

# Schnelle Lösung großer Gleichungssysteme

Anton Schüller<sup>1</sup>  
Ulrich Trottenberg<sup>1,2</sup>  
Roman Wienands<sup>2</sup>

<sup>1</sup> Fraunhofer-Institut  
Algorithmen und Wissenschaftliches Rechnen SCAI

<sup>2</sup> Mathematisches Institut  
der Universität zu Köln

Version 1.1  
28.09.2012

# Inhaltsverzeichnis

<b>Vorbemerkung: Große lineare Gleichungssysteme im Schulunterricht</b>	<b>5</b>
<b>Überblick</b>	<b>5</b>
<b>1 Große Gleichungssysteme</b>	<b>7</b>
1.1 Anwendungsbeispiele, die auf große Gleichungssysteme führen . . . . .	7
1.2 Fortschritt durch Mathematik und schnelle Rechner bei der Lösung großer Gleichungssysteme . . . . .	9
1.3 Schnelle und weniger schnelle Verfahren . . . . .	10
1.4 Darstellung großer linearer Gleichungssysteme . . . . .	10
<b>2 Direkte Verfahren zur Lösung linearer Gleichungssysteme</b>	<b>12</b>
2.1 Einführung . . . . .	12
2.1.1 Cramersche Regel und Determinanten . . . . .	12
2.1.2 Rundungsfehler . . . . .	18
2.2 Idee des Gaußschen Eliminationsverfahrens . . . . .	19
2.2.1 Zielsystem: Matrix in Dreiecksgestalt . . . . .	19
2.2.2 Der erste Eliminationsschritt . . . . .	20
2.2.3 Fortführung des Verfahrens . . . . .	21
2.2.4 Nachteile des Gaußschen Eliminationsverfahrens . . . . .	22
<b>3 Iterative Lösung von linearen Gleichungssystemen</b>	<b>23</b>
3.1 Jacobi-Verfahren und Gauss-Seidel-Verfahren . . . . .	23
3.1.1 Explizite Formeln für Jacobi- und Gauss-Seidel-Verfahren . . . . .	23
3.1.2 Zwei konkrete Beispiele . . . . .	25

<i>Inhaltsverzeichnis</i>	3
3.2 Das SOR-Verfahren . . . . .	26
3.3 Zur Konvergenz iterativer Verfahren . . . . .	27
3.3.1 Zur Konvergenz des Jacobi-Verfahrens, starkes Zeilensummenkriterium . . . . .	27
3.3.2 Ein Beispiel . . . . .	27
3.3.3 Zusammenhang zum Gauss-Seidel-Verfahren . . . . .	28
<b>4 Ableitungen von Funktionen, Differentialgleichungen, Randwertaufgaben</b>	<b>29</b>
4.1 Ableitung einer Funktion . . . . .	29
4.1.1 Berechnung von Ableitungen, Differenzenquotienten . . . . .	30
4.1.2 Beispiele für Ableitungen von Funktionen . . . . .	31
4.1.3 Einfache Rechenregeln für Ableitungen . . . . .	32
4.2 Gewöhnliche Differentialgleichungen . . . . .	32
4.3 Funktionen mehrerer Veränderlicher, partielle Ableitungen, partielle Differentialgleichungen . . . . .	33
4.4 Diskretisierung (gewöhnlicher und partieller) Randwertaufgaben mit Differenzenverfahren . . . . .	34
4.4.1 Eine gewöhnliche Randwertaufgabe 2. Ordnung . . . . .	34
4.4.2 Diskretisierung . . . . .	35
4.4.3 Approximation der Differentialgleichung . . . . .	35
4.4.4 Resultierendes Gleichungssystem . . . . .	36
4.4.5 Matrixdarstellung . . . . .	37
4.4.6 Konvergenz einfacher iterativer Verfahren zur Lösung des GLS . . .	37
4.4.7 Die Poisson-Gleichung mit Dirichletschen Randbedingungen . . . .	38
4.4.8 Diskretisierung . . . . .	38
4.4.9 Approximation von $\Delta u$ . . . . .	39
4.4.10 Aufstellung des zugehörigen Gleichungssystems . . . . .	39
4.4.11 Diskretisierungsgenauigkeit . . . . .	40
4.4.12 Matrixdarstellung . . . . .	41
4.4.13 Matrixdarstellung bei zeilenweiser Nummerierung . . . . .	41
4.4.14 Interpretation als Bandmatrix . . . . .	41

<i>Inhaltsverzeichnis</i>	4
4.4.15 Einfache Verfahren zur Lösung des diskreten linearen Gleichungssystems . . . . .	42
<b>5 Unterrichtsmaterialien</b>	<b>43</b>
5.1 Foliensätze . . . . .	43
5.1.1 Foliensatz zu Anwendungen großer Gleichungssysteme und zur Motivation . . . . .	43
5.1.2 Foliensatz zur Einführung in das Lösen großer Gleichungssysteme .	43
5.2 Computerprogramme . . . . .	44
5.2.1 Computerprogramm: Rechenzeiten bei Determinantenberechnung nach dem Laplaceschen Entwicklungssatz . . . . .	44
5.2.2 Computerprogramm: Rechenzeiten bei Determinantenberechnung durch Transformation auf Dreiecksgestalt . . . . .	45
5.2.3 Computerprogramm zur Determinantenberechnung durch Transformation auf Dreiecksgestalt . . . . .	46
5.2.4 Computerprogramm zum Rechenzeitverhalten des Gaußschen Eliminationsverfahrens bei der Lösung linearer Gleichungssysteme .	46
5.2.5 Computerprogramm: Gaußsches Eliminationsverfahren zur Lösung eines linearen Gleichungssystems . . . . .	46
5.2.6 Computerprogramm: Gauss-Seidel-Verfahren zur Lösung (diagonal-dominanter) linearer Gleichungssysteme . . . . .	47
5.2.7 Computerprogramm: Gauss-Seidel- und SOR-Verfahren zur numerischen Lösung der zweidimensionalen Poissonschen Randwertaufgabe . . . . .	47
5.2.8 Computerprogramm: Full-Multigrid-Verfahren zur numerischen Lösung der zweidimensionalen Poissonschen Randwertaufgabe . . .	48
5.3 Arbeitsblätter . . . . .	48
Arbeitsblatt: Große Gleichungssysteme und Wettervorhersage . . . . .	49
Arbeitsblatt: Fortschritte durch Mathematik und Rechnerentwicklung in den letzten 30 Jahren . . . . .	51
Arbeitsblatt: Gaußsches Eliminationsverfahren . . . . .	53
Arbeitsblatt: Jacobi-Verfahren . . . . .	56
Arbeitsblatt: Gauss-Seidel-Verfahren . . . . .	59
Arbeitsblatt: Defekt einer Näherungslösung . . . . .	62
Arbeitsblatt: Konvergenzeigenschaften des Jacobi- und des Gauss-Seidel-Verfahrens . . . . .	64

## Vorbemerkung: Große lineare Gleichungssysteme im Schulunterricht

Große Gleichungssysteme kommen in der Praxis sehr häufig vor. Kontinuierliche Aufgaben (z.B. Differential- und Integralgleichungen) werden durch “Diskretisierung” in große Gleichungssysteme mit Tausenden, ja Millionen von Unbekannten überführt, um Probleme aus Meteorologie (Wetter- und Klimavorhersage), Strömungsmechanik (z.B. Entwurf von Autos oder Flugzeugen mit möglichst niedrigem Luftwiderstand und Treibstoffverbrauch), Strukturmechanik und vielen anderen Anwendungsfeldern zu lösen.

In der Schule bleiben die betrachteten Gleichungssysteme deutlich kleiner, und es wird häufig nicht deutlich, für welche Anwendungsfelder große Gleichungssysteme in der Praxis überhaupt eine Rolle spielen und warum die Verfügbarkeit effizienter Lösungsverfahren für derartige Problemstellungen von großer praktischer Bedeutung ist.

Auch ist allgemein kaum bekannt, dass die Fortschritte innerhalb der Angewandten Mathematik Hand in Hand mit den immer schneller gewordenen Rechnern heute die Lösung von Problemstellungen ermöglichen, die noch vor 30 Jahren als kaum lösbar erschienen. Hierzu hat insbesondere auch die Entwicklung schneller Verfahren zur Lösung großer Gleichungssysteme maßgeblich beigetragen.

Auf derartige Anwendungen und Entwicklungen gehen wir gleich zu Beginn des vorliegenden Unterrichtsmoduls in den Abschnitten 1.1–1.3 ein. Dabei vergleichen wir auch verschiedene unterschiedlich effiziente Verfahren anhand ihrer Rechenzeiten für ein typisches Beispiel.

Das beigelegte Unterrichtsmaterial eignet sich für eine Unterrichtsreihe in der Jahrgangsstufe 11 (G9) bzw. 10 (G8). Für die Diskretisierung von Differentialgleichungen (mit zugehörigen Randwerten) wird die Definition der Ableitung über den Differenzenquotienten verwendet. Die Anteile, die sich auf die Kapitel 1 und 2 beziehen, können aber teilweise auch bereits wesentlich früher eingesetzt werden, z.B. im Zusammenhang mit der Lösung von Gleichungssystemen in Klasse 8, im Differenzierungsbereich Mathematik/Naturwissenschaft der Mittelstufe (ab Klasse 8) oder im Rahmen einer Projektwoche ab Klasse 8. Das gesamte Material kann auch als Einführung in die Lösung großer Gleichungssysteme in einem Projektkurs der Oberstufe eingesetzt werden. In diesem Fall könnten die Schüler sich die Inhalte selbstständig mit diesem Skript erarbeiten.

Die Entwicklung dieses Unterrichtsmoduls wurde unterstützt durch eine Projektförderung durch die WestLB-Stiftung Zukunft NRW, für die wir uns ganz herzlich bedanken.

# Überblick

In Abschnitt 1.1 diskutieren wir exemplarisch einige Anwendungen, die auf die Lösung großer Gleichungssysteme führen. Abschnitt 1.2 zeigt am Beispiel der Lösung eines großen Gleichungssystems, welche Fortschritte in den vergangenen 30 Jahren durch neue Methoden der Angewandten Mathematik einerseits und durch die Entwicklung immer leistungsfähigerer Rechner andererseits erzielt werden konnten. In Abschnitt 1.4 beschreiben wir, wie wir große lineare Gleichungssysteme darstellen. Dabei werden auch die Begriffe Matrix und Vektor eingeführt.

Kapitel 2 beschäftigt sich mit direkten Verfahren zur Lösung linearer Gleichungssysteme. Zunächst wird am Beispiel der Cramerschen Regel gezeigt, dass explizite Lösungsformeln nicht immer sinnvoll sind, weil der Rechenaufwand zu ihrer Auswertung mit steigender Anzahl von Unbekannten viel zu schnell wachsen kann. Außerdem gehen wir in einem kurzen Exkurs auch auf die mögliche Problematik von Rundungsfehlern ein. Der Rest dieses Kapitels ist dem Gaußschen Eliminationsverfahren gewidmet. Seine Idee und die einzelnen Schritte werden ausführlich dargestellt. Vor- und Nachteile dieses Verfahrens werden diskutiert.

Kapitel 3 beschäftigt sich mit einfachen iterativen Lösungsverfahren. Jacobi- und Gauss-Seidel-Verfahren werden an Beispielen eingeführt. Hierbei wird vorausgesetzt, dass der Begriff eines iterativen Verfahrens bereits bekannt ist. Als Einführung in die Idee eines iterativen Verfahrens bietet sich etwa das Heronsche Verfahren zur Wurzelberechnung an.

Eine einfache Erweiterung des Gauss-Seidel-Verfahrens führt zum SOR-Verfahren. Abschließend gehen wir auf die Konvergenzeigenschaften des Jacobi- und des Gauss-Seidel-Verfahrens ein und diskutieren das starke Zeilensummenkriterium im Detail.

Das Auftreten großer Gleichungssysteme in Anwendungen unterschiedlichster Art hängt sehr häufig zusammen mit der numerischen Lösung von partiellen Differentialgleichungen mit geeigneten Anfangs- oder Randbedingungen. Um diesen Zusammenhang deutlich zu machen, führen wir in Kapitel 4 die Begriffe der Ableitung einer Funktion einer oder mehrerer Veränderlicher sowie der gewöhnlichen und partiellen Differentialgleichung kurz und anhand einiger Beispiele ein. Diese Einführung ist bewusst kurz gehalten und hat nicht die Absicht, eine auch nur halbwegs vollständige Einführung in die Theorie zu geben.

Der Übergang von einer Randwertaufgabe, bestehend aus einer (elliptischen) Differentialgleichung und einfachen Randbedingungen, in ein großes Gleichungssystem wird dagegen anhand eines ein- und eines zwei-dimensionalen Beispiels detailliert beschrieben. Die Idee dabei ist, Ableitungen wieder durch Differenzenquotienten zu ersetzen, dabei aber auf den Grenzübergang  $h \rightarrow 0$  zu verzichten. Die in Kapitel 3 eingeführten iterativen Lösungsverfahren können zur Lösung der so entstehenden Gleichungssysteme direkt eingesetzt werden.

Kapitel 5 enthält eine Sammlung/Beschreibung verschiedener Materialien, die für den Einsatz im Unterricht oder für das Selbststudium hilfreich sind. Hierbei handelt es sich einerseits um eine Reihe von Computerprogrammen, welche die im Skript eingeführten (und weitere) Lösungsverfahren realisieren und unter verschiedenen Aspekten beleuchten (vgl. Abschnitt 5.2). Hinzu kommen einige Arbeitsblätter samt entsprechender Musterlösungen (vgl. Abschnitt 5.3).

# Kapitel 1

## Große Gleichungssysteme

### 1.1 Anwendungsbeispiele, die auf große Gleichungssysteme führen

1. **Wetterprognose:** Das mathematische Modell, das der globalen Wetterprognose zugrundeliegt, ist ein System von partiellen Differentialgleichungen zur Bestimmung des Wetter charakterisierender Funktionen, z.B.

- Windgeschwindigkeit (Vektor aus 3 Komponenten)
- Druck
- Feuchtigkeit
- Temperatur

Für die tägliche Wettervorhersage muss dieses System, in das eine Vielzahl von Messwerten als Startwerte eingehen, mit Hilfe numerischer Verfahren gelöst werden. Hierzu wird ein Gitter durch den Raum, der die Erde umgibt, gelegt. In den Gitterpunkten werden dann Näherungswerte für die Lösungsfunktionen berechnet. Bei dem heute benutzten Wettermodell besteht das Gitter in vertikaler Richtung aus 40 parallelen Kugelschalen. In jeder dieser Kugelschalen haben die Gitterpunkte Abstände, die in Äquatornähe etwa 30 km betragen. Dies ergibt ungefähr 16 Millionen Gitterpunkte des räumlichen Gitters. Ähnlich wie der Raum durch ein Gitter, wird die Zeit in Zeitschritte unterteilt. Für eine 10 Tage Prognose braucht man rund 6500 Zeitschritte. In jedem Zeitschritt wird u.a. ein lineares Gleichungssystem mit 16 Millionen Unbekannten (für den Druck in den räumlichen Gitterpunkten) gelöst. Um regelmäßig aktuelle Vorhersagen machen zu können, stehen für die Lösung dieses Problems nur wenige Stunden zur Verfügung. Deshalb müssen extrem schnelle Lösungsverfahren eingesetzt werden.

2. **Klimaprognosen:** In Klimaprognosen beträgt ein typischer Vorhersagezeitraum nicht nur 10 Tage wie bei der Wettervorhersage, sondern z.B. 50 oder 100 Jahre. Das Modell für die Simulation der Atmosphäre ist dem der Wettervorhersage sehr ähnlich. Hinzu kommen bei Klimasimulationen aber auch noch detaillierte Modelle zur Simulation der Ozeane. Da erheblich längere Zeiträume zu berechnen sind,

verwendet man bei Klimasimulationen wesentlich größere Gitter mit erheblich weniger Gitterpunkten als bei der Wettervorhersage (Abstand zweier Gitterpunkte in Äquaturnähe: mehrere 100 km). Trotzdem beträgt die Rechenzeit selbst auf sehr schnellen Rechnern mehrere Monate.

Die Vorhersagen über das Klima in den kommenden 100 Jahren schwanken um bis zu etwa  $3^{\circ}\text{C}$ . Dabei sind die tatsächlichen Fehlerquellen für diese relativ großen Abweichungen bis heute nicht genau bekannt. Zwei mögliche Fehlerquellen sind die großen Abstände zwischen den Gitterpunkten bzw. zu ungenaue physikalische/meteorologische Modelle.

3. **Numerischer Windkanal:** Hiermit wird ein vollständiger, also dreidimensionaler (3D) Entwurf von Großraumflugzeugen auf Rechnern durchgeführt. Die dabei benötigten 3D-Gitter bestehen heute aus vielen Millionen Gitterpunkten. In jedem Gitterpunkt werden die Werte von 5 unbekannten Funktionen (Geschwindigkeiten in  $x$ -,  $y$ - und  $z$ -Richtung, Druck, Energie) berechnet.

Weitere Anwendungsbeispiele, bei denen große Gleichungssysteme auftreten, die in möglichst kurzer Zeit mit Hilfe effizienter mathematischer Methoden auf Computern gelöst werden, zeigt Abb. 1.

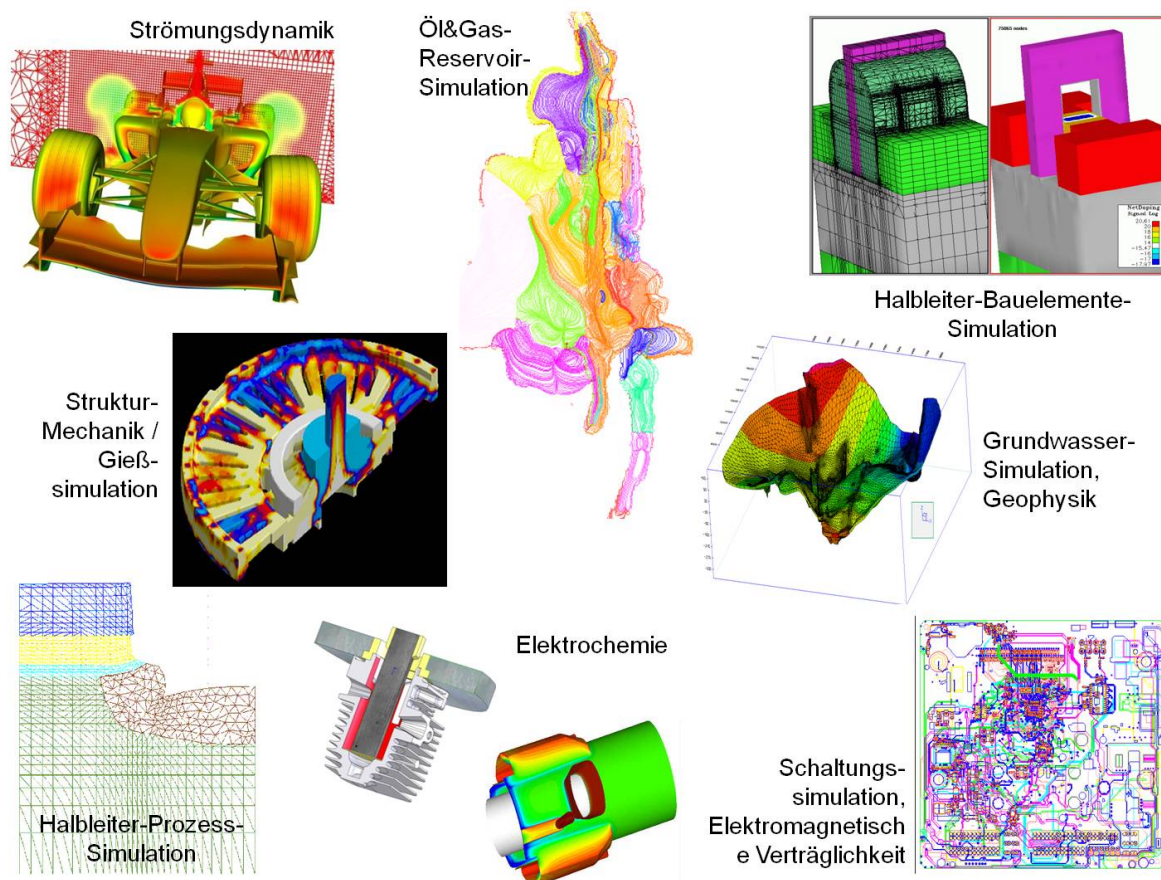


Abbildung 1: Einige weitere Anwendungsfelder, in denen große Gleichungssysteme auftreten, die auf effiziente Art und Weise gelöst werden müssen.



## 1.2 Fortschritt durch Mathematik und schnelle Rechner bei der Lösung großer Gleichungssysteme

Abb. 2 zeigt die Fortschritte, die in den letzten 30 Jahren einerseits durch die Angewandte Mathematik in Form immer effizienterer Algorithmen und andererseits durch die Entwicklung immer schnellerer Rechner erreicht worden sind, an dem durchaus typischen Beispiel der Lösung eines großen Gleichungssystems, das bei der numerischen Behandlung einer Differentialgleichung auftritt, die Wärmeleitung und Diffusion beschreibt,

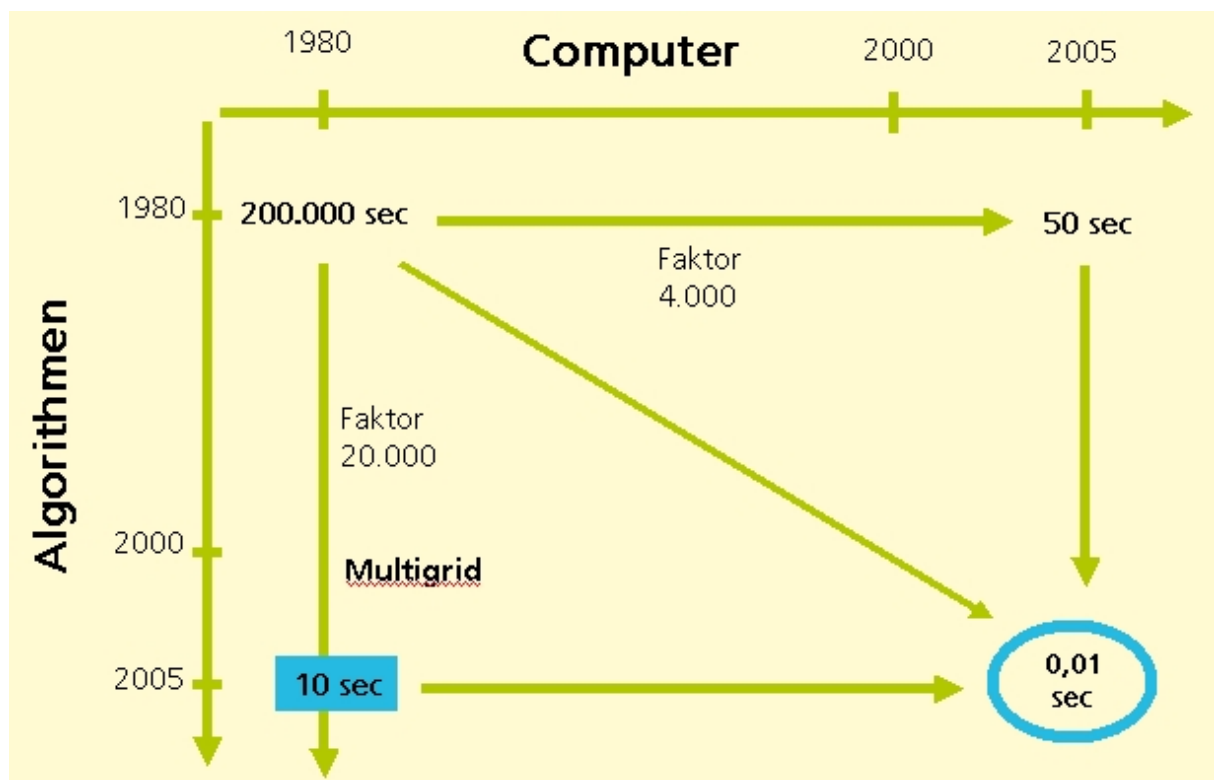


Abbildung 2: Fortschritte durch Mathematik und Rechnerentwicklung in den vergangenen 30 Jahren.

Auf der horizontalen Achse ist der allgemein bekannte Fortschritt durch die Rechnerentwicklung aufgetragen. Jeder weiß, dass Rechner in den vergangenen Jahrzehnten um ein Vielfaches (für dieses Anwendungsbeispiel um den Faktor 4000) schneller geworden sind.

Kaum bekannt ist hingegen, dass auf der Seite der Mathematik durch neue, schnellere Algorithmen Fortschritte in vergleichbarer Größenordnung erzielt worden sind. Bei dem Beispiel in Abb. 2 dauerte die Lösung des gegebenen Problems auf demselben langsamen Rechner von 1980 im Jahre 2005 nur noch 10 sec statt 200.000 sec, wenn man die 2005 bekannten schnellen mathematischen Algorithmen einsetzt statt der 1980 gebräuchlichen. Tatsächlich nutzt man heute natürlich nicht nur diese schnellen Algorithmen, sondern auch die heute vorhandenen viel schnelleren Rechner. So führen Fortschritte auf Seiten der Mathematik und der Rechnerentwicklung dazu, dass ein Problem, dessen Lösung 1980 noch etwa 200.000 sec (das sind rund 55 Stunden) dauerte, im Jahre 2005 in einem Bruchteil einer Sekunde gelöst werden konnte.

### 1.3 Schnelle und weniger schnelle Verfahren

Ziel dieses Abschnitts ist es, einen Eindruck über schnelle und weniger schnelle Verfahren zur Lösung von linearen Gleichungssystemen zu gewinnen, wie sie bei der Diskretisierung partieller Differentialgleichungen auftreten.

Wir betrachten als Beispiel die Poissonsche Randwertaufgabe (vgl. Abschnitt 4.4.7), bei der hier zur Bestimmung einer Näherungslösung ein Gitter mit je  $n$  Punkten in  $x$ - und in  $y$ -Richtung verwendet wird (vgl. Abschnitt 4.4). Wie wir in Abschnitt 4.4.10 sehen werden, ergibt sich für  $n = 1024$  ein lineares Gleichungssystem mit  $N = (n - 1)^2 \approx 10^6$  Unbekannten. Tab. 1 vergleicht die für die Lösung dieses Gleichungssystems benötigte Anzahl an Rechenoperationen und die benötigte Rechenzeit für verschiedene Verfahren auf einem Standard-PC. Sie gibt anschaulich die Leistungssteigerung wieder, die mit Hilfe von neuen Methoden der Numerischen Mathematik erzielt wurde.

Verfahren	Anzahl der Operationen	Rechenzeit
Cramersche Regel	$\sim N!$	" $\infty$ "
Gaußsches Eliminationsverfahren für Bandmatrizen	$\sim N^2$	14 h
Überrelaxationsverfahren, SOR (1960)	$\sim N^{1.5}$	5 min
Mehrgitter-Verfahren (1980)	$\sim N$	1 sec

Tabelle 1: Rechenzeiten verschiedener Verfahren zur Lösung eines Gleichungssystems mit rund 1 Million Unbekannten, das bei der numerischen Lösung einer einfachen zweidimensionalen Differentialgleichung entsteht.

### 1.4 Darstellung großer linearer Gleichungssysteme

Gegeben sei ein lineares Gleichungssystem mit  $N$  Gleichungen und  $N$  Unbekannten

$$\begin{array}{cccccc}
 a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1N}x_N & = & b_1 \\
 a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2N}x_N & = & b_2 \\
 \vdots & + & \cdots & + & \cdots & + & \vdots & = & \vdots \\
 a_{N1}x_1 & + & a_{N2}x_2 & + & \cdots & + & a_{NN}x_N & = & b_N .
 \end{array}$$

Wir schreiben dieses Gleichungssystem auch in der Form

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \cdots & \cdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

oder ganz kurz

$$Ax = b$$

mit der *Matrix*

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \cdots & \cdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \quad \text{und den Vektoren} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} .$$

**Beispiel 1.1** *Das Gleichungssystem mit  $N=3$  Unbekannten*

$$\begin{array}{rclcl} x_1 & + & & 4x_3 & = & 3 \\ 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ 2x_1 & + & 4x_2 & & & = & 6 \end{array}$$

*können wir mit*

$$A = \begin{pmatrix} 1 & 0 & 4 \\ 5 & 2 & 1 \\ 2 & 4 & 0 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} . \quad (1.1)$$

*auch schreiben als*

$$Ax = b \quad \text{oder} \quad \begin{pmatrix} 1 & 0 & 4 \\ 5 & 2 & 1 \\ 2 & 4 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} . \quad \triangle$$

# Kapitel 2

## Direkte Verfahren

### 2.1 Einführung

#### 2.1.1 Cramersche Regel und Determinanten

Man kann die Lösung eines linearen Gleichungssystems direkt als Formel hinschreiben. Wir werden in diesem Abschnitt sehen, wie diese Formel aussieht. Wir werden allerdings auch sehen, dass diese Formel nur in Ausnahmefällen hilfreich sein wird, weil der Rechenaufwand, den man braucht, um diese Formel auszuwerten, sehr schnell riesengroß wird.

**Cramersche Regel:** Wenn das aus  $N$  Gleichungen mit  $N$  Unbekannten bestehende lineare Gleichungssystem  $Ax = b$  ( $A$  ist dann eine  $N \times N$ -Matrix) genau eine Lösung hat, lässt sich diese Lösung wie folgt berechnen:

$$x_i = \frac{\det A_i(b)}{\det A} . \quad (2.1)$$

Dabei entsteht die Matrix  $A_i(b)$ , indem man in der Matrix  $A$  die  $i$ -te Spalte durch den Vektor  $b$  ersetzt. Hierbei bezeichnet  $\det A$  die sogenannte Determinante der Matrix  $A$ , die oft auch mit  $|A|$  bezeichnet wird, und  $\det A_i(b)$  bezeichnet entsprechend die Determinante der Matrix  $A_i(b)$ .

Um mit der Cramerschen Regel die Lösung eines Gleichungssystems berechnen zu können, müssen wir jetzt nur noch wissen, wie sich die Determinanten berechnen lassen.

**Determinanten für  $N=1$  und  $N=2$ :** Determinanten sind nur für quadratische Matrizen definiert. Die Determinante einer quadratischen Matrix ist immer eine Zahl. Für  $N = 1$ , d.h. für eine  $1 \times 1$ -Matrix  $A = (a)$  ist die Determinante gleich dem einzigen Element dieser Matrix:

$$\det(a) = a \text{ für alle } a \in \mathbb{R} .$$

Für eine  $2 \times 2$ -Matrix  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  gilt

$$\det A = |A| = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

**Beispiel 2.1** *Cramersche Regel an einem Beispiel von 2 Gleichungen mit 2 Unbekannten:*  
Wir betrachten das Gleichungssystem

$$\begin{aligned} 5x_1 + 2x_2 &= 1 \\ 2x_1 + 4x_2 &= 6 \end{aligned}$$

mit

$$A = \begin{pmatrix} 5 & 2 \\ 2 & 4 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 1 \\ 6 \end{pmatrix}.$$

Wollen wir die Cramersche Regel zur Lösung dieses Gleichungssystems benutzen, so benötigen wir die drei Determinanten  $\det A$ ,  $\det A_1 \begin{pmatrix} 1 \\ 6 \end{pmatrix}$  und  $\det A_2 \begin{pmatrix} 1 \\ 6 \end{pmatrix}$ , wie sie zu Beginn dieses Abschnitts eingeführt wurden (vgl. Erläuterung zu (2.1)). Damit erhalten wir

$$\begin{aligned} x_1 &= \frac{\det A_1 \begin{pmatrix} 1 \\ 6 \end{pmatrix}}{\det A} = \frac{\begin{vmatrix} 1 & 2 \\ 6 & 4 \end{vmatrix}}{\begin{vmatrix} 5 & 2 \\ 2 & 4 \end{vmatrix}} = \frac{1 \cdot 4 - 2 \cdot 6}{5 \cdot 4 - 2 \cdot 2} = \frac{4 - 12}{20 - 4} = \frac{-8}{16} = -\frac{1}{2} \\ x_2 &= \frac{\det A_2 \begin{pmatrix} 1 \\ 6 \end{pmatrix}}{\det A} = \frac{\begin{vmatrix} 5 & 1 \\ 2 & 6 \end{vmatrix}}{\begin{vmatrix} 5 & 2 \\ 2 & 4 \end{vmatrix}} = \frac{5 \cdot 6 - 1 \cdot 2}{5 \cdot 4 - 2 \cdot 2} = \frac{30 - 2}{20 - 4} = \frac{28}{16} = \frac{7}{4}. \quad \triangle \end{aligned}$$

**Determinanten für  $N \geq 3$ :** Die Determinante einer  $3 \times 3$ -Matrix  $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

lässt sich folgendermaßen berechnen:

$$\begin{aligned} \det A &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \\ &= a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned}$$

Hierbei haben wir die Berechnung der Determinante einer  $3 \times 3$ -Matrix auf die Berechnung von mehreren Determinanten von  $2 \times 2$ -Matrizen zurückgeführt. Dies kann man verallgemeinern. Allgemein gilt für  $N \times N$ -Matrizen mit  $N \geq 2$ :

$$\det A = \sum_{i=1}^N (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij} \quad (\text{sogenannte Entwicklung nach der } j\text{-ten Spalte, d.h. hier ist } j \text{ eine beliebige, aber feste Spalte})$$

und

$$\det A = \sum_{j=1}^N (-1)^{i+j} \cdot a_{ij} \cdot \det A_{ij} \quad (\text{sogenannte Entwicklung nach der } i\text{-ten Zeile, d.h. hier ist } i \text{ eine beliebige, aber feste Zeile})$$

wobei  $A_{ij}$  die  $(N-1) \times (N-1)$ -Matrix ist, die durch Streichen der  $i$ -ten Zeile und der  $j$ -ten Spalte aus  $A$  entsteht (Laplacescher Entwicklungssatz). Die oben angegebene Berechnung einer  $3 \times 3$ -Matrix entspricht einer Entwicklung nach der ersten Zeile.

**Beispiel 2.2** *Cramersche Regel an einem Beispiel von 3 Gleichungen mit 3 Unbekannten: Wir betrachten das Gleichungssystem aus Beispiel 1.1:*

$$\begin{array}{rclcl} x_1 & + & & 4x_3 & = & 3 \\ 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ 2x_1 & + & 4x_2 & & & = & 6 \end{array}$$

mit

$$A = \begin{pmatrix} 1 & 0 & 4 \\ 5 & 2 & 1 \\ 2 & 4 & 0 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}.$$

Nach der Cramerschen Regel benötigen wir für die Lösung dieses Gleichungssystems die 4 Determinanten

$$\det A, \det A_1 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}, \det A_2 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}, \det A_3 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix},$$

die wir wie folgt berechnen können:

$$\begin{aligned} \det A &= \begin{vmatrix} 1 & 0 & 4 \\ 5 & 2 & 1 \\ 2 & 4 & 0 \end{vmatrix} \\ &= 1 \cdot \begin{vmatrix} 2 & 1 \\ 4 & 0 \end{vmatrix} - 0 \cdot \begin{vmatrix} 5 & 1 \\ 2 & 0 \end{vmatrix} + 4 \cdot \begin{vmatrix} 5 & 2 \\ 2 & 4 \end{vmatrix} \\ &= 1 \cdot (2 \cdot 0 - 1 \cdot 4) + 4 \cdot (5 \cdot 4 - 2 \cdot 2) \\ &= -4 + 4 \cdot 16 = 60 \\ \det A_1 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} &= \begin{vmatrix} 3 & 0 & 4 \\ 1 & 2 & 1 \\ 6 & 4 & 0 \end{vmatrix} \quad \left( A_1 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \text{ entsteht wieder aus } A, \text{ indem wir die} \right. \\ &\quad \left. \text{erste Spalte von } A \text{ durch } b \text{ ersetzen} \right) \\ &= 3 \cdot \begin{vmatrix} 2 & 1 \\ 4 & 0 \end{vmatrix} - 0 \cdot \begin{vmatrix} 1 & 1 \\ 6 & 0 \end{vmatrix} + 4 \cdot \begin{vmatrix} 1 & 2 \\ 6 & 4 \end{vmatrix} \\ &= 3 \cdot (2 \cdot 0 - 1 \cdot 4) + 4 \cdot (1 \cdot 4 - 2 \cdot 6) \\ &= 3 \cdot (-4) + 4 \cdot (4 - 12) \\ &= -12 - 32 \\ &= -44 \end{aligned}$$

$$\begin{aligned}
\left| A_2 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \right| &= \begin{vmatrix} 1 & 3 & 4 \\ 5 & 1 & 1 \\ 2 & 6 & 0 \end{vmatrix} \\
&= 1 \cdot \begin{vmatrix} 1 & 1 \\ 6 & 0 \end{vmatrix} - 3 \cdot \begin{vmatrix} 5 & 1 \\ 2 & 0 \end{vmatrix} + 4 \cdot \begin{vmatrix} 5 & 1 \\ 2 & 6 \end{vmatrix} \\
&= 1 \cdot (1 \cdot 0 - 1 \cdot 6) - 3 \cdot (5 \cdot 0 - 1 \cdot 2) + 4 \cdot (5 \cdot 6 - 1 \cdot 2) \\
&= 1 \cdot (-6) - 3 \cdot (-2) + 4 \cdot (30 - 2) \\
&= -6 + 6 + 112 \\
&= 112
\end{aligned}$$

$$\begin{aligned}
\left| A_3 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \right| &= \begin{vmatrix} 1 & 0 & 3 \\ 5 & 2 & 1 \\ 2 & 4 & 6 \end{vmatrix} \\
&= 1 \cdot \begin{vmatrix} 2 & 1 \\ 4 & 6 \end{vmatrix} - 0 \cdot \begin{vmatrix} 5 & 1 \\ 2 & 6 \end{vmatrix} + 3 \cdot \begin{vmatrix} 5 & 2 \\ 2 & 4 \end{vmatrix} \\
&= 1 \cdot (2 \cdot 6 - 1 \cdot 4) + 3 \cdot (5 \cdot 4 - 2 \cdot 2) \\
&= 1 \cdot 8 + 3 \cdot 16 \\
&= 56
\end{aligned}$$

Als Lösung für unser Gleichungssystem erhalten wir damit

$$\begin{aligned}
x_1 &= \frac{\left| A_1 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \right|}{|A|} = -\frac{44}{60} = -\frac{11}{15} \\
x_2 &= \frac{\left| A_2 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \right|}{|A|} = \frac{112}{60} = \frac{28}{15} \\
x_3 &= \frac{\left| A_3 \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} \right|}{|A|} = \frac{56}{60} = \frac{14}{15}
\end{aligned}$$

△

An diesem Beispiel deutet sich bereits an, dass der Rechenaufwand für die Cramersche Regel bei Gleichungssystemen mit mehr als zwei Unbekannten recht hoch wird. Noch deutlicher wird dies bei größeren Werten von  $N$ . Im folgenden Beispiel berechnen wir (nur zur Abschreckung!) die Determinante einer  $4 \times 4$ -Matrix.

**Beispiel 2.3** Berechnung der Determinante einer  $4 \times 4$ -Matrix: In diesem Beispiel entwickeln wir alle auftretenden Determinanten nach der ersten Zeile.

$$\begin{aligned}
 \begin{vmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 2 & 5 & 4 \\ 8 & 7 & 6 & 5 \end{vmatrix} &= 1 \cdot \begin{vmatrix} 4 & 9 & 16 \\ 2 & 5 & 4 \\ 7 & 6 & 5 \end{vmatrix} - 2 \cdot \begin{vmatrix} 1 & 9 & 16 \\ 1 & 5 & 4 \\ 8 & 6 & 5 \end{vmatrix} + 3 \cdot \begin{vmatrix} 1 & 4 & 16 \\ 1 & 2 & 4 \\ 8 & 7 & 5 \end{vmatrix} - 4 \cdot \begin{vmatrix} 1 & 4 & 9 \\ 1 & 2 & 5 \\ 8 & 7 & 6 \end{vmatrix} \\
 &= 1 \cdot \left( 4 \cdot \begin{vmatrix} 5 & 4 \\ 6 & 5 \end{vmatrix} - 9 \cdot \begin{vmatrix} 2 & 4 \\ 7 & 5 \end{vmatrix} + 16 \cdot \begin{vmatrix} 2 & 5 \\ 7 & 6 \end{vmatrix} \right) \\
 &\quad - 2 \cdot \left( 1 \cdot \begin{vmatrix} 5 & 4 \\ 6 & 5 \end{vmatrix} - 9 \cdot \begin{vmatrix} 1 & 4 \\ 8 & 5 \end{vmatrix} + 16 \cdot \begin{vmatrix} 1 & 5 \\ 8 & 6 \end{vmatrix} \right) \\
 &\quad + 3 \cdot \left( 1 \cdot \begin{vmatrix} 2 & 4 \\ 7 & 5 \end{vmatrix} - 4 \cdot \begin{vmatrix} 1 & 4 \\ 8 & 5 \end{vmatrix} + 16 \cdot \begin{vmatrix} 1 & 2 \\ 8 & 7 \end{vmatrix} \right) \\
 &\quad - 4 \cdot \left( 1 \cdot \begin{vmatrix} 2 & 5 \\ 7 & 6 \end{vmatrix} - 4 \cdot \begin{vmatrix} 1 & 5 \\ 8 & 6 \end{vmatrix} + 9 \cdot \begin{vmatrix} 1 & 2 \\ 8 & 7 \end{vmatrix} \right) \\
 &= 1 \cdot (4 \cdot (5 \cdot 5 - 4 \cdot 6) - 9 \cdot (2 \cdot 5 - 4 \cdot 7) + 16 \cdot (2 \cdot 6 - 5 \cdot 7)) \\
 &\quad - 2 \cdot (1 \cdot (5 \cdot 5 - 4 \cdot 6) - 9 \cdot (1 \cdot 5 - 4 \cdot 8) + 16 \cdot (1 \cdot 6 - 5 \cdot 8)) \\
 &\quad + 3 \cdot (1 \cdot (2 \cdot 5 - 4 \cdot 7) - 4 \cdot (1 \cdot 5 - 4 \cdot 8) + 16 \cdot (1 \cdot 7 - 2 \cdot 8)) \\
 &\quad - 4 \cdot (1 \cdot (2 \cdot 6 - 5 \cdot 7) - 4 \cdot (1 \cdot 6 - 5 \cdot 8) + 9 \cdot (1 \cdot 7 - 2 \cdot 8)) \\
 &= 1 \cdot (4 \cdot (25 - 24) - 9 \cdot (10 - 28) + 16 \cdot (12 - 35)) \\
 &\quad - 2 \cdot (1 \cdot (25 - 24) - 9 \cdot (5 - 32) + 16 \cdot (6 - 40)) \\
 &\quad + 3 \cdot (1 \cdot (10 - 28) - 4 \cdot (5 - 32) + 16 \cdot (7 - 16)) \\
 &\quad - 4 \cdot (1 \cdot (12 - 35) - 4 \cdot (6 - 40) + 9 \cdot (7 - 16)) \\
 &= 1 \cdot (4 \cdot 1 - 9 \cdot (-18) + 16 \cdot (-23)) \\
 &\quad - 2 \cdot (1 \cdot 1 - 9 \cdot (-27) + 16 \cdot (-34)) \\
 &\quad + 3 \cdot (1 \cdot (-18) - 4 \cdot (-27) + 16 \cdot (-9)) \\
 &\quad - 4 \cdot (1 \cdot (-23) - 4 \cdot (-34) + 9 \cdot (-9)) \\
 &= 4 + 162 - 368 \\
 &\quad - 2 \cdot (1 + 243 - 544) \\
 &\quad + 3 \cdot (-18 + 108 - 144) \\
 &\quad - 4 \cdot (-23 + 136 - 81) \\
 &= -202 - 2 \cdot (-300) + 3 \cdot (-54) - 4 \cdot (32) \\
 &= -202 + 600 - 162 - 128 = 108 \quad \triangle
 \end{aligned}$$

Dieses Beispiel zeigt, dass die Berechnung von Determinanten über die Entwicklung nach Zeilen oder Spalten sehr viel Rechenaufwand erfordert, wenn nicht zufällig viele Elemente der Matrix 0 sind.

**Rechenaufwand für große Werte von  $N$ :** Berechnet man die Determinanten in der Formel (2.1) zur Bestimmung der Lösung eines linearen Gleichungssystems bei einem



Gleichungssystem mit  $N$  Unbekannten wie oben angegeben mit dem Laplaceschen Entwicklungssatz, so hat man eine Anzahl von Multiplikationen und Additionen auszuführen, die proportional zu  $N! = N \cdot (N-1) \cdot (N-2) \cdot (N-3) \cdot \dots \cdot 3 \cdot 2 \cdot 1$  ist. Damit wächst auch die Rechenzeit proportional zu  $N!$ . Schon für  $N = 50$  erhalten wir das ernüchternde Ergebnis, dass die Berechnung der Determinante auf einem schnellen Rechner rund  $10^{50}$  Jahre benötigen würde. So kann die Lösung  $x$  jedenfalls nicht gewonnen werden (außer für kleine  $N$  wie  $N = 2$  oder  $N = 3$  und in Spezialfällen).

**Effizientere Berechnung der Determinante einer Matrix:** Ist eine Matrix eine rechte obere oder linke untere Dreiecksmatrix, so ist ihre Determinante gleich dem Produkt ihrer Diagonalelemente.

### Beispiel 2.4

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 0 & 4 & 9 & 16 \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 0 & 5 \end{vmatrix} = 1 \cdot 4 \cdot 5 \cdot 5 = 100 \qquad \begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 \\ 1 & 2 & 5 & 0 \\ 8 & 7 & 6 & 5 \end{vmatrix} = 1 \cdot 4 \cdot 5 \cdot 5 = 100 \quad \triangle$$

Die Determinanten von Dreiecksmatrizen kann man also ganz einfach berechnen. Ferner kann man zeigen: Das Addieren des Vielfachen einer Zeile (oder Spalte) zu einer anderen Zeile (oder Spalte) ändert den Wert der Determinante nicht (wir verzichten hier auf einen Beweis). Dies kann man benutzen, um die Berechnung der Determinante einer allgemeinen Matrix auf die Berechnung der Determinante einer Dreiecksmatrix zurückzuführen. Wir demonstrieren dies am Beispiel der  $4 \times 4$ -Matrix aus Beispiel 2.3.

### Beispiel 2.5 *Addieren wir in*

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 2 & 5 & 4 \\ 8 & 7 & 6 & 5 \end{vmatrix}$$

das  $(-1)$ -fache der ersten Zeile zur zweiten und dritten Zeile sowie das  $(-8)$ -fache der ersten Zeile zur vierten Zeile, so erhalten wir

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 2 & 5 & 4 \\ 8 & 7 & 6 & 5 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & 2 & 0 \\ 0 & -9 & -18 & -27 \end{vmatrix}$$

Die ersten drei Zeilen der Matrix haben jetzt bereits obere Dreiecksgestalt. Zu bearbeiten bleibt lediglich noch die vierte Zeile. Um das zweite Element der vierten Zeile zu elimi-

nieren, addieren wir das  $(9/2)$ -fache der zweiten Zeile zur vierten Zeile und erhalten

$$\begin{array}{ccc} \left| \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & 2 & 0 \\ 0 & -9 & -18 & -27 \end{array} \right| & = & \left| \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 9 & 27 \end{array} \right| \\ & & \text{addieren wir nun noch das } \left(-\frac{9}{2}\right)\text{-fache} \\ & & \text{der 3. Zeile zur 4. Zeile, so erhalten wir} \\ & = & \left| \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 0 & 2 & 6 & 12 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 27 \end{array} \right| \\ & = & 1 \cdot 2 \cdot 2 \cdot 27 = 108 \end{array} \quad \triangle$$

Offensichtlich ist diese Art der Berechnung der Determinante um ein Vielfaches schneller als die Berechnung mit Hilfe des Laplaceschen Entwicklungssatzes über die Entwicklung nach Unterdeterminanten.

Bei der Berechnung der Lösung eines Gleichungssystems mit  $N$  Unbekannten sind dann aber immer noch  $N + 1$  Determinanten von  $N \times N$ -Matrizen zu berechnen. Das Gaußsche Eliminationsverfahren, das wir in Abschnitt 2.2 kennenlernen werden, löst ein solches Gleichungssystem mit einem Aufwand, der etwa der Berechnung lediglich einer solchen Determinante entspricht. Die Cramersche Regel spielt daher bei der praktischen Lösung von großen Gleichungssystemen keine Rolle.

### 2.1.2 Rundungsfehler

Rundungsfehler können bei linearen Gleichungssystemen, wenn man unbedacht vorgeht, manchmal einen katastrophalen Einfluss haben. Man kann das bereits an sehr kleinen Gleichungssystemen mit wenigen Unbekannten beobachten.

**Beispiel 2.6** Wir betrachten das Gleichungssystem  $Ax = b$  mit

$$(A|b) = \left( \begin{array}{ccc|c} 2 & 1.01 & 2.52 & 9.57 \\ 0.4 & 0.203 & -1.8 & -0.385 \\ 0.6 & -1.05 & 0.8 & -3.85 \end{array} \right).$$

Das Gaußsche Eliminationsverfahren liefert bei 4-stelliger Genauigkeit, d.h. mit dezimalen Gleitkommazahlen, deren Mantisse vierstellig ist, die exakte Lösung

$$x_1 = 1, \quad x_2 = 5, \quad x_3 = 1.$$

Wird die gleiche Rechnung jedoch mit 3-stelligen Gleitkommazahlen durchgeführt, so kommt folgende, völlig unsinnige Lösung heraus:

$$\hat{x}_1 = 3.53, \quad \hat{x}_2 = 0, \quad \hat{x}_3 = 1. \quad \triangle$$

Mit geeigneten Varianten des Gaußschen Eliminationsverfahrens kann man derartige “Katastrophen” verhindern, und daran ist man in der Praxis natürlich interessiert.

Wir werden hier keine derartigen problematischen Gleichungssysteme betrachten. Trotzdem ist es interessant zu wissen, wann dieses Problem auftreten kann. Bei zwei Gleichungen mit zwei Unbekannten beschreibt jede Gleichung eine Gerade. Sind diese Geraden nahezu parallel, dann können winzige Änderungen in den Koeffizienten der Geradengleichung (oder entsprechend im gegebenen Gleichungssystem) dazu führen, dass sich der Schnittpunkt der Geraden (d.h. die Lösung des Gleichungssystems) erheblich verschiebt.

Auch bei der zeichnerischen Lösung wird in derartigen Fällen die genaue Bestimmung des Schnittpunktes zweier Geraden problematisch.

Analog ist es im dreidimensionalen Fall kritisch, wenn die Ebenen, die durch die Gleichungen beschrieben werden, fast parallel sind, oder wenn die Schnittgeraden von je zwei Ebenen nahezu parallel sind.

## 2.2 Idee des Gaußschen Eliminationsverfahrens

Gegeben sei das lineare Gleichungssystem

$$Ax = b \quad (2.2)$$

mit

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \vdots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}.$$

Ferner nehmen wir an, dass das Gleichungssystem genau eine Lösung hat.

### 2.2.1 Zielsystem: Matrix in Dreiecksgestalt

Die Grundidee des Gaußschen Eliminationsverfahrens ist, das obige Gleichungssystem so umzuformen, dass die Lösungsgesamtheit unverändert bleibt (äquivalente Umformung) und das Zielsystem die folgende Gestalt hat

$$\begin{array}{cccccccl} r_{11}x_1 & + & r_{12}x_2 & + & \cdots & + & r_{1N}x_N & = & y_1 \\ 0 & & r_{22}x_2 & + & \cdots & + & r_{2N}x_N & = & y_2 \\ 0 & & 0 & & r_{33}x_3 & \cdots & + & r_{3N}x_N & = & y_3 \\ \vdots & & \vdots & & \ddots & & \vdots & = & \vdots \\ 0 & & 0 & \cdots & 0 & & r_{NN}x_N & = & y_N \end{array}, \quad (2.3)$$

also ein Dreiecksschema ist. Man kann dies auch in Matrixform als  $Rx = y$  mit einer oberen Dreiecksmatrix

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ 0 & r_{22} & \cdots & r_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{NN} \end{pmatrix} \quad \text{und} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

schreiben. Das Gleichungssystem  $Rx = y$  ist dann leicht zu lösen. Dabei setzen wir voraus, dass alle Zahlen in der Hauptdiagonalen der Matrix von Null verschieden sind. Wir beginnen mit der letzten Zeile und bestimmen  $x_N$ . Diesen Wert setzen wir nun in die vorletzte Zeile ein und können dann leicht  $x_{N-1}$  ermitteln usw. Im letzten Auflösungsschritt berechnen wir, da wir alle Werte  $x_N, x_{N-1}, \dots, x_2$  bereits kennen, den Wert von  $x_1$ . Damit sind alle Lösungswerte bekannt.

Dieses Vorgehen können wir folgendermaßen notieren:

$$\begin{aligned} x_N &= \frac{1}{r_{NN}} \cdot y_N \\ x_{N-1} &= \frac{1}{r_{N-1,N-1}} \cdot (y_{N-1} - r_{N-1,N}x_N) \\ x_{N-2} &= \frac{1}{r_{N-2,N-2}} \cdot (y_{N-2} - r_{N-2,N-1}x_{N-1} - r_{N-2,N}x_N) \\ &\vdots \\ x_1 &= \frac{1}{r_{11}} \cdot (y_1 - r_{12}x_2 - r_{13}x_3 - \dots - r_{1N}x_N). \end{aligned}$$

Dies können wir auch kompakt schreiben als

$$x_i = \frac{1}{r_{ii}} \left( y_i - \sum_{j=i+1}^N r_{ij}x_j \right), \quad i = N, N-1, \dots, 1. \quad (2.4)$$

Man beachte, dass der Summenausdruck in Gleichung (2.4) den Wert 0 hat, wenn das Summationsintervall leer ist, d.h. wenn  $i+1 > N$  ist (tritt hier bei  $i = N$  ein.) Man beachte ferner, dass die Rekursion (2.4) beim größten Indexwert ( $i = N$ ) beginnt und beim kleinsten ( $i = 1$ ) endet. Sie läuft also rückwärts, gegen die natürliche Reihenfolge der natürlichen Zahlen. Wir nennen den obigen Prozess “Rückwärtseinsetzen” (engl. “back substitution”).

## 2.2.2 Der erste Eliminationsschritt

Um das zu lösende Gleichungssystem (2.2) auf die gewünschte Dreiecksform (2.3) zu bringen, ohne die Lösungsmenge zu ändern, lösen wir die erste Gleichung nach  $x_1$  auf und setzen diesen Ausdruck in die anderen Gleichungen ein. Hierbei haben wir vorausgesetzt, dass  $a_{11} \neq 0$  ist. (Genau dasselbe tun wir übrigens, wenn wir das  $-a_{i1}/a_{11}$  fache der ersten Zeile zur  $i$ -ten Zeile addieren.) Allgemein erhalten wir so

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij} - \frac{a_{i1}}{a_{11}}a_{1j} \quad (j = 2, \dots, N) \\ a_{i1}^{(1)} &= 0 \\ b_i^{(1)} &= b_i - \frac{a_{i1}}{a_{11}}b_1. \end{aligned}$$

Damit haben wir die Unbekannte  $x_1$  aus allen außer der ersten Gleichung eliminiert und erhalten das GLS  $A^{(1)}x = b^{(1)}$ :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2N}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3N}^{(1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & a_{N2}^{(1)} & a_{N3}^{(1)} & \cdots & a_{NN}^{(1)} \end{pmatrix} x = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(1)} \\ \vdots \\ b_N^{(1)} \end{pmatrix}$$

### 2.2.3 Fortführung des Verfahrens

Das weitere Eliminationsverfahren läuft entsprechend ab: Die für  $Ax = b$  beschriebene Idee wird jetzt sinngemäß auf  $(A^{(1)}x^{(1)} = b^{(1)})$  angewendet, genauer: auf das Teilsystem (Restsystem)

$$\begin{pmatrix} a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2N}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3N}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{N2}^{(1)} & a_{N3}^{(1)} & \cdots & a_{NN}^{(1)} \end{pmatrix} \begin{pmatrix} x_2^{(1)} \\ x_3^{(1)} \\ \vdots \\ x_N^{(1)} \end{pmatrix} = \begin{pmatrix} b_2^{(1)} \\ b_3^{(1)} \\ \vdots \\ b_N^{(1)} \end{pmatrix}.$$

Unter der Voraussetzung, dass  $a_{22}^{(1)} \neq 0$  ist, werden jetzt die Elemente unterhalb von  $a_{22}^{(1)}$  zu Null transformiert.

Insgesamt setzt sich das Gaußsche Eliminationsverfahren aus  $N - 1$  Eliminationsschritten zusammen, durch welche ausgehend von  $Ax = b$  sukzessive die Spalten der Matrix unterhalb des Diagonalelements zu 0 gemacht (eliminiert) werden. Das Endsystem hat dann die gewünschte Dreiecksgestalt. Dieses Verfahren geht theoretisch solange gut, wie die Elemente  $a_{kk}^{(k-1)}$  ( $k = 1, \dots, N - 1$ ) von Null verschieden bleiben, da ja jedesmal durch sie geteilt wird. Sollte im Verlauf dieses Eliminationsprozesses ein solches Diagonalelement Null werden, so können wir die Reihenfolge der Gleichungen noch vertauschen.

**Beispiel 2.7** Wir betrachten das lineare Gleichungssystem

$$\begin{aligned} 5x_1 + 2x_2 + x_3 &= 1 \\ 2x_1 + 4x_2 + 2x_3 &= 6 \\ x_1 + x_2 + 4x_3 &= 3 \end{aligned}$$

mit

$$A = \begin{pmatrix} 5 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 4 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 1 \\ 6 \\ 3 \end{pmatrix}.$$

Führen wir den Eliminationsprozess so durch, wie wir ihn oben beschrieben haben, so erhalten wir nach der Elimination der Unbekannten  $x_1$  aus der zweiten und dritten Gleichung das Gleichungssystem

$$\begin{pmatrix} 5 & 2 & 1 \\ 0 & 16/5 & 8/5 \\ 0 & 3/5 & 19/5 \end{pmatrix} x = \begin{pmatrix} 1 \\ 28/5 \\ 14/5 \end{pmatrix}.$$

Nach dem zweiten Eliminationsschritt haben wir entsprechend das Gleichungssystem

$$\begin{pmatrix} 5 & 2 & 1 \\ 0 & 16/5 & 8/5 \\ 0 & 0 & 7/2 \end{pmatrix} x = \begin{pmatrix} 1 \\ 28/5 \\ 7/4 \end{pmatrix}.$$

Damit haben wir das angestrebte Zielschema erhalten und können es nun durch Rückwärtseinsetzen auflösen. Wir erhalten

$$\begin{aligned} x_3 &= \frac{7/4}{7/2} = \frac{1}{2} \\ x_2 &= \frac{28/5 - 8/5 \cdot 1/2}{16/5} = \frac{24}{16} = \frac{3}{2} \\ x_1 &= \frac{1 - 1/2 - 2 \cdot 3/2}{5} = -\frac{5/2}{5} = -\frac{1}{2} \end{aligned}$$

### 2.2.4 Nachteile des Gaußschen Eliminationsverfahrens

Das Gaußsche Eliminationsverfahren ist ein direktes Verfahren, das i.a. verwendet wird, falls  $A$  eine vollbesetzte Matrix ist. Bei vollbesetzten Matrizen wächst der Rechenaufwand des Gaußschen Eliminationsverfahrens im Wesentlichen mit der dritten Potenz der Anzahl der Unbekannten (Schreibweise:  $W = O(N^3)$ ).

In der Praxis treten sehr häufig Matrizen auf, bei denen nur wenige Diagonalen von 0 verschieden sind (Bandmatrizen). Die bei der Diskretisierung der Poisson-Gleichung entstehenden GLS etwa haben eine solche Struktur. Hier reduziert sich der Aufwand des Gaußschen Eliminationsverfahrens auf  $N^2$ . Aber auch dies ist immer noch viel zu hoch für viele Anwendungen aus der Praxis.

# Kapitel 3

## Iterative Lösung von linearen Gleichungssystemen

### 3.1 Jacobi-Verfahren und Gauss-Seidel-Verfahren

In der Praxis hat  $A$  häufig eine spezielle Struktur und/oder ist schwach besetzt (engl. sparse), d.h. die meisten Elemente von  $A$  sind 0.  $N$  ist eventuell sehr groß, z.B.  $10^6$ . Dies bedeutet, dass man Gleichungssysteme mit einer Million von Unbekannten (oder mehr) zu lösen hat.

In solchen Fällen sind iterative Lösungsverfahren eine interessante Alternative. Bei ihnen startet man mit irgendeiner Startnäherung für die Unbekannten (im Zweifelsfall nimmt man an, dass alle Unbekannten 0 sind) und reduziert den Fehler sukzessiv durch eine geeignete Iterationsvorschrift. Wir werden im nächsten Abschnitt zwei einfache iterative Verfahren kennenlernen.

#### 3.1.1 Explizite Formeln für Jacobi- und Gauss-Seidel-Verfahren

**Beispiel 3.1** Für  $N = 3$  betrachten wir das Gleichungssystem

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Unter der Voraussetzung, dass  $a_{ii} \neq 0$  ( $i = 1, 2, 3$ ), können wir jeweils die  $i$ -te Gleichung nach  $x_i$  auflösen und erhalten

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3) \\ x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3) \\ x_3 &= \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2) . \end{aligned} \tag{3.1}$$

Wenn eine Startnäherung

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{pmatrix}$$

gegeben ist, können wir diese in die rechte Seite von (3.1) einsetzen und so gemäß

$$\begin{aligned} x_1^{(1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}) \\ x_2^{(1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)}) \\ x_3^{(1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)}) , \end{aligned} \quad (3.2)$$

eine neue Näherung

$$x^{(1)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix}$$

berechnen. Dies ist der erste Schritt des **Jacobi-Verfahren (JAC)**, das auch als Gesamtschrittverfahren bezeichnet wird.

Werden in der Iterationsvorschrift (3.2) auf der rechten Seite nicht nur Werte  $x_k^{(0)}$ , sondern - soweit verfügbar - die neuen, bereits berechneten Werte  $x_k^{(1)}$  eingesetzt, erhält man den ersten Iterationsschritt des **Gauss-Seidel-Verfahrens (GS)**, das auch unter dem Namen Einzelschrittverfahren bekannt ist:

$$\begin{aligned} x_1^{(1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}) \\ x_2^{(1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}) \\ x_3^{(1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}) . \end{aligned} \quad (3.3)$$

△

Für allgemeines  $N$  (genauer: für  $Ax = b$  mit  $A \in \mathbb{R}^{N,N}$ ,  $a_{ii} \neq 0$ , ( $i = 1, \dots, N$ )) lautet die Iterationsvorschrift von JAC:

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{k=1, k \neq i}^N a_{ik}x_k^{(n)} \right)} \quad (i = 1, \dots, N) . \quad (3.4)$$

Dabei sind die  $x_k^{(0)}$  ( $k = 1, \dots, N$ ) Startnäherungen (z.B. alle 0) und  $n = 0, 1, 2, \dots$  bezeichnet den Iterationsindex.

Werden in der Iterationsvorschrift (3.4) auf der rechten Seite nicht nur Werte  $x_k^{(n)}$ , sondern - soweit verfügbar - die neuen, bereits berechneten Werte  $x_k^{(n+1)}$  eingesetzt, erhält man GS:

$$\boxed{x_i^{(n+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{k=1}^{i-1} a_{ik}x_k^{(n+1)} - \sum_{k=i+1}^N a_{ik}x_k^{(n)} \right)} \quad (i = 1, \dots, N) . \quad (3.5)$$



### 3.1.2 Zwei konkrete Beispiele

Wir betrachten zunächst das GLS

$$\begin{aligned} 10x_1 + x_2 &= 1 \\ x_1 + 10x_2 &= 10 \end{aligned}$$

mit der exakten Lösung  $x_1 = 0$  und  $x_2 = 1$ . Wir beginnen mit der Startnäherung  $x_1^{(0)} = x_2^{(0)} = 0$ . Die Iterationsvorschrift für JAC lautet

$$\begin{aligned} x_1^{(n+1)} &= \frac{1 - x_2^{(n)}}{10} \\ x_2^{(n+1)} &= \frac{10 - x_1^{(n)}}{10} , \end{aligned}$$

und wir können daraus leicht berechnen

$$\begin{aligned} (x_1^{(0)}, x_2^{(0)}) &= (0, 0) \\ (x_1^{(1)}, x_2^{(1)}) &= (0.1, 0) \\ (x_1^{(2)}, x_2^{(2)}) &= (0, 0.99) \\ (x_1^{(3)}, x_2^{(3)}) &= (10^{-3}, 0) \\ (x_1^{(4)}, x_2^{(4)}) &= (0, 1 - 10^{-4}) \\ (x_1^{(5)}, x_2^{(5)}) &= (10^{-5}, 1) . \end{aligned}$$

Für GS haben wir die Iterationsvorschrift

$$\begin{aligned} x_1^{(n+1)} &= \frac{1 - x_2^{(n)}}{10} \\ x_2^{(n+1)} &= \frac{10 - x_1^{(n+1)}}{10} , \end{aligned}$$

und wir erhalten daraus

$$\begin{aligned} (x_1^{(0)}, x_2^{(0)}) &= (0, 0) \\ (x_1^{(1)}, x_2^{(1)}) &= (0.1, 0.99) \\ (x_1^{(2)}, x_2^{(2)}) &= (10^{-3}, 1 - 10^{-4}) \\ (x_1^{(3)}, x_2^{(3)}) &= (10^{-5}, 1 - 10^{-6}) . \end{aligned}$$

Beide Verfahren konvergieren offenbar gegen die richtige Lösung  $x_1 = 0$  und  $x_2 = 1$ , wobei in diesem Fall GS schneller als JAC zu sein scheint.

Als zweites Beispiel betrachten wir das Gleichungssystem

$$\begin{pmatrix} 10 & -4 & -2 \\ -4 & 10 & -4 \\ -6 & -2 & 12 \end{pmatrix} x = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} .$$

In diesem Beispiel berechnet sich dann etwa  $x^{(n+1)}$  zu

$$\begin{aligned} x_1^{(n+1)} &= (4 \cdot x_2^{(n)} + 2 \cdot x_3^{(n)} + \underbrace{2}_{b_1}) / \underbrace{10}_{a_{11}} \\ x_2^{(n+1)} &= (4 \cdot x_1^{(n)} + 4 \cdot x_3^{(n)} + \underbrace{3}_{b_2}) / \underbrace{10}_{a_{22}} \\ x_3^{(n+1)} &= (6 \cdot x_1^{(n)} + 2 \cdot x_2^{(n)} + \underbrace{1}_{b_3}) / \underbrace{12}_{a_{33}}, \end{aligned}$$

und man erhält hieraus

$$\begin{aligned} x^{(0)} &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & x^{(1)} &= \begin{pmatrix} 0.2 \\ 0.3 \\ 0.083 \end{pmatrix} \\ x^{(2)} &= \begin{pmatrix} 0.337 \\ 0.413 \\ 0.233 \end{pmatrix} & x^{(3)} &= \begin{pmatrix} 0.412 \\ 0.528 \\ 0.321 \end{pmatrix}. \end{aligned}$$

Bei GS ist im Unterschied zu JAC wie oben beschrieben immer mit den neuesten verfügbaren Näherungen zu rechnen:

$$\begin{aligned} x_1^{(n+1)} &= (4 \cdot x_2^{(n)} + 2 \cdot x_3^{(n)} + \underbrace{2}_{b_1}) / \underbrace{10}_{a_{11}} \\ x_2^{(n+1)} &= (4 \cdot x_1^{(n+1)} + 4 \cdot x_3^{(n)} + \underbrace{3}_{b_2}) / \underbrace{10}_{a_{22}} \\ x_3^{(n+1)} &= (6 \cdot x_1^{(n+1)} + 2 \cdot x_2^{(n+1)} + \underbrace{1}_{b_3}) / \underbrace{12}_{a_{33}}. \end{aligned}$$

Wir finden

$$\begin{aligned} x^{(0)} &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & x^{(1)} &= \begin{pmatrix} 0.200 \\ 0.380 \\ 0.247 \end{pmatrix} \\ x^{(2)} &= \begin{pmatrix} 0.401 \\ 0.559 \\ 0.377 \end{pmatrix} & x^{(3)} &= \begin{pmatrix} 0.499 \\ 0.650 \\ 0.441 \end{pmatrix}. \end{aligned}$$

Die exakte Lösung dieses GLS beträgt auf zwei Stellen gerundet  $x_1 = 0.60$ ,  $x_2 = 0.74$  und  $x_3 = 0.51$ . Nach drei Iterationsschritten sind wir zwar näher an die exakte Lösung herangekommen, aber wir brauchen sicherlich noch weitere Schritte, um eine gute Genauigkeit zu erzielen.

## 3.2 Das SOR-Verfahren

Das SOR (Successive OverRelaxation) Verfahren erhält man aus GS durch eine kleine, aber außerordentlich wirkungsvolle Verallgemeinerung, indem man die Korrektur, die ein

Iterationsschritt von GS liefert, mit einem geeigneten Faktor  $\omega$  multipliziert. Dazu wird die Iterationsvorschrift (3.5) von GS als ein erster Zwischenschritt betrachtet, der die Größe  $\tilde{x}_i^{(n)}$  liefert.

$$\tilde{x}_i^{(n)} = \frac{1}{a_{ii}} \left( b_i - \sum_{k=1}^{i-1} a_{ik} x_k^{(n+1)} - \sum_{k=i+1}^N a_{ik} x_k^{(n)} \right) \quad (i = 1, \dots, N) . \quad (3.6)$$

Im SOR-Verfahren erhält man jetzt  $x_i^{(n+1)}$  durch

$$x_i^{(n+1)} = x_i^{(n)} + \omega(\tilde{x}_i^{(n)} - x_i^{(n)}) . \quad (3.7)$$

Dabei ist  $\omega$  ein Parameter, der problemangepaßt gewählt wird. Für  $\omega > 1$  (Überrelaxation) verstärkt man die Korrektur  $x_i^{(n+1)} - x_i^{(n)}$  von  $x_i^{(n)}$  die der entsprechende Iterationsschritt von GS liefern würde, für  $\omega < 1$  (Unterrelaxation) schwächt man sie ab. Für  $\omega = 1$  erhält man wieder GS.

### 3.3 Zur Konvergenz iterativer Verfahren

#### 3.3.1 Zur Konvergenz des Jacobi-Verfahrens, starkes Zeilensummenkriterium

Gegeben sei ein GLS  $Ax = b$  mit einer Matrix  $A$ , die aus den Elementen  $a_{ij}$  besteht ( $i, j = 1, \dots, N$ ). Die Bedingung

$$\sum_{\substack{k=1 \\ k \neq i}}^N |a_{ik}| < |a_{ii}| \quad (i = 1, \dots, N) .$$

heißt *starkes Zeilensummenkriterium*. (Wenn diese Bedingung erfüllt ist, d.h. wenn in jeder Zeile  $i = 1, \dots, N$  der Betrag des Diagonalelementes  $a_{ii}$  größer als die Summe der Beträge der Nichtdiagonalelemente ist, sagt man auch, die Matrix  $A$  ist “stark diagonal-dominant”.)

Man kann zeigen: Ist für eine Matrix  $A$  das starke Zeilensummenkriterium erfüllt (d.h. überwiegt in der Matrix  $A$  die Hauptdiagonale in obigem Sinn), dann konvergiert das Jacobi-Verfahren gegen die eindeutig bestimmte Lösung von  $Ax = b$ . Jeder Iterationsschritt des Jacobi-Verfahrens reduziert dann den Fehler mindestens um den Faktor

$$\frac{\sum_{\substack{k=1 \\ k \neq i}}^N |a_{ik}|}{|a_{ii}|} .$$

#### 3.3.2 Ein Beispiel

Für die Matrix

$$A = \begin{pmatrix} 10 & -4 & -2 \\ -4 & 10 & -4 \\ -6 & -2 & 12 \end{pmatrix}$$

(vgl. Beispiel in Abschnitt 3.1.2) ist das starke Zeilensummenkriterium erfüllt, denn

$$4 + 2 < 10, \quad 4 + 4 < 10, \quad 6 + 2 < 12 \ .$$

Nach der Aussage im letzten Abschnitt konvergiert also das Jacobi-Verfahren für jedes GLS  $Ax = b$  mit dieser Matrix  $A$ . Jeder Iterationsschritt reduziert den Fehler dann mindestens um den Faktor

$$\max \left\{ \frac{4}{10} + \frac{2}{10}, \frac{4}{10} + \frac{4}{10}, \frac{6}{12} + \frac{2}{12} \right\} = \frac{8}{10} = 0.8 < 1 \ .$$

### 3.3.3 Zusammenhang zum Gauss-Seidel-Verfahren

**Bemerkung 3.1** Man kann beweisen, dass die Konvergenz von GS aus der Konvergenz von JAC folgt, *wenn das starke Zeilensummenkriterium erfüllt ist*. Das starke Zeilensummenkriterium ist also auch hinreichend für die Konvergenz von GS.

Es sei jedoch hier bemerkt, dass es Fälle gibt, in denen JAC konvergiert, nicht aber GS, und umgekehrt. >>

# Kapitel 4

## Ableitungen, Differentialgleichungen, Randwertaufgaben

### 4.1 Ableitung einer Funktion

Gegeben sei eine Funktion  $u(x)$ , die für alle reellen Werte  $x$  definiert sei. Ferner lasse sich im Punkt  $x_0$  eindeutig eine Tangente an  $u$  legen (*dies wird im folgenden allgemein vorausgesetzt!*). Dann definieren wir die Ableitung der Funktion  $u$  in einem Punkt  $x_0$  als die Steigung  $m$  der Tangente von  $u$  im Punkt  $x_0$  und bezeichnen die Ableitung mit  $u'(x_0)$ .

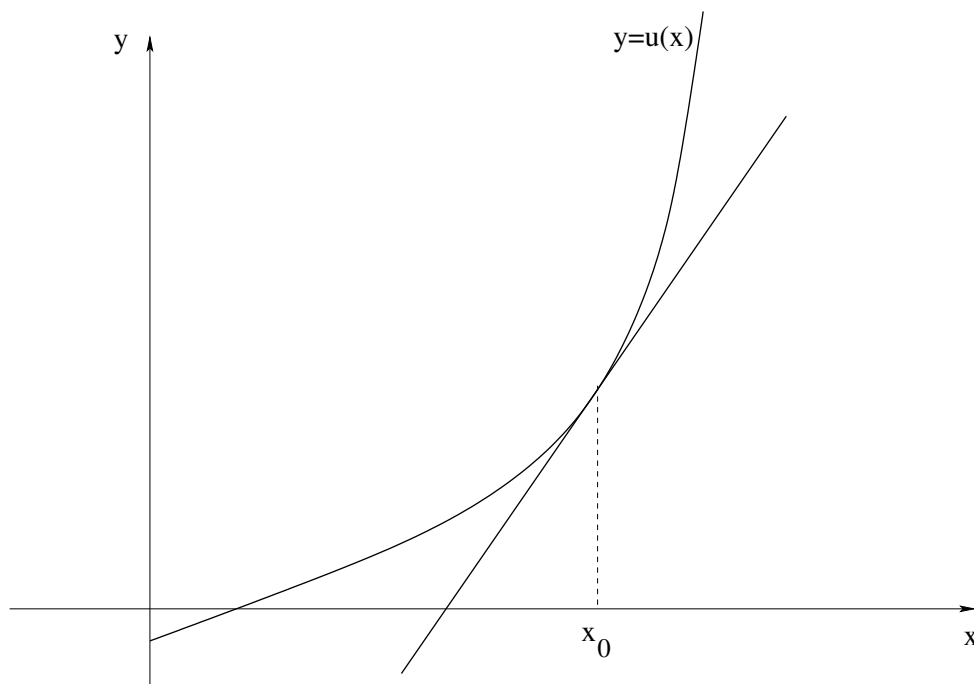


Abbildung 1: Tangente im Punkte  $x_0$  an eine Funktion  $u(x)$

**Beispiele:**

1. Die Gerade  $u(x) = mx + b$  ist in jedem Punkt ihre eigene Tangente. Sie hat die Steigung  $m$ . Damit gilt für die Ableitung der Geraden in jedem Punkt  $x$ :

$$u'(x) = m .$$

2. Die konstante Funktion  $u(x) = b$  ist eine spezielle Funktion mit der Steigung  $m = 0$ . Folglich gilt für sie in jedem Punkt  $u'(x) = 0$ .

Existiert die Ableitung z.B. in jedem Punkt, kann man  $u'$  wieder als Funktion von  $x$  begreifen und die Ableitung der Ableitung (d.h. die zweite Ableitung von  $u$ ) bestimmen. Offenbar ist für jede lineare Funktion  $u(x) = mx + b$  die zweite Ableitung identisch gleich 0, da die erste Ableitung eine Konstante ist, d.h.  $u''(x) = 0$ .

### 4.1.1 Berechnung von Ableitungen, Differenzenquotienten

Will man die Ableitung einer allgemeineren Funktion im Punkt  $x_0$  berechnen, so kann man die Steigung der Tangente approximieren, indem man die Steigung der Sekante durch die Punkte  $(x_0, u(x_0))$  und  $(x_0 + h, u(x_0 + h))$  berechnet und dann  $h$  gegen 0 streben lässt (vgl. Abb. 2). Das heißt:

$$u'(x_0) = \lim_{h \rightarrow 0} \frac{u(x_0 + h) - u(x_0)}{h} . \quad (4.1)$$

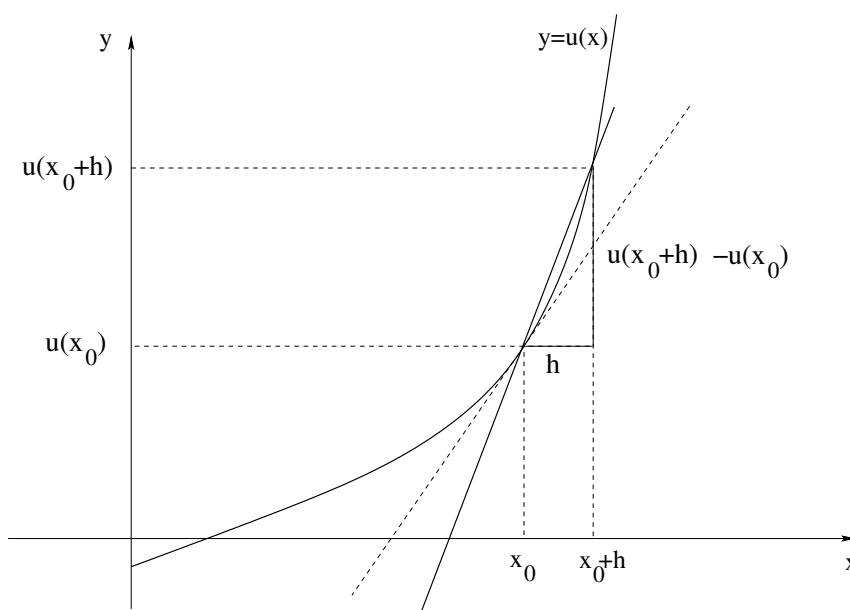


Abbildung 2: Tangente und Sekante einer Funktion

Tatsächlich kann man auch dies als die Ableitung der Funktion  $u$  im Punkt  $x_0$  definieren, sofern der Grenzwert existiert und eindeutig ist.

Ganz analog kann man auch die Sekante durch die Punkte  $(x_0, u(x_0))$  und  $(x_0-h, u(x_0-h))$  verwenden und erhält dann:

$$u'(x_0) = \lim_{h \rightarrow 0} \frac{u(x_0) - u(x_0 - h)}{h} . \quad (4.2)$$

Die Ausdrücke

$$\frac{u(x_0 + h) - u(x_0)}{h} \quad \text{und} \quad \frac{u(x_0) - u(x_0 - h)}{h}$$

werden auch als (rechtsseitiger und linksseitiger) *Differenzenquotient* bezeichnet.

### 4.1.2 Beispiele für Ableitungen von Funktionen

**Beispiel 4.1** Die Funktion  $u(x) = |x|$  hat für  $x = 0$  keine eindeutig bestimmte Tangente. Daher gilt für ihre Ableitung

$$u'(x) = \begin{cases} -1 & \text{für } x < 0 \\ +1 & \text{für } x > 0 \\ \text{nicht definiert} & \text{für } x = 0 \end{cases} . \quad \triangle$$

**Beispiel 4.2** Die Ableitung der Funktion  $u(x) = x^n$  berechnen wir als

$$\begin{aligned} u'(x) &= \lim_{h \rightarrow 0} \frac{u(x+h) - u(x)}{h} \\ &= \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} \\ &= \lim_{h \rightarrow 0} \frac{x^n + nhx^{n-1} + \frac{n(n-1)}{2}h^2x^{n-2} + \dots + \frac{n(n-1)}{2}h^{n-2}x^2 + nh^{n-1}x + h^n - x^n}{h} \\ &= \lim_{h \rightarrow 0} \frac{nhx^{n-1} + \frac{n(n-1)}{2}h^2x^{n-2} + \dots + \frac{n(n-1)}{2}h^{n-2}x^2 + nh^{n-1}x + h^n}{h} \\ &= \lim_{h \rightarrow 0} \left( nx^{n-1} + \frac{n(n-1)}{2}hx^{n-2} + \dots + \frac{n(n-1)}{2}h^{n-3}x^2 + nh^{n-2}x + h^{n-1} \right) \\ &= nx^{n-1} . \end{aligned}$$

**Beispiel 4.3** Ohne Beweis vermerken wir, dass die Ableitungen der Funktionen  $\sin(x)$ ,  $\cos(x)$  und  $e^x$  gegeben sind durch

$$\begin{aligned} (\sin x)' &= \cos(x) \\ (\cos x)' &= -\sin(x) \\ (e^x)' &= e^x \end{aligned}$$

Damit gilt für die zweiten Ableitungen dieser Funktionen

$$\begin{aligned} (\sin x)'' &= -\sin(x) \\ (\cos x)'' &= -\cos(x) \\ (e^x)'' &= e^x \end{aligned} \quad \triangle$$

### 4.1.3 Einfache Rechenregeln für Ableitungen

Seien  $u(x)$  und  $v(x)$  zwei Funktionen von  $x$ , deren Ableitungen im Punkte  $x_0$  existieren. Dann gilt

$$\begin{aligned}(u(x) + v(x))' &= u'(x) + v'(x) \\ (cu(x))' &= cu'(x) \quad c = \text{const.} \in \mathbb{R} \\ (u(x)v(x))' &= u'(x)v(x) + u(x)v'(x) .\end{aligned}$$

**Beispiele:**

$$\begin{aligned}(7x^3)' &= 7(x^3)' = 7 \cdot 3x^2 = 21x^2 \\ (5x^2 + 7x^3)' &= (5x^2)' + (7x^3)' = 5(x^2)' + 21x^2 = 10x + 21x^2 \\ (x^2 \sin x)' &= (x^2)' \sin x + x^2(\sin x)' = 2x \sin x + x^2 \cos x\end{aligned}$$

## 4.2 Gewöhnliche Differentialgleichungen

Als gewöhnliche Differentialgleichung (DGL)  $n$ -ter Ordnung bezeichnen wir eine Gleichung, in der die  $n$ -te Ableitung der Funktion  $u(x)$  vorkommt. Ferner können in dieser Gleichung auch  $u(x)$  und niedrigere Ableitungen von  $u(x)$  sowie weitere  $x$ -und  $u$ -abhängige Terme vorkommen.

Einige einfache Beispiele für Differentialgleichungen sind:

1.  $u'(x) = 0$

Lösungen dieser DGL sind alle konstanten Funktionen; denn die Steigung (Ableitung) jeder konstanten Funktion ist 0.

2.  $u'(x) = x$

Lösungen dieser DGL sind die Funktionen  $u(x) = \frac{1}{2}x^2 + c$  mit einer Konstanten  $c$ ; denn leitet man diese Funktionen ab, so erhält man  $u'(x) = x$ .

3.  $u'(x) = \cos(x)$

Lösungen dieser DGL sind die Funktionen  $u(x) = \sin(x) + c$  mit einer Konstanten  $c$ . Auch dies kann man durch Ableiten überprüfen.

4.  $u''(x) = -u(x)$

Lösungen dieser DGL sind die Funktionen  $u(x) = a \sin(x) + b \cos(x)$ . Auch dies überprüft man leicht durch Ableiten dieser Funktionen.

5.  $u'''(x) + u''(x) \cdot u'(x) - u^2(x) = r(x)$  mit einer gegebenen Funktion  $r(x)$ .

Offenbar sind in den ersten 4 Beispielen die Lösungen der Differentialgleichungen nicht eindeutig bestimmt. Dies ist typisch. Um eindeutige Lösungen erhalten zu können, ist in der Regel die Angabe einer oder mehrerer zusätzlicher Bedingungen erforderlich.



### 4.3 Funktionen mehrerer Veränderlicher, partielle Ableitungen, partielle Differentialgleichungen

Wir betrachten jetzt eine Funktion  $u$ , die von zwei Variablen  $x$  und  $y$  abhängt. Dann können wir die sogenannten partiellen Ableitungen von  $u$  nach  $x$  bzw. von  $u$  nach  $y$  untersuchen. Betrachtet man die Ableitung von  $u(x, y)$  nach  $x$ , nimmt man  $y$  als konstanten Parameter an und kann dann die Ableitung bezüglich der Variablen  $x$  wie gehabt berechnen. Bezüglich der Variablen  $y$  verfährt man genauso. Wir nehmen dabei hier und im folgenden jeweils an, dass die Ableitungen existieren.

#### Beispiele:

1.  $u(x, y) = x$

Dann gilt

$$\begin{aligned} u_x(x, y) &= 1 \\ u_y(x, y) &= 0 \end{aligned}$$

2.  $u(x, y) = xy^2$

Dann ist

$$\begin{aligned} u_x(x, y) &= y^2 \\ u_y(x, y) &= 2xy \end{aligned}$$

und für die zweiten partiellen Ableitungen nach  $x$  und  $y$  gilt

$$\begin{aligned} u_{xx}(x, y) &= 0 \\ u_{yy}(x, y) &= 2x \end{aligned}$$

Als *partielle Differentialgleichungen* bezeichnet man eine Gleichung, in der partielle Ableitungen einer Funktion mehrerer Veränderlicher vorkommen.

#### Beispiele:

1. Für eine Funktion  $u(x, y)$  betrachten wir die partielle DGL

$$u_x(x, y) = 0 \quad .$$

Offensichtlich erfüllt jede Funktion  $u(x, y) = f(y)$  diese partielle DGL.

2. Die partielle DGL

$$u_{xx} + u_{yy} = 0$$

wird z.B. erfüllt von den Funktionen

$$\begin{aligned} u(x, y) &= \text{const.} \\ u(x, y) &= \alpha x + \beta y \quad (\alpha, \beta \in \mathbb{R}) \\ u(x, y) &= \alpha \sin(x) e^y + \beta \cos(x) e^y \end{aligned}$$

und vielen weiteren.

## 3. Mögliche Lösungen der partiellen DGL

$$u_{xx} + u_{yy} = 6xy$$

sind u.a. die Funktionen

$$\begin{aligned} u(x, y) &= x^3y \\ u(x, y) &= xy^3 \\ u(x, y) &= \alpha x^3y + (1 - \alpha)xy^3 + \beta u_{hom} \quad (\alpha, \beta \in \mathbb{R}) , \end{aligned}$$

wobei  $u_{hom}(x, y)$  eine Lösung der Gleichung

$$(u_{hom})_{xx} + (u_{hom})_{yy} = 0$$

ist.

Dies kann man durch einfaches Nachrechnen verifizieren. Auch in diesen Beispielen ist die Lösung der DGL offenbar nicht eindeutig bestimmt.

Partielle Differentialgleichungen können ganz unterschiedliche Eigenschaften haben. Man unterscheidet elliptische, parabolische und hyperbolische Differentialgleichungen. Wir verzichten an dieser Stelle darauf, ein auch nur halbwegs vollständige Einführung in die Theorie und numerische Behandlung partieller Differentialgleichungen zu geben. Wir betrachten stattdessen in den folgenden Abschnitten exemplarisch zwei sogenannte “Randwertaufgaben”, deren Lösung unter geeigneten (natürlichen) Voraussetzungen – auf die wir hier nicht näher eingehen – existiert und eindeutig ist. Bei diesen Randwertaufgaben sind zusätzlich zu den DGLs vorgegebene “Randbedingungen” zu erfüllen.

## 4.4 Diskretisierung (gewöhnlicher und partieller) Randwertaufgaben mit Differenzenverfahren

In diesem Abschnitt stellen wir je eine gewöhnliche und eine partielle Randwertaufgabe vor und diskutieren, wie sich ihre Lösungen näherungsweise numerisch berechnen lassen.

Das erste Beispiel ist eine sehr einfache gewöhnliche Randwertaufgabe (RWA). Wir beschreiben den einfachsten numerischen Lösungsansatz, der auf ein lineares Gleichungssystem mit einer tridiagonalen Matrix führt. Das zweite Beispiel ist eine ebenfalls einfache, aber fundamental wichtige partielle Randwertaufgabe. Ihre effiziente numerische Behandlung ist nicht trivial. Einfache iterative Lösungsverfahren, wie sie in den vorangegangenen Abschnitten beschrieben wurden, können jedoch sehr einfach zur Lösung des bei der Diskretisierung entstehenden linearen Gleichungssystems eingesetzt werden.

### 4.4.1 Eine gewöhnliche Randwertaufgabe 2. Ordnung

Wir betrachten zunächst die gewöhnliche Differentialgleichung

$$u''(x) = f(x) \quad (x \in [0, 1]) \quad (4.3)$$

mit den Randbedingungen

$$u(0) = g_0, \quad u(1) = g_1 \quad . \quad (4.4)$$

Dabei ist die rechte Seite  $f$  eine gegebene Funktion und  $g_0, g_1 \in \mathbb{R}$  sind gegebene Randwerte. Wir suchen eine Funktion  $u(x)$ , welche die Differentialgleichung und die Randbedingungen erfüllt.

**Bemerkung 4.1** (ohne Beweis): Unter geeigneten Voraussetzungen an die Funktion  $f$  ist die oben beschriebene Aufgabe eindeutig lösbar.  $\gg$

**Bemerkung 4.2** (*Physikalische Problemstellung*) Die DGL (4.3) beschreibt physikalisch die Auslenkung  $u(x)$  einer elastischen Saite unter einer Belastung  $f(x)$ .  $\gg$

## 4.4.2 Diskretisierung

Wir wollen die Randwertaufgabe (4.3)-(4.4) nun nicht analytisch lösen (dies geht nur für bestimmte Klassen von Funktionen  $f(x)$ ), sondern eine Näherung für die Lösung  $u(x)$  durch „Diskretisierung“ numerisch berechnen. Dazu benötigen wir ein „Gitter“ und eine diskrete Näherung für die DGL.

Dazu teilen wir das Intervall  $[0, 1]$ , auf dem die DGL (4.3) gegeben ist, in  $n$  äquidistante Teilintervalle  $[x_i, x_{i+1}]$  mit

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n$$

auf. Die Punkte  $x_i$  und die Maschenweite (d.h. die Länge der Teilintervalle)

$$h = \frac{x_n - x_0}{n} = \frac{1}{n}$$

bestimmen ein eindimensionales „Gitter“ (vgl. Abb. 3), auf dem die Lösungsfunktion  $u(x)$  näherungsweise bestimmt werden soll.

$$0 = x_0 \quad | \quad x_1 \quad | \quad x_2 \quad | \quad x_3 \quad | \quad \dots \quad | \quad x_n = 1$$

Abbildung 3: Ein eindimensionales Gitter

Jetzt benötigen wir noch Näherungsformeln für die zweite Ableitung der Funktion  $u(x)$ . Die Idee ist dabei, die Ableitung durch Differenzenquotienten zu ersetzen.

## 4.4.3 Approximation der Differentialgleichung

Um zu einem linearen Gleichungssystem zu gelangen, müssen wir  $u''(x)$  durch lineare Beziehungen von  $u(x)$  ersetzen. Hierzu benutzen wir die Definition der Ableitungen durch den Grenzwert der zugehörigen Differenzenquotienten. Es gilt

$$u''(x) = \lim_{h \rightarrow 0} \frac{1}{h} (u'(x+h) - u'(x)) \quad . \quad (4.5)$$

Ferner können wir benutzen, dass

$$u'(x+h) = \lim_{h \rightarrow 0} \frac{1}{h} (u(x+h) - u(x)) \quad (4.6)$$

und

$$u'(x) = \lim_{h \rightarrow 0} \frac{1}{h} (u(x) - u(x-h)) \quad (4.7)$$

gilt (linksseitige Differenzenquotienten für  $u'(x+h)$  und  $u'(x)$ ). Setzen wir diese Beziehungen in (4.5) ein, so erhalten wir

$$u''(x) = \lim_{h \rightarrow 0} \frac{1}{h^2} (u(x-h) - 2u(x) + u(x+h)) . \quad (4.8)$$

Verzichten wir jetzt auf den Grenzübergang  $h \rightarrow 0$ , so gilt für hinreichend kleines  $h$

$$u''(x) \approx \frac{1}{h^2} (u(x-h) - 2u(x) + u(x+h)) . \quad (4.9)$$

Dabei machen wir zwar einen Fehler, der aber um so kleiner werden wird, je kleiner  $h$  gewählt wird.

#### 4.4.4 Resultierendes Gleichungssystem

Wir benutzen die obige Näherungsformel (4.9) für  $u''(x)$  jetzt in allen inneren Gitterpunkten  $x_i = x_0 + ih$  ( $i = 1, \dots, n-1$ ); in den Randpunkten  $x_0 = 0$  und  $x_n = 1$  setzen wir die gegebenen Randwerte ein. Auf diese Weise erhalten wir ein lineares Gleichungssystem für die Näherungslösung, die wir mit  $u_h(x_i)$  bezeichnen ( $i = 0, 1, \dots, n$ ):

$$\frac{1}{h^2} (u_h(x_{i-1}) - 2u_h(x_i) + u_h(x_{i+1})) = f(x_i) \quad (i = 1, 2, \dots, n-1) \quad (4.10)$$

$$u_h(x_0) = g_0 \quad (4.11)$$

$$u_h(x_n) = g_1 \quad (4.12)$$

Wir verwenden hier bewusst die Notation  $u_h(x_i)$  und nicht  $u(x_i)$ , weil wir bei der Diskretisierung (4.9) auf den Grenzübergang  $h \rightarrow 0$  verzichtet und dadurch einen (für kleines  $h$  kleinen) Fehler eingeführt haben.

Die Unbekannten  $u_h(x_i)$  in dem entstandenen linearen Gleichungssystem (4.10)-(4.12) sind Näherungen für die gesuchte Funktion  $u(x)$  in den Gitterpunkten  $x_i$ . Im Zusammenhang mit dem linearen Gleichungssystem werden diese Unbekannten oft auch kurz als  $u_i$  bezeichnet (d.h.  $u_i := u_h(x_i)$ ). Das Gleichungssystem lautet dann

$$\frac{1}{h^2} (u_{i-1} - 2u_i + u_{i+1}) = f(x_i) \quad (i = 1, 2, \dots, n-1) \quad (4.13)$$

$$u_0 = g_0 \quad (4.14)$$

$$u_n = g_1 \quad (4.15)$$

**Bemerkung 4.3 (Diskretisierungsfehler)** Wir bemerken, dass sich der (globale) “Diskretisierungsfehler”, d.h. der Fehler zwischen der diskreten Näherungslösung  $u_h(x)$  und der exakten Lösung  $u(x)$ , unter geeigneten Voraussetzungen im Wesentlichen wie

$$\text{const} \cdot h^2$$

verhält. Um den Diskretisierungsfehler klein zu machen, muss man die Schrittweite  $h$  also entsprechend klein wählen. Die Anzahl der Unbekannten im diskreten Gleichungssystem wird dann allerdings sehr groß und der Rechenaufwand zur Lösung des zugehörigen Gleichungssystems wächst entsprechend.  $\gg$

#### 4.4.5 Matrixdarstellung

Das Gleichungssystem (4.10)-(4.12) lässt sich in Matrixschreibweise folgendermaßen darstellen:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} = \begin{pmatrix} g_0 \\ h^2 f(x_1) \\ \vdots \\ \vdots \\ h^2 f(x_{n-1}) \\ g_1 \end{pmatrix} \quad (4.16)$$

**Bemerkung 4.4** (*Aufwand zur Lösung des Gleichungssystems*) Da es sich bei der Matrix des Gleichungssystems um eine Tridiagonalmatrix handelt, lässt sich die Lösung mit dem Gaußschen Eliminationsverfahren mit einem Rechenaufwand berechnen, der lediglich proportional zu  $N$  ist. Daher ist für derartige eindimensionale Randwertaufgaben der Einsatz von iterativen Verfahren noch von geringer praktischer Bedeutung.  $\gg$

#### 4.4.6 Konvergenz einfacher iterativer Verfahren zur Lösung des GLS

Löst man das Problem (4.16) mit dem GS-, JAC- oder SOR-Verfahren, so kann man zeigen, dass für die asymptotischen Konvergenzfaktoren pro Iterationsschritt (d.h. für die Faktoren, um die der Fehler asymptotisch pro Iterationsschritt des Verfahrens kleiner wird) gilt:

$$\rho(\text{JAC}) = \cos \pi h \doteq 1 - \frac{1}{2} \pi^2 h^2 \quad (4.17)$$

$$\rho(\text{GS}) = \cos^2 \pi h \doteq 1 - \pi^2 h^2 \quad (4.18)$$

$$\rho(\text{GS}(\omega_{\text{opt}})) = \frac{1 - \sin \pi h}{1 + \sin \pi h} \doteq 1 - 2\pi h \quad (4.19)$$

Dabei ist der optimale Parameter  $\omega_{\text{opt}}$  im SOR-Verfahren in diesem Beispiel

$$\omega_{\text{opt}} = \frac{2}{1 + \sin \pi h} \quad .$$

Für den Stellengewinn  $\mu$  pro Iterationsschritt bedeutet dies asymptotisch

$$\begin{array}{lll} \mu(\text{JAC}) & \doteq & \frac{1}{2} \pi^2 h^2 \quad (\text{Verfahren konvergiert ganz, ganz langsam}) \\ \mu(\text{GS}) & \doteq & \pi^2 h^2 \quad (\text{Verfahren konvergiert ganz langsam}) \\ \mu(\text{GS}(\omega_{\text{opt}})) & \doteq & 2\pi h \quad (\text{Verfahren konvergiert deutlich besser als die ersten beiden}). \end{array}$$

### 4.4.7 Die Poisson-Gleichung mit Dirichletschen Randbedingungen

Wir betrachten jetzt die zu dieser gewöhnlichen Randwertaufgabe analoge partielle Randwertaufgabe in  $\mathbb{R}^2$ :

$$\begin{aligned}\Delta u(x, y) &= f(x, y) & ((x, y) \in \Omega = (0, 1)^2 \subset \mathbb{R}^2) \\ u(x, y) &= g(x, y) & ((x, y) \in \partial\Omega.)\end{aligned}\tag{4.20}$$

Dabei bezeichnet  $\Delta$  den „Laplace-Operator“

$$\Delta u(x, y) = u_{xx}(x, y) + u_{yy}(x, y)$$

Die Funktionen  $f$  und  $g$  sind gegebene Funktionen. Gesucht ist eine Funktion  $u$  welche die „Poisson-Gleichung“

$$\Delta u(x, y) = f(x, y) \text{ in } \Omega$$

und die „Dirichlet-Randbedingungen“

$$u(x, y) = g(x, y) \text{ auf } \partial\Omega$$

erfüllt.

**Bemerkung 4.5** (*Physikalische Bedeutung der Poisson-Gleichung*) Die Poisson-Gleichung ist eine der wichtigsten partiellen Differentialgleichungen. Sie beschreibt eine Vielzahl physikalischer Phänomene, z. B.

- das elektrostatischen Potential,
- das Gravitationspotential,
- die Auslenkung einer Membran unter der Belastung  $f(x)$ ,
- eine stationäre Wärmeleitung,
- das Potential einer stationären inkompressiblen, reibungsfreien Strömung. >>

### 4.4.8 Diskretisierung

Ganz analog zum eindimensionalen (1D) Fall approximieren wir die Differentialgleichung wieder auf einem geeigneten Gitter. Dazu überdecken wir das abgeschlossene Einheitsquadrat  $\bar{\Omega} = [0, 1]^2$  durch ein Gitter. Mit der Schrittweite  $h = 1/n$  erhalten wir die Gitterpunkte

$$\bar{\Omega}_h = \{(x_i, y_j) | (x_i, y_j) = (ih, jh); i, j = 0, 1, \dots, n\} \subset \bar{\Omega}\tag{4.21}$$

(vgl. Abb 4).

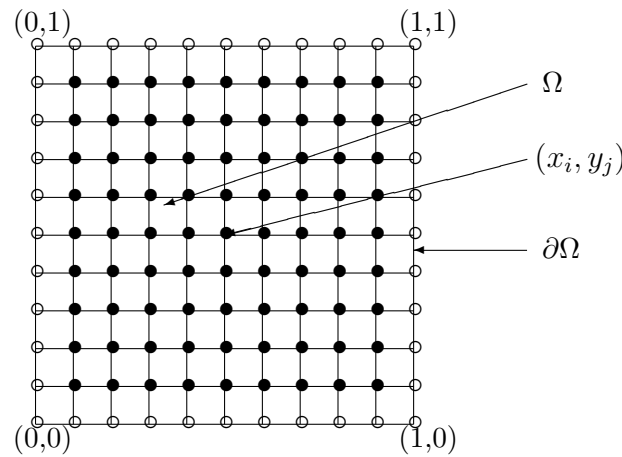


Abbildung 4: Diskretisierungsgitter im 2D Fall

#### 4.4.9 Approximation von $\Delta u$

Analog zum eindimensionalen Fall approximieren wir die partiellen Ableitungen  $u_{xx}(x, y)$ ,  $u_{yy}(x, y)$  durch die Näherungsformeln

$$\begin{aligned} u_{xx}(x, y) &\approx \frac{1}{h^2} (u(x-h, y) - 2u(x, y) + u(x+h, y)) \\ u_{yy}(x, y) &\approx \frac{1}{h^2} (u(x, y-h) - 2u(x, y) + u(x, y+h)) . \end{aligned} \quad (4.22)$$

Damit erhalten wir den diskreten Laplace-Operator

$$\Delta_h u(x, y) = \frac{1}{h^2} (u(x-h, y) + u(x+h, y) + u(x, y-h) + u(x, y+h) - 4u(x, y)) . \quad (4.23)$$

Abb. 5 veranschaulicht, von welchen Gitterpunkten Beiträge zur Berechnung des diskreten Laplace-Operators geleistet werden. Hierdurch motiviert verwenden wir auch die kompakte Schreibweise

$$\Delta_h u(x, y) = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} u(x, y) .$$

Diese kompakte Schreibweise ist äquivalent zu (4.23), sie betont aber den Zusammenhang zwischen diskreter und kontinuierlicher Randwertaufgabe, und sie macht auf einen Blick deutlich, welche Punkte des Gitters und welche zugehörigen Unbekannten an der Diskretisierung in einem Punkt beteiligt sind. Sie hat viele Vorteile, auch z. B. für die Programmierung.

#### 4.4.10 Aufstellung des zugehörigen Gleichungssystems

Hiermit können wir jetzt ein lineares Gleichungssystem für die diskrete Lösung

$$u_h(x, y) , (x, y) = (x_i, y_j) \in \overline{\Omega}_h$$

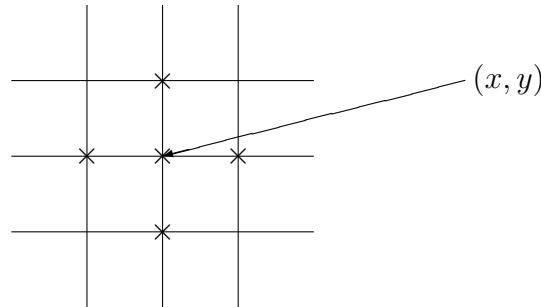


Abbildung 5: Geometrisches Muster des diskreten Laplace-Operators

aufstellen. Diese Lösung besteht aus  $(n+1)^2$  Komponenten (entsprechend den  $(n+1)^2$  Gitterpunkten von  $\bar{\Omega}_h$ ). Wie im 1D Fall unterscheiden wir wiederum die inneren Gitterpunkte

$$\Omega_h = \{(x, y) = (x_i, y_j) = (ih, jh) : i, j = 1, \dots, n-1\} \quad (4.24)$$

und die Rand-Gitterpunkte

$$\Gamma_h = \bar{\Omega}_h \setminus \Omega_h$$

(vgl. Abb. 4, dort sind die Punkte von  $\Omega_h$  durch  $\bullet$  und die Punkte von  $\Gamma_h$  durch  $\circ$  gekennzeichnet.) Das entstehende Gleichungssystem für die diskrete Lösung können wir dann in der Form

$$\begin{aligned} \Delta_h u_h &= f(x, y) & \text{für } (x, y) \in \Omega_h \\ u_h(x, y) &= g(x, y) & \text{für } (x, y) \in \Gamma_h \end{aligned} \quad (4.25)$$

schreiben. Dieses Gleichungssystem bezeichnen wir in Zukunft auch als unser *Modellproblem*.

Bei der Programmierung von effizienten Lösungsverfahren für dieses und ähnliche Probleme geht man häufig direkt von dem geometrischen Hintergrund aus. Die Unbekannten und die Gleichungen sitzen in den Punkten des Gitters. Die Matrixdarstellung des Gleichungssystems wird dann gar nicht gebraucht.

#### 4.4.11 Diskretisierungsgenauigkeit

Ohne Beweis erwähnen wir hier, dass sich der Diskretisierungsfehler, d.h. der Fehler zwischen der diskreten Näherungslösung  $u_h(x)$  und der exakten Lösung  $u(x)$  der Randwertaufgabe, unter geeigneten Voraussetzungen im Wesentlichen wie

$$\text{const} \cdot h^2$$

verhält. Um den Diskretisierungsfehler klein zu machen, muss man die Schrittweite  $h$  also auch hier entsprechend klein wählen. Die Anzahl der Unbekannten in unserem Gleichungssystem und die Anzahl der Gleichungen wird dann allerdings sehr groß (sie wächst viel schneller als im eindimensionalen Fall) und der Rechenaufwand zur Lösung des Gleichungssystems wächst entsprechend.



#### 4.4.12 Matrixdarstellung

Man kann natürlich auch wieder zu einer *Matrixdarstellung* dieses Gleichungssystems übergehen. Dann hängt die Form der Matrix davon ab, in welcher Nummerierung man (auf der linken und rechten Seite des obigen Systems) die Gitterpunkte  $(x, y) = (x_i, y_j)$  durchläuft. Man kann z.B. die Gitterpunkte zeilenweise, beginnend von links unten nach rechts oben nummeriert. Für  $n = 8$  erhält man:

$$\begin{array}{ccccccc} 43 & 44 & 45 & 46 & 47 & 48 & 49 \\ 36 & 37 & 38 & 39 & 40 & 41 & 42 \\ 29 & 30 & 31 & 32 & 33 & 34 & 35 \\ 22 & 23 & 24 & 25 & 26 & 27 & 28 \\ 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

#### 4.4.13 Matrixdarstellung bei zeilenweiser Nummerierung

Wir betrachten hier nur die Struktur, die sich bei zeilenweiser Nummerierung ergibt. Die Randbedingungen sind dabei schon eliminiert worden. Im Falle  $n = 8$  ergibt sich die folgende Blockstruktur:

$$A_N = \begin{pmatrix} B & I & 0 & 0 & 0 & 0 & 0 \\ I & B & I & 0 & 0 & 0 & 0 \\ 0 & I & B & I & 0 & 0 & 0 \\ 0 & 0 & I & B & I & 0 & 0 \\ 0 & 0 & 0 & I & B & I & 0 \\ 0 & 0 & 0 & 0 & I & B & I \\ 0 & 0 & 0 & 0 & 0 & I & B \end{pmatrix}$$

$$B = \begin{pmatrix} -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 \end{pmatrix}$$

$A_N$  ist eine tridiagonale Block-Matrix. Dabei ist  $B$  selbst wieder eine  $(7 \times 7)$ -Tridiagonalmatrix, während  $I$  die  $(7 \times 7)$ -Einheitsmatrix ist. Die Matrix  $A_N$  hat offensichtlich eine sehr spezielle Struktur. Man kann sie z. B. als eine **Bandmatrix** interpretieren.

#### 4.4.14 Interpretation als Bandmatrix

Bei der Interpretation als Bandmatrix könnte man daran denken, das Eliminationsverfahren an diese Form der Matrix anzupassen. Die Bandbreite von  $A_N$  wäre im vorliegenden Fall

$$m = 2n - 1$$

und der Rechenaufwand wäre proportional zu

$$m^2 N = (2n - 1)^2 N \sim N^2$$

Gegenüber dem Rechenaufwand für eine vollbesetzte Matrix ( $\sim N^3$ ) wäre dies zwar schon ein deutlicher Gewinn; für große Werte von  $n$  ergibt sich aber trotzdem eine unakzeptable Rechenzeit. Ein Nachteil der Interpretation als Bandmatrix ist, daß die inneren Nullen durch dieses Verfahren nicht ausgenutzt werden. Sie füllen sich innerhalb der Bandbreite mit Werten  $\neq 0$  auf.

#### 4.4.15 Einfache Verfahren zur Lösung des diskreten linearen Gleichungssystems

Zur Lösung des bei der Diskretisierung der zweidimensionalen RWA entstandenen Gleichungssystems kann z.B. das GS-Verfahren oder auch das SOR-Verfahren eingesetzt werden. Beide Verfahren lassen sich sehr einfach programmieren, wenn man in der Initialisierungsphase neben der rechten Seite des Gleichungssystems in den inneren Punkten des Gitters auch bereits die exakten Randwerte für die diskrete Näherungslösung abspeichert. Hingegen sollte man darauf verzichten, die Matrix des GLS abzuspeichern. Die Gleichungen sehen in jedem Gitterpunkt gleich aus, und die Programmierung ist wesentlich leichter, wenn man sich Schleifen über die Unbekannten oder über die Gleichungen im Innern des Gitters als Schleifen über die inneren Gitterpunkte vorstellt.

Die Aussagen zur Konvergenz dieser Verfahren für den eindimensionalen Fall in Abschnitt 4.4.6 übertragen sich direkt auf den zweidimensionalen Fall.

Das Gaußsche Eliminationsverfahren ist hier dem SOR-Verfahren bei großen Anzahlen von Unbekannten schon deutlich unterlegen (vgl. die Rechenzeiten in Tab. 1). Man beachte, dass rund 1 Mio. Unbekannte im zweidimensionalen Fall bereits bei  $h = 1/1000$  entstehen.

Zur Lösung der bei der Diskretisierung von Randwertaufgaben entstehenden Gleichungssysteme können Verfahren wie Gauss-Seidel oder das SOR-Verfahren eingesetzt werden. Sie sind aber meist nicht besonders effizient.

Es gibt viel schnellere iterative Verfahren, z.B. die sogenannten (iterativen) Mehrgitterverfahren, die selber wiederum Bausteine wie das Gauss-Seidel-Verfahren benutzen.

Noch schneller sind die sogenannten Full-Multigrid-Versionen der Mehrgitterverfahren, die wiederum den direkten Verfahren zuzuordnen sind. Sie hier detailliert zu besprechen würde jedoch den Rahmen dessen sprengen, was in der Schule möglich ist.

Dennoch haben wir bei den Programmen ein derartiges Full-Multigrid-Programm zu Demonstrationszwecken beigelegt (vgl. Abschnitt 5.2.8). Es zeigt, dass man mit effizienten Algorithmen auch Gleichungssysteme mit Hunderttausenden von Unbekannten in Sekundenschnelle lösen kann.

# Kapitel 5

## Unterrichtsmaterialien

### 5.1 Foliensätze

#### 5.1.1 Foliensatz zu Anwendungen großer Gleichungssysteme und zur Motivation

Die Datei *schnelle\_loesung\_gls\_motivation.pdf* enthält einen einführenden Foliensatz, der zwei Zielsetzungen verfolgt. Zum einen soll den Schüler/innen ein Einblick in die vielfältigen Anwendungsfelder gegeben werden, bei denen große Gleichungssysteme gelöst werden müssen. Das Beispiel der Wettervorhersage wird dabei etwas ausführlicher behandelt.

Zum anderen soll auch deutlich gemacht werden, dass es viele unterschiedliche Verfahren zum Lösen großer Gleichungssysteme gibt, die sich auch im Hinblick auf den Rechenaufwand und die Rechenzeiten um Größenordnungen unterscheiden. Gleich die Folie *Algorithmen versus Hardware* veranschaulicht, dass es nicht nur auf Seiten der Informatik (durch die Entwicklung immer schnellerer Computer), sondern auch auf Seiten der Mathematik (durch die Entwicklung immer effizienterer Algorithmen) in den letzten 30 Jahren erhebliche Fortschritte gegeben hat, die gemeinsam dafür sorgen, dass heute standardmäßig Problemstellungen gelöst werden können, die vor 30 Jahren noch riesige Rechenzeiten benötigt hätten. Diese Folie kann auch bei der Besprechung des Arbeitsblattes *Fortschritte durch Mathematik und Rechnerentwicklung in den letzten 30 Jahren* zum Einsatz kommen.

#### 5.1.2 Foliensatz zur Einführung in das Lösen großer Gleichungssysteme

Die Datei *schnelle\_loesung\_gls.pdf* enthält einen Foliensatz, der die wesentlichen mathematischen Inhalte des vorliegenden Dokuments aufbereitet und der im Unterricht eingesetzt werden kann.

Behandelt werden

- das Gaußsche Eliminationsverfahren,
- Jacobi-, Gauß-Seidel- und SOR-Verfahren als Beispiele für verschiedene einfache iterative Verfahren (der Begriff eines iterativen Verfahrens wird dabei mit dem Heron-Verfahren zur Wurzelberechnung eingeführt),
- der Ableitungsbegriff (Definition über Steigung der Tangente bzw. durch den Grenzwert des Differenzenquotienten),
- der Begriff der Differentialgleichung sowie
- die Diskretisierung von Randwertaufgaben, die direkt zu der Problemstellung großer Gleichungssysteme führt.

## 5.2 Computerprogramme

Alle beigegeführten Computerprogramme sind in der Programmiersprache Python geschrieben (Version 2.6.x oder 2.7.x). Python ist frei verfügbar und kann über die Web-Seite [www.python.org](http://www.python.org) frei heruntergeladen werden. Dort finden sich auch entsprechende Tutorials etc.

Python wurde mit dem Ziel entwickelt, das Programmieren für den Programmierer möglichst einfach zu machen. Daher kann man Python mit relativ wenig Aufwand so weit lernen, dass man alle in der Schule vorkommenden Algorithmen problemlos programmieren kann. Wir haben dies mehrfach mit (heterogenen) Gruppen von Schülerpraktikanten der Klassen 9 - 13 erprobt und sehr gute Erfahrungen mit einer zwei- bis dreitägigen Einführung in Python gemacht.

Auf Wunsch können wir auch entsprechendes Material zur Verfügung stellen.

Python ist nicht die schnellste Programmiersprache und nach unseren Erfahrungen in der Ausführung etwa einen Faktor 30 langsamer als entsprechende optimierte C-Programme. Der Aufwand, C oder eine vergleichbare Programmiersprache zu lernen oder einen komplizierten Algorithmus in C zu programmieren, ist dafür aber wesentlich höher als bei Python.

### 5.2.1 Computerprogramm: Rechenzeiten bei Determinantenberechnung nach dem Laplaceschen Entwicklungssatz

Das Programm *detcalc\_entwicklungssatz\_rechenzeiten.py* berechnet die Determinanten von  $n \times n$  Matrizen ( $n = 1, \dots, 10$ ), die vorab mit Zufallszahlen zwischen 0 und 1 besetzt werden, nach dem Laplaceschen Entwicklungssatz. Die Rechenzeiten für die Berechnung der Determinanten werden mit ausgegeben.

Mit diesem Programm kann der Lehrer demonstrieren oder die Schüler/innen können selbst ausprobieren, wie dramatisch ein Rechenaufwand, der proportional zu  $n!$  ist, anwächst.

Eine typische Ausgabe des Programms kann dann folgendermaßen aussehen:

n	time	det
1	0.000	0.903260
2	0.000	0.007563
3	0.000	0.127612
4	0.000	-0.012200
5	0.000	0.059021
6	0.015	0.024155
7	0.094	0.041348
8	0.453	-0.003574
9	4.016	0.008393
10	39.578	0.015875

Was wäre für  $n=11$  zu erwarten? Antwort: etwa  $11 \cdot 40 = 440 \text{ sec} = 7 \text{ min}$ ; für  $n=12$  entsprechend  $12 \cdot 7 \text{ min} = 84 \text{ min}$ !

Will man Determinanten von Matrizen mit  $n > 10$  berechnen, beispielsweise, um dies zu überprüfen, so reicht es im zweiten Befehl des Hauptprogramms

```
for n in range(1,11):
```

die Zahlen 1 und 11 zu ändern. Die 1 ist dann durch die erste Zahl zu ersetzen, die als kleinste Dimension einer Matrix gewählt werden soll, die Zahl anstelle von 11 muss um 1 größer gewählt werden als die Zahl, die als größte Dimension gewählt werden soll.

Will man also die Determinante einer  $11 \times 11$ -Matrix berechnen, so müsste dieser Befehl lauten:

```
for n in range(11,12):
```

Danach speichert man das modifizierte Programm ab und kann es dann laufen lassen.

### 5.2.2 Computerprogramm: Rechenzeiten bei Determinantenberechnung durch Transformation auf Dreiecksgestalt

Das Computerprogramm *detcalc\_trafo\_dreiecksmatrix\_rechenzeiten.py* berechnet die Determinanten von  $n \times n$  Matrizen ( $n=1, \dots, 100$ ), die vorab mit Zufallszahlen zwischen 0 und 1 besetzt werden, durch Transformation der Matrix auf Dreiecksgestalt und Produktbildung der entstehenden Diagonalelemente. Die Rechenzeiten für die Berechnung der Determinanten werden mit ausgegeben.

Will man Determinanten noch größerer Matrizen berechnen, reicht es im untenstehenden Befehl

```
for n in range(1,101):
```

die Zahlen 1 und 101 zu ändern. Die 1 ist durch die erste Zahl zu ersetzen, die als kleinste Dimension einer Matrix gewählt werden soll, die Zahl anstelle von 101 muss um 1 größer gewählt werden als die Zahl, die als größte Dimension gewählt werden soll.

Mit diesem Programm kann der Lehrer demonstrieren oder die Schüler/innen können selbst ausprobieren, wie ein Rechenaufwand, der proportional zu  $n^3$  ist, anwächst: deutlich langsamer als beim Programm, das im vorangegangenen Abschnitt beschrieben wurde.

Man beachte auch den ganz erheblichen Speicherplatzbedarf für die Matrix: Bei  $n = 1000$  sind  $1000 \times 1000 = 1$  Mio. Werte zu speichern. Das Programm stößt dann aus Speicherplatzgründen irgendwann an seine Grenzen.

### 5.2.3 Computerprogramm zur Determinantenberechnung durch Transformation auf Dreiecksgestalt

Das Computerprogramm *detcalc.py* berechnet die Determinante einer  $n \times n$ -Matrix, wobei die Koeffizienten der Matrix vom Benutzer eingegeben werden.

Mit diesem Programm können Lehrer und/oder Schüler/innen die Determinante einer gegebenen Matrix berechnen.

### 5.2.4 Computerprogramm zum Rechenzeitverhalten des Gaußschen Eliminationsverfahrens bei der Lösung linearer Gleichungssysteme

Das Computerprogramm *gauss\_elimination\_rechenzeiten.py* löst lineare Gleichungssystem von  $n$  Gleichungen mit  $n$  Unbekannten ( $n = 50, 100, 150, \dots, 400$ ), wobei die Koeffizienten vorab mit Zufallszahlen zwischen 0 und 1 besetzt werden. Die Rechenzeiten für die Lösungen der Gleichungssysteme werden ausgegeben.

Will man noch größere Gleichungssysteme lösen, reicht es im Befehl

```
for n in range(50,401,50):
```

die Zahlen 50, 401 und 50 zu ändern. Die erste 50 ist durch die erste Zahl zu ersetzen, die als kleinste Dimension einer Matrix gewählt werden soll, die Zahl anstelle von 401 muss um 1 größer gewählt werden als die Zahl, die als größte Dimension gewählt werden soll; die zweite 50 stellt das Inkrement dar und kann nach Bedarf ebenfalls modifiziert werden.

Mit diesem Programm kann der Lehrer demonstrieren oder die Schüler/innen können selbst ausprobieren, wie ein Rechenaufwand, der proportional zu  $n^3$  ist, anwächst: deutlich langsamer als beim Programm, das im vorangegangenen Abschnitt beschrieben wurde.

Man beachte den ganz erheblichen Speicherplatzbedarf für die Matrix: Bei  $n = 1000$  sind  $1000 \times 1000 = 1$  Mio. Werte zu speichern. Das Programm stößt dann aus Speicherplatzgründen irgendwann an seine Grenzen.

### 5.2.5 Computerprogramm: Gaußsches Eliminationsverfahren zur Lösung eines linearen Gleichungssystems

Das Computerprogramm *gauss\_elimination\_loesung\_gls.py* löst ein lineares Gleichungssystem von  $n$  Gleichungen mit  $n$  Unbekannten, wobei die Koeffizienten und die rechte Seite des Gleichungssystems vom Benutzer eingegeben werden.

Mit diesem Programm können Lehrer und/oder Schüler/innen gegebene reguläre lineare Gleichungssysteme lösen.

### 5.2.6 Computerprogramm: Gauss-Seidel-Verfahren zur Lösung (diagonaldominanter) linearer Gleichungssysteme

Das Computerprogramm *gauss\_seidel\_iteration.py* löst ein (diagonaldominantes) lineares Gleichungssystem von  $n$  Gleichungen mit  $n$  Unbekannten, wobei die Koeffizienten und die rechte Seite des Gleichungssystems vom Benutzer eingegeben werden.

Vor der Lösungsphase analysiert das Programm, ob das starke Zeilensummenkriterium erfüllt ist und gibt ggf. den Faktor aus, um den der Fehler pro Iterationsschritt mindestens sinkt. Ist das Kriterium nicht erfüllt, weist das Programm darauf hin, dass die Konvergenz des Gauss-Seidel-Verfahrens nicht gesichert ist. Trotzdem kann das Gauss-Seidel-Verfahren auch dann evtl. noch konvergieren. Es ist dann aber besonders wichtig, zu prüfen, ob das Residuum (der Defekt) auch hinreichend klein geworden ist.

Mit diesem Programm können Lehrer und/oder Schüler/innen gegebene (diagonaldominante) lineare Gleichungssysteme lösen. Man kann mit ihm auch experimentell überprüfen, wie stark die Diagonaldominanz des Gleichungssystems die Konvergenz des Verfahrens tatsächlich beeinflusst. Der Faktor, den das starke Zeilensummenkriterium als obere Schranke für die Fehlerreduktion liefert, ist in vielen Fällen tatsächlich viel zu pessimistisch.

### 5.2.7 Computerprogramm: Gauss-Seidel- und SOR-Verfahren zur numerischen Lösung der zweidimensionalen Poissonschen Randwertaufgabe

Das Computerprogramm *gs\_sor\_2d\_poisson\_rwa.py* löst die zweidimensionale Poissonsche Randwertaufgabe auf dem Einheitsquadrat mit dem Gauss-Seidel- bzw. SOR-Verfahren. Eingabeparameter sind

- nint: die Anzahl der Intervalle in x- und y-Richtung
- niter: die maximale Anzahl an Iterationsschritten
- epsilon: die Genauigkeit, mit der das diskrete GLS gelöst werden soll; die Iterationen sollen abbrechen, sobald Defekt/Residuum kleiner oder gleich epsilon sind.
- omega: Relaxationsparameter ( $0 < \omega < 2$ );  $\omega = 1$  liefert das Gauss-Seidel-Verfahren

Dieses Programm kann für mehrere Zwecke eingesetzt werden:

- Die Schüler/innen können experimentell untersuchen, wie das Konvergenzverhalten des Gauss-Seidel-Verfahrens von der Schrittweite  $h = 1/\text{nint}$  abhängt, indem sie für verschiedene Schrittweiten untersuchen, wieviele Iterationsschritte erforderlich sind, um eine feste Genauigkeit (festes epsilon) zu erreichen.
- Die Schüler/innen können anhand des Programms erkennen, dass Residuum/Defekt zwar bis in den Rundungsfehlerbereich sinken, der maximale Fehler im Vergleich zur Lösung der kontinuierlichen Randwertaufgabe, d.h. der durch die Diskretisierung eingeführte Modellfehler durch Vernachlässigung des Grenzübergangs für  $h \rightarrow 0$ , aber vorher erreicht wird.

- Die Schüler/innen können experimentell ermitteln, wie dieser Diskretisierungsfehler von der Schrittweite  $h = 1/n_{\text{int}}$  abhängt.
- Die Schüler/innen können experimentell ermitteln, für welche Werte des Relaxationsparameters  $\omega$  das Verfahren die schnellste Fehlerreduktion liefert. Sie können auch erkennen, dass dieses optimale  $\omega$  von der Schrittweite  $h = 1/n_{\text{int}}$  abhängt.

### 5.2.8 Computerprogramm: Full-Multigrid-Verfahren zur numerischen Lösung der zweidimensionalen Poissonschen Randwertaufgabe

Das Computerprogramm *full\_multigrid\_2d\_poisson\_rwa.py* löst die zweidimensionale Poissonsche Randwertaufgabe auf dem Einheitsquadrat mit einem Full-Multigrid-Verfahren. Dieses Programm startet auf einem groben Gitter bestehend aus  $2 \times 2$  Intervallen (d.h.  $3 \times 3$  Gitterpunkten) und geht dann sukzessiv auf immer feinere Gitter über, bis ein Gitter mit  $1024 \times 1024$  Intervallen (d.h.  $1025 \times 1025 = 1050625$  Gitterpunkten) erreicht ist. Auf jedem Gitter wird der Fehler der auf diesem Gitter berechneten Näherungslösung sowie die zur Berechnung benötigte Zeit ausgegeben.

Dieses Programm demonstriert eindrucksvoll, wie effizient dieses Mehrgitterverfahren im Vergleich zu Gauss-Seidel- oder auch SOR-Verfahren ist (vgl. Abschnitt 5.2.7).

Man kann auch erkennen, dass die Rechenzeit beim Übergang auf ein feineres Gitter um nicht mehr als den Faktor 4 wächst, um den auch die Anzahl der Unbekannten steigt. Ferner kann man wieder erkennen, dass der durch die Diskretisierung bedingte Fehler bei einem Übergang auf ein Gitter mit halber Schrittweite um den Faktor 4 sinkt.

## 5.3 Arbeitsblätter



## Arbeitsblatt: Große Gleichungssysteme und Wettervorhersage

In der Meteorologie werden sehr leistungsfähige Computer eingesetzt, um das Wetter der kommenden Tage zu berechnen. Dabei geht man in einzelnen kurzen Zeitschritten voran. Insgesamt werden heute für eine 10-tägige-Wettervorhersage rund 6500 Zeitschritte benötigt. In jedem Zeitschritt ist neben vielen anderen Berechnungen auch jeweils ein Gleichungssystem bestehend aus 16 Mio. Gleichungen mit 16 Mio. Unbekannten zu lösen.

Für die Lösung von Gleichungssystemen steht heute eine Vielzahl verschiedener Verfahren zur Verfügung. Tab. 1 zeigt zwei Eigenschaften einiger bekannter Verfahren für Probleme, wie sie bei der Wettervorhersage auftreten: die erforderliche Anzahl an Rechenoperationen in Abhängigkeit von den Unbekannten sowie die Rechenzeit für die Lösung eines Gleichungssystems mit rund 1 Mio. Unbekannten. So ist beispielsweise der Rechenaufwand des Gaußschen Eliminationsverfahrens hier proportional zum Quadrat der Anzahl  $N$  der Unbekannten, der des SOR-Verfahrens proportional zu  $N^{1,5} = N \cdot \sqrt{N}$  und der eines Mehrgitterverfahrens proportional zu  $N$ .

Dies bedeutet beispielsweise: Löst man mit dem Gaußschen Eliminationsverfahren ein 3-mal so großes Gleichungssystem wie in der Tabelle, so ist die erforderliche Rechenzeit  $3^2 = 9$  mal so hoch (d.h.  $9 \cdot 14\text{h} = 126\text{h}$ ), denn der Rechenaufwand wächst proportional zum Quadrat der Anzahl der Unbekannten.

Verfahren	Anzahl der Rechenoperationen	Rechenzeit
Gaußsches Eliminationsverfahren für Bandmatrizen	$\sim N^2$	14 h
Überrelaxationsverfahren, SOR-Verfahren (1960)	$\sim N^{1,5}$	5 min
Mehrgitter-Verfahren (1980)	$\sim N$	1 sec

Tabelle 1: Rechenaufwand verschiedener Verfahren und zugehörige Rechenzeiten bei der Lösung eines Gleichungssystems mit rund 1 Million Unbekannten auf einem normalen PC.

### Aufgaben:

1. Wie lang ist ein Zeitschritt bei einer 10-Tage-Wetterprognose?
2. Wie lange brauchen die in Tab. 1 angegebenen Verfahren zur Lösung eines Gleichungssystems mit
  - a) 2 Mio.
  - b) 4 Mio.
  - c) 8 Mio.
  - d) 16 Mio.Unbekannten? Beachte dabei, dass die Rechenzeiten bei allen angegebenen Verfahren außer dem Mehrgitterverfahren schneller wachsen als die Anzahl der Unbekannten. (Zur Kontrolle bei d): Gauß-Verfahren rund 5 Monate, SOR-Verfahren über 5 Stunden, Mehrgitter-Verfahren 16 sec)  
Was fällt dir auf, wenn du das Anwachsen der Rechenzeiten für die verschiedenen Verfahren miteinander vergleichst?
3. Wie lange würde ein handelsüblicher PC allein zur Lösung der in einer 10-tägigen Wettervorhersage auftauchenden Gleichungssysteme benötigen, wenn
  - a) das SOR-Verfahren
  - b) das Mehrgitter-Verfahreneingesetzt würde? Benutze die Kontrollangaben in der vorherigen Aufgabe. Interpretiere die Ergebnisse.

**Arbeitsblatt: Große Gleichungssysteme und Wettervorhersage****Lösungen:**

1. Die Länge eines Zeitschritts beträgt

$$\frac{10 \text{ Tage}}{6500} = \frac{10 \cdot 24 \cdot 3600 \text{ sec}}{6500} \approx 132 \text{ sec} ,$$

also etwas über 2 Minuten.

2. a) Gauss:  $2^2 \cdot 14 \text{ h} = 56 \text{ h}$ , also über 2 Tage  
 SOR:  $2 \cdot \sqrt{2} \cdot 5 \text{ min} \approx 14 \text{ min}$   
 Mehrgitter:  $2 \cdot 1 \text{ sec} = 2 \text{ sec}$
- b) Gauss:  $4^2 \cdot 14 \text{ h} = 224 \text{ h}$ , also über 9 Tage  
 SOR:  $4 \cdot \sqrt{4} \cdot 5 \text{ min} = 40 \text{ min}$   
 Mehrgitter:  $4 \cdot 1 \text{ sec} = 4 \text{ sec}$
- c) Gauss:  $8^2 \cdot 14 \text{ h} = 896 \text{ h}$ , also über 37 Tage  
 SOR:  $8 \cdot \sqrt{8} \cdot 5 \text{ min} \approx 113 \text{ min}$   
 Mehrgitter:  $8 \cdot 1 \text{ sec} = 8 \text{ sec}$
- d) Gauss:  $16^2 \cdot 14 \text{ h} = 3584 \text{ h}$ , also über 149 Tage (etwa 5 Monate!)  
 SOR:  $16 \cdot \sqrt{16} \cdot 5 \text{ min} = 320 \text{ min}$   
 Mehrgitter:  $16 \cdot 1 \text{ sec} = 16 \text{ sec}$

Beim Vergleich der Rechenzeiten fällt auf, dass der Unterschied zwischen den einzelnen Verfahren immer größer wird. Bei 16 Mio. Unbekannten benötigt das Gauss-Verfahren nahezu 5 Monate Rechenzeit im Vergleich zu lediglich 16 sec beim Mehrgitter-Verfahren. Das SOR-Verfahren liegt mit 320 min noch in einem erträglichen Bereich, ist aber um einen Faktor 1200 langsamer als das Mehrgitter-Verfahren.

3. a) SOR:  $6500 \cdot 320 \text{ min} = 2080000 \text{ min} \approx 34666 \text{ h} \approx 1444 \text{ Tage} \approx 4 \text{ Jahre}$   
 bzw. bei Verwendung der Kontrollangaben:  
 $6500 \cdot 320 \text{ min} = 1950000 \text{ min} \approx 32500 \text{ h} \approx 1354 \text{ Tage} \approx 3,7 \text{ Jahre} \approx 3 \text{ Jahre } 8 \text{ Monate}$
- b) Mehrgitter:  $6500 \cdot 16 \text{ sec} = 104000 \text{ sec} \approx 1733 \text{ min} \approx 29 \text{ h}$

Selbst das Mehrgitter-Verfahren benötigt auf einem handelsüblichen PC mehr als einen Tag, um allein die in der 10-Tages-Wettersimulation auftretenden Gleichungssysteme zu lösen. Da in der Praxis aber wesentlich leistungsfähigere Hochleistungsrechner eingesetzt werden, sind die auftretenden Berechnungen heute in akzeptabler Zeit möglich, wenn optimale Verfahren eingesetzt werden. Die Rechenzeiten des SOR-Verfahrens sind im Vergleich dazu noch viel zu hoch.

## Arbeitsblatt: Fortschritte durch Mathematik und Rechnerentwicklung in den letzten 30 Jahren

Abb. 1 zeigt anhand eines konkreten Beispiels die Fortschritte bei der Lösung großer Gleichungssysteme, die in 25 Jahren einerseits durch die Angewandte Mathematik in Form immer schnellerer Algorithmen (Rechenverfahren), andererseits durch immer schnellere Rechner erreicht worden sind. Das zugrundeliegende Beispiel ist ein Gleichungssystem mit weit mehr als einer Million Unbekannten, das bei der numerischen Lösung der Wärmeleitungs- oder Diffusionsgleichung auftritt.

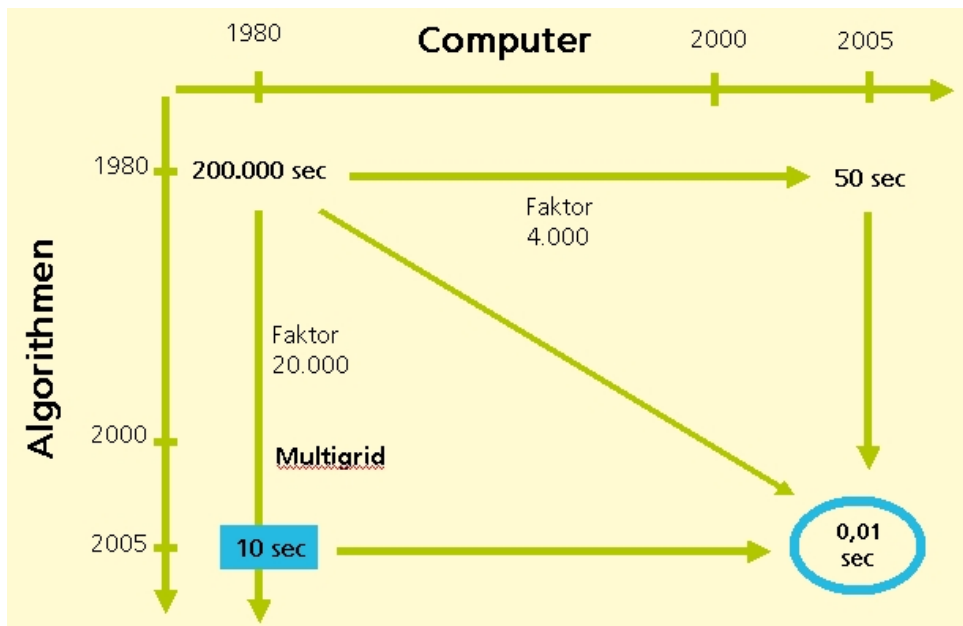


Abbildung 1: Kleinere Rechenzeiten durch Mathematik und durch schnellere Computer.

### Aufgaben:

1. Was ist auf der horizontalen Achse (d.h. der  $x$ -Achse), was auf der vertikalen Achse aufgetragen? Worin unterscheiden sich beide Achsen?
2. Wie lange dauerte die Lösung des der Abbildung zugrundeliegenden Gleichungssystems 1980 auf dem damals schnellsten Rechner und mit den damals bekannten Algorithmen?
3. Wie lange dauert die Lösung dieses Problems auf dem 2005 schnellsten Rechner bei Verwendung der 1980 bekannten Algorithmen?
4. Wie lange würde die Lösung des Problems auf dem Rechner von 1980 mit den 2005 bekannten Algorithmen dauern?
5. Wie lange dauert die Lösung des Problems mit den Algorithmen und Rechnern von 2005?
6. Vergleiche die Beschleunigungsfaktoren auf der Seite der Mathematik und auf der Seite der Rechnerentwicklung.
7. Welche Rechenzeit könnte man erwarten, wenn man die Rechenzeit von 1980 durch das Produkt der Beschleunigungsfaktoren durch Algorithmen und Rechnerentwicklung dividiert? Was fällt dir auf?

**Arbeitsblatt: Fortschritte durch Mathematik und Rechnerentwicklung Infos für Lehrkräfte:** Die Datei *schnelle.loesung\_gls\_motivation.pdf* enthält einen Foliensatz mit der Folie *Algorithmen versus Hardware*, die bei der Besprechung dieses Arbeitsblattes verwendet werden kann (vgl. Abschnitt 5.1.1).

### Lösungen:

1. Sowohl die horizontale als auch die vertikale Achse sind Achsen, auf denen die Zeit aufgetragen ist. Die horizontale Achse beschreibt Fortschritte durch schnellere Computer, die vertikale Fortschritte durch schnellere Algorithmen.
2. 1980 dauerte die Lösung des GLS  $200.000 \text{ sec} \approx 55 \text{ h}$ .
3. Auf dem 2005 schnellsten Rechner dauert die Lösung mit den im Jahr 1980 schnellsten bekannten Algorithmen 50 sec.
4. Auf dem Rechner von 1980 würde die Lösung des GLS mit dem 2005 schnellsten Verfahren 10 sec dauern.
5. Heute ist das Problem in 0,01 sec lösbar.
6. Der Beschleunigungsfaktor durch die Mathematik beträgt für dieses Beispiel 20.000, der durch die Computerentwicklung 4.000. Die Beschleunigung durch die Mathematik ist daher für das vorliegende Beispiel um einen Faktor 4 größer.
7. Multipliziert man die beiden Faktoren 4.000 und 20.000, so erhält man das Produkt 80.000.000 .  
 $200.000 \text{ sec} : 80.000.000 = 0.0025 \text{ sec}$ . Man könnte also eine um einen Faktor 4 niedrigere Rechenzeit erwarten, wenn die Beschleunigung von Algorithmen und Rechnerentwicklung unabhängig voneinander wäre.

Zusatzinfo für Lehrer: Tatsächlich ist die Effizienz der modernen Algorithmen auf den alten, langsamen Rechnern höher als auf den schnelleren modernen Computern, d.h. moderne Computer können die schnellen Algorithmen weniger optimal abarbeiten: Da diese Algorithmen sehr wenig Rechenoperationen benötigen, ist der Zugriff auf Daten im Speicher auf modernen Rechnern mit ihren leistungsfähigen Prozessoren wesentlich problematischer als auf den alten Rechnern mit wesentlich langsameren Prozessoren. Trotzdem ist die Rechenzeit von modernen Algorithmen auf modernen Computern in diesem Beispiel natürlich immer noch um einen Faktor 5000 kleiner als die der Algorithmen von 1980 (0,01 sec im Vergleich zu 50 sec).

## Arbeitsblatt: Gaußsches Eliminationsverfahren

**Aufgabe 1:** Schau dir die folgenden Gleichungssysteme genau an und löse sie, indem du mit der jeweils letzten Gleichung beginnst.

$$\begin{array}{lll} \text{(a)} & 5x_1 - x_2 = 4 & \text{(b)} \quad 3x_1 - 4x_2 + 5x_3 = -3 \\ & 4x_2 = 8 & \quad 3x_2 - 2x_3 = 4 \\ & & \quad 5x_3 = 5 \\ \text{(c)} & x_1 - x_2 - x_3 + x_4 = 1 & \\ & x_2 + x_3 + x_4 = 8 & \\ & x_3 - x_4 = 2 & \\ & x_4 = 5 & \end{array}$$

(d) Warum ist die Lösung dieser Gleichungssysteme einfach?

Das *Gaußsche Eliminationsverfahren* ist eine Methode zur Lösung von Gleichungssystemen, bei der die Gleichungssysteme zunächst in die Form umgewandelt werden, die bei allen Gleichungssystemen aus Aufgabe 1 vorliegt: In der letzten Gleichung kommt nur die letzte Unbekannte vor, in der vorletzten Gleichung nur die beiden letzten Unbekannten usw., denn dann kann man das Gleichungssystem einfach lösen. Zu einer solchen Form kommt man, indem man geeignete Vielfache der ersten Gleichungen zu den anderen Gleichungen addiert.

**Beispiel:** Wir betrachten das Gleichungssystem

$$\begin{array}{rrcr} 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ 2x_1 & + & 4x_2 & + & 2x_3 & = & 6 \\ x_1 & + & x_2 & + & 4x_3 & = & 3 \end{array}$$

Um in der zweiten Gleichung die erste Unbekannte zu eliminieren, addieren wir das  $(-\frac{2}{5})$ -fache der ersten Gleichung zur zweiten Gleichung. Ganz entsprechend addieren wir zur dritten Gleichung das  $(-\frac{1}{5})$ -fache der ersten Gleichung. Damit erhalten wir das Gleichungssystem

$$\begin{array}{rrcr} 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ & & \frac{16}{5}x_2 & + & \frac{8}{5}x_3 & = & \frac{28}{5} \\ & & \frac{3}{5}x_2 & + & \frac{19}{5}x_3 & = & \frac{14}{5} \end{array}$$

Jetzt stört nur noch die Unbekannte  $x_2$  in der letzten Gleichung. Um auch diese noch zu eliminieren, addieren wir das  $(-\frac{3}{5} : \frac{16}{5})$ -fache der zweiten Gleichung zur dritten Gleichung  $((-\frac{3}{5} : \frac{16}{5}) = (-\frac{3}{16}))$ . So erhalten wir die Zielform

$$\begin{array}{rrcr} 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ & & \frac{16}{5}x_2 & + & \frac{8}{5}x_3 & = & \frac{28}{5} \\ & & & & \frac{7}{2}x_3 & = & \frac{7}{4} \quad . \end{array}$$

**Aufgabe 2:**

- (a) Rechne dieses Beispiel im Detail nach.
- (b) Löse das entstandene Gleichungssystem, indem du mit der letzten Gleichung beginnst.

**Aufgabe 3:** Gehe wie im Beispiel vor und löse die folgenden Gleichungssysteme:

$$\begin{array}{lll} \text{(a)} & x_1 + 2x_2 = 5 & \text{(b)} \quad x_1 + 3x_2 + 2x_3 = 6 \\ & x_1 + 4x_2 = 9 & \quad 2x_1 + 7x_2 - 2x_3 = 7 \\ & & \quad 3x_1 + 8x_2 - 4x_3 = 7 \\ \text{(c)} & 2x_1 + 2x_2 + 5x_3 = 21 & \\ & 4x_1 + 5x_2 - 2x_3 = 8 & \\ & 6x_1 + 9x_2 - 5x_3 = 9 & \end{array}$$

**Arbeitsblatt: Gaußsches Eliminationsverfahren****Infos für Lehrkräfte:**

Dieses Arbeitsblatt soll dazu dienen, dass sich Schüler/innen der Oberstufe selbständig das Gaußsche Eliminationsverfahren erarbeiten.

Aufgabe 1 dient dazu, zu erkennen, dass Gleichungssysteme, die bereits in Dreiecksgestalt vorliegen, direkt und einfach gelöst werden können. Diese Erkenntnis macht man sich dann im Gaußschen Eliminationsverfahren zunutze, indem man ein gegebenes Gleichungssystem in diese Dreiecksgestalt bringt.

Dieser erste Teilschritt des Gaußschen Eliminationsverfahrens wird dann anhand eines Beispiels eingeführt. Um das Vorgehen einheitlich zu beschreiben, haben wir uns dabei darauf beschränkt, lediglich Vielfache von Gleichungen zu anderen Gleichungen zu addieren.

Es ist möglich, dass dieses Beispiel allein nicht ausreicht, um schwächeren Schülerinnen und Schülern das Gaußsche Eliminationsverfahren hinreichend zu erklären. Für diesen Fall sollte man Aufgabe 2(a) gemeinsam an der Tafel berechnen, bevor sich die Schüler/innen mit Aufgabe 3 beschäftigen. Erfahrungsgemäß sollte man dabei Wert darauf legen, detailliert zu beschreiben, wie man die Faktoren, mit denen man eine zu addierende Gleichung multipliziert, um eine neue 0 im Gleichungssystem zu erzeugen, (im Beispiel auf dem Arbeitsblatt die Zahlen  $-\frac{2}{5}$ ,  $-\frac{1}{5}$ ,  $-\frac{3}{16}$ ) bestimmt.

Man beachte, dass sich durch das hier beschriebene Vorgehen (Addieren von Vielfachen einer Gleichung zu anderen Gleichungen) der Wert der Determinante der Matrix, die das Gleichungssystem beschreibt, nicht ändert. Die Berechnung der Determinante fällt also als Nebenprodukt des Verfahrens mit ab. Die Lehrkraft kann dies als zusätzliche Information in den Unterricht einbringen. Dies bedeutet für das auf dem Arbeitsblatt angegebene Beispiel:

$$\det \begin{pmatrix} 5 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 4 \end{pmatrix} = \det \begin{pmatrix} 5 & 2 & 1 \\ 0 & \frac{16}{5} & \frac{8}{5} \\ 0 & 0 & \frac{7}{2} \end{pmatrix} = 5 \cdot \frac{16}{5} \cdot \frac{7}{2} = 56$$

**Lösungen der Aufgaben:****Aufgabe 1:**

(a)

$$\left. \begin{array}{rcl} 5x_1 - x_2 & = & 4 \\ 4x_2 & = & 8 \end{array} \right\} \Rightarrow x_2 = 2 \quad x_1 = \frac{4+x_2}{5} = \frac{6}{5}$$

(b)

$$\left. \begin{array}{rcl} 3x_1 - 4x_2 + 5x_3 & = & -3 \\ 3x_2 - 2x_3 & = & 4 \\ 5x_3 & = & 5 \end{array} \right\} \Rightarrow x_3 = 1 \quad x_2 = \frac{4+2x_3}{3} = 2 \quad x_1 = \frac{-3+4x_2-5x_3}{3} = 0$$

(c)

$$\left. \begin{array}{rcl} x_1 - x_2 - x_3 + x_4 & = & 1 \\ x_2 + x_3 + x_4 & = & 8 \\ x_3 - x_4 & = & 2 \\ x_4 & = & 5 \end{array} \right\} \Rightarrow \begin{array}{l} x_1 = 1 + x_2 + x_3 - x_4 = -1 \\ x_2 = 8 - x_3 - x_4 = -4 \\ x_3 = 2 + x_4 = 7 \\ x_4 = 5 \end{array}$$

(d) Die Gleichungssysteme haben alle Dreiecksgestalt: In der letzten Gleichung kommt nur die letzte Unbekannte vor, in der vorletzten Gleichung nur die beiden letzten Unbekannten usw.

## Aufgabe 2:

(a) trivial

(b) Aus

$$\begin{array}{rclcl} 5x_1 & + & 2x_2 & + & x_3 & = & 1 \\ & & \frac{16}{5}x_2 & + & \frac{8}{5}x_3 & = & \frac{28}{5} \\ & & & & \frac{7}{2}x_3 & = & \frac{7}{4} \end{array}$$

folgt sofort

$$\begin{aligned} x_3 &= \frac{7}{4} : \frac{7}{2} = \frac{1}{2} \\ x_2 &= \frac{5}{16} \left( \frac{28}{5} - \frac{8}{5} x_3 \right) = \frac{5}{16} \left( \frac{28}{5} - \frac{4}{5} \right) = \frac{5}{16} \cdot \frac{24}{5} = \frac{3}{2} \\ x_1 &= \frac{1}{5} (1 - 2x_2 - x_3) = \frac{1}{5} \left( 1 - 3 - \frac{1}{2} \right) = \frac{1}{5} \cdot \left( -\frac{5}{2} \right) = -\frac{1}{2} \end{aligned}$$

### Aufgabe 3:

(a)

$$\left. \begin{array}{l} x_1 + 2x_2 = 5 \\ x_1 + 4x_2 = 9 \end{array} \right\} \Rightarrow \left. \begin{array}{l} x_1 + 2x_2 = 5 \\ 2x_2 = 4 \end{array} \right\} \Rightarrow \begin{array}{l} x_1 = 1 \\ x_2 = 2 \end{array}$$

(b)

$$\left. \begin{array}{l} x_1 + 3x_2 + 2x_3 = 6 \\ 2x_1 + 7x_2 - 2x_3 = 7 \\ 3x_1 + 8x_2 - 4x_3 = 7 \end{array} \right\} \Rightarrow \left. \begin{array}{l} x_1 + 3x_2 + 2x_3 = 6 \\ x_2 - 6x_3 = -5 \\ -x_2 - 10x_3 = -11 \end{array} \right\} \Rightarrow \left. \begin{array}{l} x_1 + 3x_2 + 2x_3 = 6 \\ x_2 - 6x_3 = -5 \\ -16x_3 = -16 \end{array} \right\} \Rightarrow \begin{array}{l} x_1 = 1 \\ x_2 = 1 \\ x_3 = 1 \end{array}$$

(c)

$$\left. \begin{array}{l} 2x_1 + 2x_2 + 5x_3 = 21 \\ 4x_1 + 5x_2 - 2x_3 = 8 \\ 6x_1 + 9x_2 - 5x_3 = 9 \end{array} \right\} \Rightarrow \left. \begin{array}{l} 2x_1 + 2x_2 + 5x_3 = 21 \\ x_2 - 12x_3 = -34 \\ 3x_2 - 20x_3 = -54 \end{array} \right\} \Rightarrow \left. \begin{array}{l} 2x_1 + 2x_2 + 5x_3 = 21 \\ x_2 - 12x_3 = -34 \\ 16x_3 = 48 \end{array} \right\} \Rightarrow \begin{array}{l} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \end{array}$$

## Arbeitsblatt: Jacobi-Verfahren

In der Praxis treten häufig sehr große Gleichungssysteme auf (mehrere Millionen von Unbekannten), die eine spezielle Struktur haben, d.h. in jeder einzelnen Gleichung treten nur relativ wenige Unbekannte auf. In solchen Fällen sind iterative Lösungsverfahren eine interessante Alternative. Bei ihnen startet man mit irgendeiner Startnäherung für die Unbekannten (im Zweifelsfall nimmt man an, dass alle Unbekannten 0 sind) und reduziert den Fehler sukzessiv durch eine geeignete Iterationsvorschrift. Ein sehr einfaches iteratives Verfahren ist das Jacobi-Verfahren.

Wir betrachten ein beliebiges Gleichungssystem mit 3 Gleichungen und 3 Unbekannten:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3\end{aligned}$$

Gilt  $a_{ii} \neq 0$  für alle  $i$  ( $i = 1, 2, 3$ ), so können wir jeweils die  $i$ -te Gleichung nach  $x_i$  auflösen:

$$\begin{aligned}x_1 &= (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \\x_2 &= (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \\x_3 &= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} .\end{aligned}$$

Sind Startnäherungen  $x_1^{(0)}$ ,  $x_2^{(0)}$  und  $x_3^{(0)}$  gegeben, so können wir hieraus wie folgt neue Näherungen  $x_1^{(1)}$ ,  $x_2^{(1)}$  und  $x_3^{(1)}$  berechnen:

$$\begin{aligned}x_1^{(1)} &= (b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)})/a_{11} \\x_2^{(1)} &= (b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)})/a_{22} \\x_3^{(1)} &= (b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)})/a_{33} .\end{aligned}$$

Dies ist der erste Schritt des Jacobi-Verfahrens, das sich bei 3 Gleichungen mit 3 Unbekannten allgemein schreiben lässt als

$$\begin{aligned}x_1^{(n+1)} &= (b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)})/a_{11} \\x_2^{(n+1)} &= (b_2 - a_{21}x_1^{(n)} - a_{23}x_3^{(n)})/a_{22} \\x_3^{(n+1)} &= (b_3 - a_{31}x_1^{(n)} - a_{32}x_2^{(n)})/a_{33} \quad \text{mit } n = 0, 1, 2, 3, \dots\end{aligned}$$

**Aufgabe 1:** Führe beginnend mit der Startnäherung  $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$  zwei Iterationsschritte des Jacobi-Verfahrens aus, um eine Näherungslösung des Gleichungssystems

$$\begin{aligned}10x_1 + x_2 - x_3 &= 11 \\-x_1 + 10x_2 + x_3 &= 20 \\x_1 - x_2 + 10x_3 &= 9\end{aligned}$$

zu berechnen. Vergleiche die berechneten Näherungen mit der exakten Lösung  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 1$ .

### Aufgabe 2:

(a) Begründe, weshalb die Iterationsvorschrift des Jacobi-Verfahrens für das Gleichungssystem

$$\begin{aligned}10x_1 + x_2 &= 1 \\x_1 + 10x_2 &= 10\end{aligned} \quad \text{folgendermaßen lautet:} \quad \begin{aligned}x_1^{(n+1)} &= (1 - x_2^{(n)})/10 \\x_2^{(n+1)} &= (10 - x_1^{(n)})/10 .\end{aligned}$$

(b) Führe ausgehend von der Startnäherung  $x_1^{(0)} = x_2^{(0)} = 0$  vier Schritte des Jacobi-Verfahrens durch. Vergleiche mit der exakten Lösung  $x_1 = 0$  und  $x_2 = 1$ .



**Arbeitsblatt: Jacobi-Verfahren****Infos für Lehrkräfte:**

Mit diesem Arbeitsblatt sollen sich die Schüler/innen selbständig das Jacobi-Verfahren erarbeiten.

Bei der Bearbeitung der Aufgaben ist denkbar, dass einige Schüler/innen berechnete neue Näherungen  $x_i^{(n+1)}$  direkt in die Verfahrensvorschrift einsetzen, obwohl dort beim Jacobi-Verfahren noch die alten Näherungen  $x_i^{(n)}$  stehen. Geschieht dies systematisch, so haben diese Schüler/innen das Gauss-Seidel-Verfahren “entdeckt”, das im vorliegenden Fall (und in vielen anderen Fällen) schneller als das Jacobi-Verfahren konvergiert. Dies kann man natürlich im Unterricht aufgreifen und so das Gauss-Seidel-Verfahren einführen. Man sollte aber deutlich machen, dass es sich um zwei verschiedene Lösungsverfahren handelt.

**Lösungen der Aufgaben:**

**Aufgabe 1:** Die Iterationsvorschrift des Jacobi-Verfahrens lautet für unser Gleichungssystem:

$$\begin{aligned}x_1^{(n+1)} &= (11 - x_2^{(n)} + x_3^{(n)})/10 \\x_2^{(n+1)} &= (20 + x_1^{(n)} - x_3^{(n)})/10 \\x_3^{(n+1)} &= (9 - x_1^{(n)} + x_2^{(n)})/10\end{aligned}\quad \text{mit } n = 0, 1, 2, 3, \dots$$

Setzen wir die Startnäherung  $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$  in diese Verfahrensvorschrift ein, so erhalten wir:

$$\begin{aligned}x_1^{(1)} &= (11 - 0 + 0)/10 = 1,1 \\x_2^{(1)} &= (20 + 0 - 0)/10 = 2 \\x_3^{(1)} &= (9 - 0 + 0)/10 = 0,9\end{aligned}$$

sowie

$$\begin{aligned}x_1^{(2)} &= (11 - 2 + 0,9)/10 = 0,99 \\x_2^{(2)} &= (20 + 1,1 - 0,9)/10 = 2,02 \\x_3^{(2)} &= (9 - 1,1 + 2)/10 = 0,99\end{aligned}$$

Die berechnete Näherung ist schon nach zwei Schritten des Jacobi-Verfahrens ziemlich gut. Nach einem Schritt des Verfahrens weichen die Näherungen für  $x_1$  und  $x_3$  um 0,1 von der exakten Lösung ab,  $x_2$  ist exakt. Bei der Startnäherung hatten die Abweichungen noch bei 1 bzw. 2 gelegen; die maximale Abweichung von der exakten Lösung hat sich also im ersten Iterationsschritt um einen Faktor 20 verringert. Nach zwei Schritten liegen die Abweichungen von  $x_1$  und  $x_3$  nur noch bei 0,01; die Abweichung von  $x_2$  beträgt 0,02. Die maximale Abweichung hat sich also im zweiten Iterationsschritt von 0,1 auf 0,02 und somit um einen Faktor 5 verringert.

**Aufgabe 2:**

(a) Hier sind (mindestens) zwei Begründungen plausibel: (b)

1. Man kann wie auf dem Aufgabenblatt vorgehen und die erste Gleichung nach  $x_1$  sowie die zweite Gleichung nach  $x_2$  auflösen und dann eine neue Näherung berechnen, indem man auf den erhaltenen rechten Seiten die alte Näherung einsetzt.
2. Man kann zwei Gleichungen mit zwei Unbekannten als Sonderfall von drei Gleichungen mit drei Unbekannten betrachten und direkt die auf dem Arbeitsblatt beschriebene Iterationsvorschrift für  $x_1$  und  $x_2$  anwenden und dabei  $a_{13} = a_{23} = 0$  setzen.

Beginnen wir mit der Startnäherung  $x_1^{(0)} = x_2^{(0)} = 0$ , so erhalten wir

$$\begin{aligned}(x_1^{(0)}, x_2^{(0)}) &= (0; 0) \\(x_1^{(1)}, x_2^{(1)}) &= (0, 1; 0) \\(x_1^{(2)}, x_2^{(2)}) &= (0; 0, 99) \\(x_1^{(3)}, x_2^{(3)}) &= (10^{-3}; 0) \\(x_1^{(4)}, x_2^{(4)}) &= (0; 1 - 10^{-4}) .\end{aligned}$$

Untersuchen wir die maximalen Abweichungen der berechneten Näherungen von der exakten Lösung des Gleichungssystems, so stellen wir fest, dass sie sich pro Iterationsschritt jeweils um einen Faktor 10 reduzieren.

## Arbeitsblatt: Gauss-Seidel-Verfahren

Wir haben am Beispiel des Jacobi-Verfahrens ein erstes iteratives Verfahren zur Lösung eines gegebenen Gleichungssystems kennengelernt. Für das Gleichungssystem

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

lautet das Jacobi-Verfahren

$$\begin{aligned}x_1^{(n+1)} &= (b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)})/a_{11} \\x_2^{(n+1)} &= (b_2 - a_{21}x_1^{(n)} - a_{23}x_3^{(n)})/a_{22} \\x_3^{(n+1)} &= (b_3 - a_{31}x_1^{(n)} - a_{32}x_2^{(n)})/a_{33}\end{aligned}\quad \text{mit } n = 0, 1, 2, 3, \dots$$

Hierbei haben wir vorausgesetzt, dass  $a_{11}$ ,  $a_{22}$  und  $a_{33}$  ungleich 0 sind.

Werden in dieser Iterationsvorschrift zur Berechnung von  $x_2^{(n+1)}$  und  $x_3^{(n+1)}$  nicht nur Werte  $x_1^{(n)}$  und  $x_2^{(n)}$ , sondern - soweit bereits verfügbar - die neuen, gerade berechneten Werte  $x_1^{(n+1)}$  und  $x_2^{(n+1)}$  eingesetzt, so erhält man das **Gauss-Seidel-Verfahren**:

$$\begin{aligned}x_1^{(n+1)} &= (b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)})/a_{11} \\x_2^{(n+1)} &= (b_2 - a_{21}x_1^{(n+1)} - a_{23}x_3^{(n)})/a_{22} \\x_3^{(n+1)} &= (b_3 - a_{31}x_1^{(n+1)} - a_{32}x_2^{(n+1)})/a_{33}\end{aligned}\quad \text{mit } n = 0, 1, 2, 3, \dots$$

**Aufgabe 1:** Führe beginnend mit der Startnäherung  $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$  zwei Iterationsschritte des Gauss-Seidel-Verfahrens aus, um eine Näherungslösung des Gleichungssystems

$$\begin{aligned}10x_1 + x_2 - x_3 &= 11 \\-x_1 + 10x_2 + x_3 &= 20 \\x_1 - x_2 + 10x_3 &= 9\end{aligned}$$

zu berechnen. Vergleiche die berechneten Näherungen mit der exakten Lösung  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 1$  (und mit den beim Jacobi-Verfahren berechneten Näherungen).

### Aufgabe 2:

(a) Begründe, weshalb die Iterationsvorschrift des Gauss-Seidel-Verfahrens für das Gleichungssystem

$$\begin{aligned}10x_1 + x_2 &= 1 \\x_1 + 10x_2 &= 10\end{aligned}\quad \text{folgendermaßen lautet: } \begin{aligned}x_1^{(n+1)} &= (1 - x_2^{(n)})/10 \\x_2^{(n+1)} &= (10 - x_1^{(n+1)})/10.\end{aligned}$$

(b) Führe ausgehend von der Startnäherung  $x_1^{(0)} = x_2^{(0)} = 0$  vier Schritte des Gauss-Seidel-Verfahrens durch. Vergleiche mit der exakten Lösung  $x_1 = 0$  und  $x_2 = 1$  (und mit den beim Jacobi-Verfahren berechneten Näherungen).

**Arbeitsblatt: Gauss-Seidel-Verfahren****Infos für Lehrkräfte:**

Mit diesem Arbeitsblatt sollen sich die Schüler/innen selbständig das Gauss-Seidel-Verfahren erarbeiten. Die Kenntnis des Jacobi-Verfahrens wird hierbei vorausgesetzt.

**Lösungen der Aufgaben:**

**Aufgabe 1:** Die Iterationsvorschrift des Gauss-Seidel-Verfahrens lautet für unser Gleichungssystem:

$$\begin{aligned}x_1^{(n+1)} &= (11 - x_2^{(n)} + x_3^{(n)})/10 \\x_2^{(n+1)} &= (20 + x_1^{(n+1)} - x_3^{(n)})/10 \\x_3^{(n+1)} &= (9 - x_1^{(n+1)} + x_2^{(n+1)})/10 \quad \text{mit } n = 0, 1, 2, 3, \dots\end{aligned}$$

Setzen wir die Startnäherung  $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$  in diese Verfahrensvorschrift ein, so erhalten wir:

$$\begin{aligned}x_1^{(1)} &= (11 - 0 + 0)/10 = 1,1 \\x_2^{(1)} &= (20 + 1,1 - 0)/10 = 2,11 \\x_3^{(1)} &= (9 - 1,1 + 2,11)/10 = 1,001\end{aligned}$$

sowie

$$\begin{aligned}x_1^{(2)} &= (11 - 2,11 + 1,001)/10 = 0,9891 \\x_2^{(2)} &= (20 + 0,9891 - 1,001)/10 = 1,998 \\x_3^{(2)} &= (9 - 0,9891 + 1,998)/10 = 1,00089 .\end{aligned}$$

Auch beim Gauss-Seidel-Verfahren ist die berechnete Näherung in diesem Beispiel schon nach zwei Schritten ziemlich gut. Nach einem Schritt des Verfahrens weicht die Näherung für  $x_1$  um 0,1, für  $x_2$  um 0,11 und für  $x_3$  um 0,001 von der exakten Lösung ab. Nach zwei Schritten liegt die Abweichung von  $x_1$  bei 0,0109, die von  $x_2$  bei 0,002 und die von  $x_3$  bei 0,00089. Mit Ausnahme von  $x_1$  sind die Abweichungen von den exakten Werten deutlich kleiner als beim Jacobi-Verfahren. Auch der nächste Wert für  $x_1$  wird beim Gauss-Seidel-Verfahren deutlich besser sein als beim Jacobi-Verfahren, weil hier nur die (besseren) aktuellen Näherungswerte für  $x_2$  und  $x_3$  benutzt werden. Zum Vergleich:

Gauss-Seidel-Verfahren:  $x_1^{(3)} = 1,000289$ ; Abweichung vom exakten Wert  $\approx 3 \cdot 10^{-4}$

Jacobi-Verfahren:  $x_1^{(3)} = 0,997$ ; Abweichung vom exakten Wert  $\approx 3 \cdot 10^{-3}$

**Aufgabe 2:**

(a) Hier sind (mindestens) zwei Begründungen plausibel: (b)

1. Man kann wie auf dem Aufgabenblatt vorgehen und die erste Gleichung nach  $x_1$  sowie die zweite Gleichung nach  $x_2$  auflösen und dann eine neue Näherung berechnen, indem man auf den erhaltenen rechten Seiten die alte Näherung einsetzt.
2. Man kann zwei Gleichungen mit zwei Unbekannten als Sonderfall von drei Gleichungen mit drei Unbekannten betrachten und direkt die auf dem Arbeitsblatt beschriebene Iterationsvorschrift für  $x_1$  und  $x_2$  anwenden und dabei  $a_{13} = a_{23} = 0$  setzen.

Beginnen wir mit der Startnäherung  $x_1^{(0)} = x_2^{(0)} = 0$ , so erhalten wir beim Gauss-Seidel-Verfahren

$$\begin{aligned}(x_1^{(0)}, x_2^{(0)}) &= (0, 0) \\(x_1^{(1)}, x_2^{(1)}) &= (0.1, 0.99) \\(x_1^{(2)}, x_2^{(2)}) &= (10^{-3}, 1 - 10^{-4}) \\(x_1^{(3)}, x_2^{(3)}) &= (10^{-5}, 1 - 10^{-6}) \\(x_1^{(4)}, x_2^{(4)}) &= (10^{-7}, 1 - 10^{-8}) .\end{aligned}$$

Untersuchen wir die maximalen Abweichungen der berechneten Näherungen von der exakten Lösung des Gleichungssystems, so stellen wir fest, dass sie sich pro Iterationsschritt jeweils um einen Faktor 100 reduzieren. Damit konvergiert das Gauss-Seidel-Verfahren in diesem Beispiel deutlich schneller als das Jacobi-Verfahren, bei dem lediglich eine Fehlerreduktion um den Faktor 10 pro Iterationsschritt zu verzeichnen war.

## Arbeitsblatt: Defekt einer Näherungslösung

Wir haben mit dem Jacobi-Verfahren und/oder dem Gauss-Seidel-Verfahren iterative Verfahren zur Lösung eines gegebenen Gleichungssystems (GLS) kennengelernt. Für iterative Verfahren ist aber typisch, dass sie nur für bestimmte Klassen von GLS gegen die richtige Lösung des GLS konvergieren. In anderen Fällen kommen sie der Lösung des GLS überhaupt nicht näher oder sie entfernen sich sogar davon. Da die exakte Lösung des GLS im Allgemeinen nicht bekannt ist, ist folgende Frage bei iterativen Verfahren von grundsätzlichem Interesse: Wie kann ich feststellen, ob meine Folge von Näherungswerten für die Unbekannten gegen die Lösung des GLS strebt?

Um festzustellen, wie gut die aktuelle Näherungslösung das Gleichungssystem erfüllt, kann man für jede Gleichung den sogenannten *Defekt* berechnen, indem man von der rechten Seite jeder Gleichung die linke Seite subtrahiert und hierbei für die Unbekannten die aktuellen Näherungen einsetzt.

**Beispiel:** Nehmen wir an, wir haben für das  $3 \times 3$ -GLS

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Näherungen  $x_1^{(n+1)}$ ,  $x_2^{(n+1)}$  und  $x_3^{(n+1)}$  berechnet. Dann lauten die Defekte:

$$\begin{aligned} d_1^{(n+1)} &= b_1 - (a_{11}x_1^{(n+1)} + a_{12}x_2^{(n+1)} + a_{13}x_3^{(n+1)}) \\ d_2^{(n+1)} &= b_2 - (a_{21}x_1^{(n+1)} + a_{22}x_2^{(n+1)} + a_{23}x_3^{(n+1)}) \\ d_3^{(n+1)} &= b_3 - (a_{31}x_1^{(n+1)} + a_{32}x_2^{(n+1)} + a_{33}x_3^{(n+1)}) . \end{aligned}$$

### Aufgabe 1:

(a) Beim GLS

$$\begin{aligned} 10x_1 + x_2 &= 1 \\ x_1 + 10x_2 &= 10 \end{aligned}$$

sind die ersten Näherungen des Gauss-Seidel-Verfahrens für die Unbekannten

$$\begin{aligned} (x_1^{(0)}, x_2^{(0)}) &= (0, 0) \\ (x_1^{(1)}, x_2^{(1)}) &= (0.1, 0.99) \\ (x_1^{(2)}, x_2^{(2)}) &= (10^{-3}, 1 - 10^{-4}) \\ (x_1^{(3)}, x_2^{(3)}) &= (10^{-5}, 1 - 10^{-6}) . \end{aligned}$$

Berechne die Defekte  $d_1^{(k)}$  und  $d_2^{(k)}$  für  $k = 0, 1, 2, 3$ . Was fällt dir auf?

- (b) Begründe, warum der Defekt ein sinnvolles Mittel ist, um festzustellen, ob ein iteratives Verfahren gegen die Lösung eines GLS konvergiert.
- (c) Kann man aus der Kenntnis über den Defekt eine quantitative Aussage über die Größe des Fehlers machen?

## Arbeitsblatt: Defekt einer Näherungslösung

### Infos für Lehrkräfte:

Mit diesem Arbeitsblatt sollen sich die Schüler/innen selbständig den Begriff des Defektes einer Näherungslösung bei der Lösung eines GLS erarbeiten. Die Kenntnis des Jacobi- und/oder Gauss-Seidel-Verfahrens wird hierbei vorausgesetzt.

### Lösungen der Aufgaben:

#### Aufgabe 1:

(a) Die Defekte lauten

$$\begin{aligned} d_1^{(n+1)} &= 1 - (10x_1^{(n+1)} + x_2^{(n+1)}) \\ d_2^{(n+1)} &= 10 - (x_1^{(n+1)} + 10x_2^{(n+1)}) \end{aligned}$$

Setzen wir die Näherungslösungen ein, so ergibt sich

$$\begin{aligned} (d_1^{(0)}; d_2^{(0)}) &= (1; 10) \\ (d_1^{(1)}; d_2^{(1)}) &= (-0.99; 0) \\ (d_1^{(2)}; d_2^{(2)}) &= (-0.0099; 0) \\ (d_1^{(3)}; d_2^{(3)}) &= (-0.000099; 0) . \end{aligned}$$

Sieht man vom ersten Iterationsschritt ab, so reduziert sich der Defekt der ersten Gleichung pro Iterationsschritt um den Faktor 100. Der zweite Defekt ist in jedem Iterationsschritt 0.

**Bemerkung:** Dies ist typisch für das Gauss-Seidel-Verfahren. Die letzte Gleichung ist nach einem Iterationsschritt immer erfüllt, da sie ja gerade verwendet wurde, um eine neue Näherung für die letzte Unbekannte zu berechnen.

- (b) Wenn der Defekt von jeder Gleichung 0 ist, so hat man die exakte Lösung des GLS gefunden: rechte und linke Seite einer jeden Gleichung stimmen überein. Wird der Defekt klein, so sind rechte und linke Seite des GLS sehr ähnlich und man befindet sich in der Nähe der Lösung des GLS. Ist der Defekt noch groß, so ist man noch weit von einer Lösung des GLS entfernt.
- (c) Leider eignet sich der Defekt nicht, um eine quantitative Aussage über die Größe des Fehlers machen. Dies sieht man leicht, wenn man ein gegebenes GLS mit einem beliebigen Faktor  $k$  multipliziert. Dies ändert nichts an der Lösung des GLS oder an den mit einem Iterationsverfahren berechneten Näherungen für die Lösung, aber alle Defekte werden ebenfalls mit dem Faktor  $k$  multipliziert.

## Arbeitsblatt: Konvergenzeigenschaften des Jacobi- und des Gauss-Seidel-Verfahrens

Iterative Lösungsverfahren konvergieren oft nur für bestimmte Klassen von Gleichungssystemen (GLS) gegen die richtige Lösung des GLS. Daher ist es sehr wichtig zu wissen, für welche Klassen von GLS ein iteratives Lösungsverfahren auf jeden Fall konvergiert. Gauss-Seidel- und Jacobi-Verfahren konvergieren für die Klasse der Gleichungssysteme, bei denen die Koeffizientenmatrix “stark diagonaldominant” ist, und auch für eine Reihe anderer Gleichungssysteme.

**Definition:** Gegeben sei ein GLS  $Ax = b$  mit einer Matrix  $A$ , die aus den Elementen  $a_{ij}$  besteht ( $i, j = 1, \dots, N$ ). Gilt dann für jede Zeile der Matrix

$$\sum_{\substack{k=1 \\ k \neq i}}^N |a_{ik}| < |a_{ii}| ,$$

so nennt man die Matrix  $A$  auch *stark diagonaldominant*.

Man kann zeigen: Ist eine Matrix  $A$  stark diagonaldominant (d.h. überwiegt in der Matrix  $A$  die Hauptdiagonale in obigem Sinn), dann konvergieren das Jacobi-Verfahren und das Gauss-Seidel-Verfahren gegen die eindeutig bestimmte Lösung von  $Ax = b$ .

### Ein Beispiel:

Die Matrix

$$A = \begin{pmatrix} 10 & -4 & -2 \\ -7 & 15 & 5 \\ -6 & -3 & 12 \end{pmatrix}$$

ist stark diagonaldominant, denn es gilt:

$$\begin{aligned} 10 &> |-4| + |-2| \\ 15 &> |-7| + 5 \\ 12 &> |-6| + |-3| . \end{aligned}$$

Für jedes beliebige GLS der Form

$$\begin{aligned} 10x_1 - 4x_2 - 2x_3 &= b_1 \\ -7x_1 + 15x_2 + 5x_3 &= b_2 \\ -6x_1 - 3x_2 + 12x_3 &= b_3 \end{aligned}$$

konvergieren daher Jacobi- und Gauss-Seidel-Verfahren.

**Bemerkung:** Ist eine Matrix  $A$  weit davon entfernt, stark diagonaldominant zu sein, so ist die Chance, dass das Jacobi- oder Gauss-Seidel-Verfahren konvergieren, relativ klein.

**Aufgabe 1:** Betrachte die beiden GLS

$$\begin{aligned} 5x_1 + 22x_2 &= 1 \\ 9x_1 - x_2 &= 3 \end{aligned} \quad \text{und} \quad \begin{aligned} x_1 + 10x_2 &= 10 \\ 10x_1 + x_2 &= 1 \end{aligned}$$

- (a) Sind diese GLS stark diagonaldominant?
- (b) Kann man erwarten, dass das Jacobi- oder das Gauss-Seidel-Verfahren für eines dieser beiden GLS gegen die Lösung konvergieren?
- (c) Wie kann man diese GLS stark diagonaldominant machen? Werden jetzt Jacobi- und Gauss-Seidel-Verfahren für die beiden GLS konvergieren?



**Arbeitsblatt: Konvergenzeigenschaften des Jacobi- und des Gauss-Seidel-Verfahrens****Infos für Lehrkräfte:**

Dieses Arbeitsblatt führt den Begriff der starken Diagonaldominanz für allgemeine  $N \times N$ -Matrizen ein. Die Matrixnotation eines Gleichungssystems wird als bekannt vorausgesetzt. Ebenfalls vorausgesetzt wird die Kenntnis des Summationssymbols  $\sum$ .

**Aufgabe 1:**

- (a) Beide GLS sind weit davon entfernt stark diagonal dominant zu sein, denn die Nicht-diagonalelemente sind in allen Fällen deutlich größer als die zugehörigen Diagonalelemente: Für das erste GLS gilt  $5 < 22$  und  $|-1| < 9$  und für das zweite  $1 < 10$  und  $1 < 10$ . Für starke Diagonaldominanz müssten statt der “<”-Zeichen “>”-Zeichen gelten.
- (b) Da beide GLS weit von einer Diagonaldominanz entfernt sind, kann man nicht erwarten, dass das Jacobi- oder das Gauss-Seidel-Verfahren für eines dieser beiden GLS gegen die Lösung konvergieren.
- (c) Man kann beide GLS sehr leicht diagonal dominant machen, indem man die Gleichungen vertauscht. Dann lauten die GLS

$$\begin{array}{rclcl} 9x_1 - x_2 & = & 3 & & \\ 5x_1 + 22x_2 & = & 1 & \text{und} & \\ & & & & 10x_1 + x_2 = 1 \\ & & & & x_1 + 10x_2 = 10 \end{array}$$

und jetzt dominieren die Diagonalelemente. Daher konvergieren Jacobi- und Gauss-Seidel-Verfahren, wenn man sie hierauf anwendet.