Vorlesung Numerische Mathematik 1 & 2
Studiengang Informatik der ZHAW

Dr. sc. nat. Reto Knaack

16. September 2016

Vorbemerkung

Das vorliegende Skript wurde, basierend auf einer früheren Version, für die im Studienjahr 14/15 stattfindenden Vorlesungen Numerische Mathematik 1 & 2 im Studiengang Informatik der ZHAW komplett überarbeitet.

Das Skript basiert auf den folgenden Quellen:

- [1] 'Numerische Mathematik: Eine beispielorientierte Einführung' von Michael Knorrenschild (3. Auflage)
- [2] 'Computer Mathematik' von Walter Gander (online verfügbar unter http://www.abz.inf. ethz.ch/index.php?page=76).
- [3] Skript 'Numerische Mathematik I' von Lars Grüne (Mathematisches Institut, Universität Bayreuth)
- [4] Skript 'Einführung in die Numerische Mathematik' von F. Natterer (Institut für Numerische und instrumentelle Mathematik, Universität Münster),
- [5] Skript 'Numerische Mathematik 1' von E. Novak (Universität Jena).
- [6] 'Numerik Algorithmen', G. Englen-Müllges, K. Niederdrenk, R. Wodicka, Springer-Verlag (10. Auflage), 2011
- [7] 'Numerik für Informatiker', Huckle, Schneider, Springer, 2002
- [8] 'Mathematik für Ingenieure und Naturwissenschaftler: Band 2', L. Papula, Vieweg + Teubner Verlag (13. Auflage), 2012
- [9] 'Numerical Methods for Engineers and Scientists', A. Gilat, V. Subramaniam, Wiley, 2014
- [10] 'Numerische Mathematik', Folien zur Vorlesung, R. Massjung, 2013
- [11] 'Einführung in die Theorie der Fourierreihen und Fouriertransformationen', A. Spaenhauer, M. Steiner-Curtis, Skript, 2014

Teilweise wurden Text und Abbildungen eins zu eins aus den obigen Quellen übernommen, insbesondere aus [1] und [7]. Das Skript ist deshalb nicht als eigenständiges Werk zu sehen, sondern als Unterrichts- und Arbeitshilfe, kondensiert aus einer Vielzahl anderer Skripte und Bücher.

Es wurde versucht, die behandelten Konzepte und Algorithmen in einen historischen Kontext zu setzen, um die Verbindungen der Numerischen Mathematik mit der Informatik und der Entwicklung des Computers hervorzuheben. Die Kaptiel zur historischen Entwicklung sind in diesem Sinne als Hintergrundinformationen zu verstehen:

"It is an extremely useful thing to have knowledge of the true origins of memorable discoveries, especially those that have been found not by accident but by dint of meditation. It is not so much that thereby history may attribute to each man his own discoveries and others should be encouraged to earn like commendation, as that the art of making discoveries should be extended by considering noteworthy examples of it."[1]

[1] Leibniz, in the opening paragraph of his Historia et Origo Calculi Differentialis [1]. The quote itself was taken from Erik Meijering (2002), "A Chronology of Interpolation".

Inhaltsverzeichnis

in letzter Prüfung

[2] Hauptsächlich gemäss Kap. 1 aus [7], erweitert mit Zusatzinformationen aus Wikipedia.
[3] Übernommen in gekürzter und leicht abgeänderter Form von Kap. 1 und Kap. 5 aus [7]
[4] Kapitel hauptsächlich übernommen aus [3]

[5] Hauptsächlich gemäss Wikipedia und http://www-history.mcs.st-andrews.ac.uk/HistTopics/The_rise_of_calculus.html

[6] Gemäss Erik Merijering (2002), "A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing."
[7] Übernommen aus Wikipedia:http://de.wikipedia.org/wiki/Methode_der_kleinsten_Quadrate

Kapitel 1

Einführung in die Numerische Mathematik

Dieses Kapitel gibt eine Einführung darin, was Numerische Mathematik ist, wo die Verbindungen zur Informatik (und umgekehrt) liegen, wie sie sich im geschichtlichen Kontext entwickelte und was einige typische Fragestellungen sind, die wir im Verlauf der Vorlesung behandeln werden.

Lernziele:

- Sie kennen die Definition der Numerischen Mathematik sowie die wichtigsten Anknüpfungspunkte zur Informatik.
- Sie kennen den geschichtlichen Hintergrund der Entwicklung der Numerischen Mathematik.
- Sie kennen einige der typischen Fragestellungen der Numerischen Mathematik.

1.1 Was ist Numerische Mathematik

Die Numerische Mathematik, kurz Numerik genannt, beschäftigt sich als Teilgebiet der Mathematik mit der Konstruktion und Analyse von Algorithmen für kontinuierliche mathematische Probleme. Hauptanwendung ist dabei die Berechnung von Lösungen mit Hilfe von Computern[1]. Im Gegensatz zur analytischen Rechnung will man bei der Numerik keine geschlossenen Formeln oder algebraische Ausdrücke erhalten, sondern, wie der Name sagt, numerische Resultate. Unter einem Algorithmus verstehen wir dabei "eine endliche Menge genau beschriebener Anweisungen (arithmetische und logische Operationen und Ausführungshinweise), die in einer bestimmten Reihenfolge auszuführen sind, um mit Hilfe der eingegebenen Daten die gesuchten Ausgabedaten zu ermitteln" [6].

Interesse an solchen Algorithmen besteht meist aus einem der folgenden Gründe:

1. Es gibt zu dem Problem keine explizite Lösungsdarstellung (so zum Beispiel bei den Navier-Stokes-Gleichungen oder dem Dreikörperproblem) oder
2. die Lösungsdarstellung existiert, ist jedoch nicht geeignet, um die Lösung schnell zu berechnen oder liegt in einer Form vor, in der Rechenfehler sich stark bemerkbar machen (zum Beispiel bei vielen Potenzreihen).

Unterschieden werden zwei Typen von Verfahren: Einmal direkte, die nach endlicher Zeit bei unendlicher Rechnergenauigkeit die exakte Lösung eines Problems liefern, und auf der anderen Seite Näherungsverfahren, die – wie der Name sagt – nur Approximationen liefern. Ein direktes Verfahren ist beispielsweise das gaußsche Eliminationsverfahren, welches die exakte Lösung eines linearen Gleichungssystems ermöglicht. Näherungsverfahren sind unter anderem Quadraturformeln, die den Wert eines Integrals näherungsweise berechnen oder auch das Newton-Verfahren, das iterativ bessere Approximationen einer Nullstelle einer Funktion liefert. Da in Anwendungen die Lösungen nur auf endliche Genauigkeit benötigt werden, kann ein iteratives Verfahren auch bei der Existenz eines direkten Verfahrens sinnvoller sein, wenn es in kürzerer Zeit diese Genauigkeit liefert. Unterschiedliche Verfahren werden nach Laufzeit, Stabilität und Robustheit verglichen.

[1] Definition gemäss Wikipedia

Die Verbindung der Numerik mit der Informatik ist offensichtlich, ist doch die effiziente Berechnung numerischer Algorithmen ohne Computer meist nicht möglich. Umgekehrt kommt man bei der täglichen Arbeit mit dem Computer zwingend auf direkte oder indirekte Art mit Grundverfahren der Numerik in Berührung. Dies gilt insbesondere in den Bereichen (gemäss [7]):

- Zahldarstellung und -arithmetik
- Implementierung mathematischer Funktionen
- Computergraphik (Darstellung von Objekten)
- Bildverarbeitung (Kompression, Analyse, Bearbeitung)
- Neuronale Netze (Lernverfahren)
- Information Retrieval (Vektorraummodell)
- Chip Design (Algebraische Differentialgleichungen)
- stochastische Automaten und Markov-Ketten (Prozessverwaltung, Warteschlangen)

Die Numerische Mathematik erscheint manchmal als eine nicht sehr übersichtliche Sammlung von Rezepten für eine Vielzahl von numerischen Problemen. Dies ist irreführend. Definiert man die Informatik wie im englischen Sprachgebrauch als die "Wissenschaft vom Computer" (Computer Science), so ist die Numerische Mathematik in natürlicher Weise darin enthalten (vgl. [7]). Dies äusserst sich z.B. im wichtigen Bereich des Wissenschaftlichen Rechnens (Scientific Computing), in dem Numeriker und Informatiker zusammen mit Wissenschaftlern daran arbeiten, komplexe Anwendungsprobleme verschiedener Wissenschaftsgebiete fachübergreifend zu lösen (typisches Beispiel ist die Meteorologie mit Wetter- und Klimamodellen sowie den entsprechenden Vorhersagen). Die Numeriker beschäftigen sich dabei mit der Entwicklung effizienter Algorithmen und Methoden, die das mathematische Problem möglichst gut diskret approximieren. Die Informatiker sind zuständig für die effiziente Implementierung (bzgl. Rechenzeit, Speicherverwaltung, Cache, Parallelisierung etc., vgl. [7]). Allerdings verschwimmen die Grenzen zwischen den Disziplinen in diesem Bereich, d.h. Numeriker und Informatiker benötigen ein umfangreiches Wissen der jeweils anderen Disziplin.

1.2 Historische Entwicklung[2]

Die Anfänge der (numerischen) Mathematik reichen zurück bis zu den frühen Hochkulturen. Das erste logisch aufgebaute Zahlensystem, ein Additionssystem, dürfte auf die Ägypter zurückgehen (ca. 3000 v. Chr.). Die Babylonier kannten zu Beginn des 2. Jahrtausends vor Christus bereits ein Zahlensystem mit der Basis 60. Eine Näherung der Quadratwurzel $\sqrt{2}$ aus dieser Zeit ist in Abb. 1.1 [3] dargestellt,

Wir wollen auf die Entwicklung der Zahlensysteme an dieser Stelle nicht weiter eingehen, eine detaillierte Beschreibung findet sich in Kap. 2.1. Sowohl die Ägypter als auch die Babylonier kannten die vier Grundrechenarten sowie Näherungen für π. In der Antike (ca. 1200 v.Chr. bis ca. 600 n.Chr.) betrieben die Griechen Mathematik als Wissenschaft im Rahmen der Philosophie und prägten die strenge logische Beweisführung. Ihr Augenmerk lag dabei, wie schon bei den Ägyptern und Babyloniern, vorwiegend auf der Geometrie (als eines der Hauptwerke gelten die Schriften des Griechen Euklid von Alexandria, 3. Jhr. v.Chr.; noch heute sprechen wir von euklidischer Geometrie). Arabische Gelehrte bauten auf den griechischen und indischen Erkenntnissen auf (das uns bekannte moderne Dezimalsystem mit der Null wurde von indischen Mathematikern im Zeitraum 3. Jhr. v.Chr. bis 5. Jhr. n.Chr. entwickelt) und verbreiteten dieses Wissen über Spanien und Italien nach Europa.

Ab dem 13. Jhr. n.Chr. konnte sich das schriftliche Rechnen mit den indisch-arabischen Ziffern langsam durchsetzen und die römischen Ziffern verdrängen. Aber selbst der deutsche Rechenmeister Adam Ries (1492 - 1559) beschrieb in seinen Rechenbüchern neben dem schriftlichen Rechnen mit den arabischen Ziffern hauptsächlich noch das Rechnen mit Abakus, Linien und Steinen, das nur wenige, meist Verwaltungsbeamte, Kaufleute und Gelehrte beherrschten.

Michael Stiefel (deutscher Theologe und Mathematiker, 1487-1567) führte bereits kurze Zeit später die negativen Zahlen ein und prägte den Begriff Exponent. Dem Schweizer Jost Bürgi (Uhrmacher, Instrumentenbauer und Hofastronom, 1552-1632) wird die Einführung der Logarithmentafeln zugerechnet (ab 1588), die er 1605 dem Astronomen

[2] Hauptsächlich gemäss Kap. 1 aus [7], erweitert mit Zusatzinformationen aus Wikipedia.

[3] "Ybc7289-bw". Licensed under Creative Commons Attribution 2.5 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Ybc7289-bw.jpg#mediaviewer/File:Ybc7289-bw.jpg

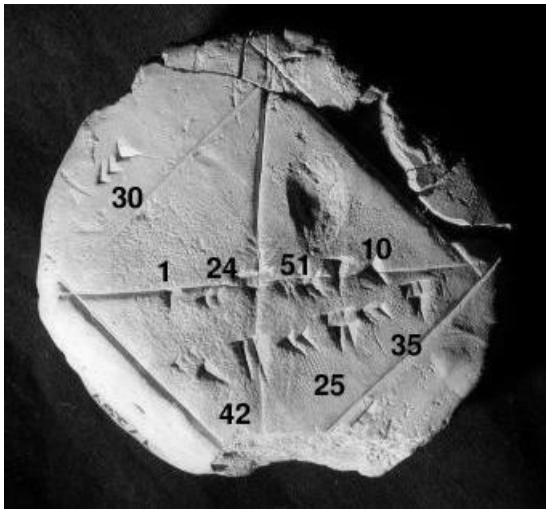


Abbildung 1.1: Babylonische Tontafel YBC 7289 von ca. 1800-1600 v.Chr. Die Näherung von $\sqrt{2}$ ist in der Diagonale eines Quadrates dargestellt mit den Symbolen für $1 + 24/60 + 51/60^2 + 10/60^3 = 1.41421296...$

Kepler (berühmt für seine drei Keplerschen Gesetze, 1571-1630) zugänglich machte. Allerdings publizierte er sie erst 1620, sechs Jahre später als Lord John Napier (schottischer Mathematiker, 1550-1617), der seither als Entdecker des natürlichen Logarithmus gilt. Die von ihm entwickelten Rechenstäbchen mit logarithmischer Skala ("Napier-Bones") erlaubten es, die Multiplikation von Zahlen auf einfache Weise auf die Addition zurückzuführen.[4] Daraus entwickelten die englischen Theologen und Mathematiker Edmund Gunter (1581-1626) und William Oughtred (1574-1660) um 1622 den ersten funktionsfähigen Rechenschieber.

Der Tübinger Universalgelehrte Wilhelm Schickard (1592-1635) entwickelte 1623 die erste Zahnrad-betriebene Rechenmaschine, die die Addition und Subtraktion von bis zu sechsstelligen Zahlen beherrschte und zur Unterstützung von Multiplikation und Division Napier-Rechenstäbchen verwendete. Der Franzose Blaise Pascal (1623-1662) stellte 1642 das erste mechanische Rechenwerk für Addition und Subtraktion mit durchlaufendem Zehnerübertrag her. Im Jahre 1671 entwickelte der deutsche Universalgelehrte Gottfried Willhelm Leibniz (1646-1716) seine Rechenmaschine REPLICA mit zwölfstelligem Anzeigewerk, die alle vier Grundrechenarten beherrschte und beschrieb kurz darauf das binäre Zahlensystem. Zusammen mit Sir Isaac Newton (1643-1727) gilt er als Begründer der Infinitesimalrechnung. Einige der Algorithmen, die wir im Verlauf der Vorlesung kennen lernen werden, gehen auf Newton zurück (z.B. die Nullstellenbestimmung nichtlinearer Funktionen, Polynominterpolation u.a.), die er unter anderem für seine bahnbrechenden Arbeiten in der Mechanik benötigte. Im Jahr 1687 erschien sein Hauptwerk Philosophia Naturalis Principa Mathematica. Noch heute sprechen wir in der Physik in der klassischen Mechanik von den drei Newtonschen Gesetzen.

Auf den Werken von Leibniz und Newton setzte der bedeutende Schweizer Mathematiker und Physiker Leonhard Euler (1707 in Basel geboren, 1783 in Petersburg gestorben) auf[5], der wegen seiner Beiträge in der Analysis und Zahlentheorie und weiteren Teilgebieten der Mathematik Berühmtheit erlangte. Nach ihm sind unter anderem die eulersche Zahl e oder die eulersche Konstante γ benannt (die ausreichend wichtig ist, um ihren Wert bis 2009 auf fast 30 Milliarden dezimale Nachkommastellen zu berechnen). Ein Grossteil der mathematischen Symbolik geht auf Euler zurück (z.B. e, π, die imaginäre Zahl i, $\sum$, $f(x)$). Seine Arbeit über das Königsberger Brückenproblem gilt als eine der ersten Arbeiten auf dem Gebiet der Graphentheorie, heute in der Informatik von grosser Bedeutung in der Komplexitätstheorie von Algorithmen.

Im 18. Jahrhundert wurden die mechanischen Rechenmaschinen weiterentwickelt (z.B. durch Johannes Polenius, Antonius Braun, Philipp Matthäus Hahn). Mit der serienmässigen Fertigung ab 1821 durch Charles Xavier Thomas (1785-1870) in Paris entstanden in der Folge Hunderte von verschiedenartigen mechanischen Rechenmaschinen.

Der englische Mathematiker Charles Babbage (1791-1871) entwarf 1822 seine Differenzenmaschine, die auch Logarithmen berechnen konnte. Die von ihm konzipierte, aber nie gebaute Analytical Engine (ab 1833) gilt als Vorläufer des modernen Computers. Er sah bereits getrennte Baugruppen für Speicher und Rechenwerk und sogar ein Druckwerk vor. Seine enge Mitarbeitern Ada Lovelace (1815-1852), ebenfalls Mathematikerin, beschrieb die Programmierung der Maschine in der Theorie und gilt als erste Programmiererin. Nach ihr wurde die Programmier-

[4] Gemäss $x \cdot y = \exp(\ln(x) + \ln(y))$ bzw. $\ln(x \cdot y) = \ln(x) + \ln(y)$.

[5] Von 1976-1995 auf der Schweizer 10 Franken Note abgebildet.

sprache Ada und die Lovelace-Medal benannt (letztere wird von der British Computer Society seit 1998 vergeben). Die Maschine sollte über Lochkarten gesteuert werden, wie sie von dem französischen Erfinder Jean-Marie Jacquard (1752-1834) zur Steuerung von Webstühlen eingeführt worden waren und der damit entscheidend zur industriellen Revolution beitrug.

Der Zeitgenosse Johann Carl Friedrich Gauss (1777-1855), deutscher Mathematiker, Physiker, Astronom und Geodät, entwickelte eine Vielzahl von Algorithmen, die auch heute noch von Bedeutung sind und seinen Namen tragen. Einige davon werden wir im Rahmen dieser Vorlesung behandeln, so das gausssche Eliminationsverfahren zur exakten Lösung linearer Gleichungssysteme und das Gauss-Seidel Verfahren zur iterativen Lösung. Daneben gibt es noch eine Vielzahl weiterer Algorithmen, die Sie zumindest teilweise in anderen Mathematik-Vorlesungen kennen lernen werden und die auf ihn zurückgehen, z.B. die gausssche Normalverteilung in der Statistik und das entsprechende gausssche Fehlerintegral.

In der Zeit von 1848 bis 1850 entwickelte der englische Mathematiker George Boole (1815-1864) die Boolesche Algebra, die Grundlage der heutigen binären Rechenschaltungen. Die technische Entwicklung wurde 1890 vorangetrieben durch die Lochkartenmaschine des amerikanischen Ingenieurs Herman Hollerith (1860-1929), die zur Volkszählung in den USA entwickelt wurde. In den dazugehörigen Lesegeräten steckten kleine Metallstäbe, die an den Stellen, wo die Karte gelocht war, einen Kontakt zuliessen, so dass elektrischer Strom fliessen konnte. Die Auswertung der Daten dauerte mit dieser Technik statt mehrer Jahre nur einige Wochen.

Der Höhepunkt der mechanischen Rechenmaschinen war wohl mit dem ersten Taschenrechner, der Curta, erreicht. Im Konzentrationslager Buchenwald vollendete der jüdische Häftling Curt Herzstark (1902-1988), ein österreichischer Erfinder und Sohn eines Rechenmaschinenherstellers, die Pläne für seinen Lilliput Rechner, der von der SS als Siegesgeschenk an den Führer vorgesehen war. Wieder in Freiheit wurde Herzstark in Liechtenstein technischer Direktor der Cortina AG zur Herstellung und Vertrieb der Curta, einer verbesserten Form des Lilliput.

Generell wurde während des zweiten Weltkrieges auf beiden Seiten erhebliche Ressourcen für die Entwicklung verbesserter Rechenmaschinen und numerischer Algorithmen eingesetzt. Die 1940er Jahre markieren deshalb den Beginn der 'modernen' Numerischen Mathematik.

Einen Meilenstein hierbei setzte der englische Mathematiker und Kryptoanalytiker Alain Turing (1912-1954) mit seinem Beitrag zur Entschlüsselung der mit der Enigma verschlüsselten deutschen Funksprüche. Unter anderem wegen seiner 1936 erschienenen Arbeit *On Computable Numbers with an Application to the "Entscheidungsproblem"* und dem damit verknüpften Begriff der Turingmaschine gilt er als einer der einflussreichsten Theoretiker der frühen Computerentwicklung und Informatik. Auf der deutschen Seite entwickelte 1941 der Bauingenieur und Erfinder Konrad Zuse (1910-1995) die Z3, den ersten Relaisrechner[6] und damit ersten funktionsfähigen Digitalrechner weltweit. Als Kompromiss zwischen der Festkomma-Zahldarstellung, in der problemlos addiert werden kann, und einer logarithmischen Darstellung, die das Multiplizieren erlaubt, wurde die noch heute übliche Gleitpunktdarstellung verwendet. Der Rechner wurde zur Berechnung von komplexen Matrizen eingesetzt, die in der Aerodynamik (dem sogn. 'Flügelflattern', welches zum Absturz von Flugzeugen führte) eine Rolle spielten. Er wurde bei einem Bombenangriff der Alliierten zerstört und im März 1945, kurz vor der Kapitulation Deutschlands, durch die Zuse Z4 ersetzt[7].

In den USA wurde 1944 der vom amerikanischen Computerpionier Howard Aiken (1900-1973) entwickelte, vollständig aus elektromechanischen Bauteilen zusammengesetzte Grossrechner Mark I in Betrieb genommen (er war ca. 2.5 Meter hoch und ca. 17 Meter lang, sein Gewicht betrug 5 Tonnen). Das Projekt wurde hauptsächlich von IBM finanziert. Der Rechner diente anfänglich dem in die USA geflüchteten, österreich-ungarischen Mathematiker John von Neumann (1903-1957) im Rahmen des Manhatten-Projekts für Berechnungen des Implosionsmechanismus der Plutoniumbombe. Von Neumann entwickelte dazu erste numerische Verfahren zur Lösung von partiellen Differentialgleichungen. 1946 stellte er (in loser Analogie zum menschlichen Hirn) die Fundamentalprinzipien eines frei programmierbaren Rechners auf (Prozessor, Speicher, Programm und Daten im Prozessor). Praktisch alle modernen Rechner beruhen auf von Neumans Rechnerarchitektur.

Ebenfalls 1946 wurde ENIAC (Electronic Numrical Integrator and Computer), der erste rein elektronische Rechner, vorgestellt. Entwickelt worden war er von den amerikanischen Computeringenieuren John Eckert und John Mauchly, welche 1951 mit dem UNIVAC I (Universal Automatic Computer) den ersten in den USA hergestellten kommerziellen Computer auf den Markt brachten. Dieser benötigte eine Leistung von bis zu 125 kW und konnte knapp 2000 Rechenoperationen pro Sekunde ausführen. 1958 entstand der erste integrierte Schaltkreis (ein Flipflop). Die Telefunken TR 4 war 1962 der erste Computer auf Basis von Transistor Bausteinen und 1967 eroberte der erste Taschenrechner den Markt. 1976 wurde der erste Home-Computer Apple I geboren, und ab 1981 begann der

[6] Ein Relais ist ein mit Strom betriebener, elektromagnetischer Schalter mit normalerweise zwei Schaltstellungen

[7] Diese überstand die Kriegswirren und wurde von 1950 bis 1955 an der ETH als zentraler Rechner eingesetzt. Sie wurde durch die Eigenentwicklung ERMETH (Elektronische Rechenmaschine der ETH) ersetzt, welche 1956-1963 in Betrieb war.

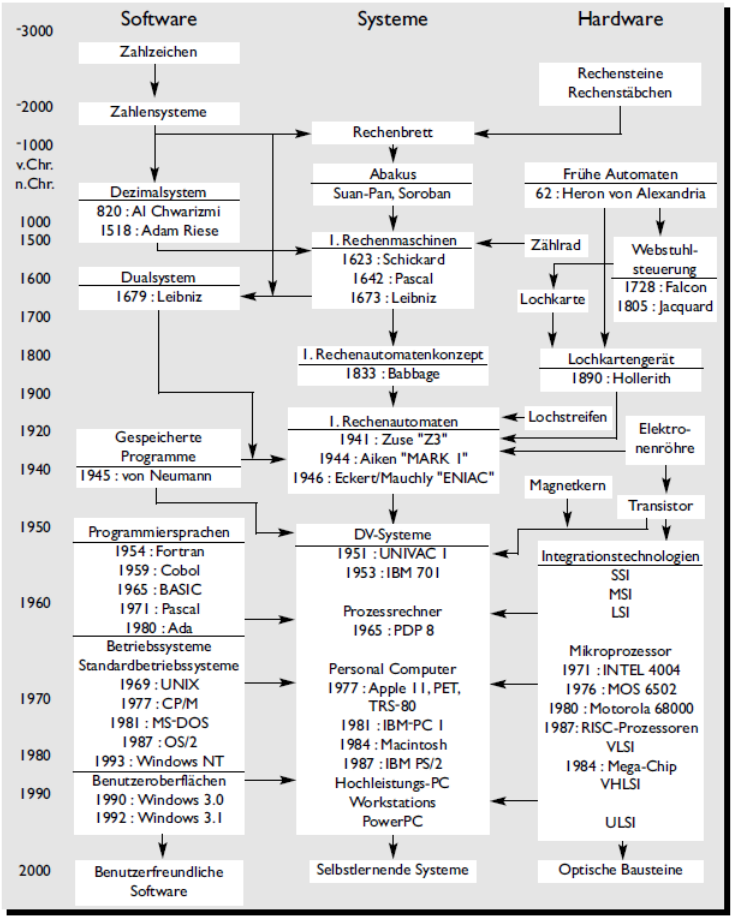


Abbildung 1.2: Zur geschichtlichen Entwicklung des Computers (aus Computergeschichte - Faszination über Jahrtausende, R. Weiss)

Siegeszug des PC.

Die Zukunft der Numerischen Mathematik wird voraussichtlich weiter stark geprägt werden von der weiteren Entwicklung des Computers. Das Moorsche Gesetz, welches die Verdopplung der Rechenleistung alle 18-24 Monate voraussagt, wird früher oder später durch physikalische Gründe ausser Kraft gesetzt werden. Ein alternativer Ansatz stellt die Idee eines Quantencomputers dar, welcher auf Basis der quantenmechanischen Zustände der Atome rechnet und dadurch unvorstellbar viele Operationen parallel durchführen kann.

1.3 Typische Fragestellungen

Einige typische Fragestellungen, die in den Vorlesungen Numerik 1 und 2 behandelt werden:

1. Numerische Lösung von Nullstellenproblemen

   Sei f :$\mathbb{R}\to\mathbb{R}$: Suche $x\epsilon\ \mathbb{R}$ mit $f(x) = 0$.

   Für die quadratische Gleichung $f(x) = x^2 + px + q$ sind die Nullstellen bekannt:

   $$x_{1,2} = -p/2 \pm \sqrt{p^2/4 - q}.$$

   Aber wie bestimmt man Nullstellen von komplizierteren Funktionen wie $f(x) = x^n + a_{n-1}x^{n-1} + ... + a_0$ oder $f(x) = \exp(x) - \sin(x)$?

   Beispiel: Iteratives Tangenten Verfahren nach Newton.

   Die Funktion $f(x)$ kann in der Umgebung von x_0 durch ihre Tangente angenähert werden: $f(x) \approx f(x_0) + f'(x_0)(x - x_0)$. Der Schnittpunkt x_1 der Tangenten mit der x-Achse ist eine erste Näherung für die Nullstelle, dort gilt $0 =$

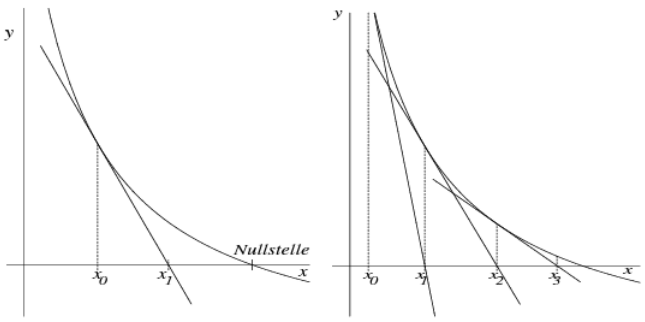


Abbildung 1.3: Newtonverfahren (aus [1])

f(x$_0$)+f'(x$_0$)(x$_1$−x$_0$) und daraus ergibt sich $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$. Durch wiederholen des Verfahrens (siehe Abbildung 1.3) erhält man nach n Schritten die Iterationsformel

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} (n = 1, 2, 3, ...).$$

2. Numerische Lösung von linearen Gleichungssystemen. Beispiel: *A(n,n)* sei eine Matrix reeller Zahlen und $\overrightarrow{b} \epsilon \mathbb{R}^n$ ein Vektor. Gesucht ist der Vektor $\overrightarrow{x} \epsilon \mathbb{R}^n$, so dass $A\overrightarrow{x} = \overrightarrow{b}$. Dieses Problem kann gemäss der linearen Algebra gelöst werden mit $\overrightarrow{x} = A^{-1} \cdot \overrightarrow{b}$, doch ist diese Lösung so nicht effizient berechenbar.

3. Numerische Lösung von nicht linearen Gleichungssystemen. Dies ist im Prinzip eine Erweiterung des Nullstellenproblems oben auf weitere Dimensionen. Zu lösen ist zum Beispiel das Problem:

$$f(x_1, x_2) := \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 + 4x_2 \\ 8x_2^3 + 4x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

   Am Term x_2^3 sieht man, dass das System nicht linear ist. Das oben vorgestellte Newton Verfahren lässt sich auf diese Fragestellung erweitern und für die Lösung solcher Systeme verwenden.

4. Interpolation. Beim typischen Interpolationsproblem sind n diskrete Wertepaare (x_i, f_i) gegeben und gesucht ist eine stetige Funktion f mit der Eigenschaft *f(x$_i$)=f$_i$* für alle x_i. Beispiel: Es soll eine interpolierende Funktion für die beiden Werte (0,1) und (1,2) gefunden werden. Eine mögliche Lösung ist $f(x) = x + 1$, aber auch *f(x)* = $\sqrt{x}$+1 oder *f(x)*=sin(πx) + x + 1 kommen in Frage (vgl. Abb. 1.4). Tatsächlich lösen unendlich viele Funktionen dieses Problem, was zeigt, dass Interpolationsprobleme nicht per se eindeutig lösbar sind.

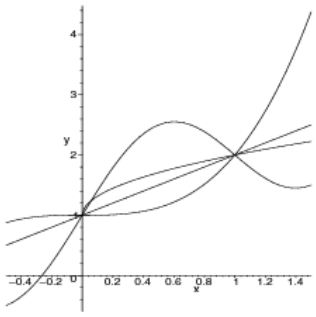


Figure 1.4: Interpolation durch die Punkte (0,1) und (1,2) (aus [1]).

5. Numerische Differentiation. In vielen Anwendungen werden Werte von Ableitungen von Funktionen benötigt. In den seltensten Fällen steht aber die Ableitung f' als Funktion, die man nur noch auswerten müsste, zur Verfügung. Es müssen also Näherungen für die Werte der Ableitung berechnet werden, z.B. gilt in der ersten Ordnung (für genügend kleines h):

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} \approx \frac{f(x_0 + h) - f(x_0)}{h} =: D1f(x_0, h)$$

6. Numerische Integration. Gegeben ist *f:* [a,b]$\longrightarrow\mathbb{R}$. Gesucht ist ein Näherungswert von

$$I = \int_a^b f(x)dx$$

Beispiel: Mittelpunkts- oder Trapezregel wie in Abb. 1.5.

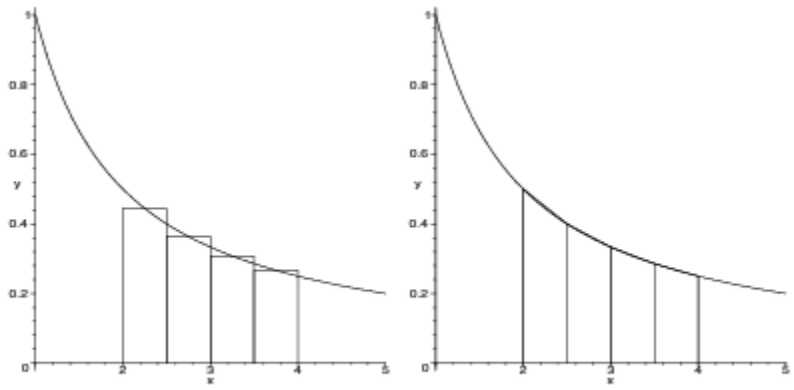


Bild 7.4 Summierte Mittelpunktsregel für $\int_2^4 x^{-1}\,dx$ und $n = 4$

Bild 7.5 Summierte Trapezregel für $\int_2^4 x^{-1}\,dx$ und $n = 4$

Figure 1.5: Beispiel zur Mittelpunkt- und Trapezregel (aus [1]).

7. Regression (auch 'Ausgleichprobleme'; behandelt in Numerik 2). Häufig sind n Wertepaare (x_i, y_i) das Resultat von Messungen und deshalb mit gewissen Unsicherheiten behaftet, welche sich z.B. als Streuung manifestieren können (sogenannte 'Scatter Plots'). Dann empfiehlt es sich, nicht eine Funktion zu suchen, die exakt durch die Wertepaare geht (wie unter 4.), sondern möglichst 'nah' an alle beobacheten Werte rankommt. Ein theoretisches Beispiel ist in Abb. 1.6 gezeigt. Offenbar gibt es dort einen Trend, dass die gemessene Grösse y (z.B. die Fläche einer bestimmten Moosart) mit der Grösse x (z.B. ein Mass für den pH Wert des Bodens) etwa linear abnimmt. Dieser Trend lässt sich als Gerade *f(x)* mittels der Methode der kleinsten Quadrate berechnen, so dass der Fehler $\sum(y_i - f(x_i))^2$minimal wird.

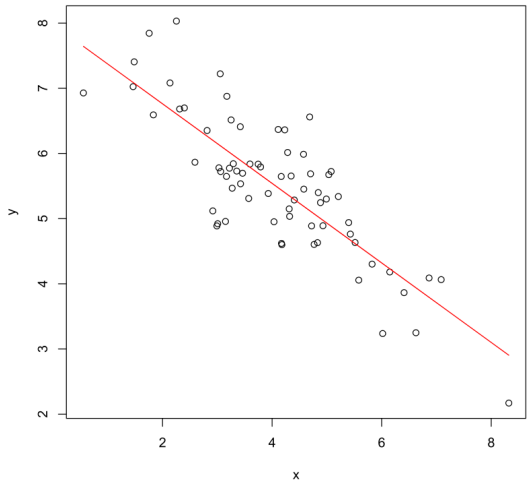


Abbildung 1.6: Sogenannter 'Scatter Plot' und die mittels linearer Regression bestimmte Näherungsgerade.

8. Differentialgleichungen (behandelt in Numerik 2). Gegeben ist eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}$, ein Intervall [a,b] und ein Anfangswert y_0. Gesucht ist eine Funktion $y : [a, b] \to \mathbb{R}$ mit

$$y'(t) = f(t, y(t))$$

für alle $t \in [a, b]$ und $y(a) = y_0$. Das Anfangswertproblem besteht also darin, eine Lösung y der gewöhnlichen Differenzialgleichung zu finden, die an der Stelle $t = a$ den vorgegebenen Werty_0 annimmt. Die Lösungen lassen sich anhand sogenannter Richtungsfelder wie in Abb. 1.7 veranschaulichen.

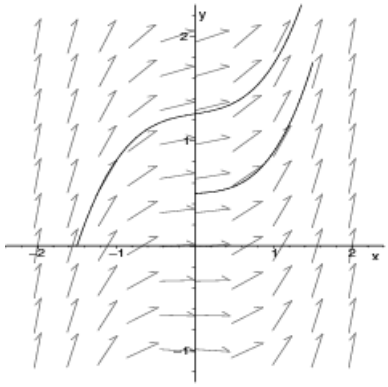


Figure 1.7: Richtungsfeld einer gewöhnlichen Differentialgleichung (aus [1]).

9. Fourier Transformation (behandelt in Numerik 2). Eine periodische Funktion $f(t)$ mit der Periode $T > 0$, also $f(t + k \cdot T) = f(t)$, kann unter gewissen Bedingungen in ihre harmonischen Schwingungen, d.h. in Sinus und Kosinusfunktionen, zerlegt werden, deren Frequenzen ganzzahlige Vielfache der Grundfrequenz $\omega = \frac{2\pi}{T}$ sind:

$$\begin{aligned} f(t) &= \frac{a_0}{2} + a_1 \cos(\frac{2\pi}{T} \cdot t) + b_1 \sin(\frac{2\pi}{T} \cdot t) + a_2 \cos(2 \cdot \frac{2\pi}{T} \cdot t) + b_2 \sin(2 \cdot \frac{2\pi}{T} \cdot t) + ... \\ &= \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cdot \cos(k\omega t) + b_k \sin(k\omega t) \end{aligned}$$

Die Koeffizienten ergeben sich zu

$$\begin{aligned} a_k &= \frac{2}{T} \int_0^T f(t) \cdot \cos(k\omega t)\, dt \\ b_k &= \frac{2}{T} \int_0^T f(t) \cdot \sin(k\omega t)\, dt \end{aligned}$$

Einfacher kann dies mit komplexen Zahlen dargestellt werden:

$$\begin{aligned} f(t) &= \sum_{k=-\infty}^{\infty} c_n \mathrm{e}^{jn\omega t} \\ c_n &= \frac{1}{T} \int_0^T f(t) \cdot \mathrm{e}^{-jn\omega t}\, dt \end{aligned}$$

Die Koeffizienten enthalten dann die Information über die Frequenz. Durch den Plot von c_n^2 erhält man dann ein sogenanntes Powerspektrum.

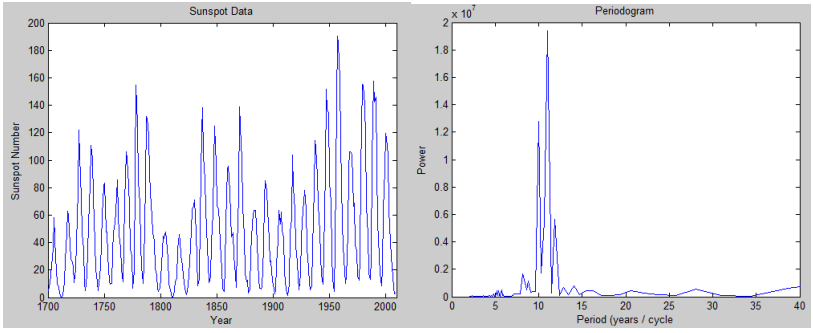


Figure 1.8: Sonnenfleckenzahlen als Funktion der Zeit und zugehöriges Powerspektrum.

Kapitel 2

Rechnerarithmetik

Dieses Kapitel beschäftigt sich damit, wie Zahlen dargestellt werden, was für Fehler dabei entstehen können und wie sich diese bei Operationen fortpflanzen.

Lernziele:

- Sie verstehen die Definition der maschinendarstellbaren Zahlen und können Gleitpunktzahlen zwischen verschiedenen Basen umrechnen.
- Sie können die Fehler, die beim Abbilden von reellen Zahlen auf Maschinenzahlen entstehen, sowie die Maschinengenauigkeit berechnen.
- Sie können die Fortpflanzung von Fehlern bei Funktionsauswertungen abschätzen und die Konditionszahl berechnen.

2.1 Zur Geschichte der Zahlendarstellung[1]

In den frühen Hochkulturen entwickelten sich unterschiedliche Konzepte zur Darstellung von Zahlen, die nach Art der Zusammenstellung und der Anordnung der Ziffern in Additionssysteme und Positionssysteme (auch Stellenwertsysteme genannt) einteilbar sind. Additionssysteme ordnen jeder Ziffer eine bestimmte Zahl zu. Im Gegensatz dazu ordnen Positions- oder Stellenwertsysteme jeder Ziffer aufgrund der relativen Position zu anderen Ziffern eine Zahl zu. So haben die beiden Ziffern 2 und 5 im Dezimalsystem je nach Zusammenstellung entweder den Wert 25 oder 52. Alle Zahlensysteme bauen dabei auf einer sogenannten ganzzahligen Grundzahl[2] $B > 1$, auch Basis genannt, auf.

Das ägyptische Additionssystem dürfte mit ca. 5000 Jahren wohl das älteste logisch aufgebaute Zahlensystem sein. Es benutzt die Basis $B = 10$. Für die ersten sieben Stufenzahlen 10^i mit $0 \leq i \leq 6$ werden spezielle Hieroglyphen als Zahlzeichen verwendet, wie in der Abb. 2.1 zu sehen. Die Ziffer 0 existierte zu diesem Zeitpunkt noch nicht, was bei dieser Darstellungsart nicht als Mangel empfunden wird.

Zu Beginn des 2. Jahrtausends v. Chr. verwendeten die babylonischen Mathematiker eine Zahlenschrift, die nur einen Nagel für eine Eins und einen offenen Winkel für die 10 benutzte. Ein Nagel konnte dabei - je nach seinem Abstand zu den anderen Winkeln oder Nägeln - eine 1 oder eine 60 bedeuten. Die Zahl $3661 = 1 \cdot 60^2 + 1 \cdot 60^1 + 1 \cdot 60^0$ wurde durch drei Nägel mit Abstand angezeigt, während drei Nägel ohne Abstand einfach für 3 standen. Das Problem der eindeutigen Darstellung der Zahl $3601 = 1 \cdot 60^2 + 0 \cdot 60^1 + 1 \cdot 60^0$wurde fast zwei Jahrtausende später gelöst, indem man als Platzhalter für eine fehlende Stelle zwei schräg hochgestellte Nägel einführte, also 3601durch zwei Nägel, getrennt durch die beiden hochgestellten Nägel, markierte. Siehe dazu auch Abb. 2.2.

Sollen die Ziffern unabhängig von der Position verwendet werden, kommen wir also um die Darstellung der Ziffer Null nicht herum. Wir sind dann z.B. in der Lage, die beiden Zahlen $701 = 7 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0$ und

[1] Übernommen in gekürzter und leicht abgeänderter Form von Kap. 1 und Kap. 5 aus [7]

[2] Vor allem wurden die Zahlen 2, 5, 10, 12, 20 und 60 benutzt. Die wohl wichtigsten Grundzahlen sind 2 und 10. Von besonderem Interesse für die Babylonier war die Zahl 60, da sie zugleich die Zahl 30, also ungefähr die Anzahl Tage in einem Monat, als auch die Zahl 12, die Anzahl Monate in einem Jahr, als Teiler besitzt.

Abbildung 2.1: Symbole zum Darstellen von Zahlen bei den antiken Ägyptern: Ein Strich war ein Einer, ein umgekehrtes U ein Zehner, die Hunderter wurden durch eine Spirale, die Tausender durch die Lotosblüte mit Stil und die Zehntausender durch einen oben leicht angewinkelten Finger dargestellt. Dem Hunderttausender entsprach eine Kaulquappe mit hängendem Schwanz. Ergänzend ohne Bild hier: Die Millionen wird durch einen Genius, der die Arme zum Himmel erhebt, repräsentiert (aus [7]).

Abbildung 2.2: Babylonische Form der Zahl $46821 = 13 \cdot 60^2 + 0 \cdot 60^1 + 21 \cdot 60^0$ als $13 \mid 0 \mid 21$ (aus [7]).

$71 = 7 \cdot 10^1 + 1 \cdot 10^0$ zu unterscheiden. Die Ziffer 0 deutet das Auslassen einer "Stufenzahl" B^i an und ermöglicht eine übersichtlichere Darstellung in der modernen Nomenklatur

$$z = \sum_{i=0}^{n} z_i B^i.$$

Die Symbole für die Ziffern $z_i \in \{0, 1, ..., b-1\}$ können wir dann als Koeffizienten der Stufenzahlen B^i interpretieren. Die Einführung der für uns heute nicht mehr wegzudenkenden Null als eigenständige Ziffer verdanken wir indischen Mathematikern.

Im 3. Jahrhundert v.Chr. wurde in Nordindien ein Zehnersystem entwickelt, das für die Ziffern 1 bis 9 graphische Zeichen benutzte, und dabei für diese Ziffern in verschiedenen Zehnerpotenzen auch verschiedene Zeichen zur Verfügung hatte, also z.B. für 9, 90, 900, usw. jeweils ein eigenes Zeichen. Im 5. Jahrhundert n.Chr. fanden indische Mathematiker heraus, dass ihr Zahlensystem sich stark vereinfachen liess, wenn man auf diese Unterscheidung der Potenzen verzichtete und statt dessen eine eigene neue Ziffer, die Null, die Auslassung einer Zehnerpotenz anzeigte. Das indische Wissen wurde durch Araber wie den Universalgelehrten Mohammed Ibn Musa al-Charismi [3](etwa 780 - 850) nach Europa weitervermittelt (siehe Abb. 2.3).

In Europa galt die Ziffer Null lange als Teufelswerk und findet sich erstmalig in einer Handschrift von 976. Bis ins Mittelalter wurden in Europa Zahlen in lateinischen Grossbuchstaben geschrieben. Im römischen Zahlensystem standen I, V, X, L, C, D und M für 1, 5, 10, 50, 100, 500 und 1000. Grössere Zahlen wurden einfach zusammengesetzt, so steht z.B. MMMDCCCLXXVI für die Zahl 3876. Zum Rechnen war dieses Zahlensystem allerdings kaum geeignet und wurde im Laufe des 13. Jahrhunderts von den arabischen Ziffern abgelöst. Dazu beigetragen hat Anfang des 13. Jahrhunderts vor allem der Mathematiker Leonardo Fibonacci[4] aus Pisa, der die Kenntnis der arabischen Ziffern mit seinem Werk '*Liber Abaci*' verbreitete. Detailliert erklärt er darin das Multiplizieren, Dividieren, Bruchrechnen, Potenzrechnen, Wurzelziehen und die Lösung von quadratischen und kubischen Gleichungen mit den arabischen Ziffern. Aus dem arabischen Wort *as-sifr* (die Leere) kreierte er den Namen *zefirum*, aus dem sich sowohl das Wort Ziffer, als auch das englische *zero* ableitet.

Das neue Zahlensystem mit der Null ermöglichte auch das Automatisieren von Rechenschritten. Im Jahre 1671 entwickelte Gottfried Wilhelm Leibniz[5] seine Rechenmaschine, die REPLICA, die bereits alle vier Grundrechenarten beherrschte. Kurz darauf beschrieb er das binäre (oder auch duale) Zahlensystem, ohne das die heutige elektroni-

[3]Gemäss Wikipedia auch al-Chwarizmi. Von seinem Namen leitet sich der Begriff Algorithmus ab. Sein Beiname al-Chwarizmi bedeutet "der Choresmier" und bezieht sich auf seine Herkunft aus diesem iranischen Volk. Sein arabisches Lehrbuch *Über das Rechnen mit indischen Ziffern* (um 825) beginnt in der mittelalterlichen lateinischen Übersetzung mit den Worten *Dixit Algorismi* (Algorismi hat gesagt).

[4]Namensgeber der Fibonacci Folge $1, 1, 2, 3, 5, 8, 13, ...$, der damit um 1202 das Wachstum einer Kaninchenpopulation beschrieb. Die Folge war auch vor ihm bereits den Arabern und Indern bekannt.

[5]1646 - 1716, deutscher Universalgelehrter und, zusammen mit seinem Zeitgenossen Sir Isaac Newton (1643 - 1727), unter anderem Urheber der Integral- und Differentialrechnung

Abbildung 2.3: Arabische und indische Symbole zum Darstellen von Zahlen: In der ersten Zeile sehen wir die indischen Ziffern des 2. Jahrhunderts n.Chr. Diese bildhaften Ziffern wurden erst von den Arabern übernommen (zweite Zeile) und später von den Europäern (dritte bis sechste Zeile: 12., 14, 15. und 16. Jhr.) immer abstrakter dargestellt (aus [7]).

sche Datenverarbeitung kaum vorstellbar wäre. Auf dieses und weitere Zahlensysteme und die Implikationen auf arithmetische Operationen wollen wir im weiteren näher eingehen.

2.2 Maschinenzahlen

Die Menge der reellen Zahlen $\mathbb{R}$ hat unendlich viele Elemente. Jede Rechenmaschine ist aber ein endlicher Automat, d.h. er kann aufgrund der beschränkten Stellenzahl nicht alle Zahlen exakt darstellen und nur endlich viele Operationen ausführen. Für eine gegebene Basis $B \in \mathbb{N}$ $(B > 1)$ kann jede reelle Zahl $x \in \mathbb{R}$ aber als

$$x = m \times B^e$$

dargestellt werden, wobei $m \in \mathbb{R}$ die Mantisse und $e \in \mathbb{Z}$ der Exponent genannt wird.

Computerintern wird üblicherweise die Basis $B = 2$ verwendet (als Binär- od. Dualzahlen benannt), dies als direkte Folge der Zustände 'Strom' / 'kein Strom' (bzw. 1 und 0) von mikroelektronischen Schaltungen. Man spricht hier von einem *Bit* ('binary digit')[6], welches genau zwei Werte, eben 0 oder 1 annehmen kann. Werden Bits zu Einheiten von 8 Bits zusammengefasst, spricht man von einem *Byte*. Weitere übliche Basen sind $B = 8$ (Oktalzahlen), $B = 10$ (Dezimalzahlen) und $B = 16$ (Hexadezimalzahlen). Für letztere benötigt man 16 verschiedene Zeichen und verwendet die Ziffern 0,1,...,9 sowie *A,...,F* (wobei $A \triangleq 10$, $B \triangleq 11$ etc., auch Kleinbuchstaben sind erlaubt).

Aufgabe 2.1:

1. Überlegen Sie sich: wie viele verschiedene Möglichkeiten gibt es, mit Binärzahlen ein Byte zu füllen?
2. Wie viele Ziffern bräuchten Sie im Hexadezimalsystem, um die gleiche Anzahl Möglichkeiten zu erhalten?
3. Was folgern Sie daraus bzgl. der Vorteile des Hexadezimalsystems?

Im Rechner stehen natürlich nur endlich viele Stellen für m und e zur Verfügung, z.B. n Stellen für m und l Stellen für e. Wir schreiben (entsprechend der englischen Schreibweise wird das Komma durch einen Punkt ersetzt):

$$\begin{aligned} m &= \pm 0.m_1 m_2 m_3 ... m_n \\ \textit{und } e &= \pm e_1 e_2 ... e_l \end{aligned}$$

[6] Gemäss [7] wurde der Begriff *bit* das erste Mal wahrscheinlich von John Tukey (amerikanischer Mathematiker, 1915 - 2000, Träger der IEEE 'Medal of Honor') verwendet, als kürzere Alternative zu *bigit* oder *binit*. Das Wort digit kommt aus dem Lateinischen und bedeutet Finger.

Definition 2.1: Maschinenzahlen / Gleitpunktzahlen

- Unter der zusätzlichen Normierungs–Bedingung $m_1 \neq 0$ (falls $x \neq 0$) ergibt sich eine eindeutige Darstellung der sogenannten ***maschinendarstellbaren Zahlen M*** zur Basis B:

$$M = \left\{ x \epsilon \mathbb{R} | \, x = \pm 0.m_1 m_2 m_3 ... m_n \cdot B^{\pm e_1 e_2 ... e_l} \right\} \cup \{0\}$$

  Dabei gilt m_i, $e_i \epsilon \{0,1,..., B\text{-}1\}$ für $i \neq 0$ und $B \in \mathbb{N},\ B > 1$)

- Der **Wert** einer solchen Zahl ist definiert als

$$\sum_{i=1}^{n} m_i B^{e-i}$$

  und ergibt gerade die (nicht normierte) Darstellung der Zahl im Dezimalsystem. Dabei ist e ebenfalls im Dezimalsystem zu nehmen, also $e = \sum_{i=1}^{l} e_i B^{l-i}$ und es gilt $e \in \mathbb{Z}$, d.h. e kann natürlich auch negativ sein. Weiter gibt es eine obere und untere Schranke: $e_{min} \leq e \leq e_{max}$.

- Man redet dann auch von einer ***n*-stelligen Gleitpunktzahl zur Basis *B*** (engl: floating point). Zahlen, die nicht in dieser Menge M liegen, müssen durch Rundung in eine maschinendarstellbare Zahl umgewandelt werden.

Bemerkungen:

- Der Exponent $e \in \mathbb{Z}$ definiert, wie wir es vom Dezimalsystem kennen, die Position des Dezimalpunktes, also z.B. $x = 112.78350 = 112.78350 \cdot 10^0 = 1127835.0 \cdot 10^{-4} = 0.11278350 \cdot 10^3$. Eine Verschiebung des Dezimalpunktes um n Stellen nach links führt (bei gleichbleibendem Wert der Zahl) zu einer Erhöhung des Exponenten auf $e + n$. Eine Verschiebung des Dezimalpunktes um n Stellen nach rechts führt entsprechend zu einer Reduktion des Exponenten auf $e - n$. Analoges gilt für sämtliche andere Basen, z.B. bei $B = 2$: $x = 11001.111 = 11001.111 \cdot 2^0 = 0.11001111 \cdot 2^5$.

- Um Missverständnisse zu vermeiden, kann die Basis explizit als Index zu einer Mantisse in Klammern angegeben werden. Wird kein Exponent angegeben, ist das gleichbedeutend mit $e = 0$, z.B. $(1011100.111)_2 = 1011100.111 \cdot 2^0 = 0.1011100111 \cdot 2^7 = (0.1011100111)_2 \cdot 2^7$.

- Die in Definition 2.1 gewählte Normierungsbedingung, kann auch durch andere Normierungsbedingungen ersetzt werden. Was wären weitere Möglichkeiten? Weshalb normiert man überhaupt? Studieren Sie hierzu die folgenden Beispiele.

Beispiele 2.1 (teilweise gemäss [1]):

1. Normierte Gleitpunktzahlen (gemäss Definition 2.1):

   (a) $x_1 = -0.2345 \cdot 10^3$ ist eine vierstellige Gleitpunktzahl im Dezimalsystem mit dem Wert

$$-\sum_{i=1}^{4} m_i \cdot 10^{3-i} = -(2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1}) = -234.5 \ (= -0.2345 \cdot 10^3)$$

   (b) $x_2 = 0.111 \cdot 2^3$ ist eine dreistellige Gleitpunktzahl im Binär-/Dualsystem mit dem Wert

$$\sum_{i=1}^{3} m_i \cdot 2^{3-i} = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 7 \ (= 0.7 \cdot 10^1)$$

   (c) $x_3 = 0.1001 \cdot 2^{-3}$ ist eine vierstellige Gleitpunktzahl im Binär-/Dualsystem mit dem Wert

$$\sum_{i=1}^{4} m_i \cdot 2^{-3-i} = 2^{-4} + 2^{-7} = 0.0703125 \ (= 0.703125 \cdot 10^{-1})$$

(d) $x_4 = 0.71537 \cdot 8^2$ ist eine fünfstellige Gleitpunktzahl im Oktalsystem mit dem Wert

$$\sum_{i=1}^{5} m_i \cdot 8^{2-i} = 7 \cdot 8^1 + 1 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 8^{-2} + 7 \cdot 8^{-3} = 57.685546875 \ (= 0.57685546875 \cdot 10^2)$$

(e) $x_5 = 0.AB3C9F \cdot 16^4$ ist eine sechsstellige Gleitpunktzahl im Hexadezimalsystem mit dem Wert

$$\sum_{i=1}^{6} m_i \cdot 16^{4-i} = 10 \cdot 16^3 + 11 \cdot 16^2 + 3 \cdot 16^1 + 12 \cdot 16^0 + 9 \cdot 16^{-1} + 15 \cdot 16^{-2} = 43836.62109375 (= 0.4383662109375 \cdot 10^5)$$

2. Nicht normierte Gleitpunktzahlen:

   Die obigen Beispiele $x_1 - x_5$ sind alle gemäss Definition 2.1 normiert, d.h. die erste Ziffer vor dem Punkt ist Null, die erste Ziffer nach dem Punkt ist ungleich Null für $x \neq 0$. Damit sind ihre Mantisse und der Exponent eindeutig definiert. Die folgenden Beispiele geben zur Illustration nicht normierte Varianten für x_1 und x_2. Es ist offensichtlich, dass eine nicht normierte Darstellung bzgl. Mantisse und Exponent nicht mehr eindeutig ist (auch wenn der Wert immer gleich bleibt). Dies ist ein Zustand, der bei der Speicherung vermieden werden muss.

   (a) $\tilde{x}_1 = -0.002345 \cdot 10^5 = -23.45 \cdot 10^2 = -234500 \cdot 10^{-3} = ...$

   (b) $\tilde{x}_2 = 0.0111 \cdot 2^4 = 1.11 \cdot 2^2 = 111 \cdot 2^0 = 11100 \cdot 2^{-2} = ...$

3. IEC/IEEE[7] - Gleitpunktzahlen. Es gibt zwei Grundformate zur Basis $B = 2$

   (a) Single precision (einfache Genauigkeit). Gesamtlänge der Zahl ist 32 Bit (4 Bytes), davon 1 Bit für das Vorzeichen, 23 Bit für die Mantisse, und 8 Bit für den Exponenten:

   `single`: (32 bit)

   ```
   V EEEEEEEE MMMMMMMMMMMMMMMMMMMMMMM
   0 1      8 9                     31
   ```

   (b) Double precision (doppelte Genauigkeit). Gesamtlänge der Zahl ist 64 Bit (8 Bytes), wobei 1 Bit für das Vorzeichen, 52 Bit für die Mantisse, und 11 Bit für den Exponenten:

   `double`: (64 bit)

   ```
   V EEEEEEEEEEE MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
   0 1        11 12                                            63
   ```

   Das Vorzeichenbit $v \ \epsilon\{0,1\}$ erzeugt das Vorzeichen über den Faktor (-1)^v, d.h. $v = 0$ entspricht einem positiven, $v = 1$ einem negativen Vorzeichen. Beim IEEE Format ist die Mantisse so normiert, so dass die Ziffer vor dem Dezimalpunkt ungleich 0 sein muss. Ausserdem wird ein von der Anzahl Bits abhängiger Biaswert im Exponent subtrahiert, womit der Exponent kein eigenes Vorzeichenbit benötigt.

4. Wieso rechnet man eigentlich mit Gleitpunktzahlen? Nimmt man, abgesehen vom Vorzeichen, an, dass 8 dezimale Speichereinheiten zur Verfügung stehen, so liessen sich damit in den folgenden Systemen die folgenden positiven Zahlen darstellen:

   (a) Ganzzahlsystem:

      i. kleinste darstellbare positive Zahl:0000 0001
      ii. grösste darstellbare positive Zahl: 9999 9999

   Es lassen sich also im positiven Bereich alle ganzen Zahlen zwischen 1 und 99 999 999 ($= 10^8 - 1$) hinterlegen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist konstant gleich 1.

   (b) Festpunktsystem mit 4 Dezimalen:

      i. kleinste darstellbare positive Zahl:0000.0001
      ii. grösste darstellbare positive Zahl: 9999.9999

[7] Institute of Electrical and Electronics Engineers

Es lassen sich im positiven Bereich Zahlen zwischen 0.0001 und 9999.9999 ($= 10^4 - 10^{-4}$) darstellen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist konstant gleich 10^{-4}.

(c) Normalisiertes Gleitpunktsystem mit 6 Mantissen- und 2 Exponenententenziffern (mit Bias -50)

i. kleinste darstellbare positive Zahl: $0.100000 \cdot 10^{-50}$

ii. grösste darstellbare positive Zahl: $0.999999 \cdot 10^{49}$

Es lassen sich im positiven Bereich Zahlen zwischen 10^{-51} und fast10^{49} darstellen. Der Abstand zwischen aufeinanderfolgenden Zahlen ist allerdings variabel, wie in Kap. 2.4 gezeigt.

Offensichtlich ist der darstellbare Zahlenbereich bei gleichem Speicherbedarf bei Gleitpunktzahlen enorm gross.

Aufgabe 2.2 [1]:

1. Wie viele Stellen benötigt man für die Mantisse, um die folgenden Zahlen als n-stellige Gleitpunktzahlen im Dezimalsystem darzustellen? Wie gross ist der zugehörige Exponent?

   $x_1 = 0.00010001$, $x_2 = 1230001$, $x_3 = \frac{4}{5}$, $x_4 = \frac{1}{3}$

2. Bestimmen Sie alle dualen positiven 3-stelligen Gleitpunktzahlen mit einstelligem positiven binären Exponenten sowie ihren dezimalen Wert.

3. Wie viele verschiedene Maschinenzahlen gibt es auf einem Rechner, der 20-stellige Gleitpunktzahlen mit 4-stelligen binären Exponenten sowie dazugehörige Vorzeichen im Dualsystem verwendet? Wie lautet die kleinste positive und die größte Maschinenzahl?

4. Verstehen Sie den folgenden 'Witz'?

   Es gibt 10 Gruppen von Menschen: diejenigen, die das Binärsystem verstehen, und die anderen.

2.3 Umrechnung zwischen den Basen

Für das gegenseitige Konvertieren von Zahlen mit unterschiedlichen Basen reicht es, als ein Bezugssystem das Dezimalsystem zu nehmen. Wir müssen also im Grunde zwei Richtungen betrachten, die Umrechnung einer Gleitkommazahl mit Basis $B \neq 10$ ins Dezimalsystem und die Umrechnung vom Dezimalsystem in eine beliebige andere Basis[8].

2.3.1 Umrechnung von einer beliebigen Basis ins Dezimalsystem

Die Umwandlung einer Gleitkommazahl mit Basis B in die zugehörige Dezimalzahl ist nichts anderes als die Berechnung des Wertes gemäss Definition 2.1, doch empfiehlt es sich, den ganzzahligen Anteil und den Dezimalteil (nach dem Dezimalpunkt) getrennt als eigenständige Polynome zu behandeln. Dies erlaubt es uns, das Horner-Schema zu verwenden, wie am folgenden Beispiel erläutert werden soll:

Beispiel 2.2

- Die (unnormierte) Binärzahl $(x)_2 = 11001.1011$ soll ins Dezimalsystem umgerechnet werden. Gemäss Definition 2.1 gilt

$$(x)_{10} = \sum_{i=1}^{n} m_i B^{e-i} = \underbrace{1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0}_{\text{ganzzahliger Anteil}} + \underbrace{1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4}}_{\text{Dezimalanteil}} = 25.6875$$

  Die sequentielle Summation dieses langen Ausdrucks ist nicht sehr effizient. Das Horner-Schema (bekannt z.B. von der Linearfaktorzerlegung von Polynomen) gibt uns dafür eine effizientere Methode, wenn wir den ganzzahligen Anteil und den Dezimalanteil getrennt als eigenständige Polynome behandeln. Das Horner-Schema erlaubt die Auswertung dieser beiden Polynome an den Stellen $x = 2$ bzw. $x = \frac{1}{2}$ sehr effizient.

[8] Wir wollen an dieser Stelle nicht auf alle möglichen Spielarten eingehen, ausführlich erläutert werden die Umrechnungsarten z.B. auf der Webseite http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm#1

– Ganzzahliger Anteil. Der Faktor 2 wird fortlaufend mit den Koeffizienten m_i $(i = 1, 2, ..., e)$ multipliziert und 'von links nach rechts' aufsummiert (mit Start beim innersten Klammerausdruck):

$$\underrightarrow{11001} = \underrightarrow{(((\mathbf{1} \cdot 2 + \mathbf{1}) \cdot 2 + \mathbf{0}) \cdot 2 + \mathbf{0}) \cdot 2 + \mathbf{1}} = 25 \ (= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1)$$

oder analog mit dem Schema (Erklärung folgt im Unterricht)

	2^4	2^3	2^2	2^1	2^0
	1	1	0	0	1
$B = 2$	$\downarrow$	2	6	12	24
	1	3	6	12	**25**

– Dezimalanteil. Der Faktor 2^{-1} bzw. $\frac{1}{2}$ wird fortlaufend mit den Koeffizienten m_i $(i = e+1, e+2, ..., n)$ multipliziert und von 'rechts nach links' aufsummiert (mit Start beim innersten Klammerausdruck):

$$.\underleftarrow{1011} = \underleftarrow{\frac{1}{2}\left(\mathbf{1} + \left(\frac{1}{2}\right)\left(\mathbf{0} + \left(\frac{1}{2}\right)\left(\mathbf{1} + \mathbf{1} \cdot \left(\frac{1}{2}\right)\right)\right)\right)} = 0.6875 \ (= 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4})$$

oder analog mit dem Schema (Erklärung folgt im Unterricht)

	$(\frac{1}{2})^4$	$(\frac{1}{2})^3$	$(\frac{1}{2})^2$	$(\frac{1}{2})^1$	$(\frac{1}{2})^0$
	1	1	0	1	0
$B = \frac{1}{2}$	$\downarrow$	0.5	0.75	0.375	0.6875
	1	1.5	0.75	1.375	**0.6875**

Wie wir in der folgenden Aufgabe sehen werden, ist das Horner-Schema gut geeignet, um den ganzzahligen Anteil zu berechnen, jedoch nur bedingt, um den Dezimalanteil zu bestimmen. Trotzdem hilft es, das Prinzip zu verstehen.

Aufgabe 2.3

- Konvertieren Sie die folgenden Zahlen mit dem Horner-Schema ins Dezimalsystem:

1. $(110001110.00101)_2$
2. $(111110101.1101)_2$
3. $(122102.102)_3$
4. $(345.2114)_6$
5. $(AFDE.BB1C)_{16}$

2.3.2 Umrechnung vom Dezimalsystem in andere Zahlensysteme

Die Umkehrung des Horner-Schemas erlaubt es, den ganzzahligen Anteil und den Dezimalanteil je für sich durch durch fortlaufendes Ausklammern der neuen Basis B (bzw. $\frac{1}{B}$) zu berechnen. Wir illustrieren dies beispielhaft anhand der Umrechnung der Zahl $(1006.687)_{10}$ ins Binär-, Octal- und Hexadezimalsystem.

2.3.2.1 Umrechnung vom Dezimal- ins Binärsystem

Die Zahl $x = 1006.687$ soll vom 10er-System ins 2er-System umgewandelt werden. Ganzzahliger Anteil und Dezimalanteil werden jeweils getrennt behandelt.

- Zunächst die Umwandlung des ganzzahligen Teils. Gehen Sie nach folgendem Verfahren vor:

1. Teilen Sie die Zahl durch 2 und notieren sich den Rest (0 oder 1).
2. Nehmen Sie das Resultate der vorherigen Division und wiederholen den Vorgang bis die Division durch 2 Null ergibt.
3. Die Ziffernfolge für den Rest ergibt (von "unten nach oben") die Binärdarstellung der Zahl.

$$\begin{array}{rcll} 1006{:}2 & = & 503 & \text{Rest: } 0 \\ 503{:}2 & = & 251 & \text{Rest: } 1 \\ 251{:}2 & = & 125 & \text{Rest: } 1 \\ 125{:}2 & = & 62 & \text{Rest: } 1 \\ 62{:}2 & = & 31 & \text{Rest: } 0 \\ 31:2 & = & 15 & \text{Rest: } 1 \\ 15:2 & = & 7 & \text{Rest: } 1 \\ 7:2 & = & 3 & \text{Rest: } 1 \\ 3:2 & = & 1 & \text{Rest: } 1 \\ 1:2 & = & 0 & \text{Rest: } 1 \end{array}$$

Das Resultat ist: 1111101110

- Nun die Umwandlung des Dezimalanteils. Gehen Sie nach folgendem Verfahren vor:

  1. Multiplizieren Sie die Zahl mit der Basis 2 und notieren Sie sich die Zahl vor dem Komma
  2. Falls diese 1 wird, schneiden Sie sie weg bis der Rest 0 ist, der Rest sich wiederholt oder die gewünschte Genauigkeit erreicht ist.
  3. Die Ziffernfolge für den Rest ergibt (von "oben nach unten") die Binärdarstellung der Zahl.

$$\begin{array}{rcll} 2 \cdot 0{,}687 & = & 1{,}374 & \text{Ziffer } 1 \\ 2 \cdot 0{,}374 & = & 0{,}748 & \text{Ziffer } 0 \\ 2 \cdot 0{,}748 & = & 1{,}496 & \text{Ziffer } 1 \\ 2 \cdot 0{,}496 & = & 0{,}992 & \text{Ziffer } 0 \\ 2 \cdot 0{,}992 & = & 1{,}984 & \text{Ziffer } 1 \\ 2 \cdot 0{,}984 & = & 1{,}968 & \text{Ziffer } 1 \\ 2 \cdot 0{,}968 & = & 1{,}936 & \text{Ziffer } 1 \\ 2 \cdot 0{,}936 & = & 1{,}872 & \text{Ziffer } 1 \\ 2 \cdot 0{,}872 & = & 1{,}744 & \text{Ziffer } 1 \\ \vdots & & \vdots & \vdots \end{array}$$

Das Resultat ist: 0.10101111...

- Die beiden Teile zusammen ergeben also das Resultat

$$1006.687 = 1111101110.10101111...$$

in unnormierter Darstellung. Wollen wir noch normieren, z.B. auf die Mantisselänge $n = 13$, so erhalten wir

$$1006.687 \approx 0.1111101110101 \cdot 2^{10}$$

Dabei haben wir von 1111101110.10101111... die ersten 13 Ziffern genommen und noch ein '0.' vorangestellt. Der Wert des Exponenten ergibt sich aus der Anzahl Ziffern für den ganzzahligen Anteil (nämlich 10).

Falls es sich bei x nicht gerade um eine Maschinenzahl handelt, lässt sich wegen der begrenzten Mantisselänge nicht verhindern, dass durch das 'Abschneiden' der binären Zahl ein Fehler gemacht wird, denn der Wert der binären Zahl $0.1111101110101 \cdot 2^{10}$ ist (aus Gründen der kompakten Notation verzichten wir hier auf das Horner-Schema)

$$\begin{aligned} &1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 \\ &+1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= 1006.625 \end{aligned}$$

Deshalb wird bei jedem Rechner jeweils ein Fehler auftreten, da die Mantisselänge immer begrenzt ist. Konkret ergibt sich in diesem Beispiel bei der Abbildung ins Dualsystem der absolute Fehler (definiert als Betrag der Differenz zwischen dem Näherungswert und dem exakten Wert):

$$| 1006.625 - 1006.687 | = 0.0620$$

Auf die verschiedenen Fehlerarten kommen wir in Kap. 2.4 noch zu sprechen. Dort werden wir zeigen, dass einfaches Abschneiden i.d.R. kein gutes Verfahren ist, um eine reelle Zahl auf die Menge der Maschinenzahlen abzubilden.

2.3.2.2 Umrechnung vom Dezimal- ins Oktalsystem

Das Vorgehen ist analog zur Umrechnung ins Dualsystem, nur das als Divisor bzw. Multiplikator nun statt der Basis 2 die Basis 8 verwendet wird. Für den ganzzahligen Anteil erhalten wir:

$$\begin{array}{rcccl} 1006{:}8 & = & 125 & \text{Rest:} & 6 \\ 125{:}8 & = & 15 & \text{Rest:} & 5 \\ 15{:}8 & = & 1 & \text{Rest:} & 7 \\ 1{:}8 & = & 0 & \text{Rest:} & 1 \end{array}$$

bzw. $(1006)_{10} = (1756)_8$. Hier haben wir die Basis zur Verdeutlichung als Index angegeben. Für den Dezimalanteil haben wir dann

$$\begin{array}{rcll} 8 \cdot 0{,}687 & = & 5{,}496 & \text{Ziffer } 5 \\ 8 \cdot 0{,}496 & = & 3{,}968 & \text{Ziffer } 3 \\ 8 \cdot 0{,}968 & = & 7{,}744 & \text{Ziffer } 7 \\ 8 \cdot 0{,}744 & = & 5{,}952 & \text{Ziffer } 5 \\ 8 \cdot 0{,}952 & = & 7{,}616 & \text{Ziffer } 7 \\ 8 \cdot 0{,}616 & = & 4{,}928 & \text{Ziffer } 4 \\ 8 \cdot 0{,}928 & = & 7{,}424 & \text{Ziffer } 7 \\ 8 \cdot 0{,}424 & = & 3{,}392 & \text{Ziffer } 3 \\ 8 \cdot 0{,}392 & = & 3{,}136 & \text{Ziffer } 3 \\ \vdots & & \vdots & \vdots \end{array}$$

und zusammen erhalten wir die unnormierte Darstellung

$$(1006.687)_{10} = (1756.537574733...)_8$$

Für die Mantisselänge $n = 13$ ergibt sich die normierte Darstellung

$$1006.687 \approx 0.1756537574733 \cdot 8^4$$

Der Wert dieser Oktalzahl ist

$$\begin{aligned} &1 \cdot 8^3 + 7 \cdot 8^2 + 5 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 8^{-2} + 7 \cdot 8^{-3} \\ &+5 \cdot 8^{-4} + 7 \cdot 8^{-5} + 4 \cdot 8^{-6} + 7 \cdot 8^{-7} + 3 \cdot 8^{-8} + 3 \cdot 8^{-9} \\ &= 1006.686999998987... \end{aligned}$$

Für den absoluten Fehler erhalten wir

$$\mid 1006.686999998987... - 1006.687 \mid \approx 1.0133 \cdot 10^{-9}$$

Offensichtlich ist der absolute Fehler hier deutlich kleiner als in der Binärdarstellung (bei gleicher Mantisselänge).

2.3.2.3 Umrechnung vom Dezimal- ins Hexadezimalsystem

Als Divisor bzw. Multiplikator verwenden wir nun die Basis 16. Zur Erinnerung: wir verwenden die Ziffern 0,1,...,9 sowie A,...,F (wobei $A \triangleq 10$, $B \triangleq 11$, $C \triangleq 12$, $D \triangleq 13$, $E \triangleq 14$, $F \triangleq 15$ in Gross- oder Kleinbuchstaben). Für den ganzzahligen Anteil erhalten wir:

$$\begin{array}{rcccccl} 1006{:}16 & = & 62 & \text{Rest:} & 14 & \rightarrow & \text{Ziffer: E} \\ 62{:}16 & = & 3 & \text{Rest:} & 14 & \rightarrow & \text{Ziffer: E} \\ 3{:}16 & = & 0 & \text{Rest:} & 3 & \rightarrow & \text{Ziffer: 3} \end{array}$$

bzw. $(1006)_{10} = (3EE)_{16}$. Für den Dezimalanteil haben wir dann

$$
\begin{array}{lll}
16 \cdot 0{,}687 = 10{,}992 & \text{Ziffer} & \text{A} \\
16 \cdot 0{,}992 = 15{,}872 & \text{Ziffer} & \text{F} \\
16 \cdot 0{,}872 = 13{,}952 & \text{Ziffer} & \text{D} \\
16 \cdot 0{,}952 = 15{,}232 & \text{Ziffer} & \text{F} \\
16 \cdot 0{,}232 = 3{,}712 & \text{Ziffer} & 3 \\
16 \cdot 0{,}712 = 11{,}392 & \text{Ziffer} & \text{B} \\
16 \cdot 0{,}392 = 6{,}272 & \text{Ziffer} & 6 \\
16 \cdot 0{,}272 = 4{,}352 & \text{Ziffer} & 4 \\
16 \cdot 0{,}352 = 5{,}632 & \text{Ziffer} & 5 \\
16 \cdot 0{,}632 = 10{,}112 & \text{Ziffer} & \text{A} \\
\vdots & \vdots & \vdots
\end{array}
$$

und zusammen erhalten wir die unnormierte Darstellung

$$(1006.687)_{10} = (3\text{EE.AFDF3B645A...})_{16}$$

Für die Mantisselänge $n = 13$ ergibt sich die normierte Darstellung

$$1006.687 \approx 0.3\text{EEAFDF3B645A} \cdot 16^3$$

Der Wert dieser Hexadezimalzahl ist

$$
\begin{aligned}
3 \cdot 16^2 + 14 \cdot 16^1 + 14 \cdot 16^0 + 10 \cdot 16^{-1} + 15 \cdot 16^{-2} + 13 \cdot 16^{-3} \\
+15 \cdot 16^{-4} + 3 \cdot 16^{-5} + 11 \cdot 16^{-6} + 6 \cdot 16^{-7} + 4 \cdot 16^{-8} \\
+5 \cdot 16^{-9} + 10 \cdot 16^{-10} = 1006.686999999999898...
\end{aligned}
$$

Für den absoluten Fehler erhalten wir

$$| \, 1006.686999999999898... - 1006.687 \, | \approx 1.1369 \cdot 10^{-13}$$

Aufgabe 2.4:

1. Konvertieren Sie die Dezimalzahl 2678.317 in die Basis $B = 5$
2. Normieren Sie Ihr Resultat auf eine Mantissenlänge von $n = 12$ und passen Sie den Exponenten entsprechend an.
3. Was für einen absoluten Fehler machen Sie durch das Abschneiden bei dieser Normierung?
4. Schreiben Sie in MATLAB eine Funktion dec_to_bin, die eine beliebige Dezimalzahl inklusive Nachkommastellen (Input) in ihre Binärzahl (Output) mit wählbarer ganzzahliger Mantisselänge (Input) und genügend grossem Exponenten berechnet.

2.4 Approximations- und Rundungsfehler

Die Maschinenzahlen sind nicht gleichmässig verteilt. Ein Beispiel für alle binären normalisierten Gleitpunktzahlen mit 4-stelliger Mantisse und 2-stelligem Exponenten ist in Abb. 2.1 dargestellt. Zwangsläufig gibt es bei jedem Rechner eine grösste (x_{max}) und kleinste positive Maschinenzahl (x_{min}). Dabei gilt für normalisierte Gleitpunktzahlen:[9]

$$
\begin{aligned}
x_{max} &= B^{e_{max}} - B^{e_{max}-n} = (1 - B^{-n}) B^{e_{max}} \\
x_{min} &= B^{e_{min}-1}
\end{aligned}
$$

[9] Wird auf die Normalisierung der Mantisse ($m_1 \neq 0$) verzichtet, führt dies zu sogenannt subnormalen Zahlen, die bis B^{m-n} hinunter reichen (IEEE Standard 754).

Aufgabe 2.5:

- Schreiben Sie die kleinste und grösste binäre positive Maschinenzahl für Abb. 2.4 explizit auf und berechnen Sie deren Wert. Stimmt das mit x_{max} und x_{min} überein?

Zahlen, die ausserhalb des Rechenbereichs [-x_{max}, x_{max}] liegen, sind im *Überlaufbereich (overflow)* und führen zum Abbruch der Rechnung (mit IEEE 754 konforme Systeme geben die Bitsequenz *inf* aus). Zahlen ungleich 0, die innerhalb des Bereichs [-x_{min}, x_{min}] liegen, führen zu einem *Unterlauf (underflow)*. Dann ist es sinnvoll, die Rechnung mit 0 weiterzuführen.

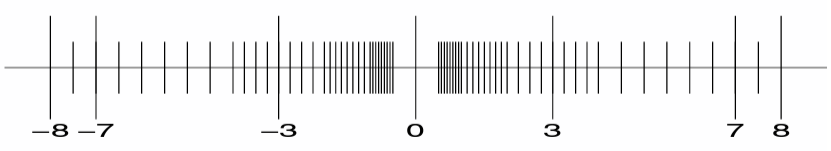


Abbildung 2.4: Alle binären Maschinenzahlen mit $n = 4$ und $0 \leq e \leq 3$ (aus [1])

Offensichtlich ist die Anzahl n der Mantissestellen von entscheidender Bedeutung für den Bereich der Zahlen, die abgebildet werden können. Dies wird eindrücklich illustriert in folgendem Beispiel:

Beispiel 2.3 [7]:

- Am 4. Juni 1996 startete zum ersten Mal eine Ariane 5-Rakete der ESA von Französisch Guyana aus. Die unbemannte Rakete hatte vier Satelliten an Bord. 36.7 Sekunden nach dem Start wurde in einem Programm versucht, den gemessenen Wert der horizontalen Geschwindigkeit von 64 Bit Gleitpunktdarstellung in 16 Bit Ganzzahldarstellung (signed Integer) umzuwandeln. Da die entsprechende Masszahl grösser war als $2^{15} = 32768$, wurde ein Überlauf erzeugt. Das Lenksystem versagte daraufhin seine Arbeit und gab die Kontrolle an eine zweite, identische Einheit ab. Diese produzierte folgerichtig ebenfalls einen Überlauf. Da der Flug der Rakete instabil wurde und die Triebwerke abzubrechen drohten, zerstörte sich die Rakete selbst. Es entstand ein Schaden von ca. 500 Millionen Dollar durch den Verlust der Rakete und der Satelliten. Die benutzte Software stammte vom Vorgängermodell Ariane 4. Die Ariane 5 flog schneller und offensichtlich wurde dies bei der Repräsentation der Geschwindigkeit nicht beachtet.

2.4.1 Rundungsfehler und Maschinengenauigkeit

Aus Abb. 2.1 wird deutlich, dass jede reelle Zahl, die von einem Rechner verwendet werden soll, aber selber keine Maschinenzahl ist, durch eine solche ersetzt werden muss. Dabei entstehen Fehler.

Definition 2.2: Absoluter / Relativer Fehler

- Hat man eine Näherung $\widetilde{x}$ zu einem exakten Wert x, so ist der Betrag der Differenz $\mid \widetilde{x} - x \mid$ der **absolute Fehler.**
- Falls $x \neq 0$, so ist $\mid \frac{\widetilde{x}-x}{x} \mid$ bzw. $\frac{|\widetilde{x}-x|}{|x|}$ der **relative Fehler** dieser Näherung.

  Bemerkung: In der Numerik ist der relative Fehler der wichtigere. Weshalb?

Natürlich sollte die Maschinenzahl dabei so gewählt werden, dass sie möglichst nahe bei der reellen Zahl liegt. Einfaches Abschneiden ist dazu nicht geeignet. Ein besseres Verfahren ist die Rundung. Beim Runden einer Zahl x wird eine Näherung unter den Maschinenzahlen gesucht, die einen minimalen absoluten Fehler $\mid rd(x) - x \mid$ aufweist. Eine n-stellige dezimale Gleitpunktzahl $\widetilde{x} = 0.m_1m_2m_3...m_n \cdot 10^e = rd(x)$, die durch die Rundung eines exakten Wertes x entstanden ist, hat also einen absoluten Fehler von höchstens

$$\mid rd(x) - x \mid \leq 0.\underbrace{00...005}_{n} \cdot 10^e = 0.5 \cdot 10^{e-n},$$

wobei die 5 an der Stelle $n+1$ nach dem Dezimalpunkt auftritt. Für eine beliebige (gerade) Basis gilt analog:

$$\text{falls } B\text{gerade:} \quad | \quad rd(x) - x \mid \leq 0.\underbrace{00...00}_{n}\frac{B}{2}\cdot B^e = \frac{B}{2}\cdot B^{e-n-1},$$

Beispiel 2.4 [6]:

- Sei $x = 180.1234567 = 0.1801234567 \cdot 10^3$. Gerundet auf eine siebenstellige Mantisse ($n = 7$) erhält man $rd(x) = 0.1801235 \cdot 10^3$ und es gilt wegen $e = 3$

$$| \, rd(x) - x \, | = 0.\underbrace{0000000}_{n=7}433 \cdot 10^3 = 0.433 \cdot 10^{-4} \leq 0.5 \cdot 10^{-4}$$

Aufgabe 2.6

1. Vergewissern Sie sich anhand einfacher Zahlenbeispiele, dass die Rundung ein besseres Verfahren für die Abbildung einer reellen Zahl auf eine Maschinenzahl darstellt als einfaches Abschneiden der überzähligen Ziffern, wie in den früheren Beispielen in Kap. 2.3.2. Was ist der maximale Fehler, der durch das Abschneiden auftreten kann?

2. Wir kennen die (allgemeinen) Rundungsregeln für das Dezimalsystem. Verallgemeinern sie diese für eine beliebige gerade Basis B. Runden Sie anschliessend die folgenden Zahlen auf eine vierstellige Mantisse, berechnen Sie den absoluten Fehler der Rundung und vergewissern Sie sich, dass $| \, rd(x) - x \, | \leq \frac{B}{2} \cdot B^{e-n-1}$. Gilt diese Relation (bei gleichen Rundungsregeln) auch für ungerade Basen?

   a) $(11.0100)_2$ b) $(11.0110)_2$ c) $(11.111)_2$ d) $(120.212)_3$ e) $(120.222)_3$ f) $(0.FFFFF)_{16}$

Für die Berechnungen bedeutet das, dass jede einzelne Operation (+, -, *, ...) auf n+1 genau gerechnet wird und das Ergebnis auf n Stellen gerundet wird (*n-stellige Gleitpunktarithmetik*). Jedes Zwischenergebnis wird also gerundet, nicht erst das Endergebnis. Das bedeutet auch, dass die einzelnen Rundungsfehler durch die Rechnung weitergetragen werden und allenfalls das Endergebnis verfälschen können. Für den maximal auftretenden relativen Fehler bei der Rundung ergibt sich bei n-stelliger Gleitpunktarithmetik im Dezimalsystem:

Definition 2.3: Maschinengenauigkeit

- Die Zahl $eps := 5 \cdot 10^{-n}$ heisst **Maschinengenauigkeit**. Bei allgemeiner Basis B gilt $eps := \frac{B}{2} \cdot B^{-n}$. Sie gibt den maximalen relativen Fehler, der durch Rundung entstehen kann.
- Alternative Definition: Die Maschinengenauigkeit ist die kleinste positive Maschinenzahl, für die auf dem Rechner $1 + eps \neq 1$ gilt.

$$| \, \frac{rd(x) - x}{x} \, | \leq 5 \cdot 10^{-n} \; (da \; x \geq 10^{e-1}).$$

Bemerkungen:

- Der Begriff Maschinengenauigkeit impliziert nicht, dass der Rechner nicht mit deutlich kleineren Zahlen $x < eps$ noch 'genau' rechnen kann.

Beispiel 2.5:

- Am Freitag, dem 25. November 1983, schloss der Aktienindex von Vancouver bei 524.811 Punkten und eröffnete am folgenden Montag, dem 28. November, bei 1098.892 Punkten. Was war passiert? Seit dem Start im Januar 1982 bei 1000 Punkten war der Aktienindex kontinuierlich gefallen, trotz florierendem Handel und guter Wirtschaftslage. Der Index wurde ca. 3000 mal am Tag neu berechnet, jeweils auf vier Dezimalstellen genau. Doch statt auf drei Dezimalstellen zu runden, wurde die vierte Dezimalstelle einfach abgeschnitten. Der dabei

maximal mögliche Fehler von 0.0009 mutet zwar klein an, doch bei 3000 Wiederholungen pro Tag konnte sich dieser Abschneidefehler auf bis zu $0.0009 \cdot 3000 = 2.7$ Punkte pro Tag aufsummieren. Über die Zeitspanne von fast zwei Jahren verlor der Index so fast die Hälfte seines Wertes. Dies wurde am 28. November basierend auf korrekter Rundung korrigiert. Grössere Auswirkungen hatte diese Korrektur offenbar nicht, da zum damaligen Zeitpunkt das Volumen an Derivaten gering war.

Aufgabe 2.7 [1]:

1. Gesucht ist eine Näherung $\widetilde{x}$ zu $x = \sqrt{2} = 1.414213562...$ mit einem absoluten Fehler von höchstens 0.001.

2. Es soll 2590 + 4 + 4 in 3-stelliger Gleitpunktarithmetik gerechnet werden (im Dezimalsystem), einmal von links nach rechts und einmal von rechts nach links. Wie unterscheiden sich die Resultate?

   Anhand der Lösung sieht man, dass es bei der n-stelligen Gleitpunktarithmetik auf die Reihenfolge der Operationen ankommt, im Unterschied zum exakten Rechnen. Als Faustregel kann man festhalten:

   Beim Addieren sollte man die Summanden in der Reihenfolge aufsteigender Beträge sortieren

3. Berechnen Sie $s_{300} := \sum_{i=1}^{300} \frac{1}{i^2}$ sowohl auf- als auch absteigend, je einmal mit 3-stelliger und 5-stelliger Gleitpunktarithmetik (in MATLAB können Sie eine Zahl x auf 3 Stellen reduzieren z.B. mit dem Befehl

   string2num(num2string(x,3))).

4. Es ist $\lim_{n\to\infty}(1+\frac{1}{n})^n = e$. Erstellen Sie eine Tabelle mit ihrem Rechner oder MATLAB für n = 1, 10, 100, ... für den Ausdruck $(1+\frac{1}{n})^n$sowie den absoluten und relativen Fehler. Erklären Sie Ihre Beobachtungen.

   Lösung:

n	$(1+\frac{1}{n})^n$	absoluter Fehler	relativer Fehler
10^0			
10^2			
10^3			
10^4			
10^5			
10^6			
10^8			
10^9			
10^{10}			
10^{15}			

5. Überlegen Sie sich einen kurzen iterativen Algorithmus, der die Maschinengenauigkeit Ihres Rechners prüft. Schliessen Sie aus dem Ergebnis, ob Ihr Rechner im Dual- oder Dezimalsystem rechnet und mit welcher Stellenzahl er operiert.

2.4.2 Fehlerfortpflanzung bei Funtionsauswertungen / Konditionierung

Wir haben gesehen, dass ein Rundungsfehler durch die Abbildung einer reellen Zahl x auf ihre Maschinenzahl $\widetilde{x}$ in die Berechnungen einfliesst. Soll nun eine Funktion $f : \mathbb{R} \longrightarrow \mathbb{R}$ an der Stelle x ausgewertet werden, wird ein zusätzlicher Fehler dadurch generiert, dass nicht $f(x)$, sondern $f(\widetilde{x})$ berechnet wird[10]. Für den fehlerbehafteten Wert $\widetilde{x}$ können wir den Fehler quantifizieren als $\Delta x = \widetilde{x} - x$ (vgl. Def. 2.2) oder

$$\widetilde{x} = x + \Delta x$$

Nun wollen wir den absoluten Fehler $\mid f(\widetilde{x}) - f(x) \mid$ und den relativen Fehler $\frac{|f(\widetilde{x})-f(x)|}{|f(x)|}$dieser Funktionsauswertung berechnen. Unter der Annahme, dass die Funktion f stetig differenzierbar ist, können wir $f(\widetilde{x})$ gemäss Taylor

[10] Tatsächlich ist die Situation noch schlimmer, denn bereits die Umsetzung der reellwertigen Funktion selbst kann in einem endlichen Rechner zu zusätzlichen Fehlern führen. Darauf wollen wir an dieser Stelle aber nicht eingehen.

entwickeln. Aus der allg. Taylor-Reihe (bekannt aus der Analysis) einer Funktion $f(x)$ um den Entwicklungspunkt x_0

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i$$

erhalten wir für die Entwicklung von $f(\widetilde{x})$ um den Entwicklungspunkt x

$$f(\widetilde{x}) = f(x + \Delta x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!}(\Delta x)^i = f(x) + f'(x)\Delta x + \frac{f''(x)}{2}(\Delta x)^2 + ...$$

wobei wir in der Taylor-Reihe x durch $\tilde{x}$ und x_0 durch $x$ersetzt haben.

Unter der Annahme $\Delta x \ll 1$ können die höheren Fehlerterme $(\Delta x)^n$ für $n \geq 2$ vernachlässigt werden und es ergibt sich die folgende Näherung

$$\begin{aligned} f(\widetilde{x}) - f(x) &\approx f'(x)\Delta x \\ &\approx f'(x)(\tilde{x} - x) \end{aligned}$$

bzw. bei beidseitiger Division durch $f(x)$ und rechtsseitiger Multiplikation mit $\frac{x}{x}$:

$$\frac{f(\widetilde{x}) - f(x)}{f(x)} \approx \frac{f'(x) \cdot x}{f(x)} \cdot \frac{\tilde{x} - x}{x}$$

Wir erhalten also die folgenden Näherungen:

- Näherung für den **absoluten Fehler bei Funktionsauswertungen**:

$$| f(\tilde{x}) - f(x) | \approx | f'(x) | \cdot | \tilde{x} - x |$$

- Näherung für den **relativen Fehler bei Funktionsauswertungen**:

$$\frac{| f(\widetilde{x}) - f(x) |}{| f(x) |} \approx \frac{| f'(x) | \cdot | x |}{| f(x) |} \cdot \frac{| \tilde{x} - x |}{| x |}$$

Definition 2.4: Konditionszahl

- Den Faktor

$$K := \frac{| f'(x) | \cdot | x |}{| f(x) |}$$

  nennt man **Konditionszahl**.

- Man unterscheidet **gut konditionierte Probleme**, d.h. die Konditionszahl ist klein, und **schlecht konditionierte Probleme** (ill posed problems) mit grosser Konditionszahl. Bei gut konditionierten Problemen wird der relative Fehler durch die Auswertung der Funktion nicht grösser.

Bemerkungen:

- $\widetilde{x}$ kann generell als fehlerbehafteter Näherungswert für x angesehen werden. Ob der Fehler nun durch Rundung oder andere Effekte verursacht wird (z.B. durch fehlerhafte Messungen) ist hierbei nicht von Belang.
- Bei Funktionsauswertungen pflanzt sich der absolute Fehler in x näherungsweise mit dem Faktor $f'(x)$ fort. Falls $| f'(x) | > 1$ wird der absolute Fehler grösser, falls $| f'(x) | < 1$ kleiner.
- Bei Funktionsauswertungen pflanzt sich der relative Fehler in x näherungsweise mit der Konditionszahl fort.

Beispiel 2.5 [1]:

- Was lässt sich über die Fehlerfortpflanzung des absoluten Fehlers für die Funktion $f(x) = \sin(x)$ aussagen? Da$|\ f'(x)\ | = |\ \cos(x)\ | \leq 1$, folgt dass der absolute Fehler in den Funktionswerten nicht grösser sein kann als in den $x-$Werten sondern eher kleiner.
- Bei der Funktion $f(x) = 1000 \cdot x$ wird wegen $f'(x) = 1000$ der absolute Fehler in der Funktionsauswertung um den Faktor 1000 grösser.
- Die Konditionszahl für das Quadrieren, also $f(x) = x^2$, ist $K = \frac{|2x| \cdot |x|}{|x^2|} = 2$, d.h. der relative Fehler verdoppelt sich in etwa. Dies ist aber noch keine schlechte Konditionierung.
- Das Polynom
$$P(x) = (x-1)^3 = x^3 - 3x^2 + 3x - 1$$
hat die dreifache Nullstelle $x_1 = x_2 = x_3 = 1$. Das nahe bei P liegende Polynom
$$Q(x) = x^3 - 3.000001x^2 + 3x - 0.999999$$
hat die Nullstellen $x_1 = 1$, $x_2 \backsimeq 1.001414$, $x_3 \backsimeq 0.998586$. Während die Koeffizienten von P um 10^{-6} gestört wurden, haben sich die Nullstellen um 10^{-3}verändert, d.h. die Störung wurde um einen Faktor 1000 verstärkt. Die Nullstellen von P sind schlecht konditioniert. Schaut man sich den Graphen von Q in der Umgebung der Nullstelle an, sieht man, wie die Rundungsfehler der Maschinenzahlen zu einer Zackenlinie führen.

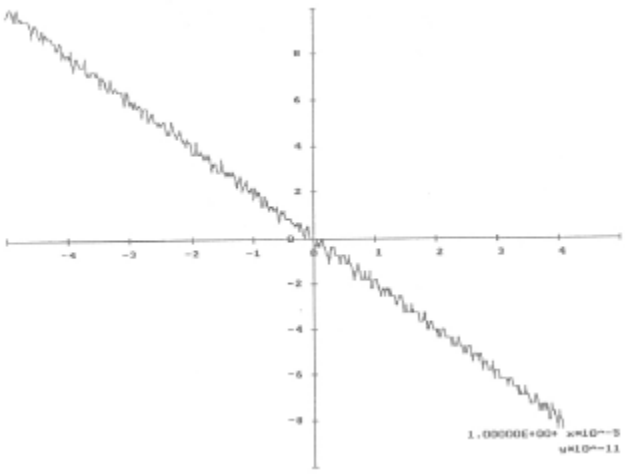


Abbildung 2.5: Das Polynom Q in der Umgebung von $x = 1$. (Abbildung aus [2])

Betrachten wir nun die relativen Fortpflanzungsfehler für die grundlegenden arithmetischen Operationen.

Fehlerfortpflanzung bei Summation

Für
$$f(x) = x + c\ (c \in \mathbb{R})$$
haben wir für die Ableitung $f'(x) = 1$ und damit
$$\frac{|\ f(\tilde{x}) - f(x)\ |}{|\ f(x)\ |} \approx \frac{|\ x\ |}{|\ x + c\ |} \cdot \frac{|\ \tilde{x} - x\ |}{|\ x\ |}$$
bzw.
$$K = \frac{|\ x\ |}{|\ x + c\ |}$$
Wenn x und die Konstante c gleiches Vorzeichen haben, gilt $K \leq 1$, dann haben wir also ein gut konditioniertes Problem. Was passiert aber, wenn x und c entgegengesetzte Vorzeichen haben und betragsmässig fast gleich gross

sind? Dann wird $|\ x + c\ |$sehr klein und somit K sehr gross, die Addition (bzw. Subtraktion) ist dann schlecht konditioniert. Dieses Phänomen nennt man auch ***Auslöschung***. Es tritt immer dann auf, wenn ungefähr gleich grosse fehlerbehaftete Zahlen voneinander abgezogen werden und das Resultat anschliessend normiert wird.

Beispiel 2.6 [7]:

- Für die beiden reellen Zahlen $r = \frac{3}{5}$ und $s = \frac{4}{7}$ mit den normierten gerundeten Repräsentationen mit fünfstelliger Mantisse, also $\tilde{r} = (0.10011)_2$ und $\tilde{s} = (0.10010)_2$, berechnen wir die Differenz $r - s = \frac{1}{35}$ näherungsweise als

$$0.10011 \cdot 2^0 - 0.10010 \cdot 2^0 = 0.00001 \cdot 2^0 = 0.10000 \cdot 2^{-4} = \frac{1}{32}.$$

  Für den relativen Fehler erhalten wir

$$\frac{\frac{1}{32} - \frac{1}{35}}{\frac{1}{35}} = 0.0938 \approx 9.4\%$$

  was viel ist (zum Vergleich, dies ist rund dreimal grösser als die Maschinengenauigkeit $2^{-5} = 0.0313$). Für die Berechnung mit dreistelliger Mantisse erhalten wir

$$0.101 - 0.101 = 0$$

  und damit einen Fehler von 100%.

Beispiel 2.7 [6]:

- Gegeben seien die drei Werte

$$\begin{aligned} x_1 &= 123.454 \cdot 10^9 \\ x_2 &= 123.446 \cdot 10^9 \\ x_3 &= 123.435 \cdot 10^9 \end{aligned}$$

  Legt man eine 5-stellige dezimale Gleitpunktarithmetik zugrunde, so wird durch Rundung

$$\begin{aligned} \tilde{x}_1 &= 0.12345 \cdot 10^{12} \\ \tilde{x}_2 &= 0.12345 \cdot 10^{12} \\ \tilde{x}_3 &= 0.12344 \cdot 10^{12} \end{aligned}$$

  und man erhält statt

$$\begin{aligned} x_1 - x_2 &= x_1 + (-x_2) = 8 \cdot 10^6 \\ x_1 - x_3 &= x_1 + (-x_3) = 19 \cdot 10^6 \end{aligned}$$

  die fehlerhaften Werte

$$\begin{aligned} \tilde{x}_1 - \tilde{x}_2 &= 0 \\ \tilde{x}_1 - \tilde{x}_3 &= 10 \cdot 10^6 \end{aligned}$$

Aufgabe 2.8:

1. Untersuchen Sie, ob die Multiplikation und die Division zweier Zahlen gut oder schlecht konditionierte Funktionsauswertungen sind.

Kapitel 3

Numerische Lösung von Nullstellenproblemen

In diesem Kapitel behandeln wir Verfahren zur näherungsweisen Lösung von nichtlinearen Gleichungen mit einer Unbekannten (die Lösung linearer Gleichungen einer Variablen ist trivial). Wie wir sehen werden, ist die Lösung von nichtlinearen Gleichungen mit einer Unbekannten äquivalent zur Bestimmung der Nullstellen einer Funktion $f : \mathbb{R} \to \mathbb{R}$mit $f(x) = 0$, deshalb der Titel.

Lernziele:

- Sie können das Bisektionsverfahren anwenden und in MATLAB programmieren.
- Sie können die Begriffe Fixpunktgleichung, Fixpunktiteration sowie anziehender bzw. abstossender Fixpunkt definieren.
- Sie können zu einer konkreten Aufgabenstellung die Fixpunktgleichung aufstellen und die entsprechende Iteration durchführen.
- Sie können dabei auftretende Fehler mittels des Banachschen Fixpunktsatzes quantifizieren.
- Sie können das Newtonverfahren, das vereinfachte Newtonverfahren sowie das Sekantenverfahren anwenden.
- Sie verstehen der Begriff der Konvergenzordnung.

Bemerkung: Die im Kapitel 2 verwendete Normierung $x = \pm 0.m_1 m_2 m_3 ... m_n \cdot B^e$ haben wir im Zusammenhang mit der Theorie der Rechnerarithmetik und der maschinendarstellbaren Zahlen zu verschiedenen Basen eingeführt. In den Ingenieurwissenschaften werden numerische Resultate aber meist als Dezimalzahlen in der Potenzschreibweise dargestellt mit vier Nachkommastellen, wobei die erste Ziffer vor dem Komma *ungleich 0* sein muss (für $x \neq 0$). Sofern wir im weiteren mit numerischen Resultaten arbeiten und es nicht ausdrücklich anders verlangt ist, werden wir also im Dezimalsystem i.d.R. mit der Normierung $x = \pm m_1.m_2 m_3 m_4 m_5 \cdot 10^{\pm e}$ mit $m_1 \neq 0$ (für $x \neq 0$) arbeiten.

3.1 Zur historischen Entwicklung

Die Fragestellung der Lösung nichtlinearer Gleichungen begleitet die (numerische) Mathematik seit ihren Anfängen. Die Babylonier (und vermutlich bereits die Ägypter) beschäftigten sich in ihrer auf die Geometrie fokussierten Mathematik unter anderem mit der Frage, wie gross die Seitenlängen x eines Quadrates mit der gegebenen Fläche A sind, also mit der Lösung der nichtlinearen Gleichung $x^2 = A$ (vgl. die Lösung in Abb. 1.1 für $A = 2$). Eng damit verwandt ist natürlich die Fragestellung des Flächeninhaltes eines rechtwinkligen Dreiecks. Der nach dem griechischen Philosophen Pythagoras von Samos (um 570-510 v.Chr.) benannte Satz $a^2 + b^2 = c^2$ war den Babyloniern bereits rund 1000 Jahre früher bekannt. Die Übersetzung einer babylonischen Tontafel (ca. 1900-1600 v.Chr.) im

Britischen Museum lautet[1]:

4 is the length and 5 the diagonal. What is the breadth?
Its size is not known.
4 times 4 is 16.
5 times 5 is 25.
You take 16 from 25 and there remains 9.
What times what shall I take in order to get 9?
3 times 3 is 9.
3 is the breadth.

Der Grieche Heron von Alexandria (1. Jhr. n.Chr.) beschrieb in seinem Werk *Metrika* (Buch der Messung) das nach ihm benannte Näherungsverfahren von Heron zur iterativen Berechnung einer (beliebigen) Quadratwurzel $x = \sqrt{A}$ für $A > 0$ mit der Iterationsvorschrift (für einen Startwert $x_0 \neq 0$):

$$x_{n+1} = \frac{x_n + \frac{A}{x_n}}{2}$$

Im Mittelalter konzentrierte man sich auf die Nullstellensuche von Polynomen. Der italienische Mathematiker Girolamo Cardano (1501-1576) veröffentlichte als erster Lösungsformeln (die Cardanischen Formeln) für kubische Gleichungen und zusätzlich Lösungen für Gleichungen vierten Grades. Bei seinen Berechnungen stiess er auf die komplexen Zahlen und zeigte (entgegen der damaligen Lehrmeinung), dass auch mit negativen Zahlen gerechnet werden kann.

Isaac Newton beschrieb im Zeitraum 1664 bis 1671 einen neuen Algorithmus zur Nullstellenbestimmung von Polynomen dritten Grades. Sein Landsmann und Mathematiker Thomas Simpson (1710-1761) formulierte dieses Verfahren unter Benutzung der Ableitung in der Iterationsvorschrift

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

was wir heute als Newton-Verfahren bezeichnen (vgl. Kap. 3.5). Tatsächlich ist das Newton-Verfahren äquivalent zum Heron-Verfahren für die Bestimmung der Nullstellen der Funktion $f(x) = x^2 - A$. Generell lässt sich das Newton-Verfahren natürlich (unter gewissen Einschränkungen bzgl. der Konvergenz) für beliebige stetig differenzierbare Funktionen $f(x)$ einsetzen, nicht nur für Polynome.

Wahrscheinlich im Zusammenhang mit dem Beweis des Zwischenwertsatzes der Analysis (siehe Kap. 3.2) konstruierte der böhmische Priester und Mathematiker Bernard Bonzano (1781-1848) um 1817 das Bisektionsverfahren[2], welches es durch fortlaufende Intervallhalbierung zuverlässig (aber langsam) erlaubt, eine Nullstelle einer stetigen Funktion zu finden (ohne Benutzung der Ableitung wie im Newton-Verfahren). Der polnische Mathematiker Stefan Banach (1892-1945) formulierte 1922 den Banachschen Fixpunktsatz zur Theorie der Fixpunktiterationen (siehe Kap. 3.4), die zur Lösung von Nullstellenproblemen in einem weit gefassten Bereich von einfachen Funktionen bis hin zu linearen oder nichtlinearen Gleichungssystemen und Differentialgleichungen reicht.

Modernere Verfahren zur Nullstellenbestimmung sind meist Kombinationen der hier bereits erwähnten und in den folgenden Unterkapiteln detaillierter vorgestellten Verfahren.

3.2 Problemstellung

Gegeben sei eine stetige Funktion $f : \mathbb{R} \to \mathbb{R}$. Gesucht sei ein Näherungswert für die (bzw. für eine) Nullstelle $\overline{x}$ von f. Natürlich ist eine Gleichung der Form $g(x) = h(x)$ äquivalent zu $f(x) \equiv g(x) - h(x) = 0$. Geometrisch bedeutet das, dass $f(x)$ an der Stelle $\overline{x}$ die x-Achse schneidet.

Aufgabe 3.1:

- Die nichtlineare Gleichung $x = \cos(x)$ lässt sich als Nullstellenproblem von $f(x) \equiv x - \cos(x) = 0$ interpretieren. Lösen Sie für$x \in [0, 1]$ auf graphischem Weg einmal die Gleichung $x = \cos(x)$ und dann die Gleichung $f(x) = x - \cos(x) = 0$.

[1] John J. O'Connor, Edmund F. Robertson: Pythagoras's theorem in Babylonian mathematics. In: MacTutor History of Mathematics archive (englisch) unter http://www-history.mcs.st-andrews.ac.uk/Indexes/HistoryTopics.html

[2] Edwards, C. H. (1979). Bolzano, Cauchy, and Continuity. The Historical Development of the Calculus (pp. 308, 309). New York, NY: Springer New York.

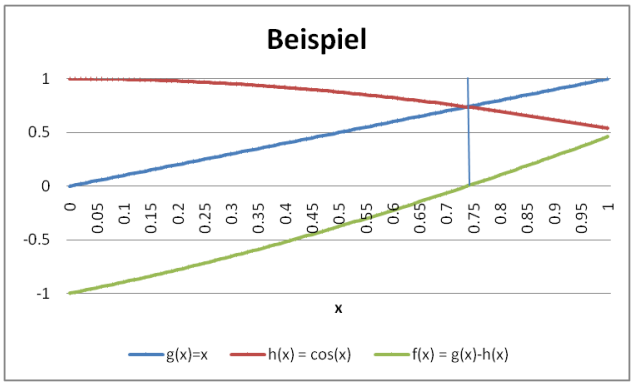


Abbildung 3.1: Graphische Lösung zu Aufgabe 3.1

Folgende Fragen sollten aber erst geklärt werden, bevor ein solches Problem gelöst werden kann:

1. Gibt es überhaupt eine Nullstelle von $f(x)$, und wenn ja, in welchem Bereich?
2. Gibt es mehrere Nullstellen? Wenn ja, welche davon sollten mit dem Rechner gefunden werden?

Zur Lösung dient der folgende Satz aus der Analysis:

Satz 3.1: Nullstellensatz von Bolzano

- Sei $f : [a, b] \to \mathbb{R}$ stetig mit $f(a) \leq 0 \leq f(b)$ oder $f(a) \geq 0 \geq f(b)$. Dann muss f in $[a, b]$ eine Nullstelle besitzen.

Wenn man also auf dem Intervall $[a, b]$ einen Vorzeichenwechsel von f feststellt, d.h. $f(a) \cdot f(b) < 0$, dann besitzt f in diesem Intervall mindestens eine Nullstelle. Im folgenden beschreiben wir ein Verfahren, dass diesen Umstand benutzt.

3.3 Bisektionsverfahren

Wir berechnen zunächst $x_1 = (a + b)/2$ und prüfen ob $f(x_1) > 0$. Wenn ja, dann verwenden wir $[a, x_1]$ als neues Näherungsintervall, wenn nein, dann muss eine Nullstelle in $[x_1, b]$ liegen. Das neue Intervall nennen wir $[a_1, b_1]$. Wiederholung des Verfahrens mit dem neuen Intervall liefert eine Intervallschachtelung, die eine Nullstelle bestimmt. Auf diese Weise kann man eine Nullstelle beliebig genau annähern. Dieses einfache Verfahren zur Bestimmung einer Nullstelle einer stetigen Funktion nennt man Bisektionsverfahren. Formal kann man das folgendermassen definieren:

Satz 3.2: Bisektionsverfahren [1]

- Gegeben sei eine stetige Funktion $f : [a, b] \longrightarrow \mathbb{R}$ mit $f(a) \cdot f(b) < 0$. In jedem der über die Rekursion für $i = 0, 1, ...$erzeugten Intervalle

$$\begin{aligned} [a_0,\, b_0] &= [a,\, b]; \\ [a_{i+1},\, b_{i+1}] &= \begin{cases} \left[a_i,\, \frac{a_i+b_i}{2}\right] & \text{falls } f\left(\frac{a_i+b_i}{2}\right) \cdot f(a_i) \leq 0 \\ \left[\frac{a_i+b_i}{2},\, b_i\right] & \text{sonst} \end{cases} \end{aligned}$$

befindet sich eine Nullstelle von f und es gilt

$$b_i - a_i = \frac{b-a}{2^i}, \text{ insbesondere also } \lim_{i\to\infty}(b_i - a_i) = 0$$

Allerdings gibt es wesentlich schnellere Verfahren, eine Nullstelle zu berechnen. Wir können es aber dazu verwenden, einen Überblick über die Lage der Nullstellen zu verschaffen, indem man nur einige Schritte durchführt. Zudem hat das Bisektionsverfahren einige sehr vorteilhafte Eigenschaften:

- Es funktioniert für allgemeine stetige Funktionen.
- Es liefert immer ein Ergebnis, vorausgesetzt, dass man geeignete Startwerte a und b finden kann (man sagt, dass das Verfahren "global konvergiert").
- Die Anzahl der Schritte, nach der die gewünschte Genauigkeit erreicht ist, hängt nur von a und b aber nicht von f ab.

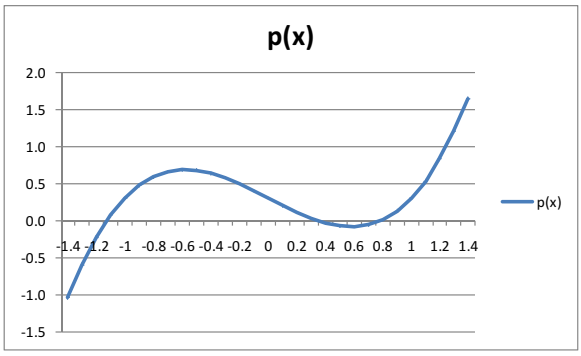
Beispiel 3.1

- Gesucht sind Intervalle, in denen sich die Nullstellen von $p(x) = x^3 - x + 0.3$ befinden.

  Lösung: p ist ein Polynom vom Grad 3 und hat maximal 3 Nullstellen. Wegen der Ähnlichkeit zum Polynom $q(x) = x^3 - x$, welches die Nullstellen -1, 0, 1 besitzt, suchen wir in dieser Umgebung. Um Vorzeichenänderungen festzustellen, berechnen wir im Intervall [-3, 3] einige Funktionswerte:

x	-3.0	-2.5	-2.0	-1.5	-1.0	-0.5	0.0	0.5	1.0	1.5	2.0	2.5	3.0
$p(x)$	-23.7	-12.8	-5.7	-1.6	0.3	0.7	0.3	-0.1	0.3	2.2	6.3	13.4	24.3

- Wir haben den ersten Vorzeichenwechsel von $p(x)$ im Intervall [-1.5, -1.0], den nächsten im Intervall [0.0, 0.5] und den dritten im Intervall [0.5, 1.0]. Nach dem Zwischenwertsatz gibt es also in jedem dieser Intervalle eine Nullstelle. Da es nur maximal 3 Nullstellen geben kann, enthält also jedes Intervall genau eine und wir haben alle Nullstellen gefunden.



- Wir wollen jetzt die Nullstelle im Intervall [0.0, 0.5] bis auf eine Stelle nach dem Komma bestimmen.

  Lösung: Der Mittelwert der Intervallgrenzen ist (0.0+0.5)/2 =0.25. Der Funktionswert $p(0) = 0.3$ und $p(0.25) = 0.07$ ist grösser als Null, also liegt die Nullstelle im Intervall [0.25,0.5]. Der Mittelwert dieser neuen Intervallgrenzen ist (0.25+0.5)/2=0.375, und p(0.375)=-0.020 ist kleiner als Null, also liegt die Nullstelle im Intervall [0.25,0.375]. Nochmaliges Ausführen ergibt (0.25+0.375)/2 = 0.3125 und mit p(0.3125)=0.018 > 0 liegt die Nullstelle im Intervall [0.3125,0.375]. Damit muss die Nullstelle also 0.3_ _ sein und wir haben sie auf eine Nachkommastelle genau bestimmt.

Aufgabe 3.2 [1]:

1. Bestimmen Sie wie im obigen Beispiel die beiden anderen Nullstellen von $p(x)$ auf eine Nachkommastelle genau.
2. Bestimmen Sie mit dem Bisektionsverfahren die Lösung von $f(x) = x^2 - 2 = 0$ auf einem geeigneten Startintervall auf vier Nachkommastellen genau.
3. Optional: Übersetzen Sie den obigen Satz zum Bisektionsverfahren in ein MATLAB Programm.

3.4 Fixpunktiteration

Die Fixpunktiteration ist eine weitere einfache Methode zur Bestimmung von Nullstellen. Sie beruht auf der Idee, dass für nichtlineare Gleichungen der Form $f(x) = F(x) - x$ die Bedingung $f(\overline{x}) = 0$ genau dann erfüllt ist, wenn $F(\overline{x}) = \overline{x}$.

Definition 3.1: Fixpunktgleichung / Fixpunkt [1]

- Eine Gleichung der Form $F(x) = x$ heisst **Fixpunktgleichung**.
- Ihre Lösungen $\overline{x}$, für die $F(\overline{x}) = \overline{x}$ erfüllt ist, heissen **Fixpunkte** (da die Funktion F die Punkte $\overline{x}$ auf sich selbst abbildet.

Anstelle eines Nullstellenproblems kann man also ein dazu äquivalentes Fixpunktproblem betrachten. Dazu muss aber $f(x) = 0$ in die Fixpunktform $F(x) = x$ gebracht werden, wozu es viele Möglichkeiten gibt. Bei dieser Überführung muss unbedingt auf Äquivalenz geachtet werden, d.h. die Lösungsmenge darf nicht verändert werden.

Beispiel 3.2:

- Die Gleichung $p(x) = x^3 - x + 0.3$ soll in Fixpunktform gebracht werden.
  Lösung: Die einfachste Möglichkeit ist $p(x) = 0 \iff F(x) \equiv x^3 + 0.3 = x$
  Aber auch $F(x) \equiv \sqrt[3]{x - 0.3} = x$ ist möglich.
- Die Gleichung $x = \cos(x)$, die wir weiter oben graphisch gelöst haben, ist bereits in der Fixpunktform.

Definition 3.2: Fixpunktiteration [1]

- Gegeben sei $F : [a, b] \to \mathbb{R}$, mit $x_0 \in [a, b]$. Die rekursive Folge

$$x_{n+1} \equiv F(x_n), \qquad n = 0, 1, 2, ...$$

  heisst Fixpunktiteration von F zum Startwert x_0.

Die 'Hoffnung' ist, dass die erzeugte Folge gegen einen Fixpunkt von F konvergiert. Fixpunktiterationen sind leicht durchzuführen und jeder Iterationsschritt benötigt nur eine Funktionsauswertung. Aus der generellen Form $F(x) = x$ folgt aber auch direkt, dass sich graphisch die Lösung ergibt als die Schnittpunkte zwischen den beiden Funktionen $y = F(x)$ und $y = x$. Allerdings können sich zwei Fixpunktiterationen zum gleichen Nullstellenproblem bzgl. ihrem Konvergenzverhalten unterscheiden.

Beispiel 3.3:

- Berechnen Sie Nullstellen von $p(x) = x^3 - x + 0.3$ mittels Fixpunktiteration.
  Lösung: Die Fixpunktiteration lautet $x_{n+1} = F(x_n) = x_n^3 + 0.3$. Wir wissen bereits aus der letzten Aufgabe, wo wir die Nullstellen zu vermuten haben, also wählen wir Startwerte aus der Umgebung, z.B. -1, 0, 1. Wir erhalten die folgende Tabelle (aus [1]):

n	x_n	x_n	x_n
0	-1	0	1
1	-0.7	0.3	1.3
2	-0.043	0.327	2.497
3	0.299920493	0.334965783	15.86881747
4	0.3269785388	0.3375838562	3996.375585
5	0.3349588990	0.3384720217	$\vdots$
6	0.3375815390	0.3387764750	$\vdots$
7	0.3384712295	0.3388812067	$\vdots$
8	0.3387762027	0.3389172778	$\vdots$
9	0.3388811129	0.3389297064	$\vdots$
10	0.3389172455	0.3389339894	$\vdots$

Während mit den beiden Startwerten -1 und 0 die Fixpunktiteration gegen 0.3389... konvergiert, divergiert sie für den Startwert 1. Auch für andere Startwerte würde man feststellen, dass die Folgen entweder gegen 0.3389... konvergieren oder dann divergieren. Die Nullstelle bzw. der Fixpunkt $x = 0.3389$ scheint die Iterationsfolgen anzuziehen, die beiden anderen Nullstellen aber nicht. Daher können sie mit dieser Iteration nicht angenähert werden.

Die obere Figur in Abbildung 3.2 zeigt die Fixpunktiteration in der Nähe des Fixpunktes $x = 0.3389$. Man sieht, dass die Folge schnell konvergiert. Was führt nun dazu, dass die Folge für die beiden anderen Fixpunkte nicht konvergiert?

Die untere Figur in Abbildung 3.2zeigt alle drei Schnittpunkte $y = F(x)$ und $y = x$. Die Vermutung liegt nahe, dass die Steigung der Funktion $y = F(x)$ verglichen mit derjenigen von $y = x$ an der Stelle der Fixpunkte $\overline{x}$ eine Rolle spielt. Dort, wo die Steigung von $F(x)$ kleiner ist als diejenige von $y = x$ (welche die Steigung 1 hat), scheint die Fixpunktiteration zu funktionieren, es muss also gelten $F'(\overline{x}) < 1$. Die Folge konvergiert schneller je kleiner $F'(\overline{x})$. Umgekehrt gilt, die Fixpunktiteration divergiert für $F'(\overline{x}) > 1$, wie es der Fall für die beiden anderen Fixpunkte ist. Diese sind nicht mit dieser Fixpunktiteration bestimmbar.

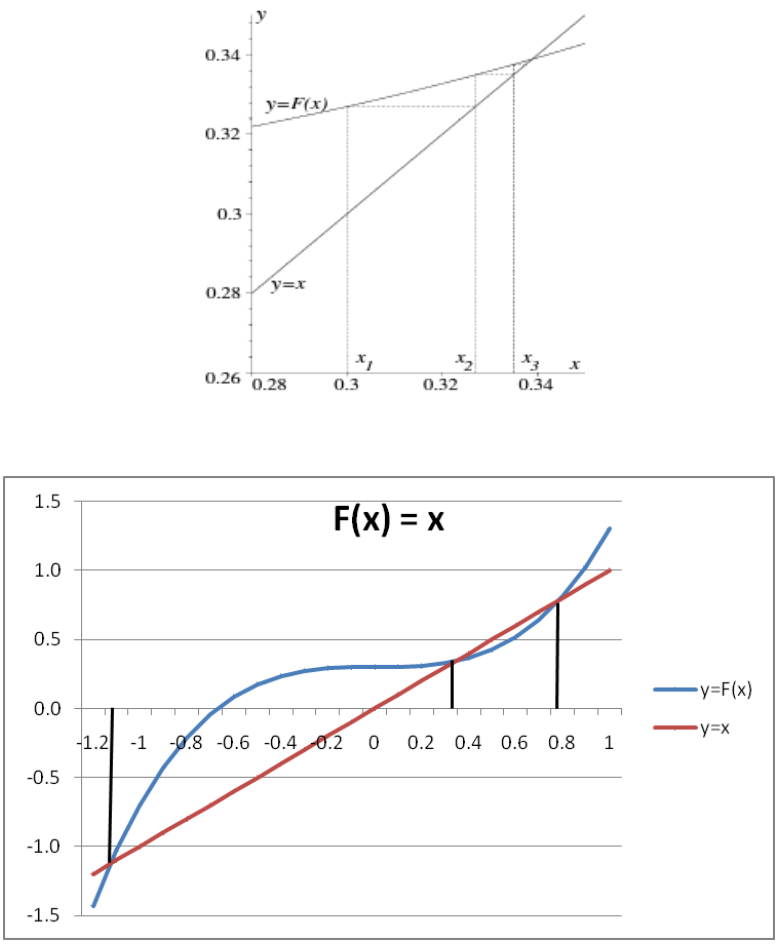


Abbildung 3.2: Oben: Fixpunktiteration zu $x_{n+1} = F(x_n) = x_n^3 + 0.3$ in der Umgebung des Fixpunktes $\overline{x}_2$ (aus [1]). Unten: Alle drei Nullstellen von $F(x)$.

Wir halten also fest:

Satz 3.2 zur Fixpunktiteration [1]:

- Sei $F : [a,b] \longrightarrow \mathbb{R}$ mit stetiger Ableitung F' und $\overline{x} \in [a,b]$ ein Fixpunkt von F. Dann gilt für die Fixpunktiteration $x_{n+1} = F(x_n)$:
  - Ist $\mid F'(\overline{x}) \mid < 1$, so konvergiert x_n gegen $\bar{x}$, falls der Startwert x_0 nahe genug bei $\bar{x}$ liegt. Der Punkt $\bar{x}$ heisst dann **anziehender Fixpunkt.**
  - Ist $\mid F'(\overline{x}) \mid > 1$, so konvergiert x_n für keinen Startwert $x_0 \neq \bar{x}$. Der Punkt $\bar{x}$ heisst dann **abstossender Fixpunkt.**

Aufgabe 3.3 [1]:

1. Überprüfen Sie anhand des obigen Satzes, welche der drei Fixpunkte $\overline{x}_1 = -1.125$, $\overline{x}_2 = 0.3389$, $\overline{x}_3 = 0.7864$ abstossend oder anziehend sind.
2. Bestimmen Sie anhand einer Fixpunktiteration die Lösung(en) von $x = \cos(x)$.
3. Prüfen Sie, ob der Fixpunkt $\overline{x}_3$=0.7864 für die Fixpunktiteration $x_{n+1} = F(x_n) = \sqrt[3]{x_n - 0.3}$ anziehend oder abstossend ist.

Was uns nun interessiert ist, welche Startwerte für eine Fixpunktiteration geeignet sind und was für Fehler wir für die n-te Fixpunktiteration erwarten müssen. Dazu dient uns der

Satz 3.3: Banachscher Fixpunktsatz [1]

- Sei $F : [a,b] \longrightarrow [a,b]$ (d.h. F bildet $[a,b]$ auf sich selber ab) und es existiere eine Konstante α mit $0 < \alpha < 1$ und
$$\mid F(x) - F(y) \mid \leq \alpha \mid x - y \mid \text{ für alle } x, y \in [a,b]$$
(d.h. F ist "Lipschitz-stetig" und "kontraktiv", α nennt man auch Lipschitz-Konstante). Dann gilt:
  - F hat genau einen Fixpunkt $\overline{x}$ in [a,b]
  - Die Fixpunktiteration $x_{n+1} = F(x_n)$ konvergiert gegen $\overline{x}$ für alle Startwerte $x_0 \in [a,b]$
  - Es gelten die Fehlerabschätzungen
$$\begin{aligned} \mid x_n - \overline{x} \mid &\leq \frac{\alpha^n}{1-\alpha} \mid x_1 - x_0 \mid \quad \text{a-priori Abschätzung} \\ \mid x_n - \overline{x} \mid &\leq \frac{\alpha}{1-\alpha} \mid x_n - x_{n-1} \mid \quad \text{a-posteriori Abschätzung} \end{aligned}$$

Bemerkungen:

- Aus $\mid F(x) - F(y) \mid \leq \alpha \mid x - y \mid$ für alle $x, y \in [a,b]$ folgt
$$\frac{\mid F(x) - F(y) \mid}{\mid x - y \mid} \leq \alpha,$$
wobei die linke Seite sämtliche möglichen Steigungen der Sekanten durch die beiden Punkte $(x, F(x))$ und $(y, F(y))$ für alle $x, y \in [a,b]$ darstellt. Aus diesem Grund kann man α als die grösstmögliche Steigung von $F(x)$ auf dem Intervall $[a,b]$ interpretieren, bzw.
$$\alpha = \max_{x_0 \in [a,b]} \mid F'(x_o) \mid$$

- Wählt man das Intervall $[a, b]$ sehr nahe um einen anziehenden Fixpunkt $\overline{x}$, so ist also $\alpha \approx \mid F'(\overline{x}) \mid$.
- In der Praxis gestaltet es sich meist schwierig, ein Intervall [a,b] zu finden, dass unter F auf sich selbst abgebildet wird. Hat man ein solches Intervall gefunden, dann sind die Fehlerabschätzungen aber recht nützlich. Wir werden diesen Satz nochmals im Zusammenhang mit der iterativen Lösung von linearen Gleichungssystemen in Kap. 4 aufgreifen.

Beispiel 3.4:

- Gesucht ist ein Intervall [a, b] und eine Konstante $\alpha < 1$, so dass der Banachsche Fixpunktsatz auf die Fixpunktiteration $x_{n+1} = F(x_n) = x_n^3 + 0.3$ anwendbar ist.

  Lösung: Wir wissen bereits, dass die Fixpunktiteration in der Nähe von $\overline{x} = 0.3389$ konvergiert. Also suchen wir in der Nähe davon ein geeignetes Intervall. Wir versuchen es zum Beispiel mit$[a, b] = [0, 0.5]$. Für jedes x in diesem Intervall gilt $F(x) = x^3 + 0.3 \geq 0.3$ und der maximale Funktionswert ist $F(0.5) = 0.125 + 0.3 = 0.425 \leq 0.5$. Also bildet F das Intervall $[0, 0.5]$ tatsächlich auf $[0, 0.5]$ ab, die erste Bedingung ist also erfüllt. Jetzt untersuchen wir, ob es eine Konstante $\alpha < 1$ gibt, so dass $\mid F(x) - F(y) \mid \leq \alpha \mid x - y \mid$für alle $x, y \in [0, 0.5]$ gilt. Aus der obigen Bemerkung wissen wir dass

  $$\alpha = \max_{x_0 \in [a,b]} \mid F'(x_o) \mid$$

  Also berechnen wir die Ableitung $F'(x)$ auf dem Intervall [0,0.5] und finden, dass der maximale Wert der Ableitung $\mid F'(x) \mid = 3x^2$ wegen ihrem monoton steigenden Verhalten bei $x = 0.5$ erreicht wird und dass $\mid F'(x) \mid = 3x^2 = 3 * 0.5^2 = 0.75 < 1$. Also setzen wir $\alpha = 0.75$.

Aufgabe 3.4 [1]:

1. Schätzen sie jetzt für das obige Beispiel mit der a-priori Abschätzung ab, wie viele Iterationen ausreichen sollten, um ausgehend von $x_0 = 0$ einen absoluten Fehler von max. 10^{-4}zu erhalten. Wenden Sie dann die a-posteriori Abschätzung an, um den absoluten Fehler zu erhalten.
2. Finden Sie mit Hilfe des Banachschen Fixpunktsatzes den Fixpunkt $\overline{x}_2$ für die Fixpunktiteration $x_{n+1} = F(x_n) = \sqrt[3]{x_n - 0.3}$ und den Startwert $x_0 = 0.7$.
3. Welche der beiden Fixpunktiterationen $x_{n+1} = F(x_n) = x_n^3 + 0.3,\ x_0 = 0$ und $x_{n+1} = F(x_n) = \sqrt[3]{x_n - 0.3}$, $x_0 = 0.7$ wird nach Ihrer Erwartung schneller konvergieren?

3.5 Das Newton-Verfahren

In diesem Abschnitt werden wir ein weiteres Verfahren zur Lösung nichtlinearer Gleichungssysteme betrachten, das bereits aus der Analysis bekannte Newton–Verfahren. Im Vergleich zu den bisher betrachteten Verfahren konvergiert dieses meist deutlich schneller. Wie wir im nächsten Abschnitt sehen werden, ist es quadratisch konvergent. Im Gegensatz zum Bisektions–Verfahren oder der Fixpunktiteration wird hier nicht allerdings nur die Funktion f selbst sondern auch ihre Ableitung benötigt. Wir setzen also voraus, dass f stetig differenzierbar ist.

Die Idee des Newton–Verfahrens ist wie folgt: Berechne die Tangente $g(x)$ von f im Punkt x_n, d.h. die Gerade

$$g(x) = f(x_n) + f'(x_n)(x - x_n).$$

Die Nullstelle von g sei x_{n+1}, dann gilt also

$$g(x_{n+1}) = 0 = f(x_n) + f'(x_n)(x_{n+1} - x_n).$$

Auflösen nach x_{n+1} liefert

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (n = 0, 1, 2, 3, ...).$$

Das gilt natürlich nur, wenn $f'(x_n) \neq 0$ erfüllt ist. Die Idee ist in Abb.3.3 graphisch dargestellt. Den Startwert sollte man in der Nähe der Nullstelle wählen, um eine schnelle Konvergenz zu erreichen. Die Konvergenz der Folge $(x_0, x_1, x_2, ...)$ ist sicher gegeben, wenn im Intervall $[a, b]$, in dem alle Näherungswerte liegen sollen, die Bedingung

$$\left| \frac{f(x) \cdot f''(x)}{[f'(x)]^2} \right| < 1$$

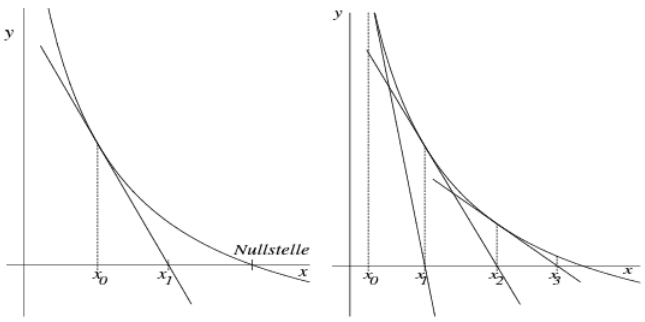


Abbildung 3.3: Newton-Verfahren (aus [1]).

erfüllt ist (hinreichende Konvergenzbedingung). In der Regel überprüft man diese Bedingung zumindest für den Startwert x_0. Ungeeignet sind Startwerte, in deren unmittelbarer Umgebung die Kurventangente nahezu parallel zur x-Achse verläuft.

Aufgabe 3.5 [1]:

1. Bestimmen Sie die Nullstellen von $f(x) = x^2 - 2 = 0$ näherungsweise mit dem Newton-Verfahren und dem Startwert $x_0 = 2$. Vergleichen Sie ihren Wert nach $n+1 = 4$ Iterationsschritten mit dem exakten Wert von $\sqrt{2}$. Auf wie vielen Nachkommastellen stimmt die Iterationslösung überein? Für welchen Startwert konvergiert die Folge nicht.?
2. Bestimmen Sie das Iterationsverfahren für $f(x) = x^2 - a = 0$ als Berechnungsmöglichkeit für $\sqrt{a}$ und vergleichen Sie das Resultat mit dem in Kap. 3.1 vorgestellten Heron-Verfahren.

Das Newton-Verfahren ist ein häufig verwendetes und schnelles Verfahren, um Nullstellen zu bestimmen. Es hat aber den Nachteil, dass man in jedem Schritt wieder eine Ableitung berechnen muss. Um das zu umgehen, kann man zu zwei vereinfachten Verfahren greifen, dem vereinfachten Newton-Verfahren und dem Sekantenverfahren.

3.5.1 Vereinfachtes Newton-Verfahren

Statt in jedem Schritt $f'(x_n)$ auszurechnen, kann man immer wieder $f'(x_0)$ verwenden. Damit ergibt sich die Rekursionsformel:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \qquad (n = 0, 1, 2, 3, ...).$$

Natürlich wird man erwarten, dass dieses Verfahren weniger gut funktioniert als das originale Newton-Verfahren. Tatsächlich konvergiert es langsamer.

3.5.2 Sekantenverfahren

Hier wird nicht der Schnittpunkt der Tangenten mit der x-Achse berechnet, sondern der Schnittpunkt von Sekanten ('Schneidenden') durch jeweils zwei Punkte $(x_0, f(x_0))$ und $(x_1, f(x_1))$ mit der x-Achse. Statt der Ableitung $f'(x_0)$ wird in der Iterationsformel dann die Steigung

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

der Sekanten eingesetzt und man erhält

$$x_2 = x_1 - \frac{f(x_1)}{\frac{f(x_1)-f(x_0)}{x_1-x_0}} = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} \cdot f(x_1)$$

und analog die Iterationsformel

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \cdot f(x_n) \qquad (n = 1, 2, 3, ...).$$

Das Sekantenverfahren ist veranschaulicht in Abbildung 3.4. Es benötigt zwei Startwerte x_0, x_1 und konvergiert langsamer, dafür benötigt es keine Ableitungen.

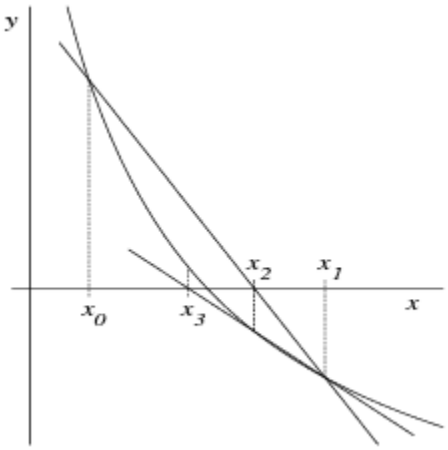


Abbildung 3.4: Sekantenverfahren (aus [1]).

3.6 Konvergenzgeschwindigkeit

Wie wir bereits angesprochen haben, unterscheiden sich die Nullstellenverfahren in ihrer Konvergenzgeschwindigkeit. Diese lässt sich durch den Begriff der Konvergenzordnung miteinander vergleichen.

Definition 3.3: Konvergenzordnung [1]

- Sei (x_n) eine gegen $\overline{x}$ konvergierende Folge. Dann hat das Verfahren die **Konvergenzordnung** q≥1 wenn es eine Konstante c $>$ 0 gibt mit

$$| x_{n+1} - \overline{x} | \leq c \cdot | x_n - \overline{x} |^q$$

für alle n. Falls $q = 1$ verlangt man noch $c < 1$. Im Fall $q = 1$ spricht man von linearer, im Fall $q = 2$ von quadratischer Konvergenz.

Beispiel 3.5:

- Sei $c = 1$ und $| x_0 - \overline{x} | \leq 0.1$. Es gilt dann also z.B. für quadratische Konvergenz nach jeder Iteration, dass der Fehler quadratisch abnimmt:

$$\begin{aligned} | x_1 - \overline{x} | &\leq | x_0 - \overline{x} |^2 \leq 0.1^2 = 10^{-2} \\ | x_2 - \overline{x} | &\leq | x_1 - \overline{x} |^2 \leq (10^{-2})^2 = 10^{-4} \\ | x_3 - \overline{x} | &\leq | x_2 - \overline{x} |^2 \leq (10^{-4})^2 = 10^{-8} \\ &\vdots \end{aligned}$$

Bemerkungen:

- Es gilt: für einfache Nullstellen konvergiert das Newton-Verfahren quadratisch, das vereinfachte Newton-Verfahren linear, und für das Sekantenverfahren gilt $q = (1 + \sqrt{5})/2 = 1.618....$

3.7 Fehlerabschätzung

Wir haben beim Banachschen Fixpunktsatz (Kap. 3.4) bereits eine Art der Fehlerabschätzung kennengelernt, benötigen dort aber die Konstante α. In der Praxis gibt es einfachere Methoden, um abzuschätzen, wie weit eine Näherung x_n nach der n-ten Iteration von der exakten Nullstelle entfernt ist. Eine einfache Möglichkeit ist es, die Funktion in der Umgebung der Näherung auszuwerten und zu überprüfen, ob ein Vorzeichenwechsel stattfindet.

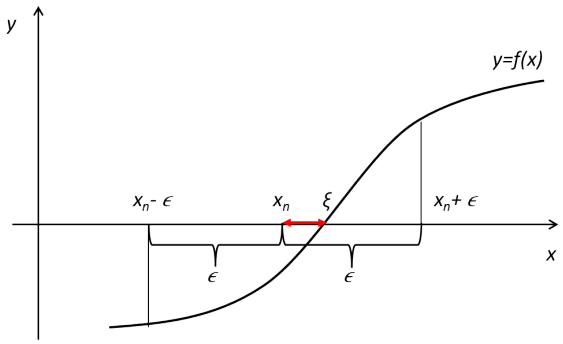


Abbildung 3.5: $f(x_n - \epsilon) \cdot f(x_n + \epsilon) < 0 \Rightarrow | \, x_n - \xi \, | < \epsilon$(nach [6]).

Daraus lässt sich gemäss dem Nullstellensatz (Kap. 3.2) schliessen, dass eine Nullstelle innerhalb des betrachteten Intervalls liegen muss und man kann abschätzen, wie weit die Näherung $x_n$von der tatsächlichen Nullstelle entfernt ist. Dieses Verfahren ist auf jedes iterative Verfahren zur Nullstellenbestimmung einer Funktion anwendbar, sofern die Nullstelle ungerade Ordnung hat (d.h. sie ist ein Schnittpunkt und nicht ein Berührungspunkt des Funktionsgraphen mit der x-Achse).

Sei x_n also ein iterativ bestimmter Näherungswert einer exakten Nullstelle ξ (ungerader Ordnung) der stetigen Funktion $f : \mathbb{R} \to \mathbb{R}$ und es gelte für eine vorgegebene Fehlerschranke / Fehlertoleranz $\epsilon > 0$

$$f(x_n - \epsilon) \cdot f(x_n + \epsilon) < 0,$$

dann muss gemäss dem Nullstellensatz im offenen Intervall $(x_n - \epsilon \, , \, x_n + \epsilon)$ eine Nullstelle ξ liegen und es gilt die Fehlerabschätzung (vgl. Abb. 3.5)

$$| \, x_n - \xi \, | < \epsilon$$

Beispiel 3.6 [1]:

- Es soll für $f(x) = x^2 - 2 = 0$ der Fehler für die Näherung $x_3$der Nullstelle mit dem Newton-Verfahren berechnet werden.

  Lösung: Es ist leicht zu sehen dass $f(x_3 - 10^{-5}) < 0$ und $f(x_3 + 10^{-5}) > 0$. Gemäss dem Nullstellensatz (Kap. 3.2) gibt es also eine Nullstelle $x \in [x_3 - 10^{-5}, \, x_3 + 10^{-5}]$ für die der absolute Fehler $| \, x - x_3 \, | \leq 10^{-5}$ist. Tatsächlich gilt $| \, \sqrt{2} - x_3 \, | \approx 2.1 \cdot 10^{-6}$

Um auch den Fall möglicher Berührungspunkten mit der x-Achse oder schlecht konditionierte Probleme abzudecken, empfiehlt es sich sich, in einem Programm zusätzliche Abbruchkriterien einzubauen, da ansonsten die Iteration vielleicht in eine Endlos-Schleife mündet. Einfachstes Mittel, ist eine Obergrenze N_{max} für die Anzahl Interationsschritte anzugeben. Notwendige (aber nicht hinreichende) Kriterien, um eine Nullstelle zu erkennen, sind für ein vorgegebenes $\epsilon > 0$ beispielsweise, dass der Funktionswert nach der n-ten Iteration kleiner wird als ϵ, also $| \, f(x_n) \, | < \epsilon$, oder auch, dass die Differenz zwischen zwei aufeinanderfolgenden Werten unterhalb eine vorgegebene Schwelle sinkt, also $| \, x_{n+1} - x_n \, | < \epsilon$. Diese Abbruchkriterien liefern aber keine Garantie, dass wir tatsächlich nahe genug an einer Nullstelle daran sind.

Aufgaben 3.6 [1]:

2.5 Bestimmen Sie alle Lösungen der Gleichung $2\sin x = x$ bis auf einen nachgewiesenen absoluten Fehler von max. 10^{-3}.

2.6 Das Bauer-Ziege-Wiese-Problem: Ein Bauer besitzt eine kreisrunde Wiese vom Radius R. Am Rand dieser Wiese bindet er eine Ziege an mit einer Leine der Länge r, und zwar so, dass die Ziege genau die Hälfte der Wiese abgrasen kann (s. Bild 2.4). Wie groß ist r?

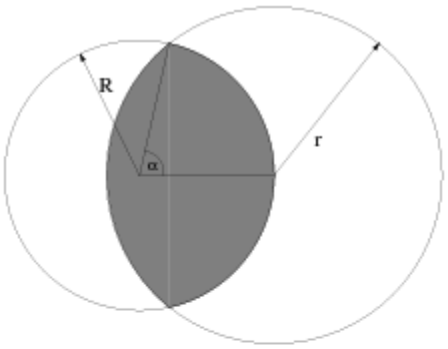


Bild 2.5

Mit dem Kosinussatz erhält man $r = R\sqrt{2\,(1-\cos\alpha)}$. Das Problem führt auf folgende Gleichung für den Winkel α (im Bogenmaß):

$$\frac{\pi}{2\,\cos\alpha} + \alpha - \pi - \tan\alpha = 0.$$

Offensichtlich kann diese Gleichung nicht durch geschicktes Umformen nach α aufgelöst werden. Die Hilfe numerischer Methoden ist daher nötig. Bestimmen Sie ein Intervall, in dem sich die gesuchte Lösung befindet und bestimmen Sie die Lösung mit einem Verfahren Ihrer Wahl bis auf einen gesicherten absoluten Fehler von 0.0001.

2.7 Wenden Sie das Newton-Verfahren, das vereinfachte Newton-Verfahren und das Sekantenverfahren zur näherungsweisen Bestimmung der Nullstelle von $f(x) = x^2 - 2$ an.

Aufgabe 3.7:

- Optional: Implementieren Sie das Sekanten-Verfahren in MATLAB. Wo würden Sie beim Versuch, das Newton-Verfahren zu implementieren, momentan noch auf Schwierigkeiten stossen?

Kapitel 4

Numerische Lösung linearer Gleichungssysteme

In diesem Kapitel behandeln wir die Lösung linearer Gleichungssysteme, die in vielen Anwendungen in der Numerik, Physik, Technik, Betriebswirtschaftslehre etc. auftreten. Beispiele sind das Newton-Verfahren für *nichtlineare* Gleichungssysteme, wo bei jedem Schritt lineare Gleichungssysteme auftreten; die Methode der kleinsten Quadrate von Gauss in der Ausgleichsrechnung; die numerische Lösung von Randwertproblemen bei gewöhnlichen und partiellen Differentialgleichungen mit Hilfe von Differenzenverfahren; bei der Interpolation mittels Splines; die Behandlung von Eigenwertproblemen in der mathematischen Physik; in der Elektrotechnik die Berechnung von Netzwerken (Ströme zu vorgegebenen Spannungen und Widerständen); in der Betriebswirtschaftslehre bei der linearen Programmierung uvm. In der Theorie ist das Problem der Auflösung linearer Gleichungssysteme vollständig gelöst, in der Praxis geht es um deren effiziente Berechnung.

Lernziele:

- Sie können lineare Gleichungssysteme selbst aufstellen.
- Sie können den Gauss-Algorithmus mit und ohne Pivotisierung sowie die $LR-$Zerlegung auf konkrete Problemstellungen anwenden.
- Sie kennen die Cholesky-Zerlegung.
- Sie können die Fehler für gestörte lineare Gleichungssysteme berechnen.
- Sie können das Jacobi- sowie das Gauss-Seidel-Verfahren anwenden und in MATLAB implementieren.
- Sie beherrschen die zugehörigen Fehlerabschätzungen.

4.1 Zur historischen Entwicklung

Auch lineare Gleichungssystem beschäftigten Mathematiker schon vor Tausenden von Jahren. Eine Aufgabe, die rund 4000 Jahre alt ist und aus Mesopotamien stammt, lautet: "Ein Viertel der Breite zur Länge addiert ergibt 7 Handbreiten, Länge und Breite addiert macht 10 Handbreiten". Natürlich beschäftigten sich auch die Ägypter mit ähnlichen Problemen, wie z.B. der folgenden Aufgabe aus dem 'Papyrus Moskau'[1] ca. 2000 v.Chr.: “Berechne die Länge und Breite eines Rechteckes der Fläche 12, wenn die Breite 3/4 der Länge ist”. In der heutigen Schreibweise würden wir das erste Beispiel als System zweier Gleichungen mit zwei Unbekannten formulieren (mit x als Breite

[1] welcher in Moskau aufbewahrt wird, daher der Name

und y als Länge[2]):

$$\begin{aligned} \frac{1}{4}x + y &= 7 \\ x + y &= 10 \end{aligned}$$

bzw. im Matrizenkalkül

$$\begin{pmatrix} \frac{1}{4} & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 7 \\ 10 \end{pmatrix}$$

Die Babylonier oder die Ägypter kannten kein Matrizenkalkül. Die Chinesen kamen dem zwischen 200 bis 100 v.Chr. schon bedeutend näher, wie im chinesischen Mathematikbuch Jiu Zhang Suanshu (dt. 'Neun Kapitel der Rechenkunst' od. 'Neun Bücher arithmetischer Technik') aus dieser Zeit festgehalten ist, welches die chinesische Mathematik und diejenige der umliegenden Länder bis ins 17. Jhr. prägte. So wurde darin bereits das Verfahren beschrieben, welches wir heute als Gauss-Algorithmus kennen[3].

Die erste systematische Untersuchung von linearen Gleichungssystemen wird Gottfried Wilhelm Leibniz (1646-1716) zugeschrieben[4]. Er führte die Formeln für Determinanten für 2x2 und 3x3 Gleichungssysteme ein. Gabriel Cramer (1704-1752) entwickelte die nach ihm benannte allgemeine Lösungsformel für Systeme von n Gleichung mit n Unbekannten. Seine Regel benötigt allerdings einen enormen Rechenaufwand von rund $n(n+1)!$ Gleitkommaoperationen. Für $n = 10$ benötigt man bereits fast 400 Mio. Punktoperationen und für $n = 20$ bereits 10^{21}. Deshalb ist die Cramersche Regel in der Praxis völlig unbrauchbar (dies gilt bereits für $n = 3$).

Der deutsche Mathematiker, Physiker, Astronom und Geodät Carl Friedrich Gauss (1777-1855) betrachtete lineare Gleichungssysteme im Zusammenhang mit astronomischen Problemen. So gelang es ihm, den Zwergplaneten Ceres im Asteroidengürtel zwischen Mars und Jupiter, der 1801 entdeckt und gleich darauf wieder verlorengegangen war, aufgrund seiner Berechnungen basierend auf der Methode der kleinsten Quadrate wieder zu finden. 1811 entwickelte er den nach ihm benannten Gauss-Algorithmus (vgl. Kap. 4.3), eines der heutigen Standardverfahren zur Lösung von linearen Gleichungssystemen. Der Gauss-Algorithmus benötigt für die Lösung eines $n \times n$ Gleichungssystem lediglich $\frac{2}{3}n^3 + \frac{5}{2}n^2 - \frac{13}{6}n$ Punktoperationen (vgl. Kap. 4.6.2), d.h. für $n = 20$ also nur rund 6000 im Gegensatz zu 10^{21} bei der Cramerschen Regel.

Ausgehend von den Untersuchungen linearer Gleichungssysteme entwickelte sich daraus das Gebiet der linearen Algebra, unter anderem basierend auf den Werken von Wiliam Rowan Hamilton (1805-1865; Vektoren, Quaternionen), Hermann Grassmann (1809-1877; endlichdimensonale Vektorräume), Arthur Cayley (1821-1895; Matrizen als algebraische Objekte), Camille Jordan (1838-1922; Jordansche Normalform), Ferdinand Georg Frobenius (1849-1917; Gruppentheorie), Maxime Bôcher (1867-1918; *Introduction to higher algebra*), Herbert Westren Trunbull (1885-1961) und Alexander Aitken (1895-1967) mit *Introduction to the Theory of Canonical Matrices* sowie Leon Mirsky (1918-1983) mit *An introduction to linear algebra.*

Beispiele aus der Praxis mit grossen linearen Gleichungssystemen [6]

1. Parabolantenne der Firma Krupp mit 100 m Durchmesser am oberen Rand (vgl. Abb. 4.1[5],[6]).

   Es handelt sich dabei um einen räumlichen Verbund aus Stäben und Balken, die geometrisch ein Rotationsparaboloid bilden. Die Berechnung muss so erfolgen, dass bei Verformung durch Neigung und Eigengewicht wegen der Richtgenauigkeit der Antenne immer wieder ein Rotationsparaboloid entsteht. Es sind jeweils ca. 5000 Gleichungen mit 5000 Unbekannten zu lösen. Nur der Empfänger muss dann jeweils in den neuen Brennpunkt nachgeführt werden. Für jede neue Einstellung beträgt die mittlere Abweichung vom idealen Paraboloid weniger als 0.6 mm (2012).

[2] Die Bezeichnung unbekannter Grössen durch Buchstaben x, y, z stammt übrigens vom französischen Mathematiker und Philosoph René Descartes (1596-1650)

[3] siehe z.B. MacTutor unter http://www-history.mcs.st-and.ac.uk/HistTopics/Matrices_and_determinants.html

[4] siehe http://www.math.kit.edu/iag2/~globke/media/geschichtela.pdf

[5] By Dr. Schorsch (photo taken by Dr. Schorsch) [GFDL (http://www.gnu.org/copyleft/fdl.html), CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0/) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

[6] „Radiotelescope effelsberg full“ von Hotstepper13 - Eigenes Werk. Lizenziert unter Creative Commons Attribution 3.0 über Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Radiotelescope_effelsberg_full.jpg#mediaviewer/ File:Radiotelescope_effelsberg_full.jpg

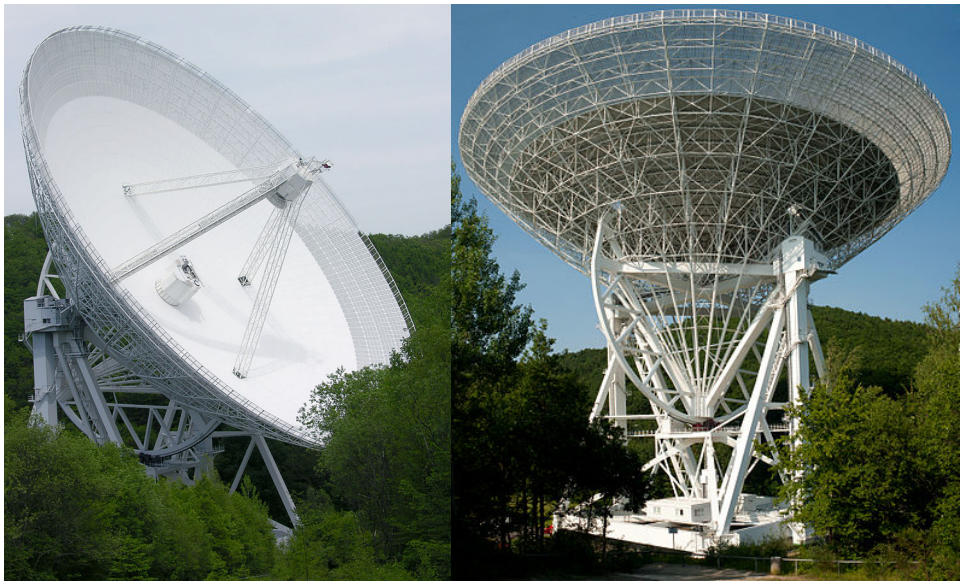


Abbildung 4.1: 100 m Radioteleskop in der Eiffel (Wikimedia Commons).

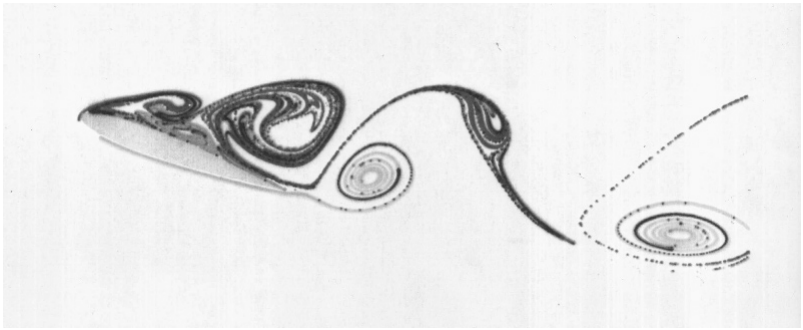


Abbildung 4.2: Simulaton einer ablösenden Strömung [6].

2. Beispiel des Aerodynamischen Instituts der RWTH Aachen. Numerische Simulation einer ablösenden Strömung um Tragflügelprofile, gerechnet mit den Navier-Stokes- Gleichungen: Wenn 3-dimensional gerechnet wird und ein $(31 \times 31 \times 51)$-Gitter mit je 4 Gleichungen verwendet wird, so erhält man nichtlineare Systeme aus 196 044 Gleichungen mit 196 044 Unbekannten, die iterativ (etwa mit 5 Iterationen) gelöst werden. Rechnet man bis zum Wirbelablösen 10 000 Zeitschritte, so ergeben sich $5 \times 10\,000 = 50\,000$ lineare Gleichungssysteme aus rund ca. 200 000 Gleichungen, die zu lösen sind.

3. Ein Finite-Element-Beispiel aus dem Institut für Bildsame Formgebung der RWTH Aachen: Bei der Simulation des Fließpress-Verfahrens zur Herstellung eines Zahnrades mit zwölf Zähnen wird unter Ausnutzung der Symmetriebedingungen mit dem Modell eines halben Zahnes gerechnet. In diesem Beispiel wird dazu ein Netz mit 2911 Knoten erstellt. Man erhält unter Berücksichtigung aller Randbedingungen insgesamt 7560 nichtlineare Gleichungen, die iterativ gelöst werden. Dabei tritt eine Bandmatrix auf.

4. Der PageRank-Algorithmus, welcher auf die Gründer von Google, Sergey Brin und Lawrence Page, zurückgeht[7], erlaubt die Klassifizierung einer Menge von verlinkten Dokumenten, z.B. der Seiten des Internets, nach ihrer "Wichtigkeit", bzw. dem *rank*. Die zugrunde liegende Idee[8] ist, dass eine Seite umso wichtiger ist, je mehr Links von anderen wichtigen Seiten auf sie zeigen. Zur Bestimmung der Wichtigkeit wird die PageRank- bzw. Google-Matrix benötigt. Diese Matrix repräsentiert einen gerichteten Graphen, wobei die Knoten des Graphen den Web-Seiten entsprechen und die Kanten den Links dazwischen.

[7] Sergei Brin, Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Computer Networks and ISDN Systems, Band 30, 1998, S. 107-117

[8] nach UZH, MAS410, Geometrie und Lineare Algebra

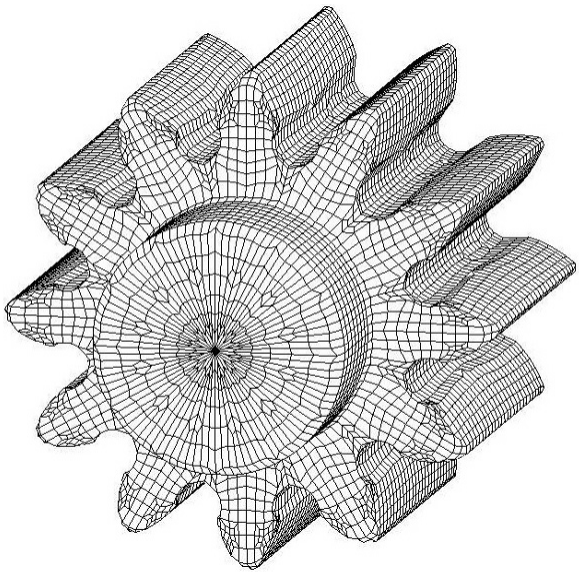


Abbildung 4.3: Fliesspressverfahren zur Herstellung eines Zahnrades [6].

Beispiel: ein einfaches Web mit 4 Seiten ist in Abb. 4.4 dargestellt. Ein Pfeil von Seite i zur Seite j entspricht einem Link. Die Bedeutung der Web-Seiten wird durch den Vektor

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

angegeben, wobei $x_i \in \mathbb{R}$ die Wichtigkeit der Seite i angibt. Für die Seite 1 ergibt sich z.B.

$$x_1 = 0 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + \frac{1}{2} \cdot x_4.$$

Die Gewichte berechnen sich daraus, dass die Seite 1 keinen Link auf sich selber $(0 \cdot x_1)$ oder von Seite 2 $(0 \cdot x_2)$ hat, jedoch je einen Link von Seite 3 und Seite 4. Da Seite 3 insgesamt nur einen ausgehenden Link aufweist, erhält dieser für Seite 1 das volle Gewicht $(1 \cdot x_3)$. Da Seite 4 aber 2 ausgehende Links aufweist, erhält der Link auf Seite 1 nur das Gewicht $\frac{1}{2}$ (also $\frac{1}{2} \cdot x_4$). Für alle vier Seiten erhält man so das lineare Gleichungssystem:

$$\begin{aligned} x_1 &= 0x_1 + 0x_2 + 1x_3 + \frac{1}{2}x_4 \\ x_2 &= \frac{1}{3}x_1 + 0x_2 + 0x_3 + 0x_4 \\ x_3 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 + 0x_3 + \frac{1}{2}x_4 \\ x_4 &= \frac{1}{3}x_1 + \frac{1}{2}x_2 + 0x_3 + 0x_4 \end{aligned}$$

Oder in Matrix-Schreibweise:

$$\boldsymbol{x} = \boldsymbol{P} \cdot \boldsymbol{x}, \quad \text{mit } \boldsymbol{P} = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix}$$

Also ist $\boldsymbol{x}$ ein Eigenvektor von $\boldsymbol{P}$ zum Eigenwert 1. Dies ist zudem eine Fixpunktgleichung und kann gemäss Kap. 3.4 iterativ gelöst werden. Mit dem Startvektor $\boldsymbol{x_0} = (1, 1, 1, 1)^T$ erhalten wir mittels der Fixpunktiteration

$$\boldsymbol{x}_{i+1} = \boldsymbol{P}\boldsymbol{x}_i$$

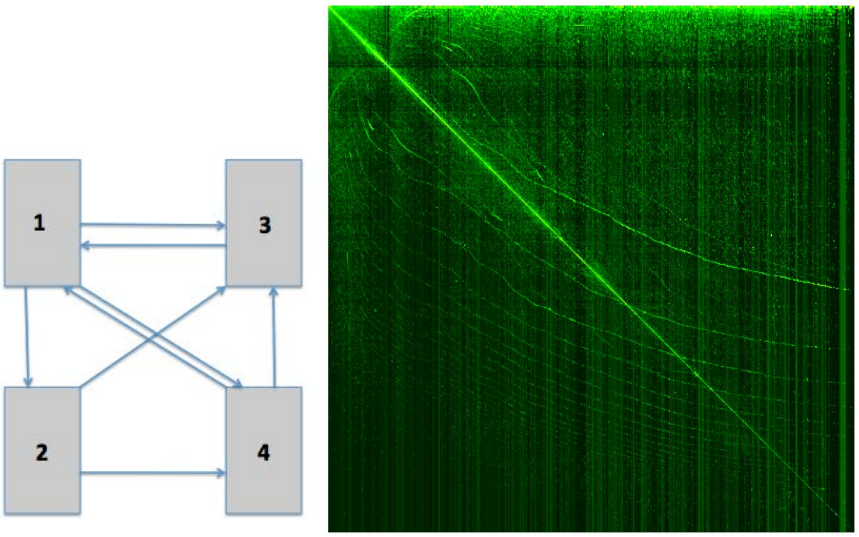


Abbildung 4.4: Links: Einfaches Web mit $n = 4$ Seiten und seinen Links[8] . Rechts: Google Matrix des Netzwerks der Cambridge Universtität aus dem Jahr 2006 mit $n = 212710$ (http://arxiv.org/abs/1106.6215, GFDL, Wikimedia Commons)

die Näherungslösung

$$\boldsymbol{x}_1 = \boldsymbol{P}\boldsymbol{x}_0 = \begin{pmatrix} 1.5000 \\ 0.3333 \\ 1.3333 \\ 0.8333 \end{pmatrix}, \ \boldsymbol{x}_2 = \boldsymbol{P}\boldsymbol{x}_1 = \begin{pmatrix} 1.75 \\ 0.5 \\ 1.0833 \\ 0.6667 \end{pmatrix}, \ \ldots, \boldsymbol{x}_{16} = \boldsymbol{P}\boldsymbol{x}_{15} = \begin{pmatrix} 1.5484 \\ 0.5161 \\ 1.1613 \\ 0.7742 \end{pmatrix}$$

Also hat die Seite 1 die höchste Wichtigkeit, Seite 3 die zweithöchste, Seite 4 die dritthöchste, und Seite 2 die vierhöchste bzw. die niedrigste Wichtigkeit. Unter Verwendung der Einheitsmatrix $\boldsymbol{I}$ lässt sich $\boldsymbol{x}$ auch mit dem aus der linearen Algebra bereits bekannten und in Kap. 4.3 nochmals detailliert beschriebenen Gauss-Algorithmus als eine Lösung der homogenen Gleichung

$$(\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$$

bestimmen.

Um auch zufälliges Hüpfen zwischen den Seiten (ohne Benützung von Links) abbilden zu können, wird die Matrix $\boldsymbol{P}$ noch modifiziert mit einer Matrix $\boldsymbol{S}$, deren Elemente alle den Wert $\frac{1}{n}$ haben bei einem Web mit n Seiten. Die **Google-Matrix** $\boldsymbol{G}$ erhält man als Überlagerung der beiden Matrizen:

$$\boldsymbol{G} = \alpha\boldsymbol{P} + (1 - \alpha)\boldsymbol{S}.$$

Dabei ist $0 \leq \alpha \leq 1$ ein Faktor, der das zufällige Hüpfen modelliert (für $\alpha = 1$ findet kein zufälliges Hüpfen statt, für $\alpha = 0$ findet ausschliesslich zufälliges Hüpfen statt). Die Erfinder des PageRank-Algorithmus wählten $\alpha = 0.85$.

4.2 Problemstellung

Gesucht ist eine Lösung zu einem linearen Gleichungssystem mit$n$ Gleichungen und n Unbekannten der Form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n \end{aligned} \tag{4.1}$$

Üblicherweise schreibt man solche Gleichungssysteme in Matrix-Form als

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{4.2}$$

mit

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n},\ \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \text{ und } \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n.$$

Bezüglich der Notation werden in diesem Skript Matrizen mit fettgedruckten Grussbuchstaben und Vektoren mit fettgedruckten Kleinbuchstaben hervorgehoben. Im obigen Gleichungssystem sind $\boldsymbol{A}$ und $\boldsymbol{b}$ gegeben, $\boldsymbol{x}$ ist gesucht. Gewisse Eigenschaften der Matrix $\boldsymbol{A} = (a_{ij})$ entscheiden darüber, was für ein Verfahren zur Lösung des Gleichungssystems (4.2) sinnvoll eingesetzt werden kann. Da die Anzahl n der Gleichungen in (4.1) der Anzahl Unbekannten $x_1, \ldots, x_n$ entspricht, ist $\boldsymbol{A}$ in (4.2) eine quadratische Matrix der Dimension $n \times n$.

Für quadratische Matrizen $\boldsymbol{A}$ wissen wir aus der linearen Algebra, dass genau dann eine eindeutige Lösung existiert, wenn die Determinante $det(\boldsymbol{A})$ nicht verschwindet (gleichbedeutend mit $\boldsymbol{A}$ ist invertierbar bzw. $\boldsymbol{A}$ ist regulär), d.h. wenn eine Matrix $\boldsymbol{A}^{-1}$ existiert, so dass $\boldsymbol{A} \cdot \boldsymbol{A}^{-1} = \boldsymbol{A}^{-1} \cdot \boldsymbol{A} = \boldsymbol{I}$, wobei $\boldsymbol{I}$ die $n \times n$ Einheitsmatrix ist (die Einträge auf der Diagonalen sind 1, alle anderen Einträge sind 0).

Bei der numerischen Lösung von Systemen der Art (4.2) unterscheidet man zwischen

- direkten Verfahren

  Mit einem direkten Verfahren erhält man mit einer endlichen Zahl von Rechenschritten die exakte Lösung (wenn man Rundungsfehler vernachlässigt)

- iterativen Verfahren

  Hier wird eine Folge von Vektoren erzeugt, die gegen die Lösung von (4.2) konvergiert.

Wir beginnen mit den direkten Verfahren. Hierfür benötigen wir die Definition der oberen bzw. unteren Dreiecksform.

Definition 4.1: Untere Dreiecksmatrix / Obere Dreiecksmatrix [6]

- Eine $n \times n$ Matrix $\boldsymbol{L} = (l_{ij})$ heisst **untere Dreiecksmatrix**, wenn $l_{ij} = 0$ für $j > i$ gilt; sie heisst **normierte untere Dreiecksmatrix**, wenn ausserdem $l_{ii} = 1$ für alle i gilt.
- Eine $n \times n$ Matrix $\boldsymbol{R} = (r_{ij})$ heisst **obere Dreiecksmatrix**, wenn $r_{ij} = 0$ für $i > j$ gilt; sie heisst **normierte obere Dreiecksmatrix**, wenn ausserdem $r_{ii} = 1$ für alle i gilt.

Beispiel 4.1

- Untere normierte Dreiecksmatrix:

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & 1 \end{pmatrix}$$

- Obere Dreiecksmatrix:

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & r_{nn} \end{pmatrix}$$

4.3 Der Gauss-Algorithmus

Das Eliminationsverfahren nach Gauss (der "Gauss-Algorithmus") ist ein anschauliches Verfahren, das zudem gut implementiert werden kann. Es beruht auf der Tatsache, dass ein lineares Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ leicht lösbar ist, falls die Matrix $\boldsymbol{A}$ als obere Dreiecksmatrix vorliegt, also alle Elemente unterhalb der Diagonalen verschwinden.

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix} \tag{4.3}$$

In diesem Fall kann man für $Ax = b$ mittels der folgenden rekursiven Vorschrift, dem sogenannten Rückwärtseinsetzen, die Komponenten von x berechnen:

$$x_n = \frac{b_n}{a_{nn}},\ x_{n-1} = \frac{b_{n-1} - a_{n-1\,n}\,x_n}{a_{n-1\,n-1}},\ \ldots\ ,\ x_1 = \frac{b_1 - a_{12}x_2 - \ldots - a_{1n}x_n}{a_{11}} \tag{4.4}$$

oder, kompakt geschrieben,

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}, \qquad i = n, n-1, ..., 1. \tag{4.5}$$

Die Idee des Gauss-Algorithmus ist nun, ein beliebiges Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ umzuformen in ein äquivalentes Gleichungssystem $\widetilde{\boldsymbol{A}}\boldsymbol{x} = \widetilde{\boldsymbol{b}}$, so dass die Matrix $\widetilde{\boldsymbol{A}}$ als obere Dreiecksmatrix vorliegt.

Bei dieser Transformation sind folgende Umformungen zugelassen:

- $z_j := z_j - \lambda z_i$ mit $i < j$ ($\lambda \in \mathbb{R}$), wobei z_i die i-te Zeile des Gleichungssystems bezeichnet
- $z_i \to z_j$: Vertauschen der i-ten und j-ten Zeile im System

Es sind also nur Zeilenvertauschungen und die Subtraktion eines Vielfachens einer Zeile von einer darunter stehenden Zeile erlaubt. Mit diesen beiden Umformungen kann jede Matrix $\boldsymbol{A}$ in die obere Dreiecksform (4.3) gebracht werden (natürlich müssen für die Lösung von$\boldsymbol{Ax} = \boldsymbol{b}$ die Umformungen auch auf $\boldsymbol{b}$ angewandt werden).

Beispiel 4.2

- Es soll folgendes System $\boldsymbol{Ax} = \boldsymbol{b}$ gelöst werden, wobei

$$\boldsymbol{A} = \begin{pmatrix} 1 & 5 & 6 \\ 7 & 9 & 6 \\ 2 & 3 & 4 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 29 \\ 43 \\ 20 \end{pmatrix}$$

Um die Matrix auf die obere Dreiecksgestalt zu bringen, müssen wir die Einträge 7, 3, und 2 unterhalb der Diagonalen eliminieren. Wir subtrahieren das 7-fache der erste Zeile von der zweiten Zeile ($z_2 \equiv z_2 - 7z_1$) und erhalten

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 2 & 3 & 4 \end{pmatrix} \text{ und } \boldsymbol{b}_1 = \begin{pmatrix} 29 \\ -160 \\ 20 \end{pmatrix}.$$

Jetzt subtrahieren 2-mal die erste Zeile von der letzten ($z_3 \equiv z_3 - 2z_1$) und erhalten

$$\boldsymbol{A}_2 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 0 & -7 & -8 \end{pmatrix} \text{ und } \boldsymbol{b}_2 = \begin{pmatrix} 29 \\ -160 \\ -38 \end{pmatrix}.$$

Im letzten Schritt subtrahieren wir 7/26-mal die zweite Zeile von der dritten ($z_3 \equiv z_3 - \frac{7}{26}z_2$):

$$\boldsymbol{A}_3 = \begin{pmatrix} 1 & 5 & 6 \\ 0 & -26 & -36 \\ 0 & 0 & \frac{22}{13} \end{pmatrix} \text{ und } \boldsymbol{b}_3 = \begin{pmatrix} 29 \\ -160 \\ \frac{66}{13} \end{pmatrix}.$$

Somit erhalten wir über Rückwärtseinsetzen gemäss (4.4) die gesuchten Komponenten von x:

$$x_3 = \frac{\frac{66}{13}}{\frac{22}{13}} = 3,\ x_2 = \frac{-160 - 3 \cdot (-36)}{-26} = 2 \text{ und } x_1 = \frac{29 - 2 \cdot 5 - 3 \cdot 6}{1} = 1.$$

Für die Programmierung des Gauss-Algorithmus sollte nun folgendermassen vorgegangen werden:

- Zuerst erzeugt man Nullen in der ersten Spalte unterhalb von a_{11} mit der Operation $z_j := z_j - \frac{a_{j1}}{a_{11}} z_i$ mit $j = 2, \ldots, n$.

  Dies geht nur, falls $a_{11} \neq 0$. Ist $a_{11} = 0$, so vertauschen wir die erste Zeile mit der i-ten Zeile, wobei $a_{i1} \neq 0$ sein muss. Falls alle Zeilen der Matrix in der ersten Zeile eine Null besitzen funktioniert die Vertauschung nicht. Dann ist allerdings auch die Matrix nicht regulär und die Lösungsmenge kann leer sein oder auch unendlich viele Elemente enthalten.

- Dieser Schritt wird nun wiederholt, in dem man mit der zweiten Spalte fortfährt und unterhalb der Diagonalen Nullen erzeugt.

Schliesslich erhält man den

Gauss-Algorithmus zur Transformation von $\boldsymbol{Ax} = \boldsymbol{b}$ auf ein oberes Dreiecksystem [1]:

- für $i = 1, ..., n-1$:

  erzeuge Nullen unterhalb des Diagonalelementes in der i-ten Spalte

  - Falls nötig und möglich, sorge durch Zeilenvertauschung für $a_{ii} \neq 0$:
    falls $a_{ii} \neq 0$: tue nichts
    falls $a_{ii} = 0$:
    $$\begin{cases} \text{falls } a_{ji} = 0 \text{ für alle } j = i+1, ..., n: \\ \quad A \text{ ist nicht regulär; stop;} \\ \text{wenn } a_{ji} \neq 0 \text{ für ein } j = i+1, ..., n: \\ \quad \text{sei } j \geq i+1 \text{ der kleinste Index mit } a_{ji} \neq 0 \\ \quad z_i \longleftrightarrow z_j \end{cases}$$
  - Eliminationsschritt:
    für $j = i+1, ..., n$ eliminiere das Element a_{ji} durch:
    $$z_j := z_j - \frac{a_{ji}}{a_{ii}} \cdot z_i$$

Aufgabe 4.1 [1]

- Bringen Sie das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ auf die obere Dreiecksform und lösen Sie nach x auf, wobei
$$\boldsymbol{A} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -3 & -2 \\ 5 & 1 & 4 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 0 \\ 5 \\ 3 \end{pmatrix}$$

Lösung:

$$i = 1,\ j = 2 \Rightarrow z_2 \equiv z_2 - \tfrac{1}{(-1)} z_1 \Rightarrow (\boldsymbol{A}_1 \mid \boldsymbol{b}_1) = \left(\begin{array}{ccc|c} -1 & 1 & 1 & 0 \\ 0 & -2 & -1 & 5 \\ 5 & 1 & 4 & 3 \end{array}\right)$$

$$i = 1,\ j = 3 \Rightarrow z_3 \equiv z_3 - \tfrac{5}{(-1)} z_1 \Rightarrow (\boldsymbol{A}_2 \mid \boldsymbol{b}_2) = \left(\begin{array}{ccc|c} -1 & 1 & 1 & 0 \\ 0 & -2 & -1 & 5 \\ 0 & 6 & 9 & 3 \end{array}\right)$$

$$i = 2,\ j = 3 \Rightarrow z_3 \equiv z_3 - \tfrac{6}{(-2)} z_2 \Rightarrow (\boldsymbol{A}_3 \mid \boldsymbol{b}_3) = \left(\begin{array}{ccc|c} -1 & 1 & 1 & 0 \\ 0 & -2 & -1 & 5 \\ 0 & 0 & 6 & 18 \end{array}\right)$$

Rückeinsetzen gemäss (4.4) liefert:

$x_3 = \frac{18}{6} = 3,\ x_2 = \frac{5-(-1)\cdot 3}{(-2)} = -4,\ x_1 = \frac{0-1\cdot(-4)-1\cdot 3}{(-1)} = -1$

Natürlich muss man, wenn man $\boldsymbol{Ax} = \boldsymbol{b}$ schon mal gelöst hat und nun die Lösung von $\boldsymbol{Ax} = \boldsymbol{c}$ für einen neuen Vektor $\boldsymbol{c}$ bestimmen will, die Matrix $\boldsymbol{A}$ nicht nochmal auf die obere Dreiecksform bringen, sondern wendet die Zeilenumformungen einfach auf den neuen Vektor $\boldsymbol{c}$ an mit anschliessendem Rückwärtseinsetzen.

Beispiel 4.3 [1]

- Es soll das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{c}$ mit der Matrix $\boldsymbol{A}$ aus der vorherigen Aufgabe gelöst werden für $\boldsymbol{c} = (13, -32, 22)^T$

  Lösung:

  $z_2 \equiv z_2 - \frac{1}{(-1)} z_1 \Rightarrow \boldsymbol{c}_1 = \begin{pmatrix} 13 \\ -19 \\ 22 \end{pmatrix}$

  $z_3 \equiv z_3 - \frac{5}{(-1)} z_1 \Rightarrow \boldsymbol{c}_2 = \begin{pmatrix} 13 \\ -19 \\ 87 \end{pmatrix}$

  $z_3 \equiv z_3 - \frac{6}{(-2)} z_2 \Rightarrow \boldsymbol{c}_3 = \begin{pmatrix} 13 \\ -19 \\ 30 \end{pmatrix}$

  Rückeinsetzen liefert die Lösung $\boldsymbol{x} = (-1, 7, 5)^T$

Eine zusätzliche Anwendung des Gauss-Algorithmus ist die Determinantenbestimmung. Wenn wir mit $\widetilde{\boldsymbol{A}}$ die obere Dreiecksmatrix von $\boldsymbol{A}$ bezeichnen, dann gilt die Beziehung

$$det(\boldsymbol{A}) = (-1)^l \cdot det(\widetilde{\boldsymbol{A}}) = (-1)^l \prod_{i=1}^{n} \widetilde{a}_{ii}$$

wobei $\widetilde{a}_{ii}$ die Diagonalelemente von $\widetilde{\boldsymbol{A}}$ sind und l die Anzahl der im Laufe des Gauss-Algorithmus vorgenommen Zeilenvertauschungen.

Aufgabe 4.2 [1]

- Bestimmen Sie die Determinante der Matrix $\boldsymbol{A}$ aus Aufgabe 4.1 mittels des Gauss-Algorithmus.

  Lösung: die obere Dreiecksform $\widetilde{\boldsymbol{A}}$ von $\boldsymbol{A}$ haben wir bereits gerechnet:

  $$\widetilde{\boldsymbol{A}} = \boldsymbol{A}_3 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix}$$

  Dabei wurde keine Zeilenvertauschung durchgeführt, d.h. die Determinante von A ist das Produkt der Diagonalelemente von $\widetilde{A}$:

  $$det(\boldsymbol{A}) = (-1) \cdot (-2) \cdot 6 = 12$$

Aufgabe 4.3 [1]

- Optional: Implementieren Sie den Gauss-Algorithmus in MATLAB und bestimmen Sie damit die Lösungen für die untenstehenden Gleichungssysteme sowie die Determinanten der Matrizen $\boldsymbol{A}_1 - \boldsymbol{A}_4$. Wer die Aufgabe lieber von Hand löst, kann dies unter Angabe aller Zwischenschritte tun.

$$\boldsymbol{A}_1\boldsymbol{x} = \begin{pmatrix} 4 & -1 & -5 \\ -12 & 4 & 17 \\ 32 & -10 & -41 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} -5 \\ 19 \\ -39 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 6 \\ -12 \\ 48 \end{pmatrix}$$

$$\boldsymbol{A}_2\boldsymbol{x} = \begin{pmatrix} 2 & 7 & 3 \\ -4 & -10 & 0 \\ 12 & 34 & 9 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 25 \\ -24 \\ 107 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 5 \\ -22 \\ 42 \end{pmatrix}$$

$$\boldsymbol{A}_3\boldsymbol{x} = \begin{pmatrix} -2 & 5 & 4 \\ -14 & 38 & 22 \\ 6 & -9 & -27 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 1 \\ 40 \\ 75 \end{pmatrix} \text{ bzw. } = \begin{pmatrix} 16 \\ 82 \\ -120 \end{pmatrix}$$

$$\boldsymbol{A}_4\boldsymbol{x} = \begin{pmatrix} -1 & 2 & 3 & 2 & 5 & 4 & 3 & -1 \\ 3 & 4 & 2 & 1 & 0 & 2 & 3 & 8 \\ 2 & 7 & 5 & -1 & 2 & 1 & 3 & 5 \\ 3 & 1 & 2 & 6 & -3 & 7 & 2 & -2 \\ 5 & 2 & 0 & 8 & 7 & 6 & 1 & 3 \\ -1 & 3 & 2 & 3 & 5 & 3 & 1 & 4 \\ 8 & 7 & 3 & 6 & 4 & 9 & 7 & 9 \\ -3 & 14 & -2 & 1 & 0 & -2 & 10 & 5 \end{pmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} -11 \\ 103 \\ 53 \\ -20 \\ 95 \\ 78 \\ 131 \\ -26 \end{pmatrix}$$

4.4 Fehlerfortpflanzung beim Gauss-Algorithmus und Pivotisierung

Im vorherigen Abschnitt haben wir Zeilen nur vertauscht, falls ein Diagonalelement im Laufe der Berechnungen Null wurde. Man kann aber Zeilenvertauschungen aber auch dazu verwenden, um Fehler z.B. durch Gleitpunktoperationen, zu minimieren.

In jedem Eliminationsschritt wurden die Zeilen bisher mit $\lambda = \frac{a_{ji}}{a_{ii}}$ multipliziert, d.h. der absolute Fehler vergrösserte sich um den Faktor $\mid \lambda \mid$(siehe Kap. 2). Wünschenswert wäre es also, wenn $\mid \lambda \mid = \mid \frac{a_{ji}}{a_{ii}} \mid < 1$. Dies lässt sich einfach dadurch erreichen, dass man vor dem Eliminationsschritt überprüft, welches Element in der Spalte betragsmässig am grössten ist und die Zeile vertauscht, so dass dieses grösste Element zum Diagonalelement wird. Dieses Vorgehen wird Spaltenpivotisierung genannt.

Gauss-Algorithmus zur Transformation von $\boldsymbol{Ax} = \boldsymbol{b}$ mit Spaltenpivotisierung [1]:

- für $i = 1, ..., n-1$:

  erzeuge Nullen unterhalb des Diagonalelementes in der i-ten Spalte

  - Suche das betragsgrösste Element unterhalb der Diagonalen in der i-ten Spalte:
    Wähle k so, dass $|a_{ki}| = \max\{|a_{ji}| \mid j = i, ..., n\}$
    $\begin{cases} \text{falls } a_{ki} = 0: & A \text{ ist nicht regulär; stop;} \\ \text{falls } a_{ki} \neq 0: & z_k \longleftrightarrow z_i; \end{cases}$
  - Eliminationsschritt:
    für $j = i+1, ..., n$ eliminiere das Element a_{ji} durch:

$$z_j := z_j - \frac{a_{ji}}{a_{ii}} \cdot z_i$$

Beispiel 4.4 [1]

- Die Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix}$$

soll mittels Spaltenpivotisierung auf die (rechts-) obere Dreiecksform gebracht werden.

Lösung:

$$\boldsymbol{A} := \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix} \overset{z_1 \longleftrightarrow z_2}{\longrightarrow} \begin{pmatrix} 4 & -2 & 6 \\ 1 & 2 & -1 \\ 3 & 1 & 0 \end{pmatrix} \overset{z_2 := z_2 - 0.25\, z_1}{\longrightarrow} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 3 & 1 & 0 \end{pmatrix}$$
$$\overset{z_3 := z_3 - 0.75\, z_1}{\longrightarrow} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 0 & 2.5 & -4.5 \end{pmatrix} \overset{z_3 := z_3 - z_2}{\longrightarrow} \begin{pmatrix} 4 & -2 & 6 \\ 0 & 2.5 & -2.5 \\ 0 & 0 & -2.5 \end{pmatrix}$$

Bermerkung: in dieser Lösung aus [1] hat es einen Fehler ... wo?

4.5 Dreieckszerlegung von Matrizen

4.5.1 Die LR-Zerlegung

In der obigen Version des Gauß–Verfahrens haben wir die Matrix $\boldsymbol{A}$ auf obere Dreiecksform gebracht und zugleich alle dafür notwendigen Operationen auch auf den Vektor $\boldsymbol{b}$ angewendet. Es gibt alternative Möglichkeiten, lineare Gleichungssysteme zu lösen, bei denen der Vektor $\boldsymbol{b}$ unverändert bleibt. Wir werden nun ein Verfahren kennen lernen, bei dem die Matrix $\boldsymbol{A}$ in ein Produkt von zwei Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ zerlegt wird, also $\boldsymbol{A} = \boldsymbol{LR}$, wobei $\boldsymbol{R}$ eine obere Dreiecksmatrix und $\boldsymbol{L}$ eine untere normierte Dreieckmatrix ist:

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & 1 \end{pmatrix}, \boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & r_{nn} \end{pmatrix} \tag{4.6}$$

Die Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$ wird als $\boldsymbol{LR}$**–Faktorisierung** oder $\boldsymbol{LR}$**–Zerlegung** bezeichnet. Das ursprüngliche Gleichungssystem

$$\boldsymbol{Ax} = \boldsymbol{b}$$

lautet dann

$$\boldsymbol{LRx} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{Ly} = \boldsymbol{b} \textit{ und } \boldsymbol{Rx} = \boldsymbol{y}$$

und lässt sich wie folgt in zwei Schritten lösen:

1. Zunächst löst man das Gleichungssystem $\boldsymbol{Ly} = \boldsymbol{b}$. Dies kann, ganz analog zum Rückwärtseinsetzen (4.4) durch Vorwärtseinsetzen geschehen:
$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}}, , \qquad i = 1, 2, ..., n. \tag{4.7}$$
2. Anschliessend löst man durch Rückwärtseinsetzen das Gleichungssystem $\boldsymbol{Rx} = \boldsymbol{y}$. Dann gilt
$$\boldsymbol{Ax} = \boldsymbol{LRx} = \boldsymbol{Ly} = \boldsymbol{b} \tag{4.8}$$
womit das System $\boldsymbol{Ax} = \boldsymbol{b}$ gelöst ist.

Wir haben beim Gauss-Algorithmus bereits gesehen, dass sich eine beliebige Matrix durch Zeilenumformungen in eine obere Dreiecksform transformieren lässt. Im Folgenden gehen wir davon aus, dass Zeilenvertauschungen nicht notwendig sind. Der Gauss-Algorithmus läßt sich dann so erweitern, dass damit eine $\boldsymbol{LR}$–Zerlegung einer invertierbaren Matrix $\boldsymbol{A}$ möglich ist. Tatsächlich gilt:

- $\boldsymbol{R}$ ist gerade die durch den Gauss-Algorithmus auf die obere Dreiecksform gebrachte Matrix $\widetilde{\boldsymbol{A}}$
- Die Elemente l_{ji} von $\boldsymbol{L}$ entsprechen gerade den berechneten Faktoren λ aus den Eliminationsschritten $z_j := z_j - \lambda_{ji} z_i$, also $l_{ji} = \lambda_{ji}$

Beispiel 4.5

- Wir berechnen für die Matrix $\boldsymbol{A}$ aus Aufgabe 4.1 die normierte untere Dreiecksmatrix $\boldsymbol{L}$ und die obere Dreiecksmatrix $\boldsymbol{R}$, so dass $\boldsymbol{A} = \boldsymbol{LR}$.

  Lösung: Wir hatten

$$\boldsymbol{A} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -3 & -2 \\ 5 & 1 & 4 \end{pmatrix}$$

  und

$$i=1,\ j=2 \Rightarrow z_2 \equiv z_2 - \underbrace{\frac{1}{(-1)}}_{l_{21}} z_1 \Rightarrow \boldsymbol{A}_1 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 5 & 1 & 4 \end{pmatrix}$$

$$i=1,\ j=3 \Rightarrow z_3 \equiv z_3 - \underbrace{\frac{5}{(-1)}}_{l_{31}} z_1 \Rightarrow \boldsymbol{A}_2 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 6 & 9 \end{pmatrix}$$

$$i=2,\ j=3 \Rightarrow z_3 \equiv z_3 - \underbrace{\frac{6}{(-2)}}_{l_{32}} z_2 \Rightarrow \boldsymbol{A}_3 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix} = \boldsymbol{R}$$

  Das heisst, wir können

$$\boldsymbol{R} = \boldsymbol{A}_3 = \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix}$$

  setzen und für die Elemente von L erhalten wir aus den 3 Eliminationsschritten die drei Elemente

$$\begin{aligned} l_{21} &= \frac{1}{(-1)} = -1 \\ l_{31} &= \frac{5}{(-1)} = -5 \\ l_{32} &= \frac{6}{(-2)} = -3 \end{aligned}$$

  und damit

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -5 & -3 & 1 \end{pmatrix}$$

  Die Probe ergibt wie gewünscht

$$\boldsymbol{LR} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -5 & -3 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 6 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -3 & -2 \\ 5 & 1 & 4 \end{pmatrix} = \boldsymbol{A}$$

Es gilt der folgende

Satz 4.1: $\boldsymbol{LR}$-Zerlegung [1]
Zu jeder regulären $n \times n$ Matrix $\boldsymbol{A}$, für die der Gauss-Algorithmus ohne Zeilenvertauschung durchführbar ist, gibt es $n \times n$ Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ mit den folgenden Eigenschaften:

- $\boldsymbol{L}$ ist eine normierte untere Dreiecksmatrix (also mit $l_{ii} = 1$ für $i = 1, ..., n$)
- $\boldsymbol{R}$ ist eine obere Dreiecksmatrix mit $r_{ii} \neq 0$ für $i = 1, ..., n$
- $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{R}$ ist die **$\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$**.

Aufwand: Die Berechnung der $\boldsymbol{LR}$-Zerlegung mit dem Gauss-Algorithmus benötigt $\frac{1}{3}(n^3 - n)$ Punktoperationen

Bemerkungen:

1. Die direkte Lösung von $\boldsymbol{Ax} = \boldsymbol{b}$ durch die Berechnung der inversen $\boldsymbol{A}^{-1}$ ist nicht praktikabel, da dies die Lösung von n linearen Gleichungssystemen erfordern würde und damit erheblich aufwendiger wäre.

2. Ein mit der $\boldsymbol{LR}$-Zerlegung (in Englisch $\boldsymbol{LU}$-decomposition) verwandter Algorithmus wird auch teilweise angewendet als Benchmark für die Rechengeschwindigkeit. Aus Wikipedia: " LU reduction is an algorithm related to LU decomposition. This term is usually used in the context of super computing and highly parallel computing. In this context it is used as a benchmarking algorithm, i.e. to provide a comparative measurement of speed for different computers. LU reduction is a special parallelized version of an LU decomposition algorithm, an example can be found in (Guitart 2001). The parallelized version usually distributes the work for a matrix row to a single processor and synchronizes the result with the whole matrix (Escribano 2000)".

3. Unter anderem ist die $\boldsymbol{LR}$-Zerlegung eine geschickte Variante, die Zwischenresultate des Gauss-Algorithmus zu speichern.

Aufgabe 4.4 [1]:

- Finden Sie für die Matrix A des linearen Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & -1 \\ 4 & -2 & 6 \\ 3 & 1 & 0 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 9 \\ -4 \\ 9 \end{pmatrix}$$

  die $\boldsymbol{LR}$-Zerlegung. Benutzen Sie dafür die folgenden Operationen der Gauss-Elimination:

$$(\boldsymbol{A} \mid \boldsymbol{b}) = \left(\begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 4 & -2 & 6 & -4 \\ 3 & 1 & 0 & 9 \end{array}\right) \xrightarrow{z_2 := z_2 - 4z_1} \left(\begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 0 & -10 & 10 & -40 \\ 3 & 1 & 0 & 9 \end{array}\right)$$

$$\xrightarrow{z_3 := z_3 - 3z_1} \left(\begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 0 & -10 & 10 & -40 \\ 0 & -5 & 3 & -18 \end{array}\right) \xrightarrow{z_3 := z_3 - 0.5z_2} \left(\begin{array}{ccc|c} 1 & 2 & -1 & 9 \\ 0 & -10 & 10 & -40 \\ 0 & 0 & -2 & 2 \end{array}\right)$$

  Berechnen Sie die Lösung $\boldsymbol{x}$ zuerst mittels Rückwärtseinsetzen direkt aus der obigen Dreiecksform und dem $\boldsymbol{b}$ Vektor. Lösen Sie anschliessend die beiden linearen Systeme $\boldsymbol{Ly} = \boldsymbol{b}$ und $\boldsymbol{Rx} = \boldsymbol{y}$ und vergleichen Sie.

- Optional: Erweitern Sie ihr unter Aufgabe 4.3 erstelltes Programm zum Gauss-Algorithmus, so dass es gleichzeitig auch die $\boldsymbol{LR}$- Zerlegung von $\boldsymbol{A}$ berechnet. Berechnen Sie damit die $\boldsymbol{LR}$-Zerlegung für die Matrixen aus Aufgabe 4.3.

4.5.1.1 Die $\boldsymbol{LR}$-Zerlegung mit Zeilenvertauschung

Sind Zeilenvertauschungen nötig, so lässt sich in der Regel keine $\boldsymbol{LR}$–Zerlegung erhalten. Allerdings lässt sich die Vertauschung der$i$-ten und j-ten Zeile in $\boldsymbol{A}$ durch eine Multiplikation von links der Form

$$\boldsymbol{P}_k = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & 1 & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & & 0 & \\ & & & & & & & & 1 \\ & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix}$$

erreichen, wobei die 0 gerade an der i-ten und j-ten Stelle auf der Diagonale steht (also $p_{ii} = p_{jj} = 0$). Die 1 erscheint dann in der i-ten Zeile und j-ten Spalte sowie der j-ten Zeile und i-ten Spalte ($p_{ij} = p_{ji} = 1$). Der ganzahlige Index $k = 1, 2, ...$ dient hier nur dazu, mehrere solcher Matrizen voneinander unterscheiden zu können, denn bei mehreren Zeilenvertauschungen können die dafür benötigten Matrizen $\boldsymbol{P}_1, ..., \boldsymbol{P}_l$ zu einer einzigen Matrix $\boldsymbol{P} = \boldsymbol{P}_l \cdot ... \cdot \boldsymbol{P}_1$ aufmultipliziert werden (bei linksseitiger Multiplikation). Die Matrix $\boldsymbol{P}$ nennt sich die Permutationsmatrix, sie ist immer regulär und es gilt $\boldsymbol{P}^{-1} = \boldsymbol{P}$.

Beispiel 4.6.1

- Die Vertauschung der 2. und 4. Zeile bei der Matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \text{ führt zu } \boldsymbol{A}^* = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix},$$

  welches sich auch durch die Multiplikation von links

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

  ausdrücken lässt, also $\boldsymbol{P}_1 \cdot \boldsymbol{A} = \boldsymbol{A}^*$.

- Die zusätzliche Vertauschung der 1. und 3. Zeile, also

$$\boldsymbol{A}^* = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix} \text{ geht über in } \boldsymbol{A}^{**} = \begin{pmatrix} 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix},$$

  lässt sich darstellen durch die Multiplikation von links mit $\boldsymbol{P}_2 \cdot \boldsymbol{A}^* = \boldsymbol{A}^{**}$:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

- Die beiden Zeilenvertauschungen können zusammengefasst werden durch Multiplikation von $\boldsymbol{P} = \boldsymbol{P}_2 \cdot \boldsymbol{P}_1$, also $\boldsymbol{P} \cdot \boldsymbol{A} = \boldsymbol{A}^{**}$ wobei

$$\boldsymbol{P} = \boldsymbol{P}_2 \cdot \boldsymbol{P}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Mit dieser Permutationsmatrix erhält man dann als $\boldsymbol{RL}$-Zerlegung

$$\boldsymbol{PA} = \boldsymbol{LR}$$

und das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ lässt sich schreiben als $\boldsymbol{PAx} = \boldsymbol{Pb}$ bzw. $\boldsymbol{LRx} = \boldsymbol{Pb}$ und in den zwei Schritten lösen:

$$\begin{aligned} \boldsymbol{Ly} &= \boldsymbol{Pb} \Rightarrow \boldsymbol{y} = ... \\ \boldsymbol{Rx} &= \boldsymbol{y} \Rightarrow \boldsymbol{x} = ... \end{aligned}$$

Wird die Zerlegung mittels Gauss-Algorithmus mit Spaltenpivotisierung (vgl. Kap. 4.4) durchgeführt, muss man also bei jeder Zeilenvertauschung die dazugehörige Permuationsmatrix berechnen. Zusätzlich muss jede Zeilenvertauschung fortlaufend auch in $\boldsymbol{L}$ für die Elemente in den jeweiligen Zeilen aber nur unterhalb der Diagonalen durchgeführt werden (die Diagonalelemente $l_{ii} = 1$ verändern ihre Position nicht). So erhält man schliesslich $\boldsymbol{L}$, $\boldsymbol{R}$ und $\boldsymbol{P}$. Dieses Verfahren nennt man auch $\boldsymbol{LR}$-Zerlegung mit **Spalten- bzw. Kolonnenmaximumstrategie**.

Beispiel 4.6.2

- Gegeben ist das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 3 & 9 & 12 & 12 \\ -2 & -5 & 7 & 2 \\ 6 & 12 & 18 & 6 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} 51 \\ 2 \\ 54 \\ 79 \end{pmatrix}$$

  Berechen Sie die $\boldsymbol{LR}-$Zerlegung von $\boldsymbol{A}$ mit Spaltenmaximumstrategie und bestimmen Sie anhand von $\boldsymbol{L}$, $\boldsymbol{R}$ und $\boldsymbol{P}$ die Lösung $\boldsymbol{x}$.

- Lösung:

1. Zeilenvertauschung von 1. Zeile mit 3. Zeile in $\boldsymbol{A}$, so dass mit $a_{31} = 6$ das betragsmässig grösste Element auf der Diagonale liegt. $\boldsymbol{P}_1$ bildet diese Zeilenvertauschung ab. Da die Elemente in $\boldsymbol{L}$ unterhalb der Diagonalen noch unbestimmt sind, hat eine Zeilenvertauschung noch keinen Einfluss.

$$\boldsymbol{A}^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ -2 & -5 & 7 & 2 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \ \boldsymbol{P}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ ? & 1 & 0 & 0 \\ ? & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix},$$

$$i = 1,\ j = 2 \Rightarrow z_2 \equiv z_2 - \tfrac{(-2)}{6} z_1 \Rightarrow \boldsymbol{A}_1^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ ? & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix}$$

$$i = 1,\ j = 3 \Rightarrow z_3 \equiv z_3 - \tfrac{3}{6} z_1 \Rightarrow \boldsymbol{A}_2^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 0 & 3 & 3 & 9 \\ 3 & 7 & 38 & 14 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{2} & ? & 1 & 0 \\ ? & ? & ? & 1 \end{pmatrix}$$

$$i = 1,\ j = 3 \Rightarrow z_4 \equiv z_4 - \tfrac{3}{6} z_1 \Rightarrow \boldsymbol{A}_3^* = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & -1 & 13 & 4 \\ 0 & 3 & 3 & 9 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{2} & ? & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

2. Zeilenvertauschung von 2. Zeile mit 3. Zeile in $\boldsymbol{A}_3^*$, auch für die Elemente in der ersten Spalte von$\boldsymbol{L}$.

$$\boldsymbol{A}^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & -1 & 13 & 4 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \ \boldsymbol{P}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \boldsymbol{\frac{1}{2}} & 1 & 0 & 0 \\ \boldsymbol{-\frac{1}{3}} & ? & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

$$i = 2,\ j = 3 \Rightarrow z_3 \equiv z_3 - \tfrac{(-1)}{3} z_2 \Rightarrow \boldsymbol{A}_1^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 14 & 7 \\ 0 & 1 & 29 & 11 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & 0 \\ \frac{1}{2} & ? & ? & 1 \end{pmatrix}$$

$$i = 2,\ j = 4 \Rightarrow z_4 \equiv z_4 - \tfrac{1}{3} z_2 \Rightarrow \boldsymbol{A}_2^{**} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 14 & 7 \\ 0 & 0 & 28 & 8 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & 0 \\ \frac{1}{2} & \frac{1}{3} & ? & 1 \end{pmatrix}$$

3. Zeilenvertauschung von 4. Zeile mit 3. Zeile in $\boldsymbol{A}_2^{**}$, auch für die Elemente in der ersten und zweiten Spalte von$\boldsymbol{L}$:

$$\boldsymbol{A}^{***} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 14 & 7 \end{pmatrix}, \ \boldsymbol{P}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \boldsymbol{\frac{1}{2}} & \boldsymbol{\frac{1}{3}} & 1 & 0 \\ \boldsymbol{-\frac{1}{3}} & \boldsymbol{-\frac{1}{3}} & ? & 1 \end{pmatrix}$$

$$i=3,\ j=4 \Rightarrow z_4 \equiv z_4 - \tfrac{1}{2}z_3 \Rightarrow \boldsymbol{A}_1^{***} = \boldsymbol{R} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix},\ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix}$$

1. Als Resultat erhalten wir damit:

$$\boldsymbol{R} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix},\ \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix},\ \boldsymbol{P} = \boldsymbol{P}_3 \cdot \boldsymbol{P}_2 \cdot \boldsymbol{P}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

und es gilt wie gewünscht

$$\boldsymbol{LR} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 3 & 9 & 12 & 12 \\ 3 & 7 & 38 & 14 \\ -2 & -5 & 7 & 2 \end{pmatrix} = \boldsymbol{PA}$$

2. Für die zu lösenden Gleichungssysteme

$$\begin{aligned} \boldsymbol{Ly} &= \boldsymbol{Pb} \\ \boldsymbol{Rx} &= \boldsymbol{y} \end{aligned}$$

erhalten wir den Vektor $\boldsymbol{y}$ durch Vorwärtseinsetzen:

$$\boldsymbol{Ly} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \boldsymbol{Pb} = \begin{pmatrix} 54 \\ 51 \\ 79 \\ 2 \end{pmatrix} \Rightarrow \boldsymbol{y} = \begin{pmatrix} 54 \\ 24 \\ 44 \\ 6 \end{pmatrix}$$

und die eigentlich gesuchte Lösung $\boldsymbol{x}$ durch Rückwärtseinsetzen:

$$\boldsymbol{Rx} = \begin{pmatrix} 6 & 12 & 18 & 6 \\ 0 & 3 & 3 & 9 \\ 0 & 0 & 28 & 8 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \boldsymbol{y} = \begin{pmatrix} 54 \\ 24 \\ 44 \\ 6 \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

4.5.2 Die Cholesky-Zerlegung

Im folgenden lernen wir ein weiteres Verfahren zur Dreieckszerlegung von Matrizen kennen, einen Spezialfall der $\boldsymbol{LR}$-Zerlegung. Dieses Verfahren ist nach seinem Entdecker André-Louis Cholesky (1875 -1918) benannt, einem französischen Mathematiker. Er entwickelte es für Anwendungen in der Geodäsie. [9] Die Cholesky-Zerlegung funktioniert nicht für allgemeine Matrizen sondern nur für *symmetrische, positiv definite* Matrizen. Falls anwendbar, ist es aber etwa um einen Faktor zwei effizienter als die allgemeine $\boldsymbol{LR}$-Zerlegung.

Definition 4.2: Symmetrische / positiv definite Matrizen [1]

- Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst symmetrisch, falls $\boldsymbol{A}^T = \boldsymbol{A}$ gilt ($\boldsymbol{A}^T$ ist die transponierte Matrix).
- Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst positiv definit, falls für alle $\boldsymbol{x} \in \mathbb{R}^n$, $x \neq 0$ gilt $\boldsymbol{x}^T\boldsymbol{Ax} > 0$.

[9] Gemäss F.R. Helmert (1843-1917), dem Begründer der mathematischen und physikalischen Theorien der modernen Geodäsie, ist die Geodäsie die Wissenschaft von der Ausmessung und Abbildung der Erdoberfläche und umfasst die Bestimmung der geometrischen Figur der Erde (Geoid), ihres Schwerefeldes und der Orientierung im Weltraum (siehe Abb. 4.5).

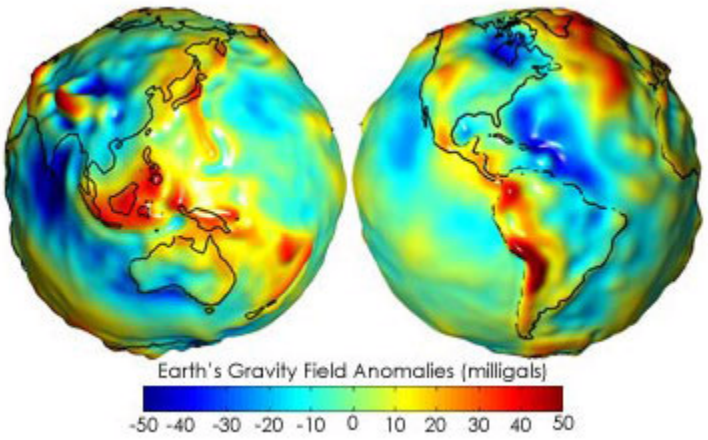


Abbildung 4.5: These "gravity anomaly" maps show where models of the Earth's gravity field based on GRACE data differ from a simplified mathematical model that assumes the Earth is perfectly smooth and featureless. Areas colored yellow, orange, or red are areas where the actual gravity field is larger than the featureless-Earth model predicts—such as the Himalayan Mountains in Central Asia (top left of the left-hand globe)—while the progressively darker shades of blue indicate places where the gravity field is less—such as the area around Hudson Bay in Canada (top center of right-hand globe). From NASA's GRACE Mission (Grvity Recovery and Climate Experiment), see http://earthobservatory.nasa.gov/Features/GRACE/.

Natürlich ist diese Bedingung eine gewisse Einschränkung, allerdings werde wir sehen, dass sie für viele Anwendungen erfüllt ist, so führt z.B. das Ausgleichsproblem i.A. auf ein Gleichungssystem mit symmetrischer und positiv definiter Matrix $\boldsymbol{A}$.

Für solche Matrizen kann nun gezeigt werden, dass immer eine $\boldsymbol{LR}$-Zerlegung existiert, wobei $\boldsymbol{R}$ hier zusätzlich die besonders einfache Form $\boldsymbol{R} = \boldsymbol{L}^T$ bzw. $\boldsymbol{R}^T = \boldsymbol{L}$ besitzt. Die Matrix $\boldsymbol{A}$ können wir dann schreiben als $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$ bzw. als $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$ und es gilt der folgende

Satz 4.2: Cholesky Zerlegung [1]
Für jede positiv definite $n \times n$ Matrix $\boldsymbol{A}$ gibt es genau eine rechts-obere Dreiecksmatrix $\boldsymbol{R}$ mit $r_{ii} > 0$ für $i = 1, ..., n$ und $\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}$. Diese Zerlegung heisst **Cholesky-Zerlegung** von $\boldsymbol{A}$.

Die Berechnung der Cholesky-Zerlegung geschieht anhand des folgenden Algorithmus, der uns die Koeffizienten r_{ij} der oberen Dreiecksmatrix $\boldsymbol{R}$ berechnet und gleichzeitig überprüft, ob $\boldsymbol{A}$ positiv definit ist:

Cholesky-Algorithmus [1]:
Gegeben sei eine symmetrische $n \times n$ Matrix $\boldsymbol{A}$. Für $i = 1, ..., n$ berechne:

- $S = a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2$ (für $i = 1$ ist also $S = a_{ii}$)
- falls $S \leq 0$, dann ist $\boldsymbol{A}$ nicht positiv definit $\rightarrow$stopp.
- falls $S > 0$:
  - $r_{ii} = \sqrt{S}$
  - für $j = i + 1, ..., n$: $r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj} \right)$

Offensichtlich ist dieser Algorithmus weniger anschaulich als die Gauss- Elimination. Auf einen Beweis soll an dieser Stelle verzichtet werden. Stattdessen veranschaulichen wir ihn an folgendem Beispiel:

Beispiel 4.7:

- Es soll geprüft werden, ob die Matrix
$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 3 & 7 & 26 \end{pmatrix}$$
positiv definit ist und wenn ja, die entsprechende Cholesky-Zerlegung berechnet werden.

- Lösung: $\boldsymbol{A}$ ist symmetrisch, also können wir den Cholesky-Algorithmus anwenden:

  - $i = 1 \Rightarrow S = a_{11} - \sum_{k=1}^{0} r_{k1}^2 = a_{11} = 1 > 0 \Rightarrow r_{11} = \sqrt{1} = 1$

    * $j = 2 \Rightarrow r_{12} = \frac{1}{r_{11}}(a_{12} - \sum_{k=1}^{0} r_{k1} r_{k2}) = a_{12} = 2$

    * $j = 3 \Rightarrow r_{13} = \frac{1}{r_{11}}(a_{13} - \sum_{k=1}^{0} r_{k1} r_{k3}) = a_{13} = 3$

  - $i = 2 \Rightarrow S = a_{22} - \sum_{k=1}^{1} r_{k2}^2 = a_{22} - r_{12}^2 = 5 - 4 = 1 > 0 \Rightarrow r_{22} = \sqrt{1} = 1$

    * $j = 3 \Rightarrow r_{23} = \frac{1}{r_{22}}(a_{23} - \sum_{k=1}^{1} r_{k2} r_{k3}) = \frac{1}{r_{22}}(a_{23} - r_{12} r_{13}) = \frac{1}{1}(7 - 2 \cdot 3) = 1$

  - $i = 3 \Rightarrow S = a_{33} - \sum_{k=1}^{2} r_{k3}^2 = a_{33} - r_{13}^2 - r_{23}^2 = 26 - 3^2 - 1^2 = 16 > 0 \Rightarrow r_{33} = \sqrt{16} = 4$

  Da bei keinem der Diagonalelemente eine Wurzel aus einer negativen Zahl gezogen werden sollte, konnte der Algorithmus bis zum Ende durchgeführt werden, d.h die Matrix ist positiv definit und wir haben
$$\boldsymbol{R} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix}$$
$$\text{bzw. } \boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 3 & 7 & 26 \end{pmatrix}$$

- Bemerkung: für eine 3×3 Matrix $\boldsymbol{A} = (a_{ij})$ können die obigen Schritte zusammengefasst werden als
$$\boldsymbol{R} = \begin{pmatrix} \sqrt{a_{11}} & \frac{a_{12}}{r_{11}} & \frac{a_{13}}{r_{11}} \\ 0 & \sqrt{a_{22} - r_{12}^2} & \frac{1}{r_{22}}(a_{23} - r_{12} r_{13}) \\ 0 & 0 & \sqrt{a_{33} - r_{13}^2 - r_{23}^2} \end{pmatrix}$$
wobei $r_{11} = \sqrt{a_{11}}$, $r_{12} = \frac{a_{12}}{r_{11}}$ etc.

Aufgabe 4.5:

- Berechnen Sie die Cholesky-Zerlegung für die folgende Matrix:
$$\boldsymbol{A}_1 = \begin{pmatrix} 4 & -2 & 6 \\ -2 & 5 & -1 \\ 6 & -1 & 26 \end{pmatrix}$$

- Optional: Implementieren Sie den Cholesky-Algorithmus in MATLAB und testen Sie Ihn an den folgenden Matrizen:

$$\boldsymbol{A}_2 = \begin{pmatrix} 9 & 12 & 6 \\ 12 & 25 & 23 \\ 6 & 23 & 78 \end{pmatrix}, \ \boldsymbol{A}_3 = \begin{pmatrix} 4 & -8 & 6 \\ -8 & 17 & -8 \\ 6 & -8 & 34 \end{pmatrix}, \ \boldsymbol{A}_4 = \begin{pmatrix} 36 & -24 & 18 \\ -24 & 17 & -8 \\ 18 & -8 & 25 \end{pmatrix},$$

$$\boldsymbol{A}_5 = \begin{pmatrix} 64 & -40 & 16 \\ -40 & 29 & -4 \\ 16 & -4 & 62 \end{pmatrix}, \ \boldsymbol{A}_6 = \begin{pmatrix} 9 & -21 & 6 \\ -21 & 49 & -14 \\ 6 & -14 & 29 \end{pmatrix}$$

$$\boldsymbol{A}_7 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 & 26 & 29 \\ 3 & 8 & 14 & 20 & 26 & 32 & 38 & 44 & 50 & 56 \\ 4 & 11 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 5 & 14 & 26 & 40 & 55 & 70 & 85 & 100 & 115 & 130 \\ 6 & 17 & 32 & 50 & 70 & 91 & 112 & 133 & 154 & 175 \\ 7 & 20 & 38 & 60 & 85 & 112 & 140 & 168 & 196 & 224 \\ 8 & 23 & 44 & 70 & 100 & 133 & 168 & 204 & 240 & 276 \\ 9 & 26 & 50 & 80 & 115 & 154 & 196 & 240 & 285 & 330 \\ 10 & 29 & 56 & 90 & 130 & 175 & 224 & 276 & 330 & 385 \end{pmatrix}$$

4.6 Fehlerrechnung und Aufwandabschätzung

4.6.1 Fehlerrechnung bei linearen Gleichungssystemen

Wie bereits in Kapitel 2 ausgeführt, können Computer nicht alle reellen Zahlen darstellen, weswegen alle Zahlen intern gerundet werden, damit sie in die endliche Menge der maschinendarstellbaren Zahlen passen. Hierdurch entstehen Rundungsfehler. Selbst wenn sowohl die Eingabewerte als auch das Ergebnis eines Algorithmus maschinendarstellbare Zahlen sind, können solche Fehler auftreten, denn auch die (möglicherweise nicht darstellbaren) Zwischenergebnisse eines Algorithmus werden gerundet. Aufgrund von diesen Fehlern aber auch wegen Eingabe– bzw. Messfehlern in den vorliegenden Daten oder Fehlern aus vorhergehenden numerischen Rechnungen, wird durch einen Algorithmus üblicherweise nicht die exakte Lösung $\boldsymbol{x}$ des linearen Gleichungssystems

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

berechnet, sondern eine Näherungslösung $\widetilde{\boldsymbol{x}}$. Um dies formal zu fassen, führt man ein "benachbartes" oder "gestörtes" Gleichungssystems

$$\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}} = \boldsymbol{b} + \triangle\boldsymbol{b}$$

ein, für das $\widetilde{\boldsymbol{x}}$ gerade die exakte Lösung ist. Dabei ist $\triangle\boldsymbol{b}$ das *Residuum* oder der *Defekt* der Näherungslösung $\widetilde{\boldsymbol{x}}$. Den Vektor $\triangle\boldsymbol{x} = \widetilde{\boldsymbol{x}} - \boldsymbol{x}$ nennen wir den Fehler der Näherungslösung $\widetilde{x}$. Da Rundung und andere Fehlerquellen i.A. nur kleine Fehler bewirken, ist es gerechtfertigt anzunehmen, dass der noch zu definierende 'Betrag' $\parallel \triangle\boldsymbol{b} \parallel$ 'klein' ist.

Das Ziel dieses Abschnittes ist es nun, aus der Grösse des Residuum $\parallel \triangle\boldsymbol{b} \parallel$ auf die Grösse des Fehlers $\parallel \triangle\boldsymbol{x} \parallel$ zu schließen. Insbesondere wollen wir untersuchen, wie sensibel die Grösse $\parallel \widetilde{\boldsymbol{x}} \parallel$ von $\parallel \triangle\boldsymbol{b} \parallel$ abhängt, d.h. ob kleine Residuen $\parallel \triangle\boldsymbol{b} \parallel$ große Fehler in$\parallel \widetilde{\boldsymbol{x}} \parallel$ hervorrufen können. Diese Analyse ist unabhängig von dem verwendeten Lösungsverfahren, da wir hier nur das Gleichungssystem selbst und kein explizites Verfahren betrachten. Um diese Analyse durchzuführen, brauchen wir das Konzept der Norm.

Definition 4.3: Vektornorm [1]
Eine Abbildung $\|.\| : \mathbb{R}^n \longrightarrow \mathbb{R}$ heisst Vektornorm, wenn die folgenden Bedingungen für alle $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ erfüllt sind:

- $\|\boldsymbol{x}\| \geq 0$ und $\|\boldsymbol{x}\| = 0 \Longleftrightarrow x = 0$
- $\|\lambda\boldsymbol{x}\| = \mid \lambda \mid \|\boldsymbol{x}\|$
- $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ "Dreiecksungleichung"

Die drei gebräuchlichsten Vektornormen sind die folgenden:

Definition 4.4: Vektornormen / Matrixnormen [1]

- Für Vektoren $\boldsymbol{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}} \in \mathbb{R}^n$ gibt es die folgenden Vektornormen:

$$\begin{aligned} \text{1-Norm, Summennorm} &: \| \boldsymbol{x} \|_1 = \sum_{i=1}^{n} | x_i | \\ \text{2-Norm, euklidische Norm} &: \| \boldsymbol{x} \|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \\ \infty\text{-Norm, Maximumnorm} &: \| \boldsymbol{x} \|_\infty = \max_{i=1,...,n} | x_i | \end{aligned}$$

- Für eine $n \times n$ Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ sind mit den Vektornormen die folgenden Matrixnormen verbunden, welche die Eigenschaften der Definition 4.3 ebenfalls erfüllen:

$$\begin{aligned} \text{1-Norm, Spaltensummennorm} &: \| \boldsymbol{A} \|_1 = \max_{j=1,...,n} \sum_{i=1}^{n} | a_{ij} | \\ \text{2-Norm, Spektralnorm} &: \| \boldsymbol{A} \|_2 = \sqrt{\rho(\boldsymbol{A}^T \boldsymbol{A})} \\ \infty\text{-Norm, Zeilensummennorm} &: \| \boldsymbol{A} \|_\infty = \max_{i=1,...,n} \sum_{j=1}^{n} | a_{ij} | \end{aligned}$$

Bemerkungen:

- Die euklidische Norm entspricht dem herkömmlichen Verständnis der Länge eines Vektors, die beiden anderen Vektornormen sind aber im Zusammenhang mit Matrixoperationen einfacher berechenbar. Insbesondere ist die 2-Norm für Matrizen hier nur zur Vollständigkeit aufgeführt, wir werden im Folgenden also nur die 1- und $\infty-$ Norm verwenden.

Beispiel 4.8 [1]:

- Berechnen Sie die 1-, 2 -, und $\infty-$ Norm des Vektors $\begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix}$ sowie die 1- und $\infty-$ Norm von $\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & -2 \\ 7 & -3 & 5 \end{pmatrix}$.

  Lösung:

$$\left\| \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} \right\|_1 = 1 + 2 + 3 = 6, \quad \left\| \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} \right\|_2 = \sqrt{1 + 2^2 + 3^2} = \sqrt{14},$$

$$\left\| \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} \right\|_\infty = \max\{1, 2, 3\} = 3$$

$$\left\| \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & -2 \\ 7 & -3 & 5 \end{pmatrix} \right\|_1 = \max\{1 + 3 + 7, 2 + 4 + 3, 3 + 2 + 5\} = 11$$

$$\left\| \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & -2 \\ 7 & -3 & 5 \end{pmatrix} \right\|_\infty = \max\{1 + 2 + 3, 3 + 4 + 2, 7 + 3 + 5\} = 15.$$

Für die Fehlerabschätzung von $\widetilde{\boldsymbol{x}}$ und $\widetilde{\boldsymbol{b}}$ gilt der folgende

Satz 4.3: Abschätzung für fehlerbehaftete Vektoren [1]

- Sei $\| \cdot \|$eine Norm, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ eine reguläre $n \times n$ Matrix und $\boldsymbol{x}$, $\widetilde{\boldsymbol{x}}$, $\boldsymbol{b}$, $\widetilde{\boldsymbol{b}} \in \mathbb{R}^n$ mit $\boldsymbol{Ax} = \boldsymbol{b}$ und $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$. Dann gilt für den absoluten und den relativen Fehler in $\boldsymbol{x}$:
  - $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \| \leq \| \boldsymbol{A}^{-1} \| \cdot \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|$
  - $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}\|} \leq \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \| \cdot \frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|}{\|\boldsymbol{b}\|}$ falls $\| \boldsymbol{b} \| \neq 0$
- Die Zahl cond$(\boldsymbol{A}) = \| \boldsymbol{A} \| \cdot \| \boldsymbol{A}^{-1} \|$ nennt man Konditionszahl der Matrix $\boldsymbol{A}$ bzgl. der verwendeten Norm.

Bemerkungen:

- Für Matrizen, deren Kondition cond$(\boldsymbol{A})$ groß ist, können sich kleine Fehler im Vektor $\boldsymbol{b}$ (bzw. Rundungsfehler im Verfahren) zu großen Fehlern im Ergebnis $\boldsymbol{x}$ verstärken. Man spricht in diesem Fall von schlecht konditionierten Matrizen.

Beispiel 4.9 [1]:

- Untersuchen Sie die Fehlerfortpflanzung im linearen Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} 2 & 4 \\ 4 & 8.1 \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix}$$

  für den Fall, dass die rechte Seite von $\widetilde{\boldsymbol{b}}$ in jeder Komponente um maximal 0.1 von $\boldsymbol{b}$ abweicht.
- Lösung: Wir betrachten das System $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$, wobei $\widetilde{\boldsymbol{b}}$ maximal um 0.1 von jeder Komponente von $\boldsymbol{b}$ abweicht. Zuerst müssen wir eine der möglichen Norm wählen. Hierfür ist die $\infty-$ Norm besonders geeignet, da wir schreiben können $\| \widetilde{\boldsymbol{b}} - \boldsymbol{b} \|_\infty \leq 0.1$. Zusätzlich haben wir $\| \boldsymbol{A} \|_\infty = 12.1$ und mit

$$\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix} = \frac{1}{2 \cdot 8.1 - 4 \cdot 4} \begin{pmatrix} 8.1 & -4 \\ -4 & 2 \end{pmatrix}$$

  erhalten wir $\| \boldsymbol{A}^{-1} \|_\infty = \frac{12.1}{0.2} = 60.5$ und für die Konditionszahl $cond(\boldsymbol{A})$ erhalten wir in der $\infty-$ Norm $cond(\boldsymbol{A})_\infty = \| \boldsymbol{A} \|_\infty \| \boldsymbol{A}^{-1} \|_\infty = 12.1 \cdot 60.5 = 732.05$. Mit dem obigen Satz gilt also

$$\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty \leq \| \boldsymbol{A^{-1}} \|_\infty \cdot \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty \leq 60.5 \cdot 0.1 = 6.05$$

$$\frac{\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty}{\| \boldsymbol{x} \|_\infty} \leq cond(\boldsymbol{A})_\infty \frac{\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty}{\| \boldsymbol{b} \|_\infty} \leq 732 \cdot \frac{0.1}{1.5} = 48.8$$

Wie ist dies nun zu interpretieren? Die Lösung $\widetilde{\boldsymbol{x}}$ des gestörten Systems $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$ wird also von der Lösung x des exakten Systems $\boldsymbol{Ax} = \boldsymbol{b}$ in jeder Komponente um maximal 6.05 abweichen (absoluter Fehler), und der relative Fehler wird maximal 48.8 betragen. Testen wir das an einem konkreten Beispiel.

Beispiel 4.10 [1]:

- Betrachten Sie obiges Beispiel und nehmen sie für die gestörte rechte Seite $\widetilde{\boldsymbol{b}} = \begin{pmatrix} 0.9 \\ 1.6 \end{pmatrix}$. Berechnen sie die Lösungen von $\boldsymbol{Ax} = \boldsymbol{b}$ und $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$. Berechnen Sie anschliessend den absoluten Fehler $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty$ und den relativen Fehler $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|_\infty}{\|\boldsymbol{x}\|_\infty}$. Vergleichen Sie mit $\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty$ und $\frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|_\infty}{\|\boldsymbol{b}\|_\infty}$.
- Lösung: Wir erhalten $\boldsymbol{x} = \begin{pmatrix} 10.5 \\ -5 \end{pmatrix}$ und $\widetilde{\boldsymbol{x}} = \begin{pmatrix} 4.45 \\ -2 \end{pmatrix}$. Mit $\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty = 0.1$ und $\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty = 6.05$ sehen wir, dass der absolute Fehler um den maximal möglichen Faktor 60.5 verstärkt worden ist. Mit $\frac{\|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|_\infty}{\|\boldsymbol{b}\|_\infty} = \frac{0.1}{1.5} = 0.0667$ und $\frac{\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \frac{6.05}{10.5} = 0.5762$ wurde der relative Fehler um einen Faktor 8.6 verstärkt, weniger als der maximal mögliche Faktor von 732.

Wir waren bisher davon ausgegangen, dass die Matrix $\boldsymbol{A}$ selbst exakt ist. Wie verhält sich die Fehlerabschätzung nun unter der Annahme, dass auch noch $\boldsymbol{A}$ fehlerbehaftet ist, wir es also mit einem Gleichungssystem

$$\widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$$

zu tun haben? Dafür gilt die folgende Fehlerabschätzung

Satz 4.4: Abschätzung für fehlerbehaftete Matrix [1]
Sei $\| \, . \, \|$eine Norm, $\boldsymbol{A}, \widetilde{\boldsymbol{A}} \in \mathbb{R}^{n \times n}$ reguläre $n \times n$ Matrizen und $\boldsymbol{x}$, $\widetilde{\boldsymbol{x}}$, $\boldsymbol{b}$, $\widetilde{\boldsymbol{b}} \in \mathbb{R}^n$ mit $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ und $\boldsymbol{A}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$. Falls

$$\mathrm{cond}(\boldsymbol{A}) \cdot \frac{\| \boldsymbol{A} - \widetilde{\boldsymbol{A}} \|}{\| \boldsymbol{A} \|} < 1$$

dann gilt:

$$\frac{\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|}{\| \boldsymbol{x} \|} \leq \frac{\mathrm{cond}(\boldsymbol{A})}{1 - \mathrm{cond}(\boldsymbol{A}) \cdot \frac{\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|}{\|\boldsymbol{A}\|}} \cdot \left(\frac{\| \boldsymbol{A} - \widetilde{\boldsymbol{A}} \|}{\| \boldsymbol{A} \|} + \frac{\| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|}{\| \boldsymbol{b} \|} \right)$$

Bemerkung:

- Für den Fall, dass $\boldsymbol{A}$ exakt gegeben ist, gilt $\frac{\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|}{\|\boldsymbol{A}\|} = 0$ und der relative Fehler für $\boldsymbol{x}$ aus Satz 4.4 reduziert sich auf den relativen Fehler in Satz 4.3.

Beispiel 4.11 [1]:

- Nehmen Sie noch einmal das Beispiel 4.9 und untersuchen Sie die Fehlerfortpflanzung unter der zusätzlichen Annahme, dass die Matrix $\boldsymbol{A}$ um maximal 0.003 elementweise gestört ist.
- Lösung: Wir hatten bereits die folgenden Grössen berechnet

  $$\| \boldsymbol{A} \|_\infty = 12.1, \ \mathrm{cond}(\boldsymbol{A}) = 732.05, \ \|\boldsymbol{b}\|_\infty = 1.5, \ \| \boldsymbol{b} - \widetilde{\boldsymbol{b}} \|_\infty \leq 0.1$$

  Wenn nun jedes Element von $\boldsymbol{A}$ um maximal 0.003 gestört wird, summiert sich diese Störung in der $\infty-$Norm auf und wir erhalten $\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|_\infty \leq 0.006$ und damit

  $$\mathrm{cond}(\boldsymbol{A}) \cdot \frac{\| \boldsymbol{A} - \widetilde{\boldsymbol{A}} \|_\infty}{\| \boldsymbol{A} \|_\infty} \leq 0.363 < 1.$$

  Wir können also die Abschätzung aus Satz 4.4 anwenden und erhalten

  $$\frac{\| \boldsymbol{x} - \widetilde{\boldsymbol{x}} \|_\infty}{\| \boldsymbol{x} \|_\infty} \leq \frac{732.05}{1 - 0.363} \left(\frac{0.006}{12.1} + \frac{0.1}{1.5} \right) \leq 77.2$$

4.6.2 Aufwandabschätzung[10]

Ein wichtiger Aspekt bei der Analyse numerischer Verfahren ist die Abschätzung, wie viel Aufwand diese Verfahren in der Regel benötigen, um zu dem gewünschten Ergebnis zu kommen. Dies hängt entscheidend von der Leistungsfähigkeit des verwendeten Computers ab. Deshalb wird nicht direkt die Zeit abgeschätzt, sondern vielmehr die Anzahl der Rechenoperationen, die ein Algorithmus benötigt. Da hierbei die Gleitkommaoperationen, also Addition, Multiplikation etc. von reellen Zahlen, die mit Abstand zeitintensivsten Operationen sind, beschränkt man sich in der Analyse üblicherweise auf diese.

Bisher haben wir nur direkte Verfahren angeschaut, welche nach einer endlichen Anzahl von Rechenschritten die 'exakte' Lösung liefern. Natürlich hängt hierbei die Anzahl Schritte von deren Dimension n der Matrix $\boldsymbol{A}$ ab. Es genügt also, die Anzahl der dafür benötigten Gleitkommaoperationen in Abhängigkeit von n zu bestimmen. Dafür benötigt man die Gleichungen

$$\sum_{i=1}^{n} i = \frac{(n+1)n}{2} \ \text{ und } \ \sum_{i=1}^{n} i^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n.$$

[10] Kapitel hauptsächlich übernommen aus [3]

Beispiel 4.12:

- Wie viele Gleitkommaoperationen benötigt das Rückwärtseinsetzen gemäss Gleichung 4.5?
- Lösung: Für jedes x_i haben wir

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}, \qquad i = n, n-1, ..., 1.$$

Betrachten wir zuerst nur die Multiplikationen und Divisionen. Für $i = n$ haben wir eine Division, für $i = n-1$ haben wir eine Multiplikation und eine Division, etc. Für $i = 1$ schliesslich haben wir $n-1$ Multiplikationen und eine Division. Das ergibt für diese Operationen also

$$1 + 2 + 3 + ... + n = \sum_{i=1}^{n} i = \frac{(n+1)n}{2}.$$

Nun schauen wir uns noch die Anzahl Additionen und Subtraktionen an. Für $i = n$ haben wir keine, für $i = n-1$ haben wir eine Subtraktion, etc., das ergibt dann

$$0 + 1 + 2 + ... + n - 1 = \sum_{i=1}^{n-1} i = \frac{(n-1+1)(n-1)}{2} = \frac{n(n-1)}{2}.$$

Für die Summe beider Operationstypen erhalten wir also

$$\frac{n^2}{2} + \frac{n}{2} + \frac{n^2}{2} - \frac{n}{2} = n^2.$$

Für das Vorwärtseinsetzen erhalten wir natürlich das gleiche Resultat.

Für die Gauss-Elimination erhält man nach einer ähnlichen Betrachtung die Anzahl Gleitkommaoperationen (ohne Pivotisierung) zu

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n.$$

Die Anzahl Operationen für die LR-Zerlegung ist identisch, wenn sie mit der Gauss-Elimination durchgeführt wurde. Für das Choleski-Verfahren erhält man (ohne Beweis)

$$\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n.$$

Für die vollständige Lösung eines linearen Gleichungssystems müssen nun die Operationen für Rückwärtseinsetzen (Gauss) bzw. Rückwärts- *und* Vorwärtseinsetzen (LR-Zerlegung und Cholesky) noch addiert werden. Für die Gauss-Elimination erhalten wir dann

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n + n^2 = \frac{2}{3}n^3 + \frac{5}{2}n^2 - \frac{13}{6}n,$$

für die LR-Zerlegung

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{13}{6}n + 2n^2 = \frac{2}{3}n^3 + \frac{7}{2}n^2 - \frac{13}{6}n,$$

für die Cholesky-Zerlegung

$$\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n + 2n^2 = \frac{1}{3}n^3 + \frac{5}{2}n^2 + \frac{1}{6}n$$

Operationen. Berücksichtigt man, dass für große n die "n^3–Terme" dominant werden, so ergibt sich, dass das Choleski–Verfahren etwa doppelt so schnell wie die Gauß– Elimination ist.

Im Vergleich dazu müssen bei der Cramerschen Regel $n+1$ Determinanten und n Quotienten bestimmt werden, was für jede Determinante mit der Regel von Leibniz $(n-1) \cdot n!$ Multiplikationen und $n! - 1$ Additionen beinhaltet. Das ergibt also

$$(n+1)\left((n-1) \cdot n! + n! - 1\right) + n = n(n+1)! - 1$$

Punktoperationen.

Um einen Eindruck von den tatsächlichen Rechenzeiten zu bekommen, nehmen wir an, dass wir einen handelsüblichen PC verwenden, der mit einer 3GHz Quad-Core CPU ausgestattet ist mit einer tatsächlichen Leistung von 30 GFLOPS (FLOPS = floating point operations per second), d.h. mit $30 \cdot 10^9$ Gleitkommaoperationen pro Sekunde (zum Vergleich: die Zuse Z3 schaffte mit einer Taktrate von 5.3 Hz rund 1 FLOPS) . Nehmen wir weiterhin an, dass wir Implementierungen der obigen Algorithmen haben, die diese Leistung optimal ausnutzen. Dann ergeben sich für $n \times n$ Gleichungssysteme die folgenden (ungefähren) Rechenzeiten

n	Gauss	LR-Zerlegung	Cholesky	Cramer
10^1	30 ns	30 ns	15 ns	0.1 s
10^2	23 μs	23 μs	12 μs	10^{143}y
10^3	22 ms	22 ms	11 ms	–
10^4	22 s	22 s	11 s	–
10^5	6 h	6 h	3 h	–
10^6	257 d	257 d	129 yd	–
10^7	704 y	704 y	352 y	–

Wie man sieht, wächst die benötigte Zeit für den Gauss-Algorithmus, die $\boldsymbol{LR}-$Zerlegung und dem Cholesky-Algorithmus um einen Faktor $10^3 = 1000$, wenn n um einen Faktor 10 erhöht wird. Für die Cramersche Regel benötigt man für $n = 10$ 'erst' 0.1 Sekunde, für $n = 20$ bereits rund 1000 Jahre, für $n = 25$ bereits 10 Mia. Jahre und für $n = 100$ läppische 10^{143} Jahre. Ein eindrückliches Beispiel, wie schnell die Fakultät wächst.

Ab $n > 10^5$ kommt aber auch für die anderen Algorithmen die Wartezeit in einen Bereich, der kaum mehr akzeptabel ist. Im nächsten Abschnitt werden wir deshalb die iterativen Verfahren kennen lernen, die zwar nicht mehr die 'exakte' Lösung berechnen, dafür aber wesentlich schneller sind.

Zum Abschluss dieses Abschnitts wollen wir aber noch ein etwas gröberes Konzept zur Aufwandsabschätzung betrachten, welches für praktische Anwendungen häufig ausreicht. Man interessiert sich dabei nicht für die genaue Anzahl durchzuführender Operationen sondern nur für eine Abschätzung bei grossen Dimensionen, d.h. wie der Aufwand sich asymptotisch für $n \to \infty$ verhält. Dabei spricht man von der Ordnung eines Algorithmus:

Definition 4.5: Ordnung [3]

- Ein Algorithmus hat die Ordnung $O(n^q)$, wenn $q > 0$ die minimale Zahl ist, für die es eine Konstante C > 0 gibt, so dass der Algorithmus für alle $n \in N$ weniger als Cn^q Operationen benötigt.

Die Zahl q ist aus den obigen Aufwandsabschätzungen einfach abzulesen. Sie entspricht gerade der höchsten auftretenden Potenzen von n. Daraus folgt, dass Vor- und Rückwärtseinsetzen sind von der Ordnung $O(n^2)$, während das Gauss-Verfahren sowie die $\boldsymbol{LR}$- und Cholesky-Zerlegung von der Ordnung $O(n^3)$ sind.

4.7 Iterative Verfahren

Wir haben bereits gesehen, dass die bisher betrachteten direkten Verfahren die Ordnung $O(n^3)$ besitzen. Für große Gleichungssysteme, die in der Praxis durchaus auftreten, führt dies wie oben gesehen zu unakzeptabel hohen Rechenzeiten. Eine Klasse von Verfahren, die eine niedrigere Ordnung hat, sind die iterativen Verfahren. Allerdings zahlt man für den geringeren Aufwand einen Preis: Man kann bei diesen Verfahren nicht mehr erwarten, eine (bis auf Rundungsfehler) exakte Lösung zu erhalten, sondern muss von vornherein eine gewisse Ungenauigkeit im Ergebnis in Kauf nehmen.

Das Grundprinzip iterativer Verfahren funktioniert dabei wie folgt: Ausgehend von einem Startvektor $\boldsymbol{x}^{(0)}$ berechnet man mittels einer Rechenvorschrift

$$F : \mathbb{R}^n \to \mathbb{R}^n$$

iterativ eine Folge von Vektoren

$$\boldsymbol{x}^{(k+1)} = F(\boldsymbol{x}^{(k)}) \text{ mit } k = 0, 1, 2, ...$$

die für $k \to \infty$ gegen die Lösung $\boldsymbol{x}$ des Gleichungssystems $\boldsymbol{Ax} = \boldsymbol{b}$ konvergieren. Wenn die gewünschte Genauigkeit erreicht ist, wird die Iteration abgebrochen und der letzte Wert $\boldsymbol{x}^{(i)}$ als Näherung des Ergebnisses verwendet.

- Bemerkung zur Notation: ein hochgestellter Index in Klammern $\boldsymbol{x}^{(k)}$ bezeichnet einen Vektor aus $\mathbb{R}^n$ nach der $k-$ten Iteration. Die Elemente des Vektors $\boldsymbol{x}^{(k)}$ werden wie üblich mit einem tiefgestellten Index bezeichnet, z.B. ist also $x_i^{(k)}$ das $i-$te Element des Vektors, bzw.

$$\boldsymbol{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_{n-1}^{(k)} \\ x_n^{(k)} \end{pmatrix}$$

Wir wollen nun versuchen, das obige Problem als Fixpunktiteration zu behandeln. Wir hatten in Kapitel 3 gesehen, dass die allgemeine Fixpunktgleichung die Form $F(x) = x$ hat, d.h. wir wollen die ursprüngliche Gleichung $\boldsymbol{Ax} = \boldsymbol{b}$ in eine ähnliche Form bringen. Dies gelingt uns, wenn wir die Matrix $\boldsymbol{A}$ zerlegen können in eine Form

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R},$$

wobei $\boldsymbol{L}$ eine untere Dreiecksmatrix sein soll (mit $l_{ii} = 0$) , $\boldsymbol{D}$ eine Diagonalmatrix und $\boldsymbol{R}$ eine obere Dreiecksmatrix (mit $r_{ii} = 0$). Die einfachste Form ist

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{R}$$

$$= \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{pmatrix} + \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix} + \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Achtung: diese Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ hier sind nicht die gleichen wie die $\boldsymbol{LR}-$Zerlegung aus Kap. 4.5.1.

4.7.1 Das Jacobi-Verfahren

Wir können mit obiger Zerlegung dann die folgende Fixpunktiteration, die auch als Jacobi- oder Gesamtschritt-Verfahren bekannt ist, durchführen:

Definition 4.6: Jacobi- bzw. Gesamtschrittverfahren [1]

- Zu lösen sei $\boldsymbol{Ax} = \boldsymbol{b}$. Die Matrix $\boldsymbol{A} = (a_{ij})$ sei zerlegt in der Form

$$\boldsymbol{A} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & 0 \end{pmatrix}}_{=: \boldsymbol{L}} + \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}}_{=: \boldsymbol{D}}$$

$$+ \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}}_{=: \boldsymbol{R}}$$

  Dann heisst die Fixpunktiteration

$$\begin{aligned} \boldsymbol{D}\boldsymbol{x}^{(k+1)} &= -(\boldsymbol{L} + \boldsymbol{R})\boldsymbol{x}^{(k)} + \boldsymbol{b} \text{ bzw.} \\ \boldsymbol{x}^{(k+1)} &= -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{R})\boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\boldsymbol{b} \end{aligned}$$

  Gesamtschrittverfahren oder **Jacobi-Verfahren**.

Dabei ist für $\boldsymbol{D}$ die inverse $\boldsymbol{D}^{-1}$ sehr einfach zu berechnen, auf der Diagonalen von $\boldsymbol{D}^{-1}$ stehen einfach die Kehrwerte der Diagonalen von $\boldsymbol{D}$, also

$$\boldsymbol{D} = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix} \Rightarrow \boldsymbol{D}^{-1} = \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{a_{33}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{a_{nn}} \end{pmatrix}$$

Die Herleitung des Jacobi-Verfahrens folgt direkt aus der Zerlegung von $\boldsymbol{A}$:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ (\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{R})\boldsymbol{x} &= \boldsymbol{b} \\ (\boldsymbol{L}+\boldsymbol{R})x+\boldsymbol{D}\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{D}\boldsymbol{x} &= -(\boldsymbol{L}+\boldsymbol{R})x+\boldsymbol{b} \\ \boldsymbol{x} &= -\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{R})\boldsymbol{x}+\boldsymbol{D}^{-1}\boldsymbol{b} \equiv F(\boldsymbol{x}) \end{aligned}$$

womit wir die Fixpunktgleichung bereits aufgestellt haben.

Beispiel 4.13 [1]:

- Wenden Sie das Jacobi-Verfahren auf das folgende System an:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \text{ mit } \boldsymbol{A} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

Lösung: Mit den Bezeichnungen aus (3.7) haben wir:

$$\boldsymbol{L} = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & -2 & 0 \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \quad \boldsymbol{R} = \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Die Iteration lautet somit:

$$\begin{aligned} \boldsymbol{x}^{(n+1)} &= -\boldsymbol{D}^{-1}((\boldsymbol{L}+\boldsymbol{R})\,\boldsymbol{x}^{(n)} - \boldsymbol{b}) \\ &= -\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix} \left(\begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & -2 & 0 \end{pmatrix} \boldsymbol{x}^{(n)} - \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix} \right) \\ &= \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{pmatrix} \boldsymbol{x}^{(n)} + \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix} \end{aligned}$$

Wir wählen als Startvektor den Nullvektor und erhalten:

i	0	1	2	3	4	5
$\boldsymbol{x}^{(i)}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}$	$\begin{pmatrix} 1.2 \\ 2.22 \\ 3.03 \end{pmatrix}$	$\begin{pmatrix} 1.0475 \\ 2.074 \\ 3.048 \end{pmatrix}$	$\begin{pmatrix} 1.0065 \\ 2.0094 \\ 3.0201 \end{pmatrix}$	$\begin{pmatrix} 0.997325 \\ 1.99858 \\ 3.00246 \end{pmatrix}$

Es sieht so aus, als konvergiere diese Folge gegen $(1,2,3)^{\mathrm{T}}$, was übrigens die Lösung des System $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ darstellt. ∎

Üblicherweise wird die Iteration programmiertechnisch nicht mit Matrixmultiplikation durchgeführt, sondern für jede Komponente des Vektors $\boldsymbol{x}$ separat, für obiges Beispiel wäre das also

$$x_1^{(k+1)} = 0.25x_2^{(k)} - 0.25x_3^{(k)} + 1.25 \tag{4.9}$$
$$x_2^{(k+1)} = 0.4x_1^{(k)} - 0.2x_3^{(k)} + 2.2 \tag{4.10}$$
$$x_3^{(k+1)} = -0.2x_1^{(k)} + 0.4x_2^{(k)} + 2.4 \tag{4.11}$$

und in der allgemeinen Form (zur einfacheren Implementation) können wir das schreiben als

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) \qquad i = 1, ..., n.$$

4.7.2 Das Gauss-Seidel-Verfahren

Wenn man nun davon ausgeht, dass nach der k-ten Iteration der Vektor $\boldsymbol{x}^{(k+1)}$ komponentenweise näher an der Lösung liegt als der Vektor vom vorherigen Iterationsschritt $\boldsymbol{x}^{(k)}$, dann ist es im obigen Beispiel vermutlich genauer, die gerade berechnete Komponente $x_1^{(k+1)}$ aus Gleichung (4.9) in die noch zu berechnende Komponente $x_2^{(k+1)}$ in Gleichung (4.10) einzusetzen. Analog setzt man anschliessend die Komponenten $x_1^{(k+1)}$ und $x_2^{(k+1)}$ in die Gleichung (4.11) ein, um $x_3^{(k+1)}$ zu erhalten. Dies führt dann auf die Iteration

$$\begin{aligned} x_1^{(k+1)} &= 0.25x_2^{(k)} - 0.25x_3^{(k)} + 1.25 \\ x_2^{(k+1)} &= 0.4x_1^{(k+1)} - 0.2x_3^{(k)} + 2.2 \\ x_3^{(k+1)} &= -0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} + 2.4 \end{aligned}$$

welche man in Matrix-Form schreiben kann als

$$\boldsymbol{x}^{(k+1)} = \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0 & 0 & -0.2 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}^{(k)} + \begin{pmatrix} 0 & 0 & 0 \\ 0.4 & 0 & 0 \\ -0.2 & 0.4 & 0 \end{pmatrix} \boldsymbol{x}^{(k+1)} + \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}.$$

Mit unseren Matrizen $\boldsymbol{L}$, $\boldsymbol{D}$, und $\boldsymbol{R}$ wird das zu

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1} \left(\boldsymbol{b} - \boldsymbol{L}\boldsymbol{x}^{(k+1)} - \boldsymbol{R}\boldsymbol{x}^{(k)} \right)$$

oder in der allgemeinen Form

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \qquad i = 1, ..., n. \tag{4.12}$$

Jetzt kann man das noch umformen, so dass alle Terme mit $\boldsymbol{x}^{(k+1)}$ auf der linken Seite erscheinen, und erhält damit das sogenannte Gauss-Seidel-Verfahren oder auch Einzelschrittverfahren.

Definition 4.7: Gauss-Seidel bzw. Einzelschrittverfahren [1]

- Zu lösen sei $\boldsymbol{Ax} = \boldsymbol{b}$. Die Matrix $\boldsymbol{A} = (a_{ij})$ sei zerlegt in der Form

$$\boldsymbol{A} = \underbrace{\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & 0 \end{pmatrix}}_{=:\boldsymbol{L}} + \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}}_{=:\boldsymbol{D}}$$

$$+ \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ & & & \ddots & \vdots \\ 0 & 0 & 0 & & \\ \vdots & \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}}_{=:\boldsymbol{R}}$$

  Dann heisst die Fixpunktiteration

$$\begin{aligned} (\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)} &= -\boldsymbol{R}\boldsymbol{x}^{(k)} + \boldsymbol{b} \text{ bzw.} \\ \boldsymbol{x}^{(k+1)} &= -(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{R}\boldsymbol{x}^{(k)} + (\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b} \end{aligned}$$

  Einzelschrittverfahren oder **Gauss-Seidel-Verfahren**.

Beispiel 4.14 [1]:

- Wenden Sie das Gauss-Seidel-Verfahren auf das System aus Beispiel 4.13 an, also:

$$\boldsymbol{Ax} = \boldsymbol{b} \text{ mit } \boldsymbol{A} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

- Lösung: Wir verwenden die allgemeine Gleichung für die Komponenten gemäss (4.12) und erhalten

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}\left(b_1 - \sum_{j=1}^{0} a_{1j}x_j^{(k+1)} - \sum_{j=2}^{3} a_{1j}x_j^{(k)}\right) \\ &= \frac{1}{4}(5 - (-1)x_2^{(k)} - 1x_3^{(k)}) = 1.25 + 0.25x_2^{(k)} - 0.25x_3^{(k)} \\ x_2^{(k+1)} &= \frac{1}{a_{22}}\left(b_2 - \sum_{j=1}^{1} a_{2j}x_j^{(k+1)} - \sum_{j=3}^{3} a_{2j}x_j^{(k)}\right) \\ &= \frac{1}{5}(11 - (-2)x_1^{(k+1)} - 1x_3^{(k)}) = 2.2 + 0.4x_1^{(k+1)} - 0.2x_3^{(k)} \\ x_3^{(k+1)} &= \frac{1}{a_{33}}\left(b_3 - \sum_{j=1}^{2} a_{3j}x_j^{(k+1)} - \sum_{j=4}^{3} a_{3j}x_j^{(k)}\right) \\ &= \frac{1}{5}(12 - 1x_1^{(k+1)} - (-2)x_2^{(k+1)}) = 2.4 - 0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} \end{aligned}$$

  was mit der weiter oben bereits hergeleiteten Iteration für dieses Beispiel natürlich übereinstimmt. Wir wählen den Null-Vektor als Startvektor, also $\boldsymbol{x}^{(0)} = (0,0,0)^T$ und setzen oben ein. Damit erhalten wir $x_1^{(1)} = 1.25$, $x_2^{(1)} = 2.2 + 0.4 \cdot 1.25 = 2.7$, $x_3^{(1)} = 2.4 - 0.2 \cdot 1.25 + 0.4 \cdot 2.7 = 3.23$, also $\boldsymbol{x}^{(1)} = (1.25, 2.7, 3.23)^T$. Weiteres Einsetzen liefert:

i	0	1	2	3	4
$\boldsymbol{x}^{(i)}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1.25\\2.7\\3.23\end{pmatrix}$	$\begin{pmatrix}1.1175\\2.001\\2.9769\end{pmatrix}$	$\begin{pmatrix}1.006025\\2.00703\\3.001607\end{pmatrix}$	$\begin{pmatrix}1.00135575\\2.0002209\\2.99981721\end{pmatrix}$

Also können wir annehmen, dass diese Folge gegen $(1,2,3)^T$ konvergiert, und zwar schneller als mit dem Jacobi-Verfahren. Natürlich hätten wir die Iterationsgleichungen auch etwas übersichtlicher aus dem Zusammenhang

$$(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)} = -\boldsymbol{R}\boldsymbol{x}^{(k)} + \boldsymbol{b}$$

herleiten können:

$$\begin{pmatrix} 4 & 0 & 0 \\ -2 & 5 & 0 \\ 1 & -2 & 5 \end{pmatrix} \boldsymbol{x}^{(k+1)} = - \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}^{(k)} + \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}.$$

Komponentenweise folgt (ohne Berechnung der Inversen $(D+L)^{-1}$)

$$\begin{aligned} 4x_1^{(k+1)} &= -(-x_2^{(k)} + x_3^{(k)}) + 5 \\ -2x_1^{(k+1)} + 5x_2^{(k+1)} &= -x_3^{(k)} + 11 \\ x_1^{(k+1)} - 2x_2^{(k+1)} + 5x_3^{(k+1)} &= 12 \end{aligned}$$

bzw. wie erwartet

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}x_2^{(k)} - \frac{1}{4}x_3^{(k)}) + \frac{5}{4} \\ x_2^{(k+1)} &= \frac{2}{5}x_1^{(k+1)} - \frac{1}{5}x_3^{(k)} + \frac{11}{5} \\ x_3^{(k+1)} &= -\frac{1}{5}x_1^{(k+1)} + \frac{2}{5}x_2^{(k+1)} + \frac{12}{5} \end{aligned}$$

4.7.3 Konvergenz

Wir haben in Kapitel 3.4 bereits Kriterien bezüglich der Konvergenz von Fixpunktiterationen kennengelernt. Diese können direkt auf die vektoriellen Fixpunktgleichungen des Jacobi- und des Gauss-Seidel-Verfahrens angewandt werden, es muss dabei nur eine Norm statt des Betragszeichens verwendet werden.

Definition 4.8: anziehender / abstossender Fixpunkt [1]

- Gegeben sei eine Fixpunktiteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{B}\boldsymbol{x}^{(n)} + \boldsymbol{b} =: F(\boldsymbol{x}^{(n)})$$

  wobei $\boldsymbol{B}$ eine $n \times n$ Matrix ist und $\boldsymbol{b} \in \mathbb{R}^n$.Weiter sei $\| \cdot \|$eine der in Kap. 4.6.1 eingeführten Normen und $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ erfülle $\overline{\boldsymbol{x}} = \boldsymbol{B}\overline{\boldsymbol{x}} + \boldsymbol{b} = F(\overline{\boldsymbol{x}})$. Dann heisst

  - $\overline{\boldsymbol{x}}$ anziehender Fixpunkt, falls $\| \boldsymbol{B} \| < 1$ gilt
  - $\overline{\boldsymbol{x}}$ abstossender Fixpunkt, falls $\| \boldsymbol{B} \| > 1$ gilt.

Unter Anwendung des Banachschen Fixpunktsatzes haben wir dann die folgende Fehlerabschätzung zur Verfügung:

Satz 4.5: Abschätzungen [1]

- Gegeben sei wie in obiger Definition eine Fixpunktiteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{B}\boldsymbol{x}^{(n)} + \boldsymbol{b} =: F(\boldsymbol{x}^{(n)})$$

und $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ sei ein bezüglich der Norm $\| \cdot \|$ anziehender Fixpunkt. Dann konvergiert die Fixpunktiteration für alle Startvektoren $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ gegen $\bar{\boldsymbol{x}}$ und es gelten die Abschätzungen

$$\| \boldsymbol{x}^{(n)} - \bar{\boldsymbol{x}} \| \leq \frac{\| \boldsymbol{B} \|^n}{1 - \| \boldsymbol{B} \|} \| \boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)} \| \quad \text{a-priori Abschätzung}$$

$$\| \boldsymbol{x}^{(n)} - \bar{\boldsymbol{x}} \| \leq \frac{\| \boldsymbol{B} \|}{1 - \| \boldsymbol{B} \|} \| \boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)} \| \quad \text{a-posteriori Abschätzung}$$

Bemerkung: Der Vergleich mit den Definitionen für das Gesamt- und Einzelschrittverfahren liefert die Matrix $\boldsymbol{B}$:

- für das Gesamtschrittverfahren (Jacobi) ist

$$\boldsymbol{B} = -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{R}),$$

- für das Einzelschrittverfahren (Gauss-Seidel) ist

$$\boldsymbol{B} = -(\boldsymbol{D} + \boldsymbol{L})^{-1}\boldsymbol{R}.$$

Ausserdem gilt mit der folgenden Definition:

Definition 4.8: Diagonaldominanz [1]

- $\boldsymbol{A}$ ist eine **diagonaldominante Matrix**, falls eines der beiden folgenden Kriterien gilt:
  - für alle $i = 1, ..., n$: $| a_{ii} | > \sum_{j=1, j \neq i}^{n} | a_{i,j} |$ (Zeilensummenkriterium)
  - für alle $j = 1, ..., n$: $| a_{jj} | > \sum_{i=1, i \neq j}^{n} | a_{i,j} |$ (Spaltensummenkriterium)

Satz 4.6: Konvergenz [1]

- **Falls $\boldsymbol{A}$ diagonaldominant ist, konvergiert das Gesamtschrittverfahren (Jacobi) und auch das Einzelschrittverfahren (Gauss-Seidel) für $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.**

Bemerkungen:

- Die Bedingung $\| \boldsymbol{B} \| < 1$ für einen anziehenden Fixpunkt $\overline{\boldsymbol{x}}$ impliziert, dass $\boldsymbol{A}$ diagonaldominant ist.
- Diagonaldominanz ist nur ein hinreichendes Kriterium. Es gibt durchaus nicht diagonaldominante Matrizen, für die die Verfahren trotzdem konvergieren kann. Ein notwendiges und hinreichendes Kriterium ist, dass der Spektralradius $\rho(\boldsymbol{B}) < 1$.

Aufgabe 4.6 [1]:

- Prüfen Sie, ob das Jacobi-Verfahren in Beispiel 4.13 konvergiert. Schätzen Sie den Fehler des Vektors $\boldsymbol{x}^{(5)}$ ab. Wie viele Schritte sollten Sie rechnen, damit der berechnete Näherungsvektor in jeder Komponente um max. 10^{-4} von der exakten Lösung $\boldsymbol{x} = (1, 2, 3)^T$ abweicht? Vergleichen Sie Ihre Fehlerabschätzung mit den wirklichen Gegebenheiten.
- Bearbeiten Sie die Aufgabenstellung nochmals, aber mit dem Gauss-Seidel-Verfahren und dem Näherungsvektor $\boldsymbol{x}^{(4)}$ aus Beispiel 4.14.

Kapitel 5

Numerische Lösung nicht linearer Gleichungssysteme

In Kapitel 3 haben wir Verfahren kennengelernt, die Nullstellen nichtlinearer Funktionen mit einer Veränderlichen zu bestimmen, also die Gleichung $f(x_0) = 0$ zu lösen für ein nichtlineares $f : \mathbb{R} \longrightarrow \mathbb{R}$. In Kapitel 4 behandelten wir dann den Fall von Systemen von n linearen Gleichungen für n Unbekannte, was man als Nullstellenbestimmung einer linearen Funktion $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ auffassen kann. In diesem Kapitel geht es nun um die Verallgemeinerung bzw. Anwendung der in Kap. 3 und 4 kennengelernten Verfahren auf Systeme von nichtlinearen Gleichungen, also um die Nullstellenbestimmung von nichtlinearen Funktionen $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$.

Lernziele:

- Sie kennen die Definition einer Funktion mit mehreren Variablen und wissen, wie diese grafisch dargestellt werden kann.
- Sie können die partiellen Ableitungen erster Ordnung einer Funktion mit mehreren Variablen definieren und berechnen.
- Sie kennen die Definition der Jacobi-Matrix und können diese für eine gegebene Funktion berechnen. Sie können damit Funktionen linearisieren.
- Sie können die in diesem Kapitel vorgestellten Algorithmen auf nichtlineare Gleichungssysteme anwenden und implementieren.
- Sie können die Unterschiede zwischen den Algorithmen erklären.

5.1 Einleitendes Beispiel

Zur historischen Entwicklung der Theorie nichtlinearer Gleichungssysteme ist in der Literatur wenig zu finden. Einer der Gründe mag sein, dass nichtlineare Gleichungssysteme im Gegensatz zu linearen Gleichungssystemen wesentlich komplexer sein können und Aussagen über die Lösbarkeit oder Konvergenz i.d.R. stark vom spezifischen Problem abhängig sind. Allgemeine Aussagen der Art, wie sie für lineare Gleichungssystemen möglich sind, sind für nichtlineare Gleichungssysteme wesentlich schwieriger. Nichtlineare Gleichungssysteme treten fast automatisch bei der Beschreibung und Lösung realer (häufig zeitabhängiger) Prozesse in Natur und Technik auf, die i.d.R. in Form von nichtlinearen Differentialgleichungen beschrieben werden (z.B. Ausbreitung elektromagnetischer Wellen im dreidimensionalen Raum).

Als einführendes Beispiel betrachten wir ein einfaches System mit zwei nichtlinearen Gleichungen und zwei Variablen (aus [6]):

$$\begin{aligned} f_1(x_1, x_2) &= x_1^2 + x_2 - 11 = 0 \\ f_2(x_1, x_2) &= x_1 + x_2^2 - 7 = 0 \end{aligned}$$

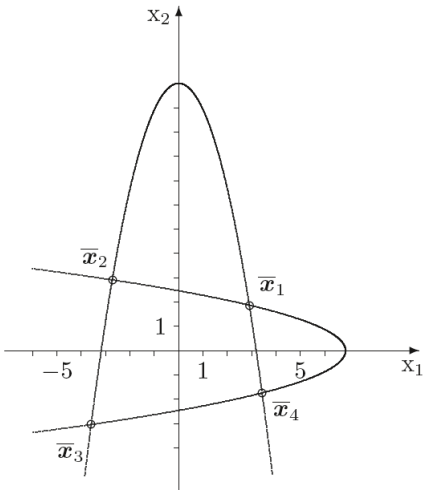


Abbildung 5.1: Grafische Darstellung der durch $f_1(x_1, x_2) = 0$ und $f_2(x_1, x_2) = 0$ implizit definierten Kurven sowie ihre Schnittpunkten (aus [6]).

Gesucht sind die Lösungen des Gleichungssystems. Diese lassen sich interpretieren als die Nullstellen der Funktion $\boldsymbol{f} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ gemäss

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2 - 11 \\ x_1 + x_2^2 - 7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Offensichtlich lässt sich ein solches System nicht in der Form $\boldsymbol{Ax} = \boldsymbol{b}$ darstellen, wie wir es von linearen Gleichungssystemen her kennen. Geometrisch lassen sich die Lösungen in diesem Beispiel bestimmen, indem wir die durch $f_1(x_1, x_2) = 0$ und $f_2(x_1, x_2) = 0$ implizit definierten Kurven in ein (x_1, x_2)-Koordinatensystem einzeichnen und die Schnittpunkte bestimmen, wie in Abb. 5.1 dargestellt. Dabei lautet die explizite Darstellung der Kurven

$$\begin{aligned} x_2 &= -x_1^2 + 11 \\ x_2 &= \sqrt{-x_1 + 7} \end{aligned}$$

und die Schnittpunkte sind

$$\bar{\boldsymbol{x}}_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \bar{\boldsymbol{x}}_2 = \begin{pmatrix} -2.8 \\ 3.2 \end{pmatrix}, \bar{\boldsymbol{x}}_3 = \begin{pmatrix} -3.8 \\ -3.3 \end{pmatrix}, \bar{\boldsymbol{x}}_4 = \begin{pmatrix} 3.4 \\ -1.7 \end{pmatrix}$$

5.2 Funktionen mit mehreren Variablen

Wie wir bei diesem einführenden Beispiel gesehen haben, ist für die numerische Lösung von nichtlinearen Gleichungssystemen das Verständnis von Funktionen mit mehreren Variablen unabdingbar. Deshalb wollen wir die wichtigsten Begriffe, insbesondere denjenigen der partiellen Ableitung, in diesem Abschnitt nochmals in Erinnerung rufen bzw. neu einführen, sofern nötig[1].

5.2.1 Definition einer Funktion mit mehreren Variablen

Die Definition für Funktionen

$$\begin{aligned} f : \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\mapsto y = f(x) \end{aligned}$$

mit der abhängigen Variablen y und der unabhängigen Variablen x kennen wir bereits aus der Analysis und erweitern Sie analog auf Funktionen mit mehreren Variablen:

[1] Wir orientieren uns dabei am Kapitel IV "Differential- und Integralrechnung für Funktionen von mehreren Variablen" aus "Mathematik für Ingenieure und Naturwissenschaftler: Band 2" von Papula [8], greifen aber nur die für uns wichtigsten Punkte auf und passen sie, soweit sinnvoll, an.

Definition 5.1: Funktionen mit mehreren Variablen

- Unter einer Funktion mit n unabhängigen Variablen $x_1, ..., x_n$ und einer abhängigen Variablen y versteht man eine Vorschrift, die jedem geordneten Zahlentupel $(x_1, x_2, ..., x_n)$ aus einer Definitionsmenge $D \subset \mathbb{R}^n$ genau ein Element y aus einer Wertemenge $W \subset \mathbb{R}$ zuordnet. Symbolische Schreibweise:

$$\begin{array}{rcl} f: \ D \subset \mathbb{R}^n & \longrightarrow & W \subset \mathbb{R} \\ (x_1, x_2, ..., x_n) & \mapsto & y = f(x_1, x_2, ..., x_n) \end{array}$$

- Da das Ergebnis $y \in \mathbb{R}$ ein Skalar (eine Zahl) ist, redet man auch von einer **skalarwertigen** Funktion.

Bemerkungen:

- Die obige Definition lässt sich einfach erweitern auf beliebige **vektorwertige** Funktionen, die nicht einen Skalar, sondern einen Vektor als Wert zurückgeben:

$$\boldsymbol{f}: \mathbb{R}^n \longrightarrow \mathbb{R}^m,$$

mit

$$\boldsymbol{f}(x_1, ..., x_n) = \begin{pmatrix} y_1 = f_1(x_1, x_2, ..., x_n) \\ y_2 = f_2(x_1, x_2, ..., x_n) \\ \vdots \\ y_m = f_m(x_1, x_2, ..., x_n) \end{pmatrix},$$

wobei die m Komponenten $f_i: \mathbb{R}^n \longrightarrow \mathbb{R}$ $(i = 1, ..., m)$ von $\boldsymbol{f}$ wieder skalarwertige Funktionen sind, entsprechend Def. 5.1.

- Wie bei einem Vektor $\boldsymbol{x}$ stellen wir zur besseren Unterscheidbarkeit vektorwertige Funktionen $\boldsymbol{f}$ fett gedruckt dar, im Gegensatz zu einem Skalar x und einer skalarwertigen Funktion f.

- Wir werden uns bei der Lösung nichtlinearer Gleichungssysteme auf vektorwertige Funktionen $\boldsymbol{f}: \mathbb{R}^n \longrightarrow \mathbb{R}^n$ konzentrieren.

Beispiele 5.1:

1. Addition und Multiplikation:

   Wir können die Addition (bzw. Subtraktion) und die Multiplikation (bzw. Division) auffassen als skalarwertige Funktionen $f: \mathbb{R}^2 \longrightarrow \mathbb{R}$ mit

$$\begin{array}{rcl} f(x, y) & = & x + y \\ g(x, y) & = & x \cdot y \end{array}$$

2. Ohmsches Gesetz:

   Die an einem ohmschen Widerstand R abfallende Spannung U hängt vom Widerstand R und der Stromstärke I gemäss dem ohmschen Gesetz $U = R \cdot I$ ab. Also haben wir für die abhängige Variable $U = f(R, I) = RI$ die skalarwertige Funktion $f: \mathbb{R}^2 \longrightarrow \mathbb{R}$ mit den unabhängigen Variablen R und I. Häufig schreibt man auch direkt

$$U \ = \ U(R, I) = R \cdot I$$

   und bringt dadurch die Abhängigkeit der Variable U von den unabhängigen Variablen R und I zum Ausdruck, wie wir es auch bereits vom eindimensionalen Fall kennen, z.B. $y = y(x)$.

3. Reihenschaltung von Widerständen:

   Bei der Reihenschaltung von n ohmschen Widerständen $R_1, R_2, ..., R_n$ ergibt sich der Gesamtwiderstand R gemäss

$$R = R(R_1, R_2, ..., R_n) = R_1 + R_2 + ... + R_n$$

4. Vektorwertige Funktionen:

   Im einleitenden Beispiel in Kap. 5.1 haben wir bereits eine vektorwertige nichtlineare Funktion $\boldsymbol{f} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ kennengelernt. Ein weiteres Beispiel für $\boldsymbol{f} : \mathbb{R}^3 \longrightarrow \mathbb{R}^4$ ist

$$f(x_1, x_2, x_3) = \begin{pmatrix} x_1^2 + x_2^2 \\ x_1^2 + x_3^2 \\ x_2^2 + x_3^2 \\ x_1^2 + x_2^2 + x_3^2 \end{pmatrix},$$

Aufgabe 5.1:

1. Geben Sie je ein konkretes (nicht zu kompliziertes) Beispiel einer skalarwertigen Funktion $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$ sowie einer vektorwertigen Funktion $\boldsymbol{g} : \mathbb{R}^4 \longrightarrow \mathbb{R}^3$ an.

2. Geben sie die lineare Funktion $\boldsymbol{f} : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ an, für die die Lösung $\boldsymbol{x}$ des linearen Gleichungssystems

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \text{ mit } \boldsymbol{A} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \text{ und } \boldsymbol{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

   gerade $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ ergibt.

5.2.2 Darstellungsformen

Insbesondere bei der Interpretation von Resultaten einer Berechnung oder Messung ist eine geeignete Darstellungsform der Daten oft entscheidend für das eigene Verständnis und das Verständnis anderer. Während Funktionen mit einer unabhängigen Variablen noch einfach darstellbar sind, wird es bei Funktionen mehrerer Variablen schnell schwierig und unübersichtlich. Wir gehen hier auf die wichtigsten Darstellungsformen ein, ohne Anspruch auf Vollständigkeit.

5.2.2.1 Analytische Darstellung

Die Funktion liegt in Form einer Gleichung vor. Man unterscheidet:

- Explizite Darstellung: die Funktionsgleichung ist nach einer Variablen aufgelöst

$$y = f(x_1, x_2, ..., x_n)$$

  Beispiel: $y = 2 \cdot \mathrm{e}^{x_1^2 + x_2^2}$

- Implizite Darstellung: die Funktionsgleichung ist nicht nach einer Variablen aufgelöst (deshalb handelt es sich hier um eine Funktion mit nur $n-1$ unabhängigen Variablen ... warum?).

$$F(x_1, x_2, ..., x_n) = 0$$

  Beispiel: $x_1^2 + x_2^2 + x_3^2 - 1 = 0$

5.2.2.2 Darstellung durch Wertetabelle

Betrachten wir den Fall $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$. Setzt man in die (als bekannt) vorausgesetzte Funktionsgleichung $z = f(x, y)$ für die beiden unabhängigen Variablen x und y der Reihe nach bestimmte Werte ein, so erhält man eine Wertetabelle bzw. Matrix (siehe Abb. 5.2).

5.2.2.3 Grafische Darstellung

Wir beschränken uns hier auf skalarwertige Funktionen mit zwei unabhängigen Variablen $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$, für die es noch anschauliche grafische Darstellungsmöglichkeiten gibt. Dazu betrachten wir die Funktion $z = f(x, y)$ in einem dreidimensionalen kartesischen Koordinatensystem mit den Koordinatenachsen x, y, z.

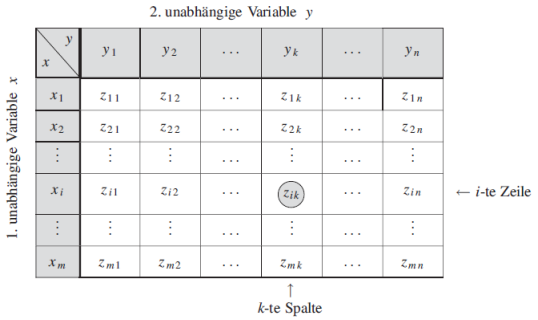


Abbildung 5.2: Wertetabelle für $z = f(x, y)$ für verschiedene Werte von x und y (aus [8]).

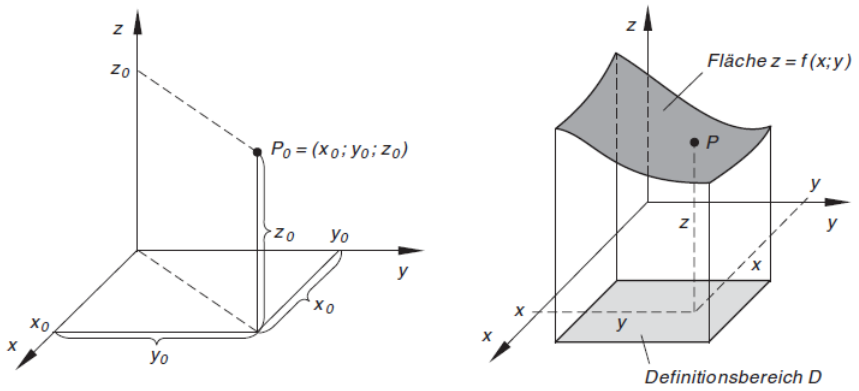


Abbildung 5.3: Links: kartesische Koordinaten eines Raumpunktes. Rechts: Darstellung einer Funktion $z = f(x, y)$ als Fläche im Raum (aus [8]).

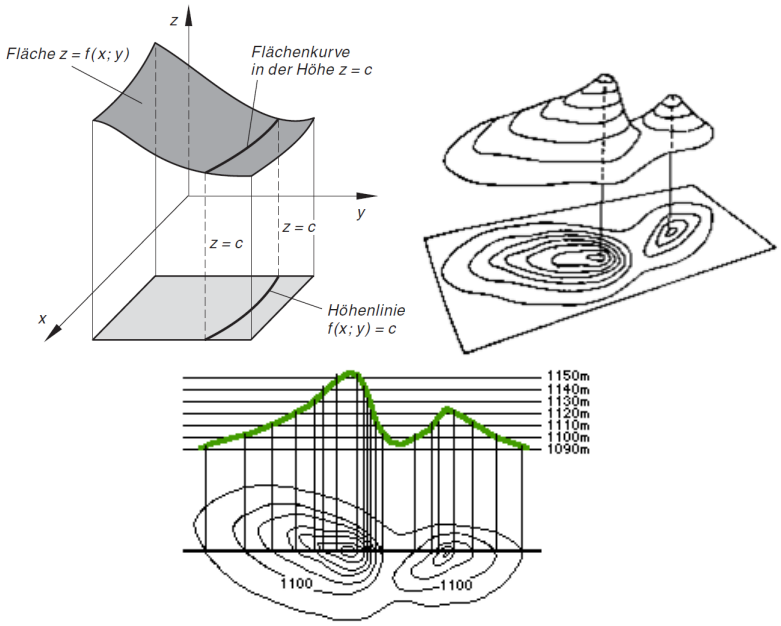


Abbildung 5.4: Zum Prinzip einer Höhenlinie (aus [8]) bzw. eines Höhenliniendiagramms.

- Darstellung einer Funktion als Fläche im Raum:

  Die Funktion f ordnet jedem Punkt $(x, y) \in D$ in der Ebene einen Wert $z = f(x, y)$ zu, der als Höhenkoordinate verstanden werden kann. Durch die Anordnung der Punkte $(x, y, f(x, y))$ im dreidimensionalen Koordinatensystem wird eine über dem Definitionsbereich D liegende Fläche ausgezeichnet (siehe Abb. 5.3).

- Schnittkurvendiagramm

  Wird die Fläche $z = f(x, y)$ bei einer konstanten Höhe $z = const.$ geschnitten, ergibt sich eine Schnittkurve. Wird diese in die $(x, y)-$Ebene projiziert, spricht man von einer Höhenlinie bzw. bei der Abbildung von einem Höhenliniendiagramm., wie wir es z.B. von Wanderkarten her kennen. Natürlich kann man auch andere Schnitte als $z = const.$ (Schnittebene parallel zur $(x, y)-$Ebene) wählen, z.B. $x = const.$ (Schnittebene parallel zur $(y, z)-$Ebene) oder $y = const.$ (Schnittebene parallel zur $(x, z)-$Ebene). Siehe Abb. 5.4.

5.2.3 Partielle Ableitungen

Für eine Funktion $f : \mathbb{R} \longrightarrow \mathbb{R}$ mit einer Variablen ist die Ableitung an der Stelle x_0 bekanntlich definiert als

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x},$$

aus geometrischer Sicht entspricht dies der Steigung $m = f'(x_0)$ der im Punkt $(x_0, f(x_0))$ angelegten Kurventangente t mit der Tangentengleichung

$$t(x) = f(x_0) + f'(x_0)(x - x_0)$$

Wir werden die Definition der Ableitung jetzt erweitern auf Funktionen mit mehreren unabhängigen Variablen. Die Idee dabei ist, jede unabhängige Variable einzeln zu betrachten und sämtliche anderen unabhängigen Variablen "einzufrieren" (d.h. als festgelegte Parameter zu behandeln). So kann man ein $n-$dimensionales Problem auf n eindimensionale Probleme reduzieren und die obige Definition der Ableitung verwenden.

Betrachten wir der Einfachheit halber eine Funktion mit zwei unabhängigen Variablen $z = f(x, y)$ und auf der dadurch definierten Fläche den Punkt P mit den Koordinaten (x_0, y_0, z_0), wobei $z_0 = f(x_0, y_0)$. Wir legen durch den Flächenpunkt P zwei Schnittebenen, die erste verläuft parallel zur $(x, z)-$Ebene, die zweite zur $(y, z)-$Ebene (Abb.5.5). Wir erhalten so durch den Punkt P zwei Schnittkurven K_1 und K_2 auf der Fläche $z = f(x, y)$. Dabei hängt die Funktionsgleichung von K_1 nur noch von der Variablen x ab (d.h. für K_1 gilt $z = f(x, y_0) =: g(x)$, denn y_0 ist fixiert). Analog hängt die Funktionsgleichung von K_2 nur noch von der Variablen y ab (d.h. für K_2 gilt $z = f(x_0, y) =: h(y)$, denn x_0 ist fixiert). Die beiden Schnittkurven sind ebenfalls in Abb.5.5 dargestellt.

Indem wir nun diese beiden Schnittkurven $g(x) = f(x, y_0)$ und $h(y) = f(x_0, y)$ gemäss unserer bisherigen Definition je einmal an der Stelle x_0 bzw. y_0 ableiten, erhalten wir die Steigung der Tangenten an die Fläche $z = f(x, y)$ im Punkt P, einmal in $x-$Richtung und einmal in $y-$Richtung. Konkret berechnen wir die beiden Grenzwerte:

$$\begin{aligned} g'(x_0) &= \lim_{\Delta x \to 0} \frac{g(x_0 + \Delta x) - g(x_0)}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x} =: \frac{\partial f}{\partial x}(x_0, y_0) \\ h'(y_0) &= \lim_{\Delta y \to 0} \frac{h(y_0 + \Delta y) - h(y_0)}{\Delta y} = \lim_{\Delta y \to 0} \frac{f(x_0, y_0 + \Delta y) - f(x_0, y_0)}{\Delta y} =: \frac{\partial f}{\partial y}(x_0, y_0) \end{aligned}$$

Wir bezeichnen diese Grenzwerte als partielle Ableitung 1. Ordnung von f an der Stelle (x_0, y_0).

Definition 5.2 [8]: Partielle Ableitungen 1. Ordnung
Unter den partiellen Ableitungen 1. Ordnung einer Funktion $z = f(x, y)$ and der Stelle (x, y) werden die folgenden Grenzwerte verstanden (falls sie vorhanden sind):

- Partielle Ableitung 1. Ordnung nach x:

$$\frac{\partial f}{\partial x}(x, y) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

- Partielle Ableitung 1. Ordnung nach y:

$$\frac{\partial f}{\partial y}(x, y) = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

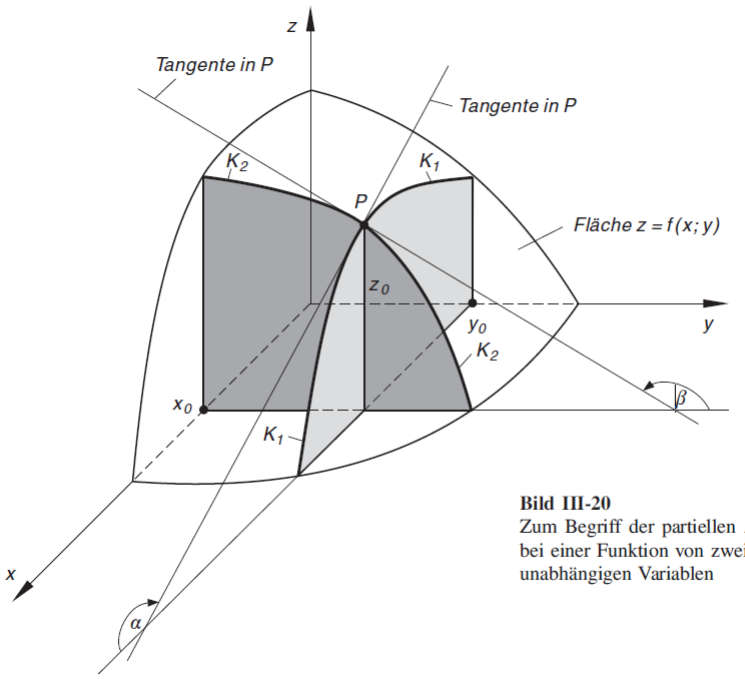
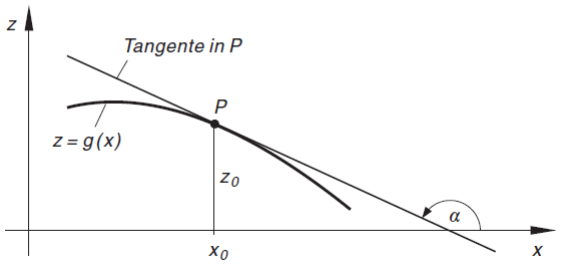


Bild III-20
Zum Begriff der partiellen Ableitung bei einer Funktion von zwei unabhängigen Variablen

Schnittkurve K_1: $z = f(x; y_0) = g(x)$



Schnittkurve K_2: $z = f(x_0; y) = h(y)$

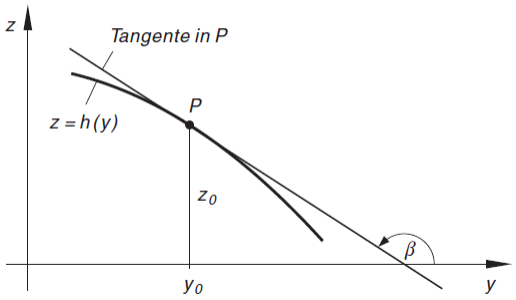


Abbildung 5.5: Fläche $z = f(x, y)$ mit dem Punkt $P = (x_0, y_0, z_0)$ und den Schnittflächen sowie den Schnittkurven K_1 und K_2 (aus [8]).

Bemerkungen:

1. Weitere übliche Symbole sind $f_x(x,y),\ f_y(x,y)$ oder in abgekürzter Schreibweise $f_x,\ f_y$ bzw. $\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y}$.

2. Geometrische Interpretation der partiellen Ableitungen der Funktion $z = f(x,y)$ an der Stelle (x_0, y_0):

   (a) $\frac{\partial f}{\partial x}(x_0,y_0)$ ist die Steigung der Flächentangente im Flächenpunkt $P = (x_0, y_0, z_0)$ in positiver $x-$Richtung

   (b) $\frac{\partial f}{\partial y}(x_0,y_0)$ ist die Steigung der Flächentangente im Flächenpunkt $P = (x_0, y_0, z_0)$ in positiver $y-$Richtung

3. Formal erhalten wir die partielle Ableitung $\frac{\partial f}{\partial x}(x,y)$, indem wir die Funktion $z = f(x,y)$ zunächst als eine nur von x abhängige Funktion betrachten und nach der Variablen x differenzieren. Während dieser Differentiation wird die Variable y als konstante Grösse (Parameter) betrachtet. Für das Differenzieren selbst gelten dann die bereits aus der Analysis bekannten Ableitungsregeln für Funktionen von einer unabhängigen Variablen.

   - Beispiel:

$$\begin{aligned} z &= f(x,y) = 3xy^3 + 10x^2y + 5y + 3y \cdot \sin(5xy) \\ \frac{\partial f}{\partial x}(x,y) &= f_x(x,y) = 3 \cdot 1 \cdot y^3 + 10 \cdot 2x \cdot y + 0 + 3y \cdot \cos(5xy) \cdot 5 \cdot 1 \cdot y \end{aligned}$$

4. Analog erhalten wir die partielle Ableitung $\frac{\partial f}{\partial y}(x,y)$, indem wir die Funktion $z = f(x,y)$ zunächst als eine nur von y abhängige Funktion betrachten und nach der Variablen y differenzieren. Während dieser Differentiation wird die Variable x als konstante Grösse (Parameter) betrachtet. Für das Differenzieren selbst gelten dann die bereits aus der Analysis bekannten Ableitungsregeln für Funktionen von einer unabhängigen Variablen.

   - Beispiel:

$$\begin{aligned} z &= f(x,y) = 3xy^3 + 10x^2y + 5y + 3y \cdot \sin(5xy) \\ \frac{\partial f}{\partial y}(x,y) &= f_y(x,y) = 3x \cdot 3y^2 + 10x^2 \cdot 1 + 5 \cdot 1 + (3 \cdot 1 \cdot \sin(5xy) + 3y \cdot \cos(5xy) \cdot 5x \cdot 1) \end{aligned}$$

5. Die partielle Differentiation wird somit auf die gewöhnliche Differentiation, d. h. auf die Differentiation einer Funktion von einer Variablen zurückgeführt. Die Ableitungsregeln sind daher die gleichen wie bei den Funktionen einer Variablen. So lautet beispielsweise die Produktregel bei zwei unabhängigen Variablen, d. h. für eine Funktion vom Typ $z = f(x,y) = u(x,y) \cdot v(x,y)$:

   - $f_x = u_x \cdot v + u \cdot v_x$
   - $f_y = u_y \cdot v + u \cdot v_y$

6. Für Funktionen mit mehr als zwei unabhängigen Variablen geht man analog vor. Sei $y = f(x_1, x_2, ..., x_n)$ eine Funktion mit n unabhängigen Variablen. Es lassen sich nun n partielle Ableitungen 1. Ordnung bilden gemäss

$$\frac{\partial f}{\partial x_k}(x_1, ..., x_k, ..., x_n) = \lim_{\Delta x_k \to 0} \frac{f(x_1, ..., x_k + \Delta x_k, ..., x_n) - f(x_1, ..., x_k, ..., x_n)}{\Delta x_k} \ (k = 1, ..., n)$$

   Dabei werden wieder alle anderen Variablen ausser x_k als konstante Grösse angenommen und es wird nach x_k abgeleitet.

Aufgabe 5.2: Berechnen Sie die partiellen Ableitungen erster Ordnung für die folgenden Funktionen

1. $z = f(x,y) = x^2y^4 + \mathrm{e}^x \cdot \cos y + 10x - 2y^2 + 3$
2. $z = f(x,y) = xy^2 \cdot (\sin x + \sin y)$
3. $z = f(x,y) = \ln(x + y^2) - \mathrm{e}^{2xy} + 3x$

   Lösungen:

   1. $f_x = 2xy^4 + \mathrm{e}^x \cdot \cos y + 10,\ f_y = 4x^2y^3 - \mathrm{e}^x \cdot \sin y - 4y$
   2. $f_x = y^2(\sin x + \sin y) + xy^2 \cdot \cos x,\ f_y = 2xy(\sin x + \sin y) + xy^2 \cdot \cos y$
   3. $f_x = \frac{1}{x+y^2} - 2y \cdot \mathrm{e}^{2xy} + 3,\ f_y = \frac{2y}{x+y^2} - 2x \cdot \mathrm{e}^{2xy}$

5.2.4 Linearisierung von Funktionen mit mehreren Variablen

Wieder ausgehend vom eindimensionalen Fall haben wir für die an eine Funktion $y = f(x)$ im Punkt $(x_0, f(x_0))$ angelegten Kurventangente g die Tangentengleichung

$$g(x) = f(x_0) + f'(x_0)(x - x_0)$$

und wir wissen, dass in einer Umgebung von x_0 die Funktion $y = f(x)$ durch die lineare Tangente angenähert ('linearisiert') werden kann, also $f(x) \approx f(x_0)+f'(x_0)(x-x_0)$ gilt. Nun wollen wir dies auf Funktionen mit mehreren Variablen erweitern und führen dafür die sogenannte Jacobi-Matrix $Df(x)$ ein, welche die einfache Ableitung $f'(x)$ ersetzen wird.

Definition 5.3: Jacobi-Matrix / Linearisierung / Tangentialebene

- Sei $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ mit $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} y_1 = f_1(\boldsymbol{x}) \\ y_2 = f_2(\boldsymbol{x}) \\ \vdots \\ y_m = f_m(\boldsymbol{x}) \end{pmatrix}$ und $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T \in \mathbb{R}^n$. Die **Jacobi-Matrix** enthält sämtliche partiellen Ableitung 1. Ordnung von $\boldsymbol{f}$ und ist definiert als

$$\boldsymbol{Df}(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \cdots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \cdots & \frac{\partial f_m}{\partial x_n}(x) \end{pmatrix}$$

- Die "verallgemeinerte Tangentengleichung"

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}^{(0)}) + \boldsymbol{Df}(\boldsymbol{x}^{(0)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(0)})$$

  beschreibt eine lineare Funktion und es gilt $\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{g}(\boldsymbol{x})$ in einer Umgebung eines gegebenen Vektors $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})^T \in \mathbb{R}^n$. Man spricht deshalb auch von der **Linearisierung** der Funktion $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ in einer Umgebung von $\boldsymbol{x}^{(0)}$ (ein hochgestellter Index in Klammern $\boldsymbol{x}^{(k)}$ bezeichnet wie bisher einen Vektor aus $\mathbb{R}^n$ nach der $k-$ten Iteration).

- Für den speziellen Fall $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ mit $y = f(x_1, x_2)$ liefert die linearisierte Funktion

$$g(x_1, x_2) = f(x_1^{(0)}, x_2^{(0)}) + \frac{\partial f}{\partial x_1}(x_1^{(0)}, x_2^{(0)}) \cdot \begin{pmatrix} x_1 - x_1^{(0)} \end{pmatrix} + \frac{\partial f}{\partial x_2}(x_1^{(0)}, x_2^{(0)}) \cdot \begin{pmatrix} x_2 - x_2^{(0)} \end{pmatrix}$$

  die Gleichung der **Tangentialebene**. Sie enthält sämtliche im Flächenpunkt $P = (x_1^{(0)}, x_2^{(0)}, f(x_1^{(0)}, x_2^{(0)}))$ an die Bildfläche von $y = f(x_1, x_2)$ angelegten Tangenten.

Beispiele 5.2:

1. Betrachten wir konkret nochmals die Funktion $\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2 - 11 \\ x_1 + x_2^2 - 7 \end{pmatrix}$ aus dem einführenden Beispiel von Kap. 5.1 und linearisieren sie in der Umgebung von $\boldsymbol{x}^{(0)} = (1, 1)^T$. Für die Jacobi-Matrix erhalten wir

$$\boldsymbol{Df}(x_1, x_2) = \begin{pmatrix} 2x_1 & 1 \\ 1 & 2x_2 \end{pmatrix}$$

   und an der Stelle $\boldsymbol{x}^{(0)} = (1, 1)^T$ gilt

$$\boldsymbol{Df}(x_1^{(0)}, x_2^{(0)}) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

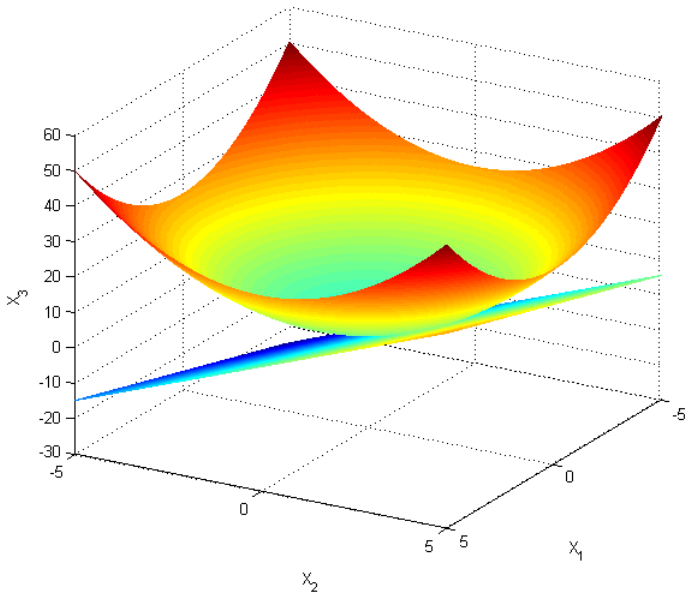


Abbildung 5.6: Grafische Darstellung der Fläche $x_3 = f(x_1, x_2) = x_1^2 + x_2^2$ sowie der Tangentialebene durch den Flächenpunkt (1,2,5) gemäss Bsp. 5.2.

Damit erhalten wir

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}) &= \boldsymbol{f}(\boldsymbol{x}^{(0)}) + \boldsymbol{Df}(\boldsymbol{x}^{(0)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(0)}) \\ \boldsymbol{g}(x_1, x_2) &= \begin{pmatrix} -9 \\ -5 \end{pmatrix} + \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix} \\ &= \begin{pmatrix} -9 + 2(x_1 - 1) + (x_2 - 1) \\ -5 + (x_1 - 1) + 2(x_2 - 1) \end{pmatrix} = \begin{pmatrix} 2x_1 + x_2 - 12 \\ x_1 + 2x_2 - 8 \end{pmatrix} \end{aligned}$$

2. Betrachten wir konkret die Funktion $f(x_1, x_2) = x_1^2 + x_2^2$. Wir linearisieren sie in der Umgebung von $\boldsymbol{x}^{(0)} = (1, 2)^T$. Wir erhalten für die Jacobi-Matrix

$$\boldsymbol{Df}(x_1, x_2) = \begin{pmatrix} 2x & 2y \end{pmatrix}$$

und damit

$$\begin{aligned} g(\boldsymbol{x}) &= f(\boldsymbol{x}^{(0)}) + \boldsymbol{Df}(\boldsymbol{x}^{(0)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(0)}) \\ g(x_1, x_2) &= 5 + \begin{pmatrix} 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 2 \end{pmatrix} \\ &= 5 + 2(x_1 - 1) + 4 \cdot (x_2 - 2) \\ &= 2x_1 + 4x_2 - 5. \end{aligned}$$

Dies ist nichts anderes als die Gleichung der Tangentialebene im kartesischen $(x_1, x_2, x_3)-$Koordinatensystem an die durch $x_3 = f(x_1, x_2) = x_1^2 + x_2^2$ definierte Fläche im Flächenpunkt $P = (1, 2, 5)$, wie in Abb. 5.6 dargestellt.

5.3 Problemstellung zur Nullstellenbestimmung für nichtlineare Systeme

Die allgemeine Problemstellung zur Nullstellenbestimmung für nichtlineare Gleichungssysteme lautet:

Definition 5.4 [1]:

- Gegeben sei $n \in \mathbb{N}$ und eine Funktion $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. Gesucht ist ein Vektor $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ mit $\boldsymbol{f}(\bar{\boldsymbol{x}}) = 0$.
- Komponentenweise bedeutet dies: Gegeben sind n Funktionen $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, die die Komponenten von $\boldsymbol{f}$ bilden. Gesucht ist ein Vektor $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ mit $f_i(\bar{\boldsymbol{x}}) = 0$ (für $i = 1, ..., n$). Dann heisst $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ eine Lösung des **Gleichungssystem**s

$$\boldsymbol{f}(x) = \boldsymbol{f}(x_1, ..., x_n) = \begin{pmatrix} f_1(x_1, ..., x_n) \\ f_2(x_1, ..., x_n) \\ \vdots \\ f_n(x_1, ..., x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Während es für lineare Gleichungssysteme relativ einfache Kriterien bezüglich der Lösbarkeit und der Anzahl von Lösungen gibt, ist diese Frage bei nichtlinearen Gleichungssystemen erheblich schwieriger zu beantworten, d.h. es gibt keine einfachen Methoden um festzustellen, ob ein nichtlineares Gleichungssystem lösbar ist und wie viele Lösungen es hat. Deshalb entscheidet die Wahl einer " geeigneten Startnäherung" meist über Erfolg oder Misserfolg der eingesetzten numerischen Verfahren.

Beispiel 5.3:

- Lösen Sie das folgende nichtlineare Gleichungssystem mit zwei Unbekannten:

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 8x_2^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Lösung: Auflösen der ersten Gleichung nach x_2 und einsetzen in die zweite Gleichung liefert die drei Lösungen $(0, 0)$, $(-2, 1)$, $(2, -1)$.

5.4 Das Newton-Verfahren für Systeme

Im Abschnitt 3.5 haben wir für die Nullstellenbestimmung einer Funktion $f : \mathbb{R} \longrightarrow \mathbb{R}$ mit einer Variablen das Newton-Verfahren hergeleitet. Aus der Linearisierung der Funktion f mittels der Tangente g in einer Umgebung der Stelle x_n

$$f(x) \approx g(x) = f(x_n) + f'(x_n)(x - x_n)$$

folgte die Iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (n = 0, 1, 2, 3, ...).$$

In Definition 5.3 haben wir die Jacobi-Matrix und die Linearisierung eingeführt. Für den für uns interessanten Fall von $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ lautet die Jacobi-Matrix

$$\boldsymbol{Df}(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \cdots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \frac{\partial f_n}{\partial x_2}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}$$

und durch Linearisierung erhalten wir

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x}^{(n)}) + \boldsymbol{Df}(\boldsymbol{x}^{(n)}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(n)}).$$

(wobei $\boldsymbol{x}^{(n)}$ wie üblich den Näherungs-Vektor für die Nullstelle $\boldsymbol{x}$ nach der n-ten Iteration beschreibt). Wenn der Vektor $\boldsymbol{x}^{(n+1)}$ eine Nullstelle von $\boldsymbol{f}$ ist, gilt:

$$\boldsymbol{f}(\boldsymbol{x}^{(n+1)}) = \mathbf{0} \approx \boldsymbol{f}(\boldsymbol{x}^{(n)}) + \boldsymbol{Df}(\boldsymbol{x}^{(n)}) \cdot (\boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)}).$$

Durch Auflösen nach $\boldsymbol{x}^{(n+1)}$ erhalten wir dann die Iterationsvorschrift

$$\boxed{\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \left(\boldsymbol{Df}(\boldsymbol{x}^{(n)})\right)^{-1} \cdot \boldsymbol{f}(\boldsymbol{x}^{(n)})}$$

wieder in Analogie zum eindimensionalen Fall

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Es wird aber nie die Inverse der Jacobi-Matrix berechnet, sondern die obige Gleichung wird zur Lösung eines linearen Gleichungssystems verwendet, indem man die Substitution

$$\boldsymbol{\delta}^{(n)} := -\left(\boldsymbol{Df}(\boldsymbol{x}^{(n)})\right)^{-1} \cdot \boldsymbol{f}(\boldsymbol{x}^{(n)})$$

als lineares Gleichungssystem auffasst gemäss

$$\boldsymbol{Df}(\boldsymbol{x}^{(n)})\boldsymbol{\delta}^{(n)} = -\boldsymbol{f}(\boldsymbol{x}^{(n)})$$

und so $\boldsymbol{\delta}^n$ bestimmen und anschliessend $\boldsymbol{x}^{(n+1)} := \boldsymbol{x}^{(n)} + \boldsymbol{\delta}^{(n)}$ berechnen kann.

5.4.1 Quadratisch-konvergentes Newton-Verfahren

Das obige Vorgehen führt zum folgenden Algorithmus:

Newton-Verfahren für Systeme [1]:
Gesucht sind Nullstellen von $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. Sei $\boldsymbol{x}^{(0)}$ ein Startvektor in der Nähe einer Nullstelle. Das Newton-Verfahren zur näherungsweisen Bestimmung dieser Nullstelle lautet:

- für $n = 0, 1, ...$:
  - Berechne $\boldsymbol{\delta}^{(n)}$ als Lösung des linearen Gleichungssystems
    $$\boldsymbol{Df}(\boldsymbol{x}^{(n)})\boldsymbol{\delta}^{(n)} = -\boldsymbol{f}(\boldsymbol{x}^{(n)})$$
  - Setze
    $$\boldsymbol{x}^{(n+1)} := \boldsymbol{x}^{(n)} + \boldsymbol{\delta}^{(n)}$$

Bemerkungen:

1. Mögliche Abbruchkriterien, $\epsilon > 0$ (gmäss [6]):
   (a) $n \geq n_{max}$, $n_{max} \in \mathbb{N}$
   (b) $\| \boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)} \| \leq \| \boldsymbol{x}^{(n+1)} \| \cdot \epsilon$
   (c) $\| \boldsymbol{x}^{(n+1)} - \boldsymbol{x}^{(n)} \| \leq \epsilon$
   (d) $\| \boldsymbol{f}\left(\boldsymbol{x}^{(n+1)}\right) \| \leq \epsilon$
2. Es kann passieren, dass mit dem Newton-Verfahren statt einer Nullstelle von $\boldsymbol{f}$ ein lokales Minimum $\boldsymbol{x}_{min}$ gefunden wird, das ungleich $\mathbf{0}$ ist. In diesem Falle ist $\boldsymbol{Df}(\boldsymbol{x}_{min})$ aber immer nicht regulär. Siehe untenstehendes Beispiel 5.5.

Beispiel 5.4 [1]:

- Wenden Sie das Newton-Verfahren auf das nichtlineare Gleichungssystem aus Beispiel 5.3 an:

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 8x_2^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Lösung: Für die Jacobi-Matrix erhalten wir

$$\boldsymbol{Df}(x_1, x_2) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1, x_2) & \frac{\partial f_1}{\partial x_2}(x_1, x_2) \\ \frac{\partial f_2}{\partial x_1}(x_1, x_2) & \frac{\partial f_2}{\partial x_2}(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 4 & 24x_2^2 \end{pmatrix}$$

Wir wählen den Startvektor $\boldsymbol{x}^{(0)} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$. Daraus ergibt sich für die erste Iteration das lineare Gleichungssystem

$$\boldsymbol{Df}(4,2)\delta^{(0)} = -\boldsymbol{f}(4,2) \Rightarrow \begin{pmatrix} 2 & 4 \\ 4 & 96 \end{pmatrix} \boldsymbol{\delta}^{(0)} = -\begin{pmatrix} 16 \\ 80 \end{pmatrix} \Rightarrow \boldsymbol{\delta}^{(0)} = -\begin{pmatrix} \frac{76}{11} \\ \frac{6}{11} \end{pmatrix}$$

und für den ersten Newton-Schritt

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \boldsymbol{\delta}^{(0)} = \begin{pmatrix} -\frac{32}{11} \\ \frac{16}{11} \end{pmatrix} = \begin{pmatrix} -2.909... \\ 1.4545... \end{pmatrix}.$$

Die weiteren Schritte sind

i	0	1	2	3	4
$\boldsymbol{x}^{(i)}$	$\begin{pmatrix} 4 \\ 2 \end{pmatrix}$	$\begin{pmatrix} -2.909 \\ 1.455 \end{pmatrix}$	$\begin{pmatrix} -2.302 \\ 1.151 \end{pmatrix}$	$\begin{pmatrix} -2.051 \\ 1.025 \end{pmatrix}$	$\begin{pmatrix} -2.0018 \\ 1.0009 \end{pmatrix}$

Die Folge konvergiert gegen $(-2, 1)^T$, damit haben wir eine der drei Nullstellen gefunden.

Aufgabe 5.3:

- Finden Sie für das obige Beispiel Startvektoren, so dass das Newton-Verfahren mit diesen Startvektoren gegen die beiden anderen Nullstellen von $\boldsymbol{f}$ konvergiert.

Man sieht, dass das Newton-Verfahren konvergiert, wenn der Startvektor nahe genug bei einer Nullstelle liegt. Allgemein gilt:

Satz 5.1: Quadratische Konvergenz des Newton-Verfahrens für Systeme[1]
Das Newton-Verfahren konvergiert quadratisch für nahe genug an einer Nullstelle $\overline{\boldsymbol{x}}$ liegende Startvektoren, wenn $\boldsymbol{Df}(\overline{\boldsymbol{x}})$ regulär und $\boldsymbol{f}$ dreimal stetig differenzierbar ist.

Aufgabe 5.4 [1]:

- Das nichtlineare System

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 5x_1^2 - x_2^2 \\ x_2 - 0.25(\sin x_1 + \cos x_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

besitzt in der Nähe von $\boldsymbol{x} = (0.25, 0.25)^T$ eine Lösung. Bestimmen Sie mit dem Newton–Verfahren eine Näherungslösung, die bezüglich der euklidischen Norm eine Genauigkeit von 10^{-5} besitzt.

Beispiel 5.5 [1]:

- Das Newtonverfahren für das nichtlineare System

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1^3 - x_2 - 1 \\ x_1^2 - x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

konvergiert für den Startvektor $\boldsymbol{x}^{(0)} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ gegen $\boldsymbol{x} = \begin{pmatrix} 0 \\ -0.5 \end{pmatrix}$. Da aber $\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix} \neq 0$ ist das keine Nullstelle. Man sieht entsprechend, dass $\boldsymbol{Df}(\boldsymbol{x}) = \begin{pmatrix} 0 & -1 \\ 0 & -1 \end{pmatrix}$ nicht regulär ist.

5.4.2 Vereinfachtes Newton-Verfahren

Der Aufwand pro Schritt kann reduziert werden, wenn nicht bei jedem Schritt die Jacobi-Matrix $\boldsymbol{Df}(x^{(n)})$ ausgewertet, sondern immer wieder $\boldsymbol{Df}(x^{(0)})$ verwendet wird. Dies ist in Analogie zu Abschnitt 3.5.1 das vereinfachte Newtonverfahren:

Vereinfachtes Newton-Verfahren für Systeme [1]:
Gesucht sind Nullstellen von $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. Sei $\boldsymbol{x}^{(0)}$ ein Startvektor in der Nähe einer Nullstelle. Das vereinfachte Newton-Verfahren zur näherungsweisen Bestimmung dieser Nullstelle lautet:

- für $n = 0, 1, ...$:
  - Berechne $\boldsymbol{\delta}^{(n)}$ als Lösung des linearen Gleichungssystems

$$\boldsymbol{Df}(\boldsymbol{x}^{(0)})\boldsymbol{\delta}^{(n)} = -\boldsymbol{f}(\boldsymbol{x}^{(n)})$$

  - Setze

$$\boldsymbol{x}^{(n+1)} := \boldsymbol{x}^{(n)} + \boldsymbol{\delta}^{(n)}$$

Bemerkung:

- Das vereinfachte Newton-Verfahren konvergiert nur noch linear und nicht mehr quadratisch.

Beispiel 5.6 [1]:

Wenden Sie das vereinfachte Newton-Verfahren auf das nichtlineare Gleichungssystem aus Beispiel 5.3 an:

$$\boldsymbol{f}(x_1, x_2) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + 8x_2^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Lösung: Wir wählen als Startvektor wieder $\boldsymbol{x}^{(0)} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$. Der erste Schritt des vereinfachten Newton-Verfahrens ist identisch mit dem ersten Schritt des Newton-Verfahrens. Im zweiten Schritt verwenden wir erneut die Jacobi-Matrix aus dem ersten Schritt; es ist also zu lösen

$$\boldsymbol{Df}(\boldsymbol{x}^{(0)})\,\boldsymbol{\delta}^{(1)} = \boldsymbol{Df}(4,2)\,\boldsymbol{\delta}^{(1)} = -\boldsymbol{f}(\boldsymbol{x}^{(1)}) = -\boldsymbol{f}(-2.09, 1.45)$$
$$\iff \begin{pmatrix} 2 & 4 \\ 4 & 96 \end{pmatrix} \boldsymbol{\delta}^{(1)} = -\begin{pmatrix} 6 \cdot 10^{-9} \\ -12.98 \end{pmatrix} \iff \boldsymbol{\delta}^{(1)} = \begin{pmatrix} 0.2951 \\ -0.1475 \end{pmatrix}$$

Damit haben wir nach einem Newton-Schritt

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} + \boldsymbol{\delta}^{(1)} = \begin{pmatrix} -2.614 \\ 1.307 \end{pmatrix}$$

Einige weitere Iterierte:

i	0	1	2	5	10
$\boldsymbol{x}^{(i)}$	$\begin{pmatrix} 4 \\ 2 \end{pmatrix}$	$\begin{pmatrix} -2.909 \\ 1.455 \end{pmatrix}$	$\begin{pmatrix} -2.614 \\ 1.307 \end{pmatrix}$	$\begin{pmatrix} -2.258 \\ 1.129 \end{pmatrix}$	$\begin{pmatrix} -2.0817 \\ 1.041 \end{pmatrix}$

Offensichtlich konvergiert die Folge gegen $(-2, 1)^{\mathrm{T}}$, jedoch deutlich langsamer als das Newton-Verfahren. ∎

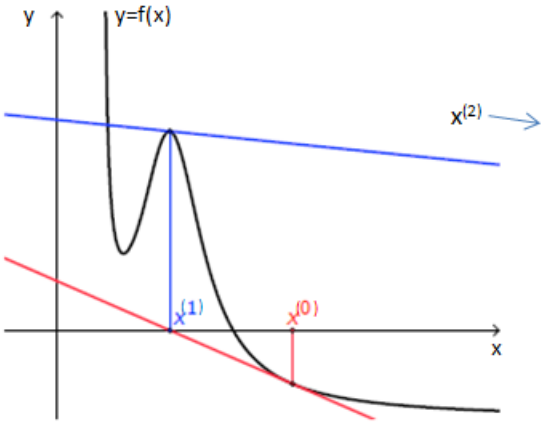


Abbildung 5.7: Eindimensionales Beispiel, in dem $x^{(1)}$ fast auf ein lokales Maximum zu liegen kommt und deshalb $f'(x^{(1)})$ beliebig klein bzw. $\left(f'(x^{(1)})\right)^{-1}$ beliebig gross wird. Die Iteration $x^{(2)}$ 'reisst' demzufolge aus und ist keine bessere Näherung für die Nullstelle von $f(x)$ als $x^{(1)}$.

5.4.3 Gedämpftes Newton-Verfahren

Falls beim $n-$ten Iterationsschritt die Jacobi-Matrix $\boldsymbol{Df}(x^{(n)})$ schlecht konditioniert (bzw. nicht oder fast nicht invertierbar) ist, kann wegen

$$\boldsymbol{\delta}^{(n)} := -\left(\boldsymbol{Df}(\boldsymbol{x}^{(n)})\right)^{-1} \cdot \boldsymbol{f}(\boldsymbol{x}^{(n)})$$

nicht generell erwartet werden, dass

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \boldsymbol{\delta}^{(n)}$$

eine bessere Näherung für die Nullstelle darstellt als $\boldsymbol{x}^{(n)}$. Unter Umständen entfernt sich in diesem Fall $\boldsymbol{x}^{(n+1)}$ sogar sehr weit von der eigentlichen Nullstelle, wie in Abb. 5.7 für einen eindimensionale Fall gezeigt ist. Falls dies der Fall ist, macht es Sinn,

$$\boldsymbol{x}^{(n)} + \boldsymbol{\delta}^{(n)}$$

zu verwerfen und es beispielsweise mit

$$\boldsymbol{x}^{(n)} + \frac{\boldsymbol{\delta}^{(n)}}{2}$$

zu probieren (d.h. wir verkleinern bzw. dämpfen die Schrittweite $\boldsymbol{\delta}^{(n)}$) und diesen Wert zu akzeptieren, sofern für die Länge des Vektors

$$\| \boldsymbol{f}\left(\boldsymbol{x}^{(n)} + \frac{\boldsymbol{\delta}^{(n)}}{2}\right) \|_2 < \| \boldsymbol{f}\left(\boldsymbol{x}^{(n)}\right) \|_2$$

gilt, da wir ja eine Iteration von $\| \boldsymbol{f}\left(\boldsymbol{x}^{(n)}\right) \|_2$ gegen $\mathbf{0}$ erreichen wollen. Das heisst, wir akzeptieren einen Iterationsschritt erst, wenn gilt

$$\| \boldsymbol{f}\left(\boldsymbol{x}^{(n+1)}\right) \|_2 < \| \boldsymbol{f}\left(\boldsymbol{x}^{(n)}\right) \|_2$$

Dies beschreibt der folgende Algorithmus:

Gedämpftes Newton-Verfahren für Systeme [6]:
Gesucht sind Nullstellen von $\boldsymbol{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. Sei $\boldsymbol{x}^{(0)}$ ein Startvektor in der Nähe einer Nullstelle, $k_{max} \in \mathbb{N}$ sei vorgegeben. Das gedämpfte Newton-Verfahren zur näherungsweisen Bestimmung dieser Nullstelle lautet:

- für $n = 0, 1, \ldots$:
  - Berechne $\boldsymbol{\delta}^{(n)}$ als Lösung des linearen Gleichungssystems

$$\boldsymbol{Df}(\boldsymbol{x}^{(n)})\boldsymbol{\delta}^{(n)} = -\boldsymbol{f}(\boldsymbol{x}^{(n)})$$

– Finde das minimale $k \in \{0, 1, ..., k_{max}\}$ mit

$$\| \boldsymbol{f}\left(\boldsymbol{x}^{(n)} + \frac{\boldsymbol{\delta}^{(n)}}{2^k}\right) \|_2 < \| \boldsymbol{f}\left(\boldsymbol{x}^{(n)}\right) \|_2$$

– Falls kein minimales k gefunden werden kann, rechne mit $k = 0$ weiter

– Setze

$$\boldsymbol{x}^{(n+1)} := \boldsymbol{x}^{(n)} + \frac{\boldsymbol{\delta}^{(n)}}{2^k}$$

Bemerkungen [6]:

1. Natürlich kann man auch für das vereinfachte Newton-Verfahren die analoge Dämpfung einbauen.
2. Umfangreiche Tests haben ergeben, dass das gedämpfte Newton-Verfahren im Allgemeinen weit besser ist als das normale Newton-Verfahren oder andere, hier nicht behandelte Verfahren wie das Gradientenverfahren.
3. Die Dämpfungsgrösse k_{max} ist stark vom jeweiligen Problem abhängig. Das Verfahren kann bei gleichem Startvektor bei verschiedener Vorgabe von k_{max}einmal konvergieren und einmal divergieren; es kann insbesondere für verschiedene k_{max} auch gegen verschiedene Nullstellen konvergieren. Sofern nichts über sinnvolle Werte bekannt ist, kann zunächst mit $k_{max} = 4$ gerechnet werden.

Aufgabe 5.5 (in den Übungen):

- Der Druck, der benötigt wird, damit ein grosser, schwerer Gegenstand in einem weichen, auf einem harten Untergrund liegenden, homogenen Boden absinkt, kann über den Druck vorhergesagt werden, der zum Absinken kleinerer Gegenstände in demselben Boden benötigt wird. Speziell der Druck p, der benötigt wird, damit ein runder flacher Gegenstand vom Radius r um d cm tief in den weichen Boden sinkt, kann über eine Gleichung der Form

  $$p = k_1 e^{k_2 r} + k_3 r$$

  approximiert werden, wobei k_1, k_2 und k_3 Konstanten mit $k_2 > 0$ sind, die von d und der Konsistenz des Bodens, aber nicht vom Radius des Gegenstandes abhängen. Der harte Untergrund liege in einer Entfernung $D > d$ unter der Oberfläche.

  a) Bestimmen Sie die Werte von k_1, k_2 und k_3, falls angenommen wird, dass ein Gegenstand vom Radius 1 cm einen Druck von 10 N/cm^2 benötigt, um 30 cm tief in einen schlammigen Boden zu sinken, ein Gegenstand vom Radius 2 cm einen Druck von 12 N/cm^2 benötigt, um 30 cm tief zu sinken und ein Gegenstand vom Radius 3 cm einen Druck von 15 N/cm^2 benötigt, um ebensoweit abzusinken (vorausgesetzt, der Schlamm ist tiefer als 30 cm). Benutzen Sie den Startvektor

  $$\boldsymbol{k}^{(0)} = \begin{pmatrix} 10 \\ 0.1 \\ -1 \end{pmatrix}$$

  .

  b) Sagen Sie aufgrund Ihrer Berechnungen aus Übung a) die minimale Grösse eines runden Gegenstandes voraus, der eine Belastung von 500 N aushält und dabei weniger als 30 cm tief sinkt.