenschritte mit Rundungsfehlern behaftet sind und daher das Ergebnis sowieso gestört ist, muß die Tatsache, daß wir der Lösung nicht "berechnen", sondern nur "annähern", an sich noch kein Nachteil sein.

Ein *iteratives Verfahren* beruht nun auf einer *Iterationsfunktion* $F : \mathbb{R}^n \to \mathbb{R}^n$ und bestimmt eine Folge von Werten $x^{(j)} \in \mathbb{R}^n$, $j \in \mathbb{N}$, durch die Vorschrift

$$x^{(j+1)} = F\left(x^{(j)}\right), \qquad j \in \mathbb{N}_0, \tag{4.1}$$

aus einem *Startwert* $x^{(0)}$. Was folgt nun aus "Konvergenz" gegen einen *Grenzwert* x^*? Anschaulich gesprochen bedeutet das ja, daß für "hinreichend großes" j wir immer $x^{(j)} \sim x^*$ haben, wobei der Fehler "beliebig klein" werden kann. Ist nun j "groß genug", dann ist ja wohl $j + 1$ erst recht "groß genug" und damit liefert (4.1), daß

$$x^* \sim x^{(j+1)} = F\left(x^{(j)}\right) \sim F\left(x^*\right) \qquad \Longrightarrow \qquad x^* = F\left(x^*\right),$$

die "Lösung" x^* ist also ein sogenannter[83] *Fixpunkt* von F. Der theoretische Hintergrund für alle derartigen Verfahren ist also die Beantwortung der folgenden Frage:

[80] Für Erbsenzähler: Sowas bezeichnet man meist als einen "Algorithmus" zur Lösung des Problems.

[81] Was die ganz und gar nicht einfache Frage einschließt, wie man herausfindet, ob eine Näherung auch eine gute Näherung ist.

[82] Natürlich gibt es heute auch Computeralgebra–Systeme wie zum Beispiel `Derive`, `Maple` oder `Mathematica`, die auch das können, aber eine exakte, normalerweise rationale Arithmetik hat den Pferdefuß der wachsenden Komplexität der Rechnung – Zähler und Nenner haben immer mehr Stellen.

[83] Und nicht ein "so genannter"!

Unter welchen Voraussetzungen an die Funktion F und den Startwert $x^{(0)}$ konvergiert die Iterationsfolge

$$x^{(j)} = F\left(x^{(j-1)}\right), \qquad j \in \mathbb{N},$$

gegen einen Fixpunkt $x^* = F(x^*)$.

Apropos "Konvergenz" – was ist das eigentlich? Man sagt, daß eine Folge $x^{(j)}$, $j \in \mathbb{N}$, gegen einen *Grenzwert*

$$y = \lim_{j\to\infty} x^{(j)}$$

konvergiert[84], wenn man sich eine beliebig kleine Distanz vorgeben kann und trotzdem nur *endlich viele* der Folgenglieder vom Grenzwert einen größeren Abstand haben. Anders gesagt: Gibt man sich eine beliebig kleine Zahl $\varepsilon > 0$ vor, dann gibt es eine Zahl $N(\varepsilon)$ so daß alle $x^{(j)}$, $j \geq N(\varepsilon)$ näher als ε an x^* liegen – verwenden wir also eine Norm, um den Abstand zu messen, so heißt dies, daß $\left\|x^{(j)} - x^*\right\| < \varepsilon$, $j \geq N(\varepsilon)$.

4.1 Kontraktionen

Nach dieser kurzen Begriffsklärung sollten wir uns nun aber wieder der Frage zuwenden, wann eine Iterationsfolge

$$x^{(j)} = F\left(x^{(j-1)}\right), \quad j \in \mathbb{N}, \qquad x^{(0)} \in \mathbb{R}^n, \tag{4.2}$$

gegen einen Grenzwert x^* konvergiert. Dieses Thema wird in allen möglichen Variationen in verschiedenen Bereichen der Mathematik gespielt und doch ist es im wesentlichen ein Begriff, der hier immer wieder die zentrale Rolle spielt, nämlich der Begriff der *Kontraktion*.

Definition 4.1 *Eine Abbildung $F : \mathbb{R}^n \to \mathbb{R}^n$ heißt* Kontraktion *(bezüglich einer Norm*[85] *$\|\cdot\|$), wenn es eine Zahl $0 < \rho < 1$ gibt, so daß*

$$\|F(x) - F(y)\| \leq \rho \, \|x - y\|, \qquad x, y \in \mathbb{R}^n. \tag{4.3}$$

Eine Kontraktion verringert also den Abstand zwischen zwei Punkten und zwar nicht auf irgendeine Art und Weise, sondern um einen festen und von diesen Punkten unabhängigen Faktor $\rho < 1$, siehe Abb. 4.1. Was bedeutet das nun für "unsere" Iterationsfolge? Nun, betrachtet man den Abstand zweier aufeinanderfolgender Punkte, dann ist

$$\left\|x^{(j+1)} - x^{(j)}\right\| = \left\|F\left(x^{(j)}\right) - F\left(x^{(j-1)}\right)\right\| \leq \rho \left\|x^{(j)} - x^{(j-1)}\right\| \leq \cdots \leq \rho^j \left\|x^{(1)} - x^{(0)}\right\|,$$

die Punkte rücken also sehr schnell sehr nahe zusammen, wie auch aus Abb. 4.2 ersichtlich ist. Und tatsächlich trügt die Anschauung aus diesem Bild nicht, der Folge bleibt tatsächlich

[84] Dies ist eine (wenn auch etwas oberflächliche) Definition der Begriffe *Konvergenz* und *Grenzwert* sowie des Symbols $\lim_{j\to\infty}$!

[85] "Normalerweise", aber was ist bei Mathematikern schon normal, definiert man den Begriff einer Kontraktion nur auf *metrischen Räumen*, wo man einen etwas schwächeren Abstandsbegriff verwendet. Genauer: Jede Norm definiert eine Metrik, aber nicht jede Metrik kommt von einer Norm.

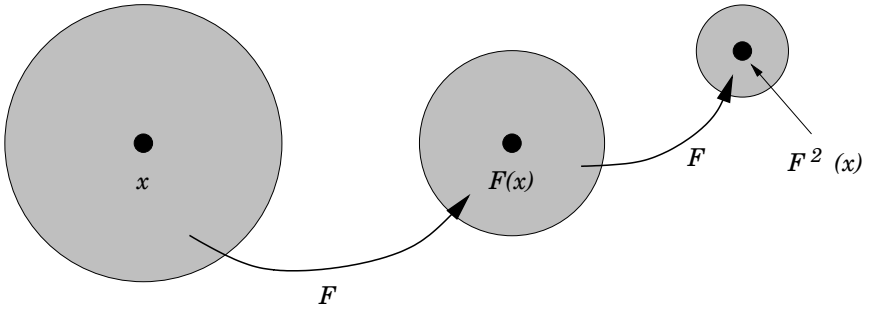


Abbildung 4.1: Die Wirkungsweise einer Kontraktion. Sie bildet die Menge aller Punkte, die von einem Punkt x einen bestimmten Abstand haben (den Kreis) auf eine Menge von Punkten ab, die von dem Punkt $F(x)$ höchstens ρ–fachen Abstand haben, was ja weniger ist. Hier ist übrigens $\rho = 1/2$.

nichts anderes übrig, als gegen einen Grenzwert zu konvergieren. Aber: So einfach ist es nicht, denn die "Geschwindigkeit", mit der die Punkte der Iterationsfolge zusammenrücken, spielt im Beweis sehr wohl eine Rolle. Wie dem auch sein, wir haben nun unser zentrales theoretisches Hilfsmittel für die Konvergenz dieser Iterationsfolgen zur Hand, nämlich den *Banachschen*[86] *Fixpunktsatz.*

Satz 4.2 *Ist $F : \mathbb{R}^n \to \mathbb{R}^n$ eine Kontraktion, dann konvergiert die Iterationsfolge (4.2) unabhängig vom Startwert $x^{(0)}$ gegen einen eindeutigen Fixpunkt $x^* = F(x^*)$.*

Nun sind aber nicht die Kontraktionen allein des Rätsels Lösung, man kann mit wenig Aufwand die Sache verallgemeinern:

1. Man kann sich auf *lokale* Kontraktionen beschränken: Anstatt die Kontraktivitätsbedingung (4.3) für *alle* Vektoren zu fordern, verlangt man nur, daß sie für alle x, y aus einem *Teilbereich* gilt und, und das braucht man, daß $F(D) \subseteq D$, das heißt, daß die Funktion F diesen "magischen" Bereich wieder in sich selbst abbildet. Aber selbst wenn diese Eigenschaft der "Selbstabbildung" vorliegt, stellt die Lokalität ein im wesentlichen unlösbares Problem dar: Wenn man will, daß die Sache gegen einen Fixpunkt (= Lösung) in D konvergiert, dann muß der *Startwert* $x^{(0)}$ auch schon in D liegen. Mit anderen Worten, wir müssen bereits eine halbwegs gute Idee haben, wo die Lösung liegt!

2. Man kann fordern, daß nicht F selbst, sondern eine mehrfache Hintereinanderschaltung von F eine Kontraktion ist, sagen wir beispielsweise

$$F^m = \underbrace{F \circ \cdots \circ F}_{m}, \qquad m \in \mathbb{N}.$$

[86] Stefan Banach, 30.3.1892–31.8.1945, "deutsch"-polnisch-ukrainischer Mathematiker, einer der Väter der Funktionalanalysis.

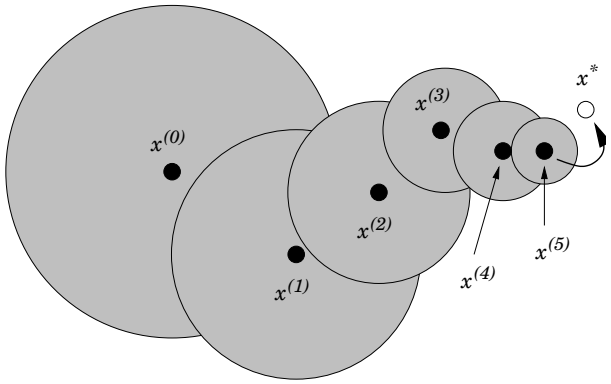


Abbildung 4.2: Eine Iterationsfolge bezüglich einer Kontraktion. Der Abstand zwischen zwei aufeinanderfolgenden Punkten muß ja das ρ^j–fache (hier: $\rho = \frac{3}{4}$) des Abstands zwischen den beiden ersten Werten der Iterationsfolge, $x^{(0)}$ und $x^{(1)}$, betragen.

Denn dann ist, für $j \in \mathbb{N}$,

$$x^{(j)} \;=\; F^j\left(x^{(0)}\right) = \underbrace{\underbrace{F\circ\cdots\circ F}_{m}\circ\cdots\circ\underbrace{F\circ\cdots\circ F}_{m}}_{k}\circ\underbrace{F\circ\cdots\circ F}_{k'}\left(x^{(0)}\right),$$

$$j = mk + k', \quad 0 \le k' < m,$$

also, mit $G = F^m$,

$$x^{(j)} = G^k\left(x^{(k')}\right).$$

Da G nun eine Kontraktion ist, konvergiert also für jedes $k' \in \{0, \ldots, m-1\}$ die Folge $y^{(k)} = G^k\left(x^{(k')}\right)$ gegen einen Grenzwert x^*, der noch dazu vom Startwert $x^{(k')}$ ist – und damit konvergiert auch die Folge $x^{(j)}$, $j \in \mathbb{N}$, gegen x^*.

4.2 Iterative Verfahren für lineare Gleichungssysteme

Zuerst wenden wir die "neue, iterative Idee" auf ein bekanntes Problem an, nämlich auf die Bestimmung der Lösungen eines linearen Gleichungssystems $Ax = b$. Dazu wählen wir erst mal eine beliebige *invertierbare* Matrix $B \in \mathbb{R}^{n\times n}$ und schreiben

$$b = Ax = (A - B + B)\,x = (A - B)\,x + Bx \qquad \Longleftrightarrow \qquad Bx = b - (A - B)\,x.$$

Da B invertierbar ist, können wir beide Seiten von links mit B^{-1} multiplizieren und erhalten, daß

$$x = B^{-1}b - B^{-1}(A - B)\,x = B^{-1}b + \left(I - B^{-1}A\right)x =: F(x), \tag{4.4}$$

und schon wird aus der Lösung x des linearen Gleichungssystems $Ax = b$ ein Fixpunkt der Funktion F und umgekehrt. Die Funktion F hat ja die Gestalt[87]

$$F(x) = \underbrace{\left(I - B^{-1}A\right)}_{=:G} x + \underbrace{B^{-1}b}_{=:h} = Gx + h, \qquad G \in \mathbb{R}^{n\times n},\ h \in \mathbb{R}^n.$$

Wann ist sowas nun eine Kontraktion? Und vor allem: Kann man, für ein vorgegebenes $A \in \mathbb{R}^{n\times n}$ immer eine invertierbare[88] Matrix B finden, so daß das iterative Verfahren konvergiert.

Beispiel 4.3 *Beginnen wir mit dem Fall $n = 1$ – warum sich das Leben nicht erst einmal einfach machen? Als "Norm" können wir jetzt den ganz normalen Absolutbetrag verwenden und erhalten, daß*

$$|F(x) - F(y)| = |Gx + h - Gy - h| = |Gx - Gy| = |G|\,|x - y|, \tag{4.5}$$

also ist F genau dann eine Kontraktion, wenn $|G| < 1$. Nun, das ist leicht zu bekommen: Hat man $0 \neq A \in \mathbb{R}$ gegeben, dann setzt man eben $B = A$ und es ist

$$G = 1 - B^{-1}A = 1 - \frac{A}{B} = 1 - 1 = 0,$$

was nicht nun nicht verboten sondern sogar sehr wünschenswert ist – das iterative Verfahren konvergiert immer in einem *Schritt.*

Wann ist nun aber für $n > 1$ die Abbildung F aus (4.4) eine Kontraktion. Eigentlich steht schon fast alles in (4.5)! Mit einer p–Norm erhalten wir nämlich, daß[89]

$$\|F(x) - F(y)\|_p = \|Gx + h - Gy - h\|_p \leq \|G\|_p\,\|x - y\|_p$$

und deswegen gilt:

F ist eine Kontraktion bezüglich der Norm $\|\cdot\|_p$, wenn $\|G\|_p < 1$ ist.

Ob eine Abbildung F eine Kontraktion ist, hängt natürlich auch vom verwendeten "Abstandsbegriff", also von der verwendeten Norm ab. So ist beispielsweise die Abbildung $F(x) = Ax$,

$$A = \begin{bmatrix} \frac{2}{3} & \frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} \end{bmatrix}$$

[87] Solche Funktionen der Form $F(x) = Gx + h$ bezeichnet man als *affine* Funktionen, im Gegensatz zu den *linearen* Funktionen, bei denen $h = 0$ sein muß.

[88] In der Praxis muß man dies zu "einfach invertierbare" verschärfen, denn was hilft es uns, wenn die Invertierung von B genauso aufwendig ist wie das Lösen des Gleichungssystems $Ax = b$.

[89] Hier nutzen wir aus, daß die p–Matrixnormen $\|\cdot\|_p$ aus (3.21) in Definition 3.19 mit den p–Vektornormen *verträglich* sind, das heißt, daß $\|Ax\|_p \leq \|A\|_p\,\|x\|_p$ ist. In gewissem Sinne sind die Matrixnormen sogar so "gebastelt", daß sie gerade diese Forderung erfüllen.

eine Kontraktion bezüglich der Norm $\|\cdot\|_\infty$, aber *keine* Kontraktion bezüglich der $\|\cdot\|_1$. Es ist nämlich, für $e_1 = [1,0]^T$, einfach nachzurechnen, daß $\|e_1\|_1 = \|e_1\|_\infty = 1$ und außerdem gilt

$$F(e_1) - F(0) = \begin{bmatrix} \frac{2}{3} & \frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{2}{3} \end{bmatrix} \qquad \Longrightarrow \qquad \begin{cases} \|F(e_1) - F(0)\|_1 = \frac{4}{3} > 1, \\ \|F(e_1) - F(0)\|_\infty = \frac{2}{3} < 1. \end{cases}$$

Anders wird die Sache, wenn wir nur fordern, daß "nur" *irgendeine* Potenz A^m, $m \in \mathbb{N}$, von A eine Kontraktion sein muß, was ja für die Konvergenz ausreichend ist. Dies ist genau dann der Fall, wenn der *Spektralradius*[90]

$$\rho(A) = \limsup_{r\to\infty} \|A^r\|_p^{1/r}.$$

kleiner als 1 ist. Diese Definition des Spektralradius ist, das kann man zeigen, unabhängig vom gewählten p und außerdem ist $\rho(A) \le \|A\|_p$ für alle $1 \le p \le \infty$.

Wir werden jetzt zwei Verfahren angeben, das Jacobi–Verfahren und das Gauß–Seidel–Verfahren, die in einer besonderen Wahl der Matrix B (natürlich in Abhängigkeit von A) bestehen und dann Typen von Matrizen angeben, für die diese Verfahren funktionieren. Auch wenn wir uns die Beweise natürlich schenken werden, läuft das alles immer auf ein Ziel hinaus:

> *Man zeige, daß für die Wahl von B im Verfahren $\rho(I - B^{-1}A) < 1$ ist, wobei es manchmal schon genügt, ein p zu finden, so daß $\|I - B^{-1}A\| < 1$ ist.*

Natürlich könnten wir, wie im Beispiel oben wieder $B = A$ wählen und hätten "Konvergenz" in einem Schritt – nicht überraschend, denn die Iteration wäre dann

$$x^{(j+1)} = \underbrace{B^{-1}}_{=A^{-1}} b + \underbrace{\left(I - B^{-1}A\right)}_{=0} x^{(j)} = A^{-1}b,$$

aber das wussten wir ja vorher schon, wie wir ein Gleichungssystem $Ax = b$ lösen können, wenn wir die Matrix A invertieren können.

Wie also wählt man dann B? Nun, schreiben wir die Iteration

$$x^{(j+1)} = B^{-1}b + \left(I - B^{-1}A\right) x^{(j)}, \qquad j \in \mathbb{N}_0, \tag{4.6}$$

nochmal in der *impliziten*[91] Form

$$Bx^{(j+1)} = b + (B - A)x^{(j)}, \qquad j \in \mathbb{N}_0, \tag{4.7}$$

[90] Was ist nun wieder der "lim sup"? Nun, wenn eine Folge nicht konvergiert, aber beschränkt bleibt, dann gibt es einen oder mehrere Werte, in deren nächster Nachbarschaft sich wieder *unendlich* viele Folgenglieder tummeln – sowas nennt man einen *Häufungspunkt*. Hat man nur einen Häufungspunkt, dann ist dieser der Grenzwert und die Folge konvergiert, hat man mehrere Häufungspunkte, dann gibt es einen größten (kann auch $+\infty$ sein!) und den nennt man *limes superior* oder eben, kurz und bündig "lim sup".

[91] "Implizit" bedeutet, daß nicht der zu berechnende Wert, $x^{(j+1)}$ in dieser Gleichung steht, sondern nur eine Funktion $f\left(x^{(j+1)}\right) = Bx^{(j+1)}$. Mathesinologie eben.

```
%%  JacobiIt.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Iterationsschritt im Jacobi-Verfahren
%%  Eingabe:
%%    A    Matrix
%%    x    Vektor
%%    b    Vektor, rechte Seite

function y = JacobiIt( A,x,b )
  n = length( A );
  y = x;

  y( 1 ) = ( b( 1 ) - [ 0,A( 1,2:n ) ] * x ) / A( 1,1 );
  for j = 2:n-1
    y( j ) = ( b( j ) - [ A( j,1:j-1 ),0,A( j,j+1:n ) ] * x ) / A( j,j );
  end
  y( n ) = ( b( n ) - [ A( n,1:n-1 ),0 ] * x ) / A( n,n );
```

Programm 4.1 `JacobiIt.m`: Ein Iterationsschritt bei der Jacobi–Iteration.

dann besteht der einfachste Fall ist sicherlich in der Auswahl

$$B = \text{diag}\,[a_{jj} \,:\, j = 1, \ldots, n]\,,$$

vorausgesetzt natürlich, daß $a_{jj} \neq 0$, $j = 1, \ldots, n$. Und das ist auch bereits das *Jacobi–Verfahren*, in der deutschsprachigen Literatur auch gerne als *Gesamtschrittverfahren* bezeichnet:

$$\begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{bmatrix} x^{(j+1)} = b - \begin{bmatrix} 0 & a_{12} & \ldots & a_{1n} \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & \ldots & a_{n,n-1} & 0 \end{bmatrix} x^{(j)} \tag{4.8}$$

beziehungsweise

$$x_k^{(j+1)} = \left(b_k - \sum_{r \neq k} a_{kr} x_r^{(j)} \right) / a_{kk}, \qquad k = 1, \ldots, n. \tag{4.9}$$

Bei der *Gauß–Seidel–Iteration*, dem *Einzelschrittverfahren*, setzt man

$$B = \begin{bmatrix} a_{11} & & 0 \\ \vdots & \ddots & \\ a_{n1} & \ldots & a_{nn} \end{bmatrix},$$

```
%%  GaussSeidelIt.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Iterationsschritt im Gauss-Seidel-Verfahren
%%  Eingabe:
%%    A    Matrix
%%    x    Vektor

function y = GaussSeidelIt( A,x,b )
  n = length( A );

  x( 1 ) = ( b( 1 ) - [ 0,A( 1,2:n ) ] * x ) / A( 1,1 );
  for j = 2:n-1
    x( j ) = ( b( j ) - [ A( j,1:j-1 ),0,A( j,j+1:n ) ] * x ) / A( j,j );
  end
  x( n ) = ( b( n ) - [ A( n,1:n-1 ),0 ] * x ) / A( n,n );
  y = x;
```

Programm 4.2 `GaussSeidelIt.m`: Ein Iterationsschritt im Gauß–Seidel–Verfahren.

also, mit Hilfe unserer wohlbekannten und hoffentlich auch ebenso geschätzten Vorwärtselimination,

$$\begin{bmatrix} a_{11} & & 0 \\ \vdots & \ddots & \\ a_{n1} & \dots & a_{nn} \end{bmatrix} x^{(j+1)} = b - \begin{bmatrix} 0 & a_{21} & \dots & a_{n1} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,1} \\ & & & 0 \end{bmatrix} x^{(j)}, \tag{4.10}$$

beziehungsweise

$$x_k^{(j+1)} = \left(b_k - \sum_{r=1}^{k-1} a_{kr} x_r^{(j+1)} - \sum_{r=k+1}^{n} a_{kr} x_r^{(j)} \right) / a_{kk}, \qquad k = 1, \dots, n. \tag{4.11}$$

Diese Iterationsschritte sind in den beiden Programmen 4.1 und 4.2 implementiert, die vollständige Iteration in Interessant ist nun natürlich die Frage, wann solche Iterationsverfahren auch wirklich funktionieren, das heißt, wann sie konvergieren. Dazu zwei Typen von Matrizen, die in der Praxis erfreulich oft vorkommen.

Definition 4.4 *Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt*

1. positiv definit*, wenn*

$$x^T A x > 0, \qquad x \in \mathbb{R}^n, \quad x \neq 0.$$

```
%%  Jacobi.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Loesung von Gleichungssystemen mit dem Jacobi-Verfahren
%%  Eingabe:
%%    A    Matrix
%%    b    Vektor, rechte Seite
%%    tol  Genauigkeit

function [y,iter] = Jacobi( A,b,tol )
  x = b; y = JacobiIt( A,x,b ); iter = 1;

  while ( ( norm( x - y ) / ( 1 + norm( x ) ) ) > tol )
    x = y;
    y = JacobiIt( A,x,b );
    iter = iter + 1;
  end
```

Programm 4.3 `Jacobi.m`: Das Jacobi–Verfahren. Es wird iteriert, bis zwei aufeinanderfolgende Vektoren einander "hinreichend nahe" gekommen sind. Dies ist nur eine Abbruchbedingung von vielen! Der zweite Rückgabewert ist die Anzahl der verwendeten Iterationen.

```
%%  GaussSeidel.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Loesung von Gleichungssystemen mit dem Gauss-Seidel-Verfahren
%%  Eingabe:
%%    A    Matrix
%%    b    Vektor, rechte Seite
%%    tol  Genauigkeit

function [y,iter] = GaussSeidel( A,b,tol )
  x = b; y = GaussSeidelIt( A,x,b ); iter = 1;

  while ( ( norm( x - y ) / ( 1 + norm( x ) ) ) > tol )
    x = y;
    y = GaussSeidelIt( A,x,b );
    iter = iter + 1;
  end
```

Programm 4.4 `GaussSeidel.m`: Das Gauß–Seidel–Verfahren. Alles, was beim Jacobi–Verfahren gesagt wurde, gilt natürlich auch hier.

2. diagonaldominant *nach Zeilen bzw. Spalten, wenn*

$$|a_{jj}| > \sum_{k \neq j} |a_{jk}| \qquad \textit{bzw.} \qquad |a_{jj}| > \sum_{k \neq j} |a_{kj}|\,, \qquad j = 1, \ldots, n.$$

Offensichtlich ist Diagonaldominanz ja einfach zu überprüfen, die Diagonalelemente müssen halt eben (betragsmäßig) so groß sein, daß alle anderen Elemente der zugehörigen Zeile (bzw. Spalte) "keine Chance" dagegen haben. Übrigens ist es weder unmöglich noch verboten, daß eine Matrix nach Zeilen *und* Spalten diagonaldominant ist – man denke nur an Diagonalmatrizen[92].

Wie aber sieht es mit der positiven Definitheit aus? Nun, eigentlich ist sie nichts anderes, als die Positivität von Quadratzahlen[93], nur eben auf Matrizen übertragen. Ist nämlich $B \in \mathbb{R}^{n \times n}$ eine *invertierbare* Matrix[94], dann ist die Matrix $A = B^T B$ symmetrisch,

$$A^T = \left(B^T B\right)^T = B^T \left(B^T\right)^T = B^T B = A,$$

und positiv definit, denn für $x \neq 0$ ist

$$x^T A x = x^T B^T B x = \underbrace{(Bx)^T}_{\neq 0} \underbrace{(Bx)}_{\neq 0} = \sum_{j=1}^{n} (Bx)_j^2 > 0.$$

Umgekehrt hat übrigens auch jede symmetrische, positiv definite Matrix A eine "Wurzel" B, die man sogar als untere Dreiecksmatrix wählen und mit dem sogenannten *Cholesky–Verfahren* auch explizit berechnen kann.

Nun aber zu unserem eigentlichen Ziel, nämlich der Beantwortung der Frage, wann, genauer, für welche Typen von Matrizen, die iterativen Verfahren denn nun konvergieren. Und, welch ein Zufall, es sind die gerade definierten Matrizenklassen.

Satz 4.5 *Sei $A \in \mathbb{R}^{n \times n}$.*

1. *Ist A symmetrisch und positiv definit, dann konvergiert das Gauß–Seidel–Verfahren.*
2. *Ist A nach Zeilen diagonaldominant, dann konvergieren das Jacobi–Verfahren und das Gauß–Seidel–Verfahren.*

Beispiel 4.6 *Sehen wir uns mal an, wie es nun mit der Konvergenzgeschwindigkeit (wieviele Iterationen braucht man) für die Verfahren aussieht. Dazu betrachten wir die Matrizen*

$$\begin{bmatrix} n & 1 & \ldots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \ldots & 1 & n \end{bmatrix} \in \mathbb{R}^{n \times n},$$

[92] Zugegeben, ein etwas billiges Beispiel.
[93] Ist $x \neq 0$ eine reelle Zahl, dann ist $x^2 > 0$.
[94] Also "$\neq 0$".

```
%%  auxnMat.m (Numerik HaF)
%%  ------------------------------------------------------
%%  n x n - Matrix mit n auf der Hauptdiagonalen, sonst 1.
%%  Eingabe:
%%    n    Groesse

function L = auxnMat( n )
  L = (n-1) * eye( n ) + ones( n );
```

Programm 4.5 `auxnMat.m`: Die Matrizen aus Beispiel 4.6.

die von Programm 4.5 geliefert werden, und sehen uns mal an, wieviele Iterationsschritte Jacobi und Gauß–Seidel bei $b = [1, \ldots, 1]^T$ brauchen, um eine Lösung von vorgegebener Genauigkeit zu erreichen.

	10^{-6}		10^{-10}		10^{-15}	
n	Jacobi	Gauß–Seidel	Jacobi	Gauß–Seidel	Jacobi	Gauß–Seidel
4	52	10	84	16	124	23
10	148	11	235	17	344	24
20	310	11	490	17	$\star$	25
50	812	12	1268	18	$\star$	25

Wie erklärt sich nun dieses Verhalten – hierbei bedeutet $\star$ übrigens, daß das Verfahren iteriert und iteriert, ohne zu einer Lösung zu kommen? Ist Jacobi das dümmere Verfahren? Natürlich so allgemein nicht! Den Grund sehen wir, wenn wir uns den Spektralradius der Iterationsmatrizen ansehen, der von der Funktion `JGSpecRad` *aus Programm 4.6 berechnet wurde:*

	$n = 4$	$n = 10$	$n = 20$	$n = 50$
Jacobi	0.75000	0.90000	0.95000	0.98000
Gauß–Seidel	0.18987	0.20152	0.20828	0.21285

Nachdem sich die Konvergenzgeschwindigkeit nach j Iterationen wie ρ^j verhält, schneidet hier natürlich das Jacobi–Verfahren hier sehr viel schlechter ab.

4.3 Iteration für nichtlineare Gleichungen

Jetzt verlassen wir die Welt der *linearen* Gleichungssysteme und versuchen, allgemeine, *nichtlineare* Gleichungen zu lösen – das dafür aber nur für den Fall $n = 1$. Wir suchen also jetzt, für eine gegebene Funktion f und einen Wert $y \in \mathbb{R}$ einen Wert $x \in \mathbb{R}$, so daß $f(x) = y$. Dabei genügt es uns völlig, *eine* Lösung zu finden, denn im allgemeinen können solche Gleichungen so viele Lösungen haben, wie sie wollen, man denke nur an $\sin x = 1$. Das Mittel der Wahl sollen aber hier auch wieder *iterative* Verfahren sein. Dazu ein paar Beobachtungen:

```
%%  SpecRad.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Spektralradius einer Matrix als betragsgroesster Eigenwert
%%  Eingabe:
%%    A    Matrix


function r = SpecRad( A )
  r = max( abs( eig( A ) ) );
```

Programm 4.6 `SpecRad.m`: Berechnung des Spektralradius einer Matrix als deren betragsgrößter Eigenwert (was auch immer das ist) über die `Matlab`/ `Octave`–Funktion `eig`.

```
%%  JGSpecRad.m (Numerik HaF)
%%  ------------------------------------------------------------
%%  Spektralradius der Iterationsmatrizen fuer Jacobi und
%%    Gauss-Seidel.
%%  Eingabe:
%%    A    Matrix


function [rJ,rG] = JGSpecRad( A )
  n = length( A );
  Ja = eye( n ) - diag( diag( A ) )^(-1) * A;
  Ga = eye( n ) - tril( A )^(-1) * A;

  rJ = SpecRad( Ja );
  rG = SpecRad( Ga );
```

Programm 4.7 `JGSpecRad.m`: Die Iterationsmatrizen des Jacobi- und Gauß–Seidel–Verfahrens und deren Spektralradius.

1. Wir können anstelle von $f(x) = y$ auch $0 = f(x) - y$ oder
$$x = \underbrace{f(x) + x - y}_{=:F(x)}$$
schreiben und schon haben wir unser Fixpunktproblem. Wir werden aber sehr bald sehen, daß es sehr wohl auch darauf ankommt, *wie* wir eine Gleichung in ein Fixpunktproblem umschreiben, wenn wir auch eine Lösung erhalten wollen[95].

2. Lineare Verfahren waren besonders einfach: Ist $F(x) = ax - b$ und suchen wir nach einem Fixpunkt von f, so haben wir eine Kontraktion (und damit Konvergenz für *jeden* Startwert), wenn es ein $\rho \in (0, 1)$ gibt, so daß
$$\rho\,|x - y| > |F(x) - F(y)| = |ax - ay| = |a|\,|x - y|, \qquad x \neq y,$$
also genau dann, wenn $a < 1$. Und andernfalls ($a \geq 1$) funktioniert die Iteration für *keinen* Startwert, denn es ist ja
$$\left|x^{(j+1)} - x^{(j)}\right| = \left|F\left(x^{(j)}\right) - F\left(x^{(j-1)}\right)\right| = |a|\,\left|x^{(j)} - x^{(j-1)}\right|, \qquad j \in \mathbb{N},$$
weswegen eine Annäherung der Punkte aneinander, eine notwendige Voraussetzung für Konvergenz, nicht auftreten kann.

3. Für beliebige f geht das nicht mehr. Sehen wir uns zum Beispiel $F(x) = x^2$ an, dann hat diese Funktion *zwei*[96] Fixpunkte, nämlich $x = 0$ und $x = 1$. Gegen den ersten Fixpunkt, $x = 0$, konvergiert das Verfahren nun für jeden Startwert $x^{(0)}$ mit $\left|x^{(0)}\right| < 1$, gegen den zweiten Fixpunkt dagegen nur, wenn $x^{(0)} = \pm 1$. Übrigens: Ist $\left|x^{(0)}\right| > 1$, dann ist $\lim_{j\to\infty} x^{(j)} = \infty$.

Definition 4.7 *Ein Iterationsverfahren $x^{(j+1)} = F\left(x^{(j)}\right)$ heißt* lokal konvergent*, wenn es ein "echtes" Intervall $[a, b]$, $a < b$, gibt, so daß die Folge $x^{(j)}$, $j \in \mathbb{N}$, für alle Startwerte aus dem offenen Intervall*[97] *(a, b) gegen eine Fixpunkt konvergiert.*

Denken wir nochmal kurz zurück an die Bemerkungen nach Satz 4.2, dem Banachschen Fixpunktsatz, dann wissen wir, was des Rätsels Lösung sein wir, nämlich der Begriff der *lokalen* Kontraktion, allerdings jetzt nur noch als *hinreichende* Bedingung[98], wie uns unser Beispiel $F(x) = x^2$ wieder einmal deutlich vor Augen führt, denn diese Funktion ist eine Kontraktion, wenn
$$\rho\,|x - y| > \left|x^2 - y^2\right| = |(x + y)(x - y)| = |x + y|\,|x - y|\,,$$
also wenn $|x + y| \leq \rho < 1$, das heißt, wenn
$$x, y \in \left(-\frac{1}{2} + \varepsilon, \frac{1}{2} - \varepsilon\right), \qquad 0 < \varepsilon.$$

[95] Zugegeben, das ist eine Zusatzbedingung, aber ja wohl keine allzu abwegige.
[96] Also nix mehr mit Eindeutigkeit.
[97] Für unsere Zwecke reicht es, daß dies das Intervall *ohne* die Randpunkte ist.
[98] Also eine Bedingung, die das Funktionieren garantiert, ohne dafür unbedingt nötig zu sein.

Man beachte, daß $\varepsilon = 0$ *nicht* erlaubt ist, denn dann kann $|x + y|$ beliebig nahe an 1 herankommen und man kann kein ρ mehr "dazwischenquetschen".

So, jetzt aber mal ein konkretes Beispiel für ein absolut klassisches Verfahren zur Berechnung der *Quadratwurzel* $\sqrt{q}$, $q \in \mathbb{Q}$[99], einer Zahl, das man als *Heron*[100]*–Verfahren* bezeichnet, das man aber bereits in babylonischen Keilschrifttexten, siehe [9, S. 125–127], findet. Erstaunlicherweise waren die Rechenmethoden der Babylonier sehr algebraisch orientiert[101], im Gegensatz zu den Rechenmeistern der Renaissance, die im wesentlichen mit geometrischen Konstruktionen im Sinne von Euklid argumentierten, siehe z.B. [6]. Es spricht übrigens für die "Alten", daß sie die Unmöglichkeit, eine Quadratwurzel direkt zu berechnen[102] in Betracht gezogen hatten – so wird bereits in der babylonischen Mathematik das Heron–Verfahren zur *näherungsweisen* Berechnung von Quadratwurzeln verwendet.

Beispiel 4.8 *(Heron–Verfahren)*
Zu $y \in \mathbb{Q}$ berechnet man die Iterationsfolge

$$x^{(0)} = y, \qquad x^{(j+1)} = \frac{1}{2}\left(x^{(j)} + \frac{y}{x^{(j)}}\right), \quad j \in \mathbb{N}_0, \tag{4.12}$$

die gegen $\sqrt{y}$ konvergiert. Tatsächlich ist zuerst einmal $\sqrt{y}$ ein Fixpunkt der Funktion $F(x) = \frac{1}{2}(x + y/x)$, da

$$F(\sqrt{y}) = \frac{1}{2}(\sqrt{y} + y/\sqrt{y}) = \sqrt{y},$$

den Konvergenzbeweis schenken wir uns, nicht aber die geometrische Interpretation des Iterationsschritts beim Heron–Verfahren, ganz im Sinne der Rechenmeister, in Abb. 4.3.

Übung 4.1 Zeigen Sie: Ist $a > b$, dann ist auch $\frac{a+b}{2} > \frac{2ab}{a+b}$, das "Querformat" des Rechtecks bleibt also erhalten.

Das Heron–Verfahren ist aber nicht die einzige Fixpunktiteration, die $\sqrt{y}$ als Fixpunkt hat. Amüsant wird das Ganze dann dadurch, das nun wahrlich nicht alle derartigen Iterationen auch wirklich konvergieren müssen.

Beispiel 4.9 *Beschränkt man sich*[103] *auf den Fall $y > 1$, so ist $\sqrt{y}$ ein Fixpunkt der Funktionen*

$$\begin{aligned} F_1(x) &= \frac{1}{2}\left(x + \frac{y}{x}\right), \qquad \textit{(Heron)} \\ F_2(x) &= \frac{y}{x}, \\ F_3(x) &= 2x - \frac{y}{x}. \end{aligned}$$

[99] $\mathbb{Q}$ steht für den Körper der rationalen Zahlen, das heißt, alle Zahlen, die als Quotient $r = \frac{p}{q}$ darstellbar sind, wobei p und q ganze Zahlen sind. Die Wurzeln liegen im allgemeinen nicht mehr in diesem Körper, was bereits die Pythagoräer, sehr zum Leidwesen ihres religionsähnlichen Weltbilds der harmonischen Verhältnisse natürlicher Zahlen wußten, aber die "Erweiterung" zu den heute so beliebten reellen Zahlen erfolgte erst sehr viel später, nämlich im 19. Jahrhundert.

[100] Ja, der mit dem Dreieck ...

[101] Behauptet zumindest [8], den ich aber auch nur als Zitat kenne.

[102] Außer, wenn man sie aus einer Qudratzahl zieht! Ein Verfahren dafür findet man sogar bei Adam Riese [10].

[103] Eine echte Einschränkung ist das nicht, da $\sqrt{1/y} = 1/\sqrt{y}$ ist

```
%%  Heron.m (Numerik HaF)
%%  ---------------------------------------------------------
%%  Heron-Verfahren fuer Quadratwurzel
%%  Eingabe:
%%    y     Zahl
%%    tol   Toleranz

function [x,it] = Heron( y,tol )
  x = y;t = 0; it = 0;

  while ( abs( x-t ) > tol )
    it = it+1;
    t = x;
    x = ( x + y/x ) / 2;
  end
```

Programm 4.8 `Heron.m`: Das Heronverfahren zur Quadratwurzelberechnung.

Starten wir nun alle drei Verfahren wieder ganz naiv mit $x^{(0)} = y$, dann liefert F_1 brav die Wurzel aus y, während wir mit F_2 die Folge $x^{(2j)} = y$, $x^{(2j+1)} = 1$ erhalten[104] *und unter Verwendung von F_3 ist sogar*

$$x^{(1)} = 2y - 1, \qquad x^{(2)} = 4y - 2 - \underbrace{\frac{y}{2y-1}}_{<1} > 4y - 3$$

und letztendlich

$$x^{(j)} > 2^j y - \left(2^j - 1\right) \quad \to \infty, \qquad j \to \infty,$$

die Folge divergiert *also kräftig.*

Das soll's aber nun sein an "Theorie", jetzt sehen wir uns die beiden "bedeutendsten" Verfahren für nichtlineare Gleichungen an.

4.4 Das Bisektionsverfahren und die Regula Falsi

Wir verabschieden uns nun einmal kurz vom Fixpunktproblem und wenden uns dem Problem zu, eine *Nullstelle* einer Funktion f zu bestimmen, also einen Wert $x \in \mathbb{R}$ so daß $f(x) = 0$ ist. Dazu müssen wir eine Annahme an die Funktion f machen, nämlich die, daß f keine Sprünge hat, sondern eine Funktion ist, die "in einem Zug" gezeichnet werden kann. Als Konsequenz aus der Stetigkeit erhält man den sogenannten Zwischenwertsatz:

[104] Es ist $F_2(y) = 1$ und $F_2(1) = y$!

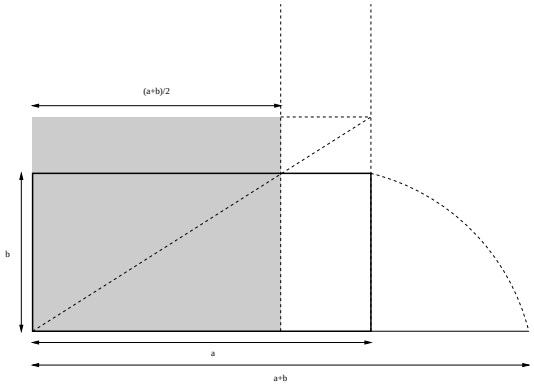


Abbildung 4.3: Ein Iterationsschritt (4.12) des Heron–Verfahrens geometrisch: Konstruiere zu vorgegebenen Seitenlängen a und b das flächengleiche Rechteck mit den Seitenlängen $a' = \frac{a+b}{2}$ und $b' = \frac{2ab}{a+b}$. Der Wert für b' ergibt sich aus dem (elementargeometrischen) *Strahlensatz*, der die Verhältnisse $\frac{b'}{b} = \frac{a}{(a+b)/2}$ liefert, was man nur noch mit b durchmultiplizieren muß.

Für $x, y \in \mathbb{R}$ nimmt eine stetige Funktion zwischen x und y jeden Wert zwischen $f(x)$ und $f(y)$ mindestens einmal an.

Das hat nun natürlich sofort eine Auswirkung auf "unser" Problem: Kennt man zu einer Funktion f zwei Punkte x, y, so daß $f(x) < 0$ und $f(y) > 0$, dann gibt es zwischen x und y mindestens einen Punkt x^*, so daß $f(x^*) = 0$ – den müssen wir nur noch finden.

Die einfachste Methode hierfür ist das *Bisektionsverfahren*, manchmal auch als *Intervallschachtelung* bekannt. Hat[105] man nämlich zwei Punkte x und y, so daß $f(x) < 0$ und $f(y) > 0$ ist, dann bildet man den Mittelpunkt $z = (x + y)/2$ und unterscheidet drei Fälle:

1. $f(z) = 0$. Das ist natürlich der beste Fall, denn damit haben wir ja die Nullstelle nicht nur gesucht sondern sogar gefunden.

2. $f(z) > 0$. Dann übernimmt z die Rolle von y, das heißt, wir setzen $x' = x$ und $y' = z$.

3. $f(z) < 0$. Jetzt wird eben x durch z ersetzt, also $x' = z$ und $y' = y$.

Ganz egal, welcher der beiden letzten Fälle nun eintritt, wir haben auf alle Fälle wieder zwei Punkte mit unterschiedlichen Vorzeichen, also $f(x') < 0$ sowie $f(y') > 0$, aber nun ist

$$x' - y' = \left\{ \begin{array}{c} x - z \\ z - y \end{array} \right\} = \left\{ \begin{array}{c} x - (x+y)/2 \\ (x+y)/2 - y \end{array} \right\} = \frac{x-y}{2},$$

das heißt die beiden Intervallgrenzen liegen nur noch halb so weit auseinander wie vorher. Damit ist das Verfahren aber klar: Wir starten mit Punkten x_0 und y_0, an denen f unterschiedliches Vorzeichen hat, und bilden dann mit der obigen Vorschrift Paare $(x_1, y_1), (x_2, y_2), \ldots$

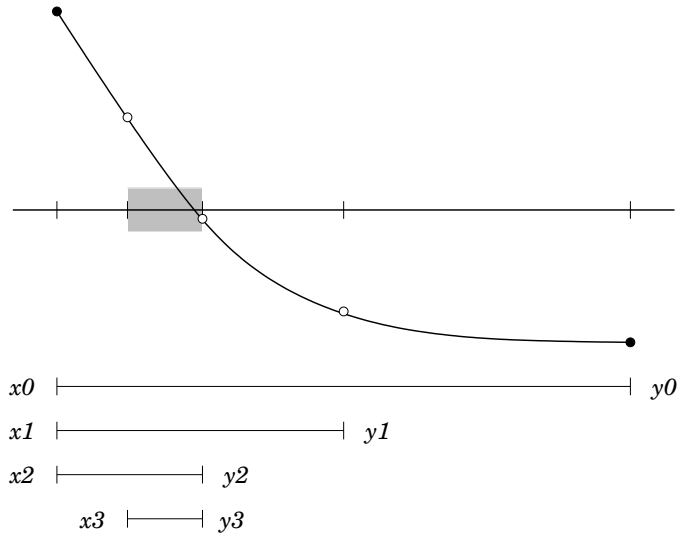


Abbildung 4.4: Die Vorgegehensweise beim Bisektionsverfahren. Ersetzt wird immer der Endpunkt, dessen Vorzeichen mit dem Vorzeichen am Mittelpunkt übereinstimmt. Nach jedem Iterationsschritt dieses Verfahrens besteht – wegen der Stetigkeit der Funktion – nun immer die Gewißheit, daß sich zwischen den Endpunkten eine Nullstelle befindet, nach unseren drei Schritten hier also irgendwo im schraffierten Intervall.

und so weiter, die eine Nullstelle immer besser einschließen. Und nachdem irgendwann die beiden Werte sich nur noch um einen "Rundungsfehler" unterscheiden, haben wir also die Nullstelle in Rechengenauigkeit gefunden. Ein paar Bemerkungen zum Bisektionsverfahren:

1. Das Bisektionsverfahren ist mit Sicherheit die einfachste Methode, um Nullstellen zu bestimmen. Trotzdem kann man sogar beweisen, daß es kein noch so kompliziertes Verfahren gibt, daß für *alle* stetigen Funktionen besser[106] ist als das Bisektionsverfahren. Das heißt übrigens weder, daß man sich die Suche nach anderen Verfahren sparen kann, noch daß dem Bisektionsverfahren irgendeine inhärente Genialität innewohnt, es heißt lediglich, daß die stetigen Funktionen einfach eine zu große Menge darstellen.

2. Die Bedingung, zwei Punkte mit unterschiedlichem Vorzeichen zu haben ist natürlich nicht notwendig, um eine Nullstelle zwischen x und y zu erhalten, siehe Abb. 4.5 (a), noch garantiert diese Bedingung die Existenz nur einer Nullstelle, siehe Abb. 4.5 (b).

3. Das Hauptproblem beim Bisektionsverfahren besteht aber nach wie vor in der Bestimmung der Startpunkte x_0, y_0. Glücklich kann sich schätzen, wer diese Information aus der zugrundeliegenden Funktion ablesen kann oder aus Problemheuristiken heraus kennt.

[105] Woher man die hat? Gute Frage!

[106] Im Sinne von schnellerer Konvergenz.

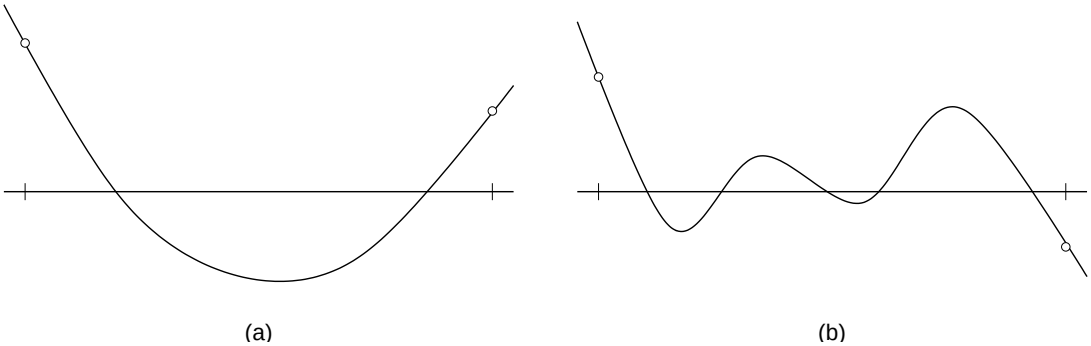


Abbildung 4.5: Im Beispiel (a) kann das Bisektionsverfahren nicht angewandt werden, da die Anfangsbedingung nicht erfüllt ist, obwohl man nach einem Iterationsschritt und Austausch eines beliebigen Endpunkts "in business" wäre, im Beispiel (b) ist die Anfangsbedingung zwar erfüllt, welche Nullstelle aber gefunden wird, ist mehr oder weniger eine Glückssache.

Ansonsten bleibt nur eine wilde Herumprobiererei, um zwei solche Punkte durch Glück und Zufall[107] zu finden.

Übung 4.2 Bestimmen Sie Startwerte x_0, y_0 für die Berechnung der Nullstelle der Funktion $f(x) = x^2 - a$, also für die Bestimmung von $\sqrt{a}$.

Eine erste Verbesserung des Bisektionsverfahrens läßt sich mit einer einfachen Heuristik motivieren: Ist einer der beiden bekanten Werte $f(x)$ und $f(y)$ betraglich sehr groß und der andere hingegen eher von kleinem Betrag, dann könnte man ja zumindest hoffen, daß die Nullstelle eher nahe bei dem kleinen Wert liegt. Zu diesem Zweck wählt man, wie in Abb. 4.6 skizziert, z nicht mehr einfach als den Mittelpunkt von x und y, sondern sieht sich die Gerade

$$\ell(t) = f(x)\frac{t-y}{x-y} + f(y)\frac{x-t}{x-y}$$

an, die die Eigenschaft hat[108], daß $\ell(x) = f(x)$ und $\ell(y) = f(y)$ ist und bestimmt z so, daß

$$0 = \ell(z) = \frac{(z-y)f(x) - (z-x)f(y)}{x-y} = \frac{z\,(f(x)-f(y)) - (y\,f(x) - x\,f(y))}{x-y},$$

also

$$z = \frac{y\,f(x) - x\,f(y)}{f(x) - f(y)}. \tag{4.13}$$

Mit diesem Wert z verfährt man dann wieder wie beim Bisektionsverfahren und erhält so wieder immer kleinere Intervalle, die die Lösung (hoffentlich) bessern und besser einschließen.

Bei Adam Riese [10] liest sich Gleichung (4.13) wie folgt:

[107] Der moderne Fachbegriff hierfür ist *Monte–Carlo–Verfahren*.

[108] Wer's nicht glaubt soll einfach x und y einsetzen.

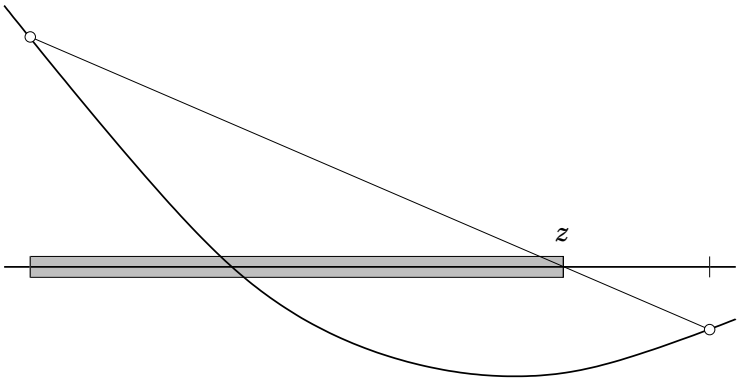


Abbildung 4.6: Ein Iterationsschritt der Regula Falsi. Der "Testpunkt" wird nicht einfach als Intervallmitte gewählt, sondern als (eindeutige!) Nullstelle der Verbindungsstrecke zwischen $f(x)$ und $f(y)$. Weitergemacht wird also mit dem schraffierten Bereich.

Regula Falsi oder Position.

Wirdt gefaßt von zweyen falschen zahlen / welche der auffgab nach / mit fleiß examinirt sollen werden / in massen das fragstück begeren ist / sagen sie der warheit zu viel / so bezeichne sie mit dem zeichen + plus / wo aber zu wenig / so beschreib sie mit dem zeichen − minus genannt. Als dann nimb ein lügen von der andern / was da bleibt / behalt für den theiler / multiplicir darnach im Creuz ein falsche zahl mit der andern lügen / nimb eins vom andern / vnd das da bleibt theil ab mit fürgemachtem theiler / so kompt berichtung der frag.

4.5 Das Sekantenverfahren und das Newtonverfahren

Dadurch, daß Bisektionsverfahren und Regula Falsi ein Intervall, an dessen Endpunkten die Funktion unterschiedliche Vorzeichen hat, in ein *kleineres* Intervall umwandeln, das dieselbe Eigenschaft hat, besitzen diese beiden Verfahren eine Art eingebaute "Konvergenzgarantie", aber eben, wie schon mehrfach gesagt, um den Preis, daß man solche Startwerte erst einmal finden muß.

Trotzdem gibt uns die Regula Falsi eine neue Idee mit auf den Weg: Man bildet ja hierbei den "neuen Punkt" z dadurch, daß man eine Gerade, die durch zwei Punkte auf der Funktion gelegt wird, eine sogenannte *Sekante*, mit der "Nulllinie" schneidet. Damit diese Methode – der Schnitt einer Sekante mit der Nulllinie – aber funktioniert, ist es überhaupt nicht nötig, daß diese Sekante durch zwei Stellen geht, an denen die Funktion unterschiedliche Vorzeichen hat. Und schon haben wir ein neues Verfahren erhalten: Wir starten mit *zwei* Punkten x_0 und x_1 und bestimmen x_2, indem wir die Sekante durch f an diesen beiden Punkten mit der Nullinie schneiden. Dann werfen wir x_0 weg und verwenden x_1 und x_2, um auf dieselbe Art und Weise den Punkt x_3 zu berechnen und hoffen, daß die so generierte Folge gegen eine Nullstelle der

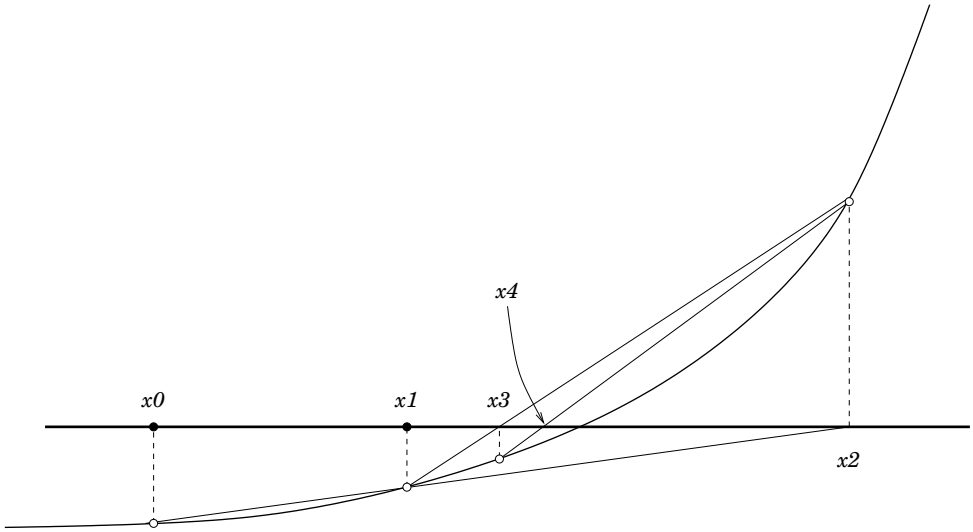


Abbildung 4.7: Die ersten Schritte des Sekantenverfahrens. Die "neuen" Punkte werden ermittelt, indem die beiden letzten "alten" Punkte verbunden werden und die Verbindungslinie mit der Nulllinie geschnitten wird.

Funktion konvergiert. Die Rechenvorschrift beim Sekantenverfahren ist

$$x_{j+1} = \frac{x_j f(x_{j-1}) - x_{j-1} f(x_j)}{f(x_{j-1}) - f(x_j)} = x_j - \frac{x_j - x_{j-1}}{f(x_j) - f(x_{j-1})} f(x_j), \qquad j \in \mathbb{N}. \tag{4.14}$$

Und in der Tat funktioniert diese Methode unter gewissen Voraussetzungen ganz prima.

Satz 4.10 *Ist f in einer Umgebung*[109] *einer* einfachen *Nullstelle*[110] *x^* zweimal stetig differenzierbar und liegen die Anfangswerte x_0, x_1 nahe genug bei x^*, dann konvergiert die Folge x_j, $j \in \mathbb{N}$, aus (4.14) gegen x^* und zwar deutlich schneller als beim Bisektionsverfahren oder bei der Regula Falsi.*

Rücken nun die beiden Punkte, an denen man die Sekante bildet, näher und näher zusammen, so wird, immer vorausgesetzt, die Funktion f ist gutartig, genauer gesagt *differenzierbar*[111], aus der Sekante "in der Grenze" die *Tangente* an die Kurve, die nun nicht mehr in zwei Punkten schneidet, sondern an einem Punkt berührt. Insbesondere ist in diesem Fall

$$\lim_{x_j \to x_{j+1}} \frac{f(x_j) - f(x_{j-1})}{x_j - x_{j-1}} = f'(x_j),$$

[109] Der Begriff "Umgebung" ist in der Mathematik strikt und klar definiert, was uns hier aber nicht weiter interessieren soll.

[110] Also $f'(x^*) \neq 0$, das garantiert einen "echten" Vorzeichenwechsel, die Nulllinie wird hier nicht nur berührt, sondern wirklich überquert.

[111] Das heißt, das diese Funktion dort eine *Ableitung* besitzt, ein Begriff, an den man sich vage aus Schulzeiten erinnern könnte.

was eingesetzt in die Formel (4.14) die Berechnungsvorschrift

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}, \qquad j \in \mathbb{N}, \tag{4.15}$$

liefert, das *Newtonverfahren*. Außerdem hält (4.15) noch eine Überraschung für uns bereit: Der Punkt x_{j+1} ist verschwunden! Wir können also wieder mit *einem* Startwert x_0 beginnen und ausgehend davon unsere Iterationsfolge $x_1, x_2, \ldots$ mittels (4.15) berechen, siehe Abb. 4.8.

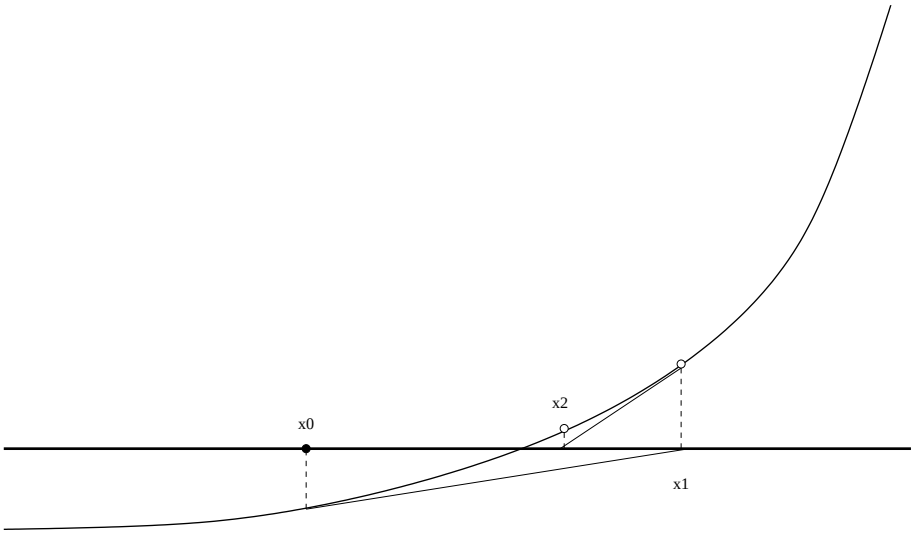


Abbildung 4.8: Die ersten Iterationsschritte des Newton–Verfahrens.

Satz 4.11 *Ist f in einer Umgebung einer einfachen Nullstelle x^* zweimal stetig differenzierbar und liegt der Anfangswert x_0 nahe genug bei x^*, dann konvergiert die Folge x_j, $j \in \mathbb{N}$, aus (4.15) gegen x^* und zwar sogar noch schneller als beim Sekantenverfahren.*

Noch ein paar Bemerkungen:

1. Auch wenn man sich bei diesen beiden Verfahren die Suche nach Punkten sparen kann, an denen die Funktion unterschiedliches Vorzeichen hat, besteht trotzdem noch ein Problem: Der Startwert muß bereits hinreichend nahe bei der Nullstelle liegen.

2. Heimlich, still und leise sind die Kontraktionen zurückgekehrt! Schaut man sich Iterationsfunktionen

$$F(x) = x - \frac{x - x'}{f(x) - f(x')}\, f(x) \qquad \text{bzw.} \qquad F(x) = x - \frac{f(x)}{f'(x)}$$

aus (4.14) und (4.15) einmal genau an, dann ist jede Nullstelle x^* von f ein Fixpunkt dieser Iterationsfunktionen. Und tatsächlich stammen die mysteriösen "Umgebungen" aus den Konvergenzsätzen von einer Forderung nach *lokaler Kontraktivität*.

3. Passiert es während der Iteration, daß $f(x_{j-1}) = f(x_j)$ (Sekantenverfahren) oder $f'(x_j) = 0$ (Newtonverfahren), dann hat man ein Problem, denn man teilt durch Null und das Verfahren muß abgebrochen werden. Noch schlimmer ist es aber, wenn nicht Gleichheit gilt, sondern nur "fast" Gleichheit, denn dann wird x_{j+1} dadurch gebildet, daß riesige Zahlen von x_j abgezogen werden und der Punkt landet also "ganz weit draußen", von wo das Iterationsverfahren dann wieder von vorne beginnen kann.

Beispiel 4.12 *Das Newtonverfahren in seiner vollen Allgemeinheit geht wohl auf Newton*[112] *zurück – kein Wunder, denn vorher war ja noch nicht einmal der Begriff der Ableitung da*[113]*. Trotzdem ist es in Spezialfällen älter, zum Beispiel, wenn man die Nullstelle der Funktion* $f(x) = x^2 - a$ *berechnen will, denn dann ist*[114] $f'(x) = 2x$ *und somit*

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)} = x_j - \frac{x_j^2 - a}{2x_j} = \frac{2x_j^2 - x_j^2 + a}{2x_j} = \frac{1}{2}\left(x_j + \frac{a}{x_j}\right),$$

was nichts anderes ist als unser gutes altes Heron–Verfahren aus Beispiel 4.8. Dieses konvergiert übrigens, wie man zeigen kann, für alle *Startwerte* $x_0 \geq \sqrt{a}$ *gegen* $\sqrt{a}$*.*

[112] Isaac Newton, ???, Mathematiker und Physiker in Cambridge, später auch "Master of mint", wobei "mint" "Münze" bedeutet!

[113] Der Begriff der Ableitung, ja die gesamte Infinitesimalrechnung wurden – unabhängig voneinander – von Newton und Leibnitz entwickelt.

[114] Schulbildung!

Thou shalt not covet, but tradition
Approves all forms of competition.

"The latest decalogue",
Arthur Hugh Clough, 1862

5 Unter– und überbestimmte Gleichungssysteme

Jetzt aber wieder zurück zu unseren linearen Gleichungssystemen der Form (3.11)

$$Ax = b, \qquad A \in \mathbb{R}^{m\times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m,$$

fordern jetzt aber nicht mehr unbedingt, daß $m = n$ sein soll – eine kleine Erweiterung mit grosser Wirkung, denn die ganze Theorie mit eindeutiger Lösbarkeit und so klappte und klappt eben nur dann, wenn $m = n$ ist.

Erinnern wir uns nochmal, wofür die Größen m und n stehen, nämlich für die Anzahl der Gleichungen und die Anzahl der Variablen. Geht man davon aus, daß immer eine Gleichung eine Variable "bestimmt", so nennt man ein lineares Gleichungssystem

überbestimmt, wenn $m > n$ ist, also mehr Bestimmungsgleichungen da sind als Variablen die bestimmt werden können,

unterbestimmt, wenn $m < n$ ist, wenn also nicht alle Variablen durch Gleichungen festgelegt werden können.

Der leichtere Fall ist in der Tat der Fall, daß wir es mit einem unterbestimmten Gleichungssystem zu tun haben, denn, salopp gesagt, können wir so ein Gleichungssystem lösen, indem wir uns für bestimmte Variablen einfach nicht interessieren. Mathematisch etwas exakter können wir im "Normalfall"[115] immer m Spalten, sagen wir mit Index $j_1, \ldots, j_m$, finden[116], so daß die resultierende $m \times m$–Matrix invertierbar ist:

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} & \ldots & \boxed{a_{1,j_1}} & \ldots & \boxed{a_{1,j_m}} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots & \ldots & \vdots & \ldots & \vdots \\ a_{m1} & \ldots & \boxed{a_{m,j_1}} & \ldots & \boxed{a_{m,j_m}} & \ldots & a_{mn} \end{bmatrix}$$

Wenn wir nun unsere Variablen $x_1, \ldots, x_n$ geeignet umnumeriert haben, können wir annehmen, daß dies die Spalten $1, \ldots, m$ sind, also unsere Matrix A von der Form

$$A = \begin{bmatrix} \boxed{a_{11}} & \ldots & \boxed{a_{1m}} & a_{1,m+1} & \ldots & a_{1n} \\ \boxed{\vdots} & \ddots & \boxed{\vdots} & \vdots & \ddots & \vdots \\ \boxed{a_{m1}} & \ldots & \boxed{a_{mm}} & a_{m,m+1} & \ldots & a_{mn} \end{bmatrix}$$

[115] Das entspricht der Invertierbarkeit einer quadratischen Matrix und bedeutet, daß man jede Matrix um einen beliebig kleinen Betrag stören kann und diese Eigenschaft erhält sowie daß diese Eigenschaft sich von hinreichend kleinen Störungen nicht stören läßt.

[116] Zur Erinnerung: m, die Anzahl der Zeilen, ist hier ja kleiner als n, die Anzahl der Spalten

ist, wobei die Matrix $\widetilde{A} = \left[a_{jk} : \begin{smallmatrix} j=1,\ldots,m \\ k=1,\ldots,m \end{smallmatrix}\right]$ nun *invertierbar* ist. Und nun ist es ganz einfach, $Ax = b$ zu lösen:

Man bestimme $\widetilde{x} = [x_1, \ldots, x_m]^T$ als Lösung von $\widetilde{A}\widetilde{x} = b$, dann ist

$$x = \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \in \mathbb{R}^n$$

natürlich eine Lösung von $Ax = \widetilde{A}\widetilde{x} = b$.

Wesentlich interessanter ist aber der Fall eines *überbestimmten* Gleichungssystems, denn die haben im Normalfall *gar keine* Lösung. Es muß schon viel zusammenkommen (z.B. $b = 0$), wenn so ein Gleichungssytem lösbar sein soll.

Übung 5.1 Geben Sie je eine rechte Seite $b \neq 0$ an, für die das Gleichungssystem $Ax = b$ eine oder keine Lösung hat, wobei

$$A = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Beispiel 5.1 *Ein typisches*[117] *Problem aus statistischen Anwendungen besteht darin, durch eine Reihe von (ungenauen) Meßdaten $(x_1, y_1), \ldots, (x_n, y_n)$, die von einem näherungsweise linearen Bildungsgesetz stammen, das heißt*

$$y_j = ax_j + b, \qquad j = 1, \ldots, n$$

für unbekannte Modellparameter a und b. Ziel ist es dann, aus den Meßwerten die Modellparameter zu bestimmen. Das führt uns zu dem im Falle $n \gg 2$ grandios überbestimmten Gleichungssystem

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{=:A} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{=:x} = \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{=:b}.$$

Nun wäre das Leben ja einfach, wenn die Daten wirklich von einer Geraden stammen würden. Oftmals spielen aber sowohl Meß- wie auch Modellfehler[118] *eine nicht zu vernachlässigende Rolle, das heißt, das Gleichungssystem ist nicht mehr lösbar.*
Machen wir's nun ganz konkret und setzen in `Octave`*–Notation*

[117] Aber natürlich stark vereinfachtes

[118] Beispielsweise könnte die "lineare" Abhängigkeit nur in erster Näherung oder unter Vernachlässigung vieler nicht so entscheidender aber dennoch existenter Zusatzfaktoren existieren.

```
octave> x = (0:0.1:1)';
```

und

```
octave> y = 3*x + ones(11,1) + 0.1 * sin(10*x);
```

Anders gesagt, wir betrachten die Funktion $3 * a + b$*, aber eben "leicht" gestort, nämlich um* $\frac{\sin(10a)}{10}$.

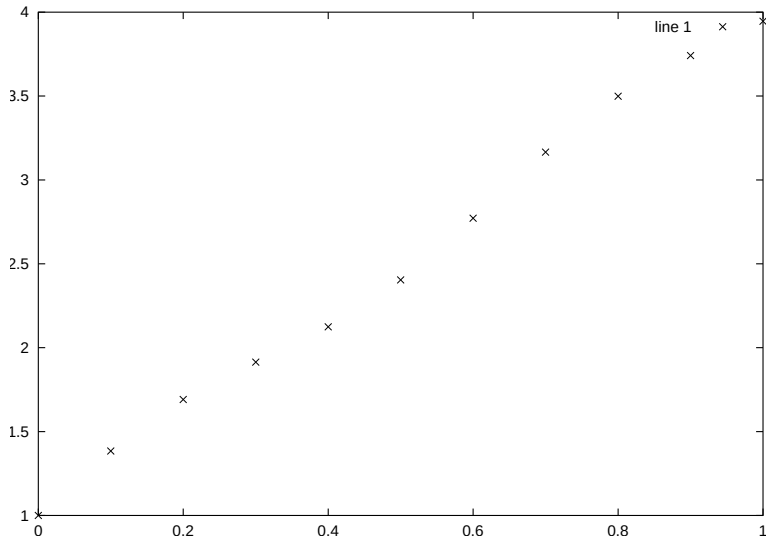


Abbildung 5.1: Verteilung der "nahezu" linearen Daten aus Beispiel 5.1.

Was also tun? Nun, wenn wir ein Gleichungssystem nicht *exakt* lösen können, dann wählen wir eben x so, daß es "so gut wie möglich" gelöst wird, daß also der Fehler $r = b - Ax$ so klein wie möglich wird. Bei "Größe" im Zusammenhang mit Vektoren erinnern wir uns[119] natürlich sofort an den Begriff der *Norm* und können unser Problem also wie folgt konkretisieren:

Bestimme x *so, daß*

$$\|b - Ax\| = \min_{y \in \mathbb{R}^n} \|b - Ay\|, \tag{5.1}$$

wobei $\|\cdot\|$ *eine vorgegeben Norm ist.*

Die Natur der Norm, also der verwendet Abstandsbegriff, spielt eine ganz gewaltige Rolle bei der Lösung und es ist durchaus vernünftig, in manchen Anwendungen die eine, in anderen Anwendungen aber eine andere Norm zu verwenden. Hier wollen wir uns aber auf die *euklidische* Norm $\|x\|_2 = \sqrt{x^T x}$ beschränken, bei der die Lösung des Minimierungsproblems eine besonders schöne geometrische Interpretation besitzt und auch numerisch besonders gut zu berechnen ist. Unser Problem ist also:

[119] Hoffentlich!

Bestimme x so, daß

$$\|b - Ax\|_2 = \min_{y \in \mathbb{R}^n} \|b - Ay\|_2 . \tag{5.2}$$

Es gibt nun zwei Möglichkeiten, wie wir unser Minimum aus (5.2) bestimmen, entweder, indem wir analytisch vorgehen und (5.2) ausmultiplizieren[120], nach x ableiten[121] und $= 0$ setzen, oder, indem wir geometrisch argumentieren – wir wollen hier letzteres machen, weil es einfach anschaulicher ist.

Wenn wir eine Matrix $A \in \mathbb{R}^{m \times n}$ haben, so können wir uns die Menge aller Punkte im $\mathbb{R}^m$ anschauen, die wir bekommen, indem wir *alle* möglichen Punkte $x \in \mathbb{R}^n$ von rechts dranmultiplizieren, also

$$\{y = Ax \ : \ x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m.$$

Zwei solcher Gebilde, sogenannte *lineare Teilräume* oder *lineare Unterräume* sind in Abb. 5.2

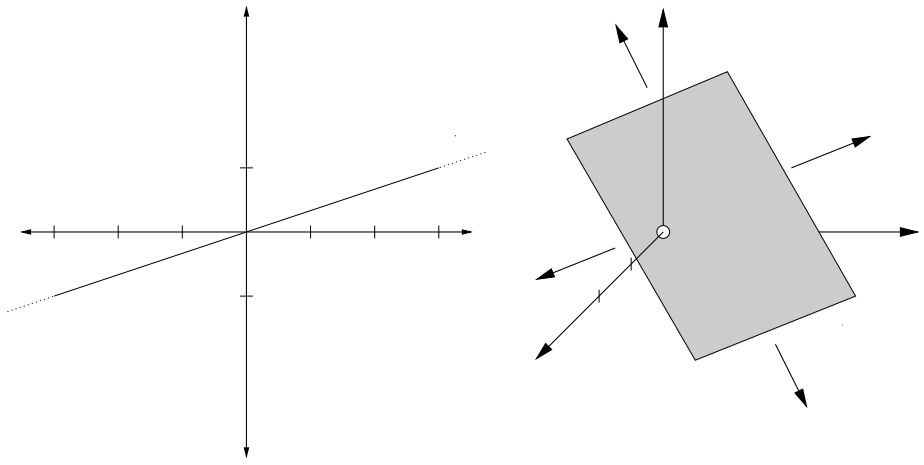


Abbildung 5.2: Zwei lineare Teilräume. Links der Fall $m = 2$,$n = 1$, rechts der Fall $m = 3$, $n = 2$. Man beachte: solche Geraden/Ebenen enthalten *immer* den Nullpunkt, da $A0 = 0$.

gezeigt. Im einfachsten Fall, $n = 1$, erhält man immer die Gerade, die vom "Spaltenvektor" $A \in \mathbb{R}^{m \times 1}$ und seinen Vielfachen erzeugt wird, in höheren Dimensionen erhält man Ebenen. Wie erhält man nun denjenigen Punkt der Form $y = Ax$, der am nächsten bei einem vorgegebenen $b \in \mathbb{R}^m$ liegt – dieser löst ja schließlich unser Minimierungsproblem. Wie Abb. 5.3 zeigt, ist das derjenige Punkt $y = Ax$ für den der Fehlervektor $b - y = b - Ax$ auf $y = Ax$ *senkrecht* steht, für den also

$$0 = (b - Ax)^T Ax$$

gilt. Nun, $b - y$ steht aber genau senkrecht auf *alle* Vektoren der Form Ax, $x \in \mathbb{R}^n$, wenn

$$(b - y)^T A = 0 \tag{5.3}$$

[120] Genauer, indem wir $\|b - Ax\|_2^2 = (b - Ax)^T (b - Ax)$ berechnen, das ist ein nach oben offenes Paraboloid.

[121] Das geht nicht nur für "einfache" x, sondern auch für Vektoren; das Ergebnis ist dann allerdings ein Vektor!

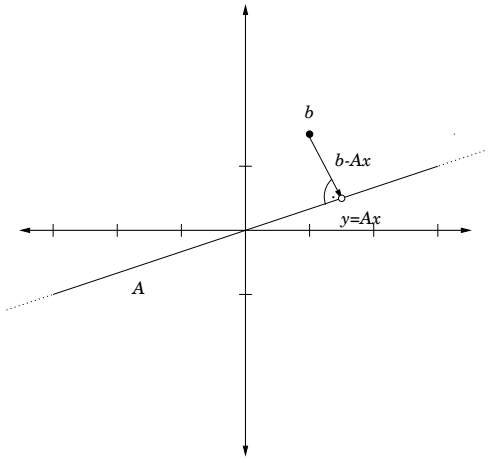


Abbildung 5.3: Der Punkt auf der Gerade, der am nächsten bei b liegt, ist der Fußpunkt des Lots von b auf die Gerade $A\mathbb{R}$. .

Ist nämlich (5.3) erfüllt, dann ist für beliebiges $x \in \mathbb{R}^n$

$$(b-y)^T Ax = \underbrace{\left((b-y)^T A\right)}_{=0^T} x = 0.$$

Ist umgekehrt $x := A^T(b-y) \neq 0$ (und das ist ein Vektor aus $\mathbb{R}^n$ wie man sich einfach überlegt), so ist

$$(b-y)^T Ax = \underbrace{\left((b-y)^T A\right)}_{=(A^T(b-y))^T} A^T(b-y) = \left\|A^T(b-y)\right\|_2^2 > 0.$$

Also muß unsere "Optimallösung" $y = Ax$ die Gleichung (5.3) erfüllen, was uns[122]

$$0 = \left((b-Ax)^T A\right)^T = A^T(b-Ax) = A^T - A^T Ax$$

und somit die *Normalengleichung*

$$A^T Ax = A^T b, \qquad A^T A \in \mathbb{R}^{n\times n}, \tag{5.4}$$

liefert, deren Lösung $x \in \mathbb{R}^n$ gleichzeitig die Lösung unseres *Ausgleichsproblems* oder Minimierungsproblems (5.2) ist. Und damit sind wir natürlich "back in business", denn wie man Gleichungssysteme löst, das wissen wir ja inzwischen, so daß Programm 5.1 eine Lösung für das Minimierungsproblem liefert.

Beispiel 5.2 *Nun können wir die "optimale" Gerade aus Beispiel 5.1 bestimmen, indem wir die Matrix*

[122] Die Transponierte jeder beliebigen, auch nichtquadratischen Nullmatrix ist wieder eine Nullmatrix.

```
%%  Ausgleich.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Loese Ausgleichsproblem durch Loesung der Normalengleichung
%%  Eingabe:
%%    A    Matrix
%%    b    rechte Seit

function x = Ausgleich( A,b )
  Y = A' * A;
  y = A' * b;

  %% Eingebaute Loesungsroutine von Octave ...
  x = A \ y;
```

Programm 5.1 `Ausgleich.m`: Lösung des Minimierungsproblems über die Normalengleichung (5.4). Wir verwenden hier die *eingebaute* Lösungsroutine von `Matlab`bzw. `Octave`, die ist auf jeden Fall schneller und auch stabiler (da wesentlich ausgetüftelter) als unsere "selbstgestrickten" Verfahren.

```
octave> A = [ x,ones( length(x), 1 ) ];
```

bestimmen und dann über

```
octave> a = Ausgleich ( A,y )
a =

  2.9756
  1.0250
```

die Parameter bestimmen, die, wie man sieht, noch sehr nahe bei den Werten der "Ausgangsgeraden" liegen. Die Qualität dieser Approximation sieht man in Abb 5.4 und tatsächlich trügt die recht gute Optik nicht, denn der quadratische Fehler ergibt sich numerisch als

```
octave> z = y - ( a(1)*x + a(2)*ones(11,1) ); z'*z
ans = 0.047547
```

Das heißt also, wir haben unser Problem gelöst und können es getrost ad actas legen? Schön wäre es, aber jetzt schlägt der Unterschied zwischen theoretischer und praktischer Lösung eines Problems durch. Ganz abgesehen davon, daß die Berechnung von $A^T A$ ja durchaus Zeit und Rechenleistung kostet, bereiten uns die numerischen Eigenschaften dieser Matrix erheblich mehr Kopfzerbrechen:

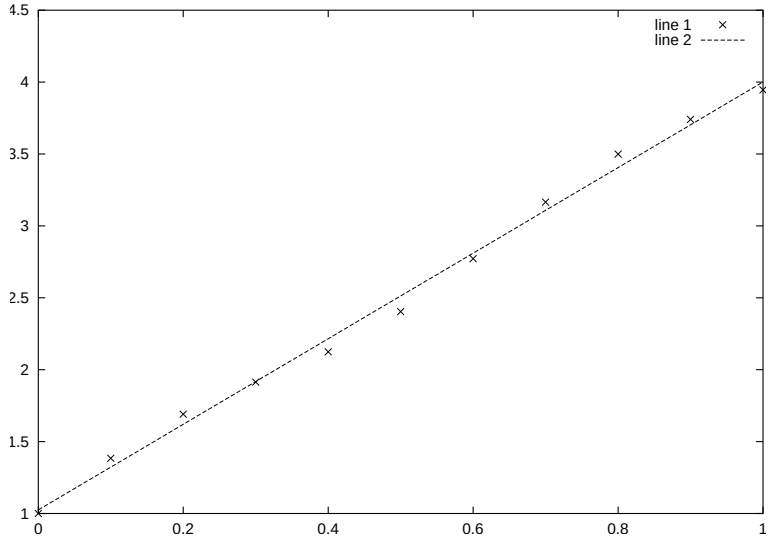


Abbildung 5.4: Die “optimale” lineare Funktion aus Beispiel 5.2 und ihre Approximationsgüte. .

Die Konditionszahl der Matrix $A^T A$ beträgt

$$\kappa_2\left(A^T A\right) \sim \kappa_2\left(A\right)^2.$$

Mit anderen Worten: Der Fehler bei der numerischen Lösung des Minimierungsproblems über die Normalengleichung kann dramatisch steigen! Man kann übrigens sehr einfach $n+1 \times n$–Matrizen angeben, für die tatschlich $\kappa_2\left(A^T A\right) = \kappa_2\left(A\right)^2$ gilt.
Aber, und das ist die gute Nachricht, wir müssen nicht von vorne anfangen, sondern uns lediglich ein anderes Hilfsmittel besorgen, nämlich eine passende Zerlegung. Welche Zerlegung[123] hier mehr Sinn macht, das sehen wir ziemlich schnell, wenn wir erkennen, daß

$$\left\|Q^T x\right\|_2 = \sqrt{x^T Q^T Q x} = \sqrt{x^T I x} = \sqrt{x^T x} = \left\|x\right\|_2,$$

wenn nur Q eine *orthogonale* Matrix ist. Diese Beobachtung führt letztendlich dazu[124], daß für alle orthogonalen Matrizen $Q \in \mathbb{R}^m$, $Q' \in \mathbb{R}^n$ und für jede Matrix $A \in \mathbb{R}^{m \times n}$

$$\kappa_2\left(QAQ'\right) = \kappa_2(A).$$

gilt:

Multiplikation mit orthogonalen Matrizen verändert[125] *die Konditionszahl nicht. Deswegen sagt man, die Konditionszahl κ_2 einer Matrix ist* orthogonal invariant.

[123] Nun gut, so viele kennen wir nun auch wieder nicht, aber zwei sind es doch!
[124] Ist gar nicht so schwer, wer will, kann’s gerne mal versuchen.
[125] Und in diesem Kontext ist “Veränderung” normalerweise synonym für “Verschlechterung”.

```
%%  JacobiQR2.m (Numerik HaF)
%%  ------------------------------------------------------
%%  Berechne QR-Zerlegung mittels Jacobirotationen
%%  Version 2 fuer nichtquadratische Matrizen
%%  Ueberschreiben der Matrix A
%%  Eingabe:
%%    A    Matrix

function [Q,R] = JacobiQR( A )
  [m,n] = size( A );
  Q = eye( m );

  for j = 1:n-1
    for k = j+1:m
      Qjk = RotMat( [ A( j,j ), A( k,j ) ],j,k,n );
      A = Qjk * A;
      Q = Q * Qjk';
    end
  end

  R = A;
```

Programm 5.2 `JacobiQR2.m`: Noch eine QR–Zerlegung, diesmal aber auch für nichtquadratische Matrizen.

Probieren wir es also einmal mit der QR–Zerlegung. Dabei halten wir zuerst fest, daß es für die Vorgehensweise dort völlig egal war, ob die Matrix quadratisch war oder nicht, denn man ist einfach Spalte für Spalte vorgegangen. Die entsprechende Variante unseres "Drehverfahrens" ist in Programm 5.2 angegeben. Allerdings hat jetzt die QR–Zerlegung die Form

$$\mathbb{R}^{m\times n} \ni A = QR = \underbrace{\begin{bmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & \dots & * \end{bmatrix}}_{\in\mathbb{R}^{m\times m}} \underbrace{\begin{bmatrix} * & \dots & * \\ & \ddots & \vdots \\ 0 & & * \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}}_{\in\mathbb{R}^{m\times n}}, \tag{5.5}$$

das heißt, die Matrix R besteht aus einer $n \times n$–Rechtsdreiecksmatrix, unter die noch $m - n$

```
##  Ausgleich2.m (Numerik HaF)
##  ----------------------------------------------------
##  Loese Ausgleichsproblem ueber QR-Zerlegung
##  Eingabe:
##    A    Matrix
##    b    rechte Seite

function x = Ausgleich2( A,b )
  [m,n] = size( A );
  [Q,R] = JacobiQR2( A );

  RR = R( 1:n,1:n );
  bb = Q' * b;

  x = RR \ bb( 1:n );
```

Programm 5.3 `Ausgleich2.m`: Lösung des Ausgleichsproblems über eine QR–Zerlegung. .

Leerzeilen gepackt sind. Mit so einer Zerlegung ist nun

$$A^TA = R^T \underbrace{Q^TQ}_{=I} R = \underbrace{\left[\widetilde{R}^T\, 0\right]}_{=R^T} \underbrace{\begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix}}_{=R} = \widetilde{R}^T\widetilde{R}$$

Also wird unsere Normalengleichung zu

$$A^TAx = A^Tb \qquad \Longleftrightarrow \qquad \widetilde{R}^T\widetilde{R}x = R^TQ^Tb = \widetilde{R}^T \left[\left(Q^Tb\right)_j \,:\, j = 1,\ldots,n\right],$$

was wir unter der Voraussetzung daß $\widetilde{R}$ invertierbar ist[126] sogar zu

$$\widetilde{R}\,x = \left[\left(Q^Tb\right)_j \,:\, j = 1,\ldots,n\right] \tag{5.6}$$

vereinfachen können und das wir nun numerisch stabil mit einer einfachen Rücksubstitution lösen können. Dieses Verfahren ist in Programm 5.3 angegeben.

Beispiel 5.3 *Lösen wir Beispiel 5.1 mittels Programm 5.3, erhalten wir*

```
octave> a = Ausgleich2( A,y )
a =

  3.0113
  1.0000
```

[126] Unser "Normalfall"!

```
##  Ausgleich3.m (Numerik HaF)
##  ----------------------------------------------------------
##  Loese Ausgleichsproblem ueber QR-Zerlegung
##  Eingabe:
##    A    Matrix
##    b    rechte Seite

function x = Ausgleich3( A,b )
  [m,n] = size( A );
  [Q,R] = qr( A );

  RR = R( 1:n,1:n );
  bb = Q' * b;

  x = RR \ bb( 1:n );
```

Programm 5.4 `Ausgleich3.m`: Lösung des Ausgleichsproblems unter Verwendung der "eingebauten" QR–Zerlegung.

und erhalten einen Fehler von

```
octave> z = y - ( a2(1)*x + a2(2)*ones(11,1) ); z'*z
ans = 0.049519
```

also etwas schlechter *als in Beispiel 5.2! Ist also unsere Theorie falsch? Nein, natürlich nicht, denn wir müssen berücksichtigen, daß wir in Programm 5.1 ausschließlich auf die schon sehr ausgefeilten Lösungsmethoden*[127] *von* `Octave` *zurückgreifen konnten, während wir uns in Programm 5.3 auf unsere "Hausmacher"–QR–Zerlegung verlassen mußten, die zwar einfach und anschaulich, aber weder besonders effizient noch besonders stabil ist*[128]*. Benutzen wir hingegen die "eingebaute" QR–Zerlegung von* `Octave`[129] *wie in Programm 5.4, dann bekommen wir wieder dasselbe Ergebnis wie in Beispiel 5.2.*

Trotzdem steht unsere Theorie im Moment noch auf sehr wackligen Füßen, denn alles was wir bisher geschafft haben, war zu zeigen, daß die "bessere" Methode zumindest mal nicht schlechter funktioniert als die "schlechtere". Hier ist zumindest mal ein Extrembeispiel, das zeigt, daß wirklich böse Sachen passieren können.

[127] Tatsächlich sind wir hier nicht so "optimal" wie wir sein könnten. Das allgemeine Lösungsverfahren basiert auf einer Gauß–Elimination mit Spaltenpivotsuche und Skalierung, für *symmetrische, positiv definite* Matrizen wie $A^T A$ gibt es aber ein spezialisiertes Verfahren, das sogenannte *Cholesky–Verfahren*. Um das generelle Problem der schlechteren Konditioniertheit kommen wir aber auch mit diesem Verfahren nicht herum!

[128] Was ausschließlich an der Implementierung liegt!

[129] Und das ist die generelle "Message" hier: Wer das Rad neu erfinden will, der muß von Rädern eine Menge verstehen, und zwar auch viel mehr als man in einer "normalen" Mathematik–Grundvorlesung lernen würde.

Beispiel 5.4 *Für $n \in \mathbb{N}$ und $t > 0$ betrachten wir die Matrix*

$$\begin{bmatrix} 1 & \dots & 1 \\ t & & \\ & \ddots & \\ & & t \end{bmatrix}.$$

Vergleichen wir nun die Fehler $\|b - Ax\|_2^2$ für die beiden Verfahren aus den Programmen 5.1 und 5.4, so erhalten wir für eine bestimmte rechte Seite[130] *die folgenden Resultate*

n	t	*Normalform (Prog. 5.1)*	*QR–Zerlegung (Prog. 5.4)*
10	10^{-3}	0.19915	0.19915
10	10^{-6}	0.19915	0.19915
10	10^{-9}	5.00143	0.19915
100	10^{-3}	0.00016172	0.00016172
100	10^{-6}	0.00016212	0.00016172
100	10^{-9}	5.0268×10^{1}	1.6172×10^{-4}
1000	10^{-6}	0.00066650	0.00066255

Das "seltsame" an diesem Beispiel ist die Tatsache, daß die beiden Verfahren lange mehr oder weniger "gleichwertig" sind und daß erst für $t \sim 10^{-8}$ etwas passiert, dann aber richtig. Übrigens liefert uns `Octave` *in diesen Fällen auch gleich noch die Information, was passiert, denn wir erhalten beispielsweise für $n = 100$, $t = 10^{-7}$ die Meldung*

```
warning: matrix singular to machine precision, rcond = 9.99201e-17
```

womit natürlich die Matrix $A^T A$ gemeint ist, die bei der Normalengleichung auftaucht. Um hingegen einen "graduellen" Übergang zu finden, muß der Computer schon ganz schön rechnen ($n = 1000$).

[130] Unter Verwendung der mehr oder weniger "zufälligen" Auswahl `!b = sin( (0:n) )'`.

Uns ist in alten mæren
wunders viel geseit
von Helden lobebæren
von grôzer arebeit

Das Nibelungenlied

Literatur

[1] N. Faletta. *Paradoxon*. Fischer Logo. Fischer Taschenbuch Verlag, 1988.

[2] W. Gautschi. *Numerical Analysis. An Introduction*. Birkhäuser, 1997.

[3] D. Goldberg. What every computer scientist should know about floating–point arithmetic. *ACM Computing Surveys*, **23** (1991), 5–48.

[4] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 1996.

[5] C. G. J. Jacobi. Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen. *Crelle's Journal*, **30** (1846), 51–94.

[6] D. Jörgensen. *Der Rechenmeister*. Rütten & Loenig, 1999. Roman.

[7] W. Kahan. How Cray's arithmetic hurts scientific computation (and what might be done about it). Manuscript for Cray User Group meeting, Toronto, 1990.

[8] J. Oates. *Babylon*. Thames and Hudson, 1979. Deutsche Ausgabe: Gustav Lübbe Verlag, 1983. Lizenzausgabe für Gondrom Verlag, 1990.

[9] W. Popp. *Wege des exakten Denkens. Vier Jahrtausende Mathematik*. Weltbild Verlag, 1987. Originalausgabe Franz Ehrenwirth Verlag, 1981.

[10] A. Riese. *Rechenbuch / auff Linien und Ziphren / in allerley Handhierung / Geschäfften unnd Kauffmannschafft*. Franck. Bey. Chr. Egen. Erben, 1574. Facsimile: Verlag Th. Schäfer, Hannover, 1987.

[11] T. Sauer. Numerische Mathematik I. Vorlesungsskript, Friedrich–Alexander–Universität Erlangen–Nürnberg, Justus–Liebig–Universität Gießen, 2000. `http://www.math.uni-giessen.de/tomas.sauer`.

[12] M. Woitschach. *Gödel, Götzen und Computer. Eine Kritik der unreinen Vernunft*. Horst Poller Verlag, Stuttgart, 1986.

Index