

LEHRBUCH

Sören Bartels

Numerik 3x9

Drei Themengebiete in jeweils
neun kurzen Kapiteln



Springer Spektrum

Springer-Lehrbuch

Sören Bartels

Numerik 3x9

Drei Themengebiete in jeweils neun kurzen Kapiteln

Sören Bartels
Abteilung für Angewandte Mathematik
Universität Freiburg
Freiburg, Deutschland

ISSN 0937-7433

ISBN 978-3-662-48202-5

DOI 10.1007/978-3-662-48203-2

ISBN 978-3-662-48203-2 (eBook)

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Spektrum

© Springer-Verlag Berlin Heidelberg 2016

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürfen.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer-Verlag GmbH Berlin Heidelberg ist Teil der Fachverlagsgruppe Springer Science+Business Media
(www.springer.com)

Vorwort

Die numerische Mathematik ist mit der Entwicklung und Analyse von Verfahren zur Lösung mathematischer Aufgaben befasst. Sie ist ein fester Bestandteil der Mathematik-Ausbildung an Hochschulen, spielt jedoch häufig eine untergeordnete Rolle, obwohl alle mathematischen Disziplinen aus dem Ziel resultieren, konkrete und praxisrelevante Probleme zu lösen. Dieser wichtige Aspekt kommt in der hochentwickelten modernen Mathematik oft zu kurz. Andererseits wird die Numerik häufig mit der zeitaufwendigen und mitunter weniger spannenden technischen Umsetzung von mathematischen Konzepten gleichgesetzt, was zu einem falschen Eindruck der Inhalte der Numerik führt. Tatsächlich gehen viele der numerischen Verfahren auf Wissenschaftler wie Gauß und Newton zurück, die mit Hilfe dieser Verfahren tiefgreifende Fragestellungen der Naturwissenschaften von Hand quantifizieren und verstehen wollten. Der Einsatz des Computers sollte also den Umgang mit numerischen Methoden vereinfachen und nicht erschweren.

In diesem Lehrbuch sollen die wichtigsten Ideen und Konzepte zur algorithmischen Lösung einiger grundlegender mathematischer Aufgaben diskutiert und die wesentlichen Schwierigkeiten der praktischen Umsetzung untersucht werden. Dabei sind stets drei Fragestellungen zu berücksichtigen:

- Ist es möglich, ein Verfahren zur näherungsweisen Lösung eines mathematischen Problems anzugeben?
- Wie wirken sich Störungen zum Beispiel durch Rundung der eingegebenen Daten auf die numerische Lösung aus?
- Wie hoch ist der Aufwand eines Verfahrens, um eine vorgegebene Genauigkeit zu erreichen?

Anhand klassischer Probleme, wie der Lösung linearer Gleichungssysteme, der Berechnung von Eigenwerten einer Matrix, der numerischen Integration von Funktionen, der näherungsweisen Lösung nichtlinearer Gleichungen und der Approximation von Lösungen von Differenzialgleichungen werden diese Fragestellungen behandelt.

Bei der Ausarbeitung des Lehrstoffs habe ich mich an den Darstellungen verschiedener Skripte, Lehrbücher und Monografien, die am Ende dieses Buchs aufgeführt sind, orientiert. Sollte ich in der Darstellung des Stoffs an der einen oder anderen Stelle zu sehr

einer Quelle gefolgt sein, so ist dies als Würdigung einer besonders gelungenen Ausarbeitung zu verstehen. Der nachfolgende Text erhebt keinerlei Anspruch auf Originalität seines Inhalts, sein Ziel ist es ausschließlich, Studenten der Mathematik, Ingenieur- und Naturwissenschaften eine weitere Möglichkeit zur Einarbeitung in die Grundlagen der numerischen Mathematik zu bieten.

Die Darstellung des klassischen Stoffs soll grundlegende Methoden der Numerik beispielhaft illustrieren. Auf Optimalität der Verfahren oder größte Allgemeingültigkeit der zugehörigen Aussagen wurde dabei bewusst verzichtet. Bei der numerischen Lösung konkret vorliegender, möglicherweise praxisrelevanter Probleme ist daher unbedingt die Spezialliteratur, die ebenfalls am Ende des Buchs auszugshaft dargestellt ist, zu konsultieren. Die im Text aufgeführten Anwendungsbeispiele sind zur Motivation und Illustration gedacht und sollten nicht als reale Fallbeispiele interpretiert werden. Für spezielle Anwendungen ist es in der Regel stets erforderlich, die für idealisierte Modellsituationen entwickelten Verfahren den besonderen Eigenschaften des vorliegenden Problems anzupassen. Dieses Buch soll den Leser auf diese Herausforderung vorbereiten.

Der Text resultiert aus Vorlesungen an den Universitäten Bonn und Freiburg. Zahlreichen Assistenten und Tuto ren danke ich für Korrekturhinweise und Verbesserungsvorschläge. Frau Lea Heusler danke ich für das sorgfältige Korrekturlesen des Manuskripts.

Freiburg, Juli 2014

Sören Bartels

Inhaltsverzeichnis

Teil I Numerische lineare Algebra

1	Grundlegende Konzepte	3
1.1	Aufgabenstellung	3
1.2	Kondition und Stabilität	4
1.3	Aufwand	7
1.4	Lernziele, Quiz und Anwendung	8
2	Operatornorm und Konditionszahl	9
2.1	Vektornormen	9
2.2	Matrixnormen	10
2.3	Konditionszahl	12
2.4	Lernziele, Quiz und Anwendung	13
3	Matrixfaktorisierungen	15
3.1	Dreiecksmatrizen	15
3.2	<i>LU</i> -Zerlegung	16
3.3	Cholesky-Zerlegung	18
3.4	Lernziele, Quiz und Anwendung	21
4	Eliminationsverfahren	23
4.1	Gaußsches Eliminationsverfahren	23
4.2	Pivot-Strategie	26
4.3	Lernziele, Quiz und Anwendung	28
5	Ausgleichsprobleme	31
5.1	Gaußsche Normalengleichung	31
5.2	Householder-Transformationen	33
5.3	<i>QR</i> -Zerlegung	35
5.4	Lösung des Ausgleichsproblems	37
5.5	Lernziele, Quiz und Anwendung	37

6	Singulärwertzerlegung und Pseudoinverse	39
6.1	Singulärwertzerlegung	39
6.2	Pseudoinverse	40
6.3	Lernziele, Quiz und Anwendung	41
7	Das Simplex-Verfahren	43
7.1	Lineare Programme	43
7.2	Der Simplex-Schritt	45
7.3	Lernziele, Quiz und Anwendung	48
8	Eigenwertaufgaben	49
8.1	Lokalisierung	49
8.2	Konditionierung	51
8.3	Potenzmethode	52
8.4	<i>QR</i> -Verfahren	56
8.5	Jacobi-Verfahren	57
8.6	Lernziele, Quiz und Anwendung	61
9	Iterative Lösungsmethoden	63
9.1	Inexakte Lösung	63
9.2	Banachscher Fixpunktsatz	63
9.3	Lineare Iterationsverfahren	65
9.4	Jacobi- und Gauß–Seidel-Verfahren	66
9.5	Diagonaldominanz und Irreduzibilität	67
9.6	Konvergenz	69
9.7	Lernziele, Quiz und Anwendung	70

Teil II Numerische Analysis

10	Allgemeine Konditionszahl und Gleitkommazahlen	75
10.1	Konditionierung	75
10.2	Gleitkommazahlen	77
10.3	Rundung	78
10.4	Stabilität	79
10.5	Lernziele, Quiz und Anwendung	81
11	Polynominterpolation	83
11.1	Lagrange-Interpolation	83
11.2	Interpolationsfehler	85
11.3	Neville-Schema	86
11.4	Tschebyscheff-Knoten	88

11.5	Hermite-Interpolation	90
11.6	Lernziele, Quiz und Anwendung	92
12	Interpolation mit Splines	93
12.1	Splines	93
12.2	Kubische Splines	95
12.3	Berechnung kubischer Splines	98
12.4	Lernziele, Quiz und Anwendung	99
13	Diskrete Fourier-Transformation	101
13.1	Trigonometrische Interpolation	101
13.2	Fourier-Basen	103
13.3	Schnelle Fourier-Transformation	105
13.4	Lernziele, Quiz und Anwendung	107
14	Numerische Integration	109
14.1	Quadraturformeln	109
14.2	Newton–Cotes-Formeln	111
14.3	Summierte Quadraturformeln	112
14.4	Gauß-Quadratur	114
14.5	Extrapolation	117
14.6	Experimentelle Konvergenzordnung	118
14.7	Lernziele, Quiz und Anwendung	119
15	Nichtlineare Probleme	121
15.1	Nullstellensuche und Minimierungsprobleme	121
15.2	Approximation von Nullstellen	122
15.3	Eindimensionale Minimierung	126
15.4	Mehrdimensionale Minimierung	127
15.5	Lernziele, Quiz und Anwendung	129
16	Methode der konjugierten Gradienten	131
16.1	Quadratische Minimierung	131
16.2	Konjugierte Suchrichtungen	132
16.3	Berechnung A -konjugierter Richtungen	133
16.4	CG-Verfahren	135
16.5	Konvergenz des CG-Verfahrens	136
16.6	Lernziele, Quiz und Anwendung	138
17	Dünnbesetzte Matrizen und Vorkonditionierung	141
17.1	Dünnbesetzte Matrizen	141
17.2	Vorkonditioniertes CG-Verfahren	142

17.3	Weitere Vorkonditionierungsmatrizen	144
17.4	Lernziele, Quiz und Anwendung	146
18	Mehrdimensionale Approximation	149
18.1	Gitter und Triangulierungen	149
18.2	Approximation auf Tensorproduktgittern	151
18.3	Zweidimensionale Fourier-Transformation	153
18.4	Approximation auf Triangulierungen	154
18.5	Lernziele, Quiz und Anwendung	157
 Teil III Numerik gewöhnlicher Differenzialgleichungen		
19	Gewöhnliche Differenzialgleichungen	161
19.1	Grundlagen	161
19.2	Das Räuber-Beute-Modell	162
19.3	Gleichungen höherer Ordnung	163
19.4	Autonome Gleichungen	164
19.5	Zweikörperprobleme	165
19.6	Explizite Lösungen	166
19.7	Lernziele, Quiz und Anwendung	166
20	Existenz, Eindeutigkeit und Stabilität	169
20.1	Existenz und Eindeutigkeit	169
20.2	Lemma von Gronwall	172
20.3	Stabilität	173
20.4	Lernziele, Quiz und Anwendung	175
21	Einschrittverfahren	177
21.1	Euler-Verfahren	177
21.2	Konsistenz	179
21.3	Diskretes Gronwall-Lemma und Konvergenz	181
21.4	Verfahren höherer Ordnung	184
21.5	Lernziele, Quiz und Anwendung	185
22	Runge–Kutta-Verfahren	187
22.1	Motivation	187
22.2	Runge–Kutta-Verfahren	188
22.3	Wohlgestelltheit	190
22.4	Konsistenz	192
22.5	Lernziele, Quiz und Anwendung	196

23	Mehrschrittverfahren	199
23.1	Allgemeine Mehrschrittverfahren	199
23.2	Konsistenz	201
23.3	Adams-Verfahren	202
23.4	Prädiktor-Korrektor-Verfahren	204
23.5	Lernziele, Quiz und Anwendung	206
24	Konvergenz von Mehrschrittverfahren	209
24.1	Differenzengleichungen	209
24.2	Nullstabilität	211
24.3	Konvergenz	213
24.4	Lernziele, Quiz und Anwendung	215
25	Steife Differenzialgleichungen	217
25.1	Steifheit	217
25.2	A -Stabilität	218
25.3	Gradientenflüsse	222
25.4	Wärmeleitungsgleichung	223
25.5	Lernziele, Quiz und Anwendung	225
26	Schrittweitensteuerung	227
26.1	A -posteriori Fehlerkontrolle	227
26.2	Adaptiver Algorithmus	230
26.3	Kontrollverfahren	230
26.4	Extrapolation	231
26.5	Lernziele, Quiz und Anwendung	231
27	Symplektische, Schieß- und dG-Verfahren	233
27.1	Hamiltonsche Systeme	233
27.2	Symplektische Verfahren	235
27.3	Schießverfahren	239
27.4	Diskontinuierliche Galerkin-Verfahren	240
27.5	Lernziele, Quiz und Anwendung	241

Teil IV Aufgabensammlungen

28	Aufgaben zur numerischen linearen Algebra	245
28.1	Grundlegende Konzepte	245
28.2	Operatornorm und Konditionszahl	247
28.3	Matrixfaktorisierungen	250
28.4	Eliminationsverfahren	254
28.5	Ausgleichsprobleme	257

28.6	Singulärwertzerlegung und Pseudoinverse	261
28.7	Das Simplex-Verfahren	264
28.8	Eigenwertaufgaben	267
28.9	Iterative Lösungsmethoden	271
29	Aufgaben zur numerischen Analysis	275
29.1	Allgemeine Konditionszahl und Maschinenzahlen	275
29.2	Polynominterpolation	278
29.3	Interpolation mit Splines	281
29.4	Diskrete Fourier-Transformation	284
29.5	Numerische Integration	288
29.6	Nichtlineare Probleme	291
29.7	Methode der konjugierten Gradienten	294
29.8	Dünnbesetzte Matrizen und Vorkonditionierung	298
29.9	Mehrdimensionale Approximation	301
30	Aufgaben zur Numerik gewöhnlicher Differenzialgleichungen	305
30.1	Gewöhnliche Differenzialgleichungen	305
30.2	Existenz, Eindeutigkeit und Stabilität	309
30.3	Einschrittverfahren	311
30.4	Runge–Kutta-Verfahren	315
30.5	Mehrschrittverfahren	318
30.6	Konvergenz von Mehrschrittverfahren	321
30.7	Steife Differenzialgleichungen	324
30.8	Schrittweitensteuerung	327
30.9	Symplektische, Schieß- und dG-Verfahren	330

Teil V Anhänge

31	Aussagen der linearen Algebra	335
31.1	Skalarprodukt von Vektoren	335
31.2	Determinante quadratischer Matrizen	335
31.3	Bild und Kern linearer Abbildungen	336
31.4	Eigenwerte und Diagonalisierbarkeit	337
31.5	Jordansche Normalform	338
32	Aussagen der Analysis	339
32.1	Stetige und differenzierbare Funktionen	339
32.2	Mittelwertsatz und Taylor-Polynome	340
32.3	Landau-Symbole	340
32.4	Fundamentalsatz der Algebra	341
32.5	Mehrdimensionale Differentialrechnung	342

33 Einführung in C	343
33.1 Struktur	343
33.2 Bibliotheken	343
33.3 Typen	344
33.4 Kontrollanweisungen	345
33.5 Logische Ausdrücke und Inkremeante	345
33.6 Funktionen	346
33.7 Pointer	346
33.8 Dynamische Arrays	347
33.9 Arbeiten mit Matrizen	347
33.10 Zeitmessung, Speicherung und Pakete	349
34 Einführung in Matlab	351
34.1 Aufbau	351
34.2 Listen und Arrays	351
34.3 Matrixoperationen	352
34.4 Manipulation von Arrays	353
34.5 Elementare Funktionen	353
34.6 Schleifen und Kontrollanweisungen	353
34.7 Text- und Grafikausgabe	354
34.8 Erstellen neuer Funktionen	354
34.9 Verschiedene Befehle	355
34.10 Dünnsbesetzte Matrizen	355
34.11 Beispiele	356
34.12 Freie Alternativen	357
35 Beispielprogramme in Matlab und C	359
35.1 LU-Zerlegung und Lösen von Dreieckssystemen	359
35.2 Polynominterpolation und Neville-Schema	362
35.3 Numerische Lösung gewöhnlicher Differenzialgleichungen	365
Anhang A – Weiterführende Themen	369
Anhang B – Literaturhinweise	371
Anhang C – Notation	375
Sachverzeichnis	377

Teil I

Numerische lineare Algebra

1.1 Aufgabenstellung

Die Numerik beschäftigt sich mit der praktischen Berechnung mathematischer Objekte wie beispielsweise

$$\int_0^1 e^{-x^2} dx, \quad \sin(20), \quad \min_{x \in [0,1]} F(x), \quad f(x) = 0, \quad Ax = b, \quad Ax = \lambda x,$$
$$y' = f(t, y).$$

Abstrakt lässt sich dies als Auswertung einer Abbildung formulieren.

Definition 1.1 Eine *mathematische Aufgabe* besteht in der Auswertung einer Abbildung $\phi: X \rightarrow Y$ bei $x \in X$.

Dabei ist zum Beispiel $\phi(x) = A^{-1}x$ oder $\phi(x) = \sin(x)$. Viele der oben aufgeführten Objekte sind nicht durch geschlossene Formeln definiert und können eventuell nur *approximativ* angegeben werden. Zudem stehen auf Computern nur endlich viele sogenannte *Maschinenzahlen* zur Verfügung, sodass nicht jede reelle Zahl exakt eingegeben und elementare Rechenoperationen wie $1/3$ nur näherungsweise bestimmt werden können. Dies führt auf *Rundungsfehler*. Weitere Fehlerquellen sind *Modellfehler*, die bei der vereinfachten mathematischen Beschreibung eines realen Vorgangs auftreten, sowie *Datenfehler*, die durch Messungen verursacht sein können. Viele dieser Ungenauigkeiten sind unvermeidbar und daher ist es in der Regel weder notwendig noch sinnvoll, ein mathematisches Problem exakt zu lösen. Durch *approximativer Lösung* lässt sich der *Rechenaufwand* häufig erheblich verringern. Die Berechnung der Determinante einer Matrix $A \in \mathbb{R}^{n \times n}$ mit dem Laplaceschen Entwicklungssatz führt beispielsweise auf $n!$ Rechenoperationen, was

für große Dimensionen n kaum in vertretbarer Zeit zu realisieren ist. Häufig lässt sich jedoch mit polynomiellem Aufwand zumindest approximativ eine Faktorisierung $A \approx LR$ mit Dreiecksmatrizen $L, R \in \mathbb{R}^{n \times n}$ konstruieren, mit deren Hilfe sich die Determinante $\det A \approx \det L \det R$ mit einem Aufwand bestimmen lässt, der mit n vergleichbar ist. Das Lösen linearer Gleichungssysteme $Ax = b$ ist damit eng verbunden. In der Praxis wird in der Regel nicht die inverse Matrix A^{-1} explizit bestimmt, sondern das Gleichungssystem direkt gelöst. Der Ausdruck $x = A^{-1}b$ steht daher in der Numerik für die Lösung des Gleichungssystems $Ax = b$ und nicht für die Multiplikation von b mit A^{-1} . Allgemeiner werden in der Numerik die folgenden typischen Fragestellungen diskutiert:

- Berechenbarkeit von Problemen (Algorithmik)
- Einfluss von Störungen (Konditionierung und Stabilität)
- Fehler zwischen berechneter und exakter Lösung (Konvergenz)
- Aufwand von Verfahren (Komplexität)

Ein wichtiges Ziel ist es, einen guten Kompromiss zwischen Genauigkeit und Aufwand eines Verfahrens zu erreichen. Dies wird für folgende Probleme untersucht:

- Lineare Gleichungssysteme
- Eigenwertprobleme
- Interpolation von Funktionen
- Integration von Funktionen
- Nullstellensuche und Optimierung
- Anfangswertprobleme

1.2 Kondition und Stabilität

Wir betrachten ein Beispiel, das die Auswirkungen von Störungen auf die Lösung eines Problems illustriert.

Beispiel 1.1 Für jedes $\varepsilon \in \mathbb{R} \setminus \{0\}$ ist die eindeutige Lösung des Gleichungssystems

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 + \varepsilon \end{bmatrix} x = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

gegeben durch $x = [2, 0]^\top$. Wir nehmen an, dass ε sehr klein ist und stören die rechte Seite in der zweiten Komponente, das heißt wir betrachten

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 + \varepsilon \end{bmatrix} \tilde{x} = \begin{bmatrix} 2 \\ 2 + \varepsilon \end{bmatrix}.$$

Die eindeutige Lösung ist gegeben durch $\tilde{x} = [1, 1]^\top$. Obwohl die Störung beliebig klein ist, unterscheiden sich die Lösungen x und \tilde{x} sehr stark.

Die Auswirkungen von Störungen auf die Lösung eines Problems führen auf den Begriff der Konditionierung.

Definition 1.2 Eine mathematische Aufgabe $\phi(x)$ heißt *schlecht konditioniert (an der Stelle x)*, wenn kleine Störungen der Daten große relative Fehler in der Lösung verursachen, das heißt wenn eine Störung \tilde{x} existiert mit

$$\frac{|\phi(\tilde{x}) - \phi(x)|}{|\phi(x)|} \gg \frac{|\tilde{x} - x|}{|x|},$$

wobei $x \neq 0$ und $\phi(x) \neq 0$ gelte. Andernfalls heißt die Aufgabe *gut konditioniert*.

Die Relation $a \gg b$ bedeutet, dass a wesentlich größer ist als b , zum Beispiel $a \geq 100b$. Was als wesentlich größer gilt, ist im Allgemeinen jedoch problemabhängig. Die Multiplikation zweier Zahlen ist gut konditioniert.

Satz 1.1 Die Aufgabe $\phi(x, y) = xy$ ist gut konditioniert in dem Sinne, dass für $x, y \in \mathbb{R}$ mit $x, y \neq 0$ und folglich $\phi(x, y) \neq 0$ sowie Störungen $\tilde{x}, \tilde{y} \in \mathbb{R}$ die relativen Fehler

$$\varepsilon_\phi = \frac{|\phi(\tilde{x}, \tilde{y}) - \phi(x, y)|}{|\phi(x, y)|}, \quad \varepsilon_x = \frac{|\tilde{x} - x|}{|x|}, \quad \varepsilon_y = \frac{|\tilde{y} - y|}{|y|}$$

die Abschätzung

$$\varepsilon_\phi \leq \varepsilon_x + \varepsilon_y + \varepsilon_x \varepsilon_y$$

erfüllen. Sind ε_x und ε_y klein, ist demnach der relative Fehler ε_ϕ ebenfalls klein.

Beweis Es gilt

$$\varepsilon_\phi = \frac{|\tilde{x}\tilde{y} - xy|}{|xy|} = \frac{|(\tilde{x}-x)\tilde{y} + x(\tilde{y}-y)|}{|xy|} \leq \frac{|\tilde{x}-x|}{|x|} \frac{|\tilde{y}-y|}{|y|} + \frac{|\tilde{y}-y|}{|y|}$$

und dies impliziert die Behauptung. \square

Bemerkung 1.1 Ebenfalls gut konditioniert sind die Addition zweier positiver oder zweier negativer Zahlen und die Inversion von Null verschiedener Zahlen. Schlecht konditioniert ist hingegen die Subtraktion nahezu gleich großer Zahlen, wie unten gezeigt wird.

Die gute Konditionierung einer Aufgabe ist offensichtlich notwendig, um das Problem numerisch sinnvoll lösen zu können, da Rundungsfehler andernfalls große Fehler verursachen könnten.

Definition 1.3 Ein *Verfahren* oder *Algorithmus* zur (näherungsweisen) Lösung einer Aufgabe ϕ ist eine Abbildung $\tilde{\phi} : X \rightarrow Y$, die durch die Hintereinanderausführung elementarer, möglicherweise rundungsfehlerbehafteter Rechenoperationen definiert ist, im einfachsten Fall

$$\tilde{\phi} = f_J \circ f_{J-1} \circ \cdots \circ f_1.$$

Beispiel 1.2 (i) Die Aufgabe $\phi(x) = x^4$ lässt sich realisieren durch $\tilde{\phi} = f \circ f$, mit der vom Rechner bereitgestellten Multiplikation $f(x) = x \square x$.

(ii) Die Wurzel $\phi(x) = \sqrt{x}$ einer Zahl $x > 0$ ist nach Heron gegeben als Grenzwert jeder Folge $z_{n+1} = (z_n + x/z_n)/2$ mit $z_0 > 0$. Damit kann $\tilde{\phi}$ als J -malige Anwendung der Iterationsvorschrift mit Initialisierung $z_0 = 1$ definiert werden.

Für eine Aufgabe sind in der Regel verschiedene Verfahren denkbar, aber selbst wenn die Aufgabe gut konditioniert ist, führen nicht alle Verfahren auf gute Ergebnisse, da sich Rundungsfehler im Laufe der Ausführung eines Verfahrens unterschiedlich auswirken können.

Beispiel 1.3 Die durch die Funktion

$$\phi(x) = \frac{1}{x} - \frac{1}{x+1} = \frac{1}{x(x+1)}$$

definierte Aufgabe ist für große Zahlen $x \in \mathbb{R}$ gut konditioniert, denn für eine Störung $\tilde{x} = (1 + \varepsilon_x)x$ mit einer kleinen Zahl ε_x erhalten wir

$$\phi(x) - \phi(\tilde{x}) = \frac{(1 + \varepsilon_x)x((1 + \varepsilon_x)x + 1) - x(x + 1)}{(1 + \varepsilon_x)x((1 + \varepsilon_x)x + 1)x(x + 1)} \approx \frac{2\varepsilon_x x^2}{x^4}.$$

Damit folgt für den relativen Fehler $\varepsilon_\phi \leq 2\varepsilon_x$, sofern $x \geq 1$ gilt. Die numerische Realisierung kann über die Verfahren

$$\tilde{\phi}_1(x) = \left(\frac{1}{x} \right) - \left(\frac{1}{x+1} \right), \quad \tilde{\phi}_2(x) = \frac{1}{(x(x+1))}$$

erfolgen, wobei die Klammerung die Reihenfolge der Ausführung der Operationen festlegt. Numerische Experimente zeigen, dass $\tilde{\phi}_1$ und $\tilde{\phi}_2$ für große Zahlen x stark voneinander abweichen.

Definition 1.4 Ein Algorithmus $\tilde{\phi}$ heißt *instabil*, wenn es eine Störung \tilde{x} von x gibt, so dass der durch Rundungsfehler und Störungen verursachte relative Fehler erheblich größer ist als der nur durch die Störung verursachte Fehler, das heißt, falls $\phi(x) \neq 0$ und

$$\frac{|\tilde{\phi}(\tilde{x}) - \phi(x)|}{|\phi(x)|} \gg \frac{|\phi(\tilde{x}) - \phi(x)|}{|\phi(x)|}.$$

Ein Algorithmus heißt *stabil*, wenn er nicht instabil ist.

Bemerkung 1.2 Notwendig für die Stabilität eines Algorithmus ist, dass jeder einzelne Rechenschritt eine gut konditionierte Aufgabe ist.

Der obige Algorithmus $\tilde{\phi}_1$ ist instabil aufgrund sogenannter *Auslöschungseffekte*, die bei der Subtraktion nahezu gleich großer Zahlen auftreten.

Beispiel 1.4 Für $x = 0.677354$ und $y = 0.677335$ ist $\phi(x, y) = x - y = 0.000019 = 0.19 \cdot 10^{-4}$. Für die Störung $\tilde{x} = (1 + \varepsilon_x)x$ mit $\varepsilon_x = 1.0 \cdot 10^{-4}$ folgt

$$\varepsilon_\phi = \frac{|\phi(\tilde{x}, y) - \phi(x, y)|}{|\phi(x, y)|} = \frac{\varepsilon_x x}{x - y} = \frac{0.677354 \cdot 10^{-4}}{0.19 \cdot 10^{-4}} \approx 3.565021$$

Die Störung von 0.01 % bewirkt also einen relativen Fehler von über 350 %.

Die Subtraktion nahezu gleich großer Zahlen ist also eine schlecht konditionierte Aufgabe.

Bemerkung 1.3 Der durch Rundung und näherungsweises Lösen verursachte Fehler bei der Lösung einer Aufgabe lässt sich mittels der Konditionierung der Aufgabe und der Stabilität des Verfahrens abschätzen, denn es gilt

$$|\phi(x) - \tilde{\phi}(\tilde{x})| \leq |\phi(x) - \phi(\tilde{x})| + |\phi(\tilde{x}) - \tilde{\phi}(\tilde{x})|. \quad \square$$

1.3 Aufwand

Neben der Stabilität eines numerischen Verfahrens ist der Rechenaufwand eine wichtige Größe.

Definition 1.5 Für eine Aufgabe $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ und ein zugehöriges Verfahren $\tilde{\phi} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ist der *Aufwand* die Anzahl der benötigten Rechenoperationen von $\tilde{\phi}$.

Eine exakte Bestimmung des Aufwands ist in der Regel nicht notwendig und es wird die Abhängigkeit von der Problemgröße n untersucht. Dabei ist die sogenannte *Landau-Notation* hilfreich.

Definition 1.6 Die Folge $(a_n)_{n \in \mathbb{N}}$ ist (*asymptotisch*) von der *Ordnung* der Folge $(b_n)_{n \in \mathbb{N}}$, falls Zahlen $c > 0$ und $N \in \mathbb{N}$ existieren, sodass $|a_n| \leq c|b_n|$ für alle $n \geq N$ gilt. In diesem Fall verwenden wir die *Landau-Notation* $a_n = \mathcal{O}(b_n)$.

Für den Aufwand a_n eines Verfahrens ist interessant, ob dieser von einer polynomiellen Ordnung n^p ist.

Beispiel 1.5 (i) Die Multiplikation eines Vektors $x \in \mathbb{R}^n$ mit einer fixierten Zahl $a \in \mathbb{R}$ ist von der Ordnung $\mathcal{O}(n)$.

(ii) Das Gaußsche Verfahren zur Lösung eines linearen Gleichungssystems besitzt den Aufwand $\mathcal{O}(n^3)$, während die Cramersche Regel auf einen Aufwand der Ordnung $\mathcal{O}(n!)$ führt.

1.4 Lernziele, Quiz und Anwendung

Sie sollten den Begriff der Konditionierung einer mathematischen Aufgabe erklären und anhand von Beispielen illustrieren können. Ferner sollten Sie die Stabilität eines Algorithmus definieren und mögliche Probleme wie Auslöschungseffekte beschreiben können. Die Landau-Notation sollten Sie erläutern und den Aufwand grundlegender Matrix-Operationen bestimmen können.

Quiz 1.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Es gilt $n^p = \mathcal{O}(\ln(1 + n))$ für jedes $0 < p \leq 1$	
Ein stabiler Algorithmus impliziert eine gute Konditionierung	
In der Praxis ist mit Auslöschungseffekten eher nicht zu rechnen	
Die Hintereinanderausführung zweier gut konditionierter Aufgaben ist gut konditioniert	
Ist ein lineares Gleichungssystem gut konditioniert für eine rechte Seite, so ist es für jede rechte Seite gut konditioniert	

Anwendung 1.1 Eine Vereinigung von n Ländern beschließt die Einführung einer Gemeinschaftswährung. Die Umrechnungskurse implizieren feste Wechselkurse zwischen den Landeswährungen, die mit m_{ij} bezeichnet seien. Es gilt $m_{ji} = m_{ij}^{-1}$. Für die praktische Umsetzung sollen Approximationen \tilde{m}_{ij} geeignet gewählt werden.

- (i) Wie groß dürfen die relativen Fehler $\varepsilon_{ij} = (\tilde{m}_{ij} - m_{ij})/m_{ij}$ höchstens sein, damit sich bei fünfmaligem beliebigem Wechseln höchstens eine relative Abweichung von 0.01 % ergibt?
- (ii) Alternativ können die Umrechnungskurse so gerundet werden, dass zum Beispiel sechs signifikante Dezimalstellen erhalten bleiben, das heißt etwa $\tilde{m}_{ij} = 0.00123456$ oder $\tilde{m}_{ij} = 12.3456$ falls $m_{ij} = 0.00123456789$ beziehungsweise $m_{ij} = 12.3456789$. Ist dieses Vorgehen sinnvoller?

2.1 Vektornormen

Um die Begriffe der Konditionierung und Stabilität präzisieren zu können, müssen Abstände zwischen Punkten im \mathbb{R}^n beziehungsweise Längen von Vektoren gemessen werden können.

Definition 2.1 Eine *Norm* auf \mathbb{R}^n ist eine Abbildung $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ mit den folgenden Eigenschaften:

- (i) $\|x\| = 0 \implies x = 0$ für alle $x \in \mathbb{R}^n$ (Definitheit);
- (ii) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{R}^n$ (Dreiecksungleichung);
- (iii) $\|\lambda x\| = |\lambda| \|x\|$ für alle $\lambda \in \mathbb{R}$ und $x \in \mathbb{R}^n$ (Homogenität).

Beispiel 2.1 Die ℓ^p -Normen sind für $1 \leq p \leq \infty$ und $x = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ definiert durch

$$\|x\|_p = \begin{cases} \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}, & p < \infty, \\ \max_{j=1,\dots,n} |x_j|, & p = \infty. \end{cases}$$

Die Norm $\|\cdot\|_2$ heißt *Euklidische Norm* und erfüllt $\|x\|_2^2 = x \cdot x = x^\top x$.

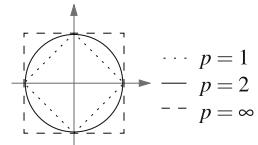
Bemerkungen 2.1 (i) Die ℓ^p -Normen sind äquivalent in dem Sinne, dass für alle $1 \leq p, q \leq \infty$ eine Konstante $c_{pq} > 0$ existiert, sodass für alle $x \in \mathbb{R}^n$ gilt

$$c_{pq}^{-1} \|x\|_p \leq \|x\|_q \leq c_{pq} \|x\|_p.$$

Die Konstante c_{pq} ist abhängig von p, q und n .

Abb. 2.1 Niveaumengen

$N_p(1)$ verschiedener ℓ^p -Normen in \mathbb{R}^2



(ii) Die ℓ^p -Normen unterscheiden sich durch ihre Niveaumengen

$$N_p(1) = \{x \in \mathbb{R}^n : \|x\|_p = 1\},$$

s. Abb. 2.1. □

2.2 Matrixnormen

Wir identifizieren im Folgenden stets Matrizen $A \in \mathbb{R}^{m \times n}$ mit linearen Abbildungen $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$, wobei dabei die jeweiligen kanonischen Basen gewählt seien. Eine lineare Abbildung wird auch als linearer Operator bezeichnet.

Definition 2.2 Für Normen $\|\cdot\|_{\mathbb{R}^m}$ und $\|\cdot\|_{\mathbb{R}^n}$ auf \mathbb{R}^m beziehungsweise \mathbb{R}^n ist die (*induzierte*) **Operatornorm** für alle $A \in \mathbb{R}^{m \times n}$ definiert durch

$$\|A\|_{op} = \sup_{x \in \mathbb{R}^n, \|x\|_{\mathbb{R}^n}=1} \|Ax\|_{\mathbb{R}^m}.$$

Die Operatornorm misst, wie stark Niveaumengen verformt werden.

Beispiel 2.2 Durch eine symmetrische Matrix $A \in \mathbb{R}^{2 \times 2}$ wird die kreisförmige Niveaumenge $N_2(1)$ auf eine Ellipse abgebildet, die im Kreis mit Radius $\|A\|_2$ enthalten ist.

Die Operatornorm definiert eine Norm mit folgenden Eigenschaften.

Lemma 2.1 Zu fixierten Normen $\|\cdot\|$ auf \mathbb{R}^n beziehungsweise \mathbb{R}^m sei $\|\cdot\|_{op}$ die induzierte Operatornorm auf $\mathbb{R}^{m \times n}$. Dann gilt:

- (i) $\|\cdot\|_{op}$ definiert eine Norm auf $\mathbb{R}^{m \times n}$;
- (ii) $\|A\|_{op} = \sup_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\| = \inf\{c > 0 : \forall x \in \mathbb{R}^n \|Ax\| \leq c\|x\|\}$;
- (iii) für $A \neq 0$ und $x \in \mathbb{R}^n$ mit $\|x\| \leq 1$ und $\|Ax\| = \|A\|_{op}$ folgt $\|x\| = 1$;
- (iv) das Infimum und das Supremum in (ii) werden angenommen.

Beweis Übungsaufgabe. □

Bemerkung 2.2 Aus (ii) folgt $\|Ax\| \leq \|A\|_{op}\|x\|$ für alle $x \in \mathbb{R}^n$.

Für einige ℓ^p -Normen lassen sich die induzierten Operatornormen explizit angeben. Die Einträge einer Matrix $A \in \mathbb{R}^{m \times n}$ seien mit a_{ij} , $1 \leq i \leq m$, $1 \leq j \leq n$, bezeichnet.

Beispiele 2.3 (i) Die ℓ^1 -Norm auf \mathbb{R}^m und \mathbb{R}^n induziert die *Spaltensummennorm*

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^m |a_{ij}| .$$

(ii) Die ℓ^∞ -Norm auf \mathbb{R}^m und \mathbb{R}^n induziert die *Zeilensummennorm*

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^n |a_{ij}| .$$

(iii) Die ℓ^2 -Norm auf \mathbb{R}^m und \mathbb{R}^n induziert die *Spektralnorm*

$$\|A\|_2 = \sqrt{\rho(A^\top A)} = (\max \{|\lambda| : \lambda \text{ ist Eigenwert von } A^\top A\})^{1/2} .$$

Die Zahl $\rho(A^\top A)$ heißt *Spektralradius* von $A^\top A$.

Einige weitere Eigenschaften der Operatornorm sind die folgenden.

Lemma 2.2 Seien Normen auf \mathbb{R}^ℓ , \mathbb{R}^m und \mathbb{R}^n fixiert und die induzierten Operatornormen mit $\|\cdot\|$ bezeichnet.

- (i) Für $A \in \mathbb{R}^{\ell \times m}$ und $B \in \mathbb{R}^{m \times n}$ gilt $\|AB\| \leq \|A\|\|B\|$.
- (ii) Die Einheitsmatrix $I_n \in \mathbb{R}^{n \times n}$ erfüllt $\|I_n\| = 1$.
- (iii) Jede induzierte Operatornorm auf $\mathbb{R}^{n \times n}$ erfüllt $\|A\|_{op} \geq |\lambda|$ für alle symmetrischen Matrizen $A \in \mathbb{R}^{n \times n}$ und jeden Eigenwert λ von A .

Beweis Nach dem vorigen Lemma gilt $\|ABx\| \leq \|A\|\|Bx\| \leq \|A\|\|B\|\|x\|$ und dies impliziert $\|AB\| \leq \|A\|\|B\|$. Die anderen Aussagen folgen unmittelbar aus der Definition der Operatornorm. \square

Die Euklidische Norm lässt sich in naheliegender Weise auf $\mathbb{R}^{m \times n}$ definieren, ist aber keine induzierte Operatornorm.

Beispiel 2.4 Die *Frobenius-Norm* einer Matrix $A \in \mathbb{R}^{m \times n}$ ist definiert durch $\|A\|_{\mathcal{F}} = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2}$. Sie ist keine induzierte Operatornorm für $n > 1$, da $\|I_n\|_{\mathcal{F}} = \sqrt{n}$ gilt. Auch die skalierte Frobeniusnorm $n^{-1/2}\|A\|_{\mathcal{F}}$ ist keine induzierte Operatornorm, denn diese verletzt die Eigenschaft $\|A\|_{op} \geq |\lambda|$ für jeden Eigenwert λ von A .

2.3 Konditionszahl

Mit Hilfe des Begriffs der Operatornorm lässt sich die Konditionierung eines linearen Gleichungssystems präzisieren.

Satz 2.1 Sei $\|\cdot\|$ eine Operatornorm auf $\mathbb{R}^{n \times n}$. Sei $A \in \mathbb{R}^{n \times n}$ regulär und seien $x, \tilde{x}, b, \tilde{b} \in \mathbb{R}^n$, sodass

$$Ax = b, \quad A\tilde{x} = \tilde{b}.$$

Dann gilt

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|b - \tilde{b}\|}{\|b\|}$$

Beweis Es gilt $\|x - \tilde{x}\| = \|A^{-1}(b - \tilde{b})\| \leq \|A^{-1}\| \|b - \tilde{b}\|$ sowie $\|b\| = \|Ax\| \leq \|A\| \|x\|$ beziehungsweise $\|x\| \geq \|A\|^{-1} \|b\|$. Damit folgt

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \frac{\|A^{-1}\| \|b - \tilde{b}\|}{\|x\|} \leq \frac{\|A^{-1}\| \|b - \tilde{b}\|}{\|A\|^{-1} \|b\|},$$

also die behauptete Abschätzung. \square

Das Produkt $\|A\| \|A^{-1}\|$ kontrolliert die Verstärkung des relativen Fehlers beim Lösen eines linearen Gleichungssystems.

Definition 2.3 Die *Konditionszahl* einer regulären Matrix $A \in \mathbb{R}^{n \times n}$ bezüglich der durch die Norm $\|\cdot\|$ auf \mathbb{R}^n induzierten Operatornorm ist definiert durch

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

Im Fall einer ℓ^p -Norm schreiben wir cond_p statt $\text{cond}_{\|\cdot\|_p}$.

Bemerkungen 2.3 (i) Die Konditionszahl einer Matrix ist stets nach unten durch 1 beschränkt, denn für jede Operatornorm gilt $1 = \|AA^{-1}\| \leq \|A\| \|A^{-1}\| = \text{cond}_{\|\cdot\|}(A)$.

(ii) Ist A symmetrisch mit Eigenwerten $\lambda_1, \dots, \lambda_n$, so gilt

$$\text{cond}_2(A) = \frac{\max_{j=1,\dots,n} |\lambda_j|}{\min_{k=1,\dots,n} |\lambda_k|}.$$

Wir betrachten die Konditionszahl der Matrix aus dem eingehenden Beispiel 1.1, in dem Störungen der rechten Seite große Fehler verursachten.

Beispiel 2.5 Die Matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{bmatrix}$ besitzt die Eigenwerte $\lambda_{1,2} = 1 + \varepsilon/2 \pm (1 + \varepsilon^2/4)^{1/2}$. Mit der Taylor-Approximation $(1+x)^{1/2} \approx 1+x/2$ folgt, dass für kleine Zahlen ε gilt $\lambda_1 \approx 2 + \varepsilon/2$ und $\lambda_2 \approx \varepsilon/2$. Damit folgt $\text{cond}_2(A) \approx 4\varepsilon^{-1}$, was das empfindliche Verhalten zugehöriger Gleichungssysteme gegenüber Störungen erklärt.

Geometrisch interpretiert misst die Konditionszahl die durch die lineare Abbildung A definierte Verzerrung, ist jedoch unabhängig von uniformen Skalierungen.

Beispiel 2.6 Für eine symmetrische Matrix $A \in \mathbb{R}^{2 \times 2}$ beschreibt $\text{cond}_2(A)$ das Verhältnis der Radien der Ellipse $A(N_2(1))$.

2.4 Lernziele, Quiz und Anwendung

Ihnen sollten verschiedene Charakterisierungen der Operatornorm sowie einige konkrete Beispiele bekannt sein. Sie sollten die Konditionszahl definieren und deren Bedeutung für die approximative Lösung linearer Gleichungssysteme erklären können.

Quiz 2.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Ist ein lineares Gleichungssystem bezüglich einer Operatornorm gut konditioniert, so auch bezüglich jeder anderen Operatornorm	
Für $A, B \in \mathbb{R}^{n \times n}$ und $\lambda, \mu \in \mathbb{R}$ und eine beliebige Operatornorm $\ \cdot\ $ gilt $\ \lambda A + \mu B\ \leq \lambda \ A\ + \mu \ B\ $	
Für $A = \begin{bmatrix} -2 & 4 \\ 0 & 5 \end{bmatrix}$ gilt $\ A\ _\infty = 6$ und $\ A\ _1 = 9$	
Für $A \in \mathbb{R}^{m \times n}$ und $B \in \mathbb{R}^{n \times p}$ gilt $\ker AB = \ker A$	
Ist λ ein Eigenwert von A , so gilt $\ A\ \leq \lambda $ für jede Operatornorm	

Anwendung 2.1 Die Routen zweier in einer Ebene fliegenden Flugzeuge seien gegeben durch $t \mapsto x^i + tv^i$ mit $x^i, v^i \in \mathbb{R}^2$ für $i = 1, 2$, wobei $\|v^i\|_2 = 350 \text{ km/h}$ gelte. Berechnen Sie den Schnittpunkt der Linien und die Zeitpunkte, an denen sich die Flugzeuge an diesem befinden. Wie groß dürfen Messfehler bei der Bestimmung der Anfangspositionen x^i , $i = 1, 2$, höchstens sein, damit der Fehler bei der Berechnung des Schnittpunkts weniger als 5 km beträgt?

3.1 Dreiecksmatrizen

In kanonischer Weise lassen sich lineare Gleichungssysteme lösen, die durch eine Dreiecksmatrix definiert sind. Dies motiviert die Faktorisierung von Matrizen mittels Dreiecksmatrizen. Wir folgen in diesem Kapitel der Darstellung in [8].

Definition 3.1 Eine Matrix $L \in \mathbb{R}^{n \times n}$ heißt *untere Dreiecksmatrix*, falls $\ell_{ij} = 0$ für $i < j$ gilt. Eine Matrix $U \in \mathbb{R}^{n \times n}$ heißt *obere Dreiecksmatrix*, falls U^\top untere Dreiecksmatrix ist. Eine Dreiecksmatrix $D \in \mathbb{R}^{n \times n}$ heißt *normalisiert*, falls $d_{ii} = 1$ für $i = 1, 2, \dots, n$ gilt.

Gleichungssysteme mit regulärer Dreiecksmatrix lassen sich mittels Rückwärts- beziehungsweise Vorwärtssubstitution lösen. Die Diagonalelemente einer regulären Dreiecksmatrix U sind wegen $0 \neq \det U = u_{11}u_{22} \dots u_{nn}$ von Null verschieden.

Algorithmus 3.1 (Rückwärtssubstitution) Seien $U \in \mathbb{R}^{n \times n}$ eine reguläre obere Dreiecksmatrix und $b \in \mathbb{R}^n$. Berechne $x \in \mathbb{R}^n$ durch:

$$\text{for } i = n : -1 : 1; \quad x_i = \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right) / u_{ii}; \quad \text{end}$$

Bemerkung 3.1 Im i -ten Schritt werden $n - i$ viele Multiplikationen und Subtraktionen sowie eine Division durchgeführt, sodass der Gesamtaufwand der Rückwärtssubstitution gegeben ist durch

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

Die Mengen der regulären unteren und oberen Dreiecksmatrizen bilden Gruppen.

Lemma 3.1 Seien $U, V \in \mathbb{R}^{n \times n}$ obere Dreiecksmatrizen. Dann ist UV eine obere Dreiecksmatrix und falls U regulär ist, so ist auch U^{-1} eine obere Dreiecksmatrix mit Diagonaleinträgen u_{ii}^{-1} , $i = 1, 2, \dots, n$.

Beweis Übungsaufgabe. □

3.2 LU-Zerlegung

Ist eine Faktorisierung $A = LU$ einer regulären Matrix $A \in \mathbb{R}^{n \times n}$ in eine untere (*lower*) und eine obere (*upper*) Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ beziehungsweise $U \in \mathbb{R}^{n \times n}$ gegeben, so lässt sich das lineare Gleichungssystem $Ax = b$ in zwei Schritten lösen:

- (i) Löse $Ly = b$.
- (ii) Löse $Ux = y$.

Es gilt dann $Ax = (LU)x = L(Ux) = Ly = b$. Störungen werden im ersten Schritt mit $\text{cond}(L)$ und im zweiten mit $\text{cond}(U)$ verstärkt, insgesamt also mit $\text{cond}(L)\text{cond}(U)$. Das Verfahren ist also nur stabil, falls $\text{cond}(L)\text{cond}(U) \approx \text{cond}(A)$ gilt. Dies ist im Allgemeinen nicht der Fall.

Beispiel 3.1 Für $A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 0 \end{bmatrix}$ mit $0 < \varepsilon \ll 1$ ist $A^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -\varepsilon \end{bmatrix}$ und es gilt $\|A\|_\infty = \|A^{-1}\|_\infty = 1 + \varepsilon$ also $\text{cond}_\infty(A) = (1 + \varepsilon)^2 \approx 1$. Eine Faktorisierung ist gegeben durch

$$L = \begin{bmatrix} 1 & 0 \\ \varepsilon^{-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \varepsilon & 1 \\ 0 & -\varepsilon^{-1} \end{bmatrix}.$$

Es gilt $\|L\|_\infty = \|L^{-1}\|_\infty = 1 + \varepsilon^{-1}$ sowie $\|U\|_\infty = \varepsilon^{-1}$, $\|U^{-1}\|_\infty = 1 + \varepsilon^{-1}$, also

$$\text{cond}_\infty(L) = (1 + \varepsilon^{-1})^2 \approx \varepsilon^{-2}, \quad \text{cond}_\infty(U) = (1 + \varepsilon^{-1})/\varepsilon \approx \varepsilon^{-2}.$$

Definition 3.2 Eine Faktorisierung $A = LU$ mit unterer Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ und oberer Dreiecksmatrix $U \in \mathbb{R}^{n \times n}$ heißt *LU-Zerlegung* von A . Sie heißt *normalisiert*, falls L normalisiert ist, das heißt auf der Diagonalen von L stehen nur Einsen.

Satz 3.1 Für eine reguläre Matrix $A \in \mathbb{R}^{n \times n}$ sind folgende Aussagen äquivalent:

- (i) Es existiert eine eindeutig bestimmte normalisierte LU-Zerlegung von A .
- (ii) Alle Untermatrizen $A_k = (a_{ij})_{1 \leq i,j \leq k} \in \mathbb{R}^{k \times k}$ von A sind regulär.

Beweis (i) \implies (ii). Gilt $A = LU$ und ist A regulär, so sind auch L und U regulär, denn $0 \neq \det(A) = \det(L)\det(U)$. Ferner sind sämtliche Untermatrizen L_k und U_k regulär, da beispielsweise $\det(L) = \ell_{11}\ell_{22} \dots \ell_{nn}$ gilt. Da für jede Untermatrix A_k die Zerlegung $A_k = L_k U_k$ gilt, folgt die Regularität von A_k .

(ii) \implies (i). Für $n = 1$ ist die Implikation klar und sie sei für $n - 1$ bewiesen. Dann existiert eine eindeutig bestimmte normalisierte LU-Zerlegung $A_{n-1} = L_{n-1}U_{n-1}$. Die Vektoren $[b^\top, a_{nn}]$ und $[c^\top, a_{nn}]$ seien die letzte Spalte beziehungsweise Zeile von A . Zum Beweis der Aussage für n genügt es zu zeigen, dass eindeutig bestimmte Vektoren $\ell, u \in \mathbb{R}^{n-1}$ und $r \in \mathbb{R}$ existieren mit

$$\begin{bmatrix} A_{n-1} & b \\ c^\top & a_{nn} \end{bmatrix} = \begin{bmatrix} L_{n-1} & 0 \\ \ell^\top & 1 \end{bmatrix} \begin{bmatrix} U_{n-1} & u \\ 0 & r \end{bmatrix} = \begin{bmatrix} L_{n-1}U_{n-1} & L_{n-1}u \\ (U_{n-1}^\top\ell)^\top & \ell^\top u + r \end{bmatrix}.$$

Wegen $A_{n-1} = L_{n-1}U_{n-1}$ ist dies äquivalent zu

$$b = L_{n-1}u, \quad c = U_{n-1}^\top\ell, \quad a_{nn} = \ell^\top u + r.$$

Da L_{n-1} und U_{n-1} regulär sind, existieren eindeutig bestimmte Lösungen u und ℓ , die dann r eindeutig festlegen. \square

Beispiele 3.2 (i) Ist A positiv definit, das heißt gilt $Ax \cdot x > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$, oder strikt diagonaldominant, das heißt gilt $\sum_{j=1, \dots, n, j \neq i} |a_{ij}| < |a_{ii}|$ für $i = 1, 2, \dots, n$, so besitzt A eine LU-Zerlegung.

(ii) Die Matrix $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ besitzt keine LU-Zerlegung.

Die LU-Zerlegung einer Matrix lässt sich sehr einfach bestimmen.

Lemma 3.2 Ist $A = LU$ eine normalisierte LU-Zerlegung von A , so folgt

$$a_{ik} = u_{ik} + \sum_{j=1}^{i-1} \ell_{ij} u_{jk}, \quad a_{ki} = \ell_{ki} u_{ii} + \sum_{j=1}^{i-1} \ell_{kj} u_{ji}.$$

Beweis Wegen $\ell_{ij} = 0$ für $j > i$ und $\ell_{jj} = 1$ gilt

$$a_{ik} = \sum_{j=1}^n \ell_{ij} u_{jk} = \sum_{j=1}^i \ell_{ij} u_{jk} = u_{ik} + \sum_{j=1}^{i-1} \ell_{ij} u_{jk}$$

und wegen $u_{ji} = 0$ für $j > i$ gilt

$$a_{ki} = \sum_{j=1}^n \ell_{kj} u_{ji} = \sum_{j=1}^i \ell_{kj} u_{ji} = \ell_{ki} u_{ii} + \sum_{j=1}^{i-1} \ell_{kj} u_{ji}.$$

Dies beweist die Behauptung. \square

Die Formeln des Lemmas lassen sich nach u_{ik} für $i \leq k$ beziehungsweise wegen $u_{ii} \neq 0$ nach ℓ_{ki} für $k > i$ auflösen.

Algorithmus 3.2 (LU -Zerlegung) Die Matrix $A \in \mathbb{R}^{n \times n}$ besitze eine normalisierte LU -Zerlegung. Die nichttrivialen Einträge von L und U sind gegeben durch:

```

for i = 1 : n
    for k = i : n;    $u_{ik} = a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} u_{jk};$    end
        for k = i + 1 : n;    $\ell_{ki} = \left( a_{ki} - \sum_{j=1}^{i-1} \ell_{kj} u_{ji} \right) / u_{ii};$    end
    end

```

Bemerkungen 3.2 (i) Die Berechnung von u_{ik} erfordert $i - 1$ Multiplikationen und Subtraktionen, bei der von ℓ_{ki} ist zusätzlich eine Division erforderlich, sodass im i -ten Schritt

$$(n - i + 1)2(i - 1) + (n - i)(2(i - 1) + 1) = (4n + 5)i - 4i^2 - (3n + 2)$$

Operationen anfallen. Durch Summation über $i = 1, 2, \dots, n$ ergibt sich der Gesamtrechenaufwand $2n^3/3 + \mathcal{O}(n^2)$.

(ii) Die Einträge von A können sukzessive durch die nichttrivialen Einträge von L und U überschrieben werden, es ist also kein zusätzlicher Speicherplatz notwendig.

3.3 Cholesky-Zerlegung

Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch, so sind lediglich $n(n+1)/2$ viele Einträge von A relevant und es ist naheliegend, nach einer Faktorisierung $A = LL^\top$ mit einer unteren Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ zu suchen. Notwendig dafür ist, dass A symmetrisch und positiv semidefinit ist, denn die Faktorisierung impliziert, dass

$$\begin{aligned} A^\top &= (LL^\top)^\top = LL^\top = A, \\ x^\top Ax &= x^\top (LL^\top)x = (L^\top x)^\top (L^\top x) = \|L^\top x\|_2^2 \geq 0. \end{aligned}$$

Ist A oder L regulär, so folgt, dass A positiv definit sein muss. In diesem Fall sind die Bedingungen für die Existenz der Cholesky-Zerlegung auch hinreichend und implizieren deren Eindeutigkeit.

Definition 3.3 Die Matrix $A \in \mathbb{R}^{n \times n}$ heißt *positiv definit*, falls für alle $x \in \mathbb{R}^n \setminus \{0\}$ gilt, dass $x^\top Ax > 0$. Gilt nur $x^\top Ax \geq 0$ für alle $x \in \mathbb{R}^n$, so heißt A *positiv semidefinit*.

Lemma 3.3 Sei A symmetrisch und positiv definit. Dann gilt $\det A > 0$ und alle Untermatrizen $A_k = (a_{ij})_{1 \leq i,j \leq k}$ sind positiv definit.

Beweis Übungsaufgabe. □

Definition 3.4 Eine Faktorisierung $A = LL^\top$ mit einer unteren Dreiecksmatrix L heißt *Cholesky-Zerlegung*.

Satz 3.2 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann existiert eine eindeutig bestimmte untere Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ mit $A = LL^\top$ und $\ell_{ii} > 0$ für $i = 1, 2, \dots, n$.

Beweis Ist $n = 1$, so gilt $a_{11} > 0$ und die Konstruktion folgt durch Wahl von $\ell_{11} = \sqrt{a_{11}}$. Die Teilmatrix $A_{n-1} = (a_{ij})_{1 \leq i,j \leq n-1}$ ist positiv definit und symmetrisch und es sei $A_{n-1} = L_{n-1}L_{n-1}^\top$ eine Faktorisierung mit den genannten Eigenschaften. Es sei $[b^\top, a_{nn}]$ die letzte Zeile von A und es sind ein Vektor $c \in \mathbb{R}^{n-1}$ und eine Zahl $\alpha > 0$ zu konstruieren, sodass

$$\begin{bmatrix} A_{n-1} & b \\ b^\top & a_{nn} \end{bmatrix} = \begin{bmatrix} L_{n-1} & 0 \\ c^\top & \alpha \end{bmatrix} \begin{bmatrix} L_{n-1}^\top & c \\ 0 & \alpha \end{bmatrix} = \begin{bmatrix} L_{n-1}L_{n-1}^\top & L_{n-1}c \\ (L_{n-1}c)^\top & \alpha^2 + c^\top c \end{bmatrix}$$

gilt. Da $A_{n-1} = L_{n-1}L_{n-1}^\top$ ist dies äquivalent zu den Gleichungen $L_{n-1}c = b$ und $c^\top c + \alpha^2 = a_{nn}$. Da L_{n-1} positive Diagonaleinträge hat, ist L_{n-1} regulär und c ist eindeutig bestimmt. Um die zweite Gleichung mit einer reellen Zahl $\alpha > 0$ lösen zu können, müssen wir $\alpha^2 = a_{nn} - c^\top c > 0$ beweisen. Es gilt

$$\det A = \det \begin{bmatrix} L_{n-1} & 0 \\ c^\top & \alpha \end{bmatrix} \det \begin{bmatrix} L_{n-1}^\top & c \\ 0 & \alpha \end{bmatrix} = \alpha^2 (\det L_{n-1})^2.$$

Wegen $\det A > 0$ und $\det L_{n-1} > 0$ folgt $\alpha^2 > 0$, das heißt es existiert ein eindeutiges $\alpha > 0$, welches die Faktorisierung komplettiert. □

Die Faktorisierungen lassen sich wieder durch Koeffizientenvergleich bestimmen.

Lemma 3.4 Gilt $A = LL^\top$ so folgt

$$a_{ik} = \begin{cases} \ell_{ik}\ell_{kk} + \sum_{j=1}^{k-1} \ell_{ij}\ell_{kj} & \text{für } i > k, \\ \ell_{kk}^2 + \sum_{j=1}^{k-1} \ell_{kj}^2 & \text{für } i = k. \end{cases}$$

Beweis Da $\ell_{kj} = 0$ für $j > k$ gilt, folgt

$$a_{ik} = \sum_{j=1}^n \ell_{ij} \ell_{kj} = \sum_{j=1}^k \ell_{ij} \ell_{kj}$$

und dies impliziert die Behauptung. \square

Die Identitäten lassen sich nach ℓ_{kk} und ℓ_{ik} auflösen.

Algorithmus 3.3 (LL^\top -Zerlegung) Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Die nichttrivialen Einträge von L sind gegeben durch:

```

for k = 1 : n
     $\ell_{kk} = \left( a_{kk} - \sum_{j=1}^{k-1} \ell_{kj}^2 \right)^{1/2}$ 
    for i = k + 1 : n;
         $\ell_{ik} = \left( a_{ik} - \sum_{j=1}^{k-1} \ell_{ij} \ell_{kj} \right) / \ell_{kk};$ 
    end
end

```

Bemerkung 3.3 Der Algorithmus berechnet die Cholesky-Zerlegung mit $n^3/3 + \mathcal{O}(n^2)$ Operationen.

Beispiel 3.3 Die Matrix $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ ist positiv definit, falls $a > 0$ und $ca - b^2 > 0$ gilt.

In diesem Fall erhält man $A = LL^\top$ mit

$$L = \begin{bmatrix} a^{1/2} & 0 \\ b/a^{1/2} & (c - b^2/a)^{1/2} \end{bmatrix}.$$

Die Lösung eines linearen Gleichungssystems lässt sich mit Hilfe der Cholesky-Zerlegung folgendermaßen bestimmen:

$$(i) \text{ Löse } Ly = b. \quad (ii) \text{ Löse } L^\top x = y.$$

Um zu zeigen, dass dies einen stabilen Algorithmus definiert, verwenden wir, dass die Spektralnorm einer Matrix $M \in \mathbb{R}^{n \times n}$ gegeben ist durch

$$\|M\|_2^2 = \rho(M^\top M) = \max\{|\lambda| : \lambda \text{ ist Eigenwert von } M^\top M\}.$$

Ist M symmetrisch, so gilt $\|M\|_2 = \rho(M)$.

Satz 3.3 Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, so gilt für die Cholesky-Zerlegung $A = LL^\top$, dass

$$\text{cond}_2(L) = \text{cond}_2(L^\top) = (\text{cond}_2(A))^{1/2}.$$

Beweis Die symmetrischen aber im Allgemeinen verschiedenen Matrizen $L^\top L$ und LL^\top haben dieselben Eigenwerte, denn da L regulär ist, gilt für alle $x \in \mathbb{R}^n$ und $\lambda \in \mathbb{R}$

$$L^\top Lx = \lambda x \iff LL^\top(Lx) = \lambda(Lx).$$

Mit $\rho(LL^\top) = \rho(L^\top L)$ folgt $\|L\|_2 = \|L^\top\|_2$ und analog $\|L^{-1}\|_2 = \|L^{-\top}\|_2$. Dies impliziert $\text{cond}_2(L) = \text{cond}_2(L^\top)$. Mit $LL^\top = A$ und da A symmetrisch ist, gilt

$$\|L\|_2^2 = \|L^\top\|_2^2 = \rho(LL^\top) = \rho(A) = \|A\|_2$$

sowie

$$\|L^{-1}\|_2^2 = \rho(L^{-\top}L^{-1}) = \rho((LL^\top)^{-1}) = \rho(A^{-1}) = \|A^{-1}\|_2.$$

Mit den Identitäten folgt insgesamt

$$\text{cond}_2(L) = \|L\|_2\|L^{-1}\|_2 = \|A\|_2^{1/2}\|A^{-1}\|_2^{1/2} = (\text{cond}_2(A))^{1/2}.$$

Dies beweist die Behauptung. \square

3.4 Lernziele, Quiz und Anwendung

Sie sollten die LU - und Cholesky-Faktorisierungen definieren, hinreichende und notwendige Bedingungen für deren Existenz benennen sowie Algorithmen zur praktischen Berechnung herleiten können. Aufwands- und Stabilitätseigenschaften der Lösung linearer Gleichungssysteme mittels der Faktorisierungen sollten Sie erläutern können.

Quiz 3.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Gilt $x^\top Ax < 0$ für alle $x \in \mathbb{R}^n$, so besitzt A eine LU -Zerlegung	
Besitzt A eine LU -Zerlegung und ist A symmetrisch, so folgt $U = L^\top$	
Ist A invertierbar mit Cholesky-Zerlegung $A = LL^\top$, so definiert $L^{-\top}L$ eine Cholesky-Zerlegung von A^{-1}	
Ist eine Cholesky-Zerlegung $A = LL^\top$ gegeben, so lässt sich das lineare Gleichungssystem $Ax = b$ mit dem Aufwand $\mathcal{O}(n^2)$ lösen	
Ist A symmetrisch und invertierbar, so ist A positiv definit	

Anwendung 3.1 Zur Bewertung von Finanzderivaten wie beispielsweise Optionen ist die Simulation mehrdimensionaler Brownscher Bewegungen erforderlich. Dazu werden n -dimensionale Zufallsvariablen benötigt, die einer korrelierten Normalverteilung folgen, das heißt $X \sim N(\mu, \Sigma)$ mit einem Erwartungswert $\mu \in \mathbb{R}^n$ und einer symmetrischen, positiv definiten Kovarianzmatrix $\Sigma \in \mathbb{R}^{n \times n}$, das heißt es gilt

$$\Sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)]$$

sowie $\mu_i = E(X_i)$ für $i, j = 1, 2, \dots, n$. Ist Y eine standard-normalverteilte vektorielle Zufallsvariable, das heißt gilt $Y \sim N(0, I_n)$, und ist $\Sigma = LL^\top$ die Cholesky-Zerlegung von Σ , so erhält man mittels $X = \mu + LY$ eine Zufallsvariable mit $X \sim N(\mu, \Sigma)$. In MATLAB lässt sich eine Realisierung von X mit Hilfe von Pseudozufallsvariablen erzeugen durch `X=mu+L*randn(n, 1)`. Verwenden Sie $n = 3$

$$\Sigma = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 5 & 1 \\ 0 & 1 & 5 \end{bmatrix}, \quad \mu = \begin{bmatrix} -5 \\ 0 \\ 5 \end{bmatrix},$$

generieren Sie 1000 Realisierungen der Variable X und stellen Sie die Histogramme der Komponenten X_i mit Hilfe des Kommandos `hist` im Bereich $[-10, 10]$ für $i = 1, 2, 3$ dar.

Eliminationsverfahren

4.1 Gaußsches Eliminationsverfahren

Lineare Gleichungssysteme treten in verschiedensten Bereichen der Mathematik auf. Sie erlauben es, aus gewissen äußereren, messbaren Größen (approximativ) innere Größen zu bestimmen, die oft nicht direkt zugänglich sind.

Beispiel 4.1 Lässt sich der Gesamtwert von in einem Glas befindlichen Münzen durch das Gewicht und das Volumen bestimmen?

Das Gaußsche Verfahren überführt ein lineares Gleichungssystem sukzessive in ein äquivalentes lineares Gleichungssystem mit oberer Dreiecksmatrix. Wir folgen in diesem Kapitel der Darstellung in [8].

Algorithmus 4.1 (Gauß-Elimination) Seien $A \in \mathbb{R}^{n \times n}$ und $b \in \mathbb{R}^n$.

- (1) Setze $A^{(1)} = A$ und $b^{(1)} = b$ sowie $k = 1$.
- (2) Für $A^{(k)}$ gelte $a_{ij}^{(k)} = 0$ für und $1 \leq j \leq k-1$ und $i \geq j+1$ und mit $\ell_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ für $i = k+1, \dots, n$ sei die normalisierte untere Dreiecksmatrix $L^{(k)} \in \mathbb{R}^{n \times n}$ wie folgt definiert:

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & a_{1n}^{(1)} \\ \ddots & & & \vdots \\ & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & \vdots & & \vdots \\ & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}, \quad L^{(k)} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & -\ell_{k+1,k} & \\ & & \vdots & \ddots \\ & & -\ell_{nk} & 1 \end{bmatrix}.$$

Dann gilt für $A^{(k+1)} = L^{(k)}A^{(k)}$, dass $a_{ij}^{(k+1)} = 0$ für $1 \leq j \leq k$ und $i \geq j + 1$, das heißt

$$A^{(k+1)} = \begin{bmatrix} a_{11}^{(1)} & \dots & & a_{1n}^{(1)} \\ \ddots & & & \vdots \\ & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} \\ & \vdots & & \vdots \\ & a_{n,k+1}^{(k+1)} & \dots & a_{nn}^{(k+1)} \end{bmatrix}.$$

Setze ferner $b^{(k+1)} = L^{(k)}b^{(k)}$.

(3) Stoppe falls $k + 1 = n$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (2).

Satz 4.1 Ist $A \in \mathbb{R}^{n \times n}$ regulär, so ist das Gauß-Verfahren genau dann durchführbar, wenn A eine LU-Zerlegung besitzt. Das Verfahren liefert dann die normalisierte LU-Zerlegung mit $U = A^{(n)}$ und $L = (L^{(n-1)} \dots L^{(1)})^{-1}$. Die modifizierte rechte Seite $y = b^{(n)}$ ist gegeben durch $y = L^{-1}b$ und die Lösung des linearen Gleichungssystems $Ax = b$ durch die Lösung des Systems $Ux = y$.

Beweis

(i) Die Matrix A besitze eine LU-Zerlegung. Das Gauß-Verfahren ist durchführbar, falls $a_{kk}^{(k)} \neq 0$ in jedem Schritt gilt. Wir betrachten die linke, obere $k \times k$ -Teilmatrix $A_k^{(k)}$ von $A^{(k)} = L^{(k-1)} \dots L^{(1)}A$, das heißt

$$A_k^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{kk}^{(k)} \end{bmatrix}.$$

Mit den linken, oberen Teilmatrizen $L_k^{(j)}$ von L und A_k von A gilt dann

$$A_k^{(k)} = L_k^{(k-1)} \dots L_k^{(1)} A_k.$$

Da die normalisierten Dreiecksmatrizen $L_k^{(j)}$ regulär sind, ist auch $A_k^{(k)}$ genau dann regulär, wenn A_k regulär ist. Dies ist nach dem Satz über die Existenz der LU-Zerlegung gegeben. Damit folgt $0 \neq \det A_k^{(k)} = a_{11}^{(1)}a_{22}^{(2)} \dots a_{kk}^{(k)}$ also $a_{kk}^{(k)} \neq 0$ und das Verfahren ist durchführbar.

(ii) Ist umgekehrt das Gauß-Verfahren durchführbar, so ist $U = A^{(n)} = L^{(n-1)} \dots L^{(1)}A$ eine obere Dreiecksmatrix und es genügt zu zeigen, dass $L = (L^{(n-1)} \dots L^{(1)})^{-1}$

eine normalisierte obere Dreiecksmatrix ist. Mit dem k -ten kanonischen Basisvektor $e_k \in \mathbb{R}^n$ und

$$\ell_k = [0, \dots, 0, \ell_{k+1,k}, \dots, \ell_{nk}]^\top$$

ist $L^{(k)} = I_n - \ell_k e_k^\top$. Mit $e_k^\top \ell_k = 0$ folgt

$$\begin{aligned} L^{(k)}(I_n + \ell_k e_k^\top) &= (I_n - \ell_k e_k^\top)(I_n + \ell_k e_k^\top) = I_n - \ell_k e_k^\top + \ell_k e_k^\top - \ell_k e_k^\top \ell_k e_k^\top \\ &= I_n, \end{aligned}$$

das heißt $(L^{(k)})^{-1} = I_n + \ell_k e_k^\top$. Mit vollständiger Induktion folgt

$$L = (L^{(n-1)} \dots L^{(1)})^{-1} = (L^{(1)})^{-1} \dots (L^{(n-1)})^{-1} = I_n + \sum_{j=1}^{n-1} \ell_j e_j^\top$$

beziehungsweise

$$L = \begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ \ell_{n1} & \dots & \ell_{n,n-1} & 1 \end{bmatrix}.$$

Dies zeigt, dass $A = LU$ die normalisierte LU-Zerlegung von A ist. \square

Bemerkung 4.1 Der Beweis zeigt, dass zur Bestimmung von L keine zusätzlichen Berechnungen erforderlich sind.

Für die Durchführung des Gauß-Verfahrens müssen die Matrizen $L^{(k)}$ nicht explizit aufgestellt werden.

Algorithmus 4.2 (Gauß-Verfahren) Sei $A \in \mathbb{R}^{n \times n}$ eine LU-zerlegbare Matrix und $b \in \mathbb{R}^n$. Berechne die LU-Zerlegung und den Vektor $y = U^{-1}b$ durch:

```

for k = 1 : n - 1
    for i = k + 1 : n;    $\ell_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ ;    $b_i^{(k+1)} = b_i^{(k)} - \ell_{ik} b_k^{(k)}$ ;   end;
        for j = k + 1 : n;    $a_{ij}^{(k+1)} = a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}$ ;   end;
    end

```

Bemerkung 4.2 Der Algorithmus liefert mit $(2/3)n^3 + \mathcal{O}(n^2)$ Rechenschritten die nicht-trivialen Einträge der LU-Zerlegung der Matrix A und die modifizierte rechte Seite y . Die Einträge von U sind gegeben durch $u_{ij} = a_{ij}^{(i)}$. Die Matrix A kann mit den berechneten Größen überschrieben werden.

4.2 Pivot-Strategie

Das oben definierte Gauß-Verfahren ist einerseits nicht für jede Matrix durchführbar und kann andererseits zu Instabilitäten führen.

Beispiel 4.2 Das lineare Gleichungssystem

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

ist für $0 \leq \varepsilon < 1/2$ gut konditioniert und besitzt die Lösung $x_1 = 1/(1 - \varepsilon)$ und $x_2 = (1 - 2\varepsilon)/(1 - \varepsilon)$. Der erste Schritt der Rückwärtssubstitution im Gauß-Verfahren liefert zunächst eine Approximation für x_2 , die für sehr kleine Zahlen ε unter Berücksichtigung von Rundung gegeben ist durch $\tilde{x}_2 = 1$. Wird dieses Ergebnis zur Berechnung von \tilde{x}_1 in der Gleichung $\varepsilon\tilde{x}_1 + \tilde{x}_2 = 1$ verwendet, so ergibt sich $\tilde{x}_1 = 0$, was keine gute Näherung von x_1 ist. Betrachtet man hingegen das äquivalente Gleichungssystem

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix},$$

so treten im Gauß-Verfahren keine Instabilitäten auf.

Die Vermeidung von Instabilitäten im Gauß-Verfahren erfolgt durch eine *Pivotsuche*. Dazu wird das obige Vorgehen in der k -Schleife vor der i -Schleife wie folgt erweitert:

- bestimme $p \in \{k, \dots, n\}$ mit $|a_{pk}^{(k)}| = \max_{i=k, \dots, n} |a_{ik}^{(k)}|$;
- vertausche die Zeilen p und k in $[A^{(k)}|b^{(k)}]$ und erhalte $[\tilde{A}^{(k)}|\tilde{b}^{(k)}]$;
- eliminiere Einträge in $[\tilde{A}^{(k)}|\tilde{b}^{(k)}]$ und erhalte $[A^{(k+1)}|b^{(k+1)}]$.

Praktisch wird das Vertauschen nicht tatsächlich durchgeführt, sondern entsprechende Indizes werden umbenannt, indem ein Vektor $\pi \in \mathbb{N}^n$ definiert wird, der die Vertauschungen beschreibt:

- initialisiere π mit $\pi = [1, \dots, n]$;
- sollen die Zeilen k und p vertauscht werden, so vertausche $\pi(k)$ und $\pi(p)$.

Die Vertauschung von Zeilen lässt sich auch mit einer *Permutationsmatrix* darstellen. Es gilt $\tilde{A}^{(k)} = P^{(k)}A^{(k)}$, wobei $P^{(k)}$ durch Vertauschen der Zeilen p und k in der Einheitsmatrix I_n entsteht.

Bemerkung 4.3 Statt der Spalten-Pivotsuche kann auch eine *totale Pivotsuche* durchgeführt werden, wobei dann auch Spalten vertauscht werden. Dies führt jedoch zu einem hohen Aufwand.

Satz 4.2 Ist $A \in \mathbb{R}^{n \times n}$ regulär und $b \in \mathbb{R}^n$, so ist das Gauß-Verfahren mit Pivotsuche durchführbar. Es liefert die normalisierte LU-Zerlegung $PA = LU$ mit $|\ell_{ij}| \leq 1$ für alle $1 \leq i, j \leq n$ sowie die modifizierte rechte Seite $b^{(n)} = L^{-1}Pb$. Dabei ist $P = P^{(n-1)} \dots P^{(1)}$.

Beweis Das Verfahren ist genau dann nicht durchführbar, wenn im k -ten Schritt mit $1 \leq k \leq n-1$ gilt, dass $|a_{pk}^{(k)}| = \max_{i=k, \dots, n} |a_{ik}^{(k)}| = 0$, das heißt die k -te Spalte der Matrix $A^{(k)}$ hat ab dem Diagonalelement nur verschwindende Einträge,

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & & \dots & a_{1,n}^{(1)} \\ \ddots & & & \vdots & \\ & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \dots & a_{k-1,n}^{(k-1)} \\ & 0 & a_{k,k+1}^{(k)} & \dots & a_{k,n}^{(k)} \\ \vdots & \vdots & \dots & \vdots & \\ 0 & a_{n,k+1}^{(k)} & \dots & a_{n,n}^{(k)} & \end{bmatrix},$$

damit sind die ersten k Spalten von k linear abhängig und folglich ist $A^{(k)}$ nicht regulär. Dann kann aber auch A nicht regulär sein, denn $A^{(k)}$ geht aus A durch reguläre Transformationen hervor. Dies ist ein Widerspruch und es folgt $\max_{i=k, \dots, n} |a_{ik}^{(k)}| > 0$. Für die Koeffizienten von L gilt $\ell_{ik} = a_{ik}^{(k)} / a_{pk}^{(k)}$ und nach Wahl von $a_{pk}^{(k)}$ folgt $|\ell_{ik}| \leq 1$. Zum Nachweis der Zerlegung $PA = LU$ bemerken wir mit $(P^{(k)})^{-1} = P^{(k)}$, dass

$$\begin{aligned} A^{(1)} &= A, \\ A^{(2)} &= L^{(1)}P^{(1)}A^{(1)} = L^{(1)}P^{(1)}A, \\ A^{(3)} &= L^{(2)}P^{(2)}A^{(2)} = L^{(2)}P^{(2)}L^{(1)}P^{(1)}A = L^{(2)}[P^{(2)}L^{(1)}P^{(2)}][P^{(2)}P^{(1)}]A, \\ A^{(4)} &= L^{(3)}P^{(3)}A^{(3)} = L^{(3)}[P^{(3)}L^{(2)}P^{(3)}][P^{(3)}P^{(2)}L^{(1)}P^{(2)}P^{(3)}][P^{(3)}P^{(2)}P^{(1)}]A \end{aligned}$$

und entsprechenden Identitäten für $A^{(5)}, \dots, A^{(n)}$. Mit

$$\hat{L}^{(k)} = P^{(n-1)}P^{(n-2)} \dots P^{(k+1)}L^{(k)}P^{(k+1)} \dots P^{(n-2)}P^{(n-1)}$$

gilt

$$A^{(n)} = \hat{L}^{(n-1)} \dots \hat{L}^{(1)} PA.$$

Die Matrix $A^{(n)} = U$ ist obere Dreiecksmatrix und mit $L^{(k)} = I_n - \ell_k e_k^\top$ folgt, dass

$$\hat{L}^{(k)} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\hat{\ell}_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & -\hat{\ell}_{nk} & & 1 \end{bmatrix}$$

und $L = (\hat{L}^{(n-1)} \dots \hat{L}^{(1)})^{-1}$ ist normalisierte untere Dreiecksmatrix. \square

Bemerkungen 4.4 (i) Zur Lösung von $Ax = b$ mit Hilfe einer LU -Zerlegung $PA = LU$ löst man die Gleichungssysteme $Ly = Pb$ und $Ux = y$. Im modifizierten Gauß-Verfahren gilt $y = b^{(n)} = L^{-1}Pb$ und man löst $Ux = b^{(n)}$.

(ii) In einer Implementation muss der Vektor π angelegt werden, um U und L aus der überschriebenen Matrix A zu gewinnen, das heißt es wird zusätzlicher Speicherplatz benötigt. Der Aufwand für das Gauß-Verfahren mit Pivotsuche beträgt ebenfalls $2n^3/3 + \mathcal{O}(n^2)$ Operationen.

4.3 Lernziele, Quiz und Anwendung

Sie sollten das Gaußsche Eliminationsverfahren motivieren und anwenden können sowie dessen Beziehungen zur LU -Zerlegung erklären können. Die Bedeutung von Pivot-Strategien sollten Sie veranschaulichen können.

Quiz 4.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Mit dem Gaußschen Eliminationsverfahren lässt sich die Inverse A^{-1} einer LU -zerlegbaren Matrix A mit dem Aufwand $\mathcal{O}(n^4)$ bestimmen	
Ist $A \in \mathbb{R}^{n \times n}$ positiv definit, so ist keine Pivotsuche notwendig, um das Gaußsche Eliminationsverfahren durchzuführen zu können	
Sind $L^{(1)}, L^{(2)}, \dots, L^{(n-1)}$ die Eliminationsmatrizen im Gaußschen Eliminationsverfahren für ein Gleichungssystem mit Systemmatrix A , so ist der Faktor L in der LU -Zerlegung von A gegeben durch $L = L^{(1)}L^{(2)}\dots L^{(n-1)}$	
Die Pivotsuche verhindert das Auftreten von Auslöschungseffekten	
Permutationsmatrizen erhält man durch Zeilenvertauschungen in der Einheitsmatrix	

Anwendung 4.1 Die Verbrennung von Traubenzucker wird durch die chemische Reaktionsgleichung



beschrieben. Dabei ist eine minimale, ganzzahlige Lösung $x = [x_1, x_2, \dots, x_4]^\top \neq 0$ zu bestimmen, sodass dieselbe Anzahl von Atomen jedes beteiligten Stoffs auf der linken und rechten Seite steht. Wie lässt sich das Gaußsche Eliminationsverfahren modifizieren, um eine Lösung zu konstruieren?

5.1 Gaußsche Normalengleichung

In vielen Anwendungen treten *überbestimmte Gleichungssysteme* auf, das heißt für $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $b \in \mathbb{R}^m$ ist $x \in \mathbb{R}^n$ gesucht, sodass

$$Ax \approx b.$$

Das Problem ist im Allgemeinen nicht exakt lösbar.

Beispiel 5.1 Zu Messdaten (t_i, y_i) , $i = 1, 2, \dots, m$, ist $c \in \mathbb{R}$ gesucht mit $y_i \approx ct_i$. Die Zahl c beschreibt dann die Steigung einer Geraden, die die Punktpaare möglichst gut approximiert, s. Abb. 5.1.

Definition 5.1 Durch $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ wird das *Ausgleichsproblem*

$$\text{Minimiere } x \mapsto \|Ax - b\|_2^2$$

definiert. Für $x \in \mathbb{R}^n$ heißt $r = Ax - b$ *Residuum* von x .

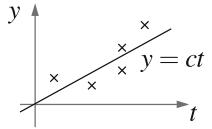
Die Betrachtung des Ausgleichsproblems wird aufgrund der verwendeten Euklidischen Norm auch als *Methode der kleinsten Quadrate* bezeichnet.

Satz 5.1 *Die Lösungen des Ausgleichsproblems sind genau die Lösungen der Gaußschen Normalengleichung*

$$A^\top Ax = A^\top b,$$

insbesondere existiert eine Lösung $x \in \mathbb{R}^n$. Ist $z \in \mathbb{R}^n$ eine weitere Lösung, so gilt $Ax = Az$ und die zugehörigen Residuen stimmen überein.

Abb. 5.1 In klassischen Ausgleichsproblemen ist eine Annäherung gegebener Messwerte durch eine Gerade gesucht



Beweis Nach Resultaten der linearen Algebra gilt

$$\mathbb{R}^m = \text{Im } A + \ker A^\top$$

und diese Zerlegung ist direkt und orthogonal. Ein Nachweis folgt aus Betrachtung der Menge $(\text{Im } A)^\perp$. Damit existieren zu $b \in \mathbb{R}^m$ eindeutig bestimmte Vektoren $y \in \text{Im } A$ und $r \in \ker A^\top$ mit $y \cdot r = 0$ und $b = y + r$. Ferner existiert ein $x \in \mathbb{R}^n$ mit $y = Ax$. Damit folgt

$$A^\top b = A^\top y + A^\top r = A^\top Ax + 0 = A^\top Ax,$$

das heißt x löst die Normalengleichung. Um zu zeigen, dass x auch Lösung des Ausgleichsproblems ist, sei $z \in \mathbb{R}^n$ beliebig. Mit $r = b - Ax$ und $A^\top r = 0$ folgt

$$\begin{aligned} \|b - Az\|_2^2 &= \|(b - Ax) + A(x - z)\|_2^2 \\ &= \|b - Ax\|_2^2 + 2r \cdot A(x - z) + \|A(x - z)\|_2^2 \\ &= \|b - Ax\|_2^2 + 2(A^\top r) \cdot (x - z) + \|A(x - z)\|_2^2 \\ &= \|b - Ax\|_2^2 + \|A(x - z)\|_2^2 \\ &\geq \|b - Ax\|_2^2. \end{aligned}$$

Also ist x eine Minimalstelle und somit Lösung des Ausgleichsproblems. Gleichheit gilt genau dann, wenn $A(x - z) = 0$ also $x - z \in \ker A = \ker A^\top A$ gilt, also genau dann, wenn auch z die Normalengleichung erfüllt. Insbesondere folgt, dass $A(x - z) = 0$ gilt, wenn $z \in \mathbb{R}^n$ eine weitere Lösung ist. \square

Bemerkung 5.1 Die Identität $A^\top r = 0$ besagt, dass r senkrecht oder normal zu den Spalten von A ist.

Lemma 5.1 Die Matrix $A^\top A$ ist symmetrisch und positiv semidefinit. Sie ist positiv definit genau dann, wenn $\ker A = \{0\}$ gilt, das heißt wenn A injektiv ist beziehungsweise die Spaltenvektoren von A linear unabhängig sind, also $\text{rank } A = n$ falls $m \geq n$. In diesem Fall ist die Lösung der Normalengleichung eindeutig.

Beweis Es gilt $(A^\top A)^\top = A^\top A$ und

$$x^\top (A^\top A)x = (Ax)^\top (Ax) = \|Ax\|_2^2 \geq 0$$

mit Gleichheit genau dann, wenn $Ax = 0$ gilt. Dies impliziert die Behauptung, da positiv definite Matrizen regulär sind. \square

Bemerkung 5.2 Die Konditionszahl von $A^\top A$ ist im Allgemeinen größer als die von A , denn für $m = n$ und eine reguläre Matrix $A \in \mathbb{R}^{n \times n}$ gilt

$$\text{cond}_2(A^\top A) = \|A^\top A\|_2 \|(A^\top A)^{-1}\|_2 = \frac{\lambda_{\max}(A^\top A)}{\lambda_{\min}(A^\top A)} = \text{cond}_2(A)^2,$$

also $\text{cond}_2(A^\top A) \geq \text{cond}_2(A)$, da $\text{cond}_2(A) \geq 1$.

Aufgrund dieser Beobachtung werden Ausgleichsprobleme nicht mit Hilfe der Normalgleichung gelöst.

5.2 Householder-Transformationen

Da die Euklidische Norm invariant ist unter Rotationen, gilt

$$\|\tilde{Q}(Ax - b)\|_2 = \|Ax - b\|_2$$

für jede Rotation \tilde{Q} und allgemeiner für orthogonale Matrizen. Wir werden versuchen, eine orthogonale Matrix Q zu konstruieren, sodass QA eine verallgemeinerte obere Dreiecksgestalt hat, was eine einfache Lösung des Ausgleichsproblems ermöglicht.

Definition 5.2 Die Matrix $Q \in \mathbb{R}^{\ell \times \ell}$ heißt *orthogonal*, falls $Q^\top Q = I_\ell$ gilt. Die Menge der orthogonalen Matrizen wird mit $O(\ell)$ bezeichnet.

Lemma 5.2 Für alle $P, Q \in O(\ell)$ gilt $PQ \in O(\ell)$, $Q^{-1} = Q^\top \in O(\ell)$, $\|Qx\|_2 = \|x\|_2$ für alle $x \in \mathbb{R}^\ell$ sowie $\text{cond}_2(Q) = 1$.

Beweis Für alle $x \in \mathbb{R}^\ell$ gilt

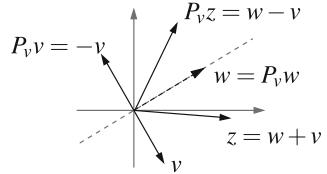
$$\|Qx\|_2^2 = (Qx)^\top (Qx) = x^\top (Q^\top Q)x = x^\top x = \|x\|_2^2$$

und damit $\|Q\|_2 = 1$. Aus den Identitäten

$$\begin{aligned} QQ^\top &= (QQ^\top)^\top = I_\ell^\top = I_\ell, \\ Q^{-\top}Q^{-1} &= (QQ^\top)^{-1} = (Q^\top Q)^{-\top} = I_\ell^{-\top} = I_\ell, \\ (PQ)^\top(PQ) &= Q^\top(P^\top P)Q = Q^\top Q = I_\ell \end{aligned}$$

folgt $PQ \in O(\ell)$ und $Q^{-1} = Q^\top \in O(\ell)$. Diese Eigenschaften implizieren $\text{cond}_2(Q) = \|Q\|_2\|Q^{-1}\|_2 = 1$. \square

Abb. 5.2 Householder-Transformationen definieren Spiegelungen an einer Ebene



Definition 5.3 Für $v \in \mathbb{R}^\ell$ mit $\|v\|_2 = 1$ heißt die Matrix $P_v = I_\ell - 2vv^\top$ *Householder-Transformation*.

Householder-Transformationen realisieren Spiegelungen an der zu v senkrechten Ebene, s. Abb. 5.2.

Lemma 5.3 Jede Householder-Transformation $P_v = I_\ell - 2vv^\top$ ist symmetrisch und orthogonal. Es gilt $P_v v = -v$ und $P_v w = w$ für alle $w \in \mathbb{R}^\ell$ mit $w \cdot v = 0$.

Beweis Übungsaufgabe. □

Jeder Vektor $x \in \mathbb{R}^\ell \setminus \{0\}$ lässt sich mit einer Householder-Transformation in \mathbb{R}^ℓ auf ein Vielfaches des kanonischen Basisvektors $e_1 \in \mathbb{R}^\ell$ abbilden.

Lemma 5.4 Sei $x \in \mathbb{R}^\ell \setminus \{0\}$ und $x \notin \text{span}\{e_1\}$ und definiere mit $\sigma = \text{sign}(x_1)$ falls $x_1 \neq 0$ und $\sigma = 1$ sonst

$$v = \frac{x + \sigma \|x\|_2 e_1}{\|x + \sigma \|x\|_2 e_1\|_2}.$$

Dann gilt

$$P_v x = (I_\ell - 2vv^\top)x = -\sigma \|x\|_2 e_1.$$

Beweis Da $x \notin \text{span}\{e_1\}$ ist v wohldefiniert und es gilt $\|v\|_2 = 1$. Weiter ist

$$\|x + \sigma \|x\|_2 e_1\|_2^2 = \|x\|_2^2 + 2\sigma \|x\|_2 x \cdot e_1 + \sigma^2 \|x\|_2^2 \|e_1\|_2^2 = 2(x + \sigma \|x\|_2 e_1)^\top x.$$

Mit $\tilde{v} = x + \sigma \|x\|_2 e_1$ gilt

$$2\tilde{v}^\top x = 2(x + \sigma \|x\|_2 e_1)^\top x = \|x + \sigma \|x\|_2 e_1\|_2^2 = \|\tilde{v}\|_2^2$$

und somit wegen $v = \tilde{v}/\|\tilde{v}\|_2$

$$P_v x = (I_\ell - 2vv^\top)x = x - 2v \frac{\tilde{v}^\top x}{\|\tilde{v}\|_2} = x - v\|\tilde{v}\|_2 = -\sigma \|x\|_2 e_1.$$

Dies beweist die Behauptung. □

Bemerkung 5.3 Die Einführung von σ vermeidet Auslöschungseffekte.

5.3 QR-Zerlegung

Mit Hilfe von Householder-Transformationen werden wir schrittweise die ersten Spalten von Teilmatrizen von A auf Vielfache kanonischer Basisvektoren e_1 entsprechender Länge transformieren und somit eine obere Dreiecksstruktur erzeugen.

Satz 5.2 Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{rank } A = n$. Dann existieren $Q \in O(m)$ und eine verallgemeinerte obere Dreiecksmatrix $R \in \mathbb{R}^{m \times n}$, das heißt es gilt $r_{ij} = 0$ für $i > j$, sodass

$$A = QR = Q \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{22} & \dots & & r_{2n} \\ \ddots & & \vdots & \\ & & & r_{nn} \end{bmatrix}.$$

Ferner gilt $|r_{ii}| > 0$ für alle $1 \leq i \leq n$. Die Faktorisierung heißt QR-Zerlegung.

Beweis Im ersten Schritt setzen wir $A_1 = A$. Es bezeichne $x = a_1 \in \mathbb{R}^m$ die erste Spalte von A_1 . Ist x ein Vielfaches von e_1 , so setzen wir $Q_1 = I_m$. Andernfalls definieren wir $Q_1 = P_v$ wie im vorigen Lemma. Es folgt $Q_1 a_1 = r_{11} e_1$ mit $|r_{11}| = \|Q_1 a_1\|_2 = \|a_1\|_2 > 0$ und somit

$$Q_1 A_1 = \begin{bmatrix} r_{11} & r_1^\top \\ & A_2 \end{bmatrix}$$

mit einer Matrix $A_2 \in \mathbb{R}^{(m-1) \times (n-1)}$ und einem Vektor $r_1 \in \mathbb{R}^{n-1}$. Im zweiten Schritt sei $a_2 \in \mathbb{R}^{m-1}$ die erste Spalte von A_2 und $\tilde{Q}_2 \in \mathbb{R}^{(m-1) \times (m-1)} \in O(m-1)$ die Einheitsmatrix I_{m-1} oder eine Householder-Transformation $\tilde{Q}_2 = P_{\tilde{v}}$, sodass $\tilde{Q}_2 a_2 = r_{22} e_1 \in \mathbb{R}^{m-1}$ mit $|r_{22}| = \|\tilde{Q}_2 a_2\| = \|a_2\|_2 > 0$. Damit gilt

$$\tilde{Q}_2 A_2 = \begin{bmatrix} r_{22} & r_2^\top \\ & A_3 \end{bmatrix}$$

mit einer Matrix $A_3 \in \mathbb{R}^{(m-2) \times (n-2)}$ und einem Vektor $r_2 \in \mathbb{R}^{n-2}$. Wir können daher schreiben

$$Q_2 Q_1 A = \begin{bmatrix} 1 & \\ & \tilde{Q}_2 \end{bmatrix} Q_1 A = \begin{bmatrix} r_{11} & r_1^\top \\ & \tilde{Q}_2 A_2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_1^\top & \\ & r_{22} & r_2^\top \\ & & A_3 \end{bmatrix}.$$

Die ersten beiden Zeilen bleiben in den nachfolgenden Schritten unverändert. Die Matrix Q_2 ist orthogonal, insbesondere ist sie die Householder-Transformation zum Vektor $v = [0, \tilde{v}]^\top$, wobei $\tilde{v} = 0$ im Fall $\tilde{Q}_2 = I_{m-1}$ sei. Nach n Schritten erhalten wir die Faktorisierung

$$Q_n Q_{n-1} \dots Q_1 A = R.$$

Da jede Householder-Transformation orthogonal und symmetrisch ist, gilt $Q_j^{-1} = Q_j^\top = Q_j$ für $j = 1, 2, \dots, n$. Damit ergibt sich

$$(Q_n Q_{n-1} \dots Q_1)^{-1} = Q_1^{-1} Q_2^{-1} \dots Q_n^{-1} = Q_1^\top Q_2^\top \dots Q_n^\top = Q_1 Q_2 \dots Q_n$$

und mit $Q = Q_1 Q_2 \dots Q_n$ folgt die behauptete Faktorisierung $A = QR$. Die Einträge r_{ii} , $i = 1, 2, \dots, n$, von R erfüllen $r_{ii} = \|a_i\|_2 \neq 0$, da A sonst nicht vollen Rang hätte. \square

Bemerkungen 5.4 (i) Im Fall $m = n$ ist die Faktorisierung bis auf Vorzeichen der Diagonaleinträge von R eindeutig bestimmt, denn gilt $A = QR = Q'R'$, so folgt, dass $E = (Q')^{-1}Q = R'R^{-1}$ eine obere Dreiecksmatrix in $O(n)$ ist. Da E^{-1} eine obere Dreiecksmatrix ist, kann die Identität $E^\top = E^{-1}$ jedoch nur dann gelten, wenn E eine Diagonalmatrix mit Diagonalelementen in $\{\pm 1\}$ ist und damit folgt $Q = Q'E$ sowie $R = ER'$.

(ii) Die Householder-Transformationen werden nicht über Matrix-Matrix-Multiplikationen realisiert, denn es gilt mit $w = A^\top v$

$$P_v A = (I_m - 2vv^\top)A = A - 2v(v^\top A) = A - 2vw^\top.$$

(iii) Die Vektoren v_i , $i = 1, 2, \dots, n$, die die Householder-Transformationen definieren, lassen sich an den frei werdenden Stellen von A abspeichern, wobei $v_i = 0$ sei, falls $Q_i = I_m$ ist. Es gilt ferner

$$Q = \prod_{i=1}^n (I_m - 2v_i v_i^\top).$$

Algorithmus 5.1 (QR-Zerlegung) Sei $A \in \mathbb{R}^{m \times n}$ mit $\text{rank } A = n$. Initialisiere $A_1 = A$ und $i = 1$.

- (1) Sei $a_i \in \mathbb{R}^{m-i+1}$ die erste Spalte des rechten unteren Blocks $A_i \in \mathbb{R}^{(m-i+1) \times (n-i+1)}$ von A .
- (2) Gilt $a_i = e_1$, so fahre fort mit (5).
- (3) Definiere $\tilde{v} = a_i + \sigma \|a_i\|_2 e_1$ und $v = \tilde{v}/\|\tilde{v}\|_2$.
- (4) Ersetze den Block A_i durch $A_i - vw^\top$ mit $w = 2A_i^\top v$.
- (5) Stoppe falls $i = n$; andernfalls erhöhe $i \rightarrow i + 1$ und wiederhole Schritt (1).

Bemerkung 5.5 Im i -ten Iterationsschritt werden

- $4(m - i + 1) + 4$ Operationen zur Berechnung von v ,
- $(m - i + 2)(n - i + 1)$ Operationen zur Berechnung von w ,
- $(m - i)(n - i + 1)$ Operationen zur Berechnung von $A_i - vw^\top$

benötigt. Insgesamt ist der Aufwand zur Berechnung der Faktorisierung damit $2mn^2 - (2/3)n^3 + \mathcal{O}(mn)$. Im Fall $m = n$ ist die Berechnung also doppelt so teuer wie die der LU -Zerlegung.

5.4 Lösung des Ausgleichsproblems

Wir verwenden die QR -Zerlegung, um ein stabiles Verfahren für das Ausgleichsproblem zu konstruieren.

Satz 5.3 Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{rank } A = n$. Mit der QR -Zerlegung $A = QR$ sowie

$$Q^\top b = \begin{bmatrix} c \\ d \end{bmatrix}, \quad Q^\top A = R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}$$

mit $c \in \mathbb{R}^n$, $d \in \mathbb{R}^{m-n}$ und einer oberen Dreiecksmatrix $\hat{R} \in \mathbb{R}^{n \times n}$ ist die Lösung des durch A und b definierten Ausgleichsproblems gegeben durch $\hat{R}x = c$.

Beweis Mit $\|Qz\|_2 = \|z\|_2$ für alle $z \in \mathbb{R}^m$ und $Q^\top Q = I_m$ gilt

$$\|b - Ax\|_2^2 = \|Q(Q^\top b - Q^\top Ax)\|_2^2 = \left\| \begin{bmatrix} c \\ d \end{bmatrix} - \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}x \right\|_2^2 = \|\hat{R}x - c\|_2^2 + \|d\|_2^2.$$

Da $\text{rank } A = n$ gilt, ist \hat{R} regulär. Die rechte Seite wird offensichtlich für $x = \hat{R}^{-1}c$ minimal. \square

Bemerkung 5.6 Aus $Q \in O(n)$ folgt für reguläre Matrizen $A \in \mathbb{R}^{n \times n}$, dass $\text{cond}_2(R) = \text{cond}_2(A)$. Die QR -Zerlegung definiert damit einen stabilen Algorithmus.

5.5 Lernziele, Quiz und Anwendung

Ihnen sollten Anwendungen bekannt sein, die auf Ausgleichsprobleme führen, und Sie sollten die Gaußsche Normalengleichung herleiten und ihre wichtigsten Eigenschaften aufzeigen können. Die Konstruktion der QR -Zerlegung einer Matrix sollten Sie erklären und deren Bedeutung bei der Lösung von Ausgleichsproblemen beschreiben können.

Quiz 5.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Das Ausgleichsproblem besitzt stets eine Lösung	
Gilt $\text{rank } A = n \leq m$, so ist $A^\top A$ invertierbar	
Durch $I_n - 2(v^\top v)^{-2}vv^\top$ wird eine Householder-Transformation definiert, sofern $v \in \mathbb{R}^n \setminus \{0\}$ gilt	
Ist Q orthogonal, so sind sowohl die Zeilen- als auch die Spaltenvektoren von Q paarweise orthogonal	
Für jede Vektornorm $\ \cdot\ $ auf \mathbb{R}^n , jede orthogonale Matrix $Q \in O(n)$ und jeden Vektor $x \in \mathbb{R}^n$ gilt $\ Qx\ = \ x\ $	

Anwendung 5.1 Theoretische Überlegungen zu zwei physikalischen Vorgängen führen zu der Annahme, dass die Größen y und t einem Zusammenhang der Form $y(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$ und die Größen z und v dem Zusammenhang $z(v) = c/v$ genügen. Mit Experimenten werden Messdaten (t_i, y_i) beziehungsweise (v_i, z_i) für $i = 1, 2, \dots, m$ bestimmt. Formulieren Sie Ausgleichsprobleme zur näherungsweisen Bestimmung von c_0, c_1, \dots, c_3 beziehungweise c und stellen Sie die zugehörigen Gaußschen Normalengleichungen auf. Wie lässt sich die Gültigkeit der Vermutung über den Zusammenhang nach Berechnung der Koeffizienten beurteilen?

6.1 Singulärwertzerlegung

Die symmetrische und positiv semidefinite Matrix $A^\top A \in \mathbb{R}^{n \times n}$ für $A \in \mathbb{R}^{m \times n}$ spielt eine wichtige Rolle im Ausgleichsproblem. Sie ist diagonalisierbar und es existiert eine Orthonormalbasis bestehend aus Eigenvektoren v_1, v_2, \dots, v_n mit zugehörigen Eigenwerten

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p > \lambda_{p+1} = \dots = \lambda_n = 0$$

mit $0 \leq p \leq n$, wobei Eigenwerte gemäß ihrer Vielfachheit gegebenenfalls mehrfach aufgezählt werden. Für $i = 1, 2, \dots, p$ definieren wir $u_i = \lambda_i^{-1/2} A v_i$. Für $1 \leq i, j \leq p$ gilt dann

$$\begin{aligned} u_i^\top u_j &= \lambda_i^{-1/2} \lambda_j^{-1/2} (A v_i)^\top (A v_j) = (\lambda_i \lambda_j)^{-1/2} v_i^\top (A^\top A v_j) \\ &= (\lambda_i \lambda_j)^{-1/2} v_i^\top (\lambda_j v_j) = \lambda_j (\lambda_i \lambda_j)^{-1/2} v_i^\top v_j = \delta_{ij}. \end{aligned}$$

Die Vektoren (u_1, u_2, \dots, u_p) bilden also ein Orthonormalsystem im \mathbb{R}^m , genauer eine Orthonormalbasis des Unterraums $\text{Im } A$. Wir ergänzen es durch Vektoren $(u_{p+1}, u_{p+2}, \dots, u_m)$ zu einer Orthonormalbasis (u_1, u_2, \dots, u_m) des \mathbb{R}^m . Es gilt

$$A^\top u_i = \lambda_i^{-1/2} A^\top A v_i = \lambda_i^{1/2} v_i, \quad i = 1, 2, \dots, p.$$

Aus $\ker A^\top A = \ker A$ folgt $\text{Im } A = \text{span}\{u_1, u_2, \dots, u_p\}$ und somit die Inklusion $\{u_{p+1}, \dots, u_m\} \subset (\text{Im } A)^\perp = \ker A^\top$ beziehungsweise

$$A^\top u_i = 0, \quad i = p + 1, \dots, m.$$

Unter Verwendung von $\sigma_i = \lambda_i^{1/2}$, $i = 1, 2, \dots, p$, erhalten wir folgenden Satz.

Satz 6.1 Sei $A \in \mathbb{R}^{m \times n}$. Dann existieren Zahlen $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$ und Orthonormalbasen $(u_i)_{i=1,\dots,m}$ des \mathbb{R}^m und $(v_j)_{j=1,\dots,n}$ des \mathbb{R}^n mit den Eigenschaften

$$\begin{aligned} Av_i &= \sigma_i u_i, & A^\top u_i &= \sigma_i v_i, & i &= 1, 2, \dots, p, \\ Av_j &= 0, & A^\top u_k &= 0, & j &= p+1, \dots, n, k = p+1, \dots, m. \end{aligned}$$

Die Zahlen σ_i^2 , $i = 1, 2, \dots, p$, sind genau die von Null verschiedenen Eigenwerte von $A^\top A$. Für

$$U = [u_1, \dots, u_m] \in \mathbb{R}^{m \times m}, \quad V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$$

gilt $U \in O(m)$ und $V \in O(n)$ und mit

$$\Sigma = \begin{bmatrix} \sigma_1 & & 0 & \dots & 0 \\ & \ddots & \vdots & & \vdots \\ & & \sigma_p & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$$

folgt

$$A = U \Sigma V^\top = \sum_{i=1}^p \sigma_i u_i v_i^\top, \quad A^\top = V \Sigma^\top U^\top = \sum_{i=1}^p \sigma_i v_i u_i^\top.$$

Die Faktorisierung heißt Singulärwertzerlegung und im Englischen singular value decomposition (SVD).

Beweis Die Aussagen folgen unmittelbar aus der Konstruktion. □

6.2 Pseudoinverse

Mit Hilfe der Singulärwertzerlegung lässt sich der Begriff der inversen Matrix auf nicht-reguläre und nichtquadratische Matrizen verallgemeinern.

Definition 6.1 Ist $A = U \Sigma V^\top$ die Singulärwertzerlegung von A und $\Sigma^+ \in \mathbb{R}^{n \times m}$ definiert durch

$$\Sigma^+ = \begin{bmatrix} \sigma_1^{-1} & & 0 \\ & \ddots & \vdots \\ & & \sigma_p^{-1} & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times m},$$

so heißt $A^+ = V \Sigma^+ U^\top = \sum_{i=1}^p \sigma_i^{-1} v_i u_i^\top \in \mathbb{R}^{n \times m}$ *Pseudoinverse* oder *Moore–Penrose-Inverse* von A .

Bemerkungen 6.1 (i) Nach Wahl der Vektoren u_i, v_j gilt $\ker A^+ = \ker A^\top$ und $\text{Im } A^+ = \text{Im } A^\top$.

(ii) Die Matrix A^+ ist die eindeutig bestimmte Lösung in $\mathbb{R}^{n \times m}$ der algebraischen Gleichungen

$$AXA = A, \quad XAX = X, \quad (AX)^\top = AX, \quad (XA)^\top = XA.$$

Beispielsweise gilt wegen $U^\top U = I_m$ und $V^\top V = I_n$, dass

$$A^+ AA^+ = (V \Sigma^+ U^\top)(U \Sigma V^\top)(V \Sigma^+ U^\top) = V \Sigma^+ \Sigma \Sigma^+ U^\top = V \Sigma^+ U^\top = A^+.$$

Mit der Pseudoinversen lässt sich das Ausgleichsproblem lösen.

Satz 6.2 *Der Vektor $A^+ b$ ist unter allen Lösungen des Ausgleichsproblems diejenige mit minimaler Norm.*

Beweis Mit $A^+ AA^+ = A^+$ und $\ker A^+ = (\text{Im } A)^\perp$ folgt

$$AA^+ b - b \in \ker A^+ = (\text{Im } A)^\perp = \ker A^\top,$$

das heißt $A^\top A(A^+ b) = A^\top b$ beziehungsweise, dass $A^+ b$ Lösung der Gaußschen Normalengleichung ist. Ist $z \in \mathbb{R}^n$ eine weitere Lösung, so gilt wegen $\ker A^\top A = \ker A$, dass

$$A^\top A(A^+ b - z) = 0 \iff A(A^+ b - z) = 0.$$

Mit $w = A^+ b - z \in \ker A$ folgt aus $A^+ b \in \text{Im } A^+ = (\ker A)^\perp$, dass $(A^+ b) \cdot w = 0$ und für $z = A^+ b - w$ erhalten wir

$$\|z\|_2^2 = \|A^+ b\|_2^2 + \|w\|_2^2 \geq \|A^+ b\|_2^2.$$

Damit ist $A^+ b$ eine Lösung mit minimaler Norm. □

Bemerkung 6.2 Gilt $\text{rank } A = n \leq m$, so folgt aus $A^+ b = (A^\top A)^{-1} A^\top b$ für alle $b \in \mathbb{R}^m$, dass $A^+ = (A^\top A)^{-1} A^\top$ und insbesondere $A^+ = A^{-1}$ falls $n = m$.

6.3 Lernziele, Quiz und Anwendung

Sie sollten die Ideen zur Konstruktion der Singulärwertzerlegung einer Matrix beschreiben und die Definition der Pseudoinversen sowie deren Bezug zu Ausgleichsproblemen konkretisieren können.

Quiz 6.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Die Quadrate der Singulärwerte einer Matrix sind die Eigenwerte von AA^\top	
Für den ersten Singulärwert σ_1 von A gilt $\sigma_1 = \ A\ _2$	
Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch, so definiert die Singulärwertzerlegung eine Diagonalisierung von A	
Eine Lösung des Ausgleichsproblems wird definiert durch die Lösung des linearen Gleichungssystems $A^+x = b$	
Es existiert eine Lösung $z \in \mathbb{R}^n$ des Ausgleichsproblems mit der Eigenschaft $\ A^+b\ _2 = \ z\ _2$	

Anwendung 6.1 Die Matrix $A \in \mathbb{R}^{m \times n}$ beschreibe bestimmte Daten wie beispielsweise die Graustufen eines Bildes. Um die Daten zu komprimieren, wird zunächst die Singulärwertzerlegung $A = \sum_{i=1}^p \sigma_i u_i v_i^\top$ bestimmt. Für $\varepsilon > 0$ und $i = 1, 2, \dots, p$ sei

$$\tilde{\sigma}_i = \begin{cases} \sigma_i, & \text{falls } \sigma_i \geq \varepsilon, \\ 0, & \text{falls } \sigma_i < \varepsilon. \end{cases}$$

Es sei

$$\tilde{A} = \sum_{i=1}^p \tilde{\sigma}_i u_i v_i^\top.$$

Zeigen Sie, dass

$$\|A - \tilde{A}\|_{\mathcal{F}} \leq p\varepsilon$$

und $\operatorname{rank} \tilde{A} \leq \operatorname{rank} A$.

7.1 Lineare Programme

In Anwendungen wie der Minimierung von Produktionskosten treten lineare Optimierungsprobleme mit linearen Ungleichungs-Nebenbedingungen auf. Um solche Probleme prägnant zu formulieren, verwenden wir im Folgenden die Schreibweise $a \leq b$ für Vektoren $a, b \in \mathbb{R}^m$, falls $a_i \leq b_i$ für $i = 1, 2, \dots, m$ gilt. Wir folgen in diesem Kapitel der Darstellung in [10].

Definition 7.1 Ein *lineares Programm* ist eine Optimierungsaufgabe

$$\text{Minimiere } g(y) = p^\top y \text{ unter der Nebenbedingung } Uy \leq d$$

mit gegebenen $p \in \mathbb{R}^\ell$, $U \in \mathbb{R}^{q \times \ell}$ und $d \in \mathbb{R}^q$. Ein lineares Programm ist in *Normalform*, wenn es sich in der Form

$$\text{Minimiere } f(x) = c^\top x \text{ unter der Nebenbedingung } Ax = b, x \geq 0$$

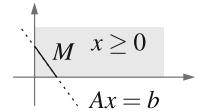
mit gegebenen $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ schreiben lässt.

Bemerkung 7.1 Durch Einführung zusätzlicher Variablen lässt sich jedes lineare Programm in Normalform überführen. Dabei zerlegt man $y_i = v_i - w_i$ mit $v_i, w_i \geq 0$ und schreibt eine Ungleichung $Uy \leq d$ als Gleichung $Uy + z = d$ mit $z \geq 0$. Die neue Variable ist dann der Vektor $x = [v, w, z]$.

Definition 7.2 Die *zulässige Menge* eines linearen Programms in Normalform ist $M = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$.

Bemerkung 7.2 Die zulässige Menge ist konvex und kann leer, einelementig, beschränkt oder unbeschränkt sein, s. Abb. 7.1.

Abb. 7.1 Die zulässige Menge M eines linearen Programms ist der Schnitt konvexer Mengen



Definition 7.3 Ein Punkt $x \in M$ heißt *Ecke*, wenn er sich nicht als echte Konvexitätskombination in M schreiben lässt, das heißt für alle $z, y \in M$ und $\lambda \in (0, 1)$ mit $x = \lambda z + (1 - \lambda)y$ folgt $x = y = z$.

Wir werden im Folgenden stets annehmen, dass M nicht leer ist. Ohne Beweis verwenden wir die folgenden Resultate.

Satz 7.1 Die zulässige Menge M sei nicht leer und beschränkt.

- (i) Die Menge M hat endlich viele Ecken $y^1, y^2, \dots, y^L \in M$ und diese spannen M auf, das heißt $M = \{x = \sum_{i=1}^L \lambda_i y^i : \lambda_i \in [0, 1], \sum_{i=1}^L \lambda_i = 1\}$.
- (ii) Das lineare Programm besitzt eine Lösung und das Minimum wird in einer Ecke von M angenommen.

Bemerkung 7.3 Ist M unbeschränkt, so kann das Problem lösbar oder unlösbar sein.

Um ein lineares Programm zu lösen, genügt es also Ecken zu betrachten. Die Menge der möglichen Lösungen wird damit auf endlich viele Punkte reduziert.

Definition 7.4 Die Indexmenge I_x einer Ecke $x \in M$ besteht aus den Indizes der von Null verschiedenen Komponenten

$$I_x = \{i \in \{1, 2, \dots, n\} : x_i > 0\}$$

und es sei $J_x = \{1, 2, \dots, n\} \setminus I_x$. Die Mengen I_x und J_x werden als geordnet angesehen und für einen Vektor $z \in \mathbb{R}^n$ sowie die Matrix $A \in \mathbb{R}^{m \times n}$ mit Spaltenvektoren $(a_i : i = 1, 2, \dots, n)$ schreiben wir

$$\begin{aligned} z_{I_x} &= (z_i : i \in I_x), & z_{J_x} &= (z_j : j \in J_x), \\ A_{I_x} &= (a_i : i \in I_x), & A_{J_x} &= (a_j : j \in J_x). \end{aligned}$$

Wenn aus dem Kontext klar ist, welche Ecke gemeint ist, wird der Index x bei I_x und J_x weggelassen. Für $z \in \mathbb{R}^n$ gilt dann

$$Az = A_I z_I + A_J z_J.$$

Satz 7.2 Für die Ecken von M gelten die folgenden Aussagen.

- (i) Ein Punkt $x \in M$ ist Ecke genau dann, wenn die Spaltenvektoren $(a_i : i \in I_x)$ linear unabhängig sind.
- (ii) Jede Ecke $x \in M$ ist durch ihre Indexmenge eindeutig festgelegt.

Beweis

- (i) Ist x keine Ecke, so existieren $y, z \in M \setminus \{x\}$ und $\lambda \in (0, 1)$ mit $x = \lambda y + (1 - \lambda)z$. Es gilt $I_{y-z} \subset I_x = I$, da aus $x_i = 0$ und $y, z \geq 0$ auch $y_i = z_i = 0$ folgt. Ferner gilt $0 = b - b = A(y - z) = A_I(y - z)_I$ und wegen $(y - z)_I \neq 0$ folgt die lineare Abhängigkeit der Spalten der Matrix A_I . Sind umgekehrt die Spalten von A_I linear abhängig, so existiert $\tilde{y} \neq 0$ mit $A_I \tilde{y} = 0$ und \tilde{y} lässt sich durch Nullen zu $y \in \mathbb{R}^n$ mit $y_I = \tilde{y}$ und $I_y \subset I_x$ ergänzen. Da $x_i > 0$ für alle $i \in I_x$ gilt dann mit $\varepsilon > 0$ hinreichend klein, dass $x \pm \varepsilon y \geq 0$. Ferner gilt $Ay = 0$ und somit $A(x \pm \varepsilon y) = b$, also $x \pm \varepsilon y \in M$. Mit $\lambda = 1/2$ ist dann aber $x = \lambda(x + \varepsilon y) + (1 - \lambda)(x - \varepsilon y)$ eine echte Konvexitätskombination und x keine Ecke.
- (ii) Ist x eine Ecke so sind nach (i) die Spaltenvektoren von A_I linear unabhängig und aus $b = Ax = A_I x_I$ folgt, dass x_I eindeutig bestimmt ist. \square

Die Anzahl der durch $A \in \mathbb{R}^{m \times n}$ definierten Gleichheits-Nebenbedingungen $Ax = b$ in einem linearen Programm in Normalform ist in der Regel geringer als die Anzahl der Unbekannten, das heißt es gilt $m \leq n$.

Definition 7.5 Eine Ecke $x \in M$ heißt *entartet*, wenn $|I_x| < m$ gilt. Andernfalls heißt sie *nichtentartet*.

Bemerkung 7.4 Ist $x \in M$ eine nichtentartete Ecke und gilt $\text{rank } A = m$, so ist $|I_x| = m$ und die Matrix $A_I \in \mathbb{R}^{m \times m}$ ist invertierbar.

7.2 Der Simplex-Schritt

Zur Lösung eines linearen Programms ist es ausreichend, die Ecken der zulässigen Menge zu betrachten. Ausgehend von einer Ecke wird dabei eine neue konstruiert, sodass der Funktionswert verkleinert wird. Es gelte $\text{rank } A = m$ und M sei nichtleer. Wir gehen wie folgt vor:

- (1) Es sei $x \in M$ eine Ecke und falls diese entartet ist, sei I_x so zu einer m -elementigen Menge I ergänzt, dass A_I regulär ist. Es sei $J = \{1, 2, \dots, n\} \setminus I$.
- (2) Für alle $z \in M$ folgt aus $b = Az = A_I z_I + A_J z_J$, dass

$$z_I = A_I^{-1}b - A_I^{-1}A_J z_J. \quad (7.1)$$

Damit sind die Komponenten bezüglich I eindeutig durch die bezüglich J festgelegt; insbesondere gilt wegen $x_J = 0$, dass $x_I = A_I^{-1}b$. Für den Funktionswert $f(z) = c^\top z$ folgt

$$\begin{aligned} c^\top z &= c_I^\top z_I + c_J^\top z_J = c_I^\top (A_I^{-1}b - A_I^{-1}A_J z_J) + c_J^\top z_J \\ &= c_I^\top x_I + (c_J^\top - c_I^\top A_I^{-1}A_J)z_J = c^\top x + (c_J - A_J^\top A_I^{-1}c_I)^\top z_J. \end{aligned}$$

Mit $u_J = c_J - A_J^\top A_I^{-1}c_I$ gilt also

$$f(z) = f(x) + u_J^\top z_J. \quad (7.2)$$

Eine Verringerung des Funktionswerts ist wegen $z \geq 0$ also nur dann möglich, wenn u_J in einer Komponente negativ ist. Andernfalls ist \underline{x} bereits Lösung des Problems.

(3) Es gelte $u_r < 0$ für ein $r \in J$. Wir machen den Ansatz

$$z_j = 0, \quad j \in J \setminus \{r\}, \quad z_r = t$$

mit einer zu wählenden Zahl $t \geq 0$. Aus $Az = b$ folgt mit (7.1)

$$z_I = A_I^{-1}b - A_I^{-1}A_J z_J = x_I - tA_I^{-1}a_r,$$

sodass z eindeutig durch t bestimmt ist, und es ergibt sich mit (7.2)

$$f(z) = f(x) + tu_r \leq f(x).$$

Es ist noch $z \geq 0$ sicherzustellen.

(4) Es sei $d = A_I^{-1}a_r$, sodass $z_I = x_I - td$. Gilt $d \leq 0$, so folgt $z \geq 0$ für jede Wahl von $t \geq 0$ und $f(z) \rightarrow -\infty$ für $t \rightarrow \infty$, das heißt M ist unbeschränkt und das Problem ist nicht lösbar.

(5) Es gelte $d_i > 0$ für ein $i \in I$. Die Bedingung $z \geq 0$ ist erfüllt, sofern

$$z_i = x_i - td_i \geq 0$$

gilt. Um den Funktionswert $f(z)$ maximal zu reduzieren und gleichzeitig $z \geq 0$ zu gewährleisten, wählen wir

$$t = \min_{i \in I, d_i > 0} \frac{x_i}{d_i} = \frac{x_s}{d_s}.$$

Damit folgt insbesondere $z_s = 0$. Ist x nichtentartet, so gilt wegen $x_i > 0$ für alle $i \in I$, dass $t > 0$ und der Funktionswert echt reduziert wird.

(6) Wir zeigen, dass $z \in M$ eine Ecke ist. Es gilt $I_z \subset I^{\text{neu}} = (I_x \setminus \{s\}) \cup \{r\}$ und nach Satz 7.2 genügt es zu zeigen, dass die Vektoren $(a_i : i \in I^{\text{neu}})$ linear unabhängig sind.

Seien $\gamma_i, i \in I^{\text{neu}}$, sodass

$$0 = \sum_{i \in I^{\text{neu}}} \gamma_i a_i = \sum_{i \in I \setminus \{s\}} \gamma_i a_i + \gamma_r a_r$$

gilt. Mit $d = A_I^{-1}a_r$ beziehungsweise $a_r = A_I d$ folgt

$$\begin{aligned} 0 &= \sum_{i \in I \setminus \{s\}} \gamma_i a_i + \gamma_r \sum_{i \in I} d_i a_i \\ &= \sum_{i \in I \setminus \{s\}} (\gamma_i + \gamma_r d_i) a_i + \gamma_r d_s a_s. \end{aligned}$$

Da die Vektoren $(a_i : i \in I)$ linear unabhängig sind, folgt $\gamma_i + \gamma_r d_i = 0$ für $i \in I \setminus \{s\}$ und $\gamma_r d_s = 0$. Wegen $d_s \neq 0$ impliziert dies $\gamma_i = 0$ für alle $i \in I^{\text{neu}}$. Also ist z eine Ecke.

Wir haben insgesamt den folgenden Satz bewiesen.

Satz 7.3 Es gelte $\text{rank } A = m$ und $x \in M$ sei eine Ecke. Mit $I_x \subset I$, sodass $A_I \in \mathbb{R}^{m \times m}$ regulär ist, sei $J = \{1, 2, \dots, n\} \setminus I$. Gilt $u = c_J - A_J^\top A_I^{-1} c_I \geq 0$, so ist x Lösung des Problems. Ist $u_r < 0$ für ein $r \in J$, so definiere $d = A_I^{-1}a_r$. Gilt $d \leq 0$, so ist das Problem nicht lösbar. Ist $d_s > 0$ mit $s \in I$, sodass $t = \min_{i \in I, d_i > 0} \frac{x_i}{d_i} = \frac{x_s}{d_s}$, so wird durch

$$x_i^{\text{neu}} = \begin{cases} x_i - t d_i, & i \in I \setminus \{s\}, \\ t, & i = r, \\ 0, & i \in (J \setminus \{r\}) \cup \{s\}, \end{cases}$$

eine Ecke x^{neu} von M definiert mit der Eigenschaft

$$f(x^{\text{neu}}) \leq f(x).$$

Ist x nichtentartet, so ist die Ungleichung strikt.

Bemerkungen 7.5 (i) Das Simplex-Verfahren besteht in der wiederholten Anwendung des Simplex-Schritts, bis der Fall $d \leq 0$ für Unlösbarkeit eintritt, das hinreichende Abbruchkriterium $u \geq 0$ für eine Ecke erfüllt ist oder eine Ecke ein zweites Mal durchlaufen wird. Da es nur endlich viele Ecken gibt, bricht das Verfahren stets ab.

(ii) Es können sogenannte Zyklen auftreten, das heißt man kehrt zu einer bereits besuchten Ecke zurück ohne dass eine Reduktion eintritt oder das Minimum erreicht ist. Dies wird jedoch für praxisrelevante Probleme nicht beobachtet.

(iii) Die neu konstruierte Ecke x^{neu} kann entartet sein, selbst wenn x nichtentartet ist.

(iv) Es gibt $\binom{n}{m}$ viele Ecken, sodass im schlechtesten Fall $\mathcal{O}(n!)$ viele Ecken durchlaufen werden müssten, um zum Minimum zu gelangen. In der Praxis wird nur polynomieller

Aufwand bezüglich n beobachtet, aber es gibt Beispiele, in denen 2^n viele Simplex-Schritte erforderlich sind.

(v) Die algorithmische Konstruktion einer Ecke zur Initialisierung des Verfahrens ist keineswegs trivial.

7.3 Lernziele, Quiz und Anwendung

Sie sollten geometrische Eigenschaften linearer Programme erläutern und die wesentlichen Ideen des Simplex-Schritts erklären können.

Quiz 7.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Jedes lineare Programm in Normalform besitzt eine Lösung	
Der Simplex-Schritt realisiert eine echte Reduktion der zu minimierenden Funktion	
Der Punkt $x = 0$ ist stets eine Ecke der zulässigen Menge	
Jede Ecke ist durch ihre Nulleinträge eindeutig festgelegt	
Im Simplex-Verfahren nimmt die Anzahl der Null-einträge der durchlaufenen Ecken strikt ab	

Anwendung 7.1

- Ein Produkt lagere an den Orten A_1, \dots, A_m in den jeweiligen Mengen a_1, \dots, a_m und es werde an den Orten B_1, \dots, B_n in den Mengen b_1, \dots, b_n benötigt. Es bezeichne c_{ij} die Kosten für den Transport einer Mengeneinheit des Produkts von A_i nach B_j . Formulieren Sie ein lineares Programm in Normalform, um die Gesamtkosten für den Transport des Produkts zu minimieren.
- Ein Produzent von Streusalz erhält den Auftrag, 50 Tonnen Streusalz nach Rom, 20 nach Paris und 30 nach Berlin zu liefern. In Lagern in Prag und Amsterdam sind 40 beziehungsweise 60 Tonnen verfügbar. Wie groß sind die optimalen Transportmengen, wenn die Kosten pro 10 Tonnen Transportmenge in Euro gemäß Tab. 7.1 gegeben sind. Verwenden Sie zur Lösung die MATLAB-Routine `linprog`.

Tab. 7.1 Transportkosten je Tonne Streusalz

	Rom	Paris	Berlin
Prag	700	600	200
Amsterdam	800	300	400

8.1 Lokalisierung

Die Berechnung einzelner oder sämtlicher Eigenwerte einer Matrix und gegebenenfalls zugehöriger Eigenvektoren bezeichnet man als *Eigenwertaufgaben*. Im Allgemeinen ist es schwierig und ineffizient die Nullstellen eines charakteristischen Polynoms zu bestimmen, da schon die Auswertung des Polynoms mit hohem Aufwand verbunden ist.

Satz 8.1 Sei $A \in \mathbb{R}^{n \times n}$ und $\lambda \in \mathbb{C}$ ein Eigenwert von A . Dann gilt

$$\lambda \in \bigcup_{i=1}^n K_i, \quad K_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, \dots, n, j \neq i} |a_{ij}|\}.$$

Die Mengen K_i heißen Gerschgorin-Kreise.

Beweis Es gelte $Ax = \lambda x$ für ein $x \in \mathbb{C}^n \setminus \{0\}$. Dann existiert ein i mit $|x_j| \leq |x_i|$ für alle $j = 1, 2, \dots, n$, und $x_i \neq 0$. Es gilt

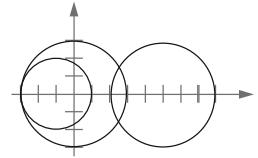
$$\lambda x_i = (Ax)_i = \sum_{j=1}^n a_{ij} x_j$$

und nach Division durch $x_i \neq 0$ folgt

$$\lambda - a_{ii} = \sum_{j=1, \dots, n, j \neq i} a_{ij} \frac{x_j}{x_i}.$$

Die Dreiecksungleichung und $|x_j|/|x_i| \leq 1$ implizieren $\lambda \in K_i$ und damit die Behauptung. \square

Abb. 8.1 Gerschgorin-Kreise
in Beispiel 8.1



Beispiel 8.1 Für die nachfolgende Matrix $A \in \mathbb{R}^{3 \times 3}$ ergeben sich die Gerschgorin-Kreise K_1, K_2, K_3 , s. Abb. 8.1.

$$A = \begin{bmatrix} 5 & 1 & 2 \\ 1 & -1 & 1 \\ 2 & 1 & 0 \end{bmatrix}, \quad \begin{aligned} K_1 &= \{z \in \mathbb{C} : |z - 5| \leq 3\}, \\ K_2 &= \{z \in \mathbb{C} : |z + 1| \leq 2\}, \\ K_3 &= \{z \in \mathbb{C} : |z| \leq 3\}. \end{aligned}$$

Im Fall symmetrischer Matrizen lassen sich die Eigenwerte durch Extremwerte des sogenannten *Rayleigh-Quotienten* charakterisieren.

Satz 8.2 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch. Für den maximalen beziehungsweise minimalen Eigenwert von A gilt

$$\lambda_{\min} = \min_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^\top A x}{\|x\|_2^2}, \quad \lambda_{\max} = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^\top A x}{\|x\|_2^2}.$$

Beweis Es sei $(v_1, v_2, \dots, v_n) \subset \mathbb{R}^n$ eine Orthonormalbasis des \mathbb{R}^n bestehend aus Eigenvektoren zu den Eigenwerten $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \in \mathbb{R}$ der Matrix A . Zu $x \in \mathbb{R}^n$ existieren $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}$, sodass $x = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n$ und es gilt

$$Ax = \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 + \dots + \alpha_n \lambda_n v_n.$$

Die Orthonormalität $v_i \cdot v_j = \delta_{ij}$, $1 \leq i, j \leq n$, der Vektoren v_1, v_2, \dots, v_n impliziert

$$\begin{aligned} x^\top x &= \left(\sum_{i=1}^n \alpha_i v_i \right) \cdot \left(\sum_{j=1}^n \alpha_j v_j \right) = \sum_{i,j=1}^n \alpha_i \alpha_j v_i \cdot v_j = \sum_{i=1}^n \alpha_i^2, \\ x^\top Ax &= \left(\sum_{i=1}^n \alpha_i v_i \right) \cdot \left(\sum_{j=1}^n \alpha_j A v_j \right) = \sum_{i,j=1}^n \alpha_i \lambda_j \alpha_j v_i \cdot v_j = \sum_{i=1}^n \lambda_i \alpha_i^2. \end{aligned}$$

Daraus folgt

$$x^\top A x \geq \lambda_n \sum_{i=1}^n \alpha_i^2 = \lambda_n \|x\|_2^2 = \lambda_{\min} \|x\|_2^2,$$

wobei Gleichheit für $x = v_n$ gilt. Die Aussage für $\lambda_1 = \lambda_{\max}$ folgt analog. \square

8.2 Konditionierung

Eine Matrix $A \in \mathbb{R}^{n \times n}$ ist komplex diagonalisierbar, wenn es eine reguläre Matrix $V \in \mathbb{C}^{n \times n}$ und eine Diagonalmatrix $D \in \mathbb{C}^{n \times n}$ gibt, sodass $A = VDV^{-1}$ gilt. In dieser Situation lässt sich folgendes Resultat von Bauer und Fike beweisen.

Satz 8.3 *Sei $A \in \mathbb{R}^{n \times n}$ komplex diagonalisierbar mit $A = VDV^{-1}$, sei $E \in \mathbb{R}^{n \times n}$ und sei $\tilde{\lambda} \in \mathbb{C}$ ein Eigenwert von $A + E$. Dann existiert ein komplexer Eigenwert von A , sodass*

$$|\tilde{\lambda} - \lambda| \leq \text{cond}_2(V) \|E\|_2,$$

wobei die Operatornorm und Konditionszahl in naheliegender Weise für komplexe Matrizen verallgemeinert seien.

Beweis Ist $\tilde{\lambda}$ auch ein Eigenwert von A , so ist die Aussage trivial. Sei im Folgenden $\tilde{\lambda}$ kein Eigenwert von A , sodass $\tilde{\lambda}I_n - A$ invertierbar ist. Ist $x \in \mathbb{C}^n$ ein Eigenvektor von $A + E$ zum Eigenwert $\tilde{\lambda}$, so gilt

$$Ex = (A + E)x - Ax = \tilde{\lambda}x - Ax = (\tilde{\lambda}I_n - A)x,$$

also $x = (\tilde{\lambda}I_n - A)^{-1}Ex$, das heißt 1 ist Eigenwert von $(\tilde{\lambda}I_n - A)^{-1}E$. Damit folgt

$$\begin{aligned} 1 &\leq \|(\tilde{\lambda}I_n - A)^{-1}E\|_2 = \|(\tilde{\lambda}VV^{-1} - VDV^{-1})^{-1}E\|_2 \\ &= \|V(\tilde{\lambda}I_n - D)^{-1}V^{-1}E\|_2 \leq \|V\|_2 \|(\tilde{\lambda}I_n - D)^{-1}\|_2 \|V^{-1}\|_2 \|E\|_2 \\ &= \text{cond}_2(V) \max_{\lambda \in \sigma(A)} |\tilde{\lambda} - \lambda|^{-1} \|E\|_2, \end{aligned}$$

wobei das Maximum über alle komplexen Eigenwerte λ von A gebildet wird. Mit der Identität $\max_{x \in X} |x|^{-1} = (\min_{x \in X} |x|)^{-1}$ folgt die Behauptung. \square

Bemerkungen 8.1 (i) Nicht jede Matrix ist komplex diagonalisierbar, jedoch ist jede Matrix komplex trigonalisierbar.

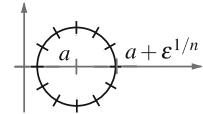
(ii) Normale Matrizen, das heißt Matrizen mit der Eigenschaft $AA^\top = A^\top A$, sind komplex diagonalisierbar mit unitärer Transformationsmatrix V , das heißt V erfüllt $\overline{V}^\top V = I_n$.

Korollar 8.1 *Sei $A \in \mathbb{R}^{n \times n}$ normal, $E \in \mathbb{R}^{n \times n}$ und $\tilde{\lambda}$ ein Eigenwert von $A + E$. Dann existiert ein Eigenwert von A mit*

$$|\lambda - \tilde{\lambda}| \leq \|E\|_2$$

Beweis Da A diagonalisierbar ist mit einer unitären Matrix $V \in \mathbb{C}^{n \times n}$ und da $\text{cond}_2(V) = 1$ gilt, folgt die Abschätzung aus dem vorangegangenen Satz. \square

Abb. 8.2 Die Eigenwerte der Matrix A_ε aus Beispiel 8.2 liegen auf einer Kreislinie mit Mittelpunkt a und Radius $\varepsilon^{1/n}$



Für normale, insbesondere symmetrische Matrizen ist die Bestimmung der Eigenwerte damit ein gut konditioniertes Problem. Im Allgemeinen ist dies nicht der Fall.

Beispiel 8.2 Ist $p(t) = t^n + a_{n-1}t^{n-1} + \dots + a_1t + a_0$ ein beliebiges Polynom, so gilt $p(t) = (-1)^n \det(A - tI_n)$ mit der Frobenius-Begleitmatrix

$$A = \begin{bmatrix} 0 & & & -a_0 \\ 1 & 0 & & -a_1 \\ \ddots & \ddots & & \vdots \\ & 1 & 0 & -a_{n-2} \\ & & 1 & -a_{n-1} \end{bmatrix},$$

insbesondere entsprechen die komplexen Nullstellen von p den komplexen Eigenwerten von A . Für $a \in \mathbb{R} \setminus \{0\}$ und $\varepsilon > 0$ hat das Polynom $p_0(t) = (t - a)^n$ die n -fache Nullstelle $\lambda = a$, während das Polynom $p_\varepsilon(t) = (t - a)^n - \varepsilon$ die Nullstellen $\lambda_k = a - \varepsilon^{1/n} e^{i2\pi k/n}$, $k = 1, 2, \dots, n$, besitzt, s. Abb. 8.2. Die Polynome p_0 und p_ε unterscheiden sich nur im konstanten Koeffizienten und für die Differenz $A - A_\varepsilon$ der zugehörigen Begleitmatrizen gilt $\|A - A_\varepsilon\|_\ell = \varepsilon$ für $\ell \in \{1, 2, \infty\}$. Es gilt $|\lambda - \lambda_k| = \varepsilon^{1/n}$ und für die relativen Fehler folgt

$$\begin{aligned} \frac{|\lambda - \lambda_k|}{|\lambda|} &= \frac{\varepsilon^{1/n}}{|a|} \frac{\|A\|_\ell}{\|A\|_\ell} \frac{\|A - A_\varepsilon\|_\ell}{\varepsilon} \\ &= \frac{\varepsilon^{1/n}}{|a|} \frac{\|A\|_\ell}{\|A\|_\ell} \frac{\|A - A_\varepsilon\|_\ell}{\|A\|_\ell}. \end{aligned}$$

Der Faktor $\varepsilon^{(1-n)/n}$ ist unbeschränkt für $\varepsilon \rightarrow 0$, sofern $n > 1$ gilt.

8.3 Potenzmethode

Sei $A \in \mathbb{R}^{n \times n}$ reell diagonalisierbar mit Eigenwerten $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$ und zugehörigen linear unabhängigen Eigenvektoren $v_1, v_2, \dots, v_n \in \mathbb{R}^n$, für die $\|v_i\|_2 = 1$, $i = 1, 2, \dots, n$, gelte. Für jedes $x \in \mathbb{R}^n$ mit

$$x = \sum_{i=1}^n \alpha_i v_i$$

ergibt die k -malige Anwendung von A auf v

$$A^k x = A^{k-1} \left(\sum_{i=1}^n \lambda_i \alpha_i v_i \right) = \dots = \sum_{i=1}^n \lambda_i^k \alpha_i v_i.$$

Ist λ_1 der betragsmäßig größte Eigenwert, so folgt für k hinreichend groß, dass

$$A^k x \approx \alpha_1 \lambda_1^k v_1.$$

Wir betrachten die Normen $\|A^k x\|_2$ sowie $\|A^{k+1} x\|_2$ und bilden deren Quotient, sodass wegen $\|v_1\|_2 = 1$ folgt

$$\frac{\|A^{k+1} x\|_2}{\|A^k x\|_2} \approx |\lambda_1|.$$

Mit dieser Beobachtung kann man den dominierenden Eigenwert einer Matrix bestimmen.

Algorithmus 8.1 (von Mises-Potenzmethode) Seien $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n \setminus \{0\}$ und $\varepsilon_{\text{stop}} > 0$. Setze $x_0 = x/\|x\|_2$, $\mu_0 = 0$ und $k = 0$.

- (1) Berechne $\tilde{x}_{k+1} = Ax_k$, $\mu_{k+1} = \|\tilde{x}_{k+1}\|_2$ und $x_{k+1} = \tilde{x}_{k+1}/\|\tilde{x}_{k+1}\|_2$.
- (2) Stoppe falls $|\mu_{k+1} - \mu_k| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Bemerkung 8.2 Induktiv zeigt man, dass $x_k = A^k x / \|A^k x\|_2$ gilt. Um ein Verlassen des Bereichs darstellbarer Zahlen zu vermeiden, muss in jedem Iterationsschritt eine Normierung durchgeführt werden.

Wir zeigen, dass die Folge $(x_k)_{k \in \mathbb{N}}$ einen normierten Eigenvektor zum betragsmäßig maximalen Eigenwert von A approximiert, dessen Betrag durch die Folge $(\mu_k)_{k \in \mathbb{N}}$ angehert wird.

Satz 8.4 Es gelte $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$ und es sei $x = \sum_{i=1}^n \alpha_i v_i$ mit den normierten Eigenvektoren v_1, v_2, \dots, v_n von A . Gilt $\alpha_1 \neq 0$ so folgt mit $q = |\lambda_2|/|\lambda_1| < 1$ und $k \geq K$, dass

$$|\|Ax_k\|_2 - |\lambda_1|| \leq 4\|A\|_2 c q^k$$

mit einer von k unabhängigen Konstanten $c \geq 0$.

Beweis Für jedes $k \geq 0$ gilt

$$A^k x = \lambda_1^k \alpha_1 \left(v_1 + \sum_{i=2}^n \frac{\lambda_i^k}{\lambda_1^k} \frac{\alpha_i}{\alpha_1} v_i \right) = \lambda_1^k \alpha_1 (v_1 + w_k),$$

wobei $w_k \in \mathbb{R}^n$ durch die Summe definiert sei. Es folgt

$$\|w_k\|_2 \leq q^k \sum_{i=2}^n \frac{|\alpha_i|}{|\alpha_1|} = cq^k.$$

Ferner ist

$$\begin{aligned} x_k &= \frac{A^k x}{\|A^k x\|_2} = \frac{\lambda_1^k \alpha_1 (v_1 + w_k)}{|\lambda_1^k \alpha_1| \|v_1 + w_k\|_2} \\ &= \text{sign}(\lambda_1^k \alpha_1) v_1 + \text{sign}(\lambda_1^k \alpha_1) \left(\frac{v_1 + w_k}{\|v_1 + w_k\|_2} - v_1 \right) \\ &= \text{sign}(\lambda_1^k \alpha_1) v_1 + r_k. \end{aligned}$$

Mit der umgekehrten Dreiecksungleichung $\|a\|_2 - \|b\|_2 \leq \|a + b\|_2$ sowie der gewöhnlichen Dreiecksungleichung $\|a - b\|_2 \leq \|a\|_2 + \|b\|_2$ folgt

$$1 - cq^k \leq \|v_1\|_2 - \|w_k\|_2 \leq \|v_1 + w_k\|_2 \leq \|v_1\|_2 + \|w_k\|_2 \leq 1 + cq^k$$

und es folgt für k hinreichend groß, sodass $cq^k \leq 1/2$ gilt,

$$\begin{aligned} \|r_k\|_2 &= \left\| \frac{v_1(1 - \|v_1 + w_k\|_2) + w_k}{\|v_1 + w_k\|_2} \right\| \\ &\leq \frac{|1 - \|v_1 + w_k\|_2| + \|w_k\|_2}{\|v_1 + w_k\|_2} \leq \frac{2cq^k}{1 - cq^k} \leq 4cq^k. \end{aligned}$$

Für $\tilde{x}_{k+1} = Ax_k$ liefert die obige Darstellung von x_k , dass

$$\tilde{x}_{k+1} = \lambda_1 \text{sign}(\lambda_1^k \alpha_1) v_1 + Ar_k$$

und somit

$$\|\tilde{x}_{k+1} - \lambda_1 \text{sign}(\lambda_1^k \alpha_1) v_1\|_2 \leq \|Ar_k\|_2 \leq 4\|A\|_2 cq^k.$$

Eine weitere Anwendung der umgekehrten Dreiecksungleichung zeigt unter Verwendung von $\tilde{x}_{k+1} = Ax_k$ und $\|v_1\|_2 = 1$ die Behauptung. \square

Bemerkungen 8.3 (i) In jedem Schritt der Iteration verringert sich der Approximationsfehler um den Faktor $q < 1$.

(ii) Die letzte Ungleichung im Beweis zeigt, dass $\lambda_1 < 0$ genau dann gilt, wenn die Vorzeichen von $(x_k)_{k=1,2,\dots}$ alternieren, und dass die Folge $(x_k)_{k=1,2,\dots}$ bis auf den Faktor $\text{sign } \lambda_1^k$ gegen einen Eigenvektor konvergiert.

(iii) Die Voraussetzung $\alpha_1 \neq 0$ muss im konkreten Fall sichergestellt werden. Aufgrund von Rundungsfehlern kann dies zwar angenommen werden, allerdings kann dann die Konstante $c \sim 1/|\alpha_1|$ sehr groß werden.

Im Fall symmetrischer Matrizen lässt sich eine bessere Konvergenzaussage beweisen, die impliziert, dass der Fehler in jedem Schritt um den Faktor q^2 verringert wird.

Satz 8.5 Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch, so gilt unter den Voraussetzungen des vorigen Satzes

$$|\lambda_1 - x_k^\top A x_k| \leq 2\|A\|_2 c^2 q^{2k}.$$

Beweis Ist A symmetrisch, so können die Eigenvektoren (v_1, v_2, \dots, v_n) im Beweis des vorigen Satzes als orthonormal angenommen werden und es gilt

$$A^k x = \lambda_1^k \alpha_1 (v_1 + w_k)$$

mit $v_1 \cdot w_k = 0$. Es sei

$$\gamma_k = \text{sign}(\lambda_1^k \alpha_1) \|v_1 + w_k\|_2^{-1}$$

und wegen $\|v_1 + w_k\|_2^2 = \|v_1\|_2^2 + \|w_k\|_2^2 \geq 1$ gilt $|\gamma_k| \leq 1$. Es folgt

$$x_k = \frac{A^k x}{\|A^k x\|_2} = \frac{\lambda_1^k \alpha_1 (v_1 + w_k)}{|\lambda_1^k \alpha_1| \|v_1 + w_k\|_2} = \gamma_k v_1 + \gamma_k w_k$$

und damit

$$\begin{aligned} (\lambda_1 I_n - A)x_k &= \gamma_k \lambda_1 v_1 + \gamma_k \lambda_1 w_k - \gamma_k A v_1 - \gamma_k A w_k \\ &= \gamma_k (\lambda_1 I_n - A) w_k. \end{aligned}$$

Da $A w_k \in \text{span}\{v_2, v_3, \dots, v_n\}$ gilt, ist der Vektor auf der rechten Seite orthogonal zu v_1 . Mit der Cauchy-Schwarz-Ungleichung $a^\top b \leq \|a\|_2 \|b\|_2$ und $\gamma_k \leq 1$ folgt, dass

$$\begin{aligned} |x_k^\top (\lambda_1 I_n - A)x_k| &= \gamma_k^2 |(v_1 + w_k)^\top (\lambda_1 I_n - A) w_k| \\ &\leq \|w_k\|_2 \|\lambda_1 (I_n - A)\|_2 \|w_k\|_2 \\ &\leq (|\lambda_1| + \|A\|_2) \|w_k\|_2^2 \\ &= 2\|A\|_2 \|w_k\|_2^2, \end{aligned}$$

wobei in der letzten Gleichung $|\lambda_1| = \|A\|_2$ verwendet wurde. Schließlich folgt mit

$$\lambda_1 - x_k^\top A x_k = x_k^\top (\lambda_1 I_n - A) x_k$$

und $\|w_k\|_2 \leq cq^k$, dass

$$|\lambda_1 - x_k^\top A x_k| \leq 2\|A\|_2 \|w_k\|_2^2 \leq 2\|A\|_2 c^2 q^{2k}$$

und somit die Behauptung. \square

Bemerkungen 8.4 (i) Gilt $0 < |\lambda_n| < |\lambda_{n-1}| \leq \dots \leq |\lambda_1|$, so liefert die Potenzmethode mit A^{-1} statt A eine Approximation von $|\lambda_n|^{-1}$. Dies wird als *inverse Iteration* bezeichnet.

(ii) Wendet man die Potenzmethode auf die Matrix $(A - \mu I_n)^{-1}$ an, so konvergiert sie unter geeigneten Voraussetzungen gegen den Eigenwert, der am nächsten bei μ liegt.

(iii) Der dominierende Eigenwert kann mehrfach auftreten, das heißt die Voraussetzung der Sätze lässt sich auf $|\lambda_1| = \dots = |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n| \geq 0$ abschwächen.

8.4 QR-Verfahren

Das *QR*-Verfahren berechnet unter geeigneten Voraussetzungen Approximationen sämtlicher Eigenwerte einer Matrix.

Algorithmus 8.2 (QR-Verfahren) Sei $A \in \mathbb{R}^{n \times n}$. Setze $A_0 = A$ und $k = 0$.

- (1) Bestimme die *QR*-Zerlegung $A_k = Q_k R_k$ und setze $A_{k+1} = R_k Q_k$.
- (2) Stoppe falls $\|A_{k+1} - A_k\| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Das Verfahren lässt sich als verallgemeinerte Potenzmethode interpretieren und die Iterierten sind ähnlich zueinander.

Lemma 8.1([5]) Es gilt

$$\begin{aligned} A_{k+1} &= Q_k^\top A_k Q_k = (Q_0 \dots Q_k)^\top A (Q_0 \dots Q_k), \\ A^{k+1} &= (Q_0 \dots Q_k)(R_k \dots R_0). \end{aligned}$$

Beweis Aus $A_{k+1} = R_k Q_k$ und $A_k = Q_k R_k$ beziehungsweise $R_k = Q_k^\top A_k$ folgt $A_{k+1} = Q_k^\top A_k Q_k$ und die wiederholte Anwendung dieses Arguments beweist die erste Gleichung. Die zweite Gleichung ist klar für $k = 0$ und sie gelte für ein $k \geq 0$. Dann folgt aus

$$Q_{k+1} R_{k+1} = A_{k+1} = (Q_0 \dots Q_k)^\top A (Q_0 \dots Q_k),$$

dass

$$(Q_0 \dots Q_k) Q_{k+1} R_{k+1} (R_k \dots R_1) = A(Q_0 \dots Q_k)(R_k \dots R_1) = AA^{k+1} = A^{k+2}$$

und dies beweist die Aussage. \square

Mit Hilfe dieses Lemmas und einer Stabilitätseigenschaft der QR -Zerlegung lässt sich das folgende Resultat beweisen, siehe beispielsweise [5].

Satz 8.6 Sei $A \in \mathbb{R}^{n \times n}$ diagonalisierbar mit $A = VDV^{-1}$ derart, dass für die Eigenwerte $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$ gilt

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > 0$$

und die Inverse der Matrix V eine LU-Zerlegung besitzt. Dann gilt

$$\|\text{diag}(A_k) - \text{diag}(D)\|_2 \leq cq^k$$

mit $q = \max_{1 \leq i < j \leq n} |\lambda_j| / |\lambda_i|$ und einer Konstanten $c \geq 0$.

Bemerkungen 8.5 (i) In der Praxis beobachtet man Konvergenz unter deutlich schwächeren Voraussetzungen an A .

(ii) Im Allgemeinen führt ein Schritt im QR -Verfahren auf einen Aufwand der Ordnung $\mathcal{O}(n^3)$. Wird A zunächst mittels Householder-Transformationen auf eine sogenannte Hessenberg-Matrix

$$\hat{A} = H^\top AH = \begin{bmatrix} \hat{a}_{11} & \dots & & \hat{a}_{1n} \\ \hat{a}_{21} & \hat{a}_{22} & \dots & \hat{a}_{2n} \\ & \hat{a}_{32} & \dots & \hat{a}_{3n} \\ & & \ddots & \vdots \\ & & & \hat{a}_{n,n-1} & \hat{a}_{nn} \end{bmatrix}$$

transformiert, das heißt gilt $\hat{a}_{ij} = 0$ für $i > j + 1$, so lässt sich die QR -Zerlegung mit Givens-Rotationen in $\mathcal{O}(n^2)$ Schritten bestimmen.

8.5 Jacobi-Verfahren

Der Satz über die Gerschgorin-Kreise zeigt, dass die Diagonaleinträge einer Matrix Approximationen der Eigenwerte definieren, die besonders gut sind, wenn die Nichtdiagonalelemente klein sind. Im Jacobi-Verfahren werden diese Einträge einer symmetrischen Matrix sukzessive mit Ähnlichkeitstransformationen verringert. Wir folgen der Darstellung in [7].

Definition 8.1 Für $A \in \mathbb{R}^{n \times n}$ sei

$$\mathcal{N}(A) = \|A\|_{\mathcal{F}}^2 - \sum_{i=1}^n a_{ii}^2 = \sum_{1 \leq i, j \leq n, i \neq j} a_{ij}^2.$$

Offensichtlich ist A eine Diagonalmatrix genau dann, wenn $\mathcal{N}(A) = 0$ gilt. Allgemeiner lässt sich zeigen, dass zu jedem Diagonaleintrag a_{jj} , $1 \leq j \leq n$, ein Eigenwert λ mit der Eigenschaft $|\lambda - a_{jj}| \leq \sqrt{\mathcal{N}(A)}$ existiert. Aus dem Gerschgorinschen Kreisesatz erhält man die schwächere Aussage, dass zu jedem Eigenwert λ von A ein Diagonaleintrag a_{jj} existiert, sodass $|\lambda - a_{jj}| \leq (n-1)^{1/2} \sqrt{\mathcal{N}(A)}$ gilt.

Definition 8.2 Für $c, s \in \mathbb{R}$ mit $c^2 + s^2 = 1$ und $1 \leq p, q \leq n$ wird eine *Givens-Rotation* $G_{pq} \in O(n)$ definiert durch

$$(G_{pq})_{ij} = \begin{cases} 1, & i = j, i \neq p, \\ 1, & i = j, i \neq q, \\ c, & i = p, j = p, \\ c, & i = q, j = q, \\ s, & i = q, j = p, \\ -s, & i = p, j = q, \\ 0 & \text{sonst.} \end{cases} \quad G_{pq} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & & & -s & & c \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix}$$

Im folgenden Lemma wird verwendet, dass die Frobenius-Norm invariant ist unter orthogonalen Transformationen, das heißt dass $\|Q^\top M\|_{\mathcal{F}} = \|MQ\|_{\mathcal{F}} = \|M\|_{\mathcal{F}}$ für alle $M \in \mathbb{R}^{n \times n}$ und $Q \in O(n)$ gilt. Dies folgt zum Beispiel aus $\|M\|_{\mathcal{F}}^2 = \text{tr}(M^\top M)$ und $\|M\|_{\mathcal{F}} = \|M^\top\|_{\mathcal{F}}$.

Lemma 8.2 Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch und G_{pq} eine beliebige Givens-Rotation, so gilt für $B = G_{pq}^\top A G_{pq}$, dass

$$\mathcal{N}(B) = \mathcal{N}(A) - 2(a_{pq}^2 - b_{pq}^2),$$

wobei $b_{pq} = cs(a_{qq} - a_{pp}) + (c^2 - s^2)a_{pq}$.

Beweis Man überprüft direkt, dass die Einträge der symmetrischen Matrix B gegeben sind durch $b_{ij} = a_{ij}$, sofern $i, j \notin \{p, q\}$, sowie

$$\begin{aligned} b_{pp} &= c^2 a_{pp} + 2c s a_{pq} + s^2 a_{qq}, \\ b_{qq} &= s^2 a_{pp} - 2c s a_{pq} + c^2 a_{qq}, \\ b_{pq} &= b_{qp} = c s (a_{qq} - a_{pp}) + (c^2 - s^2) a_{pq}, \\ b_{ip} &= c a_{ip} + s a_{iq}, \quad i \in \{1, 2, \dots, n\} \setminus \{p, q\}, \\ b_{iq} &= -s a_{ip} + c a_{iq}, \quad i \in \{1, 2, \dots, n\} \setminus \{p, q\}. \end{aligned}$$

Mit $\|B\|_{\mathcal{F}} = \|A\|_{\mathcal{F}}$ folgt

$$\begin{aligned} \mathcal{N}(B) &= \|B\|_{\mathcal{F}}^2 - \sum_{i=1}^n a_{ii}^2 + \sum_{i=1}^n (a_{ii}^2 - b_{ii}^2) \\ &= \|A\|_{\mathcal{F}}^2 - \sum_{i=1}^n a_{ii}^2 + (a_{pp}^2 - b_{pp}^2 + a_{qq}^2 - b_{qq}^2) \\ &= \mathcal{N}(A) + a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2. \end{aligned}$$

Die Formeln für die Einträge von B zeigen, dass

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{pq} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

Identifizieren wir diese Identität mit $\hat{b} = g^\top \hat{a} g$, so folgt $\|\hat{b}\|_{\mathcal{F}}^2 = \|\hat{a}\|_{\mathcal{F}}^2$ beziehungsweise

$$b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 = a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2,$$

also

$$a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2 = 2(b_{pq}^2 - a_{pq}^2)$$

und dies impliziert die Behauptung. \square

Kann die Givens-Rotation G_{pq} so gewählt werden, dass $b_{pq} = 0$ gilt, so ergibt sich eine Verringerung der Nichtdiagonaleinträge. Durch Betrachtung von $c = \cos(\alpha)$, $s = \pm \sin(\alpha)$ und $D = \cos(2\alpha)$ erhält man folgendes Resultat.

Lemma 8.3 Ist $a_{pq} \neq 0$ und die Matrix G_{pq} definiert durch $c = \sqrt{(1+D)/2}$ und $s = -\text{sign}(a_{pq}) \sqrt{(1-D)/2}$ mit

$$D = \frac{a_{pp} - a_{qq}}{\left((a_{pp} - a_{qq})^2 + 4a_{pq}^2\right)^{1/2}} \in [-1, 1]$$

so gilt $b_{pq} = 0$.

Beweis Übungsaufgabe. □

Um eine größtmögliche Reduktion von $\mathcal{N}(A)$ zu erzielen, ist es naheliegend, das betragsmäßig größte Nichtdiagonalelement von A auszuwählen.

Satz 8.7 Ist a_{pq} das betragsmäßig größte Nichtdiagonalelement von A , so gilt mit der im vorigen Lemma definierten Givens-Rotation G_{pq} für die Matrix $B = G_{pq}^\top A G_{pq}$ und mit $\varepsilon_n = 2/(n(n-1))$, dass

$$\mathcal{N}(B) \leq (1 - \varepsilon_n) \mathcal{N}(A),$$

Beweis Nach Wahl von a_{pq} gilt $\mathcal{N}(A) \leq n(n-1)a_{pq}^2$. Damit folgt

$$\mathcal{N}(B) = \mathcal{N}(A) - 2a_{pq}^2 \leq \left(1 - \frac{2}{n(n-1)}\right) \mathcal{N}(A),$$

also die behauptete Abschätzung. □

Aus dem Satz folgt die Konvergenz des folgenden Verfahrens.

Algorithmus 8.3 (Jacobi-Verfahren) Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch. Setze $A_0 = A$ und $k = 0$.

- (1) Seien p, q die Indizes des betragsmäßig größten Nichtdiagonalelements von A_k und wähle die Givens-Rotation G_{pq} , sodass für $A_{k+1} = G_{pq}^\top A_k G_{pq}$ der Eintrag $(A_{k+1})_{pq}$ verschwindet.
- (2) Stoppe falls $\mathcal{N}(A_{k+1}) \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Bemerkungen 8.6 (i) Im Allgemeinen werden $\mathcal{O}(n^2 \log(1/\varepsilon_{\text{stop}}))$ viele Iterationen benötigt, um $\mathcal{N}(A_{k+1}) \leq \varepsilon_{\text{stop}}$ zu garantieren.

(ii) Ein bereits zu Null transformierter Eintrag kann im Laufe des Verfahrens wieder von Null abweichen.

(iii) Das Verfahren konstruiert eine Faktorisierung $A = GDG^\top$ mit einer orthogonalen Matrix G und einer näherungsweisen Diagonalmatrix D , insbesondere sind damit die Spaltenvektoren von G näherungsweise Eigenvektoren von A .

(iv) Da die Suche nach dem maximalen Nichtdiagonalelement sehr aufwendig ist, arbeitet man in der Praxis eher sukzessive alle Nichtdiagonalelemente ab und wiederholt dies so oft, bis $\mathcal{N}(A_k)$ hinreichend klein ist. Dieses Vorgehen bezeichnet man als *zyklisches Jacobi-Verfahren*.

8.6 Lernziele, Quiz und Anwendung

Ihnen sollten verschiedene Eigenwertaufgaben und deren Konditionierung bekannt sein. Sie sollten in der Lage sein, verschiedene Verfahren zur numerischen Lösung von Eigenwertaufgaben herzuleiten und deren Konvergenz- und Aufwandseigenschaften präzisieren können.

Quiz 8.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Ist $A \in \mathbb{R}^{n \times n}$ regulär, so ist die Berechnung der Eigenwerte von A ein gut konditioniertes Problem	
Die Matrizen A und A^\top besitzen dieselben Eigenwerte und Eigenvektoren	
Die Konvergenzgeschwindigkeit der Potenzmethode ist abhängig vom Verhältnis des betragsmäßig größten zum betragsmäßig kleinsten Eigenwert	
Die Durchführung eines Schritt des QR-Verfahrens erfordert einen Aufwand der Ordnung $\mathcal{O}(n^3)$	
Das Jacobi-Verfahren ist für jede diagonalisierbare Matrix durchführbar und konvergent	

Anwendung 8.1 Die Zahlen 1, 2, 3 seien Indikatoren für die Verständlichkeit einer Mathematik-Vorlesung, wobei 1 für sehr gut, 2 für gut und 3 für wenig verständlich stehe. Die Wahrscheinlichkeit, dass auf eine Vorlesung der Verständlichkeit j eine Vorlesung der Verständlichkeit i folgt sei mit p_{ij} bezeichnet und es gelte

$$P = \begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.5 & 0.2 & 0.1 \\ 0.4 & 0.5 & 0.3 \end{bmatrix}.$$

Auf eine sehr gut verständliche Vorlesung folgt also mit 40% Wahrscheinlichkeit eine wenig verständliche Vorlesung. Bezeichnet der Vektor $x_0 \in [0, 1]^3$ die Verständlichkeit der aktuellen Vorlesung, so sind die Wahrscheinlichkeiten der Verständlichkeitsindikatoren der k Vorlesungen später stattfindenden Veranstaltung gegeben durch $x_k = P^k x_0$.

- (i) Testen Sie experimentell die Konvergenz der Folge $(x_k)_{k \geq 0}$, wobei x_0 durch kanonische Basisvektoren im \mathbb{R}^3 definiert sei, das heißt nach wie vielen Schritten gilt $\|x_k - x_{k+1}\|_1 \leq 10^{-5}$? Was bedeutet dies für die Verständlichkeit der Vorlesungen?

- (ii) Angenommen, die Folge $(x_k)_{k \geq 0}$ wird stationär, das heißt es gilt $x_k \approx x^*$ für alle $k \geq K$ mit einer hinreichend großen Zahl $K \geq 0$. Wie lässt sich x^* charakterisieren?
- (iii) Testen Sie fünf mit `rand(3, 1)` generierte, skalierte Startvektoren $x_0 \in [0, 1]^3$ mit $\|x_0\|_1 = 1$ und charakterisieren Sie die stationären Punkte. Betrachten Sie dazu die Eigenwerte und -vektoren von P , die Sie in MATLAB mit `[V, D] = eig(P)` bestimmen können.

9.1 Inexakte Lösung

Aufgrund von Modell- und Datenfehlern sowie numerischer Rundung ist es im Allgemeinen nicht notwendig und sinnvoll, ein lineares Gleichungssystem exakt im Sinne der Rechnerarithmetik zu lösen. Wir werden die Lösung eines Gleichungssystems durch eine Folge von Approximationslösungen annähern, und die Iteration abbrechen, wenn die Gleichung hinreichend gut erfüllt ist. Dieser Ansatz führt in vielen Fällen zu einer erheblichen Reduktion des Aufwands. Wir folgen in diesem Kapitel der Darstellung in [8].

9.2 Banachscher Fixpunktsatz

Der Banachsche Fixpunktsatz definiert ein Verfahren, das unter geeigneten Voraussetzungen die Lösung einer Fixpunktgleichung approximiert.

Definition 9.1 Eine Abbildung $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ heißt *Kontraktion* bezüglich einer Norm $\|\cdot\|$ auf \mathbb{R}^n , wenn es eine Zahl $q < 1$ gibt, sodass für alle $x, y \in \mathbb{R}^n$ gilt

$$\|\Phi(x) - \Phi(y)\| \leq q \|x - y\|.$$

Für Kontraktionen führt die folgende Fixpunktiteration zu konvergenten Approximationen.

Algorithmus 9.1 Sei $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine Kontraktion und $x^0 \in \mathbb{R}^n$. Setze $k = 0$.

- (1) Definiere $x^{k+1} = \Phi(x^k)$.
- (2) Stoppe falls $\|x^{k+1} - x^k\| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Satz 9.1 Ist $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine Kontraktion, so besitzt Φ einen eindeutigen Fixpunkt $x^* \in \mathbb{R}^n$, das heißt es gilt $\Phi(x^*) = x^*$. Für jeden Startwert $x^0 \in \mathbb{R}^n$ definiert die Fixpunktiteration $x^{k+1} = \Phi(x^k)$ für $k = 0, 1, 2, \dots$, eine Folge von Approximationen von x^* mit der Eigenschaft

$$\|x^k - x^*\| \leq \frac{q^k}{1-q} \|x^1 - x^0\|,$$

insbesondere konvergiert die Folge $(x^k)_{k \in \mathbb{N}}$ gegen x^* .

Beweis Die Abbildung Φ hat höchstens einen Fixpunkt, denn sind $x^*, y^* \in \mathbb{R}^n$ Fixpunkte, so gilt

$$\|x^* - y^*\| = \|\Phi(x^*) - \Phi(y^*)\| \leq q \|x^* - y^*\|$$

und da $q < 1$ ist, folgt $x^* = y^*$. Die durch das Verfahren $x^{k+1} = \Phi(x^k)$ definierte Folge ist eine Cauchy-Folge, denn aus

$$\|x^k - x^{k+1}\| = \|\Phi(x^{k-1}) - \Phi(x^k)\| \leq q \|x^{k-1} - x^k\|$$

folgt induktiv

$$\|x^k - x^{k+1}\| \leq q^k \|x^0 - x^1\|$$

und mit der Dreiecksungleichung für $n \geq m$

$$\begin{aligned} \|x^m - x^n\| &= \|x^m - x^{m+1} + x^{m+1} - x^{m+2} + x^{m+2} - \cdots - x^{n-1} + x^{n-1} - x^n\| \\ &\leq \sum_{k=m}^{n-1} \|x^k - x^{k+1}\| \leq \sum_{k=m}^{n-1} q^k \|x^0 - x^1\| = \|x^0 - x^1\| q^m \sum_{k=0}^{n-m-1} q^k \\ &= \|x^0 - x^1\| q^m \frac{1 - q^{n-m}}{1 - q} \leq \|x^0 - x^1\| \frac{q^m}{1 - q}. \end{aligned}$$

Als Cauchy-Folge hat $(x^k)_{k \in \mathbb{N}}$ einen Grenzwert $x^* \in \mathbb{R}^n$ und für diesen folgt mit der Lipschitz-Stetigkeit von Φ , dass

$$x^* = \lim_{k \rightarrow \infty} x^{k+1} = \lim_{k \rightarrow \infty} \Phi(x^k) = \Phi(x^*).$$

Damit ist x^* Fixpunkt von Φ . Die Fehlerabschätzung folgt aus der obigen Abschätzung durch Grenzübergang $n \rightarrow \infty$. \square

Bemerkungen 9.1 (i) Aus der Fehlerabschätzung lässt sich bestimmen, wieviele Iterations schritte notwendig sind, um eine vorgegebene Fehlertoleranz zu erreichen.

(ii) Die Tatsache, dass das Verfahren für jede Wahl des Startwerts x^0 gegen die Lösung konvergiert, bezeichnet man als *globale Konvergenz*.

9.3 Lineare Iterationsverfahren

Wir wollen die Kontraktionseigenschaft für affin-lineare Abbildungen $\Phi(x) = Mx + s$ untersuchen. Offensichtlich ist die Abbildung Φ eine Kontraktion, wenn eine Operatornorm $\|\cdot\|_{op}$ auf $\mathbb{R}^{n \times n}$ existiert mit $\|M\|_{op} < 1$. Der *Spektralradius* einer Matrix $M \in \mathbb{R}^{n \times n}$ ist definiert durch

$$\rho(M) = \max\{|\lambda| : \lambda \text{ ist komplexer Eigenwert von } M\}.$$

Der folgende Satz zeigt, dass es ausreichend ist, $\rho(M) < 1$ zu zeigen, um eine Kontraktions-eigenschaft zu garantieren. Man beachte, dass $\rho(M)$ für $n \geq 2$ keine Norm auf $\mathbb{R}^{n \times n}$ definiert.

Satz 9.2 Für $M \in \mathbb{R}^{n \times n}$ gilt

$$\rho(M) = \inf \{\|M\|_{op} : \|\cdot\|_{op} \text{ ist induzierte Operatornorm auf } \mathbb{C}^{n \times n}\}.$$

Beweis

- (i) Sei $\lambda \in \mathbb{C}$ ein Eigenwert von M mit $\rho(M) = |\lambda|$ und $x \in \mathbb{C}^n \setminus \{0\}$ ein zugehöriger Eigenvektor. Dann gilt für jede Norm auf \mathbb{C}^n , dass

$$\rho(M)\|x\| = \|\lambda x\| = \|Mx\| \leq \|M\|_{op}\|x\|,$$

also $\rho(M) \leq \|M\|_{op}$.

- (ii) Die Matrix M ist komplex trigonalisierbar, das heißt es existieren eine invertierbare Matrix $T \in \mathbb{C}^{n \times n}$ und eine obere Dreiecksmatrix $R \in \mathbb{C}^{n \times n}$ mit

$$R = T^{-1}MT = \begin{bmatrix} \lambda_1 & r_{12} & \dots & r_{1n} \\ & \ddots & & \vdots \\ & & \lambda_{n-1} & r_{n-1,n} \\ & & & \lambda_n \end{bmatrix}$$

und den komplexen Eigenwerten $\lambda_1, \lambda_2, \dots, \lambda_n$ von M . Für $\varepsilon > 0$ sei $D_\varepsilon \in \mathbb{R}^{n \times n}$ die Diagonalmatrix mit Diagonalelementen $1, \varepsilon, \varepsilon^2, \dots, \varepsilon^{n-1}$. Dann wird durch

$$\|x\|_\varepsilon = \|D_\varepsilon^{-1}T^{-1}x\|_\infty$$

eine Norm auf \mathbb{C}^n definiert. Für die zugehörige Operatornorm gilt

$$\begin{aligned} \|M\|_\varepsilon &= \sup_{x \neq 0} \frac{\|D_\varepsilon^{-1}T^{-1}Mx\|_\infty}{\|D_\varepsilon^{-1}T^{-1}x\|_\infty} \stackrel{x=TD_\varepsilon y}{=} \sup_{y \neq 0} \frac{\|D_\varepsilon^{-1}T^{-1}MTD_\varepsilon y\|_\infty}{\|y\|_\infty} \\ &= \sup_{y \neq 0} \frac{\|D_\varepsilon^{-1}RD_\varepsilon y\|_\infty}{\|y\|_\infty} = \|D_\varepsilon^{-1}RD_\varepsilon\|_\infty \end{aligned}$$

mit der Zeilensummennorm $\|\cdot\|_\infty$. Direktes Nachrechnen zeigt

$$D_\varepsilon^{-1} R D_\varepsilon = \begin{bmatrix} \lambda_1 & \varepsilon r_{12} & \dots & \varepsilon^{n-1} r_{1n} \\ & \ddots & & \vdots \\ & & \lambda_{n-1} & \varepsilon r_{n-1,n} \\ & & & \lambda_n \end{bmatrix}$$

und damit folgt, sofern $\varepsilon \leq 1$ gilt,

$$\begin{aligned} \|M\|_\varepsilon &= \|D_\varepsilon^{-1} R D_\varepsilon\|_\infty = \max_{i=1,2,\dots,n} \left(|\lambda_i| + \sum_{j=i+1}^n \varepsilon^{j-i} |r_{ij}| \right) \\ &\leq \max_{i=1,2,\dots,n} |\lambda_i| + \varepsilon \|R\|_\infty = \rho(M) + \varepsilon \|R\|_\infty. \end{aligned}$$

Da $\varepsilon > 0$ beliebig klein gewählt werden kann, folgt die Behauptung. \square

Korollar 9.1 Gilt $\rho(M) < 1$, so ist die Abbildung $\Phi : x \mapsto Mx + b$ eine Kontraktion.

Beispiel 9.1 Das Richardson-Verfahren zur approximativen Lösung des linearen Gleichungssystems $Ax = b$ ist für $\omega > 0$ definiert durch $M = I_n - \omega A$ und $c = \omega b$, das heißt

$$x^{k+1} = Mx^k + c = x^k - \omega(Ax^k - b).$$

Ist A symmetrisch und positiv definit, so sind sämtliche Eigenwerte von A positiv, und für ω hinreichend klein folgt $\rho(I_n - \omega A) < 1$. Gilt $x^{k+1} = x^k$, so ist x^k Lösung von $Ax = b$.

9.4 Jacobi- und Gauß-Seidel-Verfahren

Basierend auf einfachen Zerlegungen von Matrizen lassen sich iterative Verfahren definieren.

Definition 9.2 Für $A \in \mathbb{R}^{n \times n}$ sind der untere, Diagonal- und obere Anteil $L, U, D \in \mathbb{R}^{n \times n}$ von A definiert durch

$$d_{ij} = \begin{cases} a_{ii}, & i = j, \\ 0, & i \neq j, \end{cases} \quad \ell_{ij} = \begin{cases} a_{ij}, & i > j, \\ 0, & i \leq j, \end{cases} \quad u_{ij} = \begin{cases} a_{ij}, & i < j, \\ 0, & i \geq j. \end{cases}$$

Da $A = L + D + U$ gilt, ist das Gleichungssystem $Ax = b$ äquivalent zu

$$Lx + Dx + Ux = b$$

und Iterationsverfahren lassen sich dadurch definieren, dass x in den verschiedenen Ter- men durch x^k oder x^{k+1} ersetzt wird, beispielsweise

$$Lx^k + Dx^{k+1} + Ux^k = b \iff x^{k+1} = -D^{-1}(A - D)x^k + D^{-1}b.$$

Für einen stationären Punkt beziehungsweise im Fall $x^{k+1} = x^k$ ist das Gleichungssystem offensichtlich erfüllt. Eine Alternative zu diesem Vorgehen ist

$$Lx^{k+1} + Dx^{k+1} + Ux^k = b \iff x^{k+1} = -(L + D)^{-1}Ux^k + (L + D)^{-1}b.$$

Definition 9.3 Das *Jacobi*- und *Gauß–Seidel*-Verfahren sind definiert durch

$$\begin{aligned} M^J &= -D^{-1}(A - D), & c^J &= D^{-1}b \\ M^{GS} &= -(L + D)^{-1}U, & c^{GS} &= (L + D)^{-1}b. \end{aligned}$$

Bemerkungen 9.2 (i) Im Jacobi-Verfahren ist in jedem Iterationsschritt ein lineares Gleichungssystem mit Diagonalmatrix und beim Gauß–Seidel-Verfahren mit unterer Dreiecksmatrix zu lösen.

(ii) Es ist zu erwarten, dass das Gauß–Seidel-Verfahren bessere Konvergenzeigenschaften als das Jacobi-Verfahren hat, da die Matrix $L + D$ in der Regel eine bessere Approximation von A ist als die Matrix D .

9.5 Diagonaldominanz und Irreduzibilität

Wir wollen hinreichende Bedingungen an eine Matrix formulieren, die die Kontraktions-eigenschaft eines Iterationsverfahrens implizieren.

Definition 9.4 Die Matrix $A \in \mathbb{R}^{n \times n}$ heißt *diagonaldominant*, falls für $i = 1, 2, \dots, n$ gilt

$$\sum_{j=1, \dots, n, j \neq i} |a_{ij}| \leq |a_{ii}|$$

und diese Ungleichung strikt ist für ein $i_0 \in \{1, 2, \dots, n\}$. Ist sie strikt für alle $i = 1, 2, \dots, n$, so heißt A *strikt diagonaldominant*.

Beispiel 9.2 Für die Matrizen

$$A_1 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}$$

gilt, dass A_1 diagonaldominant aber nicht strikt diagonaldominant und A_2 strikt diagonaldominant ist.

Bemerkungen 9.3 (i) Ist A strikt diagonaldominant, so gilt $a_{ii} \neq 0$ für $i = 1, 2, \dots, n$ und D ist regulär. Für die Iterationsmatrix $M^J = -D^{-1}(A - D)$ des zugehörigen Jacobi-Verfahrens gilt $m_{ii}^J = 0$ und somit aufgrund der Diagonaldominanz für $i = 1, 2, \dots, n$

$$\sum_{j=1}^n |m_{ij}^J| = \sum_{j=1, \dots, n, j \neq i} \frac{|a_{ij}|}{|a_{ii}|} = \frac{1}{|a_{ii}|} \sum_{j=1, \dots, n, j \neq i} |a_{ij}| < 1.$$

Das bedeutet $\|M^J\|_\infty < 1$ und somit $\rho(M^J) < 1$.

(ii) Für strikt diagonaldominante Matrizen gilt $\rho(M^{GS}) \leq \rho(M^J)$.

Strikte Diagonaldominanz ist im Allgemeinen eine zu einschränkende Bedingung.

Definition 9.5 Die Matrix $A \in \mathbb{R}^{n \times n}$ heißt *reduzibel*, falls disjunkte, nichtleere Indexmengen $I, J \subset \{1, 2, \dots, n\}$ existieren, sodass $I \cup J = \{1, 2, \dots, n\}$ und $a_{ij} = 0$ für alle Paare $(i, j) \in I \times J$. Andernfalls heißt A *irreduzibel*.

Beispiel 9.3 Die Matrix

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 4 & 5 \\ 6 & 0 & 7 \end{bmatrix}$$

ist reduzibel mit $I = \{1, 3\}$ und $J = \{2\}$.

Bemerkung 9.4 Für reduzible Matrizen lässt sich das Lösen des linearen Gleichungssystems $Ax = b$ zerlegen. Ist für $X, Y \subset \{1, 2, \dots, n\}$ die Teilmatrix A_{XY} definiert durch $A_{XY} = (a_{ij})_{i \in X, j \in Y}$ und der Teilvektor x_Y durch $x_Y = (x_k)_{k \in Y}$, so gilt $A_{II}x_I = b_I$ und $A_{JJ}x_J = b_J - A_{JI}x_I$.

Lemma 9.1 Ist M irreduzibel und diagonaldominant, so ist M regulär mit $m_{ii} \neq 0$ für $i = 1, 2, \dots, n$.

Beweis Ist M nicht regulär, so existiert $x \in \mathbb{R}^n \setminus \{0\}$ mit $Mx = 0$ und aus der i -ten Zeile der Identität folgt

$$|m_{ii}x_i| \leq \sum_{j=1, \dots, n, j \neq i} |m_{ij}| |x_j|.$$

Definiere $I = \{i : |x_i| = \|x\|_\infty\}$ und $J = \{j : |x_j| < \|x\|_\infty\}$. Dann gilt $I \neq \emptyset$ und $I \cup J = \{1, 2, \dots, n\}$. Es gilt auch $J \neq \emptyset$, denn andernfalls würde $|x_j| = \|x\|_\infty$ für $j = 1, 2, \dots, n$ und damit

$$|m_{ii}| \leq \sum_{j=1, \dots, n, j \neq i} |m_{ij}|$$

gelten, was der Diagonaldominanz, die strikte Ungleichheit in umgekehrter Richtung für ein i_0 garantiert, widerspricht. Folglich gilt $J \neq \emptyset$ und aufgrund der Irreduzibilität existieren $i \in I$ und $j \in J$ mit $m_{ij} \neq 0$ und somit

$$|m_{ii}| \leq \sum_{j=1, \dots, n, j \neq i} |m_{ij}| \frac{|x_j|}{\|x\|_\infty} < \sum_{j=1, \dots, n, j \neq i} |m_{ij}|$$

im Widerspruch zur Diagonaldominanz von M . Folglich ist M regulär. Die Regularität und die Diagonaldominanz von M implizieren $m_{ii} \neq 0$ für $i = 1, 2, \dots, n$, denn andernfalls wäre eine Zeile von M identisch Null, was der Regularität von M widersprechen würde. \square

9.6 Konvergenz

Das vorangegangene Lemma erlaubt es, die Konvergenz des Jacobi- und Gauß-Seidel-Verfahrens zu beweisen.

Satz 9.3 Ist A irreduzibel und diagonaldominant, so sind Jacobi- und Gauß-Seidel-Verfahren durchführbar und konvergent, das heißt M^J und M^{GS} sind wohldefiniert und erfüllen $\rho(M^J) < 1$ und $\rho(M^{GS}) < 1$.

Beweis

- (i) Nach dem vorigen Lemma gilt $a_{ii} \neq 0$ für $i = 1, 2, \dots, n$ und somit ist $M^J = -D^{-1}(A - D)$ wohldefiniert. Wir zeigen, dass $M^J - \mu I_n$ für alle $\mu \in \mathbb{C}$ mit $|\mu| \geq 1$ regulär ist, sodass $\rho(M^J) < 1$ folgt. Da Irreduzibilität unabhängig ist von den Diagonalelementen einer Matrix sind mit A auch $A - D$ und $M^J = -D^{-1}(A - D)$ irreduzibel. Ebenso ist $M = M^J - \mu I_n$ irreduzibel. Mit der Diagonaldominanz von A folgt für $i = 1, 2, \dots, n$, dass

$$\sum_{j=1, \dots, n, j \neq i} |m_{ij}| = \sum_{j=1, \dots, n, j \neq i} |m_{ij}^J| = \sum_{j=1, \dots, n, j \neq i} \frac{|a_{ij}|}{|a_{ii}|} \leq 1 \leq |\mu| = |m_{ii}|,$$

wobei die Ungleichung strikt ist für ein $i_0 \in \{1, 2, \dots, n\}$. Folglich ist M diagonaldominant für jedes $\mu \in \mathbb{C}$ mit $|\mu| \geq 1$ und zusammen mit der Irreduzibilität folgt die Regularität von M .

- (ii) Wiederum impliziert $a_{ii} \neq 0$ für $i = 1, 2, \dots, n$, dass $M^{GS} = -(L + D)^{-1}U$ wohldefiniert ist. Für $\mu \in \mathbb{C}$ mit $|\mu| \geq 1$ sei $M = M^{GS} - \mu I_n$. Da $L + D$ regulär ist, ist M genau dann regulär, wenn

$$\tilde{M} = -(L + D)M = -(L + D)(-(L + D)^{-1}U - \mu I_n) = U + \mu L + \mu D$$

regulär ist. Mit $A = U + L + D$ ist auch \tilde{M} irreduzibel. Ferner ist \tilde{M} diagonaldominant, denn für $i = 1, 2, \dots, n$ gilt aufgrund der Diagonaldominanz von A , dass

$$\begin{aligned} \sum_{j=1, \dots, n, j \neq i} |\tilde{m}_{ij}| &= |\mu| \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^n |a_{ij}| \leq |\mu| \sum_{j=1, \dots, n, j \neq i} |a_{ij}| \\ &\leq |\mu| |a_{ii}| = |\tilde{m}_{ii}|, \end{aligned}$$

wobei strikte Ungleichung für ein $i_0 \in \{1, 2, \dots, n\}$ gilt. Also ist \tilde{M} diagonaldominant und das vorangegangene Lemma impliziert die Regularität von \tilde{M} . \square

Bemerkungen 9.5 (i) Im Fall der Konvergenz führen häufig wenige Iterationsschritte zu einer guten Näherungslösung. Da jeder Iterationsschritt im Jacobi- und Gauß-Seidel-Verfahren $\mathcal{O}(n^2)$ viele Operation erfordert, kann damit der typische Aufwand $\mathcal{O}(n^3)$ direkter Lösungsmethoden wie der Gauß-Elimination verringert werden.

(ii) Beide Voraussetzungen des Satzes werden benötigt, um eine Kontraktionseigenschaft der Iterationsmatrizen zu garantieren, wie das Beispiel

$$A = \begin{bmatrix} 1/2 & 1/2 \\ 0 & 1 \end{bmatrix}, \quad M^J = -D^{-1}(A - D) = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

zeigt.

9.7 Lernziele, Quiz und Anwendung

Sie sollten iterative Verfahren zur Lösung linearer Gleichungssysteme herleiten und deren Vorteile im Vergleich zu anderen Methoden aufzeigen können. Hinreichende Bedingungen für die Konvergenz linearer Iterationsverfahren sollten Sie benennen können. Strukturelle Eigenschaften von Matrizen, die die Konvergenz der Verfahren sicher stellen, sollten Sie erklären und deren Bedeutung veranschaulichen können.

Quiz 9.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Ist $A \in \mathbb{R}^{n \times n}$ irreduzibel und $D \in \mathbb{R}^{n \times n}$ eine Diagonalmatrix, so ist auch $A - D$ irreduzibel	
Ist $A \in \mathbb{R}^{n \times n}$ diagonaldominant, so ist A regulär	
Für symmetrische Matrizen stimmen Jacobi- und Gauß-Seidel-Verfahren überein	
Die Eigenschaft $a_{ii} \neq 0$ einer Matrix $A \in \mathbb{R}^{n \times n}$ ist notwendig für die Wohldefiniertheit des Jacobi- und des Gauß-Seidel-Verfahrens	
Ist $A - \mu I_n$ regulär für alle $\mu \in \mathbb{C}$ mit $ \mu \geq 1/4$, so gilt $ \lambda < 1/4$ für alle Eigenwerte von A	

Anwendung 9.1 In Anwendungen wie der Beschreibung des elastischen Verhaltens von Stabtragwerken treten reguläre Matrizen $A \in \mathbb{R}^{n \times n}$ auf, bei denen sehr viele Einträge verschwinden. In diesen Fällen ist es häufig sinnvoll, ein iteratives Verfahren zu implementieren, ohne die Matrix A komplett abzuspeichern. Zeigen Sie, dass sich das Jacobi- und das Gauß-Seidel-Verfahren in der Form

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

beziehungsweise

$$x_i^{k+1} = a_{ii}^{-1} \left(b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k \right)$$

für $i = 1, 2, \dots, n$ schreiben lässt. Vereinfachen Sie diese Formeln für den Fall von Matrizen mit endlicher *Bandweite* $w > 0$, das heißt für den Fall, dass $a_{ij} = 0$ für $|i - j| > w$ gilt.

Teil II

Numerische Analysis

10.1 Konditionierung

Wir betrachten die Auswirkungen von Störungen bei der Auswertung einer mathematischen Aufgabe $\phi(x)$, die durch eine Abbildung $\phi : X \rightarrow Y$ zwischen normierten Vektorräumen definiert ist. Dabei werden Störungen \tilde{x} von x additiv als Summe $\tilde{x} = x + \Delta x$ mit $\Delta x = \tilde{x} - x$ dargestellt. Die folgende Definition aus [2] verallgemeinert den Begriff der Konditionszahl für allgemeine mathematische Aufgaben.

Definition 10.1 Die (*relative*) *Konditionszahl* $\kappa_\phi(x)$ der Funktion $\phi : X \rightarrow Y$ bei $x \neq 0$ mit $\phi(x) \neq 0$ ist das Infimum aller $\kappa \geq 0$, für die ein $\delta > 0$ existiert, sodass

$$\frac{\|\phi(x + \Delta x) - \phi(x)\|}{\|\phi(x)\|} \leq \kappa \frac{\|\Delta x\|}{\|x\|}$$

für alle $\Delta x \in X$ mit $\|\Delta x\|/\|x\| \leq \delta$ gilt. Die Aufgabe $\phi(x)$ heißt *schlecht konditioniert*, falls $\kappa_\phi(x) \gg 1$ gilt, und *schlecht gestellt*, falls $\kappa_\phi(x)$ nicht definiert ist.

Beispiele 10.1 (i) Die durch $\phi(x) = |x|^s$ definierte Aufgabe ist gut konditioniert genau dann, wenn $s \geq 1$ gilt.

(ii) Die Aufgabe $\phi(x) = \text{sign}(x)$ ist schlecht gestellt bei $x = 0$.

Satz 10.1 Ist ϕ differenzierbar bei x , so gilt

$$\kappa_\phi(x) = \frac{\|D\phi(x)\| \|x\|}{\|\phi(x)\|}.$$

Beweis Es gilt

$$\phi(x + \Delta x) - \phi(x) = D\phi(x)[\Delta x] + \psi(\Delta x),$$

mit einer Funktion ψ , die $\psi(\Delta x)/\|\Delta x\| \rightarrow 0$ für $\Delta x \rightarrow 0$ erfüllt. Damit existiert für jedes $\varepsilon > 0$ ein $\delta > 0$, sodass für alle Δx mit $\|\Delta x\|/\|x\| \leq \delta$ gilt

$$\left\| \frac{\phi(x + \Delta x) - \phi(x)}{\|\Delta x\|} - \frac{D\phi(x)[\Delta x]}{\|\Delta x\|} \right\| \leq \varepsilon.$$

Daraus folgt

$$\frac{\|\phi(x + \Delta x) - \phi(x)\|}{\|\phi(x)\|} \leq \left(\varepsilon + \frac{\|D\phi(x)[\Delta x]\|}{\|\Delta x\|} \right) \frac{\|\Delta x\|}{\|\phi(x)\|}.$$

Nach Definition der Operatornorm gilt $\|D\phi(x)[\Delta x]\| \leq \|D\phi(x)\| \|\Delta x\|$, wobei Gleichheit für geeignete Δx eintritt. Da $\varepsilon > 0$ beliebig klein gewählt werden kann, folgt die Behauptung. \square

Im Fall linearer Gleichungssysteme ist die Konditionszahl durch die Kondition der Matrix beschränkt.

Bemerkungen 10.1 (i) Für $\phi(b) = A^{-1}b$ gilt $D\phi(b) = A^{-1}$ und mit der Identität $\|b\| = \|A(A^{-1}b)\| \leq \|A\| \|A^{-1}b\|$ folgt, dass

$$\kappa_\phi(b) = \frac{\|A^{-1}\|}{\|A^{-1}b\|} \|b\| \leq \|A^{-1}\| \|A\| = \text{cond}(A).$$

Ferner existiert ein $b \in \mathbb{R}^n$, sodass Gleichheit gilt.

(ii) Um Einflüsse von Störungen der Matrix A zu untersuchen, betrachten wir die Abbildung $\phi(A) = A^{-1}b$. Aus der Konstanz von $A \mapsto A\phi(A) = b$ folgt $D\phi(A)[E] = -A^{-1}EA^{-1}b$ und mit der Abschätzung $\|D\phi(A)\| \leq \|A^{-1}\| \|A^{-1}b\|$ ergibt sich

$$\kappa_\phi(A) \leq \frac{\|A^{-1}\| \|A^{-1}b\| \|A\|}{\|A^{-1}b\|} = \text{cond}(A),$$

das heißt Fehler in A werden ebenfalls mit dem Faktor $\text{cond}(A)$ verstärkt.

Auch Auslöschungseffekte werden von der Konditionszahl erfasst.

Beispiele 10.2 (i) Für $\phi(x_1, x_2) = x_1 + x_2$ gilt $D\phi(x_1, x_2) = [1, 1]$ und somit

$$\kappa_\phi(x_1, x_2) = \frac{\|[1, 1]\|_1 \|(x_1, x_2)\|_1}{|x_1 + x_2|} = \frac{|x_1| + |x_2|}{|x_1 + x_2|},$$

sodass die Aufgabe schlecht konditioniert ist, falls $x_1 \approx -x_2$ gilt, das heißt wenn Auslöschungseffekte auftreten können.

(ii) Anschaulich ist das senkrechte Aufstellen eines Stifts ein schlecht konditioniertes Problem, während das Aufstellen einer Dose im Allgemeinen gut konditioniert ist.

10.2 Gleitkommazahlen

Auf digitalen Rechnern stehen nur endlich viele Zahlen zur Verfügung, die in der Regel wie folgt definiert sind. Wir folgen der Darstellung in [7].

Definition 10.2 Für eine Basis $b \geq 2$, eine Präzision $p \geq 1$ und Exponentenschränken $e_{\min} \leq e_{\max}$ mit $b, p, e_{\min}, e_{\max} \in \mathbb{Z}$ ist die Menge der *Gleitkommazahlen* oder *Maschinenzahlen* definiert durch

$$G = \{ \pm mb^{e-p} : m, e \in \mathbb{Z}, 0 \leq m \leq b^p - 1, e_{\min} \leq e \leq e_{\max} \}.$$

Eine Gleitkommazahl $g \in G$ heißt *normalisiert*, falls $m \geq b^{p-1}$ gilt. In den Fällen $b = 2$ beziehungsweise $b = 10$ spricht man vom *Dual-* und *Dezimalsystem*.

Beispiel 10.3 Für $b = 10$, $p = 3$, $e_{\min} = -2$ und $e_{\max} = 2$ besteht G aus allen Zahlen der Form $\pm m \cdot 10^{-r}$ mit $0 \leq m \leq 999$ und $1 \leq r \leq 5$, beispielsweise

$$-783 \cdot 10^{-5}, \quad 400 \cdot 10^{-3}, \quad 40 \cdot 10^{-2},$$

wobei nur die ersten beiden Zahlen normalisiert sind.

Bemerkungen 10.2 (i) Jede Gleitkommazahl $g \in G$ lässt sich als b -adische Summe darstellen, das heißt es gilt

$$g = \pm b^e (d_1 b^{-1} + d_2 b^{-2} + \cdots + d_p b^{-p})$$

mit Ziffern $d_1, d_2, \dots, d_p \in \{0, 1, \dots, b-1\}$ und $e_{\min} \leq e \leq e_{\max}$. Für normalisierte Gleitkommazahlen ist diese Darstellung eindeutig definiert mit $d_1 \neq 0$.

(ii) Für alle $g \in G$ gilt $b^{e_{\min}-1} \leq |g| \leq b^{e_{\max}}(1 - b^{-p})$.

(iii) Für $b = 10$ gilt $g = \pm 10^e \cdot 0.d_1d_2\dots d_p$ und das Komma beziehungsweise der Punkt ist von e abhängig gleitend.

Beispiel 10.4 Im Standard 754R des *Institute of Electrical and Electronics Engineers* (IEEE) sind die Formate *single* beziehungsweise *double precision* definiert durch

$$\begin{array}{llll} b = 2, & e_{\min} = -125, & e_{\max} = 128, & p = 24, \\ b = 2, & e_{\min} = -1021, & e_{\max} = 1024, & p = 53. \end{array}$$

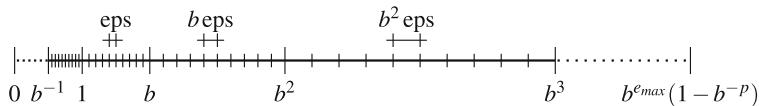


Abb. 10.1 Schematische Darstellung der Anordnung von Maschinenzahlen

Der relative Fehler bei der Approximation reeller Zahlen durch Maschinenzahlen ist beschränkt durch die sogenannte *Maschinengenauigkeit*.

Definition 10.3 Die *Maschinengenauigkeit* ist mit der kleinsten Zahl $g_{\text{eps}} \in G$, für die $g_{\text{eps}} > 1$ gilt, definiert durch $\text{eps} = g_{\text{eps}} - 1 = \min_{g>1} g - 1$

Bemerkung 10.3 Es gilt $g_{\text{eps}} = b^1(b^{-1} + 0 \cdot b^{-2} + \dots + 0 \cdot b^{-p+1} + b^{-p}) = 1 + b^{1-p}$ und somit $\text{eps} = b^{1-p}$.

Beispiele 10.5 (i) Die Maschinenzahlen zwischen b^e und b^{e+1} sind gleichmäßig in einem Abstand von $b^e \text{eps}$ angeordnet, s. Abb. 10.1.

(ii) Für die IEEE-754R-Formate *single* und *double* gilt $\text{eps} = 2^{-23} \approx 1.2 \cdot 10^{-7}$ beziehungsweise $\text{eps} = 2^{-52} \approx 2.2 \cdot 10^{-16}$.

10.3 Rundung

Rundungsabbildungen approximieren reelle Zahlen durch Maschinenzahlen.

Definition 10.4 Für eine Menge von Maschinenzahlen G heißt eine Abbildung $\text{rd} : [-g_{\text{max}}, g_{\text{max}}] \rightarrow G$ *Rundungsabbildung*, falls für jede reelle Zahl $x \in [-g_{\text{max}}, g_{\text{max}}]$ gilt, dass $|\text{rd}(x) - x| = \min_{g \in G} |x - g|$.

Bemerkungen 10.4 (i) Liegt x genau zwischen zwei Maschinenzahlen, so wird in IEEE-Standards die Maschinenzahl ausgewählt, deren letzte Ziffer gerade ist.

(ii) Man spricht von *Overflow* und *Underflow*, wenn $|x| > g_{\text{max}}$ beziehungsweise $|x| < g_{\text{min}}$ gilt. Im zweiten Fall wird in der Regel auf Null gerundet, wobei jedoch ein großer Fehler auftritt. Im denormalisierten IEEE Standard werden weitere Maschinenzahlen in einer Umgebung der Null verwendet.

(iii) Zusätzlich zu den Zahlen in G gibt es meist noch den Wert *NaN*, der beispielsweise für undefinierte Ausdrücke verwendet wird und für *Not-a-Number* steht.

Lemma 10.1 Für jedes $x \in \mathbb{R}$ mit $|x| \in [g_{\text{min}}, g_{\text{max}}]$ gilt

$$\frac{|x - \text{rd}(x)|}{|x|} \leq \frac{1}{2} \text{eps},$$

das heißt es existiert ein $\delta \in \mathbb{R}$ mit $|\delta| \leq \text{eps}/2$ und $\text{rd}(x) = (1 + \delta)x$.

Beweis Da die Maschinenzahlen in jedem Intervall $[b^e, b^{e+1}]$ gleichmäßig im Abstand $b^e \text{eps}$ angeordnet sind, existiert ein $\ell \geq 0$ mit $b^e + \ell b^e \text{eps} \leq x \leq b^e + (\ell + 1)b^e \text{eps}$ und es sei g die obere oder untere Schranke mit $|x - g| \leq (1/2)b^e \text{eps}$. Da $|x| \geq b^e$ folgt die Behauptung. \square

Definition 10.5 Das *Standardmodell der Gleitkommaarithmetik* fordert, dass für alle $x, y \in \mathbb{R}$ mit $|x|, |y| \in [g_{\min}, g_{\max}]$ und jede arithmetische Standardoperation $\text{op} \in \{+, -, *, :\}$ mit $|x \text{ op } y| \in [g_{\min}, g_{\max}]$ sowie deren numerische Realisierung $\text{op}_G : G \times G \rightarrow G$ ein $\delta \in \mathbb{R}$ mit $|\delta| \leq \text{eps}/2$ existiert, sodass

$$\text{rd}(x) \text{ op}_G \text{ rd}(y) = (x \text{ op } y)(1 + \delta).$$

Bemerkungen 10.5 (i) Es wird häufig weiter vereinfachend angenommen, dass $\text{rd}(x) \text{ op}_G \text{ rd}(y) = \text{rd}(x \text{ op } y)$ gilt.

(ii) Das Standardmodell wird von den IEEE-Standards erfüllt, die auf gängigen Rechnern realisiert sind.

(iii) Bei vielen Operationen können sich Rundungsfehler akkumulieren und relevant werden. Man spricht in diesem Fall auch von Fehlerfortpflanzung.

10.4 Stabilität

Es bezeichne $\tilde{\phi} : X \rightarrow Y$ ein numerisches Verfahren, das heißt eine endliche Folge rundungsfehlerbehafteter Grundoperationen. Bei Rundung des Arguments x gilt

$$\phi(x) - \tilde{\phi}(x + \Delta x) = (\phi(x) - \phi(x + \Delta x)) + (\phi(x + \Delta x) - \tilde{\phi}(x + \Delta x)),$$

wobei der erste Term auf der rechten Seite durch die Konditionierung von ϕ kontrolliert wird und der zweite die Stabilität des Verfahrens beschreibt. Letztere ist abhängig von der theoretisch frei wählbaren Rundungsgenauigkeit ε und wir schreiben im Folgenden auch $\tilde{\phi}_\varepsilon$ statt $\tilde{\phi}$, um dies zu kennzeichnen. Die folgende Definition aus [2] präzisiert diesen Ansatz.

Definition 10.6 Der *Stabilitätsindikator* $\sigma_{\tilde{\phi}}(x)$ des numerischen Verfahrens $\tilde{\phi}$ ist das Infimum aller $\sigma \geq 0$ für die ein $\delta > 0$ existiert, sodass

$$\frac{\|\phi(x) - \tilde{\phi}_\varepsilon(x)\|}{\|\phi(x)\|} \leq \sigma \kappa_\phi(x) \varepsilon$$

für jedes $0 \leq \varepsilon \leq \delta$ gilt. Das Verfahren $\tilde{\phi}$ heißt *instabil*, falls $\sigma_{\tilde{\phi}}(x) \gg 1$ gilt. Andernfalls heißt das Verfahren (*vorwärts-)**stabil*.

Genaue Stabilitätsanalysen sind im Allgemeinen äußerst aufwendig. Die folgenden Konzepte werden üblicherweise in der Praxis angewendet.

Bemerkung 10.6 Bei der *linearen Vorwärtsanalyse* wird jedes Zwischenergebnis z_i als rundungsbehaftet angesehen und durch $(1 + \varepsilon_i)z_i$ mit $|\varepsilon_i| \leq \varepsilon$ ersetzt. Produkte der Form $\varepsilon_i \varepsilon_j$ werden in der Rechnung vernachlässigt. Die Division wird bezüglich ε linearisiert, das heißt beispielsweise

$$(x(1 + \varepsilon))^{-1} \approx (1 - \varepsilon)x^{-1}.$$

Ein einfacher zu prüfendes, aber sehr einschränkendes Stabilitätskriterium ist die sogenannte Rückwärtsstabilität.

Definition 10.7 Der *Rückwärtsstabilitätsindikator* $\rho_{\tilde{\phi}}(x)$ einer Operation $\tilde{\phi}_\varepsilon : X \rightarrow Y$ bei x ist das Infimum aller $\rho \geq 0$, für die ein $\delta > 0$ existiert, sodass es für alle $0 \leq \varepsilon \leq \delta$ ein $\Delta x \in X$ gibt mit $\phi(x + \Delta x) = \tilde{\phi}_\varepsilon(x)$ und

$$\frac{\|\phi^{-1}(\phi(x)) - \phi^{-1}(\tilde{\phi}_\varepsilon(x))\|}{\|\phi^{-1}(\phi(x))\|} = \frac{\|\Delta x\|}{\|x\|} \leq \rho\varepsilon.$$

Das Verfahren heißt *rückwärtsstabil*, sofern nicht $\rho_{\tilde{\phi}}(x) \gg 1$ gilt.

Bemerkung 10.7 Ist $\tilde{\phi}_\varepsilon$ rückwärtsstabil, so ist $\tilde{\phi}_\varepsilon$ stabil, denn es gilt

$$\frac{1}{\kappa_\phi(x)} \frac{\|\tilde{\phi}_\varepsilon(x) - \phi(x)\|}{\varepsilon \|\phi(x)\|} = \frac{1}{\kappa_\phi(x)} \frac{\|\phi(x + \Delta x) - \phi(x)\|}{\varepsilon \|\phi(x)\|} \leq \frac{\|\Delta x\|}{\varepsilon \|x\|}$$

und daher $\sigma_{\tilde{\phi}}(x) \leq \rho_{\tilde{\phi}}(x)$.

Beispiele 10.6 (i) Die Gleitkomma-Realisierung der Aufgabe $\phi(x) = 1 + x$ ist nicht rückwärtsstabil für kleine Zahlen x , denn es gilt

$$|\phi^{-1}(1 + x + \Delta x) - \phi^{-1}(1 + x)| / |\phi^{-1}(1 + x)| = |\Delta x| / |x| \gg |\Delta x|.$$

Offensichtlich ist $\tilde{\phi} = \phi$ jedoch stabil für kleine Zahlen x .

(ii) Die Cramersche Regel ist nicht rückwärtsstabil aber vorwärtsstabil für lineare Gleichungssysteme der Dimension 2.

10.5 Lernziele, Quiz und Anwendung

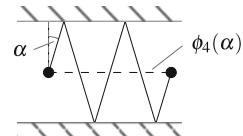
Sie sollten die allgemeine Konditionszahl einer mathematischen Aufgabe begreiflich machen und an Beispielen illustrieren können. Die Bedeutung von Gleitkommazahlen und deren Genauigkeit für numerische Berechnungen sollten Sie verdeutlichen können. Den Begriff des Stabilitätsindikators eines numerischen Verfahrens sollten Sie präzisieren können.

Quiz 10.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Ist $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ Lipschitz-stetig und gilt $\ \phi(x)\ \geq 1$ für alle $x \in \mathbb{R}^n$, so ist ϕ gut konditioniert	
Für $b = 10$, $p = 4$, $e_{\min} = -3$, $e_{\max} = 3$ ist $-13 \cdot 10^{-2}$ eine normalisierte Gleitkommazahl	
Gilt $\text{rd}(x) = 0$, so folgt $ x < \text{eps}$	
Die Maschinengenauigkeit eps beschränkt den absoluten Fehler bei der Approximation reeller Zahlen durch Gleitkommazahlen	
Ist ϕ schlecht konditioniert, so ist jedes numerische Verfahren $\tilde{\phi}$ stabil	

Anwendung 10.1 Auf einem Billiardtisch der Breite 1 werde eine sich auf der Mittellinie befindende Kugel mit dem Winkel $\alpha \in (0, \pi/2)$ angestoßen. Für eine fixierte Zahl $n \in \mathbb{N}$ sei $\phi_n(\alpha)$ der Abstand zur Ausgangsposition auf der Mittellinie, mit dem die Kugel diese nach n Bandenberührungen überquert, s. Abb. 10.2. Leiten Sie eine Formel für $\phi_n(\alpha)$ her, bestimmen Sie die Konditionszahl $\kappa_{\phi_n}(\alpha)$ und interpretieren Sie diese.

Abb. 10.2 Konditionierungsuntersuchung eines Billiard-Stoßes



11.1 Lagrange-Interpolation

Als Interpolation bezeichnet man die Approximation einer gegebenen Funktion in einem endlich-dimensionalen Raum von Funktionen wie beispielsweise Polynomen beschränkten Grades. Da nur die Koeffizienten bezüglich einer Basis abgespeichert werden müssen, ist dies für die numerische Weiterverarbeitung oder tabellarische Erfassung einer Funktion vorteilhaft. Im Folgenden bezeichne

$$\mathcal{P}_n = \left\{ \sum_{i=0}^n a_i x^i : a_0, a_1, \dots, a_n \in \mathbb{R} \right\}$$

den Raum der Polynome vom maximalen Grad $n \in \mathbb{N}_0$.

Bemerkung 11.1 Es gilt $\dim \mathcal{P}_n = n + 1$ und die Monome (x^0, x^1, \dots, x^n) bilden eine Basis von \mathcal{P}_n .

Definition 11.1 Die *Lagrange-Interpolationsaufgabe* sucht für gegebene, paarweise verschiedene *Stützstellen* (oder *Knoten*) $a \leq x_0 < x_1 < \dots < x_n \leq b$ und gegebene *Stützwerte* y_0, y_1, \dots, y_n ein Polynom $p \in \mathcal{P}_n$ mit $p(x_i) = y_i$ für $i = 0, 1, \dots, n$, s. Abb. 11.1.

Die Interpolationsaufgabe lässt sich direkt mit einer speziellen Basis von \mathcal{P}_n lösen.

Abb. 11.1 Lagrange-Interpolationsaufgabe

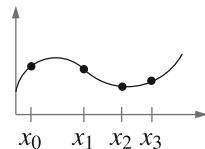
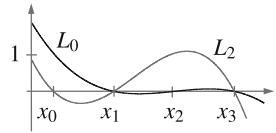


Abb. 11.2 Mit den Lagrange-Polynomen L_i lässt sich die Interpolationsaufgabe lösen



Definition 11.2 Die den Stützstellen $x_0 < x_1 < \dots < x_n$ zugeordneten Lagrange-Polynome $L_0, L_1, \dots, L_n \in \mathcal{P}_n$ sind definiert durch

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} = \frac{(x - x_1)}{(x_i - x_1)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)}.$$

Bemerkung 11.2 Es gilt $L_i(x_j) = \delta_{ij}$ für alle $0 \leq i, j \leq n$, s. Abb. 11.2.

Satz 11.1 Die Lagrange-Interpolationsaufgabe wird eindeutig durch

$$p = \sum_{i=0}^n y_i L_i$$

gelöst. Dieses Polynom wird als (Lagrange-)Interpolationspolynom bezeichnet.

Beweis Aus $L_i(x_j) = \delta_{ij}$ folgt $p(x_j) = y_j$ für $j = 0, 1, \dots, n$, das heißt p ist eine Lösung. Ist $q \in \mathcal{P}_n$ eine weitere Lösung, so gilt für $r = p - q \in \mathcal{P}_n$, dass $r(x_j) = 0$ für $j = 0, 1, \dots, n$, das heißt r hat $n + 1$ Nullstellen woraus $r = 0$ und somit $p = q$ folgt.

□

Bemerkung 11.3 Ist (q_0, q_1, \dots, q_n) eine Basis von \mathcal{P}_n , so ist die Lösung der Lagrange-Interpolationsaufgabe darstellbar als Linearkombination $p = \sum_{i=0}^n c_i q_i$, wobei der Koeffizientenvektor $c = [c_0, \dots, c_n]^\top$ das reguläre lineare Gleichungssystem $Vc = y$ mit $y = [y_0, y_1, \dots, y_n]^\top$ und der Vandermonde-Matrix $V \in \mathbb{R}^{(n+1) \times (n+1)}$ mit Einträgen $v_{ij} = q_i(x_j)$ löst. Für die Wahl der Lagrange-Polynome folgt $V = I_n$. Wählt man hingegen die Monombasis (x^0, x^1, \dots, x^n) , so ist V im Allgemeinen schlecht konditioniert.

11.2 Interpolationsfehler

Häufig repräsentieren die Werte y_0, y_1, \dots, y_n Funktionswerte einer Funktion f und man ist an der Größe des Fehlers $f - p$ interessiert.

Satz 11.2 Sei $f \in C^{n+1}([a, b])$ und es gelte $f(x_i) = y_i$ für $i = 0, 1, \dots, n$. Für die Lösung $p \in \mathcal{P}_n$ der Lagrange-Interpolationsaufgabe und jedes $x \in [a, b]$ existiert dann ein $\xi \in [a, b]$, sodass gilt

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x - x_j).$$

Beweis Sei $x \in [a, b]$. Gilt $x \in \{x_0, x_1, \dots, x_n\}$, so ist die Aussage klar und es gelte $x \neq x_i$ für $i = 0, 1, \dots, n$. Mit dem Stützstellenpolynom

$$w(y) = \prod_{j=0}^n (y - x_j) = y^{n+1} + a_n y^n + \dots + a_0 \in \mathcal{P}_{n+1}$$

für $y \in [a, b]$ sei

$$F(y) = (f(x) - p(x)) w(y) - (f(y) - p(y)) w(x).$$

Dann gilt $F(x_i) = 0$ für $i = 0, 1, \dots, n$ sowie $F(x) = 0$, das heißt F hat mindestens $n+2$ verschiedene Nullstellen. Nach dem Satz von Rolle hat F' zwischen zwei Nullstellen von F eine Nullstelle, das heißt F' hat mindestens $n+1$ verschiedene Nullstellen. Die wiederholte Anwendung dieses Arguments zeigt, dass die Ableitung $F^{(n+1)}$ mindestens eine Nullstelle $\xi \in [a, b]$ besitzt. Damit folgt

$$0 = F^{(n+1)}(\xi) = (f(x) - p(x))(n+1)! - f^{(n+1)}(\xi)w(x)$$

und dies ist die behauptete Identität. \square

Korollar 11.1 Für den Interpolationsfehler gilt

$$\|f - p\|_{C^0([a,b])} \leq \frac{\|f^{(n+1)}\|_{C^0([a,b])}}{(n+1)!} (b-a)^{n+1}.$$

Das Korollar impliziert, dass die Lagrange-Interpolationspolynome für $n \rightarrow \infty$ gleichmäßig gegen f konvergieren, sofern der Abstand $b - a$ verringert oder die Anzahl der Stützstellen erhöht wird und dabei die Ableitungen von f nicht zu rasch wachsen. Letzteres ist im Allgemeinen aber nicht der Fall.

Beispiel 11.1 Sei $f : [-1, 1] \rightarrow \mathbb{R}$ definiert durch $f(x) = (1 + 25x^2)^{-1}$ und seien die Stützstellen äquidistant gewählt, das heißt $x_i = -1 + 2i/n$ für $i = 0, 1, \dots, n$. Dann konvergiert die Folge der Lagrange-Interpolationspolynome $(p_n)_{n \in \mathbb{N}}$ von f nicht punktweise gegen f für $n \rightarrow \infty$, da der Ausdruck $\|f^{(n+1)}\|_{C^0([-1,1])}$ zu rasch wächst. Das Interpolationspolynom ist in Abb. 11.6 gezeigt.

11.3 Neville-Schema

Die direkte Auswertung des Interpolationspolynoms an einer Stelle $x \in [a, b]$ ist aufwendig und unter Umständen instabil. Das Neville-Schema erlaubt eine Berechnung von $p(x)$ mit $\mathcal{O}(n^2)$ Rechenoperationen. Wir folgen den Darstellungen in [7,8].

Definition 11.3 Für $n + 1$ Stützstellen und -werte $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ sowie $0 \leq j \leq n$ und $0 \leq i \leq n - j$ sei $p_{i,j} \in \mathcal{P}_j$ das eindeutig bestimmte Lagrange-Interpolationspolynom mit $p_{i,j}(x_k) = y_k, k = i, i + 1, \dots, i + j$, s. Abb. 11.3.

Bemerkung 11.4 Es gilt $p_{i,0}(x) = y_i$ für $i = 0, 1, \dots, n$ und $p_{0,n}(x) = p(x)$ für $x \in [a, b]$.

Satz 11.3 Mit der Initialisierung $p_{i,0}(x) = y_i$ für $i = 0, 1, \dots, n$ gilt

$$p_{i,j}(x) = \frac{(x - x_i)p_{i+1,j-1}(x) - (x - x_{i+j})p_{i,j-1}(x)}{x_{i+j} - x_i}$$

für $i = 0, 1, \dots, n - j$.

Beweis Für $j = 0$ ist die Aussage offensichtlich richtig und sie gelte für $j - 1$. Sei $0 \leq i \leq n - j$ und es bezeichne $q(x)$ die rechte Seite der behaupteten Identität für $p_{i,j}$. Wegen $p_{i+1,j-1}, p_{i,j-1} \in \mathcal{P}_{j-1}$ gilt $q \in \mathcal{P}_j$. Zudem gilt $q(x_i) = p_{i,j-1}(x_i) = y_i$ sowie $q(x_{i+j}) = p_{i+1,j-1}(x_{i+j}) = y_{i+j}$. Für $k = i + 1, i + 2, \dots, i + j - 1$ gilt wegen

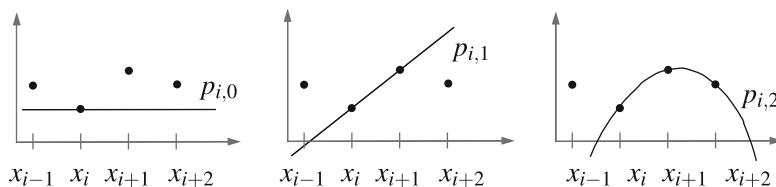


Abb. 11.3 Im Neville-Schema wird das Interpolationspolynom schrittweise konstruiert

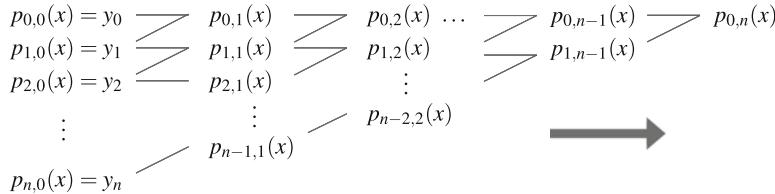


Abb. 11.4 Schematische Darstellung des Neville-Schemas; die Auswertung erfolgt von links nach rechts

$p_{i+1,j-1}(x_k) = p_{i,j-1}(x_k) = y_k$, dass

$$\begin{aligned} q(x_k) &= \frac{(x_k - x_i)p_{i+1,j-1}(x_k) - (x_k - x_{i+j})p_{i,j-1}(x_k)}{x_{i+j} - x_i} \\ &= \frac{(x_k - x_i)y_k - (x_k - x_{i+j})y_k}{x_{i+j} - x_i} = y_k. \end{aligned}$$

Die Eindeutigkeit des Interpolationspolynoms impliziert $q = p_{i,j}$. \square

Bemerkung 11.5 Das Neville-Schema sollte nicht rückwärts in rekursiver Form realisiert werden, da sonst viele Größen mehrfach berechnet werden. Stattdessen sollten die Werte $p_{i,j}(x)$ sukzessive vorwärts ausgewertet werden, was auf den Aufwand $\mathcal{O}(n^2)$ führt und in Abb. 11.4 illustriert ist.

Bemerkung 11.6 Eng verbunden mit dem Neville-Schema ist das Verfahren der *dividierten Differenzen*, das die Koeffizienten λ_j , $j = 0, 1, \dots, n$, des Lagrange-Interpolationspolynoms bezüglich der *Newton-Basis* (q_0, q_1, \dots, q_n) definiert durch $q_0 = 1$ und

$$q_j(x) = \prod_{k=0}^{j-1} (x - x_k),$$

$j = 1, 2, \dots, n$, das heißt $p(x) = \sum_{j=0}^n \lambda_j q_j(x)$, bestimmt. Mit der Initialisierung $y_{i,0} = y_i$, $i = 0, 1, \dots, n$, und der Iterationsvorschrift

$$y_{i,j} = \frac{y_{i+1,j-1} - y_{i,j-1}}{x_{i+j} - x_i}$$

für $1 \leq j \leq n$ und $0 \leq i \leq n - j$ gilt $\lambda_j = y_{0,j}$, $j = 0, 1, \dots, n$. Die Auswertung des Interpolationspolynoms erfolgt dann effizient mit dem *Horner-Schema*, das heißt mittels der Darstellung

$$p(x) = \lambda_0 + (x - x_0)[\lambda_1 + (x - x_1)[\lambda_2 + \dots [\lambda_{n-1} + (x - x_{n-1})\lambda_n] \dots]].$$

Diese Art der Auswertung des Interpolationspolynoms hat die nützliche Eigenschaft, dass weitere Stützpaare einfach hinzugefügt werden können. Das Schema ist zudem dann gut geeignet, wenn der Wert des Polynoms p an mehreren Stellen benötigt wird.

11.4 Tschebyscheff-Knoten

Eine Möglichkeit, den Interpolationsfehler bei der Lagrange-Interpolation zu verringern, besteht in der Optimierung der Stützstellen, sodass das Stützstellenpolynom

$$w(x) = \prod_{j=0}^n (x - x_j)$$

im Intervall $[a, b]$ möglichst gleichmäßig kleine Werte annimmt. Ohne Beschränkung der Allgemeinheit betrachten wir den Fall $[a, b] = [-1, 1]$.

Definition 11.4 Für $n \in \mathbb{N}_0$ ist das n -te *Tschebyscheff-Polynom* für $t \in [-1, 1]$ definiert durch

$$T_n(t) = \cos(n \arccos t),$$

s. Abb. 11.5. Die Nullstellen eines Tschebyscheff-Polynoms heißen *Tschebyscheff-Knoten*.

Die Tschebyscheff-Polynome haben bemerkenswerte Eigenschaften.

Lemma 11.1

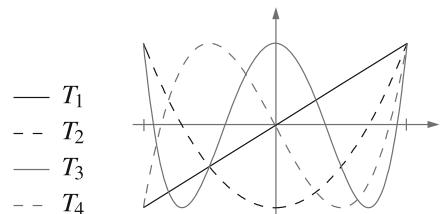
- (i) Es gilt $|T_n(t)| \leq 1$ für alle $t \in [-1, 1]$.
- (ii) Mit $T_0(t) = 1$ und $T_1(t) = t$ gilt

$$T_{n+1}(t) = 2t T_n(t) - T_{n-1}(t)$$

für alle $t \in [-1, 1]$. Insbesondere gilt $T_n \in \mathcal{P}_n|_{[-1,1]}$ und für $n \geq 1$ folgt $T_n(t) = 2^{n-1} t^n + q_{n-1}(t)$ mit $q_{n-1} \in \mathcal{P}_{n-1}|_{[-1,1]}$.

- (iii) Für $n \geq 1$ hat T_n die Nullstellen $t_j = \cos((j + 1/2)\pi/n)$, $j = 0, 1, \dots, n-1$, und die $n+1$ Extremstellen $s_j = \cos(j\pi/n)$, $j = 0, 1, \dots, n$.

Abb. 11.5 Die Nullstellen der Tschebyscheff-Polynome definieren die Tschebyscheff-Knoten



Beweis Übungsaufgabe. □

Die Nullstellen der Tschebyscheff-Polynome definieren eine optimale Wahl der Stützstellen im Sinne des folgenden Satzes.

Satz 11.4 Es seien $t_0, t_1, \dots, t_n \in [-1, 1]$ die Nullstellen des Tschebyscheff-Polynoms T_{n+1} . Dann gilt

$$\min_{x_0, \dots, x_n \in [-1, 1]} \max_{x \in [-1, 1]} \prod_{j=0}^n |x - x_j| = \max_{x \in [-1, 1]} \prod_{j=0}^n |x - t_j| = 2^{-n}.$$

Beweis Aus dem vorangegangenen Lemma folgt $T_{n+1}(x) = 2^n \prod_{j=0}^n (x - t_j)$ sowie $\max_{x \in [-1, 1]} |T_{n+1}(x)| = 1$, was die zweite behauptete Identität beweist. Angenommen, die Stützstellen t_0, t_1, \dots, t_n sind nicht optimal, das heißt es existieren x_0, x_1, \dots, x_n derart, dass für $w(x) = \prod_{j=0}^n (x - x_j)$ gilt, dass $\max_{x \in [-1, 1]} |w(x)| < 2^{-n}$. Da $w(x) = x^{n+1} + r_n(x)$ und $T_{n+1}(x) = 2^n x^{n+1} + q_n(x)$ mit $q_n, r_n \in P_n$, folgt, dass $p = 2^{-n} T_{n+1} - w = 2^{-n} q_n - r$ ein Polynom vom Grad n ist, also $p \in P_n$. Da T_{n+1} in seinen $n+2$ Extremstellen s_0, s_1, \dots, s_{n+1} die Werte ± 1 mit wechselndem Vorzeichen annimmt und $|w(x)| < 2^{-n}$ gilt, folgt dass s_0, s_1, \dots, s_{n+1} auch Extremstellen von p mit alternierenden Vorzeichen sind. Dies impliziert, dass p in $[-1, 1]$ mindestens $n+1$ Nullstellen besitzt. Dies hat $p = 0$ zur Folge, was aber im Widerspruch zu $2^{-n} |T_{n+1}(s_j)| = 2^{-n} > |w(s_j)|$ steht. □

Bemerkungen 11.7 (i) Für den Interpolationsfehler mit Tschebyscheff-Knoten im Intervall $[-1, 1]$ ergibt sich die Abschätzung

$$\|f - p\|_{C^0([-1, 1])} \leq 2^{-n} \frac{\|f^{(n+1)}\|_{C^0([-1, 1])}}{(n+1)!}.$$

(ii) Für allgemeine Intervalle $[a, b]$ konstruiert man die optimalen Stützstellen mit Hilfe einer affin-linearen Transformation $\psi : [-1, 1] \rightarrow [a, b]$.

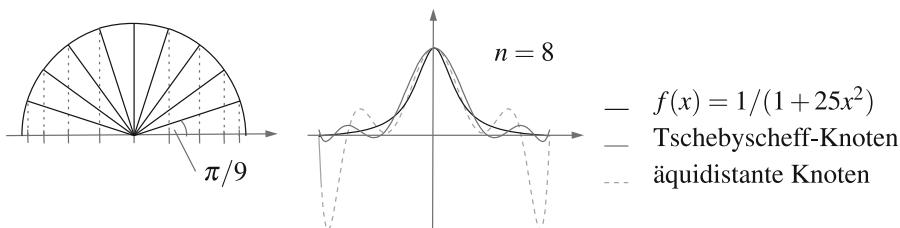


Abb. 11.6 Interpolation der Funktion $f(x) = 1/(1 + 25x^2)$ mit äquidistant verteilt sowie Tschebyscheff-Knoten

(iii) Die Tschebyscheff-Knoten entsprechen der vertikalen Projektion von gleichverteilten $n + 1$ Punkten auf einem Halbkreis, s. Abb. 11.6.

(iv) Für die Interpolation mit Tschebyscheff-Knoten kann man zeigen, dass für Lipschitz-stetige Funktionen gleichmäßige Konvergenz gilt. Insbesondere gilt punktweise Konvergenz für die Funktion $f(x) = 1/(1 + 25x^2)$. Das Interpolationspolynom mit 9 Tschebyscheff-Knoten ist in Abb. 11.6 gezeigt.

11.5 Hermite-Interpolation

Bei der Interpolation von glatten, das heißt sehr oft stetig differenzierbaren, Funktionen ist es sinnvoll, Ableitungen an Stützstellen vorzuschreiben, um den Approximationsfehler bei festgehaltener Anzahl von Stützstellen zu verringern, s. Abb. 11.7.

Definition 11.5 Die *Hermite-Interpolationsaufgabe* sucht für Stützstellen $a \leq x_0 < x_1 < \dots < x_n \leq b$ und Stützwerte $y_i^{(0)}, y_i^{(1)}, \dots, y_i^{(\ell_i)}$ für $i = 0, 1, \dots, n$ mit Zahlen $\ell_i \in \mathbb{N}_0$ ein Polynom $p \in \mathcal{P}_N$, sodass

$$p(x_i) = y_i^{(0)}, \quad p'(x_i) = y_i^{(1)}, \quad \dots, \quad p^{(\ell_i)}(x_i) = y_i^{(\ell_i)}$$

für $i = 0, 1, \dots, n$ mit $N = \sum_{i=0}^n (\ell_i + 1) - 1$ gilt.

Bei der Hermite-Interpolationsaufgabe sind $N + 1 = \sum_{i=0}^n (\ell_i + 1)$ Bedingungen zu erfüllen, sodass es sinnvoll ist, den Polynomraum \mathcal{P}_N zu verwenden.

Satz 11.5 Die Hermite-Interpolationsaufgabe ist eindeutig lösbar.

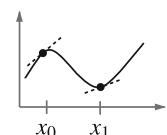
Beweis Die lineare Abbildung $T : \mathcal{P}_N \rightarrow \mathbb{R}^{N+1}$ sei definiert durch

$$Tp = [p(x_0), p'(x_0), \dots, p^{(\ell_1)}(x_0), \dots, p(x_n), p'(x_n), \dots, p^{(\ell_n)}(x_n)]^\top.$$

Gilt $Tp = 0$, so hat p die Nullstellen x_i , $i = 0, 1, \dots, n$, mit Vielfachheiten $\ell_i + 1$ und es folgt

$$p(x) = \alpha \prod_{i=0}^n (x - x_i)^{\ell_i + 1}$$

Abb. 11.7 Bei der Hermite-Interpolation sind neben Funktionswerten auch Ableitungen an den Knoten vorgeschrieben



mit einer Zahl $\alpha \in \mathbb{R}$. Unter Berücksichtigung der Vielfachheiten hat $p \in \mathcal{P}_N$ also insgesamt $N + 1$ Nullstellen und der Fundamentalsatz der Algebra impliziert $p = 0$. Damit ist T injektiv und als lineare Abbildung zwischen Räumen gleicher Dimension auch bijektiv. Dies impliziert die eindeutige Lösbarkeit der Hermite-Interpolationsaufgabe. \square

Bemerkung 11.8 Im Fall einer einzigen Stützstelle x_0 und Stützwerten $y_0^{(j)} = f^{(j)}(x_0)$, $j = 0, 1, \dots, \ell_0$, liefert die Hermite-Interpolationsaufgabe das ℓ_0 -te Taylor-Polynom von f an der Stelle x_0 .

Zur Herleitung einer Fehlerabschätzung beschränken wir uns auf den Fall $\ell_0 = \ell_1 = \dots = \ell_n = \ell$ für ein $\ell \geq 0$, sodass $N + 1 = (\ell + 1)(n + 1)$ gilt.

Satz 11.6 Für $f \in C^{N+1}([a, b])$ sei $p \in \mathcal{P}_N$ das Hermite-Polynom mit $p^{(k)}(x_i) = f^{(k)}(x_i)$, $0 \leq i \leq n$, $0 \leq k \leq \ell$, für gegebene Stützstellen $a \leq x_0 < x_1 < \dots < x_n \leq b$. Für jedes $x \in [a, b]$ existiert ein $\xi \in [a, b]$ mit

$$f(x) - p(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} \prod_{i=0}^n (x - x_i)^{\ell+1}$$

insbesondere gilt also

$$\|f - p\|_{C^0([a,b])} \leq \frac{\|f^{(N+1)}\|_{C^0([a,b])}}{(N+1)!} \prod_{i=0}^n |x - x_i|^{\ell+1}.$$

Beweis Gilt $x \in \{x_0, x_1, \dots, x_n\}$, so ist die Aussage klar, und es sei $x \in [a, b] \setminus \{x_0, x_1, \dots, x_n\}$ im Folgenden. Für $y \in [a, b]$ definiere

$$w(y) = \prod_{i=0}^n (y - x_i)^{\ell+1} \in \mathcal{P}_{N+1}$$

und

$$F(y) = (f(x) - p(x)) w(y) - (f(y) - p(y)) w(x).$$

Die Funktion F hat die $(\ell + 1)$ -fachen Nullstellen x_0, x_1, \dots, x_n sowie die einfache Nullstelle x und zwischen zwei benachbarten Nullstellen hat F' eine davon verschiedene Nullstelle. Damit hat F' nach dem Satz von Rolle neben den ℓ -fachen Nullstellen bei x_0, x_1, \dots, x_n weitere $n + 1$ Nullstellen, also insgesamt mindestens $2n + 2$ Nullstellen. Zwischen all diesen Nullstellen hat F'' weitere Nullstellen, sofern $\ell \geq 2$ gilt, das heißt F'' hat $(n + 1) + (2n + 1) = 3n + 2$ Nullstellen. Induktiv prüft man, dass $F^{(\ell)}$ mindestens $(n + 1) + (\ell n + 1)$ viele Nullstellen hat. Bei jedem Ableiten von $F^{(\ell)}$ verringert sich die Anzahl der Nullstellen um eine und damit hat die Ableitung $F^{(\ell+n+1+\ell n)} = F^{(N+1)}$ noch eine Nullstelle $\xi \in [a, b]$. Damit gilt

$$0 = F^{(N+1)}(\xi) = (f(x) - p(x))(N+1)! - (f^{(N+1)}(\xi) - 0) w(\xi)$$

und dies impliziert die behaupteten Aussagen. \square

Bemerkung 11.9 Bei der Hermite-Interpolation mit 3 Knoten und Vorgabe der Funktionswerte sowie zwei Ableitungen an jedem Knoten erhält man also eine vergleichbare Genauigkeit wie bei der Lagrange-Interpolation mit 9 Knoten.

11.6 Lernziele, Quiz und Anwendung

Ihnen sollten verschiedene Interpolationsaufgaben bekannt sein. Entsprechende Fehlerabschätzungen sollten Sie beweisen können. Die Möglichkeiten der Verbesserung von Interpolationsresultaten durch verschiedene Wahlen von Stützstellen sollten Sie erläutern und an Beispielen veranschaulichen können.

Quiz 11.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Die Lagrange-Polynome erfüllen die Identität $\sum_{i=0}^n L_i(x) = 1$ für alle $x \in [a, b]$	
Um die Funktion $f(x) = \sin(x)$ im Intervall $[0, 1]$ tabellarisch mit einem Fehler von höchstens 0.01 zu erfassen, genügt die Angabe von vier Funktionswerten	
Tschebyscheff-Knoten sind die Extremstellen der Tschebyscheff-Polynome	
Das Neville-Schema berechnet die Koeffizienten des Lagrange-Interpolationspolynoms bezüglich der Monom-Basis	
Die Hermite-Interpolationsaufgabe mit vier Stützstellen und Vorgabe der ersten und zweiten Ableitungen an den Stützstellen führt auf 8 Bedingungen	

Anwendung 11.1 Eine Presse zur Herstellung mechanischer Bauteile werde über eine Spindel angesteuert. Dabei führt der Spindelweg $0 \leq s \leq \ell$ zu einem Durchmesser $d(s)$ des Bauteils. Um Bauteile vorgegebenen Durchmessers herzustellen, muss also ein geeigneter Spindelweg angegeben werden. Tests mit der Maschine führen auf die Messwerte in Millimetern

$$(s, d(s)) = (0.10, 0.098), (0.20, 0.043), (0.35, 0.122), (0.40, 0.157).$$

Konstruieren Sie eine Funktion, die für einen vorgegebenen Radius auf Basis dieser Daten einen sinnvollen Spindelweg angibt.

12.1 Splines

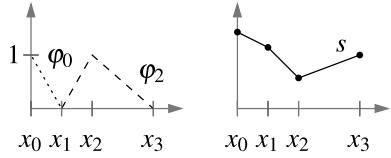
Die Interpolation mit Polynomen erfordert hohe Regularitätseigenschaften von Funktionen, um kleine Fehler zu garantieren. Um auch Funktionen, die beispielsweise nur $f \in C^2([a, b])$ erfüllen, mit hoher Genauigkeit zu approximieren, wird das Intervall $[a, b]$ in Teilintervalle zerlegt und auf jedem Teilintervall eine polynomiale Interpolation durchgeführt. An den Übergängen zwischen den Teilintervallen müssen geeignete Stetigkeitsbedingungen gestellt werden. Wir folgen in diesem Kapitel den Darstellungen in [5,6,7].

Definition 12.1 Für eine durch $a = x_0 < x_1 < \dots < x_n = b$ definierte Partitionierung \mathcal{T}_n von $[a, b]$, heißt eine Funktion $s : [a, b] \rightarrow \mathbb{R}$ Spline vom (polynomiellen) Grad $m \in \mathbb{N}_0$ und von der (Differenzierbarkeits-)Ordnung $k \in \mathbb{N}_0$, falls $s \in C^k([a, b])$ und $s|_{[x_{i-1}, x_i]} \in \mathcal{P}_m|_{[x_{i-1}, x_i]}$, $i = 1, 2, \dots, n$, gilt. Es bezeichne $S^{m,k}(\mathcal{T}_n)$ den Raum aller Splines vom Grad m bezüglich \mathcal{T}_n . Splines vom Grad $m = 1, 2, 3$ der Ordnung $m - 1$ heißen *lineare*, *quadratische* beziehungsweise *kubische* Splines.

Bemerkung 12.1 Häufig wird bei Splines nur die Differenzierbarkeitsordnung $k = m - 1$ betrachtet und dann $S^m(\mathcal{T}_n)$ statt $S^{m,m-1}(\mathcal{T}_n)$ geschrieben. Dies ist die maximale Ordnung, für die der Polynomraum \mathcal{P}_m ein echter Teilraum von $S^{m,k}(\mathcal{T}_n)$ ist. Für $k \geq m$ gilt hingegen $\mathcal{P}_m|_{[a,b]} = S^{m,k}(\mathcal{T}_n)$.

Satz 12.1 Für gegebene Werte y_0, y_1, \dots, y_n existiert genau ein linearer Spline $s \in S^{1,0}(\mathcal{T}_n)$ mit $s(x_i) = y_i$ für $i = 0, 1, \dots, n$. Dieser ist gegeben durch $s = \sum_{i=0}^n y_i \varphi_i$ mit den Hufunktionen $(\varphi_0, \varphi_1, \dots, \varphi_n) \in S^{1,0}(\mathcal{T}_n)$, die durch $\varphi_i(x_j) = \delta_{ij}$ für $0 \leq i, j \leq n$ definiert sind, s. Abb. 12.1.

Abb. 12.1 Den Knoten x_0 und x_2 zugeordnete Hutfunktionen φ_0 , φ_2 und lineare Splinefunktion s



Beweis Die Funktion $\varphi_i \in S^{1,0}(\mathcal{T}_n)$ ist für $x \in [a, b]$ gegeben durch

$$\varphi_i(x) = \begin{cases} (x - x_{i-1})/(x_i - x_{i-1}), & x \in [x_{i-1}, x_i], \\ (x_{i+1} - x)/(x_{i+1} - x_i), & x \in [x_i, x_{i+1}], \\ 0, & \text{sonst.} \end{cases}$$

Aus der Darstellung $s = \sum_{i=0}^n y_i \varphi_i$ folgt die eindeutige Lösbarkeit der Interpolationsaufgabe. \square

Satz 12.2 Die Dimension des Raums $S^{m,m-1}(\mathcal{T}_n)$ ist $n + m$.

Beweis Für $m = 1$ folgt die Aussage aus dem vorangegangenen Satz und es sei $(\varphi_0, \varphi_1, \dots, \varphi_n)$ die aus den Hutfunktionen bestehende Basis von $S^{1,0}(\mathcal{T}_n)$. Für $i = 0, 1, \dots, n$ sei r_i eine $(m-1)$ -te Stammfunktion von φ_i , das heißt $r_i^{(m-1)} = \varphi_i$. Dann gilt $(r_0, r_1, \dots, r_n) \subset S^{m,m-1}(\mathcal{T}_n)$. Zudem ist die Monombasis $(x^0, x^1, \dots, x^{m-2})$ in $S^{m,m-1}(\mathcal{T}_n)$ enthalten und wir zeigen, dass

$$(r_0, r_1, \dots, r_n, x^0, x^1, \dots, x^{m-2})$$

eine Basis von $S^{m,m-1}(\mathcal{T}_n)$ ist. Sei dazu $s \in S^{m,m-1}(\mathcal{T}_n)$. Da $s^{(m-1)} \in S^{1,0}(\mathcal{T}_n)$ gilt, existieren c_0, c_1, \dots, c_n mit

$$s^{(m-1)} = \sum_{i=0}^n c_i \varphi_i$$

und $(m-1)$ -maliges Integrieren führt auf

$$s(x) = \sum_{i=0}^n c_i r_i(x) + \sum_{j=0}^{m-2} d_j x^j$$

mit geeigneten Integrationskonstanten d_0, d_1, \dots, d_{m-2} . Zum Nachweis der linearen Unabhängigkeit seien c_0, c_1, \dots, c_n und d_0, d_1, \dots, d_{m-2} , sodass

$$\sum_{i=0}^n c_i r_i(x) + \sum_{j=0}^{m-2} d_j x^j = 0$$

für alle $x \in [a, b]$ gilt. Durch $(m - 1)$ -maliges Differenzieren folgt

$$\sum_{i=0}^n c_i r_i^{(m-1)} = \sum_{i=0}^n c_i \varphi_i = 0$$

und damit $c_0 = c_1 = \dots = c_n = 0$. Dies hat $\sum_{j=0}^{m-2} d_j x^j = 0$ zur Folge, was wiederum $d_0 = d_1 = \dots = d_{m-2} = 0$ impliziert. \square

Bemerkungen 12.2 (i) Bei $n + 1$ Stützstellen müssen neben $n + 1$ Interpolationsbedingungen $s(x_i) = y_i$, $i = 0, 1, \dots, n$, weitere $m - 1$ Bedingungen gestellt werden, um $s \in S^{m,m-1}(\mathcal{T}_n)$ eindeutig festzulegen.

(ii) Ist m ungerade und $f \in C^{m+1}([a, b])$, so kann ein interpolierender Spline $s \in S^{m,(m-1)/2}(\mathcal{T}_n)$ durch stückweise Lagrange- beziehungsweise Hermite-Interpolation definiert werden mit der Eigenschaft

$$\|f - s\|_{C^0([a,b])} \leq \frac{h^{m+1}}{(m+1)!} \|f^{(m+1)}\|_{C^0([a,b])},$$

wobei $h = \max_{i=1,\dots,n} (x_i - x_{i-1})$ die maximale Gitterweite der Partitionierung \mathcal{T}_n bezeichne.

12.2 Kubische Splines

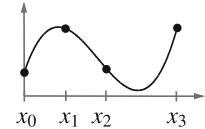
Während lineare Splines Knicke haben und quadratische Splines unstetige zweite Ableitungen besitzen, die bei praxisrelevanten Auflösungen gut wahrgenommen werden können, erscheinen kubische Splines als sehr glatt.

Definition 12.2 Für eine Partitionierung $\mathcal{T}_n = \{x_0, x_1, \dots, x_n\}$ des Intervalls $[a, b]$ und Stützwerte y_0, y_1, \dots, y_n besteht die *Interpolationsaufgabe mit kubischen Splines* in der Bestimmung einer Funktion $s \in S^{3,2}(\mathcal{T}_n)$ mit $s(x_i) = y_i$ für $i = 0, 1, \dots, n$ unter Berücksichtigung einer der folgenden Randbedingungen:

- *natürliche Randbedingungen*, das heißt $s''(a) = 0$ und $s''(b) = 0$;
- *vollständige oder Hermite-Randbedingungen*, das heißt $s'(a) = y_0^{(1)}$ und $s'(b) = y_n^{(1)}$ mit gegebenen Zahlen $y_0^{(1)}, y_n^{(1)} \in \mathbb{R}$;
- *periodische Randbedingungen*, das heißt $s'(a) = s'(b)$ und $s''(a) = s''(b)$, wobei zusätzlich $y_0 = y_n$ gelte.

Bemerkung 12.3 Die kubische Spline-Interpolation lässt sich interpretieren als das Fixieren einer dünnen Holzleiste durch gegebene Punkte, s. Abb. 12.2. Der Begriff *Spline* ist die englische Bezeichnung eines im Schiffsbau verwendeten langen, sehr biegsamen Lineals, das im Deutschen auch als *Straklatte* bezeichnet wird.

Abb. 12.2 Kubische Splines sind stückweise glatte, durch vorgegebene Punkte verlaufende C^2 -Kurven



Interpolierende kubische Splines sind minimal für eine linearisierte Biegeenergie, wie die folgende Aussage zeigt.

Satz 12.3 Sei $s \in S^{3,2}(\mathcal{T}_n)$ eine Lösung einer kubischen Spline-Interpolationsaufgabe und sei $g \in C^2([a,b])$ eine beliebige Funktion, die die Interpolationsbedingungen $g(x_i) = y_i$, $i = 0, 1, \dots, n$, sowie dieselben Randbedingungen wie s erfüllt. Dann gilt

$$\int_a^b |s''(x)|^2 dx + \int_a^b |(s - g)''(x)|^2 dx = \int_a^b |g''(x)|^2 dx.$$

Beweis Es gilt

$$\begin{aligned} \int_a^b |g''|^2 dx &= \int_a^b |s'' + (g - s)''|^2 dx \\ &= \int_a^b |s''|^2 dx + \int_a^b |(g - s)''|^2 dx + 2 \int_a^b s''(g - s)'' dx \end{aligned}$$

und es genügt zu zeigen, dass das letzte Integral auf der rechten Seite verschwindet. Aus den Randbedingungen folgt

$$s''(a)(g'(a) - s'(a)) = s''(b)(g'(b) - s'(b)).$$

Partielle Integration auf jedem Teilintervall $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$, zeigt

$$\begin{aligned} \int_a^b s''(g - s)'' dx &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s''(g - s)'' dx \\ &= \sum_{i=1}^n \left(- \int_{x_{i-1}}^{x_i} s'''(g - s)' dx + (s''(g - s)') \Big|_{x_{i-1}}^{x_i} \right). \end{aligned}$$

Für die Summe der Randterme ergibt sich unter Verwendung von $s \in C^2([a, b])$ und der Randbedingungen bei $x_0 = a$ und $x_n = b$, dass

$$\begin{aligned} \sum_{i=1}^n (s''(g-s)')|_{x_{i-1}}^{x_i} &= \sum_{i=1}^n (s''(x_i)(g-s)'(x_i) - s''(x_{i-1})(g-s)'(x_{i-1})) \\ &= s''(x_n)(g-s)'(x_n) - s''(x_0)(g-s)'(x_0) = 0. \end{aligned}$$

Da s''' auf jedem Intervall (x_{i-1}, x_i) konstant ist, beispielsweise mit Wert c_i , folgt unter Verwendung des Hauptsatzes der Infinitesimalrechnung und der Interpolationsbedingungen $s(x_i) = g(x_i)$ für $i = 0, 1, \dots, n$, dass

$$\sum_{i=1}^n \int_{x_{i-1}}^{x_i} s'''(g-s)' \, dx = \sum_{i=1}^n c_i ((g-s)(x_i) - (g-s)(x_{i-1})) = 0.$$

Insgesamt ist damit

$$\int_a^b s''(g-s)'' \, dx = 0$$

gezeigt und die Aussage des Satzes bewiesen. \square

Aus dem vorangegangenen Satz folgt die Wohlgestelltheit der Interpolationsaufgabe, die hier nur für vollständige Randbedingungen nachgewiesen wird.

Satz 12.4 *Es existiert eine eindeutige Lösung der Interpolationsaufgabe mit kubischen Splines und vollständigen Randbedingungen.*

Beweis Sind $s, g \in S^{3,2}(\mathcal{T}_n)$ zwei Lösungen der Interpolationsaufgabe, so folgt aus dem vorigen Satz, dass

$$\int_a^b |(s-g)''|^2 \, dx = 0$$

und somit $(s-g)'' = 0$ beziehungsweise $s(x) - g(x) = p + qx$ in $[a, b]$. Aus $s(x_i) - g(x_i) = 0$ für $i = 0$ und $i = n$ folgt $p = q = 0$ und somit $s = g$. Im Fall der vollständigen Randbedingungen folgt mit $\dim S^{3,2}(\mathcal{T}_n) = n + 3$, dass die lineare Abbildung

$$T_H : S^{3,2}(\mathcal{T}_n) \rightarrow \mathbb{R}^{n+3}, s \mapsto (s(x_0), \dots, s(x_n), s'(x_0), s'(x_n))$$

injektiv und somit auch bijektiv ist. \square

12.3 Berechnung kubischer Splines

Aufgrund der Regularitätsbedingung $s \in C^2([a, b])$ lassen sich interpolierende kubische Splines nicht lokal bestimmen und es muss ein lineares Gleichungssystem gelöst werden, um eine Darstellung in der Monombasis auf jedem Teilintervall zu erhalten.

Satz 12.5 Für eine Partitionierung $x_0 < x_1 < \dots < x_n$ und gegebene Interpolationswerte y_0, y_1, \dots, y_n sei $s \in S^{3,2}(\mathcal{T}_n)$ mit $s(x_i) = y_i$, $i = 0, 1, \dots, n$. Dann sind die Ableitungen $\gamma_i = s''(x_i)$, $i = 0, 1, \dots, n$, gegeben als Lösung des Gleichungssystems

$$h_i \frac{\gamma_i}{6} + \frac{(h_{i+1} + h_i)}{2} \frac{4\gamma_{i+1}}{6} + h_{i+1} \frac{\gamma_{i+2}}{6} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i},$$

für $i = 0, 1, \dots, n-2$, wobei $h_i = x_{i+1} - x_i$ sei. Mit den Größen

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{\gamma_i}{2} h_i - \frac{d_i}{6} h_i^2, \quad d_i = \frac{\gamma_{i+1} - \gamma_i}{h_i}$$

folgt dann auf jedem Teilintervall $[x_i, x_{i+1}]$, $i = 0, 1, \dots, n-1$, die Darstellung

$$s|_{[x_i, x_{i+1}]} = y_i + b_i(x - x_i) + \frac{\gamma_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3.$$

Beweis Ist $s \in S^{3,2}(\mathcal{T}_n)$ mit $s(x_i) = y_i$ und $s''(x_i) = \gamma_i$ so existieren $b_i, d_i \in \mathbb{R}$, $i = 0, 1, \dots, n-1$, mit

$$s|_{[x_i, x_{i+1}]}(x) = p_i(x) = y_i + b_i(x - x_i) + \frac{\gamma_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3.$$

- (i) Die Stetigkeit von s bei x_{i+1} , das heißt die Identität $p_i(x_{i+1}) = p_{i+1}(x_{i+1})$ führt auf die Gleichung

$$y_i + b_i h_i + \frac{\gamma_i}{2} h_i^2 + \frac{d_i}{6} h_i^3 = y_{i+1} \iff b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{\gamma_i}{2} h_i - \frac{d_i}{6} h_i^2$$

für $i = 0, 1, \dots, n-1$, womit b_i durch d_i, y_i, y_{i+1} und γ_i festgelegt wird.

- (ii) Die Stetigkeit von s'' bei x_{i+1} beziehungsweise die Identität $p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$ sowie $s''(x_n) = \gamma_n$ führt auf die Gleichung

$$\gamma_i + d_i h_i = \gamma_{i+1} \iff d_i = \frac{\gamma_{i+1} - \gamma_i}{h_i}$$

für $i = 0, 1, \dots, n-1$, womit d_i durch γ_i und γ_{i+1} festgelegt ist.

- (iii) Die Stetigkeit von s' bei x_{i+1} , das heißt die Identität $p'_i(x_{i+1}) = p'_{i+1}(x_{i+1})$, ist gleichbedeutend mit

$$b_i + h_i \gamma_i + \frac{d_i}{2} h_i^2 = b_{i+1} \iff b_{i+1} - b_i = h_i \gamma_i + \frac{d_i}{2} h_i^2$$

für $i = 0, 1, \dots, n-2$. Verwendet man in dieser Identität die obigen Darstellungen von b_i und b_{i+1} sowie d_i und d_{i+1} , so ergeben sich die behaupteten $n-1$ Gleichungen für die Koeffizienten γ_i , $i = 0, 1, 2, \dots, n$. \square

Im vorangegangenen Satz wurden $n-1$ Gleichungen hergeleitet, die von den $n+1$ Ableitungen $\gamma_i = s''(x_i)$, $i = 0, 1, \dots, n$, erfüllt werden müssen. Die Hinzunahme von zwei Randbedingungen vervollständigt das Gleichungssystem.

Beispiel 12.1 Für ein äquidistantes Gitter, das heißt es gilt $h_i = h$ für $i = 0, 1, \dots, n-1$, und die natürlichen Randbedingungen $s''(x_0) = s''(x_n) = 0$ beziehungsweise $\gamma_0 = \gamma_n = 0$ sind die Größen $\gamma_1, \gamma_2, \dots, \gamma_{n-1}$ als Lösung des tridiagonalen linearen Gleichungssystems

$$\frac{1}{6} \begin{bmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n-1} \end{bmatrix}$$

gegeben, mit $r_i = (y_{i+1} - 2y_i + y_{i-1})/h^2$, $i = 1, 2, \dots, n-1$. Die strikt diagonaldominante Systemmatrix ist regulär und somit existiert eine eindeutige Lösung.

Bemerkung 12.4 Für einen interpolierenden kubischen Spline lässt sich die Fehlerabschätzung

$$\|f - s\|_{C^0([a,b])} \leq ch^4 \|f^{(4)}\|_{C^0([a,b])}$$

mit der maximalen Gitterweite $h = \max_{i=1,\dots,n} (x_i - x_{i-1})$ und einer von f unabhängigen Konstante $c > 0$ beweisen, sofern $f \in C^4([a, b])$ gilt. Dies entspricht der Konvergenzordnung einer stückweisen Lagrange-Interpolation mit kubischen Polynomen, jedoch erhält man dabei keine interpolierende Funktion im Raum $C^2([a, b])$.

12.4 Lernziele, Quiz und Anwendung

Sie sollten Spline-Räume definieren und deren Dimensionen bestimmen können. Für kubische Splines sollten Sie eine Minimalitätseigenschaft und deren Berechnung konkretisieren können.

Quiz 12.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Jede Spline-Funktion ist einmal stetig differenzierbar	
Es gilt $S^{1,0}(\mathcal{T}_n) \cap S^{3,2}(\mathcal{T}_n) = \{0\}$, wobei 0 die konstante Funktion mit Wert 0 bezeichne	
Sind $q \in \mathcal{P}_m$ und $x_0 < x_1 < \dots < x_n$ eine Partitionierung \mathcal{T}_n von $[a, b]$, so gilt $q _{[a,b]} \in S^{m,m-1}(\mathcal{T}_n)$	
Interpolierende kubische Spline-Funktionen minimieren eine linearisierte Biegenergie unter interpolierenden C^2 -Funktionen	
Die Berechnung eines kubischen Splines führt auf ein lineares Gleichungssystem mit diagonaldominanter, irreduzibler Systemmatrix	

Anwendung 12.1 Glatte Kurven wie beispielsweise kubische Spline-Funktionen finden vielfältige Anwendungen in der Computergrafik zur Berechnung und Darstellung von Kurven und Flächen. Mit wenigen Informationen können so komplexe grafische Objekte wie CAD-Modelle oder *Postscript*-Schriftsätze beschrieben werden. Neben dem Speicheranforderung ist auch die effiziente Weiterverarbeitung wie Skalierung oder Rotation der Objekte ein wichtiger Aspekt. Eng verbunden mit Spline-Funktionen sind sogenannte *Bézier-Kurven*, die für gegebene Punkte $P_0, P_1, \dots, P_n \in \mathbb{R}^2$ und $t \in [0, 1]$ definiert sind durch

$$z(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i.$$

- (i) Zeigen Sie, dass für $n = 2$ die Darstellung

$$z(t) = (1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2]$$

gilt und interpretieren Sie diese Formel geometrisch.

- (ii) Zeigen Sie, dass mit der Initialisierung $z_i^0(t) = P_i$, $i = 0, 1, \dots, n$, und der Rekursionsvorschrift

$$z_i^j(t) = (1-t)z_i^{j-1}(t) + tz_{i+1}^{j-1}(t)$$

für $t \in [0, 1]$, $j = 1, 2, \dots, n$, und $i = 0, 1, \dots, n-j$ die Identität $z = z_0^n$ folgt.

- (iii) Implementieren Sie basierend auf der Formel aus (ii) eine rekursive Funktion `y = de_casteljau(j, i, P)` zur Auswertung der Kurve z für gegebene Punkte P_0, P_1, \dots, P_n an einer Stelle $t \in [0, 1]$. Verwenden Sie Ihr Programm, um die durch die Punkte $P_0 = (0, 0)$, $P_1 = (1, 1)$, $P_2 = (2, 0)$ und $P_3 = (3, 2)$ definierte Kurve grafisch darzustellen.

13.1 Trigonometrische Interpolation

Viele in Anwendungen auftretende Signale beziehungsweise Funktionen entstehen durch Überlagerungen von Grundschwingungen verschiedener Frequenzen, das heißt es gilt nach geeigneter Transformation auf das Intervall $[0, 2\pi]$

$$f(x) = \sum_{\ell=0}^{\infty} (c_{\ell} \cos(\ell x) + d_{\ell} \sin(\ell x)),$$

s. Abb. 13.1. Tatsächlich lässt sich jede Riemann-integrierbare Funktion so darstellen und es ist naheliegend, Funktionen mit trigonometrischen Funktionen zu interpolieren. Im Vergleich zur Approximation beispielsweise mit Monomen sind viele Koeffizienten klein und in der Praxis vernachlässigbar.

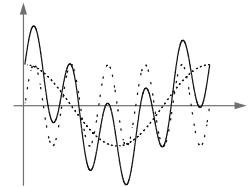
Definition 13.1 Für $m \in \mathbb{N}$, $n = 2m$ und Stützstellen $x_j = 2\pi j/n$ sowie Stützwerte $y_j \in \mathbb{R}$, $j = 0, 1, \dots, n-1$, besteht die *reelle trigonometrische Interpolationsaufgabe* in der Bestimmung von $a_{\ell}, b_{\ell} \in \mathbb{R}$, $\ell = 1, \dots, m-1$, und $a_0, a_m \in \mathbb{R}$, sodass für

$$T(x) = \frac{a_0}{2} + \sum_{\ell=1}^{m-1} (a_{\ell} \cos(\ell x) + b_{\ell} \sin(\ell x)) + \frac{a_m}{2} \cos(mx)$$

die Identität $T(x_j) = y_j$ für $j = 0, 1, \dots, n-1$ gilt.

Die reelle trigonometrische Interpolationsaufgabe lässt sich übersichtlicher im Komplexen darstellen. Es bezeichne $i = \sqrt{-1} \in \mathbb{C}$ die imaginäre Einheit.

Abb. 13.1 Funktionen lassen sich häufig als Summe von Sinus-Schwingungen darstellen



Definition 13.2 Die *komplexe trigonometrische Interpolationsaufgabe* besteht in der Bestimmung von $\beta_k \in \mathbb{C}$, $k = 0, 1, \dots, n-1$, sodass für $x_j = 2\pi j/n$ und $y_j \in \mathbb{C}$, $j = 0, 1, \dots, n-1$, sowie

$$p(x) = \beta_0 + \beta_1 e^{ix} + \beta_2 e^{i2x} + \dots + \beta_{n-1} e^{i(n-1)x} = \sum_{k=0}^{n-1} \beta_k e^{ikx}$$

die Identität $p(x_j) = y_j$ für $j = 0, 1, \dots, n-1$ gilt.

Die reelle und die komplexe trigonometrische Interpolationsaufgabe sind im folgenden Sinne äquivalent zueinander.

Satz 13.1 Seien $n = 2m$ und $y_0, y_1, \dots, y_{n-1} \in \mathbb{R}$. Die Koeffizienten β_k , $k = 0, \dots, n-1$, lösen die komplexe trigonometrische Interpolationsaufgabe genau dann, wenn die Koeffizienten a_0, a_m und a_ℓ, b_ℓ , für $\ell = 1, 2, \dots, m-1$, definiert durch

$$a_0 = 2\beta_0, \quad a_\ell = \beta_\ell + \beta_{2m-\ell}, \quad b_\ell = i(\beta_\ell - \beta_{2m-\ell}), \quad a_m = 2\beta_m,$$

die durch y_0, y_1, \dots, y_{n-1} definierte reelle trigonometrische Interpolationsaufgabe lösen.

Beweis Es gilt $e^{-i\ell x_j} = e^{-i2\pi\ell j/n} = e^{i2\pi(n-\ell)j/n} = e^{i(n-\ell)x_j}$ und mit $e^{ix} = \cos(x) + i \sin(x)$ folgt

$$\begin{aligned} \cos(\ell x_j) &= \operatorname{Re}(e^{i\ell x_j}) = \frac{e^{i\ell x_j} + e^{-i\ell x_j}}{2} = \frac{e^{i\ell x_j} + e^{i(n-\ell)x_j}}{2}, \\ \sin(\ell x_j) &= \operatorname{Im}(e^{i\ell x_j}) = \frac{e^{i\ell x_j} - e^{-i\ell x_j}}{2i} = \frac{e^{i\ell x_j} - e^{i(n-\ell)x_j}}{2i}. \end{aligned}$$

Mit $1/i = -i$ und $n = 2m$ impliziert dies, dass

$$\begin{aligned} &\frac{a_0}{2} + \sum_{\ell=1}^{m-1} (a_\ell \cos(\ell x_j) + b_\ell \sin(\ell x_j)) + \frac{a_m}{2} \cos(mx_j) \\ &= \frac{a_0}{2} + \sum_{\ell=1}^{m-1} \frac{a_\ell - ib_\ell}{2} e^{i\ell x_j} + \sum_{\ell=1}^{m-1} \frac{a_\ell + ib_\ell}{2} e^{i(n-\ell)x_j} + \frac{a_m}{2} \frac{e^{imx_j} + e^{imx_j}}{2}. \end{aligned}$$

Ein Koeffizientenvergleich führt auf $\beta_0 = a_0/2$, $\beta_\ell = (a_\ell - ib_\ell)/2$, $\beta_{n-\ell} = (a_\ell + ib_\ell)/2$ und $\beta_m = a_m/2$, woraus die Behauptung folgt. \square

Bemerkungen 13.1 (i) In der Situation des vorangegangenen Satzes gilt $p(x_j) = T(x_j)$, $j = 0, 1, \dots, n - 1$, aber im Allgemeinen gilt $p \neq T$.

(ii) Aufgrund der Identität $e^{ikx} = (e^{ix})^k$ spricht man auch von trigonometrischen Polynomen.

(iii) Bessere Approximationseigenschaften werden mit Funktionen der Form $r(x) = \sum_{k=-m}^{m-1} \delta_k e^{ikx}$ erzielt, was sich jedoch auf die komplexe trigonometrische Interpolationsaufgabe zurückführen lässt.

13.2 Fourier-Basen

Schreibt man die Interpolationsbedingungen $p(x_j) = y_j$ in vektorieller Form, so erhält man

$$y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \sum_{k=0}^{n-1} \beta_k \begin{bmatrix} e^{ikx_0} \\ e^{ikx_1} \\ \vdots \\ e^{ikx_{n-1}} \end{bmatrix} = \sum_{k=0}^{n-1} \beta_k \omega^k.$$

Diese Identität lässt sich als Basiswechsel von der Darstellung des Vektors y bezüglich der kanonischen Basis im \mathbb{R}^n auf eine Darstellung mit den Vektoren

$$\omega^k = [e^{ikx_0}, e^{ikx_1}, \dots, e^{ikx_{n-1}}]^\top$$

interpretieren. Notwendig und hinreichend für die Lösbarkeit der komplexwertigen trigonometrischen Interpolationsaufgabe ist damit, dass die Vektoren $(\omega^k)_{k=0, \dots, n-1}$ eine Basis des \mathbb{C} -Vektorraums \mathbb{C}^n definieren.

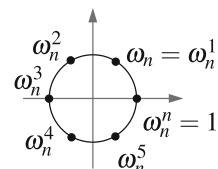
Definition 13.3 Für $n \in \mathbb{N}$ sei $\omega_n = e^{i2\pi/n}$ die n -te Einheitswurzel, s. Abb. 13.2, und für $k = 0, 1, \dots, n - 1$ sei $\omega^k \in \mathbb{C}^n$ durch

$$\omega^k = [\omega_n^{0k}, \omega_n^{1k}, \dots, \omega_n^{(n-1)k}]^\top$$

definiert. Die Familie $(\omega^0, \omega^1, \dots, \omega^{n-1}) \subset \mathbb{C}^n$ heißt Fourier-Basis.

Die Struktur der Fourier-Basisvektoren motiviert die Numerierung von Vektoren in \mathbb{C}^n mit den Indizes $j = 0, 1, \dots, n - 1$. Ebenso werden Matrizen im Folgenden beginnend

Abb. 13.2 Die Potenzen der n -ten Einheitswurzel ω_n sind gleichmäßig auf dem Einheitskreis verteilte komplexe Zahlen



mit 0 indiziert. Das Skalarprodukt zweier Vektoren $a, b \in \mathbb{C}^n$ ist definiert durch $a \cdot b = a^\top \bar{b} = \sum_{j=0}^{n-1} a_j \bar{b}_j$.

Lemma 13.1 Die Vektoren $(\omega^k)_{k=0,\dots,n-1}$ bilden eine Orthogonalbasis des \mathbb{C} -Vektorraums \mathbb{C}^n , das heißt es gilt $\omega^k \cdot \omega^\ell = n\delta_{k\ell}$.

Beweis Übungsaufgabe. □

Zur Lösung der komplexen trigonometrischen Interpolationsaufgabe muss also die darstellende Matrix des Basiswechsels bestimmt werden.

Lemma 13.2 Der Basiswechsel von der Fourier-Basis zur Basis $(e_0, e_1, \dots, e_{n-1})$, bestehend aus den kanonischen Basisvektoren, wird durch die Matrix

$$T_n = [\omega^0, \omega^1, \dots, \omega^{n-1}] \in \mathbb{C}^{n \times n}$$

mit Inverser $T_n^{-1} = (1/n)\bar{T}_n^\top$ realisiert. Für alle $y = \sum_{j=0}^{n-1} y_j e_j \in \mathbb{C}^n$ gilt also $y = \sum_{k=0}^{n-1} \beta_k \omega^k$ mit $\beta = (1/n)\bar{T}_n^\top y$.

Beweis Für $y \in \mathbb{C}^n$ sei $\beta = [\beta_0, \beta_1, \dots, \beta_{n-1}]^\top$ der Koeffizientenvektor bezüglich der Fourier-Basis $(\omega^k)_{k=0,\dots,n-1}$, das heißt es gelte $y = \sum_{k=0}^{n-1} \beta_k \omega^k$. Damit folgt

$$y^\top \bar{\omega}^\ell = \left(\sum_{k=0}^{n-1} \beta_k \omega^k \right)^\top \bar{\omega}^\ell = \sum_{k=0}^{n-1} \beta_k \omega^{k,\top} \bar{\omega}^\ell = n\beta_\ell,$$

also $\beta_\ell = (1/n)y^\top \bar{\omega}^\ell = (1/n)(\bar{\omega}^\ell)^\top y$ beziehungsweise in Vektornotation

$$\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{n-1} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} (\bar{\omega}^0)^\top \\ \vdots \\ (\bar{\omega}^{n-1})^\top \end{bmatrix} y = \frac{1}{n} \bar{T}_n^\top y.$$

Die Identität $\omega^k \cdot \omega^m = n\delta_{km}$ impliziert, dass $T_n \bar{T}_n^\top = nI_n$ beziehungsweise $T_n^{-1} = (1/n)\bar{T}_n^\top$ gilt. □

Definition 13.4 Die Abbildung $y \mapsto \beta = (1/n)\bar{T}_n^\top y$ heißt (*diskrete*) Fouriertransformation und die Umkehrabbildung $\beta \mapsto y = T_n \beta$ wird als Fourier-Synthese bezeichnet.

Bemerkungen 13.2 (i) Die Fourier-Transformation lässt sich mittels der Fourier-Synthese und komplexen Konjugationen darstellen, denn da T_n symmetrisch ist, gilt $\beta = \frac{1}{n}\bar{T}_n^\top y = \frac{1}{n}(T_n \bar{y})$.

(ii) Mit der diskreten Fourier-Transformation wird die komplexe trigonometrische Interpolationsaufgabe gelöst. Die Fourier-Synthese realisiert die Auswertung eines trigonometrischen Polynoms an den Stützstellen.

13.3 Schnelle Fourier-Transformation

Die Matrix T_n besitzt nur n verschiedene Einträge, die in einer zyklischen Art angeordnet sind, sodass sich die Multiplikation mit T_n mit einem deutlich geringerem Aufwand als $\mathcal{O}(n^2)$ realisieren lässt.

Beispiel 13.1 ([6]) Die Fourier-Synthese $y = T_8\beta$ lässt sich unter Verwendung von $\omega_8^\ell = \omega_8^{\ell \bmod 8}$ in der Form

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^1 & \omega_8^2 & \omega_8^3 & \omega_8^4 & \omega_8^5 & \omega_8^6 & \omega_8^7 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 \\ \omega_8^0 & \omega_8^3 & \omega_8^6 & \omega_8^1 & \omega_8^4 & \omega_8^7 & \omega_8^2 & \omega_8^5 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 \\ \omega_8^0 & \omega_8^5 & \omega_8^2 & \omega_8^7 & \omega_8^4 & \omega_8^1 & \omega_8^6 & \omega_8^3 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 \\ \omega_8^0 & \omega_8^7 & \omega_8^6 & \omega_8^5 & \omega_8^4 & \omega_8^3 & \omega_8^2 & \omega_8^1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \end{bmatrix}$$

schreiben. Ein Umordnen der rechten Seite nach geraden und ungeraden Indizes führt auf

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^1 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_8^7 & \omega_8^5 \\ \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^4 & \omega_8^4 & \omega_8^4 & \omega_8^4 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^5 & \omega_8^7 & \omega_8^1 & \omega_8^3 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^6 & \omega_8^2 & \omega_8^6 & \omega_8^2 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^7 & \omega_8^5 & \omega_8^3 & \omega_8^1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \\ \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{bmatrix}.$$

Mit den Identitäten $\omega_8^{2k} = e^{i2\pi 2k/8} = e^{i2\pi k/4} = \omega_4^k$ und $\omega_8^4 = e^{i\pi} = -1$ folgt

$$\begin{bmatrix} y_0 \\ \vdots \\ y_3 \\ y_4 \\ \vdots \\ y_7 \end{bmatrix} = \begin{bmatrix} T_4 & D_4 T_4 \\ T_4 & -D_4 T_4 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_6 \\ \beta_1 \\ \vdots \\ \beta_7 \end{bmatrix},$$

wobei T_4 und D_4 definiert sind durch

$$T_4 = \begin{bmatrix} \omega_4^0 & \omega_4^0 & \omega_4^0 & \omega_4^0 \\ \omega_4^0 & \omega_4^1 & \omega_4^2 & \omega_4^3 \\ \omega_4^0 & \omega_4^2 & \omega_4^0 & \omega_4^2 \\ \omega_4^0 & \omega_4^3 & \omega_4^2 & \omega_4^1 \end{bmatrix}, \quad D_4 = \begin{bmatrix} \omega_8^0 & & & \\ & \omega_8^1 & & \\ & & \omega_8^2 & \\ & & & \omega_8^3 \end{bmatrix}.$$

Damit folgt

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = T_4 \begin{bmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \end{bmatrix} + D_4 T_4 \begin{bmatrix} \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{bmatrix}, \quad \begin{bmatrix} y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = T_4 \begin{bmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \end{bmatrix} - D_4 T_4 \begin{bmatrix} \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{bmatrix}.$$

Die Fourier-Synthese $y = T_8\beta$ lässt sich also auf zwei Fourier-Synthesen der Dimension $n = 4$ zurückführen.

Die Vorgehensweise des Beispiels lässt sich verallgemeinern.

Satz 13.2 Für $\beta \in \mathbb{C}^{2m}$ sei $D_m \in \mathbb{C}^{m \times m}$ die Diagonalmatrix mit Einträgen $(D_m)_{\ell\ell} = \omega_{2m}^\ell$, $\ell = 0, 1, \dots, m-1$. Dann ist $y = T_{2m}\beta$ gegeben durch $y = (y^1, y^2)$ mit Vektoren $y^1, y^2 \in \mathbb{C}^m$ definiert durch

$$y^1 = T_m \beta^{\text{even}} + D_m T_m \beta^{\text{odd}}, \quad y^2 = T_m \beta^{\text{even}} - D_m T_m \beta^{\text{odd}},$$

wobei $\beta^{\text{even}} = [\beta_0, \beta_2, \dots, \beta_{2m-2}]^\top$ und $\beta^{\text{odd}} = [\beta_1, \beta_3, \dots, \beta_{2m-1}]^\top$ seien.

Beweis Für $0 \leq \ell \leq m-1$ gilt unter Verwendung von $\omega_{2m}^{2k\ell} = \omega_m^{k\ell}$

$$\begin{aligned} y_\ell &= \sum_{j=0}^{2m-1} (T_{2m})_{\ell j} \beta_j = \sum_{j=0}^{2m-1} \omega_{2m}^{j\ell} \beta_j \\ &= \sum_{k=0}^{m-1} \omega_{2m}^{2k\ell} \beta_{2k} + \sum_{k=0}^{m-1} \omega_{2m}^{(2k+1)\ell} \beta_{2k+1} \\ &= \sum_{k=0}^{m-1} \omega_m^{k\ell} \beta_{2k} + \omega_{2m}^\ell \sum_{k=0}^{m-1} \omega_m^{k\ell} \beta_{2k+1} \\ &= \sum_{k=0}^{m-1} (T_m)_{\ell k} \beta_{2k} + (D_m)_{\ell\ell} \sum_{k=0}^{m-1} (T_m)_{\ell k} \beta_{2k+1}, \end{aligned}$$

also $y^1 = T_m \beta^{\text{even}} + D_m T_m \beta^{\text{odd}}$. Für $\ell \geq m$ führt eine analoge Rechnung unter Berücksichtigung von $\omega_{2m}^\ell = \omega_{2m}^{m+\ell \bmod m} = -\omega_{2m}^{\ell \bmod m}$ auf die behauptete Identität. \square

Im Satz wird ein Problem der Größe n mit Aufwand $\mathcal{A}(n)$ auf zwei Probleme der Größe $n/2$ mit Aufwand $\mathcal{A}(n/2)$ reduziert. Das Zusammensetzen der Lösungen der Teilprobleme erfordert dabei den Rechenaufwand $3n/2$. Das Vorgehen lässt sich für Dimensionen $n = 2^\ell$, $\ell = \log_2(n) \in \mathbb{N}$, verallgemeinern und iterieren. Für den Rechenaufwand erhalten wir so

$$\mathcal{A}(n) \rightarrow 2\mathcal{A}(n/2) + \frac{3n}{2} \rightarrow 2\left(2\mathcal{A}(n/4) + \frac{3n}{2}\right) + \frac{3n}{2} \rightarrow \dots \rightarrow 2^\ell \mathcal{A}(1) + \ell \frac{3n}{2}.$$

Da $\mathcal{A}(1) = 1$ gilt, beträgt der Aufwand des resultierenden Verfahrens etwa $n(1 + (3/2)\log_2 n)$ (komplexe) arithmetische Operationen. Dies ersetzt den Aufwand $\mathcal{O}(n^2)$ einer Matrix-Vektor-Multiplikation durch den deutlich geringeren Rechenaufwand $\mathcal{O}(n \log_2(n))$.

13.4 Lernziele, Quiz und Anwendung

Sie sollten die grundlegenden Ideen der diskreten Fourier-Transformation erklären und die Aufwandsreduktion der schnellen Fourier-Transformation beschreiben können.

Quiz 13.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Als komplexer Vektorraum hat \mathbb{C}^n die Dimension n und als reeller Vektorraum die Dimension $2n$	
Ist ω_n die n -te Einheitswurzel, so gilt $\omega_n^{n/2} = -1$ genau dann, wenn n gerade ist	
Die komplexe trigonometrische Interpolationsaufgabe wird durch $\beta = T_n y$ mit der Fourier-Matrix T_n gelöst	
Die Matrix $S_n = (1/\sqrt{n})T_n$ definiert eine Isometrie auf \mathbb{C}^n , das heißt es gilt $\ S_n y\ _2 = \ y\ _2$ für alle $y \in \mathbb{C}^n$	
Für reelle Stützwerte y_0, y_1, \dots, y_{n-1} ist die Lösung der komplexen trigonometrischen Interpolationsaufgabe reellwertig	

Anwendung 13.1 Die diskrete Fourier-Transformation berechnet eine Frequenzzerlegung eines gegebenen Signals. Um nur relevante Informationen weiterzuverarbeiten, können dabei berechnete Koeffizienten, die im Vergleich zu anderen klein sind, häufig vernachlässigt werden. Zudem können Koeffizienten, die zu Frequenzen gehören, die in der jeweiligen Anwendung nicht wahrnehmbar sind, eliminiert werden. Der Vektor $y = [y_0, y_1, \dots, y_{n-1}]^\top \in \mathbb{R}^n$ sei definiert durch $y_j = \sin(2\pi j/n) + (1/10)\xi_j$,

$j = 0, 1, \dots, n - 1$, wobei ξ_j für einen normalverteilten Zufallswert stehe, der in MATLAB mit `randn` generiert werden kann. Verwenden Sie die MATLAB-Routine `fft`, um die Fourier-Transformierte $\beta \in \mathbb{C}^n$ zu bestimmen, und eliminieren Sie Koeffizienten β_k , für die

$$|\beta_k| \leq \theta \max_{\ell=0,1,\dots,n-1} |\beta_\ell|$$

gilt, das heißt ersetzen Sie solche Koeffizienten durch Null. Verwenden Sie die Rücktransformation `ifft`, um einen Vektor $\tilde{y} \in \mathbb{C}^n$ zu erhalten. Interpretieren Sie die Vektoren y und \tilde{y} als Werte einer Funktion und stellen Sie diese für $n = 256$ und verschiedene Werte von θ grafisch dar.

14.1 Quadraturformeln

Ziel der numerischen Integration oder Quadratur ist die Approximation eigentlicher Integrale

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

welche sich nicht explizit mit Hilfe einer Stammfunktion berechnen lassen.

Definition 14.1 Eine *Quadraturformel* auf dem Intervall $[a, b]$ ist eine lineare Abbildung $Q : C^0([a, b]) \rightarrow \mathbb{R}$ der Form

$$Q(f) = \sum_{i=0}^n w_i f(x_i)$$

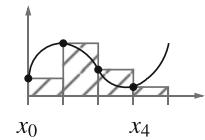
mit (*Quadratur-*)*Punkten* $(x_i)_{i=0,\dots,n}$ und (*Quadratur-*)*Gewichten* $(w_i)_{i=0,\dots,n}$. Die Zahl $\|Q\| = (b-a)^{-1} \sum_{i=0}^n |w_i|$ ist ihr *Stabilitätsindikator*.

Bemerkungen 14.1 (i) Ist $a = x_0 < x_1 < \dots < x_n = b$, so kann das Riemann-Integral approximiert werden durch

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i)$$

und die rechte Seite definiert eine Quadraturformel mit Gewichten $w_i = x_{i+1} - x_i$ für $i = 0, 1, \dots, n-1$ und $w_n = 0$, s. Abb. 14.1.

Abb. 14.1 Riemannsche Summen definieren einfache Quadraturformeln



(ii) Für jede Quadraturformel gilt

$$|Q(f)| \leq \left(\sum_{i=0}^n |w_i| \right) \|f\|_{C^0([a,b])} = \|Q\|(b-a) \|f\|_{C^0([a,b])}.$$

Definition 14.2 Die Quadraturformel Q heißt *exakt vom Grad r*, falls $Q(p) = I(p)$ für alle $p \in \mathcal{P}_r$ gilt.

Lässt sich eine Funktion f gut durch Polynome annähern, so liefert eine Quadraturformel mit hohem Exaktheitsgrad gute Approximationen des Integrals.

Satz 14.1 Sei Q exakt vom Grad $r \geq 0$. Dann gilt $\sum_{i=0}^n w_i = b - a$ und für alle $f \in C^0([a,b])$

$$|I(f) - Q(f)| \leq (1 + \|Q\|)(b-a) \min_{p \in \mathcal{P}_r} \|f - p\|_{C^0([a,b])}.$$

Im Fall $w_i \geq 0$, $i = 0, 1, \dots, n$, gilt $\|Q\| = 1$.

Beweis Nach Voraussetzung gilt $\sum_{i=0}^n w_i = Q(1) = I(1) = b - a$. Sei $f \in C^0([a,b])$ und $p \in \mathcal{P}_r$ beliebig. Mit $I(p) = Q(p)$, der Linearität von I und Q sowie der Dreiecksungleichung folgt

$$\begin{aligned} |I(f) - Q(f)| &= |I(f-p) - Q(f-p)| \leq |I(f-p)| + |Q(f-p)| \\ &\leq (b-a) \|f - p\|_{C^0([a,b])} + \left(\sum_{i=0}^n |w_i| \right) \|f - p\|_{C^0([a,b])} \\ &= (1 + \|Q\|)(b-a) \|f - p\|_{C^0([a,b])}. \end{aligned}$$

Da $p \in \mathcal{P}_r$ beliebig ist, folgt die Behauptung. □

Bemerkungen 14.2 (i) Mit Interpolationsabschätzungen ergeben sich quantitative Aussagen über den Quadraturfehler, das heißt beispielsweise

$$|I(f) - Q(f)| \leq (1 + \|Q\|)(b-a) \frac{\|f^{(r+1)}\|_{C^0([a,b])}}{(r+1)!} (b-a)^{r+1}.$$

Durch Verwendung von Tschebyscheff-Knoten kann diese Abschätzung noch verbessert werden.

(ii) Ist Q exakt vom Grad $2q$ und sind die Gewichte $(w_i)_{i=0,\dots,n}$ und Knoten $(x_i)_{i=0,\dots,n}$ symmetrisch bezüglich dem Intervallmittelpunkt $(a+b)/2$, so ist Q exakt sogar vom Grad $2q+1$.

(iii) Ist Q eine Quadraturformel auf $[a, b]$, so erhält man mit der Abbildung $\varphi : [a, b] \rightarrow [c, d]$, $x \mapsto c + (x - a)(d - c)/(b - a)$, und

$$\int_c^d g(y) dy = \int_a^b g(\varphi(x)) \varphi'(x) dx = \frac{d-c}{b-a} \int_a^b g(\varphi(x)) dx$$

eine Quadraturformel auf dem Intervall $[c, d]$.

14.2 Newton–Cotes-Formeln

Eine Klasse von Quadraturformeln ergibt sich durch Lagrange-Interpolation einer Funktion und anschließende exakte Integration des Interpolationspolynoms. Für gegebene Knoten $x_0 < x_1 < \dots < x_n$ und die zugehörigen Lagrange-Basispolynome

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

ist das Lagrange-Interpolationspolynom gegeben durch $p = \sum_{i=0}^n f(x_i) L_i$. Daher wird durch

$$\int_a^b p(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx = \sum_{i=0}^n w_i f(x_i) = Qf$$

eine Quadraturformel Q mit Gewichten $w_i = \int_a^b L_i(x) dx$ definiert. Da $p = f$ für alle $f \in \mathcal{P}_n$ gilt, ist diese Quadraturformel exakt vom Grad n . Sie wird als *Newton–Cotes-Formel* bezeichnet.

Satz 14.2 Die durch Stützstellen $x_0 < x_1 < \dots < x_n$ und Gewichte $w_i = \int_a^b L_i(x) dx$, $i = 0, 1, \dots, n$, definierte Newton–Cotes-Formel ist exakt vom Grad n .

Beweis Die Aussage folgt unmittelbar aus der Konstruktion der Quadraturformel. □

Für die Fälle $n = 0, 1, 2$ ergeben sich einfache Quadraturformeln, die in Abb. 14.2 dargestellt sind.

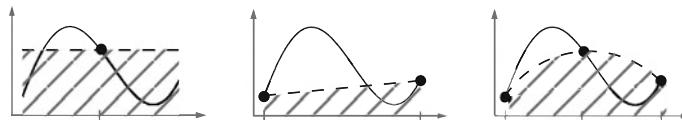


Abb. 14.2 Mittelpunkt-, Trapez- und Simpson-Regel als Spezialfälle der Newton–Cotes-Formeln für $n = 0, 1, 2$

Beispiele 14.1 (i) Für $n = 0$ und $x_0 = (a + b)/2$ ergibt sich die *Mittelpunktregel* $Q_{Mp}(f) = (b - a)f((a + b)/2)$, die exakt ist vom Grad 1.

(ii) Für $n = 1$ und $x_0 = a$, $x_1 = b$ erhält man die ebenfalls vom Grad 1 exakte *Trapezregel*

$$\int_a^b f(x) dx \approx Q_{\text{Trap}}(f) = \frac{b-a}{2} [f(a) + f(b)].$$

(iii) Für $n = 2$ und $x_0 = a$, $x_1 = (a + b)/2$, $x_2 = b$ ergibt sich die *Simpson- oder Keplersche Fassregel*

$$\int_a^b f(x) dx \approx Q_{\text{Sim}}(f) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right],$$

die aufgrund ihrer Symmetrie exakt vom Grad 3 ist.

(iv) Für $n \geq 7$ treten negative Gewichte auf, was zu Stabilitätsproblemen führen kann.

14.3 Summierte Quadraturformeln

Um hohe Genauigkeiten ohne einschränkende Regularitätsannahmen zu erzielen, kann das Intervall $[a, b]$ in kleiner werdende Teilintervalle zerlegt werden, auf denen eine Quadraturformel eines möglicherweise niedrigen Exaktheitsgrads angewendet wird.

Definition 14.3 Sei $a = a_0 < a_1 < \dots < a_N = b$ die uniforme Partitionierung des Intervalls $[a, b]$ mit Knoten $a_\ell = a + \ell(b - a)/N$, $\ell = 0, 1, \dots, N$, und sei $Q_\ell : C^0([a_{\ell-1}, a_\ell]) \rightarrow \mathbb{R}$ eine Quadraturformel auf dem Teilintervall $[a_{\ell-1}, a_\ell]$ für $\ell = 1, 2, \dots, N$. Dann heißt

$$Q^N(f) = \sum_{\ell=1}^N Q_\ell(f|_{[a_{\ell-1}, a_\ell]})$$

summierte Quadraturformel.

Beispiel 14.2 Mit der Trapezregel auf jedem Teilintervall $[a_{\ell-1}, a_\ell]$ folgt

$$\begin{aligned} Q^N f &= \sum_{\ell=1}^N \frac{a_\ell - a_{\ell-1}}{2} (f(a_\ell) - f(a_{\ell-1})) \\ &= \frac{b-a}{2N} (f(a_0) + 2f(a_1) + \cdots + 2f(a_{N-1}) + f(a_N)). \end{aligned}$$

Die Genauigkeit von summierten Quadraturformeln kann durch Verkleinerung der Teilintervalle oder durch Erhöhung des Exaktheitsgrads auf jedem Teilintervall verbessert werden.

Satz 14.3 *Besitzen die Quadraturformeln auf den Teilintervallen den Exaktheitsgrad $r \geq 0$, so gilt*

$$|I(f) - Q^N(f)| \leq (b-a)^{r+2} (1 + \max_{\ell=1,\dots,N} \|Q_\ell\|) \frac{N^{-(r+1)}}{(r+1)!} \|f^{(r+1)}\|_{C^0([a,b])}.$$

Beweis Auf jedem Teilintervall $[a_{\ell-1}, a_\ell]$ gilt

$$\left| \int_{a_{\ell-1}}^{a_\ell} f \, dx - Q_\ell(f) \right| \leq (1 + \|Q_\ell\|)(a_\ell - a_{\ell-1}) \min_{p \in \mathcal{P}_r} \|f - p\|_{C^0([a_{\ell-1}, a_\ell])}.$$

Die Fehlerabschätzungen für die Lagrange-Interpolation zeigen

$$\min_{p \in \mathcal{P}_r} \|f - p\|_{C^0([a_{\ell-1}, a_\ell])} \leq \frac{\|f^{(r+1)}\|_{C^0([a_{\ell-1}, a_\ell])}}{(r+1)!} (a_\ell - a_{\ell-1})^{r+1}.$$

Mit $a_\ell - a_{\ell-1} = (b-a)/N$ folgt

$$\begin{aligned} |I(f) - Q^N(f)| &\leq \sum_{\ell=1}^N \left| \int_{a_{\ell-1}}^{a_\ell} f \, dx - Q_\ell(f) \right| \\ &\leq \sum_{\ell=1}^N (1 + \|Q_\ell\|) \frac{(b-a)^{r+2}}{N^{r+2}} \frac{\|f^{(r+1)}\|_{C^0([a_{\ell-1}, a_\ell])}}{(r+1)!} \\ &\leq (1 + \max_{\ell=1,\dots,N} \|Q_\ell\|) N \frac{(b-a)^{r+2}}{N^{r+2}} \frac{\|f^{(r+1)}\|_{C^0([a,b])}}{(r+1)!}. \end{aligned}$$

Dies impliziert die behauptete Abschätzung. \square

Definition 14.4 Eine summierte Quadraturformel Q^N heißt *konvergent von der Ordnung $s \geq 0$* , falls

$$|Q^N(f) - I(f)| = \mathcal{O}(h^s)$$

für alle $f \in C^s([a, b])$ und $h = (b - a)/N \rightarrow 0$ gilt. In den Fällen $s = 1, 2, 3$ bezeichnet man dies als *lineare, quadratische* beziehungsweise *kubische* Konvergenz.

Beispiele 14.3 (i) Die summierte Trapezregel ist quadratisch konvergent.

(ii) Die summierte Simpson-Regel besitzt die Konvergenzordnung $s = 4$.

14.4 Gauß-Quadratur

Die Wahl von Quadraturpunkten und -gewichten beeinflusst die Genauigkeit einer Quadraturformel. Ein gewisser Exaktheitsgrad kann bei vorgegebener Anzahl von Punkten nicht überstiegen werden.

Lemma 14.1 Eine Quadraturformel mit $n + 1$ Gewichten und Quadraturpunkten besitzt den maximalen Exaktheitsgrad $2n + 1$.

Beweis Sei $Q(f) = \sum_{i=0}^n w_i f(x_i)$ und definiere $p(x) = \prod_{i=0}^n (x - x_i)^2$. Dann gilt $p \in \mathcal{P}_{2n+2}$ und p ist positiv außerhalb der Quadraturpunkte, in denen p verschwindet. Damit folgt $I(f) > 0$ sowie $Q(f) = 0$ und dies impliziert die Behauptung. \square

Wir zeigen im Folgenden, dass es tatsächlich eine Quadraturformel mit dem maximalen Exaktheitsgrad $2n + 1$ gibt. Ist eine Quadraturformel exakt vom Grad n , so sind die Gewichte bereits eindeutig festgelegt. Ist sie exakt vom Grad $2n$, so sind diese positiv.

Lemma 14.2 Eine Quadraturformel mit $n + 1$ Gewichten und Quadraturpunkten $(x_i, w_i)_{i=0,\dots,n}$ ist exakt vom Grad n genau dann, wenn gilt

$$w_i = \int_a^b L_i(x) dx$$

für $i = 0, 1, \dots, n$ mit den durch die Punkte $(x_i)_{i=0,\dots,n}$ definierten Lagrange-Basispolynomen $(L_i)_{i=0,\dots,n}$. Ist die Quadraturformel exakt vom Grad $2n$, so gilt $w_i > 0$ für $i = 0, 1, \dots, n$.

Beweis Übungsaufgabe. \square

Bei der Gauß-Quadratur werden $n + 1$ Quadraturpunkte so konstruiert, dass der maximale Exaktheitsgrad $2n + 1$ erzielt wird. Etwas allgemeiner werden dabei gewichtete Integrale der Form

$$I_\omega(f) = \int_a^b f(x)\omega(x) dx$$

mit einer nichtnegativen *Gewichtsfunktion* $\omega \in C^0(a, b)$ betrachtet. Diese Funktion sei so gewählt, dass durch

$$\langle f, g \rangle_\omega = \int_a^b f(x)g(x)\omega(x) dx$$

ein Skalarprodukt auf $C^0([a, b])$ definiert wird. Dies ist genau dann der Fall, wenn ω auf $[a, b]$ uneigentlich integrierbar ist und aus $\langle f, f \rangle_\omega = 0$ bereits $f = 0$ für jede Funktion $f \in C^0([a, b])$ folgt. Bezuglich dieses Skalarprodukts wird mit dem Gram–Schmidt-Verfahren eine Orthogonalbasis von \mathcal{P}_n bestimmt.

Satz 14.4 Es existieren Orthogonalpolynome $(\pi_j)_{j=0,\dots,n}$ derart, dass $\pi_j \in \mathcal{P}_j$ und $\langle \pi_j, \pi_k \rangle_\omega = \delta_{jk}$ für alle $0 \leq j, k \leq n$ mit $j \neq k$ gilt. Insbesondere gilt $\langle \pi_j, p \rangle_\omega = 0$ für alle $p \in \mathcal{P}_{j-1}$ und die Polynome bilden eine Basis von \mathcal{P}_n .

Beweis Übungsaufgabe. □

Lemma 14.3 Die Nullstellen jedes Orthogonalpolynoms π_j , $0 \leq j \leq n$, sind einfach, reell und liegen im Intervall (a, b) .

Beweis Die Aussage des Lemmas sei falsch für ein $j \in \{0, 1, \dots, n\}$. Hat π_j eine Nullstelle $z \in \mathbb{R} \setminus (a, b)$, so ist $p(x) = \pi_j(x)/(x - z)$ ein Polynom in \mathcal{P}_{j-1} und es folgt

$$0 = \langle \pi_j, p \rangle_\omega = \int_a^b \frac{\pi_j^2(x)}{x - z} \omega(x) dx,$$

was nicht möglich ist, da $x - z$ keine Nullstelle in (a, b) hat und π_j nicht identisch Null ist. Ist $z \in (a, b)$ eine mehrfache Nullstelle oder gilt $z \in \mathbb{C} \setminus \mathbb{R}$, so ist \bar{z} ebenfalls eine Nullstelle von π_j und es folgt $p(x) = \pi_j(x)/((x - z)(x - \bar{z})) = \pi_j(x)/|x - z|^2 \in \mathcal{P}_{j-2}$. Wiederum führt die Identität $0 = \langle \pi_j, p \rangle_\omega$ zu einem Widerspruch. □

Beispiele 14.4 (i) Für die Gewichtsfunktion $\omega(x) = (1 - x^2)^{-1/2}$ im Intervall $(-1, 1)$ ergeben sich die Tschebyscheff-Polynome.

(ii) Für $\omega(x) = 1$ im Intervall $[-1, 1]$ ergeben sich die Legendre-Polynome als Ableitungen der Ordnung n des Polynoms $(x^2 - 1)^n$, das heißt

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n.$$

Die Nullstellen des Orthogonalpolynoms π_n definieren eine Quadraturformel mit Exaktheitsgrad $2n + 1$.

Satz 14.5 Sei $\pi_{n+1} \in \mathcal{P}_{n+1}$ das $(n + 1)$ -te Orthogonalpolynom bezüglich der Gewichtsfunktion $\omega \in C^0(a, b)$. Durch die Nullstellen $(x_i)_{i=0, \dots, n}$ von π_{n+1} und die Gewichte

$$w_i = \int_a^b L_i(x) \omega(x) dx$$

für $i = 0, 1, \dots, n$ wird eine Quadraturformel $Q_\omega f = \sum_{i=0}^n w_i f(x_i)$ definiert, sodass

$$Q_\omega p = I_\omega p = \int_a^b p(x) \omega(x) dx$$

für alle $p \in \mathcal{P}_{2n+1}$ gilt.

Beweis Die im Satz definierte Quadraturformel ist wohldefiniert und nach Wahl der Gewichte gilt $I_\omega r = Q_\omega r$ für alle $r \in \mathcal{P}_n$. Ist $p \in \mathcal{P}_{2n+1}$, so erhält man durch Polynomdivision Polynome $q, r \in \mathcal{P}_n$ mit $p = q\pi_{n+1} + r$. Da $\langle q, \pi_{n+1} \rangle_\omega = 0$ gilt, folgt

$$I_\omega p = \int_a^b q(x) \pi_{n+1}(x) \omega(x) dx + \int_a^b r(x) \omega(x) dx = \langle q, \pi_{n+1} \rangle_\omega + I_\omega r = I_\omega r.$$

Mit $\pi_{n+1}(x_i) = 0, i = 0, 1, \dots, n$, folgt

$$Q_\omega p = \sum_{i=0}^n w_i (q(x_i) \pi_{n+1}(x_i) + r(x_i)) = \sum_{i=0}^n w_i r(x_i) = Q_\omega r.$$

Insgesamt also $I_\omega p = I_\omega r = Q_\omega r = Q_\omega p$. □

Beispiel 14.5 Für die Gewichtsfunktion $\omega(x) = 1$ im Intervall $[-1, 1]$ ist $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = (3x^2 - 1)/2$ und $P_3(x) = (5x^3 - 3x)/2$. Damit erhält man für $n = 0, 1, 2$ die durch

$$x_0 = 0, \quad w_0 = 2,$$

$$x_0 = -\sqrt{1/3}, \quad x_1 = \sqrt{1/3}, \quad w_0 = 1, \quad w_1 = 1,$$

$$x_0 = -\sqrt{3/5}, \quad x_1 = 0, \quad x_2 = \sqrt{3/5}, \quad w_0 = 5/9, \quad w_1 = 8/9, \quad w_2 = 5/9,$$

definierten Gaußschen Quadraturformeln.

14.5 Extrapolation

Eine summierte Quadraturformel definiert eine Funktion $T(h)$, die für eine gegebene Funktion $f \in C^0([a, b])$ und Zerlegungsfeinheiten $h = (b - a)/N$ eine Approximation des im Allgemeinen nicht direkt zugänglichen Integrals, welches mit $T(0)$ bezeichnet sei, definiert. Wir nehmen an, dass T als Funktion auf $\mathbb{R}_{\geq 0}$ gegeben ist, was beispielsweise durch geeignete Interpolation realisiert werden kann. Ist der Fehler der Quadraturformel von der Ordnung h^γ für ein $\gamma > 0$ und gilt $f \in C^\gamma([a, b])$, so folgt

$$T(h) = T(0) + \varphi(h^\gamma)$$

für eine Funktion $\varphi \in C^0(\mathbb{R})$ mit $\varphi(0) = 0$. Eine Taylorentwicklung von φ um 0, das heißt

$$\varphi(z) = c_1 z + c_2 z^2 + \cdots + c_k z^k + r_k(z)$$

mit im Allgemeinen unbekannten aber von h unabhängigen Koeffizienten c_i , $i = 1, 2, \dots, k$, und einem Restglied $r_k \in \mathcal{O}(z^{k+1})$ für $z \rightarrow 0$ führt auf

$$T(h) = T(0) + c_1 h^\gamma + c_2 h^{2\gamma} + \cdots + c_k h^{k\gamma} + r_k(h^\gamma).$$

Die Auswertung der Quadraturformel für die Feinheit $h/2$ ergibt

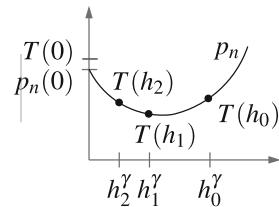
$$T(h/2) = T(0) + \frac{c_1}{2^\gamma} h^\gamma + \frac{c_2}{4^\gamma} h^{2\gamma} + \cdots + \frac{c_k}{2^{k\gamma}} h^{k\gamma} + r_k\left(\frac{h^\gamma}{2}\right).$$

Mit dieser Gleichung lässt sich der Term $c_1 h^\gamma$ in der Identität für $T(h)$ eliminieren und wir erhalten

$$T^*(h) = \frac{T(h) - 2^\gamma T(h/2)}{1 - 2^\gamma} = T(0) + c_2 \frac{1 - 2^{-\gamma}}{1 - 2^\gamma} h^{2\gamma} + \mathcal{O}(h^{3\gamma}).$$

Der berechenbare Ausdruck $T^*(h)$ definiert also eine Approximation von $T(0)$ mit einem Fehler der Ordnung $h^{2\gamma}$, was für kleine Werte von h akkurater ist als die Approximationen $T(h)$ und $T(h/2)$. Das Vorgehen ist in Abb. 14.3 skizziert.

Abb. 14.3 Extrapolation der berechneten Werte $T(h_i)$, $i = 0, 1, 2$, zur besseren Approximation des unbekannten Werts $T(0)$



Beispiel 14.6 Die Extrapolation der summierten Trapezregel mit Konvergenzordnung $s = 2$ führt auf die summierte Simpson-Regel mit Konvergenzordnung $s = 4$.

Das beschriebene Vorgehen lässt sich verallgemeinern, indem man eine Polynominterpolation der Stützwerte $T(h_i)$, $i = 0, 1, \dots, n$, durchführt. Dazu wird das Interpolationspolynom $p_n \in \mathcal{P}_n$ durch die Bedingungen

$$p_n(h_i^\gamma) = T(h_i)$$

für $i = 0, 1, \dots, n$ definiert. Der *extrapolierte Wert* $T(0) \approx p_n(0)$ kann mit Hilfe des Neville-Schemas bestimmt werden. Entsprechende Details finden sich beispielsweise in [1,7].

14.6 Experimentelle Konvergenzordnung

Die Konvergenzeigenschaften einer summierten Quadraturformel Q^N mit Schrittweite $h = (b - a)/N$ lassen sich experimentell analysieren, indem man für eine nicht-polynomielle Funktion $f \in C^s([a, b])$, deren exaktes Integral $I(f)$ bekannt ist, die Fehler

$$e_h = |I(f) - Q^N(f)|$$

für einige Schrittweiten $h > 0$ betrachtet. Aus dem Ansatz $e_h \approx c_1 h^\gamma$ folgt durch Verwendung von zwei verschiedenen Schrittweiten $h, H > 0$, dass

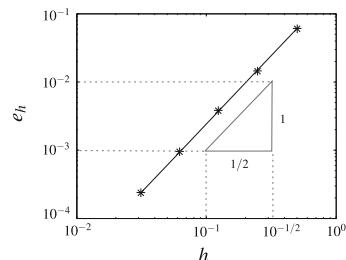
$$c_1 \approx \frac{e_h}{h^\gamma} \approx \frac{e_H}{H^\gamma}$$

und somit

$$\gamma \approx \frac{\log(e_h/e_H)}{\log(h/H)} = \frac{\log(e_h) - \log(e_H)}{\log(h) - \log(H)}.$$

Ist insbesondere $H = h/2$, so ergibt sich $\gamma = \log(e_h/e_{h/2})/\log(2)$. Berechnet man diesen Ausdruck für mehrere Schrittweiten h , so lässt sich eine (*mittlere*) *experimentelle Konvergenzordnung* durch eine Ausgleichsrechnung beziehungsweise das arithmetische Mittel

Abb. 14.4 Die experimentelle Konvergenzordnung ergibt sich als Steigung einer Ausgleichsgeraden durch Messpunkte bezüglich logarithmischer Skalierung



bestimmen. Man beachte jedoch, dass diese von der Differenzierbarkeitsordnung von f abhängen kann. Zusätzlich kann man das Konvergenzverhalten grafisch darstellen, indem man logarithmische Skalierungen der x - und y -Achsen verwendet und experimentell ermittelte Paare (h, e_h) über einen Polygonzug verbindet. Besteht tatsächlich ein Zusammenhang der Form $e_h \approx c_1 h^\gamma$, so wird der Polygonzug bezüglich der logarithmischen Skalierung die Steigung γ haben.

Beispiel 14.7 Wir betrachten die Wertepaare (h, e_h) die durch $h = 2^{-\ell}$, $\ell = 1, \dots, 5$, und $e_h = h^2/3$ gegeben sind. Eine Skizze zeigt, dass die Steigung des dadurch definierten Polygonzugs in logarithmischer Skalierung mit der Steigung der dazu parallelen Geraden durch die Punkte $(10^{-1}, 10^{-3})$ und $(10^{-1/2}, 10^{-2})$ übereinstimmt. Die logarithmische Steigung dieser Geraden ergibt sich aus dem Differenzenquotienten der Potenzen, das heißt

$$\gamma \approx \frac{\Delta_y^{\log}}{\Delta_x^{\log}} = \frac{(-2) - (-3)}{(-1/2) - (-1)} = 2.$$

Aufgrund der logarithmischen Skalierung befindet sich der Wert $10^{-1/2}$ auf der x -Achse exakt in der Mitte der Werte 10^0 und 10^{-1} , s. Abb. 14.4.

14.7 Lernziele, Quiz und Anwendung

Sie sollten den Exaktheitsgrad einer Quadraturformel definieren und darauf basierend abstrakte Fehlerabschätzungen herleiten können. Die Newton–Cotes–Formeln sollten Sie konkretisieren und an Beispielen anwenden können. Die Konstruktion der Gauß–Quadratur sollten Sie beschreiben und die Eigenschaften des Verfahrens benennen können.

Quiz 14.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Für jede Quadraturformel gilt $Q(\alpha f + \beta g) = \alpha Q(f) + \beta Q(g)$	
Ist eine Quadraturformel exakt vom Grad $r \geq 1$, so sind die Gewichte der Quadraturformel positiv	
Jede Newton-Cotes Formel mit $n+1 = 2q$ Stützstellen ist exakt vom Grad $n+2$	
Die Gauß-Quadratur verwendet die $n+1$ Nullstellen eines Orthogonalpolynoms $\pi_n \in \mathcal{P}_n$ als Quadraturpunkte	
Die Trapezformel auf dem Intervall $[-1, 1]$ approximiert das Integral der Funktion f durch $[f(-1) + f(1)]/2$	

Anwendung 14.1 Auf Basis von Stichproben und statistischen Erwägungen lässt sich das Gewicht eines Hühnereis in guter Näherung als normalverteilte Zufallsvariable X mit Erwartungswert $\mu = 57\text{ g}$ und Standardabweichung $\sigma = 7\text{ g}$ beschreiben. Die Wahrscheinlichkeit, dass das Gewicht eines Eis im Intervall $[m_1, m_2]$ liegt, ist damit gegeben durch

$$P(m_1 \leq X \leq m_2) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{m_1}^{m_2} e^{-(x-\mu)^2/(2\sigma^2)} dx.$$

- (i) Bestimmen Sie mit einer summierten Quadraturformel sowie der Identität

$$\int_0^\infty e^{-t^2/2} dt = \sqrt{\pi/2}$$

die Wahrscheinlichkeit, dass ein Ei mehr als 63 g wiegt.

- (ii) Vergleichen Sie Ihr Ergebnis mit einem Zugang der Berechnung der Wahrscheinlichkeit ohne numerische Integration mit der Identität

$$e^{-t^2} = 1 - t^2 + \frac{t^4}{2!} - \frac{t^6}{3!} + \dots$$

und der exakten Integration einiger Monome. In welchen Situationen ist dieser Zugang sinnvoll?

- (iii) Spezifizieren Sie numerisch mit vier Nachkommastellen Genauigkeit die sogenannte 68-95-99,7-Regel, die die Wahrscheinlichkeiten für die Abweichung um die ein-, zwei- beziehungsweise dreifache Standardabweichung vom Erwartungswert angibt, das heißt die Größen $P(|X - \mu| \leq j\sigma)$ für $j = 1, 2, 3$.

15.1 Nullstellensuche und Minimierungsprobleme

Für eine offene Menge $U \subset \mathbb{R}^n$ und Abbildungen $f : U \rightarrow \mathbb{R}^n$ und $g : U \rightarrow \mathbb{R}$ betrachten wir folgende Aufgaben:

- (N) Finde $x^* \in U$ mit $f(x^*) = 0$.
- (M) Finde $x^* \in U$ mit $g(x^*) = \min_{x \in U} g(x)$.

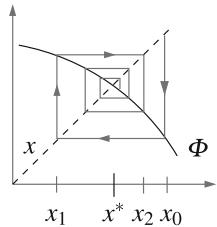
Diese Aufgabenstellungen sind über die Optimalitätsbedingung $\nabla g(x^*) = 0$ beziehungsweise über die Minimierung von $x \mapsto \|f(x)\|^2$ miteinander verbunden. Die Nullstellensuche ist darüber hinaus äquivalent zur Bestimmung eines Fixpunkts der Abbildung $\Phi(x) = f(x) + x$. Im Allgemeinen ist es weder möglich noch sinnvoll, eine Lösung exakt zu bestimmen und daher werden iterativ Folgen $(x_k)_{k=0,1,\dots}$ konstruiert, die unter geeigneten Voraussetzungen gegen eine Lösung konvergieren. Zur Klassifizierung des Konvergenzverhaltens werden folgende Begriffe verwendet.

Definition 15.1 Ein numerisches Verfahren, das eine Folge $(x_k)_{k=0,1,\dots}$ von Approximationen für eine numerische Aufgabe definiert, heißt

- (i) *global konvergent*, falls die Folge $(x_k)_{k=0,1,\dots}$ für jeden Startvektor $x_0 \in U$ gegen eine Lösung $x^* \in U$ konvergiert, und
- (ii) *lokal konvergent*, falls für jede Lösung $x^* \in U$ ein $\varepsilon > 0$ existiert, sodass die Folge $(x_k)_{k=0,1,\dots}$ für jeden Startvektor $x_0 \in B_\varepsilon(x^*) \cap U$ gegen x^* konvergiert.

Offensichtlich ist jedes global konvergente Verfahren auch lokal konvergent. Zur Charakterisierung der Konvergenzgeschwindigkeit von Verfahren nehmen wir an, dass $x_k \neq x^*$ für alle $k \in \mathbb{N}_0$ gilt.

Abb. 15.1 Beispiel einer lokal konvergenten Fixpunktiteration



Definition 15.2 Ein lokal konvergentes Verfahren heißt *konvergent von der Ordnung* $\alpha \geq 1$, falls ein $q \in \mathbb{R}$ existiert, sodass für jede Lösung $x^* \in U$, jeden Startvektor $x_0 \in B_\varepsilon(x^*) \cap U$ und die vom Verfahren generierte Folge $(x_k)_{k \in \mathbb{N}_0}$ für die Approximationfehler $\delta_k = \|x^* - x_k\|$

$$\limsup_{k \rightarrow \infty} \frac{\delta_{k+1}}{\delta_k^\alpha} = q$$

und im Fall $\alpha = 1$ zusätzlich $q \leq 1$ gilt. Ein Verfahren, das konvergent von der Ordnung α ist, heißt *linear konvergent* falls $\alpha = 1$ und $q < 1$ sowie *quadratisch konvergent* falls $\alpha = 2$ gilt. Es heißt *superlinear* oder *sublinear konvergent* falls $\alpha = 1$ und $q = 0$ beziehungsweise $\alpha = 1$ und $q = 1$ gelten.

Beispiele 15.1 (i) Ist $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine Kontraktion, so ist das Verfahren $x_{k+1} = \Phi(x_k)$ zur Approximation eines Fixpunkts von Φ global und linear konvergent.

(ii) Ist $\Phi \in C^1(\mathbb{R})$, so ist die Fixpunktiteration $x_{k+1} = \Phi(x_k)$ zur Bestimmung eines Fixpunkts x^* von Φ lokal konvergent, sofern $|\Phi'(x^*)| < 1$ gilt, s. Abb. 15.1. Gilt $|\Phi'(x^*)| > 1$, so ist das Verfahren divergent.

Bemerkung 15.1 Im sogenannten *asymptotischen Bereich*, das heißt nach hinreichend vielen Iterationen, gilt bei linearer Konvergenz eine Fehlerreduktion um den Faktor q , während sich bei quadratischer Konvergenz die Anzahl der korrekten Stellen in jedem Schritt verdoppelt.

15.2 Approximation von Nullstellen

Das Bisektionsverfahren basiert auf der Tatsache, dass jede stetige Funktion $f \in C^0([a, b])$ mit der Eigenschaft $f(a)f(b) \leq 0$ eine Nullstelle im Intervall $[a, b]$ besitzt. Ist $c \in (a, b)$ beliebig, so folgt

$$f(a)f(c) \leq 0 \quad \text{oder} \quad f(c)f(b) \leq 0$$

und das Teilintervall $[a, c]$ oder $[c, b]$ enthält mindestens eine Nullstelle von f , s. Abb. 15.2.

Abb. 15.2 Der Vorzeichenwechsel einer stetigen Funktion impliziert die Existenz einer Nullstelle

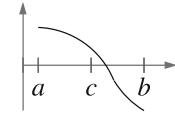
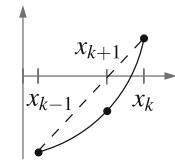


Abb. 15.3 Die Nullstelle der Sekante dient als Approximation einer Nullstelle



Algorithmus 15.1 (Bisektionsverfahren) Sei $f \in C^0([a, b])$ mit $f(a)f(b) \leq 0$ und $\varepsilon_{\text{stop}} > 0$. Setze $(a_0, b_0) = (a, b)$ und $k = 0$.

- (1) Definiere $c_k = (a_k + b_k)/2$.
- (2) Setze

$$(a_{k+1}, b_{k+1}) = \begin{cases} (a_k, c_k) & \text{falls } f(a_k)f(c_k) \leq 0, \\ (c_k, b_k) & \text{andernfalls.} \end{cases}$$

- (3) Stoppe falls $b_{k+1} - a_{k+1} \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Da das aktuelle Intervall in jedem Schritt halbiert wird, ergibt sich unmittelbar folgende Aussage.

Satz 15.1 Das Bisektionsverfahren bricht nach $J \leq 1 + \log_2 ((b - a)/\varepsilon_{\text{stop}})$ Schritten ab und das Intervall $[a_J, b_J]$ enthält eine Nullstelle.

Während das Bisektionsverfahren nur in einer Dimension sinnvoll ist, wird beim Sekantenverfahren eine Ableitung approximiert, was auch in mehreren Dimensionen möglich ist. Die einfach zu bestimmende Nullstelle der Sekante definiert den jeweils neuen Referenzpunkt, s. Abb. 15.3.

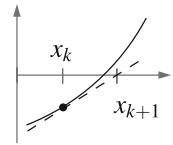
Algorithmus 15.2 (Sekantenverfahren) Sei $f \in C^0([a, b])$ mit $f(a)f(b) \leq 0$ und $\varepsilon_{\text{stop}} > 0$. Setze $x_0 = a$, $x_1 = b$ und $k = 1$.

- (1) Gilt $f(x_k) \neq f(x_{k-1})$, so definiere

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k).$$

- (2) Stoppe falls $|x_{k+1} - x_k| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Abb. 15.4 Die Nullstelle der Tangente dient als Approximation einer Nullstelle



Bemerkungen 15.2 (i) Bei der Realisierung des Sekantenverfahrens können Auslöschungseffekte auftreten und Rundungsfehler signifikant werden.

(ii) Ein alternatives Abbruchkriterium ist $|f(x_{k+1})| \leq \varepsilon_{\text{stop}}$.

(iii) Das *regula-falsi*-Verfahren kombiniert das Bisektions- mit dem Sekantenverfahren, sodass das Intervall $[x_{k-1}, x_k]$ stets eine Nullstelle enthält.

Die im Sekantenverfahren auftretende Größe $(f(x_k) - f(x_{k-1}))/(|x_k - x_{k-1}|)$ ist eine Approximation der Ableitung $f'(x_k)$. Diese Steigung definiert eine Tangente an f im Punkt $(x_k, f(x_k))$, die im Newton-Verfahren zur Bestimmung der neuen Approximation x_{k+1} verwendet wird, s. Abb. 15.4. Dies ist vor allem im Mehrdimensionalen einfacher zu realisieren, sofern die Jacobi-Matrix einfach bestimmt werden kann.

Eine Taylor-Approximation der C^1 -Abbildung $f : U \rightarrow \mathbb{R}^n$ um die Stelle x zeigt

$$0 = f(x^*) = f(x) + Df(x)(x^* - x) + \varphi(\|x^* - x\|).$$

Liegt die Approximation $x = x_k$ nahe bei x^* , so folgt unter Vernachlässigung des Terms $\varphi(\|x^* - x\|)$, dass

$$f(x_k) + Df(x_k)(x^* - x_k) \approx 0$$

und dies motiviert, dass durch

$$x_{k+1} = x_k - Df(x_k)^{-1} f(x_k) \approx x^*$$

eine verbesserte Approximation $x_{k+1} \approx x^*$ definiert wird, sofern $Df(x_k)$ regulär ist.

Algorithmus 15.3 (Newton-Verfahren) Seien $f \in C^1(U; \mathbb{R}^n)$, $x_0 \in U$ und $\varepsilon_{\text{stop}} > 0$. Setze $k = 0$.

(1) Ist $Df(x_k)$ regulär, so definiere

$$x_{k+1} = x_k - Df(x_k)^{-1} f(x_k).$$

(2) Stoppe falls $\|x_{k+1} - x_k\| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Das Newton-Verfahren ist lokal quadratisch konvergent.

Satz 15.2 Sei $f \in C^2(U; \mathbb{R}^n)$ und $x^* \in U$ eine Nullstelle von f in U , sodass $Df(x^*)$ regulär ist. Dann existiert ein $\varepsilon > 0$, sodass für jeden Startwert $x_0 \in B_\varepsilon(x^*) \cap U$ das Newton-Verfahren durchführbar und konvergent ist. Für die Iterierten $(x_k)_{k=0,1,\dots}$ gilt

$$\|x^* - x_{k+1}\| \leq c \|x^* - x_k\|^2$$

mit einer Konstanten $c \geq 0$.

Beweis Da $\det Df(x^*) \neq 0$ gilt und die Abbildung $x \mapsto \det Df(x)$ stetig ist, existiert ein $\tilde{\varepsilon} > 0$, sodass $\det Df(x) \neq 0$ und $\|Df(x)^{-1}\| \leq c_1$ für alle $x \in B_{\tilde{\varepsilon}}(x^*) \subset U$ gilt. Es gelte $x_k \in B_{\tilde{\varepsilon}}(x^*)$ für ein $k \geq 0$. Die Taylor-Entwicklung

$$0 = f(x^*) = f(x_k) + Df(x_k)(x^* - x_k) + \varphi(\|x^* - x_k\|)$$

mit einer Funktion $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, die $\varphi(t) \leq c_2 t^2$ für alle $|t| \leq c_3$ erfüllt, impliziert

$$\|f(x_k) + Df(x_k)(x^* - x_k)\| \leq c_2 \|x^* - x_k\|^2.$$

Mit der Iterationsvorschrift erhalten wir

$$x^* - x_{k+1} = Df(x_k)^{-1} (f(x_k) + Df(x_k)(x^* - x_k)).$$

Damit folgt

$$\begin{aligned} \|x^* - x_{k+1}\| &= \|Df(x_k)^{-1} (f(x_k) + Df(x_k)(x^* - x_k))\| \\ &\leq \|Df(x_k)^{-1}\| \|f(x_k) + Df(x_k)(x^* - x_k)\| \leq c_1 c_2 \|x^* - x_k\|^2. \end{aligned}$$

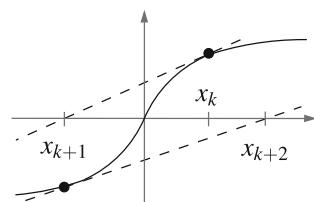
Mit $\varepsilon \leq \min\{1/(c_1 c_2), \tilde{\varepsilon}\}$ folgt, sofern $x_k \in B_\varepsilon(x^*)$ gilt, dass

$$\|x^* - x_{k+1}\| \leq c_1 c_2 \varepsilon \|x^* - x_k\| \leq \|x^* - x_k\| < \varepsilon \leq \tilde{\varepsilon}$$

und somit $x_{k+1} \in B_{\tilde{\varepsilon}}(x^*)$. Die Iteration ist daher wohldefiniert und konvergent, sofern $x_0 \in B_\varepsilon(x^*)$ gilt. \square

Bemerkung 15.3 Ist x_0 nicht nahe genug bei x^* , so kann Divergenz auftreten, s. Abb. 15.5. Dies lässt sich in vielen Fällen durch die Einführung einer Dämpfung, das heißt durch die Modifikation $x_{k+1} = x_k - \omega Df(x_k)^{-1} f(x_k)$ mit $0 < \omega < 1$, vermeiden. Im Allgemeinen gilt dann jedoch keine quadratische Konvergenz.

Abb. 15.5 Das Newton-Verfahren ist im Allgemeinen nur lokal konvergent



15.3 Eindimensionale Minimierung

Die globale Minimierung einer stetigen Funktion auf einer kompakten Menge ist ohne weitere Zusatzvoraussetzungen an die Funktion selten realisierbar. Man beschränkt sich daher in der Regel auf die Bestimmung lokaler Minimalstellen. Im Fall konvexer Funktionen sind diese bereits globale Minimalstellen.

Algorithmus 15.4 (Diskrete Suche) Sei $a = x_0 < x_1 < \dots < x_n = b$ eine Partitionierung in $n \geq 3$ Teilintervalle und $g \in C^0([a, b])$. Bestimme x_k mit $g(x_k) = \min\{g(x_1), g(x_2), \dots, g(x_{n-1})\}$, s. Abb. 15.6.

Satz 15.3 Ist x_k der von der diskreten Suche ermittelte Punkt, so enthält das Intervall $[x_{k-1}, x_{k+1}]$ eine lokale Minimalstelle $x_{loc}^* \in [a, b]$, das heißt es existiert ein $\delta > 0$ mit $g(x_{loc}^*) \leq g(x)$ für alle $x \in B_\delta(x_{loc}^*) \cap [a, b]$.

Beweis Auf dem kompakten Intervall $[x_{k-1}, x_{k+1}]$ nimmt die Funktion g ihr Minimum an. \square

Bei Intervallverkleinerungsverfahren wird die diskrete Suche auf kleiner werdende Intervalle angewendet, s. Abb. 15.7.

Algorithmus 15.5 (Intervallverkleinerung) Sei $g \in C^0([a, b])$ und $\varepsilon_{stop} > 0$. Setze $a_0 = a, b_0 = b$ sowie $k = 0$.

(1) Wähle $c_k, d_k \in (a_k, b_k)$ mit $a_k < c_k < d_k < b_k$ und setze

$$(a_{k+1}, b_{k+1}) = \begin{cases} (a_k, d_k) & \text{falls } g(c_k) \leq g(d_k), \\ (c_k, b_k) & \text{andernfalls.} \end{cases}$$

(2) Stoppe falls $b_{k+1} - a_{k+1} \leq \varepsilon_{stop}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Abb. 15.6 Das Minimum einer endlichen Menge von Funktionswerten liefert eine Approximation eines lokalen Minimums

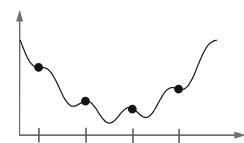
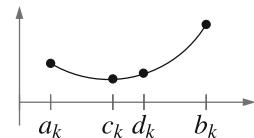


Abb. 15.7 Verkleinerung des Suchbereichs auf Basis mehrerer Funktionswerte



Bemerkung 15.4 Eine verbesserte Wahl der Punkte c_k und d_k führt auf eine gleichmäßige Verkleinerung der Intervalle und eine minimale Anzahl von Funktionsauswertungen.

15.4 Mehrdimensionale Minimierung

In mehreren Dimensionen werden in der Regel sukzessive eindimensionale Minimierungen entlang geeigneter Suchrichtungen durchgeführt. Eine sinnvolle Wahl der jeweiligen Suchrichtung ist die Richtung des steilsten Abstiegs, die durch den negativen Gradienten der zu minimierenden Funktion gegeben ist. Wir folgen den Darstellungen in [5,6].

Algorithmus 15.6 (Gradientenverfahren) Seien $g \in C^1(U)$, $x_0 \in U$, $\sigma \in (0, 1)$ und $\varepsilon_{\text{stop}} > 0$. Setze $k = 0$.

(1) Definiere $d_k = -\nabla g(x_k)$ und bestimme die maximale Zahl $\alpha_k \in \{2^{-\ell} : \ell \in \mathbb{N}_0\}$, für die die Armijo-Bedingung

$$g(x_k + \alpha_k d_k) \leq g(x_k) - \sigma \alpha_k \|d_k\|^2$$

erfüllt ist, s. Abb. 15.8.

(2) Setze $x_{k+1} = x_k + \alpha_k d_k$.

(3) Stoppe falls $\|\alpha_k d_k\| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Bemerkung 15.5 Das Verfahren führt in jedem Iterationsschritt eine diskrete Suche durch, um die Schrittweite α_k zu bestimmen. Die Existenz einer zulässigen Schrittweite folgt aus einer Übungsaufgabe. Der Parameter σ beschränkt die Schrittweite und kann bei C^2 -Funktionen auch als $\sigma = 1$ gewählt werden.

Bei der Analyse des Verfahrens wird das Abbruchkriterium ignoriert und die Konvergenz der Suchrichtungen $(d_k)_{k \in \mathbb{N}_0}$ gegen Null bewiesen.

Satz 15.4 Seien $g \in C^1(U)$, $x_0 \in U$ und $V \subset U$ offen, beschränkt und konvex, sodass

$$\tilde{V} = \{x \in U : g(x) \leq g(x_0)\} \subset V,$$

Abb. 15.8 Die Armijo-Bedingung garantiert eine vorgegebene relative Reduktion des Funktionswerts

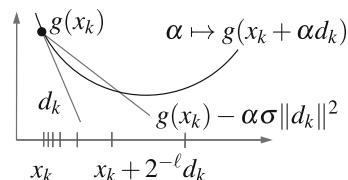
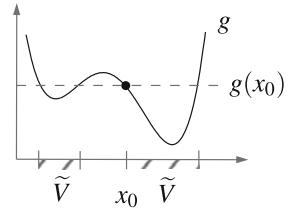


Abb. 15.9 Unterniveau-
menge \tilde{V} einer Funktion g
zum Niveau $g(x_0)$



s. Abb. 15.9. Mit $m = \max_{x \in \bar{V}} \|\nabla g(x)\|$ und

$$W = \{x + s : x \in V, s \in B_m(0)\}$$

gelte $g \in C^2(\overline{W})$. Dann folgt für die Iterierten des Gradientenverfahrens, dass $\nabla g(x_k) \rightarrow 0$ für $k \rightarrow \infty$ und $\alpha_k > (1 - \sigma)/\gamma$ mit $\gamma = \sup_{x \in W} \|D^2 f(x)\|$ gilt.

Beweis Die Folge $(g(x_k))_{k \in \mathbb{N}_0}$ ist monoton fallend, sodass $(x_k)_{k \in \mathbb{N}_0} \subset V$ und $g(x_k) \geq -c_0 = \min_{x \in \bar{V}} g(x)$ für alle $k \in \mathbb{N}_0$ gilt. Aus der Armijo-Bedingung folgt

$$\begin{aligned} g(x_0) &\geq g(x_1) + \sigma \alpha_0 \|\nabla g(x_0)\|^2 \\ &\geq g(x_2) + \sigma \alpha_1 \|\nabla g(x_1)\|^2 + \sigma \alpha_0 \|\nabla g(x_0)\|^2 \\ &\geq \dots \geq g(x_{\ell+1}) + \sigma \sum_{k=0}^{\ell} \alpha_k \|\nabla g(x_k)\|^2. \end{aligned}$$

Die Reihe $\sum_{k=0}^{\infty} \alpha_k \|\nabla g(x_k)\|^2$ ist somit konvergent und es gilt $\alpha_k \|\nabla g(x_k)\|^2 \rightarrow 0$. Es genügt daher zu zeigen, dass $\alpha_k \geq \delta > 0$ für alle $k \in \mathbb{N}_0$ und ein $\delta > 0$ gilt. Für jedes $k \in \mathbb{N}_0$ gilt entweder $\alpha_k = 1$ oder die Armijo-Bedingung ist für $2\alpha_k$ verletzt. Letzteres bedeutet

$$2\sigma \alpha_k \|\nabla g(x_k)\|^2 > g(x_k) - g(x_k + 2\alpha_k d_k).$$

Eine Taylor-Approximation impliziert, dass ein $\xi \in W$ existiert mit

$$g(x_k + 2\alpha_k d_k) = g(x_k) + \nabla g(x_k) \cdot (2\alpha_k d_k) + \frac{1}{2} (2\alpha_k)^2 D^2 g(\xi)[d_k, d_k].$$

Mit $d_k = -\nabla g(x_k)$ und $D^2 g(\xi)[d_k, d_k] \leq \gamma \|d_k\|^2$ folgt

$$2\sigma \alpha_k \|d_k\|^2 > 2\alpha_k \|d_k\|^2 - 2\gamma \alpha_k^2 \|d_k\|^2$$

beziehungsweise $(1 - \sigma)\alpha_k < \gamma \alpha_k^2$ und somit $\alpha_k > (1 - \sigma)/\gamma$ für alle $k \in \mathbb{N}_0$. \square

Bemerkungen 15.6 (i) Die Folge $(x_k)_{k \in \mathbb{N}_0}$ ist im Allgemeinen nicht konvergent. Gilt $x_k \rightarrow x_s$ für ein $x_s \in U$, so ist x_s ein stationärer Punkt von g , das heißt es gilt $\nabla g(x_s) = 0$, und x_s kann eine lokale Minimal- oder Maximalstelle oder ein Sattelpunkt sein. Lokale Maximalstellen und Sattelpunkte sind jedoch instabil im Hinblick auf Störungen, sodass das Gradientenverfahren in der Praxis meist gegen ein lokales Minimum konvergiert.

(ii) Der Beweis zeigt, dass die Werte $\|\nabla g(x_k)\|$ eine linear konvergente Folge bilden.

(iii) Die Bedingungen des Satzes sind erfüllt, sofern $g \in C^2(\mathbb{R}^n)$ gilt und g außerhalb einer kompakten Menge hinreichend groß ist.

15.5 Lernziele, Quiz und Anwendung

Ihnen sollten verschiedene Verfahren zur näherungsweisen Berechnung von Null- und Minimalstellen bekannt sein. Sie sollten die Verfahren motivieren und deren Eigenschaften darlegen können.

Quiz 15.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Konvergiert die Reihe $\sum_{k=0}^{\infty} \delta_k$, so ist die Folge $(\delta_k)_{k \geq 0}$ linear konvergent	
Die Folge $\delta_k = \sin^2(1/k)$, $k \in \mathbb{N}$, ist quadratisch konvergent gegen Null	
Das Gradientenverfahren mit einer Funktion g definiert eine konvergente Folge $(x_k)_{k \in \mathbb{N}_0}$, deren Grenzwert ein kritischer Punkt von g ist	
Hinreichend für die Konvergenz des Gradientenverfahrens ist, dass $g \in C^2(\mathbb{R}^n)$ gilt und g konvex ist	
Konvergiert das Newton-Verfahren, so gilt $\ f(x_k)\ \leq c \ x^* - x_k\ $ mit einer Konstanten $c \geq 0$ für alle $k \geq 0$	

Anwendung 15.1 Bei der Formoptimierung eines rotationssymmetrischen Trinkglases, dessen Grundfläche kreisförmig mit Durchmesser 3 cm ist, das 10 cm hoch ist und welches ein Volumen von etwa 0.21 ℓ besitzt, soll die Oberfläche minimiert werden. Die Form des Glases soll dabei durch eine kubische Kurve $s : [0, 10] \rightarrow \mathbb{R}$ beschrieben werden, sodass die Oberfläche des Glases durch

$$A(s) = 2\pi \int_0^{10} s(1 + |s'|^2)^{1/2} dx + \pi(3/2)^2$$

gegeben ist. Verwenden Sie die Stützstellen $0 = x_0 < x_1 < x_2 < x_3 = 10$, um die gesuchte Kurve mit den Werten $y_0 = 1.5$ und y_1, y_2, y_3 zu beschreiben. Für die Wahl $x_1 = 5$ und die Approximation des Volumens

$$V(s) = \pi \int_0^{10} (s(x))^2 dx$$

mit der Simpson-Regel lässt sich der Wert y_1 eliminieren. Formulieren Sie damit die Oberfläche als Funktion von y_2 und y_3 und minimieren Sie den resultierenden Ausdruck numerisch, indem Sie $A(s)$ geeignet diskretisieren.

16.1 Quadratische Minimierung

Ist $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, so ist die Lösung $x^* \in \mathbb{R}^n$ des Gleichungssystems $Ax = b$ die eindeutige Minimalstelle der Funktion

$$\phi(x) = \frac{1}{2} \|b - Ax\|_{A^{-1}}^2 = \frac{1}{2} (A^{-1}(b - Ax)) \cdot (b - Ax) \geq 0,$$

denn für jede symmetrische und positiv definite Matrix $B \in \mathbb{R}^{n \times n}$ wird durch $v \mapsto \|v\|_B = \sqrt{(Bv) \cdot v}$ eine Norm im \mathbb{R}^n definiert. Mit einer Variante des Abstiegsverfahren erhält man für eine Näherungslösung oder einen Startwert $\tilde{x} \in \mathbb{R}^n$ sowie eine Suchrichtung $\tilde{d} \in \mathbb{R}^n$ eine neue Approximation $\tilde{x} + \tilde{\alpha}\tilde{d}$ durch Minimierung von $\tilde{\psi} : t \mapsto \phi(\tilde{x} + t\tilde{d})$. Dabei gilt

$$\tilde{\psi}(t) = \phi(\tilde{x}) - t(b - A\tilde{x}) \cdot \tilde{d} + \frac{t^2}{2} (A\tilde{d}) \cdot \tilde{d}$$

und ein Differenzieren bezüglich t zeigt, dass das Minimum gegeben ist durch

$$\tilde{\alpha} = \frac{(b - A\tilde{x}) \cdot \tilde{d}}{(A\tilde{d}) \cdot \tilde{d}}.$$

Ist die Suchrichtung als negativer Gradient von ϕ bei \tilde{x} gewählt, das heißt

$$\tilde{d} = -\nabla\phi(\tilde{x}) = b - A\tilde{x},$$

so gilt für die neue Näherungslösung

$$\tilde{x}^{\text{neu}} = \tilde{x} + \tilde{\alpha}\tilde{d} = \tilde{x} + \tilde{\alpha}(b - A\tilde{x}),$$

was gerade einem Iterationsschritt eines Richardson-Verfahrens entspricht. Die wiederholte Durchführung dieser Strategie definiert eine Folge von Approximationslösungen $(x_k)_{k=0,1,\dots}$ für die eine Übungsaufgabe im Fall symmetrischer, positiv definiter Matrizen das Konvergenzverhalten

$$\|x_k - x^*\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|x_0 - x^*\|_A$$

mit $\kappa = \text{cond}_2(A)$ zeigt. Bei großen Konditionszahlen ergibt sich daher im Allgemeinen nur eine geringe Verbesserung in jedem Iterationsschritt.

16.2 Konjugierte Suchrichtungen

Die auftretenden Suchrichtungen im Abstiegsverfahren sind zwar paarweise orthogonal, führen aber nur zu langsamer Konvergenz. Eine starke Beschleunigung ergibt sich durch Verwendung sogenannter A -konjugierter Suchrichtungen. Im Folgenden wird stets vorausgesetzt, dass $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit ist. Wir folgen in diesem Kapitel der Darstellung in [6].

Definition 16.1 Die Vektoren $x, y \in \mathbb{R}^n$ heißen A -konjugiert, falls $x \cdot (Ay) = 0$ gilt.

Der Begriff der A -Konjugiertheit verallgemeinert den Begriff der Orthogonalität, denn orthogonale Vektoren sind A -konjugiert bezüglich $A = I_n$.

Lemma 16.1 Die Vektoren $d_0, d_1, \dots, d_k \in \mathbb{R}^n \setminus \{0\}$ seien paarweise A -konjugiert, das heißt es gelte $d_i \cdot Ad_j = 0$ für alle $0 \leq i, j \leq k$ mit $i \neq j$. Ist $x_0 \in \mathbb{R}^n$ und entsteht x_{j+1} aus x_j jeweils durch Minimierung von ϕ in Richtung von d_j , das heißt gilt

$$x_{j+1} = x_j + \alpha_j d_j = x_0 + \sum_{\ell=0}^j \alpha_\ell d_\ell,$$

$$\alpha_j = \frac{d_j \cdot (b - Ax_j)}{d_j \cdot Ad_j} = \frac{d_j \cdot (b - Ax_0)}{d_j \cdot Ad_j}$$

für $j = 1, 2, \dots, k$, so ist x_{j+1} das Minimum von ϕ in der Menge

$$x_0 + \text{span}\{d_0, d_1, \dots, d_j\}.$$

Beweis Für $j = 1, 2, \dots, k+1$ gilt $x_j \in x_0 + \text{span}\{d_0, d_1, \dots, d_{j-1}\}$ und mit der A -Konjugiertheit der Vektoren d_0, d_1, \dots, d_{j-1} folgt

$$d_j \cdot A(x_j - x_0) = 0$$

und somit $d_j \cdot (b - Ax_j) = d_j \cdot (b - Ax_0)$. Damit folgt

$$\begin{aligned}\phi(x_j + \alpha_j d_j) &= \phi(x_j) + \frac{\alpha_j^2}{2} d_j \cdot Ad_j - \alpha_j d_j \cdot (b - Ax_j) \\ &= \phi(x_j) + \frac{\alpha_j^2}{2} d_j \cdot Ad_j - \alpha_j d_j \cdot (b - Ax_0) \\ &= \phi(x_j) + \psi_j(\alpha_j)\end{aligned}$$

mit der quadratischen Funktion $\psi_j(t) = (t^2/2)d_j \cdot Ad_j - t d_j \cdot (b - Ax_0)$. Induktiv ergibt sich

$$\phi(x_j + \alpha_j d_j) = \phi\left(x_0 + \sum_{\ell=0}^j \alpha_\ell d_\ell\right) = \phi(x_0) + \sum_{\ell=0}^j \psi_\ell(\alpha_\ell).$$

Eine notwendige und hinreichende Bedingung für eine Minimalstelle in der Menge $x_0 + \text{span}\{d_0, \dots, d_j\}$ ist das Verschwinden der partiellen Ableitungen bezüglich der Koeffizienten α_i , $i = 0, 1, \dots, j$, das heißt

$$\frac{\partial}{\partial \alpha_i} \phi\left(x_0 + \sum_{\ell=0}^j \alpha_\ell d_\ell\right) = \psi'_i(\alpha_i) = 0$$

für $i = 0, 1, \dots, j$. Dies entspricht aber gerade der Wahl der Koeffizienten und damit ist die Aussage des Lemmas bewiesen. \square

Bemerkung 16.1 Das Lemma zeigt, dass die Koeffizienten $\alpha_1, \dots, \alpha_{n-1}$ unabhängig von einander bestimmt werden können, sofern die A -konjugierten Vektoren gegeben sind.

16.3 Berechnung A -konjugierter Richtungen

Die Bestimmung A -konjugierter Suchrichtungen erfolgt simultan mit der schrittweisen Verbesserung der Näherungslösungen. Für eine Approximation x_k ist das *Residuum* von x_k definiert durch

$$r_k = b - Ax_k.$$

Gilt $r_k = 0$, so löst x_k das Gleichungssystem $Ax = b$ und ist $x_{k+1} = x_k + \alpha_k d_k$, so gilt offenbar $r_{k+1} = r_k - \alpha_k Ad_k$.

Lemma 16.2 Für einen beliebigen Vektor $x_0 \in \mathbb{R}^n$ und $r_0 = b - Ax_0$ sowie $d_0 = r_0$ wird durch die Rekursion

$$\begin{aligned}r_{k+1} &= r_k - \alpha_k Ad_k, & d_{k+1} &= r_{k+1} - \beta_k d_k, \\ \alpha_k &= \frac{d_k \cdot r_k}{d_k \cdot Ad_k}, & \beta_k &= \frac{d_k \cdot Ar_{k+1}}{d_k \cdot Ad_k}\end{aligned}$$

eine Folge nichtverschwindender A -konjugierter Vektoren d_0, d_1, \dots, d_k bestimmt, bis $r_{k+1} = 0$ gilt. Für den durch $\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$ definierten Krylov-Raum gilt

$$\mathcal{K}_k(A, r_0) = \text{span}\{d_0, d_1, \dots, d_{k-1}\} = \text{span}\{r_0, r_1, \dots, r_{k-1}\}$$

und r_k ist orthogonal zu diesen Räumen.

Beweis Wir beweisen, dass die Vektoren d_0, d_1, \dots, d_{k-1} A -konjugiert sind und

$$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, r_1, \dots, r_{k-1}\} = \text{span}\{d_0, d_1, \dots, d_{k-1}\}$$

gilt. Für $k = 1$ sind die Aussagen klar und wir nehmen an, dass sie für ein $k \geq 1$ gelten. Wegen $r_{k-1} \in \mathcal{K}_k(A, r_0) \subset \mathcal{K}_{k+1}(A, r_0)$ sowie $d_{k-1} \in \mathcal{K}_k(A, r_0)$ und daher $Ad_{k-1} \in \mathcal{K}_{k+1}(A, r_0)$ folgt

$$r_k = r_{k-1} - \alpha_{k-1} Ad_{k-1} \in \mathcal{K}_{k+1}(A, r_0).$$

Die A -Konjugiertheit von d_0, d_1, \dots, d_{k-1} , die Identität $x_k = x_0 + \sum_{i=0}^{k-1} \alpha_i d_i$ und die Wahl von α_ℓ zeigen für $0 \leq \ell \leq k-1$, dass

$$d_\ell \cdot r_k = d_\ell \cdot (b - Ax_k) = d_\ell \cdot (b - Ax_0) - \alpha_\ell d_\ell \cdot Ad_\ell = 0.$$

Dies impliziert $r_k \perp \text{span}\{d_0, d_1, \dots, d_{k-1}\} = \mathcal{K}_k(A, r_0)$. Ist $r_k \neq 0$ also

$$\mathcal{K}_k(A, r_0) \subsetneq \text{span}\{r_0, r_1, \dots, r_k\} \subset \mathcal{K}_{k+1}(A, r_0),$$

so folgt aus Dimensionsgründen $\text{span}\{r_0, r_1, \dots, r_k\} = \mathcal{K}_{k+1}(A, r_0)$. Mit $d_k = r_k - \beta_{k-1} d_{k-1}$ erhalten wir zudem $\mathcal{K}_{k+1}(A, r_0) = \text{span}\{d_0, d_1, \dots, d_k\}$. Zum Nachweis der A -Konjugiertheit von d_0, d_1, \dots, d_k folgern wir mit der Definition von β_{k-1} und $d_k = r_k - \beta_{k-1} d_{k-1}$, dass

$$d_{k-1} \cdot Ad_k = d_{k-1} \cdot Ar_k - \frac{d_{k-1} \cdot Ar_k}{d_{k-1} \cdot Ad_{k-1}} d_{k-1} \cdot Ad_{k-1} = 0.$$

Mit der Orthogonalität $r_k \perp \text{span}\{d_0, d_1, \dots, d_{k-1}\}$, der A -Konjugiertheit der Vektoren $\{d_0, d_1, \dots, d_{k-1}\}$ und $d_k = r_k - \beta_{k-1} d_{k-1}$ sowie $Ad_\ell \in \mathcal{K}_k(A, r_0) \perp r_k$ für $0 \leq \ell \leq k-2$ erhalten wir, dass

$$d_\ell \cdot Ad_k = d_\ell \cdot Ar_k - \beta_{k-1} d_\ell \cdot Ad_{k-1} = Ad_\ell \cdot r_k - \beta_{k-1} d_\ell \cdot Ad_{k-1} = 0.$$

Damit sind die Aussagen des Lemmas bewiesen. \square

16.4 CG-Verfahren

Zur effizienten Realisierung des iterativen Verfahrens mit A -konjugierten Suchrichtungen bemerken wir, dass die Orthogonalität von r_k und d_{k-1} , das heißt $r_k \cdot d_{k-1} = 0$, die Gleichung

$$\alpha_k = \frac{d_k \cdot r_k}{d_k \cdot Ad_k} = \frac{(r_k - \beta_{k-1} d_{k-1}) \cdot r_k}{d_k \cdot Ad_k} = \frac{\|r_k\|^2}{d_k \cdot Ad_k}$$

impliziert. Aus $r_k \in \mathcal{K}_{k+1}(A, r_0) = \text{span}\{d_0, d_1, \dots, d_k\} \perp r_{k+1}$ folgt

$$d_k \cdot Ar_{k+1} = (Ad_k) \cdot r_{k+1} = \frac{1}{\alpha_k} (r_k - r_{k+1}) \cdot r_{k+1} = -\frac{d_k \cdot Ad_k}{\|r_k\|^2} \|r_{k+1}\|^2$$

und somit

$$\beta_k = \frac{d_k \cdot Ar_{k+1}}{d_k \cdot Ad_k} = -\frac{\|r_{k+1}\|^2}{\|r_k\|^2}.$$

Mit diesen Identitäten lässt sich das Verfahren der konjugierten Gradienten beziehungsweise die *conjugate gradient method* wie folgt umsetzen.

Algorithmus 16.1 (CG-Verfahren) Seien $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ und $\varepsilon_{\text{stop}} > 0$. Definiere $d_0 = r_0 = b - Ax_0$ und $k = 0$.

(1) Setze $x_{k+1} = x_k + \alpha_k d_k$, $r_{k+1} = r_k - \alpha_k Ad_k$ und $d_{k+1} = r_{k+1} - \beta_k d_k$ mit

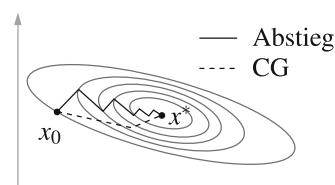
$$\alpha_k = \frac{\|r_k\|^2}{d_k \cdot Ad_k}, \quad \beta_k = -\frac{\|r_{k+1}\|^2}{\|r_k\|^2}.$$

(2) Stoppe falls $\|r_{k+1}\| \leq \varepsilon_{\text{stop}}$ gilt; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Bemerkung 16.2 Der Algorithmus terminiert nach höchstens n Schritten, da $r_n \perp \text{span}\{d_0, d_1, \dots, d_{n-1}\}$ gilt und die Vektoren d_0, d_1, \dots, d_{n-1} linear unabhängig sind, sofern nicht schon $r_k = 0$ für ein $0 \leq k \leq n - 1$ gilt. Insbesondere erhält man mit maximal n Schritten die exakte Lösung des linearen Gleichungssystems.

Der Unterschied des CG-Verfahrens zum Abstiegsverfahren ist in Abb. 16.1 schematisch illustriert.

Abb. 16.1 Das CG-Verfahren benötigt in vielen Fällen weniger Iterationsschritte als das Abstiegsverfahren



16.5 Konvergenz des CG-Verfahrens

In vielen Fällen liefert das CG-Verfahren bereits nach wenigen Schritten eine gute Approximation der Lösung des Gleichungssystems.

Satz 16.1 Für die Iterierten x_0, x_1, \dots des CG-Verfahrens und die Lösung x^* des Gleichungssystems $Ax = b$ gilt mit $\kappa = \text{cond}_2(A)$

$$\|x^* - x_k\|_A \leq 2 \left(\frac{\kappa^{1/2} - 1}{\kappa^{1/2} + 1} \right)^k \|x^* - x_0\|_A.$$

Beweis Wegen $Ax^* = b$ gilt $r_k = b - Ax_k = A(x^* - x_k)$ und mit der Minimalitätseigenschaft der Iterierten folgt

$$\begin{aligned} \|x^* - x_k\|_A^2 &= (A(x^* - x_k)) \cdot (x^* - x_k) = (A(x^* - x_k)) \cdot A^{-1}(A(x^* - x_k)) \\ &= \|b - Ax_k\|_{A^{-1}}^2 = 2\phi(x_k) = \min_{y \in x_0 + \text{span}\{d_0, \dots, d_{k-1}\}} 2\phi(y) \\ &= \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} 2\phi(y) = \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ay\|_{A^{-1}}^2 \\ &= \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} \|Ax^* - Ay\|_{A^{-1}}^2 = \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} \|x^* - y\|_A^2. \end{aligned}$$

Für jedes $y \in x_0 + \mathcal{K}_k(A, r_0)$ existiert ein Vektor $c = [c_1, c_2, \dots, c_k]^\top \in \mathbb{R}^k$ mit

$$\begin{aligned} y &= x_0 + c_1 A^0 r_0 + c_2 A^1 r_0 + \cdots + c_k A^{k-1} r_0 \\ &= x_0 + c_1 A(x^* - x_0) + \cdots + c_k A^k (x^* - x_0), \end{aligned}$$

wobei wir $r_0 = A(x^* - x_0)$ verwendet haben. Bezeichnet \mathcal{P}_k den Raum der Polynome vom maximalen Grad k , so folgt

$$\begin{aligned} \|x^* - x_k\|_A^2 &= \min_{c \in \mathbb{R}^k} \|x^* - x_0 - c_1 A(x^* - x_0) - \cdots - c_k A^k (x^* - x_0)\|_A^2 \\ &= \min_{c \in \mathbb{R}^k} \|(I_n - c_1 A - c_2 A^2 - \cdots - c_k A^k)(x^* - x_0)\|_A^2 = \min_{p \in \mathcal{P}_k, p(0)=1} \|p(A)(x^* - x_0)\|_A^2. \end{aligned}$$

Da A symmetrisch und positiv definit ist, existieren Eigenwerte $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ und zugehörige orthonormale Eigenvektoren $v_1, v_2, \dots, v_n \in \mathbb{R}^n$. Mit geeigneten Koeffizienten $\gamma_1, \gamma_2, \dots, \gamma_n$ gilt

$$x^* - x_0 = \sum_{i=1}^n (v_i \cdot (x^* - x_0)) v_i = \sum_{i=1}^n \gamma_i v_i$$

und

$$\|x^* - x_0\|_A^2 = (A(x^* - x_0)) \cdot (x^* - x_0) = \left(\sum_{i=1}^n \lambda_i \gamma_i v_i \right) \cdot \left(\sum_{j=1}^n \gamma_j v_j \right) = \sum_{i=1}^n \lambda_i \gamma_i^2.$$

Für jedes Polynom $p \in \mathcal{P}_k$ mit $p(0) = 1$ folgt unter Verwendung von $p(A)v_i = p(\lambda_i)v_i$, dass

$$\begin{aligned} \|p(A)(x^* - x_0)\|_A^2 &= \left\| \sum_{i=1}^n \gamma_i p(A)v_i \right\|_A^2 = \left\| \sum_{i=1}^n \gamma_i p(\lambda_i)v_i \right\|_A^2 \\ &= \left(\sum_{i=1}^n \gamma_i p(\lambda_i) A v_i \right) \cdot \left(\sum_{j=1}^n \gamma_j p(\lambda_j) v_j \right) = \sum_{i=1}^n \gamma_i^2 |p(\lambda_i)|^2 \lambda_i \\ &\leq \max_{i=1,\dots,n} |p(\lambda_i)|^2 \sum_{i=1}^n \gamma_i^2 \lambda_i = \max_{i=1,\dots,n} |p(\lambda_i)|^2 \|x^* - x_0\|_A^2. \end{aligned}$$

Im Fall $\lambda_1 = \lambda_2 = \dots = \lambda_n$ können wir ein Polynom $p \in \mathcal{P}_k$ mit $p(0) = 1$ finden, sodass $p(\lambda_i) = 0$ für $i = 1, 2, \dots, n$ gilt und die Aussage des Satzes ist bewiesen. Wir nehmen im Folgenden an, dass $\lambda_1 < \lambda_n$ gilt. Mit dem k -ten Tschebyscheff-Polynom $T_k \in \mathcal{P}_k$ dessen Nullstellen im Intervall $[-1, 1]$ enthalten sind und in diesem Intervall durch $T_k(s) = \cos(k \arccos(s))$ gegeben ist, setzen wir

$$q(t) = T_k\left(\frac{\lambda_n + \lambda_1 - 2t}{\lambda_n - \lambda_1}\right) / T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right).$$

Dann gilt $q \in \mathcal{P}_k$ mit $q(0) = 1$. Für $t \in [\lambda_1, \lambda_n]$ ist $(\lambda_n + \lambda_1 - 2t)/(\lambda_n - \lambda_1) \in [-1, 1]$ und aus $\max_{s \in [-1, 1]} |T_k(s)| \leq 1$ folgt

$$\max_{i=1,\dots,n} |q(\lambda_i)| \leq \left[T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right) \right]^{-1} = \left[T_k\left(\frac{\lambda_n/\lambda_1 + 1}{\lambda_n/\lambda_1 - 1}\right) \right]^{-1}.$$

Eine Übungsaufgabe zeigt

$$T_k\left(\frac{s+1}{s-1}\right) \geq \frac{1}{2} \frac{(s^{1/2} + 1)^k}{(s^{1/2} - 1)^k}$$

für $s > 1$ und damit folgt mit $\kappa = \lambda_n/\lambda_1$, dass

$$\|x^* - x_k\|_A \leq \max_{i=1,\dots,n} |q(\lambda_i)| \|x^* - x_0\|_A \leq 2 \frac{(\kappa^{1/2} - 1)^k}{(\kappa^{1/2} + 1)^k} \|x^* - x_0\|_A.$$

Dies beweist die Behauptung. \square

Beispiel 16.1 Gilt $\kappa = 100$, so erhält man beim CG-Verfahren eine Fehlerreduktion um $q \approx 0.8$ in jedem Iterationsschritt und es werden etwa 20 Schritte benötigt, um auf 1% des Anfangsfehlers zu kommen. Beim Abstiegsverfahren ergibt sich $q \approx 0.98$ und es sind mehr als 200 Schritte erforderlich.

Bemerkung 16.3 Die Konditionszahl ist für zwei Aspekte der numerischen Mathematik relevant. Einerseits beschreibt sie die Auswirkungen von Störungen auf die Lösung eines linearen Gleichungssystems und andererseits gibt sie an, wie viele Schritte bei der nähungsweisen iterativen Lösung eines Gleichungssystems erforderlich sind. Im ersten Fall ist die Wahl von Normen, die die Konditionszahl festlegen, meist durch die Anwendung vorgegeben, während im zweiten Fall die durch die Spektralnorm induzierte Konditionszahl von Interesse ist.

16.6 Lernziele, Quiz und Anwendung

Sie sollten den Begriff konjugierter Suchrichtungen erläutern und deren Bedeutung bei der iterativen Lösung linearer Gleichungssysteme beschreiben können. Hinreichende Bedingungen für die Konvergenz des CG-Verfahrens sollten Sie angeben können. Ferner sollten Sie vergleichende Aufwandsbetrachtungen durchführen können.

Quiz 16.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Mit der Cholesky-Zerlegung $A = LL^\top$ der Matrix A gilt $\ x\ _A = \ Lx\ $	
Ist $x_0 = x^*$ die Lösung von $Ax = b$, so sind die Krylov-Räume trivial, das heißt $\mathcal{K}_k = \{0\}$ für $k = 1, 2, \dots, n$	
Für $x \in \mathbb{R}^n$ und $A \in \mathbb{R}^{n \times n}$ gilt $\ x\ _A = \ Ax\ _{A^{-1}}$	
Ein Iterationsschritt des CG-Verfahrens erfordert einen Aufwand von $\mathcal{O}(n^2)$ Operationen	
Nichtverschwindende, paarweise A -konjugierte Vektoren sind linear unabhängig	

Anwendung 16.1 Zur einfachen mathematischen Beschreibung eines zweidimensionalen Diffusionsprozesses betrachten wir ein Gitter auf dem Gebiet $[0, 1]^2$ mit Gitterpunkten $x_{ij} = (i, j)h$, $i, j = 0, 1, \dots, n$ und Gitterweite $h = 1/n$. Es bezeichne u_{ij}^k die Konzentration einer Substanz in der Nähe des Gitterpunkts x_{ij} zum Zeitpunkt t_k , das heißt der Quotient der Menge von Partikeln der betrachteten Substanz im Bereich $x_{ij} + [-h/2, h/2]^2$ und des Volumens h^2 . Die Wahrscheinlichkeit, dass ein Partikel innerhalb

des Zeitintervalls $[t_k, t_{k+1}]$ der Länge τ aus der Umgebung eines Gitterpunkts in die Umgebung eines Nachbarpunkts springt sei mit p bezeichnet. Diese ist proportional zur Länge h der Grenzfläche, zur Länge τ des Zeitintervalls, zur normierten Anzahl h^{-2} der Partikel und invers proportional zur mittleren Distanz h , das heißt mit einer Diffusionskonstanten $c > 0$ gilt

$$p = c \frac{\tau}{h^2}.$$

Sinnvollerweise sollte $p \leq 1/4$ gelten. An den Randpunkten werde die Konzentration durch Entnahme oder Hinzufügen von Substanz auf Null gehalten. Die Größe f_{ij}^{k+1} bezeichne die in der Umgebung eines inneren Gitterpunkts x_{ij} im Zeitintervall $[t_k, t_{k+1}]$ hinzugefügte beziehungsweise entnommene Menge relativ zum Volumen h^2 . Für die Konzentration zum Zeitpunkt t_{k+1} gilt somit

$$u_{ij}^{k+1} = (1 - 4p)u_{ij}^k - p(u_{i-1,j}^k + u_{i+1,j}^k + u_{i,j-1}^k + u_{i,j+1}^k) + \tau f_{ij}^{k+1}$$

für innere Gitterpunkte und $u_{ij}^{k+1} = 0$ für Gitterpunkte auf dem Rand von $[0, 1]^2$. Ist die Gitterfunktion f_{ij}^k zeitlich konstant, so wird sich nach einer gewissen Zeit ein Gleichgewicht einstellen, das heißt es gilt $u_{ij}^{k+1} \approx u_{ij}^k$ für alle $0 \leq i, j \leq n$ und alle $k \geq K$.

- (i) Zeigen Sie, dass sich der Gleichgewichtszustand des Diffusionsprozesses als Lösung eines linearen Gleichungssystems mit symmetrischer und positiv definiter Systemmatrix bestimmen lässt.
- (ii) Untersuchen Sie experimentell die Abhängigkeit der Konditionszahl der Matrix A von h und bestimmen Sie die notwendige Anzahl von Iterationsschritten des Abstiegs- und des CG-Verfahrens, um eine Genauigkeit $\varepsilon_{\text{stop}} = h$ zu erzielen.
- (iii) Lösen Sie das lineare Gleichungssystem approximativ mit dem CG-Verfahren und stellen Sie die Approximationslösung mit Hilfe der MATLAB-Kommandos `meshgrid` und `surf` für den Fall $f_{ij} = 1$ grafisch dar.

17.1 Dünnbesetzte Matrizen

Das CG-Verfahren erfordert in jedem Iterationsschritt eine Matrix-Vektor-Multiplikation und ist daher besonders effizient, wenn dies mit wenig Aufwand verbunden ist. Dies ist der Fall, wenn nur wenige Einträge der Systemmatrix von Null verschieden sind. Im Folgenden repräsentiert die Matrix $A \in \mathbb{R}^{n \times n}$ stets eine Folge $(A_\ell)_{\ell \in \mathbb{N}}$ mit $A_\ell \in \mathbb{R}^{n_\ell \times n_\ell}$ mit $n_\ell \rightarrow \infty$ für $\ell \rightarrow \infty$.

Definition 17.1 Die Matrix $A \in \mathbb{R}^{n \times n}$ heißt *dünnbesetzt*, falls für die Anzahl der von Null verschiedenen Einträge $N_{nz} = |\{(i, j) : 1 \leq i, j \leq n, a_{ij} \neq 0\}|$ gilt $N_{nz} = \mathcal{O}(n)$. Der Index *nz* steht dabei für *not zero*.

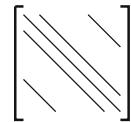
Beispiel 17.1 Bandmatrizen $A \in \mathbb{R}^{n \times n}$ mit einer von n unabhängigen Anzahl $k \in \mathbb{N}$ nichtverschwindender Nebendiagonalen, das heißt $a_{ij} \neq 0$ impliziert $|i - j| \in \{d_1, d_2, \dots, d_k\}$ mit Zahlen $d_r \in \mathbb{N}_0$, $r = 1, 2, \dots, k$, sind dünnbesetzt. Die *Bandweite* ist gegeben durch $w = \max_{r=1, \dots, k} d_r$, s. Abb. 17.1.

Um Speicherplatz zu sparen, werden dünnbesetzte Matrizen nicht als $n \times n$ -Arrays abgespeichert. Stattdessen werden beispielsweise Listen $I, J \in \mathbb{N}^{N_{nz}}$ und $X \in \mathbb{R}^{N_{nz}}$ verwendet, die die Positionen und Werte der von Null verschiedenen Einträge von A enthalten, das heißt es gilt

$$a_{ij} \neq 0 \iff \exists 1 \leq k \leq N_{nz}, (i, j) = (I_k, J_k), a_{ij} = X_k.$$

Diese Darstellung heißt *Koordinatendarstellung*. Tritt allgemeiner eine Position (i, j) wiederholt in den Indexlisten auf, so werden in der Regel die zugehörigen Werte aufsummiert. Der Speicheraufwand wird im *Compressed-Column-Storage (CCS)*-Format weiter reduziert, indem I und X wie oben definiert werden und der ℓ -te Eintrag einer Liste

Abb. 17.1 Schematische Darstellung einer Bandmatrix mit wenigen von Null verschiedenen Einträgen



$\tilde{J} \in \mathbb{N}_0^n$ spezifiziert, ab welcher Position in I und X die Einträge der ℓ -ten Spalte beginnen.

Beispiel 17.2 Für die nachfolgend definierte Matrix $A \in \mathbb{R}^{4 \times 4}$ ergeben sich die nebenstehenden Listen I, J, X und \tilde{J} :

$$A = \begin{bmatrix} 7 & 0 & 1 & 0 \\ 0 & 8 & 0 & 3 \\ 4 & 0 & 5 & 0 \\ 2 & 0 & 0 & 3 \end{bmatrix}, \quad I = [1, 3, 4, 2, 1, 3, 2, 4]^\top, \quad \tilde{J} = [1, 4, 5, 7]^\top. \\ J = [1, 1, 1, 2, 3, 3, 4, 4]^\top, \\ X = [7, 4, 2, 8, 1, 5, 3, 3]^\top,$$

Die Matrix-Vektor-Multiplikation mit einer Matrix im CCS-Format lässt sich einfach realisieren.

Bemerkung 17.1 Der Vektor $y = Az$ wird berechnet durch:

$$y = 0; \quad \text{for } \ell = 1 : N_{nz}; \quad y_{I(\ell)} = y_{I(\ell)} + X(\ell)z_{J(\ell)}; \quad \text{end}$$

17.2 Vorkonditioniertes CG-Verfahren

Die Anzahl der benötigten Iterationen des CG-Verfahrens zur approximativen Lösung des Gleichungssystems $Ax = b$ hängt von der Konditionszahl der symmetrischen und positiv definiten Systemmatrix A ab. Durch Wahl einer geeigneten invertierbaren Matrix $C \in \mathbb{R}^{n \times n}$ ist es jedoch naheliegend, das äquivalente System

$$(CA)x = Cb$$

zu betrachten. Gilt $\text{cond}(CA) \ll \text{cond}(A)$, so ist davon auszugehen, dass sich diese Umformulierung schneller und stabiler lösen lässt, sofern die Matrix C eine einfache Struktur hat, sodass sich die Multiplikation mit C effizient realisieren lässt. Bezuglich der Konditionszahl wäre die Wahl $C = A^{-1}$ optimal, jedoch wäre dann die Multiplikation mit C äquivalent zur Lösung des Ausgangsproblems $Ax = b$.

Definition 17.2 Sei $A \in \mathbb{R}^{n \times n}$ regulär. Eine reguläre Matrix $C \in \mathbb{R}^{n \times n}$ heißt *Vorkonditionierungsmatrix* für A , falls $\text{cond}(CA) \leq \text{cond}(A)$ gilt und der Rechenaufwand der Matrix-Vektor-Multiplikation $z \mapsto Cz$ geringer ist als die direkte Lösung des Gleichungssystems $Ax = b$.

Eine einfache Art der Vorkonditionierung ist die *Zeilenäquilibrierung*.

Satz 17.1 Sei $A \in \mathbb{R}^{n \times n}$ regulär und die Diagonalmatrix $C \in \mathbb{R}^{n \times n}$ für $i = 1, 2, \dots, n$ definiert durch

$$C_{ii} = \left(\sum_{j=1}^n |a_{ij}| \right)^{-1}.$$

Dann ist C ein Vorkonditionierungsmatrix für A bezüglich der Zeilensummennorm.

Beweis Die Matrix $B = CA$ erfüllt $\sum_{j=1}^n |b_{ij}| = 1$ für alle $i = 1, 2, \dots, n$, und folglich $\|B\|_\infty = 1$. Für jede Diagonalmatrix $T \in \mathbb{R}^{n \times n}$ folgt

$$\|TB\|_\infty = \max_{1 \leq i \leq n} |t_{ii}| \sum_{j=1}^n |b_{ij}| = \max_{1 \leq i \leq n} |t_{ii}| = \|T\|_\infty$$

und damit ergibt sich

$$\begin{aligned} \text{cond}_\infty(B) &= \|B^{-1}\|_\infty = \|(TB)^{-1}T\|_\infty \leq \|(TB)^{-1}\|_\infty \|T\|_\infty \\ &= \|(TB)^{-1}\|_\infty \|TB\|_\infty = \text{cond}_\infty(TB). \end{aligned}$$

Da die Abschätzung auch für $T = C^{-1}$ gilt und da die Matrix-Vektor-Multiplikation $z \mapsto Cz$ mit n Operationen realisierbar ist, ist C eine Vorkonditionierungsmatrix für A . \square

Im Allgemeinen ist die vorkonditionierte Systemmatrix CA weder symmetrisch noch positiv definit, selbst wenn A und C diese Eigenschaften besitzen, und daher ist die Konvergenz des CG-Verfahrens für das vorkonditionierte System nicht unmittelbar garantiert. Dies lässt sich jedoch durch Verwendung der Cholesky-Zerlegung $C = VV^\top$ beheben, denn es gilt

$$Ax = b \iff V^\top AVy = V^\top b, \quad y = V^{-1}x$$

und die Matrix $V^\top AV$ ist symmetrisch und positiv definit. Das vorkonditionierte CG-Verfahren löst diese Umformulierung, ohne die Cholesky-Faktorisierung explizit zu bestimmen.

Algorithmus 17.1 (Vorkonditioniertes CG-Verfahren) Seien $A, C \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ und $\varepsilon_{\text{stop}} > 0$. Definiere $r_0 = b - Ax_0$, $k = 0$ und setze $d_0 = z_0 = Cr_0$.

(1) Setze $x_{k+1} = x_k + \alpha_k d_k$ und $r_{k+1} = r_k - \alpha_k A d_k$ sowie $z_{k+1} = C r_{k+1}$ und definiere $d_{k+1} = z_{k+1} - \beta_k d_k$ mit

$$\alpha_k = \frac{r_k \cdot z_k}{d_k \cdot A d_k}, \quad \beta_k = -\frac{r_{k+1} \cdot z_{k+1}}{r_k \cdot z_k}.$$

(2) Stoppe falls $\|r_{k+1}\| \leq \varepsilon_{\text{stop}}$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole (1).

Bemerkungen 17.2 (i) Aufgrund der Umformulierung des Gleichungssystems ist es sinnvoll, die Eigenschaft $\text{cond}(V^\top A V) \leq \text{cond}(A)$ an eine Vorkonditionierungsmaatrix $C = VV^\top$ zu stellen.

(ii) Die Konstruktion geeigneter Vorkonditionierungsmaatzen basiert in der Regel auf besonderen Eigenschaften der zugrundeliegenden Anwendung.

17.3 Weitere Vorkonditionierungsmaatzen

Stationäre Iterationsverfahren der Form

$$x_{k+1} = x_k - R(Ax_k - b) = (I_n - RA)x_k + Rb$$

lassen sich als Fixpunktiterationen des Gleichungssystems

$$RAx = Rb$$

interpretieren. Sie sind konvergent, sofern $\rho(I_n - RA) < 1$ gilt, und motivieren die Wahl $C = R$ als Vorkonditionierungsmaatrix, denn dann gilt in grober Näherung $CA \approx I_n$, sodass wir $\text{cond}(CA) \approx 1$ erwarten dürfen. Dass so tatsächlich eine Vorkonditionierungsmaatrix definiert wird, muss im jeweils vorliegenden Fall überprüft werden.

Beispiele 17.3 (i) Mit der Zerlegung $A = L + D + R$ in den Diagonalanteil D sowie den strikten unteren beziehungsweise oberen Anteil L und R ist das Jacobi-Verfahren definiert durch

$$Dx_{k+1} = -(L + R)x_k + b = (D - A)x_k + b$$

beziehungsweise

$$x_{k+1} = x_k - D^{-1}(Ax_k - b),$$

was die Vorkonditionierungsmaatrix $C_J = D^{-1}$ motiviert.

(ii) Das Gauß-Seidel-Verfahren führt auf die Matrix $C_{GS} = D + L$, die im Allgemeinen nicht symmetrisch ist. Die symmetrische Gauß-Seidel-Vorkonditionierungsmatrix einer symmetrischen Matrix $A = L + D + L^\top \in \mathbb{R}^{n \times n}$ ist definiert durch

$$C_{SGS} = [(D + L)D^{-1}(D + L)^\top]^{-1}.$$

Die direkte Lösung eines dünnbesetzten Gleichungssystems mit Hilfe einer LU - oder Cholesky-Zerlegung kann ineffizient sein, da die Faktoren der Zerlegung im Allgemeinen nicht dünnbesetzt sind. Dieser Effekt wird als *Fill-In* bezeichnet. Die unvollständige Berechnung einer LU - oder Cholesky-Zerlegung kann jedoch auf eine geeignete Vorkonditionierungsmatrix führen. Dabei wird eine Besetzungstruktur $\mathcal{B} \subset \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ für die Faktoren vorgegeben und gefordert, dass

$$(LU)_{ij} = a_{ij}, \quad (i, j) \in \mathcal{B}, \quad \ell_{ij} = u_{ij} = 0, \quad (i, j) \notin \mathcal{B}.$$

Für gewisse Klassen von Matrizen lässt sich die Existenz der unvollständigen LU -beziehungsweise Cholesky-Zerlegung beweisen. Die Berechnung erfolgt, indem in den Algorithmen für die vollständigen Faktorisierungen die Einträge im Nullmuster ignoriert werden.

Algorithmus 17.2 (Unvollständige Cholesky-Zerlegung) Seien $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und $\mathcal{B} \subset \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ symmetrisch. Die nichttrivialen Einträge von L werden berechnet durch:

for $k = 1 : n$

$$\ell_{kk} = \left(a_{kk} - \sum_{j=1, \dots, k-1, (j,k) \in \mathcal{B}} \ell_{kj}^2 \right)^{1/2}$$

for $i = k + 1 : n$

$$\text{if } (i, k) \in \mathcal{B}; \quad \ell_{ik} = \left(a_{ik} - \sum_{\substack{j=1, \dots, k-1, \\ (j,k) \in \mathcal{B}, (i,j) \in \mathcal{B}}} \ell_{ij} \ell_{kj} \right) / \ell_{kk}; \quad \text{end}$$

end

end

Mit einer unvollständigen Faktorisierung lässt sich eine Vorkonditionierungsmatrix definieren.

Beispiel 17.4 Existiert die unvollständige Cholesky-Zerlegung $A = LL^\top + E$, so wird durch $C = (LL^\top)^{-1}$ eine mögliche Vorkonditionierungsmatrix definiert. Typische Definitionen für die Besetzungsstruktur sind die der gegebenen Matrix A , das heißt $\mathcal{B} = \{(i, j) : a_{ij} \neq 0\}$, was als *Zero-Fill-In* bezeichnet wird, oder es wird eine Bandweite $w \in \mathbb{N}_0$ vorgegeben und $\mathcal{B} = \{(i, j) : |i - j| \leq w\}$ definiert.

17.4 Lernziele, Quiz und Anwendung

Sie sollten den Begriff der dünnbesetzten Matrix erklären und an Beispielen veranschaulichen können. Ferner sollten Sie die grundlegenden Ideen der Verwendung einer Vorkonditionierungsmatrix beim CG-Verfahren kennen und einige Beispiele benennen können.

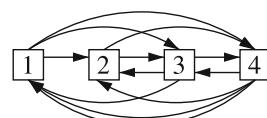
Quiz 17.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Gilt für die Anzahl der von Null verschiedenen Einträge $N_z = \{(i, j) : 1 \leq i, j \leq n, a_{ij} \neq 0\} = \mathcal{O}(n^2)$, so ist A dünnbesetzt	
Eine dünnbesetzte Matrix $A \in \mathbb{R}^{n \times n}$ wird im CCS-Format durch $\mathcal{O}(n)$ viele Informationen spezifiziert	
Das Produkt zweier dünnbesetzter Matrizen ist eine dünnbesetzte Matrix	
Jede zeilenäquilierte Matrix $A \in \mathbb{R}^{n \times n}$, das heißt es gilt $\sum_{j=1}^n a_{ij} = 1$, erfüllt $\text{cond}_\infty(A) = 1$	
Die Vorkonditionierung eines linearen Gleichungssystems führt auf ein Gleichungssystem, das mit dem Aufwand $\mathcal{O}(n)$ lösbar ist	

Anwendung 17.1 Zur Illustration von Googles PageRank-Algorithmus wird ein Modell-Internet mit N Seiten betrachtet. Es sei n_i die Anzahl der Links, die von der i -ten Seite auf andere Seiten führen. Die Variable $x_i \geq 0$ soll die Relevanz der i -ten Seite angeben und sich für jeden Link, der von der j -ten auf die i -te Seite führt, um den Wert x_j/n_j erhöhen. In der in Abb. 17.2 gezeigten Skizze gilt beispielsweise

$$x_1 = \frac{0}{2}x_2 + \frac{1}{3}x_3 + \frac{2}{4}x_4.$$

Abb. 17.2 Links in einem Modell-Internet



Insgesamt ergibt sich ein lineares Gleichungssystem zur Bestimmung des Vektors $x = [x_1, x_2, \dots, x_N]^\top$.

- (i) Zeigen Sie, dass sich die Bestimmung einer Lösung des Gleichungssystems als Eigenwertproblem $\lambda x = Ax$ mit $\lambda = 1$ formulieren lässt.
- (ii) Bestimmen Sie die Gerschgorin-Kreise für A^\top , um zu zeigen, dass $|\lambda| \leq 1$ für alle Eigenwerte von A gilt, und beweisen Sie, dass $\lambda = 1$ Eigenwert von A^\top beziehungsweise A ist.
- (iii) Bestimmen Sie mit Hilfe von MATLAB einen Eigenvektor x der Matrix A zum Eigenwert 1 mit $x_i \geq 0$, $i = 1, 2, \dots, N$, und $\|x\|_1 = 1$ für das in Abb. 17.2 gezeigte Modell-Internet.
- (iv) Führen Sie 5 Schritte der Potenzmethode mit dem Startvektor $x_0 = [1, 1, 1, 1]^\top / 4$ aus und normieren Sie dabei bezüglich der Norm $\|\cdot\|_1$.
- (v) Diskutieren Sie, ob die Matrix A in der Realität als dünnbesetzt angenommen werden kann und ob sich durch Verwendung geeigneter Speicherformate und Algorithmen zur Matrix-Vektor-Multiplikation der Aufwand verringert.

18.1 Gitter und Triangulierungen

Zur Approximation von Funktionen und Integralen in mehreren Dimensionen existieren verschiedene Ansätze, die von den Eigenschaften des zugrundeliegenden Gebiets abhängen. Als Gebiet wird dabei eine offene und zusammenhängende Menge $\Omega \subset \mathbb{R}^d$ mit $d \in \mathbb{N}$ bezeichnet, die zudem im Folgenden stets als beschränkt angenommen wird. Die einfachste Situation liegt vor, wenn Ω das Produkt von Intervallen ist, das heißt wenn Ω ein rechtwinkliges, achsenparalleles Parallelepiped der Form

$$\Omega = (a_1, b_1) \times (a_2, b_2) \times \cdots \times (a_d, b_d) = \prod_{i=1}^d (a_i, b_i)$$

ist. In diesem Fall lassen sich eindimensionale Argumente durch Tensorprodukt-Ansätze auf den mehrdimensionalen Fall übertragen.

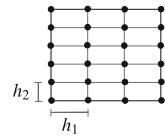
Definition 18.1 Ein (*Tensorprodukt-*)*Gitter* des Gebiets $\Omega = \prod_{i=1}^d (a_i, b_i)$ ist eine Punktmenge

$$\begin{aligned} \mathcal{G}_h &= \{x = (a_1, a_2, \dots, a_d) + (j_1 h_1, j_2 h_2, \dots, j_d h_d) : \\ &\quad 0 \leq j_i \leq n_i, i = 1, 2, \dots, d\} \end{aligned}$$

mit *Gitterfeinheiten* $h_i = (b_i - a_i)/n_i$, $n_i \in \mathbb{N}$, $i = 1, 2, \dots, d$, s. Abb. 18.1. Das Gitter heißt *uniform*, falls $h_1 = h_2 = \dots = h_d = h$ gilt.

Im Fall eines allgemeineren beschränkten Gebietes $\Omega \subset \mathbb{R}^d$ nehmen wir an, dass es einen polygonalen Rand besitzt, das heißt es existieren affin-lineare Teilaräume $H_k = \{x \in$

Abb. 18.1 Tensorproduktgitter eines Rechtecks mit Gitterfeinheiten h_1 und h_2



$\mathbb{R}^d : d_k \cdot x = c_k \}$ mit $d_k \in \mathbb{R}^d$ und $c_k \in \mathbb{R}$, sodass

$$\partial\Omega = \bigcup_{k=1}^K (\partial\Omega \cap H_k).$$

Gebiete dieser Art lassen sich in einfache Teilgebiete zerlegen. Als *Simplex* im \mathbb{R}^d bezeichnen wir eine abgeschlossene Teilmenge $T \subset \mathbb{R}^d$, die als konvexe Hülle von $d + 1$ Punkten $z_0, z_1, \dots, z_d \in \mathbb{R}^d$ gegeben ist, das heißt

$$T = \text{conv}\{z_0, z_1, \dots, z_d\} = \left\{ x \in \mathbb{R}^d : x = \sum_{i=0}^d \theta_i z_i, \theta_i \geq 0, \sum_{i=0}^d \theta_i = 1 \right\},$$

sodass T nichtentartet ist, das heißt ein nichtleeres Inneres beziehungsweise ein positives d -dimensionales Volumen besitzt. Für $d = 1, 2, 3$ sind Simplizes Intervalle, Dreiecke beziehungsweise Tetraeder, s. Abb. 18.2.

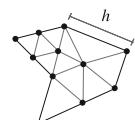
Definition 18.2 Eine (*reguläre*) *Triangulierung* des polygonal berandeten und beschränkten Gebiets Ω ist eine Menge $\mathcal{T}_h = \{T_1, T_2, \dots, T_J\}$ von Simplizes $T_j \subset \mathbb{R}^d$, $j = 1, 2, \dots, J$, sodass

$$\overline{\Omega} = \bigcup_{j=1}^J T_j$$

und der Schnitt $T_j \cap T_k$ zweier verschiedener Simplizes ist entweder leer oder ein gemeinsames Subsimplex, das heißt eine gemeinsame Ecke, Kante oder Seitenfläche. Die Simplizes einer Triangulierung werden auch als *Elemente* bezeichnet und die Menge \mathcal{N}_h der Ecken von Elementen als *Knoten*. Die Triangulierung heißt *uniform*, falls alle Elemente kongruent sind. Sie hat die (*maximale*) *Netzweite* $h > 0$, falls $\text{diam}(T) \leq h$ für alle $T \in \mathcal{T}_h$ gilt.

Auf Parallelepipedien und Simplizes werden verschiedene Polynomräume verwendet.

Abb. 18.2 Triangulierung eines zweidimensionalen Gebiets in Dreiecke



Definition 18.3 Es sei $A \subset \mathbb{R}^d$ eine abgeschlossene Menge und $k \in \mathbb{N}_0$. Die Menge der Polynome vom partiellen Grad k und vom totalen Grad k auf A sind definiert durch

$$\mathcal{Q}_k(A) = \left\{ q(x) = \sum_{0 \leq i_1, i_2, \dots, i_d \leq k} a_{i_1 i_2 \dots i_d} x_1^{i_1} x_2^{i_2} \dots x_d^{i_d} : a_{i_1 i_2 \dots i_d} \in \mathbb{R} \right\},$$

$$\mathcal{P}_k(A) = \left\{ p(x) = \sum_{\substack{0 \leq i_1, i_2, \dots, i_d \leq k \\ i_1 + i_2 + \dots + i_d \leq k}} a_{i_1 i_2 \dots i_d} x_1^{i_1} x_2^{i_2} \dots x_d^{i_d} : a_{i_1 i_2 \dots i_d} \in \mathbb{R} \right\}.$$

Bemerkung 18.1 Polynome vom partiellen Grad k sind Linearkombinationen von Tensorprodukten eindimensionaler Polynome vom Grad k .

Beispiel 18.1 Das Polynom $q(x_1, x_2) = x_1^2 x_2^3$ ist vom totalen Grad 5 und partiellen Grad 3.

18.2 Approximation auf Tensorproduktgittern

Mittels geeigneter linearer Transformationen lässt sich jedes rechtwinklige Parallelepiped auf die Menge $\Omega = (0, 1)^d$ abbilden und im Folgenden wird stets dieser Fall zusammen mit einem uniformen Tensorproduktgitter der Gitterweite $h > 0$ betrachtet.

Definition 18.4 Für eine gegebene Funktion $f \in C^0([0, 1]^d)$ und eine gegebene Gitterweite $h = 1/n$ besteht die *Tensorprodukt-Interpolationsaufgabe* in der Bestimmung eines Polynoms $q \in \mathcal{Q}_{n^d}([0, 1]^d)$ mit

$$q(x) = f(x)$$

für alle $x \in \mathcal{G}_h = \{h(i_1, i_2, \dots, i_d) : 0 \leq i_1, i_2, \dots, i_d \leq n\}$.

Satz 18.1 Die Tensorprodukt-Interpolationsaufgabe ist eindeutig lösbar.

Beweis Zur Illustration der Beweisidee betrachten wir den Fall $d = 2$. Es bezeichne $E : \mathcal{Q}_{n^2}([0, 1]^2) \rightarrow \mathbb{R}^{(n+1)^2}$ die lineare Abbildung $q \mapsto (q(x) : x \in \mathcal{G}_h)$. Sei $q \in \mathcal{Q}_{n^2}([0, 1]^2)$ mit der Eigenschaft $Eq = 0$. Für $(s, t) \in [0, 1]^2$ besitzt der Ausdruck $q(s, t)$ die Darstellungen

$$q(s, t) = \sum_{0 \leq \ell, m \leq n} a_{\ell m} s^\ell t^m = \sum_{0 \leq \ell \leq n} \left(\sum_{0 \leq m \leq n} a_{\ell m} t^m \right) s^\ell = \sum_{0 \leq \ell \leq n} b_\ell(t) s^\ell.$$

Für jedes fixierte $t_j = jh$, $j = 0, 1, \dots, n$, besitzt das Polynom $s \mapsto q(s, t_j)$ die Nullstellen $s_i = ih$, $i = 0, 1, \dots, n$, und es folgt $b_\ell(t_j) = 0$ für alle $j, \ell = 0, 1, \dots, n$. Für jedes $\ell = 0, 1, \dots, n$ besitzt das Polynom $t \mapsto b_\ell(t)$ daher die Nullstellen t_j , $j = 0, 1, \dots, n$

und es folgt $b_\ell(t) = 0$ für alle $t \in [0, 1]$ und somit $a_{\ell m} = 0$ für alle $\ell, m = 0, 1, \dots, n$ beziehungsweise $q = 0$. Damit ist E injektiv und wegen $\dim \mathcal{Q}_{n^2}([0, 1]^2) = (n + 1)^2$ auch bijektiv. \square

Die numerische Integration einer Funktion $f \in C^0([0, 1]^d)$ wird mittels der auf dem Satz von Fubini beruhenden Iterationsformel

$$I^d(f) = \int_{[0,1]^d} f(x) dx = \int_0^1 \int_0^1 \dots \int_0^1 f(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d$$

auf die Approximation eindimensionaler Integrale reduziert.

Satz 18.2 Ist $Q : C^0([0, 1]) \rightarrow \mathbb{R}$ eine Quadraturformel mit nichtnegativen Gewichten und Punkten $(w_i, t_i)_{i=0, \dots, n}$ mit Exaktheitsgrad $k \geq 0$, so wird durch

$$Q^d(f) = \sum_{i_1=0}^n \sum_{i_2=0}^n \dots \sum_{i_d=0}^n w_{i_1} w_{i_2} \dots w_{i_d} f(t_{i_1}, t_{i_2}, \dots, t_{i_d})$$

eine iterierte Quadraturformel $Q^d : C^0([0, 1]^d) \rightarrow \mathbb{R}$ definiert, die exakt ist für alle $p \in \mathcal{Q}_{k^d}([0, 1]^d)$. Ferner gilt

$$|I^d(f) - Q^d(f)| \leq \sum_{i=1}^d \sup_{\hat{x}_i \in [0, 1]^{d-1}} |I(f_{\hat{x}_i}) - Q(f_{\hat{x}_i})|,$$

wobei $f_{\hat{x}_i}$ für $\hat{x}_i = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \in [0, 1]^{d-1}$ die Abbildung

$$t \mapsto f(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_d)$$

bezeichnet.

Beweis Wir betrachten den Fall $d = 2$. Dann gilt

$$\begin{aligned} I^2(f) - Q^2(f) &= \int_0^1 \int_0^1 f(x_1, x_2) dx_1 dx_2 - \sum_{i_1=0}^n \sum_{i_2=0}^n w_{i_1} w_{i_2} f(t_{i_1}, t_{i_2}) \\ &= \int_0^1 \left[\int_0^1 f(x_1, x_2) dx_1 - \sum_{i_1=0}^n w_{i_1} f(t_{i_1}, x_2) \right] dx_2 \\ &\quad + \int_0^1 \sum_{i_1=0}^n w_{i_1} f(t_{i_1}, x_2) dx_2 - \sum_{i_1=0}^n \sum_{i_2=0}^n w_{i_1} w_{i_2} f(t_{i_1}, t_{i_2}) \\ &= \int_0^1 (If(\cdot, x_2) - Qf(\cdot, x_2)) dx_2 + \sum_{i_1=0}^n w_{i_1} (If(t_{i_1}, \cdot) - Qf(t_{i_1}, \cdot)). \end{aligned}$$

Zusammen mit der Eigenschaft $\sum_{i=0}^n w_i = 1$ ergibt sich die behauptete Aussage durch Bilden des Betrags. \square

Bemerkung 18.2 Der Aufwand der iterierten Quadraturformel wächst exponentiell bezüglich d , das heißt es sind $(n+1)^d$ Funktionsauswertungen erforderlich. Die Fehlerordnung ist hingegen dimensionsunabhängig durch den eindimensionalen Exaktheitsgrad gegeben.

18.3 Zweidimensionale Fourier-Transformation

Basierend auf der Beobachtung, dass mit einer Basis $(\omega^k)_{k=0,1,\dots,n-1}$ des \mathbb{C}^n durch die Matrizen $(\omega^k \omega^{\ell,\top})_{k,\ell=0,\dots,n-1}$ eine Basis des Vektorraums $\mathbb{C}^{n \times n}$ definiert wird, lässt sich die diskrete Fourier-Transformation auf den zweidimensionalen Fall verallgemeinern.

Satz 18.3 Für jede Matrix $Y \in \mathbb{C}^{n \times n}$ existieren eindeutig bestimmte Koeffizienten $B = (b_{k\ell})_{k,\ell=0,\dots,n-1} \in \mathbb{C}^{n \times n}$, sodass

$$Y = \sum_{k,\ell=0}^{n-1} b_{k\ell} E^{k\ell}$$

mit der durch die Matrizen $E^{k\ell} = (e^{i(j_1 k + j_2 \ell) 2\pi/n})_{j_1,j_2=0,\dots,n-1} \in \mathbb{C}^{n \times n}$ für $k, \ell = 0, 1, \dots, n-1$ definierten Orthogonalbasis bezüglich des Skalarprodukts $E : F = \sum_{j,m=0}^{n-1} E_{jm} \overline{F}_{jm}$. Mit $T_n \in \mathbb{C}^{n \times n}$ definiert durch $(T_n)_{jk} = e^{ijk 2\pi/n}$, $j, k = 0, 1, \dots, n-1$, gilt

$$Y = \frac{1}{n^2} \overline{T}_n B \overline{T}_n, \quad B = T_n Y T_n.$$

Beweis Übungsaufgabe. \square

Bemerkungen 18.3 (i) Die für die Transformation erforderlichen Matrix-Multiplikationen lassen sich mit $\mathcal{O}(n^2 \log n)$ Rechenoperationen durchführen. Dazu wird die eindimensionale schnelle Fourier-Transformation zunächst auf die Spalten von Y und anschließend auf die Zeilen der resultierenden Matrix angewendet.

(ii) Die zweidimensionale Fourier-Transformation ist die Basis von Bildkompressions-techniken wie dem jpeg-Format.

18.4 Approximation auf Triangulierungen

Mittels Triangulierungen lassen sich Spline-Räume verallgemeinern.

Definition 18.5 Für $k, m \geq 0$ und eine Triangulierung \mathcal{T}_h eines beschränkten Gebiets $\Omega \subset \mathbb{R}^d$ bezeichne

$$S^{m,k}(\mathcal{T}_h) = \{v_h \in C^k(\overline{\Omega}) : v_h|_T \in \mathcal{P}_m(T) \text{ für alle } T \in \mathcal{T}_h\}$$

den *Spline-Raum vom Grad m und der Ordnung k bezüglich \mathcal{T}_h* .

Mittels einer affin-linearen Transformation lassen sich Untersuchungen der Spline-Räume auf den Fall des Standardsimplex

$$\hat{T} = \text{conv}\{\hat{z}_0, \hat{z}_1, \dots, \hat{z}_d\}$$

zurückführen, wobei $\hat{z}_0 = 0$ und $\hat{z}_i = e_i$ für $i = 1, 2, \dots, d$ mit den kanonischen Basisvektoren $(e_1, e_2, \dots, e_d) \subset \mathbb{R}^d$ seien.

Lemma 18.1 Für $i = 0, 1, \dots, d$ sei $\hat{\varphi}_i \in \mathcal{P}_1(\hat{T})$ die eindeutig durch die Bedingungen $\hat{\varphi}_i(\hat{z}_j) = \delta_{ij}$, $j = 0, 1, \dots, d$, definierte Hutfunktion. Ist $T = \text{conv}\{z_0, z_1, \dots, z_d\} \in \mathbb{R}^d$ ein nichtentartetes Simplex, so wird durch

$$\hat{x} \mapsto \Phi_T(\hat{x}) = \sum_{i=0}^d \hat{\varphi}_i(\hat{x}) z_i$$

ein affin-linearer Diffeomorphismus $\Phi_T : \hat{T} \rightarrow T$ mit der Eigenschaft $\Phi_T(\hat{z}_i) = z_i$, $i = 0, 1, \dots, d$, definiert, s. Abb. 18.3. Das Volumen von T ist gegeben durch $|\det D\Phi_T|/d!$.

Beweis Die Hutfunktionen sind durch $\hat{\varphi}_i(\hat{x}) = \hat{x}_i$, $i = 1, 2, \dots, d$, und $\hat{\varphi}_0(\hat{x}) = 1 - \hat{x}_1 - \dots - \hat{x}_d$ für $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_d) \in \hat{T}$ gegeben und die damit definierte Abbildung Φ_T erfüllt $\Phi_T(\hat{z}_i) = z_i$, $i = 0, 1, \dots, d$. Für alle $\hat{x} \in \hat{T}$ gilt

$$\Phi_T(\hat{x}) = z_0 + Q_T \hat{x} = z_0 + [z_1 - z_0, z_2 - z_0, \dots, z_d - z_0] \hat{x}.$$

Abb. 18.3 Der Diffeomorphismus Φ_T bildet den Standardsimplex \hat{T} bijektiv auf den Simplex T ab

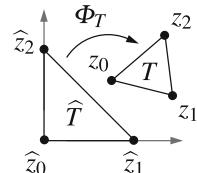
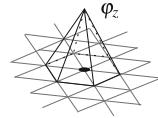


Abb. 18.4 Einem Knoten z zugeordnete Hutfunktion φ_z in einer Triangulierung



Die Determinante von Q_T ist definiert als das Volumen des Bildes des Einheitswürfels $[0, 1]^d$ unter der linearen Abbildung Q_T , womit das Volumen des Bildes des Standardsimplex gegeben ist durch $|\det Q_T|/d!$. Da dieses mit dem Volumen von T übereinstimmt und somit positiv ist, folgt dass Φ_T ein Diffeomorphismus ist. \square

Die Hutfunktionen aus dem Beweis lassen sich mit dem Diffeomorphismus Φ_T auf die Elemente transformieren und führen zum Begriff der *nodalen Basis*, mit der sich die Spline-Interpolationsaufgabe im Raum $S^{1,0}(\mathcal{T}_h)$ lösen lässt. Eine typische Hutfunktion ist in Abb. 18.4 gezeigt.

Satz 18.4 Es existiert eine eindeutig bestimmte Basis $(\varphi_z : z \in \mathcal{N}_h)$ des Raums $S^{1,0}(\mathcal{T}_h)$ mit der Eigenschaft $\varphi_z(y) = \delta_{zy}$ für alle $z, y \in \mathcal{N}_h$. Für $f \in C^0(\overline{\Omega})$ wird durch

$$\mathcal{I}_h f = \sum_{z \in \mathcal{N}_h} f(z) \varphi_z$$

der nodale Interpolant $\mathcal{I}_h f \in S^{1,0}(\mathcal{T}_h)$ mit der Eigenschaft $\mathcal{I}_h f(z) = f(z)$ für alle $z \in \mathcal{N}_h$ definiert.

Beweis Seien $z \in \mathcal{N}_h$ und $T \in \mathcal{T}_h$. Gilt $z \notin T$, so definiere $\varphi_z|_T = 0$. Andernfalls sei $i \in \{0, 1, \dots, d\}$, sodass $\Phi_T(\hat{z}_i) = z$ gilt und definiere $\varphi_z|_T = \hat{\varphi}_i \circ \Phi_T^{-1}$. Auf diese Weise werden Funktionen $(\varphi_z : z \in \mathcal{N}_h) \subset S^1(\mathcal{T}_h)$ mit der Eigenschaften $\varphi_z(y) = \delta_{zy}$ für $z, y \in \mathcal{N}_h$ definiert. Zum Nachweis, dass dies eine Basis ist, sei $s_h \in S^{1,0}(\mathcal{T}_h)$ beliebig. Durch

$$\tilde{s}_h = \sum_{z \in \mathcal{N}_h} s_h(z) \varphi_z$$

wird eine Funktion $\tilde{s}_h \in S^{1,0}(\mathcal{T}_h)$ definiert mit $\tilde{s}_h(z) = s_h(z)$ für alle $z \in \mathcal{N}_h$. Für jedes $T \in \mathcal{T}_h$ ist die Funktion $\hat{e} = (\tilde{s}_h - s_h) \circ \Phi_T$ affin-linear auf \hat{T} mit $\hat{e}(0) = 0$ und $\hat{e}(e_i) = 0$, $i = 0, 1, \dots, d$. Daraus folgt $\hat{e} = 0$ und insgesamt $s_h = \tilde{s}_h$. \square

Der Interpolationsfehler lässt sich wie im eindimensionalen Fall beschränken.

Satz 18.5 Sei $f \in C^2(\overline{\Omega})$ und \mathcal{T}_h eine reguläre Triangulierung von Ω . Dann gilt

$$\|f - \mathcal{I}_h f\|_{C^0(\overline{\Omega})} \leq \frac{h^2}{2} \|D^2 f\|_{C^0(\overline{\Omega})}.$$

Beweis Wir definieren $e = f - \mathcal{I}_h f$ und es seien $x_m \in \overline{\Omega}$ und $T \in \mathcal{T}_h$, sodass $x_m \in T$ und $|e(x_m)| = \|e\|_{C^0(\overline{\Omega})}$ gilt. Offensichtlich ist $e|_T \in C^2(T)$. Liegt x_m im Innern von T , so gilt $\nabla e(x_m) = 0$. Ist x_m eine Ecke von T , so folgt $e|_T = 0$. Liegt x_m auf einer Seite von T , so existiert ein Ecke $z \in \mathcal{N}_h \cap T$, sodass die Ableitung der Abbildung $t \mapsto e(z+t(x_m-z))$ im Punkt $t = 1$ verschwindet, das heißt es gilt $\nabla e(x_m) \cdot (x_m - z) = 0$. In jedem Fall existiert ein $z \in \mathcal{N}_h \cap T$, sodass mit einer Taylor-Approximation für ein $\xi \in T$ gilt

$$0 = e(z) = e(x_m) + \frac{1}{2}(z - x_m)^\top D^2 e(\xi)(z - x_m).$$

Da $|z - x_m| \leq h$ und $D^2 I_h f|_T = 0$ gelten, folgt die Behauptung. \square

Summierte Quadraturformeln auf triangulierten Gebieten lassen sich mit Hilfe des Referenzelements definieren.

Definition 18.6 Es sei $\hat{Q} : C^0(\hat{T}) \rightarrow \mathbb{R}$ eine durch Quadraturpunkte und -gewichte $(\hat{\xi}_i, \hat{w}_i)_{i=0,\dots,n}$ definierte Quadraturformel auf \hat{T} , das heißt $\hat{Q}\hat{f} = \sum_{i=0}^n \hat{w}_i \hat{f}(\hat{\xi}_i)$. Eine zugehörige *summierte Quadraturformel* $Q_{\mathcal{T}_h} : C^0(\overline{\Omega}) \rightarrow \mathbb{R}$ ist definiert durch

$$Q_{\mathcal{T}_h}(f) = \sum_{T \in \mathcal{T}_h} \sum_{i=0}^n |\det D\Phi_T| \hat{w}_i f(\Phi_T(\hat{\xi}_i))$$

Bemerkung 18.4 Ist die Quadraturformel $\hat{Q} : C^0(\hat{T}) \rightarrow \mathbb{R}$ exakt vom totalen Grad $m \geq 0$, das heißt werden die Integrale aller Polynome $q \in \mathcal{P}_m(\hat{T})$ exakt wiedergegeben, so ist die summierte Quadraturformel $Q_{\mathcal{T}_h}$ exakt für alle $f \in S^{m,0}(\mathcal{T}_h)$.

Beispiel 18.2 Gaußsche Quadraturformeln mit einem, drei beziehungsweise sieben Quadraturpunkten auf $\hat{T} \subset \mathbb{R}^2$ werden definiert durch $\hat{\xi} \in \mathbb{R}^{n \times 2}$ und $\hat{w} \in \mathbb{R}^n$ mit

$$\hat{\xi} = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top, \quad \hat{w} = \frac{1}{2},$$

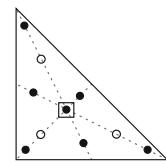
sowie

$$\hat{\xi} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix}^\top, \quad \hat{w} = \frac{1}{6}[1, 1, 1]^\top,$$

beziehungsweise mit $s = \sqrt{15}$

$$\begin{aligned} \hat{\xi} &= \frac{1}{21} \begin{bmatrix} 6-s & 9+2s & 6-s & 6+s & 6+s & 9-2s & 7 \\ 6-s & 6-s & 9+2s & 9-2s & 6+s & 6+s & 7 \end{bmatrix}^\top, \\ \hat{w} &= \frac{1}{2400} [155-s, 155-s, 155-s, 155+s, 155+s, 155+s, 270]^\top. \end{aligned}$$

Abb. 18.5 Schematische Darstellung Gaußscher Quadraturformeln auf dem Referenzdreieck



Diese Quadraturformeln sind exakt für die Polynomräume $\mathcal{P}_1(\hat{T})$, $\mathcal{Q}_2(\hat{T})$ beziehungsweise $\mathcal{P}_5(\hat{T})$; sie sind schematisch in Abb. 18.5 dargestellt.

18.5 Lernziele, Quiz und Anwendung

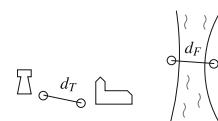
Ihnen sollten Zugänge zur Interpolation und Quadratur von Funktionen in mehreren Veränderlichen bekannt sein. Sie sollten Interpolationsabschätzungen angeben und die Probleme der Quadratur in Räumen hoher Dimension erläutern können.

Quiz 18.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Das Polynom $q(x, y) = x^2y^3z^4 + 3x^5$ besitzt den partiellen Grad 4 und den totalen Grad 5	
Es gilt $\dim \mathcal{Q}_k(\mathbb{R}^d) = (k+1)^d$	
Es gilt $\dim \mathcal{P}_k(\mathbb{R}^d) = (d+1)k$	
Es gilt $\dim S^1(\mathcal{T}_h) = \mathcal{N}_h $, wobei $ \mathcal{N}_h $ die Kardinalität von \mathcal{N}_h bezeichne	
Ist \mathcal{T}_h eine Triangulierung eines Gebiets $\Omega \subset \mathbb{R}^2$ mit Kanten \mathcal{E}_h und Knoten \mathcal{N}_h , so gilt für deren Kardinalitäten $ \mathcal{N}_h - \mathcal{E}_h + \mathcal{T}_h = 1$	

Anwendung 18.1 An einer engen Stelle soll die Breite d_F eines Flusses bestimmt werden. Dazu werden an den gegenüberliegenden Ufern Peilmarkierungen angebracht. In gewisser Entfernung zu der Stelle am Fluss befindet sich eine Stadt mit einem Kirch- und einem Wasserturm, deren Abstand mit hoher Exaktheit bekannt und mit d_T bezeichnet sei, s. Abb. 18.6. Als Hilfsmittel stehen ein Peilgerät, mit dem Winkel zwischen zwei Anpeilungspunkten gemessen werden können, und die Möglichkeit der Installation weiterer Peilmarkierungen zur Verfügung. Verwenden Sie eine Triangulierung, um die Größe

Abb. 18.6 Bestimmung einer unbekannten aus einer bekannten Distanz



d_F zu bestimmen. Mit welchen Fehlereinflüssen muss gerechnet und wie können diese minimiert werden? Wie ist es zu interpretieren, wenn die Summe der Winkel an einem inneren Knoten von 2π abweicht, dies aber nicht auf Messfehler zurückgeführt werden kann? Wie sind geografische Besonderheiten zu berücksichtigen?

Teil III

Numerik gewöhnlicher Differenzialgleichungen

19.1 Grundlagen

Viele zeitlich veränderliche Prozesse lassen sich durch sogenannte *gewöhnliche Differentialgleichungen* beschreiben. Dabei ist eine differenzierbare Funktion $y : [0, T) \rightarrow \mathbb{R}$ gesucht, die für eine gegebene Abbildung $f : (0, T) \times \mathbb{R} \rightarrow \mathbb{R}$ die Gleichung

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

für alle $t \in (0, T)$ sowie die *Anfangsbedingung* $y(0) = y_0$ für eine gegebene Zahl $y_0 \in \mathbb{R}$ erfüllt. Man bezeichnet t als *unabhängige* und y als *abhängige Variable* des *Anfangswertproblems*. Die Differentialgleichung wird häufig in der Form $y' = f(t, y)$ geschrieben, das heißt das Argument t wird bei der Funktion y und ihren Ableitungen weggelassen. Eine Differentialgleichung heißt *linear*, wenn die Abbildung $s \mapsto f(t, s)$ für alle $t \in (0, T)$ linear ist.

Beispiel 19.1 Für $k \in \mathbb{R}$ betrachten wir die Differentialgleichung $y'(t) = ky(t)$, das heißt $f(t, s) = ks$ ist unabhängig von t . Für jedes $c \in \mathbb{R}$ ist

$$y(t) = ce^{kt}$$

eine Lösung der Differentialgleichung auf jedem Intervall $(0, T)$. Durch eine Anfangsbedingung $y(0) = y_0$ wird $c = y_0$ festgelegt.

Bemerkungen 19.1 (i) Das Anfangswertproblem $y' = ky$, $y(0) = y_0$ beschreibt die Entwicklung eines Kontos mit Anfangskapital y_0 bei fester Verzinsung k pro Zeiteinheit und sofortiger Berücksichtigung des Zinseszins.

(ii) Nach dem *Newtonischen Abkühlungsgesetz* ist die Änderung der Temperatur θ eines Körpers proportional zur Differenz zur umgebenden Temperatur θ_u , das heißt $\theta'(t) = -k(\theta(t) - \theta_u)$.

(iii) Die Identität $y'(t) = ky(t)$ bedeutet, dass die Änderung von y zum Zeitpunkt t proportional ist zum Wert von y zu diesem Zeitpunkt.

(iii) Die Differenzialgleichung $y' = ky$ beschreibt auch die Entwicklung einer Population, wobei $k > 0$ gilt, wenn die Geburtenrate höher als die Sterblichkeitsrate ist.

In vielen Anwendungen werden mehrere relevante Größen gleichzeitig betrachtet, deren Werte sich gegenseitig beeinflussen. Dies führt auf *Systeme von Differenzialgleichungen*, bei denen Funktionen $y_1, y_2, \dots, y_n : [0, T] \rightarrow \mathbb{R}$ gesucht sind mit der Eigenschaft, dass

$$\begin{aligned} y'_1(t) &= f_1(t, y_1(t), y_2(t), \dots, y_n(t)), \\ &\vdots \\ y'_n(t) &= f_n(t, y_1(t), y_2(t), \dots, y_n(t)) \end{aligned}$$

für alle $t \in (0, T)$ gilt. Solche Systeme lassen sich in vektorieller Notation schreiben als $y'(t) = f(t, y(t))$, wobei $y = [y_1, y_2, \dots, y_n]^\top$ und

$$f(t, s) = \begin{bmatrix} f_1(t, s_1, s_2, \dots, s_n) \\ \vdots \\ f_n(t, s_1, s_2, \dots, s_n) \end{bmatrix}$$

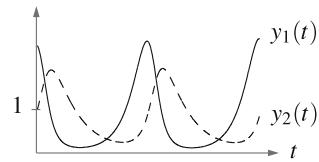
für $s = [s_1, s_2, \dots, s_n]^\top \in \mathbb{R}^n$ seien. Eine Anfangsbedingung wird dann durch einen Vektor $y_0 \in \mathbb{R}^n$ definiert.

19.2 Das Räuber-Beute-Modell

Das Räuber-Beute-Modell nach Lotka–Volterra beschreibt die Entwicklung der Anzahl von Raub- und Beutetieren wie beispielsweise Greifvögeln und Mäusen, wobei angenommen wird, dass sich die Raubtiere ausschließlich von den Beutetieren ernähren. Es seien $y_1(t)$ und $y_2(t)$ die Anzahl der Beute- beziehungsweise Raubtiere zum Zeitpunkt t in geeigneten Einheiten, sodass für $y_1 = y_2 = 1$ ein Gleichgewichtszustand eintritt, das heißt in diesem Fall entspricht die Vermehrung genau der Abnahme durch Sterben und Gefressenenwerden. Die Veränderung der Anzahl der Beutetiere y_1 ist dann proportional zu ihrer Anzahl, wobei der Proportionalitätsfaktor von der Anzahl der Raubtiere abhängt und positiv ist, wenn $y_2 < 1$ gilt, und negativ, wenn $y_2 > 1$ gilt, das heißt beispielsweise

$$y'_1(t) = \alpha(1 - y_2(t))y_1(t).$$

Abb. 19.1 Typische periodische Lösung im Räuber-Beute-Modell



Ähnlich ist die Veränderung der Anzahl der Raubtiere y_2 proportional zu ihrer Anzahl, wobei der Proportionalitätsfaktor positiv ist, falls mehr Beutetiere als im Gleichgewichtszustand zur Verfügung stehen, das heißt beispielsweise

$$y'_2(t) = \beta(y_1(t) - 1)y_2(t).$$

Ein typischer Verlauf der Populationen für den Fall $y_1(0) > 1$ und $y_2(0) = 1$ ist in Abb. 19.1 dargestellt und zeigt, dass eine große Anzahl von Beutetieren zu einer Zunahme der Raubtiere führt, bis ein kritischer Wert erreicht ist, und eine niedrige Anzahl an Raubtieren zu einer Vermehrung der Beutetiere führt.

19.3 Gleichungen höherer Ordnung

Die bisher betrachteten Gleichungen beinhalteten nur Ableitungen erster Ordnung. Allgemeiner kann man gewöhnliche Differentialgleichungen *m-ter Ordnung* betrachten, die sich abstrakt als

$$y^{(m)}(t) = f(t, y(t), y'(t), y''(t), \dots, y^{(m-1)}(t))$$

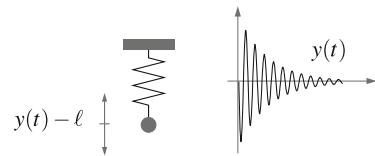
mit einer Funktion $f : (0, T) \times \mathbb{R}^m \rightarrow \mathbb{R}$ schreiben lassen. Differentialgleichungen höherer Ordnung lassen sich jedoch durch das Einführen von Hilfsvariablen als ein System von Differentialgleichungen erster Ordnung schreiben. Dazu wird $z = [z_1, z_2, \dots, z_m]^\top$ definiert durch

$$z_1 = y, \quad z_2 = y', \quad z_3 = y'', \quad \dots, \quad z_m = y^{(m-1)}$$

und das System

$$\begin{aligned} z'_1(t) &= z_2(t), \\ &\vdots \\ z'_{m-1}(t) &= z_m(t), \\ z'_m(t) &= f(t, z_1(t), z_2(t), \dots, z_m(t)) \end{aligned}$$

Abb. 19.2 Schwingungsverhalten eines gedämpften Federpendels



betrachtet, was sich in naheliegender Weise als vektorielle Differenzialgleichung $z' = \tilde{f}(t, z)$ schreiben lässt. Für Differenzialgleichungen höherer Ordnung ist es im Allgemeinen nicht ausreichend, nur den Funktionswert bei $t = 0$ vorzuschreiben. Es müssen zusätzlich die Ableitungen bis zur Ordnung $m - 1$ als Anfangsdaten vorgegeben werden, das heißt

$$y(0) = y_{0,0}, \quad y'(0) = y_{0,1}, \quad \dots, \quad y^{(m-1)}(0) = y_{0,m-1}$$

beziehungsweise mit dem oben definierten Vektor z die Bedingung $z(0) = z_0$ mit $z_0 = [y_{0,0}, y_{0,1}, \dots, y_{0,m-1}]^\top \in \mathbb{R}^m$.

Beispiel 19.2 Die Differenzialgleichung $y'' = -c^2 y$ besitzt die Lösungen $y(t) = \alpha \sin(ct)$ mit der Eigenschaft $y(0) = 0$ für jede Wahl von $\alpha \in \mathbb{R}$. Durch Vorschreiben von $y'(0)$ wird α eindeutig festgelegt.

Bemerkung 19.2 Die Auslenkung eines Federpendels, das am oberen Ende fixiert, am unteren Ende der Feder mit der Punktmasse m belastet sei und in der Ruhelage die Länge ℓ habe, genügt dem Kräftegleichgewicht

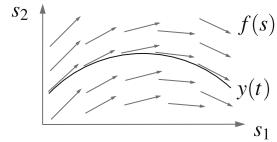
$$m y''(t) + r y'(t) + D(y(t) - \ell) = 0$$

aus Trägheitskraft, Reibungskraft und Rückstellkraft. Die Ruhelage ist mit der Gewichtskraft gegeben durch $\ell = mg/D$. Um das Schwingungsverhalten für $t > 0$ vorherzusagen, muss neben der Anfangsauslenkung $y(0)$ auch die Anfangsgeschwindigkeit $y'(0)$ bekannt sein. Ein typischer Lösungsverlauf ist in Abb. 19.2 gezeigt.

19.4 Autonome Gleichungen

Differenzialgleichungen $y'(t) = f(t, y(t))$, bei denen die Funktion f nicht von t abhängt, also $f(t, s) = \tilde{f}(s)$ gilt, heißen *autonome* Differenzialgleichungen. Durch Hinzufügen der Gleichung $z'(t) = 1$ zeigt man, dass sich jede Differenzialgleichung als System autonomer Differenzialgleichungen schreiben lässt.

Abb. 19.3 Lösungen autonomer Differenzialgleichungen sind Integralkurven des Vektorfeldes f



Bemerkung 19.3 Eine Lösung eines Systems autonomer Differenzialgleichungen $y' = f(y)$ mit $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ wird auch als *Integralkurve* des Vektorfeldes f bezeichnet, denn y lässt sich geometrisch als Kurve im \mathbb{R}^n interpretieren, deren Tangente in jedem Punkt gerade durch f vorgeschrieben ist, s. Abb. 19.3. Dies wird auch als *Phasendiagramm* bezeichnet. Aus ihm lassen sich qualitative Eigenschaften von Lösungen wie Periodizität oder Dämpfung ablesen.

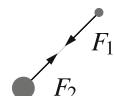
19.5 Zweikörperprobleme

Zwischen Körpern wirken anziehende Gravitationskräfte, die proportional zum Produkt der Massen und invers proportional zum Quadrat des Abstands sind. Mit dem zweiten Newtonschen Gesetz, welches besagt, dass die Impulsänderung eines Körpers beziehungsweise das Produkt der Masse und der Beschleunigung der Summe der wirkenden Kräfte entspricht, lassen sich damit Bewegungsgleichungen formulieren. Beschreiben die Funktionen $y_1, y_2 : [0, T) \rightarrow \mathbb{R}^3$ die Positionen der Mittelpunkte zweier Körper der Massen m_1, m_2 , so folgt,

$$\begin{aligned} m_1 y_1'' &= F_1(y_1, y_2) = \gamma \frac{m_1 m_2}{\|y_1 - y_2\|^2} \frac{y_2 - y_1}{\|y_1 - y_2\|}, \\ m_2 y_2'' &= F_2(y_1, y_2) = \gamma \frac{m_1 m_2}{\|y_1 - y_2\|^2} \frac{y_1 - y_2}{\|y_1 - y_2\|}, \end{aligned}$$

wobei $\gamma \approx 6.673 \cdot 10^{-11} \text{ m}^3 / (\text{kg s})$ die Gravitationskonstante sei. Man beachte dabei die entgegengesetzten Richtungen der wirkenden Kräfte. Mit den Anfangspositionen $y_1(0)$ und $y_2(0)$ sowie den Anfangsgeschwindigkeitsvektoren $y_1'(0)$ und $y_2'(0)$ lassen sich dann die Positionen der Körper vorhersagen, so lange sie einen positiven Abstand besitzen, s. Abb. 19.4.

Abb. 19.4 Zwischen Körpern wirken anziehende Gravitationskräfte



19.6 Explizite Lösungen

In speziellen Situationen lassen sich gewöhnliche Differenzialgleichungen explizit lösen. Für *separierte* Gleichungen der Form $y' = f(t)g(y)$ führt die formale Äquivalenz

$$\frac{dy}{dt} = f(t)g(y) \iff \frac{dy}{g(y)} = f(t)dt \iff \int \frac{1}{g(y)} = \int f(t)$$

mit Stammfunktionen $G(y)$ von $1/g(y)$ und $F(t) + c$ von $f(t)$ auf die Identitäten

$$G(y) = F(t) + c \iff y(t) = G^{-1}(F(t) + c).$$

Dieses Vorgehen bezeichnet man als *Separation der Variablen*. Die Methode der *Variation der Konstanten* erlaubt die Lösung von Gleichungen der Form $y' = f(t)y + h(t)$. Dazu wird zunächst die homogene Gleichung $z' = f(t)z$ gelöst und anschließend eine Funktion φ gesucht, sodass $y = \varphi z$ gilt. Mit der Produktregel ergibt sich

$$f(t)\varphi z + h(t) = y' = \varphi'z + \varphi z' = \varphi'z + \varphi f(t)z,$$

also die Bedingung $\varphi' = h/z$.

Beispiele 19.3 (i) Für die Gleichung $y' = y^2$ erhält man mit $F(t) = t$ und $G(y) = -1/y$ die Lösungen $y(t) = -1/(t + c)$.

(ii) Im Fall $y' = ky + h(t)$ erfüllt $z(t) = ce^{kt}$ die Gleichung $z' = kz$ und mit $\varphi(t) = \int_0^t h(s)c^{-1}e^{-ks} ds$ erhält man eine allgemeine Lösung.

19.7 Lernziele, Quiz und Anwendung

Sie sollten gewöhnliche Differenzialgleichungen und Anfangswertprobleme erklären und an Beispielen illustrieren können. Für einige Spezialfälle sollten Sie explizite Lösungen konstruieren können.

Quiz 19.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Die notwendige Anzahl von Anfangsdaten für die Wohlgestelltheit einer gewöhnlichen Differenzialgleichung entspricht der Ordnung der Differenzialgleichung	
Ist y Lösung der autonomen Differenzialgleichung $y' = f(y)$, so ist auch $y(t + c)$ eine Lösung für jedes $c \in \mathbb{R}$	
Die Identität $y' = y(y(t))$ definiert eine gewöhnliche Differenzialgleichung	
Die Differenzialgleichung $my' = ky$ beschreibt die Impulserhaltung eines Körpers der Masse m	
Gilt $f(s) = 0$ für ein $s \in \mathbb{R}^n$, so ist die konstante Abbildung $y(t) = s$ eine Lösung der autonomen Differenzialgleichung $y' = f(y)$	

Anwendung 19.1 Das Wachstum einer Population wird nur in einem gewissen Bereich sinnvoll durch die Differenzialgleichung $y' = ky$ beschrieben. Wenn eine Kapazitätsgrenze y_{\max} erreicht ist, wird keine weitere Zunahme der Population stattfinden. Erklären Sie, wieso dieser Effekt durch die Gleichung $y' = k(1 - y/y_{\max})y$ beschrieben werden kann und skizzieren Sie Lösungen dieser Differenzialgleichung.

20.1 Existenz und Eindeutigkeit

Ein zentrales Existenzresultat basiert auf dem Banachschen Fixpunktsatz. Dazu sei X ein Banachraum, das heißt X ist ein Vektorraum auf dem eine Norm $\|\cdot\| : X \rightarrow \mathbb{R}$ definiert ist, bezüglich der jede Cauchy-Folge in X konvergiert.

Satz 20.1 *Ist $\Psi : X \rightarrow X$ eine Kontraktion auf dem Banachraum X , das heißt existiert eine Konstante $K < 1$, sodass*

$$\|\Psi(u) - \Psi(v)\| \leq K\|u - v\|$$

für alle $u, v \in X$, so besitzt Ψ einen eindeutigen Fixpunkt $y \in X$, das heißt es gilt $\Psi(y) = y$.

Die daraus folgende Existenzaussage verwendet eine äquivalente Darstellung einer gewöhnlichen Differentialgleichung als Integralgleichung.

Lemma 20.1 *Sei $f \in C^0([0, T] \times \mathbb{R})$. Die Funktion $y \in C^1([0, T])$ erfüllt*

$$y'(t) = f(t, y(t)), \quad t \in (0, T), \quad y(0) = y_0$$

genau dann, wenn $y \in C^0([0, T])$ und

$$y(t) = y_0 + \int_0^t f(s, y(s)) \, ds$$

für alle $t \in [0, T]$ gilt.

Beweis

- (i) Sei zunächst $y \in C^1([0, T])$ eine Lösung der Differenzialgleichung, die wir mit s statt t schreiben. Der Hauptsatz der Differenzial- und Integralrechnung liefert

$$y(t) - y(0) = \int_0^t y'(s) \, ds = \int_0^t f(s, y(s)) \, ds.$$

Mit der Anfangsbedingung $y(0) = y_0$ folgt die Integralgleichung.

- (ii) Umgekehrt erfülle $y \in C^0([0, T])$ die Integralgleichung. Für $t = 0$ ergibt diese, dass $y(0) = y_0$ gilt. Für $t \in [0, T]$ und $h > 0$ folgt aus der Integralgleichung und dem Mittelwertsatz

$$\frac{y(t+h) - y(t)}{h} = \frac{1}{h} \int_t^{t+h} f(s, y(s)) \, ds = f(\eta, y(\eta))$$

für ein $\eta \in [t, t+h]$. Für $h \rightarrow 0$ folgt $\eta \rightarrow t$ und somit

$$\lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h} = f(t, y(t))$$

für alle $t \in [0, T]$. Durch Betrachtung des linksseitigen Quotienten $(y(t) - y(t-h))/h$ folgt, dass $y \in C^1([0, T])$ gilt und y die Differenzialgleichung löst. \square

Die Integraldarstellung zeigt, dass y Lösung der Fixpunktgleichung $y = \Psi[y]$ ist, wenn $\Psi : C^0([0, T]) \rightarrow C^0([0, T])$ durch

$$\Psi[y](t) = y_0 + \int_0^t f(s, y(s)) \, ds$$

definiert ist. Im folgenden *Satz von Picard–Lindelöf* wird eine Norm auf dem Raum $C^0([0, T])$ konstruiert, bezüglich der Ψ eine Kontraktion ist. Wir betrachten der Übersichtlichkeit halber nur skalare Gleichungen.

Satz 20.2 Die Abbildung $f \in C^0([0, T] \times \mathbb{R})$ sei uniform Lipschitz-stetig im zweiten Argument, das heißt es existiere ein $L \geq 0$, sodass

$$|f(t, v) - f(t, w)| \leq L|v - w|$$

für alle $t \in [0, T]$ und alle $v, w \in \mathbb{R}$ gilt. Dann besitzt das Anfangswertproblem

$$y'(t) = f(t, y(t)), \quad t \in (0, T), \quad y(0) = y_0$$

eine eindeutige Lösung $y \in C^1([0, T])$.

Beweis Der Operator Ψ sei wie oben definiert. Für jedes $u \in C^0([0, T])$ implizieren die Bedingungen an f , dass $\Psi[u] \in C^0([0, T])$ gilt. Auf $C^0([0, T])$ betrachten wir die gewichtete Norm

$$\|u\|_L = \sup_{t \in [0, T]} e^{-2Lt} |u(t)|.$$

Mit dieser Norm ist $C^0([0, T])$ vollständig und es genügt zu zeigen, dass Ψ bezüglich $\|\cdot\|_L$ eine Kontraktion ist. Für $u, v \in C^0([0, T])$ und $t \in [0, T]$ gilt

$$\begin{aligned} e^{-2Lt} |\Psi[u](t) - \Psi[v](t)| &= e^{-2Lt} \left| \int_0^t f(s, u(s)) - f(s, v(s)) \, ds \right| \\ &\leq L e^{-2Lt} \int_0^t |u(s) - v(s)| \, ds \\ &= L e^{-2Lt} \int_0^t e^{2Ls} e^{-2Ls} |u(s) - v(s)| \, ds \\ &\leq L e^{-2Lt} \|u - v\|_L \int_0^t e^{2Ls} \, ds \\ &= L e^{-2Lt} \frac{1}{2L} (e^{2Lt} - 1) \|u - v\|_L \\ &\leq \frac{1}{2} \|u - v\|_L. \end{aligned}$$

Durch Bildung des Supremums auf der linken Seite erhalten wir

$$\|\Psi[u] - \Psi[v]\|_L \leq \frac{1}{2} \|u - v\|_L,$$

das heißt $\Psi : C^0([0, T]) \rightarrow C^0([0, T])$ ist eine Kontraktion und der Banachsche Fixpunktssatz impliziert die Existenz eines eindeutigen Fixpunkts $y \in C^0([0, T])$. Nach Definition von Ψ und dem vorigen Lemma ist dies gleichbedeutend damit, dass y Lösung des Anfangswertproblems ist. \square

Der konstruktive Beweis des Banachschen Fixpunktssatzes zeigt, dass der Fixpunkt $y \in C^0([0, T])$ gegeben ist als Grenzwert der rekursiv definierten Folge

$$y^{k+1} = \Psi[y^k]$$

mit einer beliebigen Startfunktion $y^0 \in C^0([0, T])$. Diese Beobachtung kann man für die Konstruktion numerischer Verfahren zur Lösung von Anfangswertproblemen verwenden, jedoch müssen Funktionen geeignet interpoliert und integriert werden.

Bemerkung 20.1 Die Bedingung der uniformen Lipschitz-Stetigkeit an die Funktion f ist eine einschränkende Annahme. Ist f lediglich stetig, so lässt sich mit dem *Satz von Peano* die Existenz einer lokalen Lösung beweisen, das heißt es existieren $0 < T_* \leq T$ und $y \in C^1([0, T_*])$, sodass y das Anfangswertproblem auf dem Intervall $(0, T_*)$ löst.

Beispiele 20.1 (i) Das Anfangswertproblem $y' = ky$, $y(0) = y_0$, besitzt auf jedem Intervall $(0, T]$ und für jedes $k \in \mathbb{R}$ eine eindeutige Lösung.

(ii) Das Anfangswertproblem $y' = y^2$, $y(0) = y_0$, mit $y_0 > 0$ besitzt auf dem Intervall $[0, T_*]$ mit $T_* = 1/y_0$ die eindeutige Lösung $y(t) = (T_* - t)^{-1}$.

(iii) Das Anfangswertproblem $y' = y^{1/2}$, $y(0) = 0$, besitzt die Lösungen $y(t) = 0$ sowie $y(t) = t^2/4$.

Häufig besitzt die Lösung eines Anfangswertproblems höhere Regularitätseigenschaften als lediglich Differenzierbarkeit.

Satz 20.3 Gilt $f \in C^m([0, T] \times \mathbb{R}^n)$, so folgt $y \in C^{m+1}([0, T])$. Im Fall $m \geq 1$ sind Lösungen zugehöriger Anfangswertprobleme eindeutig.

Beweis Übungsaufgabe. □

20.2 Lemma von Gronwall

Das *Lemma von Gronwall* beschränkt das Wachstum der Lösung einer Differenzialgleichung.

Lemma 20.2 Seien $u \in C^0([0, T])$ und $\alpha, \beta \in \mathbb{R}$ mit $\beta \geq 0$, sodass

$$u(t) \leq \alpha + \beta \int_0^t u(s) \, ds$$

für alle $t \in [0, T]$ gilt. Dann folgt für alle $t \in [0, T]$, dass

$$u(t) \leq \alpha e^{\beta t}.$$

Beweis Es sei $v \in C^1([0, T])$ definiert durch

$$v(t) = e^{-\beta t} \int_0^t \beta u(s) \, ds.$$

Die Produktregel und die Voraussetzungen des Lemmas implizieren

$$v'(t) = -\beta e^{-\beta t} \int_0^t \beta u(s) ds + e^{-\beta t} \beta u(t) \leq \beta e^{-\beta t} \alpha.$$

Mit $v(0) = 0$ folgt

$$e^{-\beta t} \int_0^t \beta u(s) ds = v(t) = \int_0^t v'(s) ds \leq \beta \alpha \int_0^t e^{-\beta s} ds = \alpha(1 - e^{-\beta t}).$$

Multiplikation mit $e^{\beta t}$ führt auf

$$u(t) \leq \alpha + \int_0^t \beta u(s) ds \leq \alpha + \alpha e^{\beta t} (1 - e^{-\beta t}) = \alpha e^{\beta t}$$

und beweist das Lemma. \square

Bemerkung 20.2 Das Lemma von Gronwall wird häufig in differenzierlicher Form angegeben. Die Voraussetzung lautet dann $u'(t) \leq \beta u(t)$ und aus der resultierenden Ungleichung $(\log u)' = u'/u \leq \beta$ ist ersichtlich, dass u höchstens exponentiell wächst.

20.3 Stabilität

Als *Stabilität eines Anfangswertproblems* bezeichnet man die Konditionierung der zugehörigen mathematischen Aufgabe, das heißt die Auswirkungen von Störungen auf Lösungen des Anfangswertproblems. Wir nehmen dazu an, dass $y \in C^1([0, T])$ die eindeutige Lösung des Anfangswertproblems

$$y'(t) = f(t, y(t)), \quad y(0) = y_0$$

ist, und dass für Störungen \tilde{f} und \tilde{y}_0 der Funktion f und der Anfangsdaten y_0 die Funktion $\tilde{y} \in C^1([0, T])$ die eindeutige Lösung des zugehörigen gestörten Anfangswertproblems

$$\tilde{y}'(t) = \tilde{f}(t, \tilde{y}(t)), \quad \tilde{y}(0) = \tilde{y}_0$$

ist. Unter der Annahme, dass die Störungen klein sind, lässt sich zeigen, dass auch y und \tilde{y} für gewisse Zeiten nahe beieinander liegen.

Satz 20.4 Seien $f, \tilde{f} \in C^0([0, T] \times \mathbb{R})$, sodass ein $\delta > 0$ existiert mit

$$|f(t, v) - \tilde{f}(t, v)| \leq \delta$$

für alle $t \in [0, T]$ und $v \in \mathbb{R}$, und sei f uniform Lipschitz-stetig bezüglich des zweiten Arguments, das heißt es existiert ein $L \geq 0$, sodass

$$|f(t, v) - f(t, w)| \leq L|v - w|$$

für alle $t \in [0, T]$ und alle $v, w \in \mathbb{R}$ gilt. Ferner seien $y_0, \tilde{y}_0 \in \mathbb{R}$ mit $|y_0 - \tilde{y}_0| \leq \delta_0$ für ein $\delta_0 > 0$. Seien $y, \tilde{y} \in C^1([0, T] \times \mathbb{R})$ Lösungen der Anfangswertprobleme

$$\begin{aligned} y'(t) &= f(t, y(t)), & y(0) &= y_0, \\ \tilde{y}'(t) &= \tilde{f}(t, \tilde{y}(t)), & \tilde{y}(0) &= \tilde{y}_0 \end{aligned}$$

in $[0, T]$. Dann gilt

$$\sup_{t \in [0, T]} |y(t) - \tilde{y}(t)| \leq (\delta_0 + \delta T)e^{LT}.$$

Beweis Die Differenz $y - \tilde{y}$ erfüllt die Integralgleichung

$$y(t) - \tilde{y}(t) = y_0 - \tilde{y}_0 + \int_0^t f(s, y(s)) - \tilde{f}(s, \tilde{y}(s)) \, ds$$

und dies impliziert, dass

$$|y(t) - \tilde{y}(t)| \leq |y_0 - \tilde{y}_0| + \int_0^t |f(s, y(s)) - \tilde{f}(s, \tilde{y}(s))| \, ds.$$

Die Dreiecksungleichung und die Voraussetzungen an f zeigen, dass

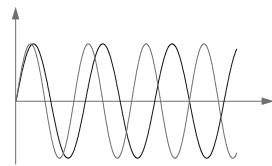
$$\begin{aligned} |f(s, y(s)) - \tilde{f}(s, \tilde{y}(s))| &\leq |f(s, y(s)) - f(s, \tilde{y}(s))| + |f(s, \tilde{y}(s)) - \tilde{f}(s, \tilde{y}(s))| \\ &\leq L|y(s) - \tilde{y}(s)| + \delta \end{aligned}$$

gilt, und mit $|y_0 - \tilde{y}_0| \leq \delta_0$ folgt

$$|y(t) - \tilde{y}(t)| \leq \delta_0 + \delta t + L \int_0^t |y(s) - \tilde{y}(s)| \, ds.$$

Abb. 20.1 Kleine Störungen

von Anfangsdaten können
sich im Langzeitverhalten
bemerkbar machen



Für die Funktion $u(t) = |y(t) - \tilde{y}(t)|$ gilt mit $\alpha = \delta_0 + \delta T$ und $\beta = L$ also

$$u(t) \leq \alpha + \beta \int_0^t u(s) \, ds.$$

Das Lemma von Gronwall impliziert $u(t) \leq \alpha e^{\beta t} \leq \alpha e^{\beta T}$, woraus die Aussage des Satzes folgt. \square

Bemerkung 20.3 Der Fehler in der Lösung der Differenzialgleichung ist proportional zu δ_0 und δ , jedoch ist der Proportionalitätsfaktor exponentiell abhängig von T und L . Das Anfangswertproblem ist also gut konditioniert beziehungsweise stabil, sofern LT hinreichend klein ist.

Beispiel 20.2 Betrachtet man zwei Federpendel mit Federkonstanten D und \tilde{D} , so geraten die Lösungen y und \tilde{y} außer Phase und die Lösungen unterscheiden sich für große Zeitpunkte stark voneinander, s. Abb. 20.1. Dies spiegelt die exponentielle Abhängigkeit vom Zeithorizont T wider.

20.4 Lernziele, Quiz und Anwendung

Sie sollten in der Lage sein, ein Anfangswertproblem als äquivalente Integralgleichung umformulieren zu können. Darauf basierend sollten Sie die Beweisideen des Satzes von Picard–Lindelöf erläutern können. Das Gronwall-Lemma sollten Sie herleiten und seine Bedeutung für die Konditionierung von Anfangswertproblemen erklären können.

Quiz 20.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Jede Lösung der Differenzialgleichung $y'' = c^2 y$ ist von der Form $y(t) = \alpha \sin(ct) + \beta \cos(ct)$	
Gilt $f \in C^1(\mathbb{R})$, so besitzt das Anfangswertproblem $y' = f(y)$, $y(0) = y_0$, für alle $y_0 \in \mathbb{R}$ und $T > 0$ eine Lösung $y \in C^1([0, T])$	
Jede Kontraktion $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ist stetig differenzierbar	
Jedes Anfangswertproblem $y' = f(t, y)$, $y(0) = y_0$, lässt sich äquivalent als Integralgleichung formulieren	
Sind $y, \tilde{y} \in C^1([0, T])$ Lösungen der Differenzialgleichung $y' = f(y)$ mit einer Lipschitz-stetigen Abbildung $f : \mathbb{R} \rightarrow \mathbb{R}$, so gilt $ y(t) - \tilde{y}(t) \leq y(0) - \tilde{y}(0) $ für alle $t \in [0, T]$	

Anwendung 20.1 Die Flugbahn einer Rakete im Schwerefeld der Erde lässt sich durch eine Vereinfachung des Zweikörperproblems beschreiben, wobei angenommen werden kann, dass der Erdmittelpunkt unverändert bleibt und als $y_{\text{Erde}} = 0$ angesetzt werden kann. Weiter sei angenommen, dass die Rakete senkrecht zur Erdoberfläche fliegt und der Treibstoff aufgebraucht ist, sodass keine eigene Beschleunigung mehr stattfindet. Zeigen Sie, dass die Höhe z der Rakete durch die Gleichung

$$z''(t) = \frac{a}{(z(t))^2}$$

mit einer geeigneten Konstanten a beschrieben wird und bestimmen Sie die Lösung für verschiedene Anfangsgeschwindigkeiten, indem Sie den Ansatz $z(t) = \alpha(t - t_0)^\beta$ verwenden. Diskutieren Sie hinreichende Bedingungen für die globale Existenz der Lösung.

21.1 Euler-Verfahren

Ein einfaches Verfahren zur numerischen Approximation von Lösungen gewöhnlicher Differenzialgleichungen der Form

$$y'(t) = f(t, y(t)), \quad y(0) = y_0$$

ergibt sich aus der Approximation der Ableitung durch einen (*Vorwärts-)*Differenzenquotienten, das heißt aus

$$y'(t) \approx \frac{y(t + \tau) - y(t)}{\tau}$$

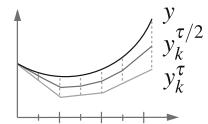
mit einer festen Schrittweite $\tau > 0$. Ist $y \in C^1([0, T])$, so konvergiert die rechte Seite für $\tau \rightarrow 0$ gegen $y'(t)$. Die Approximation führt auf

$$y(t + \tau) \approx y(t) + \tau f(t, y(t))$$

und bedeutet, dass sofern eine Approximation von y zum Zeitpunkt t bekannt ist, eine Approximation zum Zeitpunkt $t + \tau$ direkt berechnet werden kann. Beginnend mit den Anfangsdaten bei $t_0 = 0$ ergeben sich die Approximationen zu den Zeitschritten $t_k = k\tau$, $k = 1, 2, \dots, K$, wobei K die größte natürliche Zahl mit der Eigenschaft $K\tau \leq T$ sei, was durch $K = \lfloor T/\tau \rfloor$ bezeichnet wird.

Algorithmus 21.1 (Explizites Euler-Verfahren) Seien $f \in C^0([0, T] \times \mathbb{R})$, $y_0 \in \mathbb{R}$ und $\tau > 0$. Setze $k = 0$ und $K = \lfloor T/\tau \rfloor$.

Abb. 21.1 Euler-Verfahren approximieren Lösungen durch Polygonzüge



(1) Berechne

$$y_{k+1} = y_k + \tau f(t_k, y_k).$$

(2) Stoppe falls $k + 1 > K$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Geometrisch wird die Kurve $t \mapsto y(t)$ durch einen Polygonzug approximiert, der die Werte $(y_k)_{k=0,\dots,K}$ verbindet, s. Abb. 21.1. Daher wird das Verfahren auch als *Eulersches Polygonzugverfahren* bezeichnet.

Bemerkung 21.1 Im Allgemeinen stimmen die Approximationen y_k nicht mit der exakten Lösung $y(t_k)$ zu den Zeitpunkten t_k , $k = 1, 2, \dots, K$, überein.

Definition 21.1 Ein Verfahren der Gestalt

$$y_{k+1} = y_k + \tau \Phi(t_k, y_k, y_{k+1}, \tau), \quad k = 0, 1, \dots, K - 1,$$

heißt *Einschrittverfahren* mit *Inkrementfunktion* $\Phi : [0, T] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$. Ist Φ unabhängig von y_{k+1} , so wird das Verfahren als *explizites* andernfalls als *implizites Verfahren* bezeichnet.

Das *implizite Euler-Verfahren* ergibt sich aus der Verwendung des *Rückwärtsdifferenzenquotienten*

$$y'(t) \approx \frac{y(t) - y(t - \tau)}{\tau}$$

und der Auswertung der Differenzialgleichung bei t_{k+1} .

Algorithmus 21.2 (Implizites Euler-Verfahren) Seien $f \in C^0([0, T] \times \mathbb{R})$, $y_0 \in \mathbb{R}$ und $\tau > 0$. Setze $k = 0$ und $K = \lfloor T/\tau \rfloor$.

(1) Bestimme $y_{k+1} \in \mathbb{R}$ als Lösung der Gleichung

$$y_{k+1} = y_k + \tau f(t_{k+1}, y_{k+1}).$$

(2) Stoppe falls $k + 1 > K$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Bemerkungen 21.2 (i) Im Gegensatz zum expliziten ist beim impliziten Euler-Verfahren in jedem Iterationsschritt ein Gleichungssystem zu lösen. Dessen Lösbarkeit ist im jeweiligen Fall sicherzustellen.

(ii) Für das explizite und implizite Euler-Verfahren sind die Inkrementfunktionen Φ gegeben durch

$$\Phi_{\text{expl}}(t_k, y_k, y_{k+1}, \tau) = f(t_k, y_k), \quad \Phi_{\text{impl}}(t_k, y_k, y_{k+1}, \tau) = f(t_k + \tau, y_{k+1}).$$

Eine höhere Genauigkeit ergibt sich durch Verwendung der beiden Approximationen y_k und y_{k+1} .

Beispiel 21.1 Das *Mittelpunktverfahren* ist definiert durch

$$\Phi(t_k, y_k, y_{k+1}, \tau) = f(t_k + \tau/2, (y_k + y_{k+1})/2).$$

21.2 Konsistenz

Setzt man die Funktionswerte $y(t_k)$ der exakten Lösung einer Differenzialgleichung ausgewertet an den Zeitschritten t_k , $k = 0, 1, \dots, K$, in ein numerisches Verfahren ein, so lässt sich beurteilen, wie akkurat das Verfahren ist. Im Fall des expliziten Euler-Verfahrens gilt unter Verwendung der Differenzialgleichung ausgewertet bei t_k , dass

$$\frac{y(t_{k+1}) - y(t_k)}{\tau} - f(t_k, y(t_k)) = \frac{y(t_{k+1}) - y(t_k)}{\tau} - y'(t_k).$$

Eine Taylor-Approximation zeigt

$$\left| \frac{y(t_{k+1}) - y(t_k)}{\tau} - y'(t_k) \right| \leq \frac{\tau}{2} \sup_{t \in [t_k, t_{k+1}]} |y''(t)|.$$

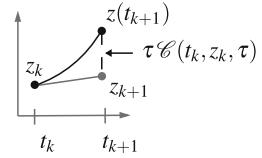
Die Werte der exakten Lösung erfüllen das numerische Verfahren also bis auf den *Konsistenzterm* $(\tau/2) \|y''\|_{C^0([0, T])}$. Um dieses Vorgehen zu verallgemeinern, betrachtet man für einen gegebenen Wert z_k zum Zeitschritt t_k das *lokale Anfangswertproblem*

$$z'(t) = f(t, z(t)), \quad t \in [t_k, t_{k+1}], \quad z(t_k) = z_k.$$

Die Abweichung der Lösung $z(t_{k+1})$ zum Zeitpunkt t_{k+1} von der durch das Einschrittverfahren definierten Approximation

$$z_{k+1} = z_k + \tau \Phi(t_k, z_k, z_{k+1}, \tau)$$

Abb. 21.2 Der Diskretisierungsfehler eines Zeitschritts definiert die Konsistenz eines Verfahrens



ist gegeben durch

$$z(t_{k+1}) - z_{k+1} = z(t_{k+1}) - z_k - \tau \Phi(t_k, z_k, z_{k+1}, \tau)$$

s. Abb. 21.2.

Definition 21.2 Der *lokale Diskretisierungsfehler* $C(t_k, z_k, \tau)$ der Inkrementfunktion Φ ist definiert durch

$$C(t_k, z_k, \tau) = \frac{z(t_{k+1}) - z_k}{\tau} - \Phi(t_k, z_k, z_{k+1}, \tau).$$

Das durch Φ definierte Verfahren heißt *konsistent von der Ordnung* $p \geq 0$, falls für alle Funktionen $f \in C^p([0, T] \times \mathbb{R})$, die uniform Lipschitz-stetig im zweiten Argument sind, und $k = 0, 1, \dots, K-1$ sowie $z_k \in \mathbb{R}$ gilt, dass

$$C(t_k, z_k, \tau) = \mathcal{O}(\tau^p)$$

für $\tau \rightarrow 0$, das heißt falls $c_1, c_2 > 0$ existieren, sodass $|C(t_k, z_k, \tau)| \leq c_1 \tau^p$ für alle $0 < \tau \leq c_2$ gilt.

Bemerkungen 21.3 (i) Ist Φ Lipschitz-stetig im dritten Argument, so kann statt z_{k+1} auch $z(t_{k+1})$ zur Bestimmung der Konsistenzordnung verwendet werden, das heißt unter Verwendung von $z_k = z(t_k)$,

$$\tilde{C}(t_k, z_k, \tau) = \frac{z(t_{k+1}) - z(t_k)}{\tau} - \Phi(t_k, z(t_k), z(t_{k+1}), \tau).$$

Im Fall $z_k = y(t_k)$ entspricht dies gerade dem Einsetzen der Funktionswerte der exakten Lösung in das numerische Schema.

(ii) Für $z_k = y(t_k)$ stimmt die lokale Lösung z auf dem Intervall $[t_k, t_{k+1}]$ mit y überein und es gilt

$$\tilde{C}(t_k, y(t_k), \tau) = \frac{y(t_{k+1}) - y(t_k)}{\tau} - \Phi(t_k, y(t_k), y(t_{k+1}), \tau).$$

Wir werden im Folgenden meist von diesem Ausdruck Gebrauch machen.

Beispiele 21.2 (i) Für das explizite Euler-Verfahren gilt unter Verwendung von $\Phi(t_k, z_k, z_{k+1}, \tau) = f(t_k, z_k)$, $z_k = z(t_k)$ und $z'(t_k) = f(t_k, z(t_k))$, dass $C(t_k, z_k, \tau) = \tilde{C}(t_k, z_k, \tau)$ mit

$$C(t_k, z_k, \tau) = \frac{z(t_{k+1}) - z(t_k)}{\tau} - z'(t_k)$$

gilt. Mit einer Taylor-Approximation folgt

$$|C(t_k, z_k, \tau)| \leq \frac{\tau}{2} \sup_{t \in [t_k, t_{k+1}]} |z''(t)|,$$

sodass das explizite Euler-Verfahren konsistent von der Ordnung $p = 1$ ist.

(ii) Eine analoge Argumentation zeigt, dass das implizite Euler-Verfahren ebenfalls die Konsistenzordnung $p = 1$ besitzt.

(iii) Ebenfalls aus Taylor-Approximationen ergibt sich die Konsistenzordnung $p = 2$ des Mittelpunktverfahrens.

21.3 Diskretes Gronwall-Lemma und Konvergenz

Die Konsistenz eines Einschrittverfahrens ist ein Maß für die Exaktheit eines Verfahrens. Darauf basierend werden wir zeigen, dass damit auch die Approximationen $(y_k)_{k=0,\dots,K}$ nahe an den exakten Funktionswerten $(y(t_k))_{k=0,\dots,K}$ liegen. Es sei dazu ein Einschrittverfahren

$$y_{k+1} = y_k + \tau \Phi(t_k, y_k, y_{k+1}, \tau)$$

mit Konsistenzordnung p gegeben, das heißt es gelte

$$\tilde{C}(t_k, y(t_k), \tau) = \frac{y(t_{k+1}) - y(t_k)}{\tau} - \Phi(t_k, y(t_k), y(t_{k+1}), \tau) = \mathcal{O}(\tau^p).$$

Die nachfolgende Fehlerabschätzung beruht auf der Interpretation der exakten Lösungswerte $(y(t_k))_{k=0,\dots,K}$ als Lösung des mit Termen der Ordnung $\mathcal{O}(\tau^p)$ gestörten numerischen Verfahrens. Dafür wird die folgende *diskrete Version des Lemmas von Gronwall* benötigt.

Lemma 21.1 Seien $(u_k)_{k=0,\dots,K}$ eine Folge nichtnegativer, reeller Zahlen und $\alpha, \beta \in \mathbb{R}$ mit $\beta \geq 0$, sodass

$$u_\ell \leq \alpha + \tau \sum_{k=0}^{\ell-1} \beta u_k$$

für alle $\ell = 0, 1, \dots, K$ gilt. Dann folgt für alle $\ell = 0, 1, \dots, K$, dass

$$u_\ell \leq \alpha \exp(\ell \tau \beta).$$

Beweis Übungsaufgabe. □

Interpretiert man die Summe als Quadraturformel, so wird die Beziehung zum kontinuierlichen Lemma von Gronwall deutlich.

Definition 21.3 Ein Einschrittverfahren heißt *konvergent von der Ordnung $p \geq 0$* , falls für alle Funktionen $f \in C^p([0, T] \times \mathbb{R})$, die uniform Lipschitz-stetig im zweiten Argument sind, Anfangsdaten $y_0 \in \mathbb{R}$ und die exakte Lösung $y \in C^{p+1}([0, T])$ sowie die Approximationen $(y_\ell)_{\ell=0,\dots,K}$ gilt

$$\max_{\ell=0,\dots,K} |y(t_\ell) - y_\ell| = \mathcal{O}(\tau^p).$$

Mit Hilfe des diskreten Gronwall-Lemmas führt die Konsistenz der Ordnung p eines Verfahrens auf die Konvergenz der Ordnung p des Verfahrens, das heißt wir erhalten eine *allgemeine Fehlerabschätzung für Einschrittverfahren*.

Satz 21.1 Das durch Φ definierte Einschrittverfahren sei wohldefiniert und konsistent von der Ordnung p . Die Inkrementfunktion Φ sei uniform Lipschitz-stetig im zweiten und dritten Argument, das heißt es existiere $M \geq 0$, sodass

$$|\Phi(t, a_1, b_1, \tau) - \Phi(t, a_2, b_2, \tau)| \leq M(|a_1 - a_2| + |b_1 - b_2|)$$

für alle $t \in [0, T]$, $a_1, a_2, b_1, b_2 \in \mathbb{R}$ und $\tau > 0$. Gilt $\tau \leq 1/(2M)$, so folgt

$$\max_{\ell=0,\dots,K} |y(t_\ell) - y_\ell| \leq 2cT\tau^p \exp(4MT)$$

mit einer von τ unabhängigen Konstanten $c \geq 0$.

Beweis Für die Funktionswerte $(y(t_k))_{k=0,\dots,K}$ gilt nach Definition des Konsistenzterms

$$y(t_{k+1}) = y(t_k) + \tau\Phi(t_k, y(t_k), y(t_{k+1}), \tau) + \tau\tilde{C}(t_k, y(t_k), \tau),$$

während die Approximationen $(y_k)_{k=0,\dots,K}$ durch

$$y_{k+1} = y_k + \tau\Phi(t_k, y_k, y_{k+1}, \tau)$$

definiert sind. Eine Subtraktion der beiden Identitäten führt auf

$$\begin{aligned} y(t_{k+1}) - y_{k+1} &= y(t_k) - y_k + \tau[\Phi(t_k, y(t_k), y(t_{k+1}), \tau) - \Phi(t_k, y_k, y_{k+1}, \tau)] \\ &\quad + \tau\tilde{C}(t_k, y(t_k), \tau). \end{aligned}$$

Mit der Dreiecksungleichung, der Lipschitz-Stetigkeit von Φ und der Konsistenzordnung p folgt

$$|y(t_{k+1}) - y_{k+1}| = |y(t_k) - y_k| + \tau M (|y(t_k) - y_k| + |y(t_{k+1}) - y_{k+1}|) + c \tau^{p+1}.$$

Mit der Definition $u_k = |y(t_k) - y_k|$ ergibt sich

$$(1 - \tau M)u_{k+1} \leq (1 + \tau M)u_k + c \tau^{p+1},$$

beziehungsweise

$$u_{k+1} \leq \frac{1 + \tau M}{1 - \tau M} u_k + \frac{c}{1 - \tau M} \tau^{p+1}.$$

Subtraktion von u_k auf beiden Seiten liefert unter Verwendung von $1 - \tau M \geq 1/2$

$$\begin{aligned} u_{k+1} - u_k &\leq \left(\frac{1 + \tau M}{1 - \tau M} - 1 \right) u_k + \frac{c}{1 - \tau M} \tau^{p+1} \\ &= \tau \frac{2M}{1 - \tau M} u_k + \frac{c\tau}{1 - \tau M} \tau^p \\ &\leq \tau 4Mu_k + 2c\tau\tau^p. \end{aligned}$$

Eine Summation über $k = 0, 1, \dots, \ell - 1$ mit $0 \leq \ell \leq K$ führt auf

$$u_\ell - u_0 \leq 4M\tau \sum_{k=0}^{\ell-1} u_k + 2cK\tau\tau^p.$$

Die Folge $(u_k)_{k=0,\dots,K}$ erfüllt also die Voraussetzungen des diskreten Gronwall-Lemmas mit

$$\alpha = u_0 + 2c(K\tau)\tau^p, \quad \beta = 4\tau M$$

und mit $u_0 = 0$ und $\ell\tau \leq K\tau \leq T$ folgt die Behauptung. \square

Bemerkungen 21.4 (i) Ähnlich wie in der Stabilitätsabschätzung hängt die Konstante in der Fehlerabschätzung kritisch von der Größe MT ab.

(ii) Der Beweis des Satzes zeigt, dass es genügt, die Anfangsdaten mit einer Genauigkeit $|y_0 - y(0)| = \mathcal{O}(\tau^p)$ zu approximieren.

Für den Spezialfall des expliziten Euler-Verfahrens lässt sich der Beweis der Konvergenzaussage vereinfachen.

Beispiel 21.3 Im Fall des expliziten Euler-Verfahrens gilt

$$\begin{aligned} y_{k+1} &= y_k + \tau f(t_k, y_k), \\ y(t_{k+1}) &= y(t_k) + \tau f(t_k, y(t_k)) + \tau C(t_k, y(t_k), \tau) \end{aligned}$$

mit $|C(t_k, y(t_k), \tau)| \leq c\tau$. Für den Fehler $u_k = |y(t_k) - y_k|$ folgt durch Subtraktion der beiden Gleichungen, sofern f uniform Lipschitz-stetig im zweiten Argument ist, dass

$$u_{k+1} \leq u_k + \tau L u_k + c\tau^2$$

und eine Summation über $k = 0, 1, \dots, \ell - 1$ für alle $0 \leq \ell \leq K$ ergibt

$$u_\ell - u_0 \leq \ell c\tau^2 + \tau \sum_{k=0}^{\ell-1} L u_k.$$

Mit $\ell\tau \leq T$ und $u_0 = 0$ führt das diskrete Gronwall-Lemma auf

$$u_\ell \leq cT\tau \exp(LT)$$

für alle $0 \leq \ell \leq K$, beziehungsweise

$$\max_{\ell=0,\dots,K} |y(t_\ell) - y_\ell| \leq cT\tau \exp(LT).$$

21.4 Verfahren höherer Ordnung

Die Konsistenzordnung der Euler-Verfahren ist durch die definierenden Taylor-Formeln gegeben. Dies motiviert eine Approximation höherer Genauigkeit zu verwenden, beispielsweise

$$\frac{y(t_{k+1}) - y(t_k)}{\tau} = y'(t_k) + \frac{\tau}{2} y''(t_k) + \mathcal{O}(\tau^2),$$

sofern $y \in C^3([0, T])$ gilt. Basierend auf dieser Identität existieren verschiedene Möglichkeiten, eine Inkrementfunktion zu konstruieren.

Beispiele 21.4 ([6,7]) (i) Differenziert man die Differenzialgleichung $y' = f(t, y)$ bezüglich t , so erhält man mit den partiellen Ableitungen $\partial_t f$ und $\partial_y f$ von f , dass

$$\begin{aligned} y''(t) &= \partial_t f(t, y(t)) + \partial_y f(t, y(t)) y'(t) \\ &= \partial_t f(t, y(t)) + \partial_y f(t, y(t)) f(t, y(t)). \end{aligned}$$

Die Verwendung dieser Identität in der obigen Formel zeigt, dass der Ausdruck

$$\frac{y(t_{k+1}) - y(t_k)}{\tau} - \frac{\tau}{2} \left(\partial_t f(t_k, y(t_k)) + \partial_y f(t_k, y(t_k)) f(t_k, y(t_k)) \right)$$

die Ableitung $y'(t_k)$ bis auf einen Fehler $\mathcal{O}(\tau^2)$ approximiert und motiviert die Verwendung der expliziten Inkrementfunktion

$$\Phi(t_k, y_k, y_{k+1}, \tau) = f(t_k, y_k) + \frac{\tau}{2} \left(\partial_t f(t_k, y_k) + \partial_y f(t_k, y_k) f(t_k, y_k) \right).$$

Die Rechnungen implizieren $C(t_k, y_k, \tau) = \mathcal{O}(\tau^2)$.

(ii) Mit zu bestimmenden Koeffizienten $a, b, c, d \in \mathbb{R}$ wird der Ansatz

$$\Phi(t, y, \tau) = af(t, y) + bf(t + c\tau, y + \tau df(t, y))$$

betrachtet und in die Definition des Konsistenzterms eingesetzt, wobei abkürzend t und y für t_k und y_k stehen. Unter Verwendung der Taylor-Approximation

$$f(t + c\tau, y + d\tau f(t, y)) = f(t, y) + \partial_t f(t, y)c\tau + \partial_y f(t, y)d\tau f(t, y) + \mathcal{O}(\tau^2)$$

ergeben sich die Bedingungen $a + b = 1$, $bc = 1/2$ und $bd = 1/2$ an die Parameter a, b, c, d für die Konsistenz der Ordnung $p = 2$. Die Lösung $a = b = 1/2$, $c = d = 1$ definiert das *Verfahren von Heun*

$$\Phi(t, y, \tau) = \frac{1}{2} \left(f(t, y) + f(t + \tau, y + \tau f(t, y)) \right)$$

und die Lösung $a = 0$, $b = 1$, $c = d = 1/2$ das *Euler–Collatz-Verfahren*

$$\Phi(t, y, \tau) = f \left(t + \frac{\tau}{2}, y + \frac{\tau}{2} f(t, y) \right).$$

Bemerkung 21.5 Die in den Verfahren auftretenden Terme

$$y_k + \theta \tau f(t_k, y_k) \approx y_k + \theta \tau y'(t_k)$$

approximieren die unbekannten Werte $y(t_{k+1})$ im Fall $\theta = 1$ und $y(t_{k+1/2}) = (y_k + y_{k+1})/2$ im Fall $\theta = 1/2$, wobei $t_{k+1/2} = t_k + \tau/2$ sei.

21.5 Lernziele, Quiz und Anwendung

Sie sollten einige Einschrittverfahren herleiten und deren Unterschiede aufzeigen können. Sie sollten den Begriff der Konsistenz motivieren und definieren sowie seine Verwendung zur Herleitung von Fehlerabschätzungen erklären können.

Quiz 21.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Die Inkrementfunktion $\Phi(t, a, b, \tau) = \alpha(a + b)/2$ definiert ein Einschrittverfahren für die Differenzialgleichung $y' = \alpha y$	
Explizite Einschrittverfahren sind stets wohldefiniert	
Der lokale Diskretisierungsfehler des expliziten Euler-Verfahrens für die Differenzialgleichung $y' = \lambda y$ ist gegeben durch $(z(t_{k+1}) - z_k)/\tau - \lambda z_k$	
Die Inkrementfunktion $\Phi(t, a, b, \tau) = f(t + \tau/2, a + \tau f(t, a)/2)$ definiert ein Verfahren der Konsistenzordnung $p = 2$	
Im Allgemeinen nimmt der Fehler $ y_\ell - y(t_\ell) $ mit jedem Zeitschritt zu	

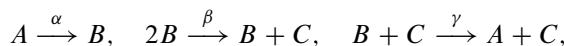
Anwendung 21.1

- (i) Die Geschwindigkeit der chemischen Reaktion zweier Stoffe A und B mit Produkt $2B$ wird nach einem Massenwirkungsprinzip durch einen Reaktionskoeffizienten α und die Differenzialgleichungen

$$C'_A = -\alpha C_A C_B, \quad C'_B = \alpha C_A C_B$$

beschrieben, wobei $C_A, C_B : [0, T] \rightarrow [0, 1]$ die jeweiligen Konzentrationen angeben. In der Reaktionsgleichung wird dies durch die Notation $A + B \xrightarrow{\alpha} 2B$ berücksichtigt. Zeigen Sie, dass die Summe der Konzentrationen $C_A + C_B$ konstant ist.

- (ii) Wir betrachten das Reaktionsschema



mit den Reaktionskoeffizienten $\alpha = 0.04$, $\beta = 3 \cdot 10^7$, $\gamma = 10^4$, das heißt beispielsweise, dass der Stoff B sehr schnell in den Stoff C umgewandelt wird. Formulieren Sie ein System von Differenzialgleichungen zur Beschreibung des Reaktionsschemas, zeigen Sie, dass die Summe der Konzentrationen konstant ist und bestimmen Sie numerisch das Maximum der Konzentration des Stoffs B , wenn zu Beginn des Vorgangs nur der Stoff A vorliegt.

- (iii) Testen Sie verschiedene MATLAB-Routinen zur numerischen Lösung des Problems im Zeitintervall $[0, 1/2]$ und kommentieren Sie die Ergebnisse.

22.1 Motivation

Die auf Taylor-Approximationen basierende Konstruktion numerischer Verfahren mit höherer Konsistenzordnung führt in der Regel auf Schemata, in denen in jedem Schritt viele Funktionsauswertungen insbesondere der Ableitungen erforderlich sind. Dies ist im Allgemeinen mit sehr hohem Aufwand verbunden. Der Ausgangspunkt der Entwicklung von Verfahren, die die Auswertung von Ableitungen von f vermeiden, ist eine lokale Integraldarstellung der Differenzialgleichung $y' = f(t, y)$. Es gilt

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} y'(s) \, ds = y(t_k) + \int_{t_k}^{t_{k+1}} f(s, y(s)) \, ds.$$

Approximiert man das Integral durch den Wert des Integranden an der Stelle t_k multipliziert mit der Länge des Integrationsbereichs $t_{k+1} - t_k = \tau$, so ergibt sich

$$y(t_{k+1}) \approx y(t_k) + \tau f(t_k, y(t_k)),$$

und dies motiviert das explizite Euler-Verfahren. Es ist naheliegend, exaktere Quadraturformeln anzuwenden, um Verfahren höherer Genauigkeit zu erhalten. Im Fall der Mittelpunktregel ergibt sich mit $t_{k+1/2} = (k + 1/2)\tau$

$$y(t_{k+1}) \approx y(t_k) + \tau f(t_{k+1/2}, y(t_{k+1/2})).$$

Der Funktionswert $y(t_{k+1/2})$ kann mit Hilfe einer Approximation von $y(t_k)$ angenähert werden, denn eine Taylor-Approximation und die Auswertung der Differenzialgleichung

bei t_k zeigen für ein $\xi \in [t_k, t_{k+1/2}]$

$$\begin{aligned} y(t_{k+1/2}) &= y(t_k) + \frac{\tau}{2}y'(t_k) + \frac{\tau^2}{4}y''(\xi) \\ &= y(t_k) + \frac{\tau}{2}f(t_k, y(t_k)) + \frac{\tau^2}{4}y''(\xi). \end{aligned}$$

Insgesamt führt dies auf das Euler–Collatz-Verfahren

$$y_{k+1} = y_k + \tau f\left(t_k + \frac{\tau}{2}, y_k + \frac{\tau}{2}f(t_k, y_k)\right).$$

Mit einer Taylor-Approximation der rechten Seite zeigt man, dass dieses Verfahren die Konsistenzordnung $p = 2$ besitzt.

22.2 Runge–Kutta-Verfahren

Ist $(\alpha_\ell, \gamma_\ell)_{\ell=1,\dots,m}$ eine Quadraturformel auf dem Intervall $[0, 1]$, so definiert $(t_k + \tau\alpha_\ell, \tau\gamma_\ell)_{\ell=1,\dots,m}$ eine Quadraturformel auf $[t_k, t_{k+1}]$ und wir erhalten

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} y'(s) \, ds \approx y(t_k) + \tau \sum_{\ell=1}^m \gamma_\ell \eta_\ell^k$$

mit den Approximationen

$$\eta_\ell^k \approx y'(t_k + \tau\alpha_\ell) = f(t_k + \tau\alpha_\ell, y(t_k + \tau\alpha_\ell)).$$

Die rechte Seite wird approximiert mittels

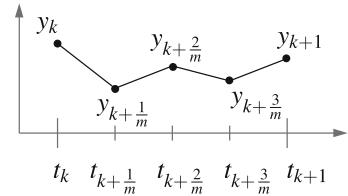
$$y(t_k + \tau\alpha_\ell) \approx y(t_k) + \tau\alpha_\ell \sum_{j=1}^m \beta_{\ell j} y'(t_k + \tau\alpha_j) \approx y(t_k) + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j^k$$

und damit lassen sich die Größen η_ℓ^k als Lösung des nichtlinearen Gleichungssystems

$$\eta_\ell^k = f\left(t_k + \tau\alpha_\ell, y_k + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j^k\right)$$

für $\ell = 1, 2, \dots, m$ bestimmen. Dies führt zu folgender Definition.

Abb. 22.1 Runge–Kutta–Verfahren verwenden implizit definierte Zwischenschritte



Definition 22.1 Für $\alpha_\ell, \beta_{\ell j}, \gamma_\ell \in \mathbb{R}$, $j, \ell = 1, 2, \dots, m$, ist ein m -stufiges Runge–Kutta–Verfahren definiert durch

$$y_{k+1} = y_k + \tau \sum_{\ell=1}^m \gamma_\ell \eta_\ell^k, \quad \eta_\ell^k = f \left(t_k + \tau \alpha_\ell, y_k + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j^k \right).$$

Anschaulich verwendet ein Runge–Kutta–Verfahren Zwischenschritte $t_{k+s/m} = t_k + \tau \alpha_s$, $s = 1, 2, \dots, m$, und zugehörige Approximationen $y_{k+s/m} = y_k + \tau \sum_{\ell=1}^s \gamma_\ell \eta_\ell^k$, um y_{k+1} zu bestimmen, s. Abb. 22.1.

Bemerkung 22.1 Runge–Kutta–Verfahren sind Einschrittverfahren, wobei die Inkrementfunktion $\Phi(t_k, y_k, \tau) = \sum_{\ell=1}^m \gamma_\ell \eta_\ell^k$ durch die Lösung eines möglicherweise nichtlinearen Problems definiert ist. Im Sinne von Einschrittverfahren sind Runge–Kutta–Verfahren explizit, jedoch ist diese Sichtweise nicht sinnvoll.

Bemerkung 22.2 Ein Runge–Kutta–Verfahren ist durch das zugehörige *Butcher–Tableau* festgelegt, in dem die Koeffizienten schematisch wie in Tab. 22.1 angeordnet werden.

Beispiele 22.1 (i) Für $m = 1$, $\alpha_1 = 0$, $\beta_{11} = 0$, $\gamma_1 = 1$ ergibt sich das explizite Euler–Verfahren:

$$\eta_1^k = f(t_k, y_k), \quad y_{k+1} = y_k + \tau \eta_1^k.$$

(ii) Das Euler–Collatz–Verfahren und das Verfahren von Heun sind durch die in Tab. 22.2 gezeigten Butcher–Tableaus definiert.

Tab. 22.1 Butcher–Tableau eines Runge–Kutta–Verfahrens

α_1	β_{11}	\dots	β_{1m}
\vdots	\vdots		\vdots
α_m	β_{m1}	\dots	β_{mm}
	γ_1	\dots	γ_m

Tab. 22.2 Butcher-Tableaus des expliziten Euler-Verfahrens der Verfahren von Euler–Collatz und Heun, sowie des Trapezverfahrens

0 0	0 0 0	0 0 0	1 1/2 1/2
	1/2 1/2 0	1 1 0	0 0 0

(iii) Das *Trapezverfahren* ergibt sich aus der Verwendung der Trapezregel, das heißt

$$y_{k+1} = y_k + \frac{\tau}{2} (f(t_k, y_k) + f(t_{k+1}, y_{k+1})) = y_k + \frac{\tau}{2} \sum_{\ell=1}^2 \eta_\ell^k$$

mit $\eta_1^k = f(t_k, y_k)$ und $\eta_2^k = f(t_k + \tau, y_{k+1})$.

22.3 Wohlgestelltheit

Die Durchführung eines Iterationsschritts mit einem Runge–Kutta-Verfahren erfordert die Lösung eines Gleichungssystems. Ist β eine strikte untere Dreiecksmatrix, so lässt sich dieses explizit lösen.

Definition 22.2 Ein Runge–Kutta-Verfahren heißt *explizit*, falls $\beta_{\ell j} = 0$ für alle $1 \leq \ell \leq j \leq n$ gilt. Es heißt *implizit* andernfalls.

Bemerkung 22.3 Ist ein Runge–Kutta-Verfahren explizit, so lassen sich die Ausdrücke η_ℓ^k sukzessive bestimmen. Für $\ell = 1, 2, \dots, m$ gilt dann

$$\eta_\ell^k = f \left(t + \tau \alpha_\ell, y_k + \tau \sum_{j=1}^{\ell-1} \beta_{\ell j} \eta_j^k \right).$$

Beispiel 22.2 Beispiele expliziter, vierstufiger Runge–Kutta-Verfahren sind das klassische Runge–Kutta-Verfahren und die 3/8-Regel, die durch die in Tab. 22.3 gezeigten Butcher-Tableaus definiert sind.

Tab. 22.3 Butcher-Tableaus des klassischen Runge–Kutta-Verfahrens und der 3/8-Regel

0	0	0	0
1/2	1/2	1/3	1/3
1/2	0 1/2	2/3	-1/3 1
1	0 0 1	1	1 -1 1

0	0	0	0
1/2	1/2	1/3	1/3
1/2	0 1/2	2/3	-1/3 1
1	0 0 1	1	1 -1 1

Im impliziten Fall muss eine Fixpunktgleichung gelöst werden. Mit den Abkürzungen $t = t_k$, $y = y_k$ und $\eta_\ell = \eta_\ell^k$ ist ein Vektor $\eta = [\eta_1, \eta_2, \dots, \eta_m]^\top$ zu bestimmen, sodass

$$\eta_1 = f(t + \tau\alpha_1, y + \tau\beta_{11}\eta_1 + \tau\beta_{12}\eta_2 + \dots + \tau\beta_{1m}\eta_m),$$

⋮

$$\eta_m = f(t + \tau\alpha_m, y + \tau\beta_{m1}\eta_1 + \tau\beta_{m2}\eta_2 + \dots + \tau\beta_{mm}\eta_m)$$

gilt, was sich abstrakt als vektorielle Gleichung $\eta = \Psi(\eta)$ schreiben lässt.

Satz 22.1 Ist $f \in C^0([0, T] \times \mathbb{R})$ uniform Lipschitz-stetig im zweiten Argument mit Lipschitz-Konstante $L \geq 0$ und gilt

$$L\tau\|\beta\|_\infty < 1,$$

mit der Zeilensummennorm $\|\beta\|_\infty = \max_{\ell=1,\dots,m} \sum_{j=1}^m |\beta_{\ell j}|$, so ist Ψ eine Kontraktion bezüglich der Maximumsnorm auf \mathbb{R}^m und es existiert ein eindeutiger Fixpunkt $\eta \in \mathbb{R}^m$ von Ψ .

Beweis Seien $\xi, \zeta \in \mathbb{R}^m$. Dann gilt

$$\begin{aligned} \|\Psi(\xi) - \Psi(\zeta)\|_\infty &= \max_{\ell=1,\dots,m} \left| f \left(t + \tau\alpha_\ell, y + \tau \sum_{j=1}^m \beta_{\ell j} \xi_j \right) \right. \\ &\quad \left. - f \left(t + \tau\alpha_\ell, y + \tau \sum_{j=1}^m \beta_{\ell j} \zeta_j \right) \right| \\ &\leq \max_{\ell=1,\dots,m} L\tau \sum_{j=1}^m |\beta_{\ell j}| \max_{j=1,\dots,m} |\xi_j - \zeta_j| \\ &= L\tau\|\beta\|_\infty\|\xi - \zeta\|_\infty. \end{aligned}$$

Der Banachsche Fixpunktsatz impliziert im Fall $L\tau\|\beta\|_\infty < 1$ die Existenz eines eindeutigen Fixpunkts. \square

Bemerkung 22.4 Ein Fixpunkt der Kontraktion $\Psi : \mathbb{R}^m \rightarrow \mathbb{R}^m$ kann mit einem beliebigen Startwert $\xi^0 \in \mathbb{R}^m$ durch die Iteration $\xi^{i+1} = \Psi(\xi^i)$ approximiert werden. Unter geeigneten Voraussetzungen kann das nichtlineare Gleichungssystem mit dem Newton-Verfahren näherungsweise gelöst werden. Ein Startwert kann dafür mit Hilfe der Approximation zum vorhergehenden Zeitschritts definiert werden.

Beispiel 22.3 Beispiele impliziter Runge–Kutta-Verfahren sind das implizite Euler-Verfahren, das Mittelpunktverfahren und das Radau-3-Verfahren, die durch die in Tab. 22.4 gezeigten Butcher-Tableaus definiert sind.

Tab. 22.4 Butcher-Tableaus
des Mittelpunkt- und des
Radau-3-Verfahrens

$\begin{array}{c c} 1 & 1 \\ \hline 1 & \end{array}$	$\begin{array}{c cc} 1/2 & 1/2 \\ \hline 1 & 1 \end{array}$	$\begin{array}{c cc} 1/3 & 5/12 & -1/12 \\ \hline 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$
--	---	--

22.4 Konsistenz

Da Runge–Kutta-Verfahren auf Quadraturformeln basieren, ist die Exaktheit der zugrundeliegenden Quadraturformel maßgebend für die Konsistenz des Verfahrens. Die Quadraturformel $(\alpha_\ell, \gamma_\ell)_{\ell=1,\dots,m}$ heißt *exakt vom Grad p*, falls

$$\int_0^1 q(s) \, ds = \sum_{\ell=1}^m \gamma_\ell q(\alpha_\ell)$$

für alle Polynome q bis zum Grad p gilt. In diesem Fall gilt für jede Funktion $\phi \in C^{p+1}([a, b])$, dass

$$\int_a^b \phi(s) \, ds = (b-a) \sum_{\ell=1}^m \gamma_\ell \phi(a + (b-a)\alpha_\ell) + \mathcal{O}((b-a)^{p+2}).$$

Im Sinne dieser Aussage ist die triviale Quadraturformel, die jedes Integral durch den Wert 0 approximiert, exakt vom Grad $p = -1$. Die Exaktheit vom Grad $p-1$ ist notwendig für die Konsistenz der Ordnung p .

Lemma 22.1 *Das durch die Koeffizienten $\alpha_\ell, \beta_{\ell j}, \gamma_\ell$, $j, \ell = 1, 2, \dots, m$, definierte Runge–Kutta-Verfahren sei konsistent von der Ordnung $p \geq 0$. Dann ist die durch $(\alpha_\ell, \gamma_\ell)_{\ell=1,\dots,m}$ definierte Quadraturformel exakt vom Grad $p-1$.*

Beweis Für $0 \leq n \leq p-1$ sei $y : [0, 1] \rightarrow \mathbb{R}$ die Lösung der Differenzialgleichung $y'(t) = f(t, y(t))$, $y(0) = 0$, mit $f(t, z) = t^n$. Offensichtlich gilt $y(t) = t^{n+1}/(n+1)$. Die Konsistenz der Ordnung p des Runge–Kutta-Verfahrens impliziert, dass für alle $\tau > 0$ die Abschätzung

$$|\tilde{C}(0, 0, \tau)| = \left| \frac{y(\tau) - y(0)}{\tau} - \Phi(0, 0, \tau) \right| \leq c \tau^p$$

gilt beziehungsweise

$$\left| \frac{1}{\tau} \frac{\tau^{n+1}}{n+1} - \sum_{\ell=1}^m \gamma_\ell (\tau \alpha_\ell)^n \right| \leq C \tau^p.$$

Eine Division dieser Ungleichung durch τ^n und der Grenzübergang $\tau \rightarrow 0$ implizieren die Exaktheit vom Grad $p - 1$ der Quadraturformel. \square

Die Umkehrung der Aussage gilt unter zusätzlichen Bedingungen an die Koeffizienten $\beta_{\ell j}$ und führt zu einer hinreichenden Bedingung für eine Konsistenzaussage, die der Darstellung in [6] folgt.

Satz 22.2 Seien $\alpha_\ell, \beta_{\ell j}, \gamma_\ell, j, \ell = 1, 2, \dots, m$, sodass die durch $(\alpha_\ell, \gamma_\ell)_{\ell=1,\dots,m}$ auf dem Intervall $[0, 1]$ definierte Quadraturformel exakt ist vom Grad $p - 1$ und dass

$$\int_0^{\alpha_\ell} q(s) ds = \sum_{j=1}^m \beta_{\ell j} q(\alpha_j)$$

für alle Polynome q bis zum Grad $p - 2$ gilt, das heißt durch $(\alpha_j, \beta_{\ell j})_{j=1,\dots,m}$ wird eine Quadraturformel auf dem Intervall $[0, \alpha_\ell]$ definiert, die exakt ist vom Grad $p - 2$. Dann besitzt das durch $\alpha_\ell, \beta_{\ell j}, \gamma_\ell$ definierte Runge–Kutta-Verfahren die Konsistenzordnung p .

Beweis Aufgrund der Voraussetzungen gilt

$$\begin{aligned} y(t + \tau) - y(t) &= \int_t^{t+\tau} f(s, y(s)) ds = \tau \int_0^1 f(t + \tau \tilde{s}, y(t + \tau \tilde{s})) d\tilde{s} \\ &= \tau \sum_{\ell=1}^m \gamma_\ell f(t + \tau \alpha_\ell, y(t + \tau \alpha_\ell)) + \mathcal{O}(\tau^{p+1}) \end{aligned}$$

sowie für $\ell = 1, 2, \dots, m$

$$\begin{aligned} y(t + \tau \alpha_\ell) - y(t) &= \int_t^{t+\tau \alpha_\ell} f(s, y(s)) ds = \tau \int_0^{\alpha_\ell} f(t + \tau \tilde{s}, y(t + \tau \tilde{s})) d\tilde{s} \\ &= \tau \sum_{j=1}^m \beta_{\ell j} f(t + \tau \alpha_j, y(t + \tau \alpha_j)) + \mathcal{O}(\tau^p). \end{aligned}$$

Für den Konsistenzterm folgt mit der Abkürzung $t = t_k$ und

$$\eta_\ell = f\left(t + \tau \alpha_\ell, y(t) + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j\right)$$

sowie $\Phi(t, y(t), \tau) = \sum_{\ell=1}^m \gamma_\ell \eta_\ell$, dass

$$\begin{aligned} |\tilde{C}(t, y(t), \tau)| &= \left| \frac{y(t + \tau) - y(t)}{\tau} - \sum_{\ell=1}^m \gamma_\ell f \left(t + \tau \alpha_\ell, y(t) + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right) \right| \\ &= \left| \sum_{\ell=1}^m \gamma_\ell \left[f(t + \tau \alpha_\ell, y(t + \tau \alpha_\ell)) - f \left(t + \tau \alpha_\ell, y(t) + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right) \right] \right| + \mathcal{O}(\tau^p) \\ &\leq L \sum_{\ell=1}^m |\gamma_\ell| \left| y(t + \tau \alpha_\ell) - y(t) - \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right| + \mathcal{O}(\tau^p) = L \sum_{\ell=1}^m |\gamma_\ell| r_\ell + \mathcal{O}(\tau^p). \end{aligned}$$

Dabei gilt

$$\begin{aligned} r_\ell &= \left| y(t + \tau \alpha_\ell) - y(t) - \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right| \\ &= \left| \tau \sum_{j=1}^m \beta_{\ell j} [f(t + \tau \alpha_j, y(t + \tau \alpha_j)) - \eta_j] \right| + \mathcal{O}(\tau^p) \\ &= \left| \tau \sum_{j=1}^m \beta_{\ell j} \left[f(t + \tau \alpha_j, y(t + \tau \alpha_j)) - f \left(t + \tau \alpha_j, y(t) + \tau \sum_{n=1}^m \beta_{jn} \eta_n \right) \right] \right| \\ &\quad + \mathcal{O}(\tau^p) \leq \tau L \sum_{j=1}^m |\beta_{\ell j}| \left| y(t + \tau \alpha_j) - y(t) + \tau \sum_{n=1}^m \beta_{jn} \eta_n \right| + \mathcal{O}(\tau^p) \\ &= \tau L \sum_{j=1}^m |\beta_{\ell j}| r_j + \mathcal{O}(\tau^p). \end{aligned}$$

Damit folgt

$$\|r\|_\infty \leq \tau L \|\beta\|_\infty \|r\|_\infty + c \tau^p$$

beziehungsweise $\|r\|_\infty \leq c(1 - \tau L \|\beta\|_\infty)^{-1} \tau^p \leq 2c \tau^p$, sofern $\tau L \|\beta\|_\infty \leq 1/2$ gilt.
Insgesamt ergibt sich

$$|\tilde{C}(t, y(t), \tau)| \leq c \tau^p$$

und dies beweist die Behauptung. □

Bemerkung 22.5 Alternativ lässt sich mit Taylor-Approximationen die Konsistenzordnung eines Runge–Kutta–Verfahrens untersuchen. Dazu verwendet man beispielsweise

mit der Abkürzung y für $y(t)$, dass

$$\begin{aligned}\frac{y(t + \tau) - y(t)}{\tau} &= y'(t) + \frac{\tau}{2} y''(t) + \mathcal{O}(\tau^2) \\ &= f(t, y) + \frac{\tau}{2} [\partial_t f(t, y) + \partial_y f(t, y) f(t, y)] + \mathcal{O}(\tau^2),\end{aligned}$$

wobei $y' = f(t, y)$ und die daraus durch Differenzieren folgende Identität $y'' = \partial_t f(t, y) + \partial_y f(t, y) y'$ ausgenutzt wurden. Für die Inkrementfunktion $\Phi(t, y(t), \tau) = \sum_{\ell=1}^m \gamma_\ell \eta_\ell$ implizieren die Taylor-Approximationen

$$\begin{aligned}\eta_\ell &= f \left(t + \tau \alpha_\ell, y + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right) \\ &= f(t, y) + \partial_t f(t, y) \tau \alpha_\ell + \partial_y f(t, y) \left(\tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right) + \mathcal{O}(\tau^2)\end{aligned}$$

sowie $\eta_j = f(t, y) + \mathcal{O}(\tau)$, dass

$$\begin{aligned}\Phi(t, y, \tau) &= \sum_{\ell=1}^m \gamma_\ell \left[f(t, y) + \partial_t f(t, y) \tau \alpha_\ell + \partial_y f(t, y) \tau \sum_{j=1}^m \beta_{\ell j} \eta_j \right] + \mathcal{O}(\tau^2) \\ &= \sum_{\ell=1}^m \gamma_\ell \left[f(t, y) + \partial_t f(t, y) \tau \alpha_\ell + \partial_y f(t, y) \tau \sum_{j=1}^m \beta_{\ell j} f(t, y) \right] + \mathcal{O}(\tau^2).\end{aligned}$$

Ein Vergleich der Koeffizienten in der resultierenden Identität für

$$\frac{y(t + \tau) - y(t)}{\tau} - \Phi(t, y(t), \tau)$$

impliziert die hinreichenden Bedingungen

$$\sum_{\ell=1}^m \gamma_\ell = 1, \quad \sum_{\ell=1}^m \gamma_\ell \alpha_\ell = \frac{1}{2}, \quad \sum_{\ell=1}^m \sum_{j=1}^m \gamma_\ell \beta_{\ell j} = \frac{1}{2}$$

für die Konsistenz zweiter Ordnung. Die letzte Bedingung kann ersetzt werden durch $\alpha_\ell = \sum_{j=1}^m \beta_{\ell j}$.

Beispiele 22.4 Das explizite Euler-Verfahren besitzt die Konsistenzordnung $p = 1$, das Mittelpunktverfahren, das Euler–Collatz-Verfahren und das Verfahren von Heun die Ordnung $p = 2$, das Radau-3-Verfahren die Konsistenzordnung $p = 3$ und das klassische Runge–Kutta-Verfahren sowie die 3/8-Regel die Ordnung $p = 4$.

Bemerkungen 22.6 (i) Explizite m -stufige Runge–Kutta-Verfahren besitzen höchstens die Konsistenzordnung $p = m$.

(ii) Durch Verwendung Gaußscher Quadraturformeln, die bei m Quadraturpunkten den Exaktheitsgrad $2m - 1$ besitzen, lassen sich implizite Runge–Kutta-Verfahren mit Konsistenzordnung $p = 2m$ konstruieren.

22.5 Lernziele, Quiz und Anwendung

Ihnen sollte die Vorgehensweise zur Motivation von Runge–Kutta-Verfahren bekannt sein und Sie sollten Butcher-Tableaus erstellen können. Hinreichende Kriterien für die Konsistenz eines Runge–Kutta-Verfahrens sollten Sie beschreiben können.

Quiz 22.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Das implizite Euler-Verfahren ist kein Runge–Kutta-Verfahren	
Ein m -stufiges Runge–Kutta-Verfahren hat mindestens die Konsistenzordnung 1	
Für jedes explizite Runge–Kutta-Verfahren der Konsistenzordnung $p = 2$ gilt $\alpha_1 = 0$	
Die Bedingung $\sum_{\ell=1}^m \gamma_\ell = 1$ ist notwendig für die Konsistenz positiver Ordnung eines Runge–Kutta-Verfahrens	
Jedes Einschrittverfahren der Konsistenzordnung p definiert eine Quadraturformel vom Exaktheitsgrad p	

Anwendung 22.1 Zwischen Partikeln wie Atomen oder Molekülen wirken sowohl anziehende als auch abstoßende Kräfte. Dabei dominieren die anziehenden für große und die abstoßenden Kräfte für kleine Distanzen. Dies wird häufig mit einem sogenannten *Lennard–Jones-Potenzial* $V(r) = -c_1 r^{-2} + c_2 r^{-4}$ beschrieben, das die wirkende Kraft durch gewisse negative Gradienten definiert, wobei ausgenutzt wird, dass die Ableitung $V'(r) = 2c_1 r^{-3} - 4c_2 r^{-5}$ negativ ist für $r^2 < r_0^2 = 2c_2/c_1$ und positiv für $r > r_0$. Mit dem Newtonschen Trägheitsgesetz können die Bahnlinien von N interagierenden Partikeln mit Einheitsmasse durch das System von Differenzialgleichungen

$$y_i'' = - \sum_{j=1, \dots, N, j \neq i} \nabla_{y_j} V(\|y_i - y_j\|) = - \sum_{j=1, \dots, N, j \neq i} V'(\|y_i - y_j\|) \frac{y_i - y_j}{\|y_i - y_j\|}$$

für $i = 1, 2, \dots, N$ mit geeigneten Anfangsdaten beschrieben werden. Auf diese Weise lassen sich Systeme von Partikeln wie beispielsweise Wassertropfen simulieren, was

jedoch auf äußerst große Systeme von Differentialgleichungen führt. Verwenden Sie verschiedene MATLAB-Routinen, um Systeme von 10–40 Partikeln, die auf einem Gitter mit Gitterweite $d = 1$ verteilt sind und keine Anfangsgeschwindigkeiten besitzen, im Zeitintervall $[0, T]$ mit $T = 100$ mit den Parametern $c_1 = 10$ und $c_2 = 2$ zu simulieren.

23.1 Allgemeine Mehrschrittverfahren

Mehrschrittverfahren basieren wie Runge–Kutta-Verfahren meist auf Quadraturformeln, jedoch vermeiden sie die Funktionsauswertungen an den Zwischenschritten und verwenden stattdessen die in den vorhergehenden Zeitschritten berechneten Werte. Ausgangspunkt ist die Integraldarstellung einer Differenzialgleichung $y' = f(t, y)$ auf dem Intervall $[t_{k+m-1}, t_{k+m}]$, das heißt

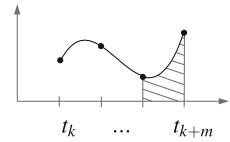
$$y(t_{k+m}) = y(t_{k+m-1}) + \int_{t_{k+m-1}}^{t_{k+m}} f(s, y(s)) \, ds.$$

Das Integral auf der rechten Seite wird mit Hilfe der Funktionswerte an den Zeitschritten $t_k, t_{k+1}, \dots, t_{k+m}$ approximiert, das heißt

$$\int_{t_{k+m-1}}^{t_{k+m}} f(s, y(s)) \, ds \approx \tau \sum_{\ell=0}^m \beta_\ell f(t_{k+\ell}, y(t_{k+\ell})).$$

Dies lässt sich als verallgemeinerte Quadraturformel interpretieren, bei der eine Funktion auf dem Intervall $[t_k, t_{k+m}]$ interpoliert und der Interpolant dann auf dem Teilintervall $[t_{k+m-1}, t_{k+m}]$ integriert wird, s. Abb. 23.1. Die Funktionswerte an den Zeitschritten müssen nur einmal bestimmt und können in späteren Zeitschritten wiederverwendet werden. Die obige Integration der Differenzialgleichung kann allgemeiner auch über ein Intervall der Form $[t_{k+m-n}, t_{k+m}]$ mit $1 \leq n \leq m$ erfolgen.

Abb. 23.1 Mehrschrittverfahren lassen sich als Anwendung einer verallgemeinerten Quadraturformel interpretieren



Definition 23.1 Ein m -Mehrschrittverfahren ist ein Verfahren der Form

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = \tau \Phi(t_k, y_k, y_{k+1}, \dots, y_{k+m}, \tau)$$

mit reellen Koeffizienten $(\alpha_\ell)_{\ell=0,\dots,m}$, wobei $\alpha_m = 1$ gelte. Das Verfahren heißt *explizit*, falls Φ nicht von y_{k+m} abhängt und *implizit* andernfalls. Es heißt *linear*, falls Koeffizienten $(\beta_\ell)_{\ell=0,\dots,m}$ existieren, sodass

$$\Phi(t_k, y_k, \dots, y_{k+m}, \tau) = \sum_{\ell=0}^m \beta_\ell f(t_{k+\ell}, y_{k+\ell}).$$

Bemerkungen 23.1 (i) Ein lineares Mehrschrittverfahren ist explizit genau dann, wenn $\beta_m = 0$ gilt.

(ii) Um einen Schritt eines Mehrschrittverfahrens durchführen zu können, müssen die Approximationen y_k, \dots, y_{k+m-1} bereits vorliegen. Zum Start des Verfahrens können Approximationen y_1, \dots, y_{m-1} mit Einschrittverfahren bestimmt werden.

Beispiele 23.1 (i) Das *leapfrog-Verfahren* ist definiert durch

$$y_{k+2} = y_k + 2\tau f(t_{k+1}, y_{k+1}).$$

(ii) Die *Adams–Bashforth-* und *Adams–Moulton-Verfahren* sind explizite beziehungsweise implizite lineare Mehrschrittverfahren der Form

$$y_{k+m} = y_{k+m-1} + \tau \sum_{\ell=0}^m \beta_\ell f(t_{k+\ell}, y_{k+\ell}),$$

(iii) Sogenannte *Backward-Differentiation-Formulas* oder *BDF-Verfahren* verwenden das Lagrange-Interpolationspolynom $q \in \mathcal{P}_m$ mit $q(t_{k+i}) = y_{k+i}$, $i = 0, 1, \dots, m$, und bestimmen y_{k+m} als Lösung der Gleichung

$$q'(t_{k+m}) = f(t_{k+m}, y_{k+m}).$$

23.2 Konsistenz

Zur Bestimmung der Genauigkeit eines Mehrschrittverfahrens wird eine lokal exakte Lösung in das numerische Schema eingesetzt.

Definition 23.2 Für $z_k \in \mathbb{R}$ sei $z : [t_k, t_{k+m}] \rightarrow \mathbb{R}$ die Lösung des Anfangswertproblems $z' = f(t, z)$ in $(t_k, t_{k+m}]$ mit $z(t_k) = z_k$. Der *lokale Konsistenzfehler* eines Mehrschrittverfahrens ist definiert durch

$$\tilde{C}(t_k, z(t_k), \tau) = \frac{1}{\tau} \sum_{\ell=0}^m \alpha_\ell z(t_{k+\ell}) - \Phi(t_k, z(t_k), \dots, z(t_{k+m}), \tau).$$

Ein Mehrschrittverfahren heißt *konsistent* von der Ordnung p , falls für alle $f \in C^p([0, T] \times \mathbb{R})$, $k = 0, 1, \dots, K-m$ und $z \in C^{p+1}([t_k, t_{k+m}])$ gilt

$$\tilde{C}(t_k, z(t_k), \tau) = \mathcal{O}(\tau^p).$$

Für lineare Mehrschrittverfahren ergeben sich einfache Kriterien für die Konsistenz der Ordnung p , wie folgendes Resultat, welches der Darstellung in [7] folgt, zeigt.

Satz 23.1 *Das lineare m -Mehrschrittverfahren*

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = \tau \sum_{\ell=0}^m \beta_\ell f(t_{k+\ell}, y_{k+\ell})$$

ist konsistent von der Ordnung $p \geq 1$ genau dann, wenn gilt

$$\sum_{\ell=0}^m \alpha_\ell = 0, \quad \sum_{\ell=0}^m (\alpha_\ell \ell^q - \beta_\ell q \ell^{q-1}) = 0, \quad q = 1, 2, \dots, p.$$

Beweis Die Taylor-Approximationen

$$\begin{aligned} z(t_{k+\ell}) &= z(t_k + \ell\tau) = \sum_{q=0}^p \frac{(\ell\tau)^q}{q!} z^{(q)}(t_k) + \mathcal{O}(\tau^{p+1}), \\ z'(t_{k+\ell}) &= z'(t_k + \ell\tau) = \sum_{q=1}^p \frac{(\ell\tau)^{q-1}}{(q-1)!} z^{(q)}(t_k) + \mathcal{O}(\tau^p) \end{aligned}$$

sowie $f(t_k, z(t_{k+\ell})) = z'(t_{k+\ell})$ zeigen

$$\begin{aligned}
 \tilde{C}(t_k, z(t_k), \tau) &= \sum_{\ell=0}^m \left[\frac{\alpha_\ell}{\tau} z(t_{k+m}) - \beta_\ell z'(t_{k+\ell}) \right] \\
 &= \sum_{\ell=0}^m \left[\frac{\alpha_\ell}{\tau} \sum_{q=0}^p \frac{(\ell\tau)^q}{q!} z^{(q)}(t_k) - \beta_\ell \sum_{q=1}^p \frac{(\ell\tau)^{q-1}}{(q-1)!} z^{(q)}(t_k) \right] + \mathcal{O}(\tau^p) \\
 &= \frac{1}{\tau} \sum_{\ell=0}^m \alpha_\ell z(t_k) + \sum_{\ell=0}^m \sum_{q=1}^p \left[\frac{\alpha_\ell}{\tau} \frac{(\ell\tau)^q}{q!} - \beta_\ell \frac{(\ell\tau)^{q-1}}{(q-1)!} \right] z^{(q)}(t_k) + \mathcal{O}(\tau^p) \\
 &= \frac{1}{\tau} \sum_{\ell=0}^m \alpha_\ell z(t_k) + \sum_{q=1}^p \sum_{\ell=0}^m \frac{\tau^{q-1}}{q!} [\alpha_\ell \ell^q - \beta_\ell q \ell^{q-1}] z^{(q)}(t_k) + \mathcal{O}(\tau^p).
 \end{aligned}$$

Unter den angegebenen Bedingungen verschwinden die beiden Summen. Die Umkehrung der Aussage folgt durch Betrachtung der Funktionen $z(t) = t^n$, $z(0) = 0$, $n = 1, 2, \dots, p$, sowie $z(t) = 1$ als Lösungen geeigneter Differenzialgleichungen. \square

23.3 Adams-Verfahren

Die Adams-Verfahren resultieren aus Diskretisierungen der lokalen Integralgleichung

$$y(t_{k+m}) = y(t_{k+m-1}) + \int_{t_{k+m-1}}^{t_{k+m}} f(s, y(s)) \, ds.$$

Für diese Verfahren gilt $\alpha_m = 1$, $\alpha_{m-1} = -1$ und $\alpha_\ell = 0$ für $\ell = 0, 1, \dots, m-2$. Bei Adams–Bashforth-Verfahren wird das Integral mit Newton–Cotes-Formeln zu den Stützstellen $t_k, t_{k+1}, \dots, t_{k+m-1}$ approximiert, das heißt

$$\int_{t_{k+m-1}}^{t_{k+m}} f(s, y(s)) \, ds \approx \tau \sum_{\ell=0}^{m-1} \beta_\ell f(t_{k+\ell}, y_{k+\ell}),$$

wobei sich die Koeffizienten $\beta_0, \beta_1, \dots, \beta_{m-1}$ durch Lagrange-Interpolation der Stützpaare $(t_{k+\ell}, w_{k+\ell})$, $\ell = 0, 1, \dots, m-1$, mit $w_{k+\ell} = f(t_{k+\ell}, y_{k+\ell})$ und anschließender Integration des Interpolationspolynoms $p_{m-1} \in \mathcal{P}_{m-1}$ auf dem Intervall $[t_{k+m-1}, t_{k+m}]$ ergeben.

Tab. 23.1 Koeffizienten der Adams–Moulton-Verfahren für $m = 1, \dots, 4$

m	β_0	β_1	β_2	β_3
1	1			
2	$-1/2$	$3/2$		
3	$5/12$	$-16/12$	$23/12$	
4	$-9/24$	$37/24$	$-59/24$	$55/24$

Beispiele 23.2 (i) Für $m = 1$ ist das Lagrange-Interpolationspolynom die konstante Funktion $p_0(s) = w_k = f(t_k, y_k)$ und es folgt

$$\int_{t_k}^{t_{k+1}} f(s, y(s)) ds \approx \tau f(t_k, y_k)$$

das heißt es gilt $\beta_0 = 1$ und $\beta_1 = 0$.

(ii) Für $m = 2$ ist das Lagrange-Interpolationspolynom $p_1 \in \mathcal{P}_1$ bezüglich der Stützpaare (t_k, w_k) und (t_{k+1}, w_{k+1}) gegeben durch

$$p_1(s) = w_k \frac{t_{k+1} - s}{\tau} + w_{k+1} \frac{s - t_k}{\tau}$$

und es folgt

$$\int_{t_{k+1}}^{t_{k+2}} f(s, y(s)) ds \approx \int_{t_{k+1}}^{t_{k+2}} p_1(s) ds = -\frac{\tau}{2} w_k + \frac{3\tau}{2} w_{k+1},$$

das heißt es gilt $\beta_0 = -1/2$, $\beta_1 = 3/2$ und $\beta_2 = 0$.

Allgemeiner ergeben sich die in Tab. 23.1 gezeigten Koeffizienten.

Bemerkungen 23.2 (i) Das Adams–Bashforth-Verfahren mit m Schritten besitzt die Konsistenzordnung m .

(ii) In jedem Schritt des Adams–Bashforth-Verfahrens ist nur eine neue Funktionsauswertung erforderlich.

Das Adams–Moulton-Verfahren mit m Schritten verwendet zusätzlich das Stützpaar (t_{k+m}, w_{k+m}) mit $w_{k+m} = f(t_{k+m}, y_{k+m})$ zur Definition des Interpolationspolynoms $p_m \in \mathcal{P}_m$ und es ergibt sich die Approximation

$$\int_{t_{k+m-1}}^{t_{k+m}} f(s, y(s)) ds \approx \tau \sum_{\ell=0}^m \beta_\ell f(t_{k+\ell}, y_{k+\ell}).$$

Tab. 23.2 Koeffizienten der Adams–Bashforth–Verfahren für $m = 1, \dots, 4$

m	β_0	β_1	β_2	β_3	β_4
1	1/2	1/2			
2	-1/12	8/12	5/12		
3	1/24	-5/24	19/24	9/24	
4	-19/720	106/720	-264/720	646/720	251/720

Die Anwendung des oben beschriebenen Vorgehens führt auf die in Tabelle 23.2 aufgeführten Koeffizienten.

Bemerkung 23.3 Das Adams–Moulton–Verfahren mit m Schritten besitzt die Konsistenzordnung $m + 1$. Es ist wohldefiniert, falls $\tau \|\beta\|_1 L < 1$ mit der uniformen Lipschitz-Konstanten L bezüglich des zweiten Arguments von f gilt.

23.4 Prädiktor-Korrektor-Verfahren

Das Adams–Moulton– und Adams–Bashforth–Verfahren lassen sich zu einem expliziten Verfahren kombinieren, das die höhere Konsistenzordnung des Adams–Moulton–Verfahrens erhält. Die Idee besteht in der Durchführung eines Schritts einer Fixpunktiteration, dem *Korrektor-Schritt*, für das Adams–Moulton–Verfahren, wobei der Startwert mit dem Adams–Bashforth–Verfahren bestimmt wird, das heißt aus einem *Prädiktor-Schritt* resultiert.

Algorithmus 23.1 (Adams–Bashforth–Moulton–Verfahren) Seien $y_0 \in \mathbb{R}$, $f \in C^0([0, T] \times \mathbb{R})$ und $\tau > 0$ sowie $m \in \mathbb{N}$. Ferner seien Startwerte $y_1, y_2, \dots, y_{m-1} \in \mathbb{R}$ gegeben. Setze $k = 0$ und $K = \lfloor T/\tau \rfloor$.

(1) Bestimme den Hilfswert $\tilde{y}_{k+m} \in \mathbb{R}$ mit dem Adams–Bashforth–Verfahren, das heißt berechne

$$\tilde{y}_{k+m} = y_{k+m-1} + \tau \sum_{\ell=0}^{m-1} \beta_\ell^{AB} f(t_{k+\ell}, y_{k+\ell}).$$

(2) Führe einen Schritt einer Fixpunktiteration des Adams–Moulton–Verfahrens mit Startwert \tilde{y}_{k+m} durch, das heißt berechne

$$y_{k+m} = y_{k+m-1} + \tau \sum_{\ell=0}^{m-1} \beta_\ell^{AM} f(t_{k+\ell}, y_{k+\ell}) + \tau \beta_m^{AM} f(t_{k+m}, \tilde{y}_{k+m}).$$

(3) Stoppe falls $k + m > K$; andernfalls erhöhe $k \rightarrow k + 1$ und wiederhole Schritt (1).

Ein Iterationsschritt des Adams–Bashforth–Moulton-Verfahrens lässt sich unter Vernachlässigung der Argumente t_k und τ schreiben als

$$y_{k+m} = y_{k+m-1} + \tau \Phi^{AM}(y_k, \dots, y_{k+m-1}, y_{k+m-1} + \tau \Phi^{AB}(y_k, \dots, y_{k+m-1})),$$

wodurch eine explizite Inkrementfunktion Φ^{ABM} definiert wird, die nicht linear ist.

Satz 23.2 *Die Funktion f sei uniform Lipschitz-stetig im zweiten Argument. Dann besitzt das m -stufige Adams–Bashforth–Moulton-Verfahren die Konsistenzordnung $m + 1$.*

Beweis Für eine lokale Lösung $z : [t_k, t_{k+m}] \rightarrow \mathbb{R}$ gilt aufgrund der Konsistenzordnung $m + 1$ des Adams–Moulton-Verfahrens

$$\begin{aligned} \tau \tilde{C}(t_k, z(t_k), \tau) &= z(t_{k+m}) - z(t_k) \\ &\quad - \tau \Phi^{AM}(z(t_k), \dots, z(t_{k+m-1}), z(t_{k+m-1}) + \tau \Phi^{AB}(z(t_k), \dots, z(t_{k+m-1}))) \\ &= z(t_{k+m}) - z(t_k) - \tau \Phi^{AM}(z(t_k), \dots, z(t_{k+m-1}), z(t_{k+m})) \\ &\quad + \tau \left[\Phi^{AM}(z(t_k), \dots, z(t_{k+m-1}), z(t_{k+m})) \right. \\ &\quad \left. - \Phi^{AM}(z(t_k), \dots, z(t_{k+m-1}) + \tau \Phi^{AB}(z(t_k), \dots, z(t_{k+m-1}))) \right] \\ &= z(t_{k+m}) - z(t_k) - \tau \Phi^{AM}(z(t_k), \dots, z(t_{k+m})) + \tau[\dots] \\ &= \mathcal{O}(\tau^{m+1}) + \tau[\dots]. \end{aligned}$$

Die uniforme Lipschitz-Stetigkeit von f beziehungsweise der Inkrementfunktion Φ^{AM} bezüglich des letzten Arguments zeigt

$$\tau|[\dots]| \leq \tau L |z(t_{k+m}) - z(t_{k+m-1}) + \tau \Phi^{AB}(t_k, z(t_k), \dots, z(t_{k+m-1}))|.$$

Aufgrund der Konsistenz der Ordnung m des Adams–Bashforth-Verfahrens gilt $\tau|[\dots]| = \mathcal{O}(\tau^{m+1})$ und damit folgt die behauptete Konsistenzordnung. \square

Bemerkung 23.4 Allgemeiner kann ein explizites Verfahren der Konsistenzordnung p_{expl} zur Bestimmung eines Startwerts verwendet und anschließend v Wiederholungen einer Fixpunktiteration mit einem impliziten Verfahren der Konsistenzordnung p_{impl} durchgeführt werden. Das resultierende Prädiktor-Korrektor-Verfahren besitzt die Konsistenzordnung $p = \min\{p_{\text{expl}} + v, p_{\text{impl}}\}$, siehe beispielsweise [7].

23.5 Lernziele, Quiz und Anwendung

Sie sollten Mehrschrittverfahren herleiten, deren Vor- und Nachteile im Vergleich zu Einschrittverfahren deutlich machen und einige Beispiele angeben können. Ein Kriterium für die Bestimmung der Konsistenzordnung eines Mehrschrittverfahrens sollten Sie herleiten können. Die Ideen der Kombination von expliziten und impliziten Mehrschrittverfahren zu Prädiktor-Korrektor-Verfahren sollten Sie erläutern können.

Quiz 23.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Adams-Moulton-Verfahren für $m \geq 1$ definieren explizite Mehrschrittverfahren, die linear sind	
Adams-Basforth-Verfahren für $m \geq 1$ definieren lineare, implizite Mehrschrittverfahren mit Konsistenzordnung $m + 1$	
Notwendig für die Konsistenz $p \geq 1$ eines Mehrschrittverfahrens ist die Bedingung $\sum_{\ell=0}^m \beta_\ell = 1$	
Das explizite Euler-Verfahren ist ein Mehrschrittverfahren mit zwei Schritten	
Prädiktor-Korrektor-Verfahren lassen sich als implizite Mehrschrittverfahren interpretieren	

Anwendung 23.1 Wechselwirkende Schwingungen führen häufig zu unerwünschten Effekten wie dem Überschwappen einer Flüssigkeit, die in einem Gefäß transportiert wird, oder dem starken Rütteln einer Waschmaschine, die bestimmte Drehzahlen durchläuft. Die mathematische Beschreibung kann in diesen Fällen zu Differenzialgleichungen führen, bei denen kleine Änderungen der Daten große Auswirkungen auf die Lösungen haben können, was auch als chaotisches Verhalten bezeichnet wird. Numerisch sind die Prozesse daher in der Regel nur für kurze Zeiten sinnvoll approximierbar. Ein Beispiel ist das Doppelpendel, bei dem an dem Arm eines Pendels ein weiteres Pendel angebracht ist, s. Abb. 23.2. Bezeichnen ϕ_1 und ϕ_2 die Auslenkungswinkel bezüglich der jeweiligen Ruhelage, so werden die Pendelbewegungen im Fall gleicher Massen und Pendellängen und

Abb. 23.2 Das Doppelpendel besteht aus zwei kombinierten Pendeln



bei einer geeigneten Skalierung der Erdbeschleunigung durch das System von Differenzialgleichungen

$$\begin{aligned}2\phi_1'' + \phi_2'' \cos(\phi_1 - \phi_2) + \phi_2' \sin(\phi_1 - \phi_2) + 2 \sin(\phi_1) &= 0, \\ \phi_2'' + \phi_1'' \cos(\phi_1 - \phi_2) - \phi_1' \sin(\phi_1 - \phi_2) + \sin(\phi_2) &= 0\end{aligned}$$

beschrieben. Simulieren Sie das System mit den Anfangsdaten $\phi_1(0) = \pi/2$, $\phi_2(0) = 0$, $\phi_1'(0) = 0$, $\phi_2'(0) = 0$ im Zeitintervall $[0, T]$ mit $T = 100$. Stören Sie die Anfangsdaten und verwenden Sie unterschiedliche MATLAB-Routinen zur numerischen Lösung. Visualisieren Sie Ihre Ergebnisse.

24.1 Differenzengleichungen

Für Einschrittverfahren impliziert die Konsistenz eines Verfahrens bereits dessen Konvergenz. Dies ist bei Mehrschrittverfahren im Allgemeinen falsch. Bei diesen definiert die Gleichung

$$\frac{1}{\tau} \sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = 0$$

Approximationen des trivialen Problems $y'(t) = 0$. Das folgende Beispiel zeigt, dass selbst eine hohe Konsistenzordnung nicht notwendigerweise zu sinnvollen Approximationen führt. Wir folgen in diesem Kapitel den Darstellungen in [1,6,7].

Beispiel 24.1 Das durch $m = 2$ und $\alpha_2 = 1$, $\alpha_1 = 4$, $\alpha_0 = -5$ und $\beta_2 = 0$, $\beta_1 = 4$, $\beta_0 = 2$ definierte Mehrschrittverfahren

$$y_{k+2} + 4y_{k+1} - 5y_k = \tau(4f(t_{k+1}, y_{k+1}) + 2f(t_k, y_k))$$

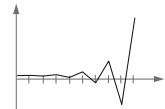
besitzt die Konsistenzordnung $p = 3$. Damit berechnete Approximationslösungen der Differenzialgleichung $y' = 0$ erfüllen $y_{k+2} + 4y_{k+1} - 5y_k = 0$. Lösungen dieser Gleichung sind gegeben durch Linearkombinationen der drei speziellen Lösungen

$$v_k = \lambda_1^k, \quad w_k = \lambda_2^k, \quad z_k = 1,$$

wobei $\lambda_1 = 1$ und $\lambda_2 = -5$ die Nullstellen des Polynoms $q(\lambda) = \lambda^2 + 4\lambda - 5$ sind. Für die Startwerte $y_0 = 1$ und $y_1 = (1 + \delta)$ folgt $\gamma_1 = 1 + \delta/6$ und $\gamma_2 = -\delta/6$ und die Lösung

$$y_k = 1 + \delta/6 - (-5)^k \delta/6$$

Abb. 24.1 Unbeschränkte Lösung der Differenzengleichung
 $y_{k+2} + 4y_{k+1} - 5y_k = 0$



ist unbeschränkt für jedes $\delta \neq 0$, s. Abb. 24.1. Damit wird die exakte Lösung des Problems $y'(t) = 0$, $y(0) = 1$, nicht sinnvoll angenähert. Der Startwert $y_1 = 1 + \delta$ lässt sich als Approximation von $y(t_1)$ eines Einschrittverfahrens interpretieren.

Definition 24.1 Zu gegebenen $\alpha_0, \alpha_1, \dots, \alpha_m \in \mathbb{R}$ mit $\alpha_m = 1$ heißt die Gleichung

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = 0$$

(lineare homogene) Differenzengleichung. Eine Folge $(y_k)_{k \geq 0}$ ist eine Lösung der Differenzengleichung, falls diese für jedes $k \in \mathbb{N}_0$ erfüllt ist.

Bemerkung 24.1 Für jeden Vektor $(y_k)_{k=0, \dots, m-1}$ von Startwerten existiert eine eindeutig bestimmte Lösung einer Differenzengleichung.

Das Verhalten von Lösungen der Differenzengleichung lässt sich mit Hilfe eines Eigenwertproblems analysieren.

Lemma 24.1 Eine Folge $(y_k)_{k \geq 0}$ ist Lösung der durch $(\alpha_\ell)_{\ell=0, \dots, m}$ definierten Differenzengleichung genau dann, wenn für die Vektoren

$$Y_k = [y_k, y_{k+1}, \dots, y_{k+m-1}]^\top$$

die Relation $Y_{k+1} = AY_k$ für $k = 0, 1, \dots$ gilt, wobei die Begleitmatrix $A \in \mathbb{R}^{m \times m}$ definiert ist durch

$$A = \begin{bmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 & \\ -\alpha_0 & -\alpha_1 & \dots & -\alpha_{m-1} & \end{bmatrix}.$$

Besitzt A die linear unabhängigen Eigenvektoren $v_1, v_2, \dots, v_m \in \mathbb{R}^m$ mit zugehörigen Eigenwerten $\lambda_1, \lambda_2, \dots, \lambda_m$ und sind $\gamma_1, \gamma_2, \dots, \gamma_m \in \mathbb{R}$ die Koeffizienten des Vektors Y_0 bezüglich dieser Basis, so folgt

$$Y_k = A^k Y_0 = \sum_{j=1}^m \lambda_j^k \gamma_j v_j.$$

Die Eigenwerte $\lambda_1, \lambda_2, \dots, \lambda_m$ sind genau die Nullstellen des charakteristischen Polynoms $q(\lambda) = \lambda^m + \alpha_{m-1}\lambda^{m-1} + \dots + \lambda\alpha_1 + \alpha_0$.

Beweis Übungsaufgabe. □

24.2 Nullstabilität

Damit ein Mehrschrittverfahren zu sinnvollen Approximationen führt, sollten Lösungen der zugehörigen homogenen Differenzengleichung beschränkt sein. Im Fall der Diagonalsierbarkeit der Begleitmatrix ist dies der Fall, sofern $|\lambda_i| \leq 1$ für $i = 1, 2, \dots, m$ gilt.

Definition 24.2 Eine lineare, homogene Differenzengleichung heißt *nullstabil*, wenn jede Lösung der Differenzengleichung beschränkt ist.

Existieren Eigenwerte, für die die algebraische Vielfachheit höher ist als die geometrische, so treten weitere Lösungen auf.

Beispiel 24.2 Die Begleitmatrix der Differenzengleichung $y_{k+2} - 2y_{k+1} + y_k = 0$ besitzt den zweifachen Eigenwert $\lambda = 1$, aber keine zugehörigen linear unabhängigen Eigenvektoren $v_1, v_2 \in \mathbb{R}^2$. Durch $y_k = k$, $k \geq 0$, wird eine unbeschränkte Lösung definiert.

Bemerkung 24.2 Eine Übungsaufgabe zeigt, dass für Eigenwerte λ der Vielfachheit $s \geq 2$ durch $y_k = k^r \lambda^k$, $k \geq 0$, für jedes $r = 0, 1, \dots, s-1$ eine Lösung der Differenzengleichung definiert wird. Notwendig für die Nullstabilität ist damit, dass Eigenwerte λ mit $|\lambda| = 1$ einfach sind.

Die folgende Definition und der folgende Satz definieren ein hinreichendes Kriterium für die Nullstabilität einer Differenzengleichung, das im Sinne der vorherigen Bemerkung auch notwendig ist.

Definition 24.3 Das Polynom $q \in \mathcal{P}_m$ erfüllt die *Dahlquistsche Wurzelbedingung*, sofern jede Nullstelle $\lambda \in \mathbb{C}$ von q die Abschätzung $|\lambda| \leq 1$ erfüllt und im Fall $|\lambda| = 1$ einfach ist.

Erfüllt das charakteristische Polynom der Begleitmatrix einer Differenzengleichung die Dahlquistsche Wurzelbedingung, so ist die Gleichung nullstabil und folglich ihre Lösungen beschränkt.

Satz 24.1 Das charakteristische Polynom $q(z) = z^m + \alpha_{m-1}z^{m-1} + \cdots + \alpha_1z + \alpha_0$ der Begleitmatrix A erfülle die Dahlquistsche Wurzelbedingung. Dann existiert eine reguläre Matrix $R \in \mathbb{C}^{m \times m}$, sodass mit der durch die Norm $x \mapsto \|Rx\|_\infty$ induzierten Operatornorm $\|B\|_R = \sup_{\|x\|_R=1} \|Bx\|_R$ gilt $\|A\|_R \leq 1$.

Beweis Seien $\lambda_1, \lambda_2, \dots, \lambda_r \in \mathbb{C}$ die komplexen Eigenwerte von A mit Vielfachheiten s_1, s_2, \dots, s_r . Der Hauptsatz über die Jordansche Normalform impliziert die Existenz einer regulären Matrix $T \in \mathbb{C}^{k \times k}$ und von Matrizen $J_i \in \mathbb{C}^{s_i \times s_i}$, $i = 1, 2, \dots, r$, sodass

$$T^{-1}AT = J = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{bmatrix}, \quad J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}.$$

Gilt $|\lambda_i| = 1$, so ist nach Voraussetzung $s_i = 1$. Sofern ein Eigenwert $|\lambda_i| < 1$ existiert, sei

$$\varepsilon = \min \{1 - |\lambda_i| : i = 1, 2, \dots, r, |\lambda_i| < 1\},$$

und andernfalls sei $\varepsilon = 1$. Es sei $D \in \mathbb{R}^{m \times m}$ die Diagonalmatrix mit den Einträgen $d_{jj} = \varepsilon^{j-1}$ für $j = 1, 2, \dots, m$. Damit gilt

$$\tilde{J} = D^{-1}T^{-1}ATD = \begin{bmatrix} \tilde{J}_1 & & & \\ & \tilde{J}_2 & & \\ & & \ddots & \\ & & & \tilde{J}_r \end{bmatrix}, \quad \tilde{J}_i = \begin{bmatrix} \lambda_i & \varepsilon & & \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon \\ & & & \lambda_i \end{bmatrix}.$$

Für die Zeilensummennorm der skalierten Jordan-Blöcke \tilde{J}_i gilt aufgrund der Wahl von ε , dass $\|\tilde{J}_i\|_\infty \leq 1$, und es folgt $\|\tilde{J}\|_\infty \leq 1$. Mit $R = D^{-1}T^{-1}$ folgt für die induzierte Operatornorm mit der Ersetzung $y = D^{-1}T^{-1}x$, dass

$$\begin{aligned} \|A\|_R &= \sup_{\|x\|_R=1} \|Ax\|_R = \sup_{\|D^{-1}T^{-1}x\|_\infty=1} \|D^{-1}T^{-1}Ax\|_\infty \\ &= \sup_{\|y\|_\infty=1} \|D^{-1}T^{-1}ATDy\|_\infty \leq \|\tilde{J}\|_\infty. \end{aligned}$$

Dies beweist die Behauptung. \square

Beispiel 24.3 Für Adams-Verfahren gilt $\alpha_m = 1$, $\alpha_{m-1} = -1$ und $\alpha_\ell = 0$ sonst, sodass das charakteristische Polynom $q(z) = z^m - z^{m-1}$ die $(m-1)$ -fache Nullstelle $\lambda = 0$ sowie die einfache Nullstelle $\lambda = 1$ besitzt. Folglich sind Adams-Verfahren nullstabil. Analog ergibt sich die allgemeine Nullstabilität von Einschrittverfahren.

24.3 Konvergenz

Für ein Mehrschrittverfahren ist die Nullstabilität der zugehörigen Differenzengleichung ein notwendiges Kriterium für die Konvergenz des Verfahrens. Jedes Mehrschrittverfahren

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = \tau \Phi(t_k, y_k, y_{k+1}, \dots, y_{k+m}, \tau)$$

lässt sich mit den Vektoren $Y_k \in \mathbb{R}^m$, $k = 0, 1, \dots, K - m + 1$, und der Funktion $\Psi : [0, T] \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$ definiert durch

$$Y_k = [y_k, y_{k+1}, \dots, y_{k+m-1}]^\top,$$

$$\Psi(t_k, Y_k, Y_{k+1}, \tau) = [0, \dots, 0, \Phi(t_k, y_k, y_{k+1}, \dots, y_{k+m}, \tau)]^\top$$

sowie der Begleitmatrix $A \in \mathbb{R}^{m \times m}$ in der Form

$$Y_{k+1} = AY_k + \tau \Psi(t_k, Y_k, Y_{k+1}, \tau),$$

darstellen. Dies ist die Struktur eines Einschrittverfahrens und eine Fehleranalyse lässt sich ähnlich durchführen. Die Gültigkeit der Dahlquistschen Wurzelbedingung erlaubt es dabei, den Einfluss der Matrix A zu kontrollieren.

Satz 24.2 *Das Mehrschrittverfahren*

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = \tau \Phi(t_k, y_k, y_{k+1}, \dots, y_{k+m}, \tau)$$

sei konsistent von der Ordnung p und das Polynom $q(z) = z^m + \alpha_{m-1}z^{m-1} + \dots + \alpha_1z + \alpha_0$ erfülle die Dahlquistsche Wurzelbedingung. Ferner sei Φ uniform Lipschitz-stetig im zweiten bis vorletzten Argument, das heißt es existiere eine Konstante $L \geq 0$, sodass für alle $t \in [0, T]$, $v, w \in \mathbb{R}^{m+1}$ und $\tau > 0$ gilt

$$|\Phi(t, v_0, \dots, v_m, \tau) - \Phi(t, w_0, \dots, w_m, \tau)| \leq L(|v_0 - w_0| + \dots + |v_m - w_m|).$$

Sind die Startwerte y_0, y_1, \dots, y_{m-1} so gewählt, dass

$$\max_{k=0, \dots, m-1} |y_k - y(t_k)| \leq C_0 \tau^p$$

mit einer von τ unabhängigen Konstanten $C_0 \geq 0$, so existieren Konstanten C_1, C_2, C_3 , sodass

$$\max_{k=0,1,\dots,K} |y_k - y(t_k)| \leq C_1 T \tau^p \exp(C_2 L T)$$

für alle $0 < \tau \leq C_3$ gilt.

Beweis In diesem Beweis steht c für eine Konstante, die sich von Rechenschritt zu Rechenschritt vergrößern kann, aber nicht von τ und K abhängt. Für $k = 0, 1, \dots, K$ sei $e_k = y_k - y(t_k)$. Nach Definition des Konsistenzterms $\tilde{C}_k = \tilde{C}(t_k, y(t_k), \tau)$ gilt

$$\sum_{\ell=0}^m \alpha_\ell e_{k+\ell} = \tau [\Phi(t_k, y_k, \dots, y_{k+m}, \tau) - \Phi(t_k, y(t_k), \dots, y(t_{k+m}), \tau) - \tilde{C}_k]$$

für $k = 0, 1, \dots, K-m$ und es bezeichne r_k die rechte Seite. Die Lipschitz-Stetigkeit von Φ und die Konsistenz des Verfahrens implizieren, dass

$$\tau|r_k| \leq \tau L \sum_{\ell=0}^m |e_{k+\ell}| + c\tau^{p+1}.$$

Mit den Vektoren

$$E_k = [e_k, e_{k+1}, \dots, e_{k+m-1}]^\top, \quad G_k = [0, \dots, 0, r_k]^\top$$

und der Begleitmatrix A folgt

$$E_{k+1} = AE_k + \tau G_k.$$

Es sei $\|\cdot\|_R$ eine Norm auf \mathbb{R}^m , sodass $\|A\|_R \leq 1$ mit der induzierten Operatornorm gilt. Die Äquivalenz von Normen auf dem Vektorraum \mathbb{R}^m zeigt, dass

$$\|\tau G_k\|_R \leq c\tau|r_k| \leq c\tau L(\|E_{k+1}\|_R + \|E_k\|_R) + c\tau^{p+1}.$$

Mit dem Schema für die Vektoren E_k folgt, dass

$$\begin{aligned} \|E_{k+1}\|_R &\leq \|AE_k\|_R + \|\tau G_k\|_R \leq \|A\|_R \|E_k\|_R + c\tau|r_k| \\ &\leq \|E_k\|_R + c\tau L(\|E_k\|_R + \|E_{k+1}\|_R) + c\tau^{p+1} \end{aligned}$$

beziehungsweise

$$(1 - c'\tau L)\|E_{k+1}\|_R \leq (1 + c''\tau L)\|E_k\|_R + c'''\tau^{p+1}.$$

Subtrahieren von $(1 - c'\tau L)\|E_k\|_R$ auf beiden Seiten führt auf

$$\|E_{k+1}\|_R - \|E_k\|_R \leq 2(c'' - c')\tau L\|E_k\|_R + 2c'''\tau^{p+1},$$

wobei $c'\tau L \leq 1/2$ beziehungsweise $1 - c'\tau L \geq 1/2$ vorausgesetzt wurde. Mit $c_1 = c'$, $c_2 = \max\{0, 2(c'' - c')\}$ und $c_3 = 2c''''$ und einer Summation dieser Gleichung über $k = 0, 1, \dots, K'$ mit $K' \leq K-m$ ergibt sich

$$\|E_{K'+1}\|_R \leq \|E_0\|_R + c_2\tau L \sum_{k=0}^{K'} \|E_k\|_R + c_3\tau^{p+1} K'.$$

Das diskrete Gronwall-Lemma und $K'\tau \leq T$ zeigen, dass

$$\max_{k=0,1,\dots,K-m+1} \|E_k\|_R \leq c_3 T \tau^p \exp(c_2 L T).$$

Da $|e_{k+\ell}| \leq c \|E_k\|_R$ für $k = 0, 1, \dots, K-m+1$ und $\ell = 0, 1, \dots, m-1$ gilt, folgt die behauptete Abschätzung. \square

24.4 Lernziele, Quiz und Anwendung

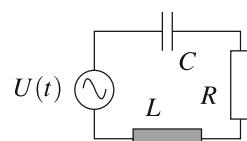
Sie sollten Stabilitätsprobleme von Mehrschrittverfahren präzisieren und die Dahlquist-sche Wurzelbedingung erklären können. Eine Beweisskizze für die Herleitung von Fehlerabschätzungen für Mehrschrittverfahren sollten Sie angeben können.

Quiz 24.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Nullstabilität ist ein notwendiges Kriterium für die Konvergenz eines Mehrschrittverfahrens positiver Konsistenzordnung	
Jedes Einschrittverfahren erfüllt die Dahlquistsche Wurzelbedingung	
Die Dahlquistsche Wurzelbedingung eines Mehrschrittverfahrens impliziert die Nullstabilität der zugehörigen Differenzengleichung	
Die Rekursionsformel $y_{k+2} = y_{k+1} - (1/4)y_k$ ist nullstabil	
Erfüllt ein Mehrschrittverfahren die Dahlquistsche Wurzelbedingung, so gilt für die zugehörige Begleitmatrix $\rho(A) < 1$	

Anwendung 24.1 Die Simulation elektrischer Schaltkreise ermöglicht die Vorhersage der auf die Bauteile abfallenden Spannungen. Als Beispiel betrachten wir einen RLC-Schaltkreis, der aus einem Widerstand (*Resistor*), einer Spule (*Inductor*) und einem Kondensator (*Capacitor*) besteht, wie in Abb. 24.2 dargestellt. Nach dem Ohmschen Gesetz ist die auf den Widerstand abfallende Spannung U_R proportional zum durchfließenden

Abb. 24.2 Schaltbild eines RLC-Schaltkreises



Strom I_R , das heißt es gilt $U_R = RI_R$. Der durch den Kondensator fließende Strom I_C ist proportional zur Spannungsänderung, das heißt es gilt $I_C = CU'_C$. An der Spule hingegen ist die abfallenden Spannung U_L proportional zur Stromänderung, das heißt es gilt $U_L = LI'_L$. Die Kirchhoffschen Gesetze besagen, dass die Summe der durch einen Knoten eines Schaltkreises fließenden Ströme Null ist und dass die Summe der zu einer Masche gehörenden Spannungen verschwindet. Für den RLC-Schaltkreis mit zeitabhängiger Spannungsquelle $U(t)$ ergeben sich daher die Gleichungen

$$\begin{aligned} U(t) &= U_R(t) + U_L(t) + U_C(t), \\ I(t) &= I_R(t) = I_L(t) = I_C(t). \end{aligned}$$

Leiten Sie die Differenzialgleichung

$$I'' + \frac{R}{L}I' + \frac{1}{LC}I = U'$$

für den durch den Schaltkreis fließenden Strom $I(t)$ her und simulieren Sie diesen für die Anfangswerte $I(0) = 0 \text{ A}$, $I'(0) = 0.5 \text{ A/s}$, die Proportionalitätsfaktoren $R = 47 \Omega$, $L = 20 \text{ mH}$, $C = 0.1 \mu\text{F}$ und die Wechselspannung $U(t) = \sin(50 \cdot 2\pi t)230 \text{ V}$. Lösen Sie das Anfangswertproblem mit verschiedenen MATLAB-Routinen und testen Sie andere Werte der Kapazität. Stellen Sie die abfallenden Spannungen U_R , U_L und U_C als Funktionen der Zeit im Intervall $[0, T]$ mit $T = 10 \text{ ms}$ vergleichend in einer Grafik dar.

25.1 Steifheit

Die Konvergenzuntersuchungen der Euler-Verfahren im wichtigen Spezialfall $y' = \lambda y$ zeigen, dass Fehlerabschätzungen für das explizite Verfahren unter einer Voraussetzung $\tau|\lambda| \leq c$ gelten, während diese Bedingung für die implizite Variante nicht notwendig ist. In Anwendungen treten Differenzialgleichungen der Form $y' = Ay$ auf, bei denen die Matrix A negative Eigenwerte besitzt. Für diese Klasse sind implizite Verfahren besonders gut geeignet. Wir folgen in diesem Kapitel den Darstellungen in [1,5,6].

Beispiel 25.1 Für $\lambda < 0$ ist die Lösung des Anfangswertproblems

$$y' = \lambda y, \quad y(0) = y_0,$$

gegeben durch $y(t) = y_0 e^{\lambda t}$ und es gilt $|y(t)| \leq |y_0|$ für alle $t \geq 0$

- (a) Mit dem expliziten Euler-Verfahren ergibt sich $y_k = (1 + \tau\lambda)^k y_0$, $k \geq 0$, und diese Folge ist genau dann beschränkt, wenn $|1 + \tau\lambda| \leq 1$ gilt, das heißt wenn $\tau \leq 2/|\lambda|$ gilt.
- (b) Mit dem impliziten Euler-Verfahren ergibt sich $y_k = (1 - \tau\lambda)^{-k} y_0$, $k \geq 0$, und da $1 - \tau\lambda \geq 1$ gilt, ist die Folge für jede Wahl von $\tau > 0$ beschränkt.

Die Schwierigkeiten expliziter Verfahren werden bei Systemen von Differenzialgleichungen noch deutlicher.

Beispiel 25.2 Für $\lambda_1, \lambda_2 < 0$ ist die Lösung des Systems

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \frac{1}{2} \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 - \lambda_2 \\ \lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

gegeben durch

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} + e^{\lambda_2 t} \\ e^{\lambda_1 t} - e^{\lambda_2 t} \end{bmatrix}.$$

Mit dem expliziten Euler-Verfahren ergeben sich die Approximationen

$$\begin{aligned} y_{1,k} &= (1 + \tau \lambda_1)^k + (1 + \tau \lambda_2)^k, \\ y_{2,k} &= (1 + \tau \lambda_1)^k - (1 + \tau \lambda_2)^k, \end{aligned}$$

und die Folge $(y_{1,k}, y_{2,k})_{k \geq 0}$ ist genau dann beschränkt, wenn $|1 + \tau \lambda_1| \leq 1$ und $|1 + \tau \lambda_2| \leq 1$ gelten. Sind beispielsweise $\lambda_1 = -1$ und $\lambda_2 = -10^\alpha$ mit $\alpha \geq 2$ so sind die Beiträge $e^{\lambda_2 t} = e^{-10^\alpha t}$ zur exakten Lösung vernachlässigbar für $t \geq 10^{-\alpha/2}$, aber die Zeitschrittweite wird durch λ_2 in der Form $\tau \leq 2/|\lambda_2| = 2 \cdot 10^{-\alpha}$ bestimmt.

Das Auftreten betragsmäßig großer, negativer Eigenwerte in einer Differenzialgleichung führt auf den Begriff der Steifheit.

Definition 25.1 Die Differenzialgleichung $y' = f(t, y)$ heißt *steif*, wenn die Jacobi-Matrix $Df(t, \bar{y}) \in \mathbb{R}^{n \times n}$ für ein $t \geq 0$ und ein $\bar{y} \in \mathbb{R}^n$ einen Eigenwert $\lambda \in \mathbb{C}$ mit der Eigenschaft $\operatorname{Re}(\lambda) \ll 0$ besitzt.

Bemerkung 25.1 Durch die Matrix $A = Df(t_*, y(t_*))$ und ihre Eigenwerte wird das lokale Verhalten einer Lösung y zum Zeitpunkt $t_* \geq 0$ insbesondere im Hinblick auf Störungen beschrieben. Die durch $y(t_* + s) = y(t_*) + z(s)$ definierte Funktion z erfüllt für kleine Werte $s \geq 0$ die lineare Differenzialgleichung

$$z'(s) = y'(t_* + s) = f(y(t_*) + z(s)) \approx f(y(t_*)) + Az(s),$$

wobei Anfangswerte $z(0) = z_0$ betrachtet werden, um die Auswirkungen von Störungen zu beurteilen. Ist A diagonalisierbar und haben die Eigenwerte ausschließlich negative Realteile, so ist die Lösung y stabil in dem Sinne, dass kleine Störungen zu keinen großen Änderungen der Lösung führen.

25.2 A-Stabilität

Zur Identifikation geeigneter numerischer Verfahren für steife Differenzialgleichungen dient folgender Stabilitätsbegriff.

Definition 25.2 Ein numerisches Verfahren heißt *A-stabil* oder *unbedingt stabil*, falls für jede komplex diagonalisierbare Matrix $A \in \mathbb{R}^{n \times n}$ deren Eigenwerte allesamt nichtpositiv sind.

tive Realteile haben, die durch das Verfahren definierten Approximationen $(y_k)_{k \geq 0}$ der Differenzialgleichung $y' = Ay$ für alle Anfangsdaten und alle Zeitschrittweiten $\tau > 0$ beschränkt sind.

Durch Diagonalisierung der Matrix A genügt es, in der Definition skalare Gleichungen zu betrachten, in denen A durch eine Zahl $\lambda \in \mathbb{C}$ mit $\operatorname{Re}(\lambda) \leq 0$ ersetzt wird.

Beispiel 25.3 Das implizite Euler-Verfahren ist A -stabil, das explizite Euler-Verfahren hingegen nicht.

Mit sogenannten Stabilitätsfunktionen lässt sich die A -Stabilität von Einschrittverfahren analysieren.

Definition 25.3 Eine Funktion $g : S \rightarrow \mathbb{C}$ mit $S \subset \mathbb{C}$ heißt *Stabilitätsfunktion* des durch die Inkrementfunktion Φ definierten Einschrittverfahrens, wenn für alle $\lambda \in \mathbb{C}$, $y_0 \in \mathbb{R}$, $\tau > 0$ mit $\tau\lambda \in S$ und alle $k \in \mathbb{N}_0$ für die Approximationen des Anfangswertproblems $y' = \lambda y$, $y(0) = y_0$, gilt

$$y_{k+1} = y_k + \tau \Phi(t_k, y_k, y_{k+1}, \tau) = g(\tau\lambda)y_k.$$

Notwendig für A -Stabilität ist im Fall der Existenz einer Stabilitätsfunktion, dass $|g(z)| \leq 1$ für alle $z \in \mathbb{C}$ mit $\operatorname{Re}(z) \leq 0$ gilt.

Beispiel 25.4 (i) Für das explizite Euler-Verfahren gilt $g(z) = 1 + z$.
(ii) Für das implizite Euler-Verfahren gilt $g(z) = 1/(1 - z)$.
(iii) Für das Trapez- beziehungsweise Mittelpunktverfahren ist $\Phi(t, y_k, y_{k+1}, \tau) = \lambda(y_k + y_{k+1})/2$ und somit $g(z) = (2 + z)/(2 - z)$.

Für Runge–Kutta-Verfahren lässt sich eine geschlossene Formel für die Stabilitätsfunktion angeben.

Lemma 25.1 Es seien $\alpha \in \mathbb{R}^m$, $\beta \in \mathbb{R}^{m \times m}$ und $\gamma \in \mathbb{R}^m$ die Koeffizienten eines Runge–Kutta-Verfahrens. Dann ist

$$g(z) = 1 + z\gamma^\top(I_m - z\beta)^{-1}e$$

mit dem Vektor $e = [1, 1, \dots, 1]^\top \in \mathbb{R}^m$ die zugehörige Stabilitätsfunktion. Für strikte untere Dreiecksmatrizen β ist g für alle $z \in \mathbb{C}$ wohldefiniert.

Beweis Für Runge–Kutta-Verfahren gilt $y_{k+1} = y_k + \tau \gamma^\top \eta^k$ mit der Lösung $\eta^k = [\eta_1^k, \eta_2^k, \dots, \eta_m^k]^\top \in \mathbb{R}^m$ des Gleichungssystems

$$\eta_\ell^k = f \left(t + \tau_\ell, y_k + \tau \sum_{j=1}^m \beta_{\ell j} \eta_j^k \right)$$

für $\ell = 1, 2, \dots, m$. Im Spezialfall $f(t, y) = \lambda y$ ergibt sich

$$\eta^k = \lambda(y_k e + \tau \beta \eta^k)$$

beziehungsweise $\eta^k = (I_m - \lambda \tau \beta)^{-1}(\lambda y_k e)$, woraus die behauptete Identität für g folgt. Ist β eine strikte untere Dreiecksmatrix, so ist die Matrix $I_m - z\beta$ für jedes $z \in \mathbb{C}$ invertierbar. \square

Mittels Vorwärtssubstitution erhält man im Fall expliziter Runge–Kutta-Verfahren polynomielle Ausdrücke. Für implizite Verfahren ergeben sich rationale Funktionen.

Beispiel 25.5 Das klassische Runge–Kutta-Verfahren ist durch $m = 4$ sowie $\alpha = [0, 1/2, 1/2, 1]^\top$, $\gamma = [1, 2, 2, 1]^\top / 6$ und $\beta \in \mathbb{R}^{4 \times 4}$ mit den nichtverschwindenden Einträgen $\beta_{21} = \beta_{32} = 1/2$ sowie $\beta_{43} = 1$ definiert. Damit folgt

$$g(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}.$$

Aus der Stabilitätsfunktion eines Einschrittverfahrens lassen sich verschiedene Aussagen ableiten.

Satz 25.1 Es sei $g : S \rightarrow \mathbb{C}$ die Stabilitätsfunktion eines Einschrittverfahrens mit der Eigenschaft, dass $\{z \in \mathbb{C} : \operatorname{Re}(z) \leq 0\} \subset S$.

- (i) Das Verfahren ist genau dann A-stabil, wenn $|g(z)| \leq 1$ für alle $z \in \mathbb{C}$ mit $\operatorname{Re}(z) \leq 0$ gilt.
- (ii) Im Fall eines expliziten Runge–Kutta-Verfahrens gilt $\lim_{|z| \rightarrow \infty} |g(z)| = \infty$, das heißt explizite Runge–Kutta-Verfahren sind nicht A-stabil.
- (iii) Ist das Verfahren konsistent von der Ordnung $p \geq 0$, so gilt $|e^z - g(z)| \leq c|z|^{p+1}$ für $z \rightarrow 0$.

Beweis

- (i) Gilt $|g(z)| \leq 1$, so folgt unmittelbar die Beschränktheit der Folge $y_k = g(\tau \lambda)^k y_0$, $k \in \mathbb{N}$.
- (ii) Für explizite Runge–Kutta-Verfahren folgt aus der Darstellung $g(z) = 1 + z \gamma^\top (I_m - z\beta)^{-1} e$, dass g ein Polynom in z ist, woraus die Behauptung folgt.

Tab. 25.1 Übersicht verschiedener Stabilitätsbegriffe

Stabilitätsbegriff	Testgleichung	Bedeutung	Beispiel
Nullstabilität	$y' = 0$	Notwendiges Konvergenzkriterium	Adams-Verfahren
A-Stabilität	$y' = \lambda y$	Vermeidung einer Schrittweitenbedingung	Trapez-Verfahren
L-Stabilität	$y' = \lambda y$	Numerische Dämpfungseigenschaft	Implizites Euler-Verfahren

- (iii) Sei $\lambda \in \mathbb{C}$. Für die Lösung $y(t) = e^{t\lambda}$, des Anfangswertproblems $y' = \lambda y$, $y(0) = 1$, und die Approximation $y_1 = g(\tau\lambda)y_0 = g(\tau\lambda)$ folgt mit der Definition des Konsistenzterms, dass

$$|e^{\tau\lambda} - g(\tau\lambda)| = |y(\tau) - y_1| \leq c|\tau\lambda|^{p+1}$$

für alle $0 < \tau \leq c'$ gilt. Mit $z = \tau\lambda$ folgt die Behauptung. \square

Korollar 25.1 Ein m -stufiges, explizites Runge–Kutta-Verfahren besitzt höchstens die Konsistenzordnung m .

Beweis Die Stabilitätsfunktion ist ein Polynom vom Grad m und dieses kann die Funktion e^z höchstens bis auf einen Fehler $\mathcal{O}(z^{m+1})$ approximieren. \square

Die unbedingte Beschränktheit von Approximationen ist eine sinnvolle Forderung an numerische Verfahren. In einigen Situationen kann ein zu rasches Abklingen der Approximationslösungen jedoch unerwünscht sein. Für das implizite Euler-Verfahren gilt beispielsweise $|g(z)| \rightarrow 0$ für $z \rightarrow -\infty$, also ein starkes Dämpfungsverhalten für große Schrittweiten. Für das Trapez-Verfahren hingegen gilt $|g(z)| \rightarrow 1$ für $|z| \rightarrow \infty$, sodass keine numerische Dämpfung für große Zeitschrittweiten eintritt, aber mit Oszillationen gerechnet werden muss.

Definition 25.4 Ein Einschrittverfahren heißt *L-stabil*, falls es A-stabil ist und $\lim_{\operatorname{Re}(z) \rightarrow -\infty} g(z) = 0$ gilt.

Die A-Stabilität beschreibt also die unbedingte Stabilität eines Verfahrens und die L-Stabilität die zusätzlichen Dämpfungseigenschaften des Verfahrens für große Schrittweiten. Eine Übersicht verschiedener Stabilitätsbegriffe ist in Tab. 25.1 dargestellt.

Bemerkungen 25.2 (i) Das implizite Euler-Verfahren ist L-stabil.

(ii) Das Trapez-Verfahren ist A-stabil, aber nicht L-stabil.

25.3 Gradientenflüsse

Eine wichtige Klasse steifer Differenzialgleichungen sind *Gradientenflüsse*, in denen die rechte Seite durch den negativen Gradienten einer Funktion gegeben ist, das heißt autonome Differenzialgleichungen der Form

$$y'(t) = -\nabla G(y(t)), \quad y(0) = y_0,$$

mit einer gegebenen Funktion $G \in C^1(\mathbb{R}^n)$. Diese Anfangswertprobleme lassen sich als kontinuierliche Abstiegsverfahren für die Minimierung der Funktion G interpretieren. Der Wert der Funktion G wird entlang des Pfades $t \mapsto y(t)$ verringert, denn multipliziert man die Differenzialgleichung mit $-y'(t)$, so folgt

$$-\|y'(t)\|^2 = -y'(t) \cdot y'(t) = \nabla G(y(t)) \cdot y'(t) = \frac{d}{dt}G(y(t)).$$

Ist die Funktion G *koerziv*, das heißt gilt $G(w) \rightarrow \infty$ für $|w| \rightarrow \infty$, so folgt, dass Lösungen beschränkt bleiben und für alle $t \in [0, \infty)$ definiert sind, selbst wenn G' nicht global Lipschitz-stetig ist. Ist G μ -konvex, das heißt existiert eine Zahl $\mu > 0$, sodass $G(s) + (\mu/2)|s|^2$ konvex ist, so ist das implizite Euler-Verfahren wohldefiniert und unbedingt stabil. Man beachte, dass Gradientenflüsse im Allgemeinen keine linearen Differenzialgleichungen der Form $y' = Ay$ definieren.

Satz 25.2 Sei $G \in C^2(\mathbb{R}^n)$ μ -konvex. Für $0 < \tau < \mu^{-1}$ und $y_0 \in \mathbb{R}^n$ wird durch

$$y_{k+1} = y_k - \tau \nabla G(y_{k+1})$$

eine Folge $(y_k)_{k \geq 0}$ eindeutig definiert und für alle $\ell \geq 0$ gilt

$$G(y_\ell) + \frac{1}{2\tau} \sum_{k=0}^{\ell-1} \|y_{k+1} - y_k\|^2 \leq G(y_0).$$

Beweis Sei $y_k \in \mathbb{R}^n$ für ein $k \geq 0$ gegeben. Die Abbildung

$$H_{k+1}(s) = \frac{1}{2\tau} \|s - y_k\|^2 + G(s)$$

ist für $0 < \tau < \mu^{-1}$ strikt konvex, das heißt $D^2 H_{k+1}(s)$ ist für alle $s \in \mathbb{R}^n$ positiv definit, und ihr einziges Minimum $y_{k+1} \in \mathbb{R}^n$ wird in einer kompakten Menge $B_r(0)$ angenommen. Für dieses gilt die Optimalitätsbedingung

$$0 = \nabla H_{k+1}(y_{k+1}) = \frac{1}{\tau}(y_{k+1} - y_k) + \nabla G(y_{k+1}).$$

Mit der Konvexitätseigenschaft

$$\nabla H_{k+1}(v)(w - v) + H_{k+1}(v) \leq H_{k+1}(w),$$

die für alle $v, w \in \mathbb{R}^n$ gilt, folgt mit $v = y_{k+1}$ und $w = y_k$, dass

$$G(y_{k+1}) + \frac{1}{2\tau} \|y_{k+1} - y_k\|^2 \leq G(y_k)$$

und die Summation dieser Abschätzung über $k = 0, 1, \dots, \ell - 1$ zeigt die Behauptung. \square

Bemerkungen 25.3 (i) Im Fall der Beschränktheit $G \geq -c$ folgt, dass $y_{k+1} - y_k \rightarrow 0$ für $k \rightarrow \infty$ gilt, und somit die Konvergenz $y_{k_\ell} \rightarrow y_*$ einer Teilfolge gegen einen stationären Punkt beziehungsweise eine Minimalstelle y_* von G , das heißt $\nabla G(y_*) = 0$.

(ii) Man beachte, dass keine Lipschitz-Stetigkeit von ∇G vorausgesetzt wurde.

(iii) Die Gleichung $y_{k+1} = y_k - \tau \nabla G(y_{k+1})$ löst man mit einer Fixpunktiteration oder dem Newton-Verfahren.

(iv) Ist G als Summe einer konvexen und einer konkaven Funktion gegeben, das heißt gilt $G = G^{cx} + G^{cv}$, so ist das *implizit-explizite* Verfahren

$$y_{k+1} = y_k - \tau \nabla G^{cx}(y_{k+1}) - \tau \nabla G^{cv}(y_k)$$

dem impliziten Verfahren aufgrund der besseren Lösbarkeit mit dem Newton-Verfahren vorzuziehen. Häufig stellen *semiimplizite* Verfahren basierend auf der Linearisierung $\nabla G(y_{k+1}) \approx \nabla G(y_k) + D^2 G(y_k)(y_{k+1} - y_k)$, das heißt

$$y_{k+1} = y_k - \tau [\nabla G(y_k) - D^2 G(y_k)(y_{k+1} - y_k)]$$

eine gute Alternative zum impliziten Verfahren dar. Dies entspricht gerade der Durchführung eines Schritts des Newton-Verfahrens für das implizite Schema.

25.4 Wärmeleitungsgleichung

Steife Differentialgleichungen treten bei der räumlichen Diskretisierung parabolischer partieller Differentialgleichungen wie beispielsweise der Wärmeleitungsgleichung auf. In einer eindimensionalen Situation ist dabei eine Funktion $u : [0, T] \times [a, b] \rightarrow \mathbb{R}$ gesucht, die das Anfangsrandwertproblem

$$\begin{aligned} \partial_t u(t, x) - c \partial_x^2 u(t, x) &= f(t, x) & (t, x) \in (0, T] \times (a, b), \\ u(0, x) &= u_0(x) & x \in [a, b], \\ u(t, a) &= 0, \quad u(t, b) = 0 & t \in (0, T] \end{aligned}$$

löst, wobei die rechte Seite $f \in C^0([0, T] \times [a, b])$ und die Anfangswerte $u_0 \in C^0([a, b])$ gegeben sind. Die Funktion u beschreibt die Temperaturverteilung in einem dünnen Metalldraht, dessen Enden konstant auf Temperatur 0 gehalten werden und zum Zeitpunkt $t = 0$ die Temperaturverteilung u_0 besitzt. Die rechte Seite f beschreibt mögliche Wärmequellen und -senken im Draht. Ähnlich wie die Approximation einer ersten Ableitung kann eine zweite Ableitung mit einem Differenzenquotienten approximiert werden. Für eine Schrittweite $h > 0$ gilt

$$\frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = u''(x) + \mathcal{O}(h^2).$$

Mit der Ortsschrittweite $h = (b-a)/M$ und den Gitterpunkten $x_j = a + jh$, $j = 0, 1, \dots, M$ sowie einer Zeitschrittweite $\tau > 0$ und den Zeitschritten $t_k = k\tau$, $k = 0, 1, \dots, K$, werden Approximationen

$$U_j^k \approx u(t_k, x_j)$$

gesucht. Das Ersetzen der Zeitableitung und der zweiten Ortsableitung mit Differenzenquotienten führt auf die Identitäten

$$\frac{1}{\tau}(U_j^{k+1} - U_j^k) - \frac{c}{h^2}(U_{j-1}^{k+1} - 2U_j^{k+1} - U_{j+1}^{k+1}) = F_j^{k+1}$$

für $k = 0, 1, \dots, K-1$, $j = 1, 2, \dots, M-1$ und $F_j^{k+1} = f(t_{k+1}, x_j)$. An den Randknoten werden die Randbedingungen $U_0^{k+1} = U_M^{k+1} = 0$ verwendet. Die Gleichungen für $j = 1, 2, \dots, M-1$ lassen sich simultan schreiben als

$$\frac{1}{\tau} \begin{bmatrix} U_1^{k+1} \\ U_2^{k+1} \\ \vdots \\ U_{M-1}^{k+1} \end{bmatrix} - \frac{c}{h^2} \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \begin{bmatrix} U_1^{k+1} \\ U_2^{k+1} \\ \vdots \\ U_{M-1}^{k+1} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} U_1^k \\ U_2^k \\ \vdots \\ U_{M-1}^k \end{bmatrix} + \begin{bmatrix} F_1^{k+1} \\ F_2^{k+1} \\ \vdots \\ F_{M-1}^{k+1} \end{bmatrix}.$$

Mit den Vektoren $\hat{U}^k = [U_1^k, U_2^k, \dots, U_{M-1}^k]^\top$ und $F^k = [F_1^k, F_2^k, \dots, F_{M-1}^k]^\top$ ist dies äquivalent zu

$$\hat{U}^{k+1} = \hat{U}^k + \tau \left(\frac{c}{h^2} A \hat{U}^{k+1} + F^{k+1} \right)$$

für $k = 0, 1, \dots, K-1$ mit den Anfangsdaten $\hat{U}^0 = [u_0(x_1), \dots, u_0(x_{M-1})]^\top$. Dies lässt sich als implizite Zeitdiskretisierung eines Systems linearer Differenzialgleichungen interpretieren, das heißt des Anfangswertproblems

$$U'(t) = \frac{c}{h^2} A U(t) + F(t), \quad U(0) = \hat{U}_0,$$

mit der symmetrischen und negativ definiten Matrix $A \in \mathbb{R}^{(M-1) \times (M-1)}$ für die $\text{cond}_2(A) \sim h^{-2}$ gilt, das heißt einer steifen Differenzialgleichung.

25.5 Lernziele, Quiz und Anwendung

Ihnen sollten Probleme expliziter Verfahren bei steifen Differenzialgleichungen bekannt sein und Sie sollten Begriffe zur Kategorisierung der Stabilität eines numerischen Verfahrens benennen und erklären können. Gradientenflüsse sollten Sie definieren und die grundlegenden Eigenschaften der Anwendung des impliziten Euler-Verfahrens beweisen können.

Quiz 25.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

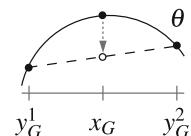
Aussage	Beurteilung
Die Differenzialgleichung $y' = e^{5t} \sin(y)$ ist steif	
Das explizite Euler-Verfahren ist A -stabil und das implizite Euler-Verfahren L -stabil	
Das Trapez-Verfahren ist konsistent von der Ordnung $p = 2$ und A -stabil	
Die Stabilitätsfunktion jedes expliziten Runge–Kutta-Verfahrens ist unbeschränkt	
Das Richardson-Verfahren zur iterativen Lösung von $Ax = b$ entspricht der Anwendung des expliziten Euler-Verfahrens auf $y' = -(Ay - b)$	

Anwendung 25.1 Wärmeleitungs- und Diffusionsprozesse sind Ausgleichsprozesse, die einen stationären, das heißt einen zeitlich unveränderlichen Zustand anstreben. Zur mathematischen Beschreibung betrachten wir einen Metallkörper und modellieren diesen als uniformes Partikelgitter. Um einen Ausgleichseffekt zu erhalten, verwenden wir die Annahme, dass die Veränderung der Temperatur an jedem inneren Gitterpunkt proportional ist zur Abweichung der Temperatur vom Mittelwert der Temperaturen an den Nachbarpunkten, das heißt

$$\partial_t \theta(t, x_G) = -\frac{c}{h^2} \left(\theta(t, x_G) - \frac{1}{|\mathcal{N}(x_G)|} \sum_{y_G \in \mathcal{N}(x_G)} \theta(t, y_G) \right),$$

wobei $\mathcal{N}(x_G)$ die Menge der Nachbarpunkte von x_G mit Kardinalität $|\mathcal{N}(x_G)|$ bezeichne und h die Gitterweite sei, s. Abb. 25.1. An den Randpunkten sei die Temperatur durch den Wert 0 und zum Zeitpunkt $t = 0$ durch Werte $\theta_0(x_G)$ vorgegeben.

Abb. 25.1 Diffusionsprozesse streben einen Gleichgewichtszustand an



- (i) Zeigen Sie, dass die rechte Seite der Differenzialgleichung im Fall einer zweimal stetig differenzierbaren Funktion θ für $h \rightarrow 0$ gegen $c\theta''(t, x_G)$ beziehungsweise $c\Delta\theta(t, x_G)$ konvergiert.
- (ii) Zeigen Sie, dass sich der Wärmeleitungsprozess als System von Differenzialgleichungen $Y' = AY$ in $(0, T]$ mit Anfangsbedingung $Y(0) = Y_0$ formulieren lässt und spezifizieren Sie die Matrix A für den Fall einer Metallplatte, die als uniformes Gitter des Gebiets $(0, 1)^2$ beschrieben wird.
- (iii) Verwenden Sie die Gitterweite $h = 1/20$ sowie das implizite und explizite Euler-Verfahren mit verschiedenen Zeitschrittweiten und zufällig generierten Anfangsdaten. Beurteilen Sie, für welche Schrittweiten Sie sinnvolle Approximationslösungen erhalten und ob es sich um eine steife Differenzialgleichung handelt.

26.1 A-posteriori Fehlerkontrolle

Wir leiten im Folgenden eine Fehlerabschätzung her, die von den berechneten Approximationen und den nichtkonstanten Schrittweiten abhängt. Die Abschätzung erlaubt die optimale lokale Anpassung der Schrittweiten und führt damit auf effiziente Verfahren, s. Abb. 26.1.

Die Folge $(y_k)_{k=0,\dots,K}$ sei durch das implizite Euler-Verfahren

$$y_{k+1} = y_k + \tau_{k+1} f(y_{k+1})$$

mit möglicherweise nichtkonstanten Schrittweiten $\tau_{k+1} = t_{k+1} - t_k > 0$ definiert. Wir identifizieren τ mit der Folge $(\tau_k)_{k=1,\dots,K}$ und nehmen an, dass $t_K = T$ gilt.

Definition 26.1 Der *affin-lineare Interpolant* $\hat{y}_\tau : [0, T] \rightarrow \mathbb{R}$ ist für $t \in [t_k, t_{k+1}]$, $k = 0, 1, \dots, K-1$, definiert durch

$$\hat{y}_\tau(t) = \frac{t - t_{k+1}}{t_k - t_{k+1}} y_k + \frac{t - t_k}{t_{k+1} - t_k} y_{k+1}.$$

Der *stückweise konstante Interpolant* $\bar{y}_\tau : [0, T] \rightarrow \mathbb{R}$ ist für $t \in (t_k, t_{k+1}]$, $k = 0, 1, \dots, K-1$, definiert durch

$$\bar{y}_\tau(t) = y_{k+1}.$$

Abb. 26.1 Adaptive Steuerung lokaler Schrittweiten

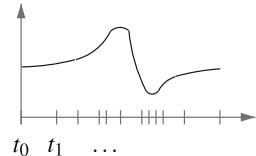
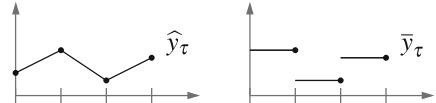


Abb. 26.2 Stückweise lineare undstückweise konstante Interpolanten gegebener Approximationswerte



Die Interpolanten sind in Abb. 26.2 beispielhaft gezeigt.

Nach Definition von \hat{y}_τ gilt für $t \in (t_k, t_{k+1})$

$$\hat{y}'_\tau(t) = \frac{1}{\tau_{k+1}}(y_{k+1} - y_k)$$

und mit der Definition von \bar{y}_τ folgt, dass sich das Euler-Verfahren für alle $t \in (t_k, t_{k+1})$ schreiben lässt in der Form

$$\hat{y}'_\tau(t) = f(\bar{y}_\tau(t)) = f(\hat{y}_\tau(t)) + (f(\bar{y}_\tau(t)) - f(\hat{y}_\tau(t))).$$

Die Funktion \hat{y}_τ löst daher die Differenzialgleichung außerhalb der Zeitschritte $(t_k)_{k=1,2,\dots,K-1}$ bis auf das Residuum

$$R_\tau = f(\hat{y}_\tau) - f(\bar{y}_\tau).$$

Diese Beobachtung lässt sich quantifizieren und führt auf eine *a-posteriori Fehlerabschätzung*.

Satz 26.1 Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ Lipschitz-stetig mit Konstante $L \geq 0$ und $y \in C^1([0, T])$ die Lösung des Anfangswertproblems $y' = f(y)$, $y(0) = y_0$. Dann gilt

$$\sup_{t \in [0, T]} |y(t) - \hat{y}_\tau(t)|^2 \leq \frac{L}{3} \left(\sum_{k=0}^{K-1} \tau_{k+1} |y_{k+1} - y_k|^2 \right) \exp(3LT).$$

Beweis Die Subtraktion der Identitäten

$$y' = f(y), \quad \hat{y}'_\tau = f(\hat{y}_\tau) + (f(\bar{y}_\tau) - f(\hat{y}_\tau))$$

zeigt, dass der Fehler $e_\tau = y - \hat{y}_\tau$ die Gleichung

$$e'_\tau = (f(y) - f(\hat{y}_\tau)) - (f(\bar{y}_\tau) - f(\hat{y}_\tau))$$

erfüllt. Die Multiplikation dieser Gleichung mit e zeigt unter Verwendung der Produktregel $(|e_\tau|^2)' = 2e'_\tau e_\tau$, dass

$$\begin{aligned} \frac{d}{dt} \frac{1}{2} |e_\tau|^2 &= e'_\tau e_\tau = (f(y) - f(\hat{y}_\tau))e_\tau - (f(\bar{y}_\tau) - f(\hat{y}_\tau))e_\tau \\ &\leq L|e_\tau|^2 + L|\hat{y}_\tau - \bar{y}_\tau||e_\tau|. \end{aligned}$$

Mit der Ungleichung $2ab \leq a^2 + b^2$ folgt

$$\begin{aligned} \frac{d}{dt} \frac{1}{2} |e_\tau|^2 &\leq L |e_\tau|^2 + \frac{L}{2} |\hat{y}_\tau - \bar{y}_\tau|^2 + \frac{L}{2} |e_\tau|^2 \\ &= \frac{3L}{2} |e_\tau|^2 + \frac{L}{2} |\hat{y}_\tau - \bar{y}_\tau|^2. \end{aligned}$$

Da e stetig und stückweise differenzierbar ist, zeigt die Integration dieser Ungleichung über $(0, t)$, dass

$$|e_\tau(t)|^2 - |e_\tau(0)|^2 \leq 3L \int_0^t |e_\tau(s)|^2 ds + L \int_0^T |\hat{y}_\tau(s) - \bar{y}_\tau(s)|^2 ds.$$

Eine Anwendung des Gronwall-Lemmas unter Berücksichtigung von $e_\tau(0) = 0$ impliziert, dass für alle $t \in [0, T]$ die Abschätzung

$$|e_\tau(t)|^2 \leq L \left(\int_0^T |\hat{y}_\tau(s) - \bar{y}_\tau(s)|^2 ds \right) \exp(3LT)$$

gilt. Auf jedem Intervall (t_k, t_{k+1}) gilt

$$\begin{aligned} \hat{y}_\tau(s) - \bar{y}_\tau(s) &= \frac{1}{\tau_{k+1}} ((t_{k+1} - s)y_k + (s - t_k)y_{k+1}) - \frac{t_{k+1} - t_k}{\tau_{k+1}} y_{k+1} \\ &= \frac{s - t_{k+1}}{\tau_{k+1}} (y_{k+1} - y_k). \end{aligned}$$

Damit folgt

$$\begin{aligned} \int_0^T |\hat{y}_\tau(s) - \bar{y}_\tau(s)|^2 ds &= \sum_{k=0}^{K-1} \frac{(y_{k+1} - y_k)^2}{\tau_{k+1}^2} \int_{t_k}^{t_{k+1}} (t_{k+1} - s)^2 ds \\ &= \sum_{k=0}^{K-1} \frac{(y_{k+1} - y_k)^2}{\tau_{k+1}^2} \frac{\tau_{k+1}^3}{3}. \end{aligned}$$

Dies beweist die Behauptung. \square

Bemerkung 26.1 Die Abschätzung des Satzes heißt *a-posteriori* Fehlerabschätzung, da sie den Approximationsfehler $y - \hat{y}_\tau$ nach Berechnung der numerischen Lösung durch berechenbare Größen beschränkt.

26.2 Adaptiver Algorithmus

Die a-posteriori Fehlerabschätzung erlaubt eine adaptive Anpassung der Zeitschrittweiten, das heißt die Schrittweite τ_{k+1} sollte so lange verkleinert werden, bis der durch $\eta_{k+1} = |y_{k+1} - y_k|$ definierte Fehlerindikator die Abschätzung $\eta_{k+1} \leq \delta$ mit einer vorgegebenen Toleranz δ erfüllt. Umgekehrt motiviert eine Ungleichung $\eta_{k+1} \leq \delta$ die Vergößerung der Schrittweite im nachfolgenden Zeitschritt.

Algorithmus 26.1 (Schrittweitensteuerung) Seien $\delta > 0$, $y_0 \in \mathbb{R}$ und $\tau_1 > 0$. Setze $k = 0$ und $t_0 = 0$.

(1) Berechne y_{k+1} mittels

$$y_{k+1} = y_k + \tau_{k+1} \Phi(t_k, y_k, y_{k+1}, \tau_{k+1}).$$

(2) Gilt $\eta_{k+1} > \delta$, so setze $\tau_{k+1} \rightarrow \tau_{k+1}/2$ und wiederhole (1).

(3) Stoppe falls $t_{k+1} = t_k + \tau_{k+1} = T$; andernfalls erhöhe $k \rightarrow k + 1$, setze $\tau_{k+1} = \min\{2\tau_k, T - t_k\}$, und wiederhole Schritt (1).

26.3 Kontrollverfahren

Liegt keine a-posteriori Fehlerabschätzung vor, so besteht eine weniger rigorose Möglichkeit der Schrittweitensteuerung in der Verwendung eines sogenannten *Kontrollverfahrens*. Dies ist ein zusätzliches Verfahren höherer Konsistenzordnung als das eigentlich verwendete Verfahren. Sind $(y_k)_{k=0,\dots,K}$ Approximationen der Ordnung $\mathcal{O}(\tau^p)$ und $(\tilde{y}_k)_{k=0,\dots,K}$ Approximationen der Ordnung $\mathcal{O}(\tau^q)$ mit $q > p$, so folgt

$$|y(t_k) - y_k| \leq |y(t_k) - \tilde{y}_k| + |\tilde{y}_k - y_k| = \mathcal{O}(\tau^q) + |\tilde{y}_k - y_k|.$$

Mit der umgekehrten Dreiecksungleichung erhält man zudem

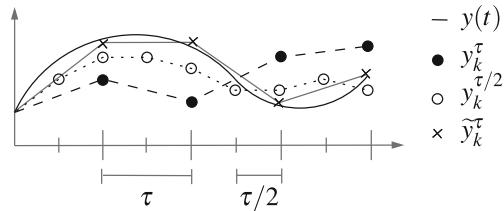
$$|\tilde{y}_k - y_k| - \mathcal{O}(\tau^q) = |\tilde{y}_k - y_k| - |y(t_k) - \tilde{y}_k| \leq |y_k - y(t_k)|.$$

Insgesamt gilt also bis auf Terme der Ordnung $\mathcal{O}(\tau^q)$, dass

$$|y(t_k) - y_k| \approx \eta_k = |\tilde{y}_k - y_k|,$$

das heißt die berechenbare Größe η_k approximiert den tatsächlichen Fehler bis auf Terme höherer Ordnung.

Abb. 26.3 Konstruktion eines Kontrollverfahrens durch Extrapolation der Approximationen y^τ und $y^{\tau/2}$



26.4 Extrapolation

Als Kontrollverfahren zur Schrittweitensteuerung kann eine extrapolierte Approximationslösung dienen. Für die mit einem Einschrittverfahren der Konsistenzordnung p berechneten Approximationen $(y_k^\tau)_{k=0,\dots,K}$ der exakten Lösung $y : [0, T] \rightarrow \mathbb{R}$ kann der Fehler $y(t_k) - y_k^\tau$ als Funktion $\varphi(\tau^p)$ angenommen werden. Eine Taylor-Entwicklung der Funktion φ führt auf die Darstellung

$$\begin{aligned} y(t_k) - y_k^\tau &= \varphi(\tau^p) = \varphi(0) + \varphi'(0)\tau^p + \frac{1}{2}\varphi''(0)\tau^{2p} + o(\tau^{2p}) \\ &= c_1\tau^p + c_2\tau^{2p} + o(\tau^{2p}). \end{aligned}$$

Wird dasselbe Verfahren mit der Schrittweite $\tau/2$ verwendet, so erhalten wir durch $y_{2k}^{\tau/2}$ eine weitere Approximation von $y(t_k)$ und dürfen die Fehlerdarstellung

$$y(t_k) - y_{2k}^{\tau/2} = \varphi((\tau/2)^p) = c_1 2^{-p} \tau^p + c_2 2^{-2p} \tau^{2p} + o(\tau^{2p})$$

annehmen. Die Multiplikation der zweiten Gleichung mit 2^p und anschließende Subtraktion von der ersten Gleichung führen auf

$$(1 - 2^p)y(t_k) - y_k^\tau + 2^p y_{2k}^{\tau/2} = c_2(1 - 2^{-p})\tau^{2p} + o(\tau^{2p}),$$

das heißt der Term $c_1\tau^p$ wird eliminiert. Daraus folgt

$$\tilde{y}_k^\tau = \frac{y_k^\tau - 2^p y_{2k}^{\tau/2}}{1 - 2^p} = y(t_k) - c_2 \frac{1 - 2^{-p}}{1 - 2^p} \tau^{2p} + o(\tau^{2p}),$$

sodass der berechenbare Ausdruck \tilde{y}_k^τ den Funktionswert $y(t_k)$ bis auf einen Fehlerterm der Ordnung $\mathcal{O}(\tau^{2p})$ approximiert. Wir haben also ein Verfahren konstruiert, bei dem der Aufwand zwar etwa verdoppelt wird, der Fehler jedoch quadriert wird. Dieses Vorgehen lässt sich rigoros analysieren und verallgemeinern.

26.5 Lernziele, Quiz und Anwendung

Sie sollten die grundlegenden Konzepte der Schrittweitensteuerung erklären und eine a-posteriori Fehlerabschätzung herleiten können. Die Rolle von Kontrollverfahren sollten Sie erklären können.

Quiz 26.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Der Approximationsfehler des impliziten Euler-Verfahrens lässt sich ohne Kenntnis der exakten Lösung nicht beschränken	
Im adaptiven Algorithmus wird eine Schrittweite bestimmt, mit der sämtliche Approximationen berechnet werden	
Für alle $a, b \in \mathbb{R}$ und $\gamma > 0$ gilt $ab \leq \gamma a^2/2 + b^2/(2\gamma)$	
Ist $y \in C^0([0, T])$ und $t_k = k\tau$ für $k = 0, 1, \dots, K$ mit $\tau = T/K$, so gilt $\max_{k=0, \dots, K-1} y(t_{k+1}) - y(t_k) \rightarrow 0$ für $\tau \rightarrow 0$	
Durch Extrapolation eines Einschrittverfahrens der Konsistenzordnung $p \geq 1$ mit Schrittweiten τ und $\tau/2$ erhält man ein Verfahren der Konsistenzordnung $p + 1$	

Anwendung 26.1 In einfachen Märkten wird der Preis p eines Produkts durch das Angebot a und die Nachfrage n bestimmt. Die Nachfrage nimmt mit steigendem Preis ab, während das Angebot mit zunehmendem Preis größer wird. Folglich ist $p \mapsto n(p)$ eine monoton fallende und $p \mapsto a(p)$ eine monoton wachsende Funktion. Ein Unterschied zwischen Angebot und Nachfrage führt zu einer Änderung des Preises, das heißt es gilt

$$p'(t) = \alpha(n(p) - a(p)).$$

- (i) Zeigen Sie, dass sich unter geeigneten Voraussetzungen an n und a stets ein Gleichgewichtszustand einstellt und dieser bei kleinen Störungen exponentiell schnell angenommen wird.
- (ii) In der Realität ist möglicherweise die Anzahl der gekauften Produkte geringer als die Nachfrage, da beispielsweise der Preis noch nicht dem tatsächlichen Wert des Produkts entspricht, jedoch bewegt sich diese Anzahl auf die Nachfrage zu. Modifizieren und erweitern Sie das Modell um diesen Verzögerungseffekt.
- (iii) Wie lässt sich die Abhängigkeit von äußeren Faktoren wie die Verfügbarkeit benötigter Rohstoffe im Modell berücksichtigen?

27.1 Hamiltonsche Systeme

Ein Hamiltonsches System beschreibt die Dynamik von N Körpern im dreidimensionalen Raum mittels einer differenzierbaren Funktion $H : \mathbb{R}^{N \times 3} \times \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}$ und dem System von Differentialgleichungen

$$q' = \partial_p H(q, p), \quad p' = -\partial_q H(q, p)$$

im Intervall $(0, T]$ mit Anfangsdaten für q und p . Die Funktionen $q_i, p_i : [0, T] \rightarrow \mathbb{R}^3$, $i = 1, 2, \dots, N$, beschreiben die Positionen und Impulse der Körper und H ist die Summe aus kinetischer und potenzieller Energie, das heißt beispielsweise

$$H(q, p) = \sum_{i=1}^N \frac{\|p_i\|^2}{2m_i} + V(q_1, q_2, \dots, q_N),$$

mit den Massen m_i , $i = 1, 2, \dots, N$, der Körper.

Beispiele 27.1 (i) Das durch die Differentialgleichung $\phi'' = -(g/\ell) \sin(\phi)$ beschriebene Fadenpendel lässt sich als Hamiltonsches System der Funktion

$$H(\phi, \psi) = \frac{1}{2m\ell^2}\psi^2 - mg\ell \cos(\phi)$$

darstellen, denn es folgt $m\ell^2\phi' = \psi$ sowie $\psi' = -mg\ell \sin(\phi)$.

(ii) Mehrkörperprobleme wie beispielsweise Sonnensysteme lassen sich durch Hamiltonsche Systeme beschreiben.

(iii) Durch $[\partial_p H(q, p), -\partial_q H(q, p)]^\top$ wird ein Tangentialvektor an den Graphen von H im Punkt (p, q) definiert. Das zugehörige Hamiltonsche System folgt damit einer Niveaulinie der Funktion H .

Hamiltonsche Systeme erfüllen Erhaltungsprinzipien für Gesamtdrehimpuls und Gesamtenergie.

Beispiel 27.2 Die Gesamtenergie eines Hamiltonschen Systems ist konstant, denn es gilt

$$\frac{d}{dt}H(q, p) = \partial_p H(q, p)p' + \partial_q H(q, p)q' = 0.$$

Fasst man die Variablen q und p in einem Vektor $z \in \mathbb{R}^{2n}$ mit $n = dN$ und $d \in \{1, 2, 3\}$ zusammen und identifiziert $H(q, p) = H(z)$, so lässt sich ein Hamiltonsches System schreiben als

$$z' = J \nabla H(z), \quad z(0) = z_0$$

mit der Matrix

$$J = \begin{bmatrix} & I_n \\ -I_n & \end{bmatrix}.$$

Die Matrix J erfüllt die Identitäten $J^\top = -J = J^{-1}$ und definiert die schiefsymmetrische Bilinearform

$$\omega(z_1, z_2) = z_1^\top J z_2.$$

Dieser Ausdruck entspricht einem orientierten Flächeninhalt des von z_1 und z_2 aufgespannten Parallelogramms. Im Fall $n = 1$ beispielsweise gilt $\omega(z_1, z_2) = \det[z_1, z_2]$ für $z_1, z_2 \in \mathbb{R}^2$.

Definition 27.1 Eine Matrix $A \in \mathbb{R}^{2n \times 2n}$ heißt *symplektisch*, falls

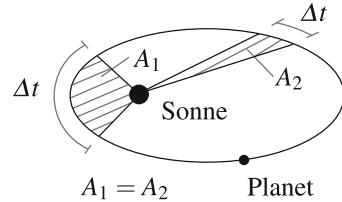
$$\omega(Az_1, Az_2) = \omega(z_1, z_2)$$

für alle $z_1, z_2 \in \mathbb{R}^{2n}$ beziehungsweise $A^\top JA = J$ gilt. Eine differenzierbare Abbildung $\Psi : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ heißt *symplektisch*, wenn ihr Differenzial $D\Psi(z)$ für alle $z \in \mathbb{R}^{2n}$ eine symplektische Matrix ist.

Symplektische Abbildungen erhalten den orientierten Flächeninhalt von Parallelogrammen. Die Symplektizität ist die charakteristische Eigenschaft Hamiltonscher Systeme.

Satz 27.1 Für ein Hamiltonsches System $z' = J \nabla H(z)$, $z(0) = z_0$, mit $H \in C^2(\mathbb{R}^{2n})$ ist der Fluss $\phi_t : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$, $z_0 \mapsto z(t)$, der einer Anfangskonfiguration z_0 den Zustand $z(t)$ zum Zeitpunkt t zuordnet, für jedes $t \in \mathbb{R}$ eine symplektische Abbildung.

Abb. 27.1 Nach dem zweiten Keplerschen Gesetz überstreicht der Fahrstrahl eines Planeten in gleichen Zeitabschnitten gleich große Segmente



Beweis Für die Abbildung $t \mapsto \phi_t(z_0)$ gilt

$$\frac{d}{dt} \phi_t(z_0) = \frac{d}{dt} z(t) = z'(t) = J \nabla H(z(t)) = J \nabla H(\phi_t(z_0)).$$

Das Differenzieren dieser Identität bezüglich z_0 führt auf

$$\frac{d}{dt} D\phi_t(z_0) = JD^2H(\phi_t(z_0))D\phi_t(z_0).$$

Zum Nachweis der Symplektizität betrachten wir $F(t) = D\phi_t(z_0)^\top JD\phi_t(z_0)$ und bemerken, dass aus $\phi_0(z_0) = z_0$ für alle $z_0 \in \mathbb{R}^{2n}$ die Identität $F(0) = J$ folgt. Für die Ableitung gilt damit unter Verwendung der Symmetrie der Hesse-Matrix D^2H , dass

$$\begin{aligned} F'(t) &= \left[\frac{d}{dt} D\phi_t(z_0) \right]^\top J [D\phi_t(z_0)] + [D\phi_t(z_0)]^\top J \left[\frac{d}{dt} D\phi_t(z_0) \right] \\ &= [D\phi_t(z_0)]^\top D^2H(\phi_t(z_0))J^\top J[D\phi_t(z_0)] \\ &\quad + [D\phi_t(z_0)]^\top J^2 D^2H(\phi_t(z_0))[D\phi_t(z_0)] = 0, \end{aligned}$$

wobei $J^\top J = I_{2n} = -J^2$ ausgenutzt wurde. Damit folgt $F(t) = J$ für alle $t \in [0, T]$ beziehungsweise die Symplektizität von ϕ_t . \square

Bemerkung 27.1 Die Keplerschen Gesetze zur Bestimmung von Planetenlaufbahnen lassen sich als Konsequenz der Symplektizität beziehungsweise der Erhaltungseigenschaften Hamiltonscher Systeme interpretieren. Das zweite Keplersche Gesetz beispielsweise postuliert, dass ein von der Sonne zum Planeten gezogener Fahrstrahl in gleichen Zeiten gleich große Flächen überstreicht, s. Abb. 27.1.

27.2 Symplektische Verfahren

Um die Dynamik eines Hamiltonschen Systems sinnvoll wiederzugeben, das heißt die Energie- und Impulserhaltungseigenschaften gut zu approximieren, sollten auch die verwendeten numerischen Verfahren symplektische Abbildungen definieren. Ein Einschritt-

verfahren der Form

$$\begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} q_k \\ p_k \end{bmatrix} + \tau \begin{bmatrix} \Phi_1(t_k, q_k, p_k, q_{k+1}, p_{k+1}, \tau) \\ \Phi_2(t_k, q_k, p_k, q_{k+1}, p_{k+1}, \tau) \end{bmatrix}$$

definiert im Fall der Wohlgestelltheit für $k = 0, 1, \dots, K - 1$ die Abbildungen

$$\Psi^{k+1} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1}).$$

Definition 27.2 Ein numerisches Verfahren heißt *symplektisch*, wenn die dadurch definierten Abbildungen $\Psi^{k+1} : (q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$, $k = 0, 1, \dots, K - 1$, für jede Hamilton-Funktion $H \in C^2(\mathbb{R}^{2n})$ symplektisch sind.

Die Symplektizität eines Verfahrens lässt sich mit dem folgenden Kriterium im Fall $n = 1$ überprüfen.

Lemma 27.1 Eine Abbildung $\Psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ist symplektisch genau dann, wenn $\det D\Psi = 1$ gilt.

Beweis Es gilt

$$D\Psi^\top JD\Psi = \begin{bmatrix} \partial_1 \Psi_1 & \partial_1 \Psi_2 \\ \partial_2 \Psi_1 & \partial_2 \Psi_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \partial_1 \Psi_1 & \partial_2 \Psi_1 \\ \partial_1 \Psi_2 & \partial_2 \Psi_2 \end{bmatrix} = \begin{bmatrix} 0 & \det D\Psi \\ -\det D\Psi & 0 \end{bmatrix},$$

woraus die Behauptung folgt. \square

Wir überprüfen die Symplektizität für einige Standardverfahren.

Beispiele 27.3 (i) Für das explizite Euler-Verfahren gilt $\Psi^{k+1} = \Psi$ mit

$$\begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} = \Psi(q_k, p_k) = \begin{bmatrix} \Psi_1(q_k, p_k) \\ \Psi_2(q_k, p_k) \end{bmatrix} = \begin{bmatrix} q_k \\ p_k \end{bmatrix} + \tau \begin{bmatrix} \partial_p H(q_k, p_k) \\ -\partial_q H(q_k, p_k) \end{bmatrix}$$

und somit

$$\begin{aligned} \partial_1 \Psi_1 &= 1 + \tau \partial_p \partial_q H, & \partial_2 \Psi_1 &= -\tau \partial_p \partial_p H, \\ \partial_1 \Psi_2 &= -\tau \partial_q \partial_q H, & \partial_2 \Psi_2 &= 1 - \tau \partial_p \partial_q H, \end{aligned}$$

sowie $\det D\Psi = 1 + \mathcal{O}(\tau^2)$, sodass das Verfahren nicht symplektisch ist.

(ii) Für das *partitionierte Euler-Verfahren*

$$\begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} q_k \\ p_k \end{bmatrix} + \tau \begin{bmatrix} \partial_p H(q_k, p_{k+1}) \\ -\partial_q H(q_k, p_{k+1}) \end{bmatrix}$$

hängt die rechte Seite von p_{k+1} ab, sodass

$$\begin{aligned}\partial_1 \Psi_1 &= \frac{\partial q_{k+1}}{\partial q_k} = 1 + \tau \partial_q \partial_p H(q_k, p_{k+1}) + \tau \partial_p^2 H(q_k, p_{k+1}) \frac{\partial p_{k+1}}{\partial q_k}, \\ \partial_2 \Psi_1 &= \frac{\partial q_{k+1}}{\partial p_k} = \tau \partial_p^2 H(q_k, p_{k+1}) \frac{\partial p_{k+1}}{\partial p_k}, \\ \partial_1 \Psi_2 &= \frac{\partial p_{k+1}}{\partial q_k} = -\tau \partial_q^2 H(q_k, p_{k+1}) - \tau \partial_q \partial_p H(q_k, p_{k+1}) \frac{\partial p_{k+1}}{\partial q_k}, \\ \partial_2 \Psi_2 &= \frac{\partial p_{k+1}}{\partial p_k} = 1 - \tau \partial_q \partial_p H(q_k, p_{k+1}) \frac{\partial p_{k+1}}{\partial p_k}.\end{aligned}$$

Die letzten beiden Gleichungen lassen sich auflösen und führen auf

$$\begin{aligned}\partial_1 \Psi_2 &= \frac{\partial p_{k+1}}{\partial q_k} = -\tau (1 + \tau \partial_q \partial_p H(q_k, p_{k+1}))^{-1} \partial_q^2 H(q_k, p_{k+1}), \\ \partial_2 \Psi_2 &= \frac{\partial p_{k+1}}{\partial p_k} = (1 + \tau \partial_q \partial_p H(q_k, p_{k+1}))^{-1}.\end{aligned}$$

Damit folgt $\det D\Psi = 1$, sodass das Verfahren symplektisch ist.

- (iii) Das Mittelpunktverfahren ist symplektisch.
- (iv) Das implizite Euler-Verfahren ist nicht symplektisch.

Bemerkung 27.2 Im Fall einer Hamilton-Funktion der Gestalt

$$H(q, p) = \sum_{i=1}^N \frac{\|p_i\|^2}{2m_i} + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N V(\|q_i - q_j\|)$$

lassen sich die durch das partitionierte Euler-Verfahren definierten Gleichungssysteme in jedem Zeitschritt explizit lösen.

Die Vorteile symplektischer Verfahren lassen sich am Beispiel des linearisierten Fadenpendels illustrieren.

Beispiel 27.4 Wir betrachten die Hamilton-Funktion

$$H(q, p) = \frac{1}{2} p^2 + \frac{1}{2} q^2$$

für die die Lösungen des Hamilton-Systems

$$q' = p, \quad p' = -q$$

durch $q(t) = a \sin(t) + b \cos(t)$ und $p(t) = a \cos(t) - b \sin(t)$ gegeben sind. Die Gesamtenergie $H(q(t), p(t))$ jeder Lösung ist dabei konstant. Mit dem Differenzenquotienten $d_t a_{k+1} = (a_{k+1} - a_k)/\tau$ und dem θ -Verfahren

$$\begin{aligned} d_t q_{k+1} &= p_{k+\theta_2} = (1 - \theta_2)p_k + \theta_2 p_{k+1}, \\ d_t p_{k+1} &= -q_{k+\theta_1} = (1 - \theta_1)q_k + \theta_1 q_{k+1}, \end{aligned}$$

lassen sich das explizite und implizite Euler-, das Mittelpunkt- beziehungsweise das partitionierte Euler-Verfahren durch die Wahlen

$$\theta = (0, 0), \quad \theta = (1, 1), \quad \theta = (1/2, 1/2), \quad \theta = (0, 1)$$

beschreiben. Wir verwenden die Formel

$$(a - b)(\theta a + (1 - \theta)b) = \frac{1}{2}(a^2 - b^2) - \frac{1 - 2\theta}{2}(a - b)^2,$$

die man durch Ergänzen von $\pm a/2$ erhält, und multiplizieren die Gleichungen des θ -Verfahrens mit $q_{k+\theta_1}$ beziehungsweise $p_{k+\theta_2}$. Die anschließende Addition der Gleichungen führt in den Fällen des expliziten und impliziten Euler- sowie des Mittelpunktverfahrens auf

$$\frac{1}{2\tau}(q_{k+1}^2 - q_k^2) + \frac{1}{2\tau}(p_{k+1}^2 - p_k^2) = \frac{1 - 2\theta_1}{2\tau}(q_{k+1} - q_k)^2 + \frac{1 - 2\theta_2}{2\tau}(p_{k+1} - p_k)^2.$$

Wir summieren über $k = 0, 1, \dots, \ell - 1$, multiplizieren mit τ und erhalten

$$H(q_\ell, p_\ell) - H(q_0, p_0) = \frac{1 - 2\theta_1}{2} \sum_{k=0}^{\ell-1} (q_{k+1} - q_k)^2 + \frac{1 - 2\theta_2}{2} \sum_{k=0}^{\ell-1} (p_{k+1} - p_k)^2.$$

Im Fall des expliziten Euler-Verfahrens ist die rechte Seite im Allgemeinen positiv und es kommt zu einem Anwachsen der Gesamtenergie, während es im Fall des impliziten Euler-Verfahrens zu einer Abnahme kommt. Für das Mittelpunktverfahren verschwindet die rechte Seite und die Energie wird exakt erhalten. Für das partitionierte Euler-Verfahren ergibt die Multiplikation der Gleichungen mit $q_{k+1/2}$ und $p_{k+1/2}$, dass

$$H(q_\ell, p_\ell) - H(q_0, p_0) = -\tau p_\ell q_\ell + \tau p_0 q_0.$$

Mit $\tau|pq| \leq \tau(p^2 + q^2)/2$ folgt

$$\frac{1 - \tau}{1 + \tau} H(q_0, p_0) \leq H(q_\ell, p_\ell) \leq \frac{1 + \tau}{1 - \tau} H(q_0, p_0).$$

Die Ergebnisse entsprechender numerischer Experimente sind in Abb. 27.2 gezeigt.

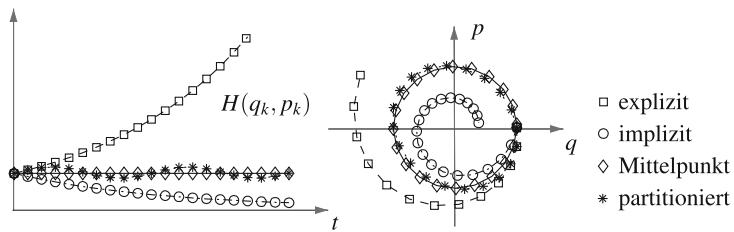


Abb. 27.2 Anwendung verschiedener Verfahren auf ein Hamiltonsches System; symplektische Verfahren wie das Mittelpunkt- und das partitionierte Euler-Verfahren erhalten physikalisch relevante Größen

27.3 Schießverfahren

Bei eindimensionalen *Randwertproblemen* ist eine Funktion $u : [a, b] \rightarrow \mathbb{R}$ gesucht, die eine Differenzialgleichung im Innern des Intervalls und Randbedingungen an beiden Intervallenden erfüllt. Ein eindimensionales Randwertproblem zweiter Ordnung lautet beispielsweise

$$\begin{aligned} u''(x) &= f(x, u(x), u'(x)), \quad x \in (a, b), \\ u(a) &= \alpha, \quad u(b) = \beta. \end{aligned}$$

Damit kann die Flugbahn eines Balls beschrieben werden, der am Ort a auf der Höhe α so abgeworfen wird, dass er am Ort b die Höhe β hat. Eindimensionale Randwertprobleme lassen sich mit den für Anfangswertprobleme konstruierten numerischen Verfahren iterativ lösen. Dazu suchen wir im obigen Modellproblem einen Parameter $s \in \mathbb{R}$, sodass die Lösung $y : [a, b] \rightarrow \mathbb{R}$ des Anfangswertproblems

$$\begin{aligned} y''(x) &= f(x, y(x), y'(x)), \quad x \in (a, b), \\ y(a) &= \alpha, \quad y'(a) = s \end{aligned}$$

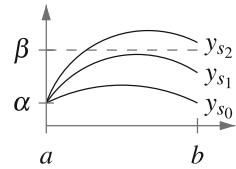
die Eigenschaft $y(b) = \beta$ besitzt und somit das Randwertproblem erfüllt. Da y vom Parameter s abhängt, schreiben wir im Folgenden y_s für die Lösung des Anfangswertproblems. Anschaulich ist die gesuchte Zahl $s \in \mathbb{R}$ der Abwurfwinkel, der notwendig ist, um die Höhe β am Ort b zu realisieren. Wir definieren die Abbildung

$$F : \mathbb{R} \rightarrow \mathbb{R}, \quad s \mapsto y_s(b) - \beta$$

und versuchen, eine Nullstelle s^* von F zu bestimmen. Mit dem Newton-Verfahren geschieht dies für einen Startwert s_0 approximativ durch die Iteration

$$s_{i+1} = s_i - \frac{F(s_i)}{F'(s_i)},$$

Abb. 27.3 Verschiedene Anfangsgeschwindigkeiten s_i führen zu unterschiedlichen Werten am finalen Zeitpunkt



wobei $F(s) = y_s(b) - \beta$ und $F'(s) = \partial_s y_s(b)$ gilt. Das sogenannte *Schießverfahren* ist in Abb. 27.3 illustriert.

Die Funktion $v(x) = \partial_s y_s(x)$ ist für gegebene $s \in \mathbb{R}$ und y_s die Lösung des nach s differenzierten Anfangswertproblems, das heißt

$$\begin{aligned} v''(x) &= f(x, y_s(x), y'_s(x))v(x) + f(x, y_s(x), y'_s(x))v'(x), \quad x \in (a, b), \\ v(a) &= 0, \quad v'(a) = 1. \end{aligned}$$

Die Funktion v ist also Lösung eines linearen Anfangswertproblems, das mit geringem Aufwand gelöst werden kann. Um Konvergenz des Newton-Verfahrens zu erhalten, muss der Startwert s_0 im Allgemeinen nahe genug an s^* liegen.

27.4 Diskontinuierliche Galerkin-Verfahren

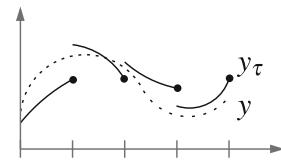
Wir multiplizieren die autonome Differenzialgleichung $y' = f(y)$ mit einer Funktion ϕ , integrieren das Produkt über das Intervall $[t_k, t_{k+1}]$ und führen eine partielle Integration durch, sodass wir die Identität

$$-\int_{t_k}^{t_{k+1}} y(t)\phi'(t) dt + y(t_k^-)\phi(t_k^-) - y(t_k^-)\phi(t_k^+) = \int_{t_k}^{t_{k+1}} f(y(t))\phi(t) dt$$

erhalten, wobei $g(t_m^\pm)$ die Grenzwerte $\lim_{\varepsilon \rightarrow 0} g(t_m \pm \varepsilon)$ für $\varepsilon > 0$ bezeichne. Dabei wurde ausgenutzt, dass y stetig ist, also $y(t_k^+) = y(t_k^-)$ gilt. Die Idee der *diskontinuierlichen Galerkin-Methode* besteht darin, unstetige Approximationen $y_\tau : [0, T] \rightarrow \mathbb{R}$ zu betrachten, und die obige Umformulierung teilweise rückgängig zu machen, um eine definierende Gleichung für y_τ herzuleiten. Wir ersetzen y durch y_τ in der obigen Gleichung und verwenden

$$-\int_{t_k}^{t_{k+1}} y_\tau(t)\phi'(t) dt = \int_{t_k}^{t_{k+1}} y'_\tau(t)\phi(t) dt + y_\tau(t_k^+)\phi(t_k^+) - y_\tau(t_{k+1}^-)\phi(t_{k+1}^-).$$

Abb. 27.4 Diskontinuierliche Galerkin-Verfahren approximieren die Lösung durch unstetige, stückweise polynomiale Funktionen



Dies führt auf die Integralgleichung

$$\int_{t_k}^{t_{k+1}} y'_\tau(t)\phi(t) dt + [y_\tau(t_k^+) - y_\tau(t_k^-)]\phi(t_k^+) = \int_{t_k}^{t_{k+1}} f(y_\tau(t))\phi(t) dt,$$

wobei $y_\tau(t_0^-) = y_0$ sei. Die Approximationslösung wird nun als stückweises Polynom $y_\tau|_{(t_k, t_{k+1})} \in \mathcal{P}_\ell|_{(t_k, t_{k+1})}$ gesucht, sodass die Integralgleichung für alle $k = 0, 1, \dots, K-1$ und alle $\phi \in \mathcal{P}_\ell|_{(t_k, t_{k+1})}$ gilt, s. Abb. 27.4.

Beispiel 27.5 Für $\ell = 0$ ergibt sich das implizite Euler-Verfahren, denn setzen wir $y_k = y_\tau|_{(t_k, t_{k+1})}$, so folgt unter Verwendung von $y_\tau'|_{(t_k, t_{k+1})} = 0$ und mit $\phi = 1$, dass $y_{k+1} - y_k = (t_{k+1} - t_k)f(y_{k+1})$ gilt.

27.5 Lernziele, Quiz und Anwendung

Sie sollten Hamiltonsche Systeme definieren und die Bedeutung symplektischer Verfahren erklären können. Schießverfahren sollten Sie motivieren und deren algorithmische Umsetzung erläutern können. Charakteristische Eigenschaften diskontinuierlicher Galerkin-Verfahren sollten Sie aufzeigen können.

Quiz 27.1 Entscheiden Sie für jede der folgenden Aussagen, ob diese wahr oder falsch ist. Sie sollten Ihre Antwort begründen können.

Aussage	Beurteilung
Hamiltonsche Systeme sind spezielle Gradientenflüsse	
Orthogonale Matrizen sind symplektisch	
Das partitionierte Euler-Verfahren besitzt die Konsistenzordnung $p = 2$	
Jedes Randwertproblem lässt sich in eindeutiger Weise als Anfangswertproblem formulieren	
Die Approximationslösung des diskontinuierlichen Galerkin-Verfahrens ist eine unstetige Funktion	

Tab. 27.1 Daten zur Simulation eines einfachen Sonnensystems

Planet	Masse	Anfangsposition	Anfangsgeschwindigkeit
Sonne	1	(0, 0, 0)	$(0, 0, 0) \cdot 10^{-3}$
Jupiter	1/1000	(−3, −4, −1)	$(5, -4, -2) \cdot 10^{-3}$
Saturn	3/10000	(10, −3, −2)	$(2, 5, 2) \cdot 10^{-3}$

Anwendung 27.1 Zur Simulation des äußeren Sonnensystems verwenden wir die Hamilton-Funktion

$$H(q, p) = \sum_{i=1}^N \frac{\|p_i\|^2}{2m_i} - \frac{\gamma}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N \frac{m_i m_j}{\|q_i - q_j\|}$$

mit den Impulsen $p_i \in \mathbb{R}^3$ und Positionen $q_i \in \mathbb{R}^3$, $i = 1, 2, \dots, N$, der betrachteten Planeten. Verwenden Sie das resultierende Hamiltonsche System und die in Tab. 27.1 angegebenen Approximationen in Sonnenmassen $SM \approx 2 \cdot 10^{30}$ kg und astronomischen Einheiten $AU \approx 150 \cdot 10^9$ m beziehungsweise AU/day zur Beschreibung von drei Planeten.

Benutzen Sie als Gravitationskonstante die einfache Näherung der heliozentrischen Gravitationskonstanten $\gamma = 3 \cdot 10^{-4} AU^3 / (SM \cdot day^2)$. Simulieren Sie das System numerisch durch Verwendung des expliziten und des partitionierten Euler-Verfahrens mit verschiedenen Schrittweiten. Stellen Sie die Laufbahnen der Planeten grafisch dar und betrachten Sie die Gesamtenergie des Systems als Funktion der Zeit. Bestimmen Sie experimentell die Länge eines Jupiter-Jahres.

Teil IV

Aufgabensammlungen

28.1 Grundlegende Konzepte

Aufgabe 28.1.1 Sei $\tilde{\phi} = f \circ g$ ein Verfahren für die mathematische Aufgabe ϕ und sei die durch g definierte Aufgabe schlecht konditioniert. Zeigen Sie, dass das Verfahren $\tilde{\phi}$ im Allgemeinen instabil ist.

Aufgabe 28.1.2 Zeigen Sie, dass die Addition zweier nichtnegativer oder nichtpositiver Zahlen gut konditioniert ist.

Aufgabe 28.1.3 Für $p > 0$, $\beta > 1$ und $j = 1, 2, 3, 4$ seien die Folgen $(a_n^{(j)})_{n \in \mathbb{N}}$ definiert durch

$$a_n^{(1)} = n^p, \quad a_n^{(2)} = \beta^n, \quad a_n^{(3)} = n!, \quad a_n^{(4)} = \log_2 n.$$

Für welche Paare $1 \leq i, j \leq 4$ gilt $a_n^{(i)} = \mathcal{O}(a_n^{(j)})$?

Aufgabe 28.1.4 Unter welchen Bedingungen an $a, b, c, d \in \mathbb{R}$ ist die Berechnung eines Schnittpunkts der beiden Geraden $x \mapsto ax + b$ und $x \mapsto cx + d$ ein gut konditioniertes Problem?

Aufgabe 28.1.5 Wie lassen sich Auslöschungseffekte bei der praktischen Berechnung der Ausdrücke

$$\frac{1 - 2x}{1 + 2x} - \frac{1}{1 + x}, \quad \frac{e^x - 1}{x}$$

für $x \neq 0$ mit $|x| \ll 1$ vermeiden?

Aufgabe 28.1.6 Diskutieren Sie die Konditionierung der Bestimmung von Nullstellen einer quadratischen Gleichung $x^2 + px + q = 0$ sowie die Stabilität ihrer Berechnung mit der pq -Formel $x_{1,2} = -p/2 \pm (p^2/4 - q)^{1/2}$. Betrachten Sie besonders die Fälle $p^2 \approx 4q$ und $p^2 \gg 4|q|$.

Aufgabe 28.1.7 Für welche $x \in \mathbb{R}$ muss bei der approximativen Berechnung $e^x \approx \sum_{k=0}^n x^k / k!$ mit Auslöschungseffekten gerechnet werden. Wie lassen sich diese vermeiden?

Aufgabe 28.1.8 Bestimmen Sie die Größenordnung des Aufwands für die Matrix-Vektor-Multiplikation, die Matrix-Matrix-Multiplikation sowie die Bestimmung der Determinante einer Matrix mit dem Laplaceschen Entwicklungssatz.

Aufgabe 28.1.9 Ein Rechner arbeite mit 10^9 Gleitkommaoperationen pro Sekunde beziehungsweise 10^9 FLOPS (*floating point operations per second*) und es seien drei Algorithmen mit Aufwand $\mathcal{O}(n)$, $\mathcal{O}(n^3)$ beziehungsweise $\mathcal{O}(n!)$ zur Lösung derselben Aufgabe gegeben. Wieviele Sekunden, Stunden, Tage oder Jahre benötigen die Algorithmen etwa für die Problemgrößen $n = 10^k$ mit $k = 1, 2, \dots, 6$?

Aufgabe 28.1.10 Sei $\phi(0)$ eine gut konditionierte Aufgabe mit $\phi'(0) = 3$. Untersuchen Sie für gegebene $x_1, x_2 \in \mathbb{R}$ die Konditionierung der Aufgabe $\phi(x_1 + x_2)$.

Projekt 28.1.1 Die Funktionen $f, g : \mathbb{R}_{>0} \rightarrow \mathbb{R}$ definiert durch

$$f(x) = \frac{1}{x} - \frac{1}{x+1}, \quad g(x) = \frac{1}{x(x+1)}$$

stimmen überein, motivieren aber zwei unterschiedliche Verfahren zur numerischen Berechnung. Bestimmen Sie für $x_k = 10^k$, $k = 1, 2, \dots, 15$, den Ausdruck

$$\delta_k = \frac{|f(x_k) - g(x_k)|}{|g(x_k)|}$$

in MATLAB und tragen Sie die Ergebnisse in eine Tabelle ein. Was beobachten Sie und wie erklären Sie die Beobachtungen?

Projekt 28.1.2 Implementieren Sie die rekursive Berechnung der Determinante einer quadratischen Matrix mit dem Laplaceschen Entwicklungssatz in MATLAB und C. Messen Sie von Hand oder mit Hilfe der Befehle `tic ... toc` beziehungsweise `clock()` die Laufzeiten für die Berechnung von $\det A$ mit der Matrix $A \in \mathbb{R}^{n \times n}$ definiert durch $a_{ii} = 2$ und $a_{ij} = (-1)^j / (n-1)$ sowie $n = 10, 20, 40, 80$.

28.2 Operatornorm und Konditionszahl

Aufgabe 28.2.1 Zu fixierten Normen $\|\cdot\|$ auf \mathbb{R}^n und auf \mathbb{R}^m bezeichne $\|\cdot\|_{op}$ die induzierte Operatornorm auf $\mathbb{R}^{m \times n}$. Beweisen Sie die folgenden Aussagen:

- (i) Die Operatornorm $\|\cdot\|_{op}$ definiert eine Norm auf $\mathbb{R}^{m \times n}$.
- (ii) Es gilt

$$\|A\|_{op} = \sup_{\|x\|=1} \|Ax\| = \inf \{c > 0 : \forall x \in \mathbb{R}^n \|Ax\| \leq c\|x\|\}$$

und das Supremum und das Infimum werden angenommen.

- (iii) Im Fall $A \neq 0$ folgt für $x \in \mathbb{R}^m$ mit $\|x\| \leq 1$ und $\|Ax\| = \|A\|_{op}$ bereits $\|x\| = 1$.
- (iv) Zeigen Sie, dass die Spektralnorm kleiner ist als jede andere Operatornorm auf dem Raum der quadratischen Matrizen.

Aufgabe 28.2.2 Für $1 \leq p < \infty$ wird auf \mathbb{R}^ℓ durch $\|x\|_p = \left(\sum_{j=1}^{\ell} |x_j|^p\right)^{1/p}$ eine Norm definiert. Die induzierte Operatornorm sei ebenfalls mit $\|\cdot\|_p$ bezeichnet.

- (i) Zeigen Sie, dass $\|A\|_1 = \max_{k=1,\dots,n} \sum_{j=1}^m |a_{jk}|$ für alle $A \in \mathbb{R}^{m \times n}$ gilt.
- (ii) Für die symmetrische Matrix $B \in \mathbb{R}^{n \times n}$ sei

$$\rho(B) = \max\{|\lambda| : \lambda \text{ ist Eigenwert von } B\}.$$

Zeigen Sie, dass $\|A\|_2 = \sqrt{\rho(A^\top A)}$ für alle $A \in \mathbb{R}^{m \times n}$ gilt.

Aufgabe 28.2.3 Zeigen Sie, dass durch $\|A\|_G = \max_{1 \leq i,j \leq n} |a_{ij}|$ eine Norm jedoch keine Operatornorm auf $\mathbb{R}^{n \times n}$ definiert wird.

Aufgabe 28.2.4 Sei $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ mit $a, b, c \in \mathbb{R}$, sodass $\det A \neq 0$ gilt. Bestimmen Sie $\text{cond}_1(A)$, $\text{cond}_2(A)$ und $\text{cond}_\infty(A)$ und diskutieren Sie, für welche Verhältnisse von a , b und c zugehörige lineare Gleichungssysteme schlecht konditioniert sind.

Aufgabe 28.2.5 Sei $A \in \mathbb{R}^{n \times n}$ invertierbar und $\|\cdot\|$ eine von der Vektornorm $\|\cdot\|$ induzierte Operatornorm auf $\mathbb{R}^{n \times n}$. Zeigen Sie, dass

$$\|A^{-1}\| = \left(\inf_{\|x\|=1} \|Ax\| \right)^{-1}$$

und $\|A^{-1}\| \geq \|A\|^{-1}$ gelten.

Aufgabe 28.2.6

- (i) Sei $A \in \mathbb{R}^{n \times n}$. Zeigen Sie, dass

$$\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty$$

gilt und überprüfen Sie die Aussage explizit für $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$.

- (ii) Zeigen Sie, dass für jede Matrix $A \in \mathbb{R}^{n \times n}$ die Abschätzungen

$$\begin{aligned} n^{-1/2} \|A\|_2 &\leq \|A\|_1 \leq n^{1/2} \|A\|_2, \\ n^{-1} \|A\|_\infty &\leq \|A\|_1 \leq n \|A\|_\infty \end{aligned}$$

gelten und geben Sie Matrizen $A \in \mathbb{R}^{n \times n}$ an, die zeigen, dass sich die Abschätzungen nicht verbessern lassen.

Aufgabe 28.2.7 Für $A \in \mathbb{R}^{n \times n}$ ist die Frobeniusnorm definiert durch $\|A\|_F^2 = \sum_{1 \leq i, j \leq n} a_{ij}^2$. Zeigen Sie, dass

$$\|A\|_F = \sqrt{\text{tr}(A^\top A)}.$$

Folgern Sie, dass die Frobeniusnorm mit der von der Euklidischen Norm induzierten Operatornorm verträglich ist in dem Sinne, dass

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n} \|A\|_2.$$

Verwenden Sie dazu die Identität $\text{tr}(A^\top A) = \lambda_1 + \dots + \lambda_n$ mit den nichtnegativen Eigenwerten $\lambda_1, \dots, \lambda_n$ von $A^\top A$. Lassen sich die Abschätzungen auch ohne Verwendung der Eigenwerte beweisen?

Aufgabe 28.2.8 Sei $A \in \mathbb{R}^{m \times n}$.

- (i) Zeigen Sie, dass $(\text{Im } A^\top)^\perp = \ker A$ mit

$$V^\perp = \{v \in \mathbb{R}^n : v \cdot w = 0 \text{ für alle } w \in V\}$$

für $V \subset \mathbb{R}^n$ und folgern Sie $\mathbb{R}^n = \text{Im } A^\top + \ker A$.

- (ii) Beweisen Sie die Dimensionsformel $n = \dim(\text{Im } A) + \dim(\ker A)$ und folgern Sie, dass $\text{rank } A = \text{rank } A^\top$, wobei für eine Matrix M der Spaltenrang von M durch $\text{rank } M = \dim \text{Im } M$ definiert sei.
 (iii) Zeigen Sie, dass

$$\ker A^\top A = \ker A.$$

Aufgabe 28.2.9

- (i) Seien $A \in \mathbb{R}^{n \times m}$ und $B \in \mathbb{R}^{m \times p}$. Mit natürlichen Zahlen $n_1, n_2, m_1, m_2, p_1, p_2$ seien $A_{ij} \in \mathbb{R}^{n_i \times m_j}, B_{jk} \in \mathbb{R}^{m_j \times p_k}$, sodass

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

gilt. Bestimmen Sie Matrizen $C_{ik} \in \mathbb{R}^{n_i \times p_k}$, sodass eine entsprechende Partitionierung auch für $C = AB$ gilt.

- (ii) Zeigen Sie, dass für jede reguläre Matrix $A \in \mathbb{R}^{n \times n}$ die Identität $(A^\top)^{-1} = (A^{-1})^\top$ gilt, welche die Schreibweise $A^{-\top}$ rechtfertigt.

Aufgabe 28.2.10 Sei $A \in \mathbb{R}^{n \times n}$ regulär und $1 \leq m \leq n$, sodass die obere linke $m \times m$ -Teilmatrix $A_{11} = (a_{ij})_{1 \leq i,j \leq m}$ ebenfalls regulär ist. Sei A zerlegt gemäß

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Zeigen Sie, dass $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ regulär ist und dass A^{-1} geben ist durch

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{bmatrix}.$$

Projekt 28.2.1 Schreiben Sie Programme in C und MATLAB, die die Operatornorm $\|\cdot\|_\infty$ einer Matrix $A \in \mathbb{R}^{m \times n}$ berechnen. Messen Sie von Hand oder mit Hilfe der Befehle `clock()` beziehungsweise `tic ... toc` für die Hilbert-Matrix $H \in \mathbb{R}^{n \times n}$ mit Einträgen $h_{ij} = 1/(i+j-1)$ $1 \leq i, j \leq n$ die Laufzeiten der Programme für $n = 10^k$, $k = 1, 2, \dots, 4$. Vergleichen Sie Ihre Programme zudem mit der Laufzeit der MATLAB-Routine `norm(H, inf)`.

Projekt 28.2.2 Die Menge $N_2(1) = \{x \in \mathbb{R}^2 : \|x\|_2 = 1\}$ lässt sich in MATLAB approximativ darstellen mittels `plot(X, Y, '-b')`, wobei `Phi=(0:dphi:2*pi)` und `X=cos(Phi), Y=sin(Phi)` beispielsweise für `dphi=0.01` seien. Plotten Sie die deformierte Menge $A(N_2(1))$ für Matrizen

$$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}, \quad \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \quad \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} c' & s' \\ s & c' \end{bmatrix}$$

mit geeigneten Zahlen $k, k_1, k_2 \in \mathbb{R}, c = \cos(\theta), s = \sin(\theta)$ für $\theta \in [0, 2\pi]$. Mit den Kommandos `hold on/off` können Sie die Mengen in einem Grafikfenster und durch Veränderung des Arguments `'-b'` mit unterschiedlichen Farben darstellen. Ersetzen Sie anschließend $N_2(1)$ durch $N_1(1)$ und $N_\infty(1)$.

28.3 Matrixfaktorisierungen

Aufgabe 28.3.1 Sei $A \in \mathbb{R}^{n \times n}$ eine positiv definite Matrix, das heißt es gelte $x^\top A x > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$.

- (i) Zeigen Sie, dass A regulär ist.
- (ii) Zeigen Sie, dass für alle $1 \leq k \leq n$ die $k \times k$ -Untermatrix $A_k = (a_{ij})_{1 \leq i,j \leq k}$ ebenfalls positiv definit ist.
- (iii) Zeigen Sie, dass alle reellen Eigenwerte von A positiv sind.

Aufgabe 28.3.2 Sei $A \in \mathbb{R}^{n \times n}$ eine strikt diagonaldominante Matrix, das heißt es gelte

$$\sum_{j=1,\dots,n, j \neq i} |a_{ij}| < |a_{ii}|, \quad i = 1, 2, \dots, n.$$

- (i) Zeigen Sie, dass die Teilmatrizen $A_k = (a_{ij})_{1 \leq i,j \leq k}$ für $k = 1, 2, \dots, n$ ebenfalls strikt diagonaldominant sind.
- (ii) Zeigen Sie, dass die Matrix A regulär ist.
Hinweis: Zeigen Sie zum Nachweis von (ii), dass für eine geeignete Norm $\|\cdot\|$ auf \mathbb{R}^n die Abschätzung $\|Ax\| > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$ gilt, und folgern Sie daraus, dass A injektiv ist.

Aufgabe 28.3.3 Zeigen Sie, dass die invertierbaren (normalisierten) unteren Dreiecksmatrizen eine Gruppe bilden, das heißt sind $L, L_1, L_2 \in \mathbb{R}^{n \times n}$ (normalisierte) untere Dreiecksmatrizen und gilt $\det L \neq 0$, so sind L^{-1} und $L_1 L_2$ ebenfalls (normalisierte) untere Dreiecksmatrizen.

Aufgabe 28.3.4 Seien $A \in \mathbb{R}^{n \times n}$, eine untere Dreiecksmatrix L sowie eine obere Dreiecksmatrix U mit $A = LU$ gegeben. Zeigen Sie, dass für $k = 1, 2, \dots, n$ und die linken, oberen $k \times k$ -Teilmatrizen A_k, L_k und U_k von A, L beziehungsweise U ebenfalls die Zerlegung $A_k = L_k U_k$ gilt.

Aufgabe 28.3.5

- (i) Zeigen Sie, dass $A_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ keine normalisierte LU -Zerlegung und $A_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ keine Cholesky-Zerlegung besitzt.

- (ii) Berechnen Sie die normalisierte LU -Zerlegung von A_3 und die Cholesky-Zerlegung von A_4 mit

$$A_3 = \begin{bmatrix} 5 & 3 & 1 \\ 10 & 8 & 8 \\ 15 & 11 & 10 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 9 & 12 & 9 \\ 12 & 41 & 22 \\ 9 & 22 & 38 \end{bmatrix},$$

sofern diese existieren.

Aufgabe 28.3.6 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit.

- (i) Zeigen Sie, dass eine eindeutig bestimmte normalisierte untere Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ und eine Diagonalmatrix $D \in \mathbb{R}^{n \times n}$ mit positiven Diagonaleinträgen existieren, sodass $A = LDL^\top$ gilt.
(ii) Entwickeln Sie ein Verfahren zur Bestimmung von L und D , das die Verwendung der Wurzelfunktion vermeidet, und bestimmen Sie die Matrizen L und D für

$$A = \begin{bmatrix} 9 & 12 & 9 \\ 12 & 41 & 22 \\ 9 & 22 & 38 \end{bmatrix}.$$

Aufgabe 28.3.7 Sei (v_1, v_2, \dots, v_n) eine Basis des \mathbb{R}^n .

- (i) Zeigen Sie, dass die durch $g_{ij} = v_i \cdot v_j$ definierte Matrix $G \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit ist.
(ii) Zeigen Sie, dass G invertierbar ist und G^{-1} ebenfalls symmetrisch und positiv definit ist.
(iii) Konstruieren Sie eine untere Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$, sodass für $W = LV$ die Identität $W^\top W = I_n$ gilt, wobei $V = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{n \times n}$ sei.

Aufgabe 28.3.8 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch mit nichtnegativen Eigenwerten. Konstruieren Sie eine symmetrische Matrix $B \in \mathbb{R}^{n \times n}$ mit $A = B^2 = BB$ und zeigen Sie, dass $\text{cond}_2(B) = \text{cond}_2(A)^{1/2}$, sofern A regulär ist.

Aufgabe 28.3.9 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Zeigen Sie, dass $\lambda_{\max}(A^{-1}) = 1/\lambda_{\min}(A)$ gilt.

Aufgabe 28.3.10

- (i) Wie lässt sich die LU -Zerlegung im Fall symmetrischer Matrizen vereinfachen und welcher Aufwand ergibt sich?
(ii) Sei $A \in \mathbb{R}^{n \times n}$ eine Bandmatrix mit Bandweite m , das heißt es gelte $a_{ij} = 0$ falls $|i - j| > m$. Wie groß ist der Aufwand der Berechnung der LU -Zerlegung, sofern diese existiert?

Projekt 28.3.1 Schreiben Sie ein C-Programm mit Funktionen `solve_upper` und `solve_lower` zur Lösung linearer Gleichungssysteme mit regulärer oberer beziehungsweise unterer Dreiecksmatrix. Die Lösungen von $Ux = b$ und $Lx = b$ sind dabei mit rückwärts beziehungsweise vorwärts laufenden Schleifen gegeben durch

$$x_j = \left(b_j - \sum_{k=j+1}^n u_{jk} x_k \right) / u_{jj}, \quad x_j = \left(b_j - \sum_{k=1}^{j-1} \ell_{jk} x_k \right) / \ell_{jj},$$

wobei die leere Summe den Wert Null habe. Testen Sie die Routinen für die Gleichungssysteme $A_\ell x = b_\ell$, $\ell = 1, 2$, mit

$$A_1 = \begin{bmatrix} 1 & 2 & 3 \\ & 4 & 5 \\ & & 6 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 6 \\ 9 \\ 6 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & & & \\ 2 & 3 & & \\ 4 & 5 & 6 & \end{bmatrix}, \quad b_2 = \begin{bmatrix} 3 \\ 12 \\ 28 \end{bmatrix}.$$

Projekt 28.3.2 Schreiben Sie ein C-Programm, das für eine LU -zerlegbare Matrix $A \in \mathbb{R}^{n \times n}$ ihre LU -Zerlegung bestimmt. Begründen Sie, warum die Einträge der Matrix A mit den berechneten Einträgen von L überschrieben werden können, sodass keine neuen Felder initialisiert werden müssen. Unter welchen Umständen sollte man die Berechnung von L abbrechen? Testen Sie die Implementierung mit den Matrizen

$$A_1 = \begin{bmatrix} 4 & 2 & 3 \\ 2 & 4 & 2 \\ 3 & 2 & 4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

um die Gleichungssystem $A_i x = b_i$, $i = 1, 2$ für $b_1 = [1, 1, 1]^\top$ und $b_2 = [1, \dots, 1]^\top$ zu lösen, wobei $A_2 \in \mathbb{R}^{n \times n}$ und $b_2 \in \mathbb{R}^n$ mit $n = 10, 20, 40, 80$ gelte. Überprüfen Sie Ihre Ergebnisse mit Hilfe der MATLAB-Befehle `l\u(A)` und `x=A\b`. Was lässt sich über die Laufzeit für die Lösung des Gleichungssystems $A_2 x = b_2$ in Abhängigkeit von n aussagen?

Projekt 28.3.3 Schreiben Sie ein C-Programm, das für eine gegebene symmetrische, positiv definite Matrix $A = (a_{ij})_{i,j=1,\dots,n} \in \mathbb{R}^{n \times n}$ ihre Cholesky-Zerlegung $A = LL^\top$ berechnet. Begründen Sie, warum die Einträge der Matrix A mit den berechneten Einträgen von L überschrieben werden können, sodass keine neuen Felder initialisiert werden müssen. Unter welchen Umständen sollte man die Berechnung von L abbrechen? Testen

Sie die Implementierung mit den Matrizen

$$A_1 = \begin{bmatrix} 4 & 2 & 3 \\ 2 & 4 & 2 \\ 3 & 2 & 4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

um die Gleichungssystem $A_i x = b_i$, $i = 1, 2$, für $b_1 = [1, 1, 1]^\top$ und $b_2 = [1, \dots, 1]^\top$ zu lösen, wobei $A_2 \in \mathbb{R}^{n \times n}$ und $b_2 \in \mathbb{R}^n$ mit $n = 10, 20, 40, 80$ gelte. Überprüfen Sie Ihre Ergebnisse mit Hilfe der MATLAB-Befehle `chol(A)` und `x=A\b`. Was lässt sich über die Laufzeit für die Lösung des Gleichungssystems $A_2 x = b_2$ in Abhängigkeit von n aussagen?

Projekt 28.3.4 Für $m \in \mathbb{N}$ und $n = m^2$ seien $B_m \in \mathbb{R}^{m \times m}$ und $A_n \in \mathbb{R}^{n \times n}$ definiert durch

$$A_n = \begin{bmatrix} B_m & -I_m & & \\ -I_m & \ddots & \ddots & \\ & \ddots & \ddots & -I_m \\ & & -I_m & B_m \end{bmatrix}, \quad B_m = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}.$$

Verwenden Sie die MATLAB-Routinen `chol` und `lu`, um Cholesky- und LU-Zerlegungen $L_n L_n^\top = A_n$ und $M_n U_n = A_n$ zu bestimmen und betrachten Sie die Fehler

$$\|A_n - L_n L_n^\top\|_\infty, \quad \|A_n - M_n U_n\|_\infty,$$

die Sie mit `norm(B, inf)` bestimmen können, für $n = 10^k$, $k = 1, 2, \dots, 6$.

28.4 Eliminationsverfahren

Aufgabe 28.4.1 Konstruieren Sie eine Permutationsmatrix $P \in \mathbb{R}^{4 \times 4}$, sodass die Matrix PA eine normalisierte LU -Zerlegung besitzt, wobei

$$A = \begin{bmatrix} -1 & 2 & 3 & 3 \\ 1 & -4 & -2 & -5 \\ 0 & -4 & 0 & -3 \\ -1 & 10 & -5 & 17 \end{bmatrix}.$$

Lösen Sie damit das lineare Gleichungssystem $Ax = b$ mit $b = [17, -23, -13, 51]^\top$.

Aufgabe 28.4.2 Verwenden Sie das Gaußsche Eliminationsverfahren ohne Pivotsuche zur Lösung des linearen Gleichungssystems $Ax = b$ mit

$$A = \begin{bmatrix} -1 & 16 & -4 & 3 \\ -3 & 20 & -22 & 0 \\ 1 & -16 & 1 & -2 \\ 3 & -6 & 4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} -24 \\ -45 \\ 20 \\ 11 \end{bmatrix}.$$

Bestimmen Sie die LU -Zerlegung von A und berechnen Sie $\det A$.

Aufgabe 28.4.3 Seien

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & -1 & 1 \\ 2 & 2 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 7 \\ 14 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} 5.5 \\ 6.5 \\ 14.5 \end{bmatrix}.$$

Berechnen Sie A^{-1} , $\text{cond}_\infty(A)$ und die Lösungen von $Ax = b$ sowie $A\tilde{x} = \tilde{b}$.

Aufgabe 28.4.4 Es sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische und positiv definite Matrix und $b^{(1)}, b^{(2)}, \dots, b^{(m)} \in \mathbb{R}^n$ verschiedene rechte Seiten. Es sei $A = LL^\top$ die Cholesky-Zerlegung von A mit der unteren Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$. Vergleichen Sie den Aufwand der folgenden beiden Vorgehensweisen zur Lösung der m linearen Gleichungssysteme $Ax^{(i)} = b^{(i)}$, $i = 1, 2, \dots, m$:

- (i) Durch Lösen der n linearen Gleichungssysteme $Az^{(j)} = e_j$ mit der Cholesky-Zerlegung von A für die kanonischen Basisvektoren $e_j \in \mathbb{R}^n$ werde die Inverse $A^{-1} = [z^{(1)}, z^{(2)}, \dots, z^{(n)}]$ bestimmt und anschließend $x^{(i)} = A^{-1}b^{(i)}$ für $i = 1, 2, \dots, m$ mittels Matrix-Vektor-Multiplikation bestimmt.
- (ii) Mit der Cholesky-Zerlegung von A werden die Lösungen von $Ax^{(i)} = b^{(i)}$ für $i = 1, 2, \dots, m$ bestimmt.

Aufgabe 28.4.5 Sei $P \in \mathbb{R}^{n \times n}$ die zur Bijektion $\pi : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ gehörende Permutationsmatrix. Zeigen Sie, dass $P^\top = P^{-1}$ und

$$P^{-1} = [e_{\pi^{-1}(1)}, e_{\pi^{-1}(2)}, \dots, e_{\pi^{-1}(n)}].$$

Aufgabe 28.4.6 Wie unterscheidet sich der Aufwand des Gaußschen Eliminationsverfahrens mit Spalten-Pivotsuche von dem mit totaler Pivotsuche?

Aufgabe 28.4.7 Zeigen Sie, dass mit den kanonischen Basisvektoren $e_1, e_2, \dots, e_m \in \mathbb{R}^m$ und $f_1, f_2, \dots, f_n \in \mathbb{R}^n$ für $A \in \mathbb{R}^{m \times n}$ gilt, dass

$$A = \sum_{i=1}^m \sum_{j=1}^n a_{ij} e_i f_j^\top.$$

Aufgabe 28.4.8 Sei $P \in \mathbb{R}^{n \times n}$ eine Permutationsmatrix, die den k -ten und ℓ -ten Eintrag eines Vektors vertauscht, wobei $\ell > k$ gelte.

- (i) Sei $A \in \mathbb{R}^{n \times n}$. Bestimmen Sie PA sowie AP .
- (ii) Sei $L = I_n - \ell_k e_k^\top$ mit dem kanonischen Basisvektor $e_k \in \mathbb{R}^n$ und einem Vektor $\ell_k = [0, \dots, 0, \ell_{k+1,k}, \dots, \ell_{n,k}]^\top$. Zeigen Sie, dass ein Vektor

$$\hat{\ell}_k = [0, \dots, 0, \hat{\ell}_{k+1,k}, \dots, \hat{\ell}_{n,k}]^\top$$

existiert, sodass mit $\hat{\ell} = I_n - \hat{\ell}_k e_k^\top$ die Identität $\hat{\ell} = PLP$ gilt.

Aufgabe 28.4.9 Sei $A \in \mathbb{R}^{m \times n}$. Konstruieren Sie ein Verfahren zur Bestimmung aller Lösungen von $Ax = 0$.

Aufgabe 28.4.10 Für $k = 1, 2, \dots, n-1$ sei $L^{(k)} = I_n - \ell_k e_k^\top$ mit Vektoren $\ell_k = [0, \dots, 0, \ell_{k+1,k}, \dots, \ell_{n,k}]^\top$ für $k = 1, 2, \dots, n-1$ und es sei $\tilde{L} = L^{(n-1)} L^{(n-2)} \dots L^{(1)}$. Zeigen Sie, dass

$$\tilde{L}^{-1} = I_n + \sum_{k=1}^{n-1} \ell_k e_k^\top.$$

Projekt 28.4.1 Schreiben Sie ein C-Programm, das für eine LU -zerlegbare Matrix $A \in \mathbb{R}^{n \times n}$ und einen Vektor $b \in \mathbb{R}^n$ das lineare Gleichungssystem $Ax = b$ mittels Gauß-Elimination löst und dabei die LU -Zerlegung von A bestimmt:

$$A = \begin{bmatrix} 1 & 7 & -2 & 3 \\ 5 & -1 & -4 & 0 \\ 8 & 1 & 3 & 5 \\ 4 & -4 & 4 & -4 \end{bmatrix}, \quad b = \begin{bmatrix} 21 \\ -9 \\ 39 \\ -8 \end{bmatrix}.$$

Dabei kann die Matrix A durch die berechneten Werte $a_{ij}^{(k+1)}$ und ℓ_{ik} überschrieben werden. Verwenden Sie Ihr Programm zur Lösung von Gleichungssystemen mit oberer Dreiecksmatrix, um das resultierende System $A^{(n)}x = b^{(n)}$ zu lösen. Testen Sie das Programm mit der oben angegebenen Matrix A und dem Vektor b .

Projekt 28.4.2 Stören Sie die rechte Seite des nachfolgenden Gleichungssystems mit dem Vektor $d \in \mathbb{R}^n$, $d_i = 10^{-5} \cos(i\pi/n)$ für $i = 1, 2, \dots, n$ und $n = 10$:

$$a_{ij} = (i + j - 1)^{-1}, \quad b_i = \sum_{k=1}^n (-1)^{k-1} / (i + k - 1), \quad x_i = (-1)^{i-1}, \quad i, j = 1, 2, \dots, n.$$

Betrachten Sie den relativen Fehler $\|x - x_d\|_2 / \|x\|_2$ und vergleichen Sie diesen mit der Konditionszahl der Matrix, die Sie mit dem MATLAB-Befehl `cond(A, 2)` bestimmen können. Kommentieren Sie die Ergebnisse.

Projekt 28.4.3 Implementieren Sie das Gaußsche Eliminationsverfahren mit Pivotsuche. Führen Sie dazu einen Vektor $\pi \in \mathbb{N}^n$ ein, der die Zeilenvertauschungen berücksichtigt. Implementieren Sie zudem ein Abbruchkriterium, das das Verfahren beendet, sofern für das Pivotelement die Abschätzung $|a_{\pi(k),k}^{(k)}| \leq 10^{-10}$ gilt. Beim Lösen des resultierenden Gleichungssystems sind in der Rückwärtssubstitution die Zeilenvertauschungen zu beachten. Testen Sie das Verfahren für das Gleichungssystem sowie $A \in \mathbb{R}^{3 \times 3}$ und $b \in \mathbb{R}^3$ definiert durch

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

28.5 Ausgleichsprobleme

Aufgabe 28.5.1 Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ sowie $x, y \in \mathbb{R}^n$. Berechnen Sie die Ableitung der Abbildung

$$t \mapsto \|A(x + ty) - b\|_2^2, \quad t \in \mathbb{R},$$

und folgern Sie die Gaußsche Normalengleichung, falls x eine Lösung des zugehörigen Ausgleichsproblems ist.

Aufgabe 28.5.2 Seien $A \in \mathbb{R}^{2 \times 1}$ und $b \in \mathbb{R}^2$ definiert durch

$$A = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

Bestimmen Sie zeichnerisch mit Hilfe eines Geodreiecks die Lösung des Ausgleichsproblems, indem Sie b orthogonal in Vektoren v, w mit $v \in \text{Im } A$ und $w \in \ker A^\top$ zerlegen.

Aufgabe 28.5.3 Eine Householder-Matrix $P \in \mathbb{R}^{m \times m}$ ist für $v \in \mathbb{R}^m$ mit $\|v\|_2 = 1$ definiert durch $P = I_m - 2vv^\top$.

- (i) Zeigen Sie, dass $P = P^\top$ und $P^{-1} = P$ gelten.
- (ii) Zeigen Sie, dass eine reelle $m \times m$ Householder-Matrix $m - 1$ Eigenwerte mit dem Wert 1 und einen Eigenwert -1 hat.
- (iii) Konstruieren Sie mit Hilfe geometrischer Überlegungen für $m = 2, 3$ eine Householder-Matrix, die einen gegebenen Vektor $x \in \mathbb{R}^m$ auf ein Vielfaches von $e_1 \in \mathbb{R}^m$ abbildet.

Aufgabe 28.5.4 Sei $D \in \mathbb{R}^{m \times m}$ eine Diagonalmatrix mit positiven Diagonaleinträgen. Die Minimierung von $x \mapsto \|D(Ax - b)\|_2^2$ realisiert beispielsweise eine unterschiedliche Gewichtung verschiedener Messergebnisse. Bestimmen Sie die zugehörige Normalengleichung.

Aufgabe 28.5.5 Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $1 < p < \infty$. Berechnen Sie die partiellen Ableitungen der Abbildung

$$x \mapsto \|Ax - b\|_p^p, \quad x \in \mathbb{R}^n.$$

Bestimmen Sie alle Zahlen p , für die die Ableitung durch eine lineare Abbildung gegeben ist.

Aufgabe 28.5.6 Berechnen Sie mit Hilfe des Householder-Verfahrens eine QR -Zerlegung für

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -\sqrt{2} & \sqrt{2}/2 \\ 0 & \sqrt{2} & 5/\sqrt{2} \end{bmatrix}$$

und lösen Sie damit die Gleichung $Ax = b$ für $b = [3\sqrt{2}, -1, 7]^\top$.

Aufgabe 28.5.7 Es sei $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix mit den Spaltenvektoren $a_1, a_2, \dots, a_n \in \mathbb{R}^n$ und (q_1, q_2, \dots, q_n) sei die daraus durch das Gram–Schmidt-Verfahren gewonnene Orthonormalbasis, das heißt

$$\tilde{q}_j = a_j - \sum_{k=1}^{j-1} (a_j \cdot q_k) q_k, \quad q_j = \frac{\tilde{q}_j}{\|\tilde{q}_j\|_2}$$

für $j = 1, 2, \dots, n$.

- (i) Zeigen Sie, dass für $R \in \mathbb{R}^{n \times n}$ definiert durch $r_{kj} = a_j \cdot q_k$ für $k < j$, $r_{kj} = 0$ für $k > j$, $r_{jj} = \|\tilde{q}_j\|_2$ für $j = 1, \dots, n$, folgt $A = QR$.
- (ii) Berechnen Sie Q und R für

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

Aufgabe 28.5.8 Sei $A \in \mathbb{R}^{m \times n}$ und $A = QR$ eine QR -Zerlegung. Zeigen Sie, dass R eine Cholesky-Zerlegung von $A^\top A$ definiert.

Aufgabe 28.5.9 Sei $A \in \mathbb{R}^{n \times n}$ regulär und $A = QR$ eine QR -Zerlegung. Zeigen Sie, dass $\text{cond}_2(A) = \text{cond}_2(R)$ gilt.

Aufgabe 28.5.10 Sei $i < j$, $\theta \in \mathbb{R}$ und definiere $B = B(i, j, \theta) \in \mathbb{K}^{m \times m}$ durch $b_{k\ell} = \delta_{k\ell}$ für $k \neq i, j$, $b_{ii} = b_{jj} = c$ und $b_{ij} = -b_{ji} = s$, mit $c = \cos(\theta)$ und $s = \sin(\theta)$, das heißt

$$B(i, j, \theta) = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & c & & s \\ & & & 1 & \\ & & & & \ddots \\ & & -s & & c \\ & & & & \\ & & & & 1 \\ & & & & & \ddots \end{bmatrix}.$$

- (i) Zeigen Sie, dass die Matrix $B(i, j, \theta)$ eine Drehung der (i, j) -Ebene um den Winkel θ bewirkt.
- (ii) Zeigen Sie, dass die sukzessive Multiplikation von $A \in \mathbb{R}^{m \times n}$ mit geeigneten $B(i, j, \theta)$ auf eine QR -Zerlegung führt.
- (iii) Ist dieses Verfahren aufwendiger als das Householder-Verfahren? Falls ja, gibt es Klassen von Matrizen, bei denen es weniger aufwendig ist?

Projekt 28.5.1 Implementieren Sie das Householder-Verfahren zur Berechnung einer QR -Zerlegung in C. Verwenden Sie Ihr Programm, um das Gleichungssystem $Ax = b$ mit der $n \times n$ Hilbert-Matrix A definiert durch $a_{ij} = (i + j - 1)^{-1}$, $1 \leq i, j \leq n$, und der rechten Seite $b = [1, 2, \dots, n]^\top$ für $n = 3$ und $n = 10$ zu lösen.

Projekt 28.5.2 Aus der Physik ist bekannt, dass Körper, die nur der Schwerkraft ausgesetzt sind, in Parabeln fliegen. Ein Körper habe die Anfangsgeschwindigkeit $v = (v_x, v_y)$ und befindet sich zum Zeitpunkt $t = 0$ am Punkt 0. Zum Zeitpunkt t befindet er sich dann am Ort $x = v_x t$, $y = v_y t - \frac{1}{2} g t^2$, wobei g die Erdbeschleunigung ist. In einer Versuchsreihe wurden die in Tab. 28.1 angegebenen Werte gemessen. Formulieren Sie ein geeignetes Ausgleichsproblem und lösen Sie dieses in MATLAB mit Hilfe der durch $[Q, R] = qr(A)$ bereitgestellten QR -Zerlegung, um die Geschwindigkeit v_y und die

Tab. 28.1 Messwerte einer Versuchsreihe

i	1	2	3	4	5	6	7
$t_i [s]$	0.1	0.2	0.6	0.9	1.1	1.2	2.0
$x_i [m]$	0.73	1.28	4.24	6.11	7.69	8.21	13.83
$y_i [m]$	0.96	1.81	4.23	5.05	5.15	4.81	0.55

Erdbeschleunigung g möglichst gut zu bestimmen. Erstellen Sie mit Hilfe des Befehls `plot` ein Schaubild, in dem die Messwerte und die berechnete Parabel aufgeführt sind. Bis zu welcher Genauigkeit ist es sinnvoll die Ergebnisse anzugeben? Welche Modellfehler, Datenfehler und Messfehler treten bei diesem Versuch auf?

28.6 Singulärwertzerlegung und Pseudoinverse

Aufgabe 28.6.1 Seien $A, B \in \mathbb{R}^{m \times n}$ und $(v_1, v_2, \dots, v_n) \subset \mathbb{R}^n$ eine Basis des \mathbb{R}^n . Zeigen Sie, dass aus $Av_i = Bv_i$ für $i = 1, 2, \dots, n$ die Gleichheit $A = B$ folgt.

Aufgabe 28.6.2 Bestimmen Sie eine Singulärwertzerlegung der Matrix

$$A = \frac{1}{4} \begin{bmatrix} 3 & 1 & -1 & -3 \\ -1 & -3 & 3 & 1 \end{bmatrix}^\top.$$

Berechnen Sie A^+ mit Hilfe der Singulärwertzerlegung sowie mittels der Identität $A^+ = (A^\top A)^{-1} A^\top$. Verwenden Sie A^+ , um das durch A und $b = [4, 1, 2, 3]^\top$ definierte Ausgleichsproblem zu lösen.

Aufgabe 28.6.3 Sei $A \in \mathbb{R}^{m \times n}$. Zeigen Sie, dass die Pseudoinverse A^+ die eindeutige Lösung der Gleichungen

$$AXA = A, \quad XAX = X, \quad (AX)^\top = AX, \quad (XA)^\top = XA$$

ist. Nehmen Sie zum Nachweis der Eindeutigkeit die Existenz einer zweiten Lösung Y an, leiten Sie die Identitäten $X = XA(YAY)(AXA)X$ sowie $Y = (YA)^\top Y(AY)^\top$ her und zeigen Sie, dass die rechten Seiten übereinstimmen.

Aufgabe 28.6.4 Zeigen Sie, dass $\text{rank } A^\top A = \text{rank } AA^\top = \text{rank } A$ gilt.

Aufgabe 28.6.5

- (i) Sei $V \subset \mathbb{R}^n$ ein Unterraum und V^\perp sein orthogonales Komplement. Zeigen Sie, dass eine eindeutig bestimmte Matrix $P_V \in \mathbb{R}^{n \times n}$ existiert mit $P_V v = v$ für alle $v \in V$ und $P_V w = 0$ für alle $w \in V^\perp$.
- (ii) Sei $A \in \mathbb{R}^{m \times n}$. Zeigen Sie, dass $A^+ A = P_{(\ker A)^\perp}$ und $AA^+ = P_{\text{Im } A}$.

Aufgabe 28.6.6 Seien $(\lambda_i, v_i) \in \mathbb{R} \times \mathbb{R}^n$, $i = 1, \dots, n$, Eigenwerte und zugehörige linear unabhängige Eigenvektoren der Matrix $A \in \mathbb{R}^{n \times n}$. Zeigen Sie, dass A die Darstellung $A = VDV^{-1}$ mit $V = [v_1, \dots, v_n]$ und $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ besitzt.

Aufgabe 28.6.7 Seien $A \in \mathbb{R}^{n \times n}$ mit Eigenwerten $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ und $\|\cdot\|_{op}$ eine Operatormodulnorm.

- (i) Zeigen Sie, dass $\|A\|_2 \leq \|A\|_{op}$ gilt.
- (ii) Zeigen Sie, dass $\max_{i=1, \dots, n} |\lambda_i| \leq \|A\|_2$ gilt.

Aufgabe 28.6.8

- (i) Sei $A \in \mathbb{R}^{n \times n}$ und $\lambda \in \mathbb{C}$ ein Eigenwert von A . Beweisen Sie die folgenden Aussagen:
- Die Zahl $\bar{\lambda}$ ist Eigenwert von A .
 - Ist A symmetrisch, so sind die Eigenwerte von A reell.
 - Ist A regulär, so ist λ^{-1} Eigenwert von A^{-1} .
 - Die Matrix A^\top besitzt den Eigenwert λ .
- (ii) Seien $A, B \in \mathbb{R}^{n \times n}$ Matrizen mit Eigenwerten λ und μ . Unter welchen Voraussetzungen ist $\lambda\mu$ Eigenwert von AB ?

Aufgabe 28.6.9

- (i) Sei $n \in \mathbb{N}$ ungerade und $Q \in SO(n)$, das heißt es gelte $Q \in \mathbb{R}^{n \times n}$ mit $Q^\top Q = I_n$ sowie $\det Q = 1$. Zeigen Sie, dass Q den Eigenwert 1 besitzt.
- (ii) Folgern Sie, dass es auf der Oberfläche eines Fußballs mindestens zwei Punkte gibt, die sich im Laufe eines Fußballspiels mindestens zweimal an derselben Stelle des umgebenden Raumes befinden.

Aufgabe 28.6.10 Zeigen Sie, dass die durch $a, b, c \in \mathbb{R}$ mit $bc > 0$ definierte Tridiagonalmatrix

$$A = \begin{bmatrix} a & b & & & \\ c & a & \ddots & & \\ & \ddots & \ddots & b & \\ & & & c & a \end{bmatrix} \in \mathbb{R}^{n \times n}$$

die Eigenwerte $\lambda_k = a + 2 \operatorname{sign}(c)\sqrt{bc} \cos(k\pi/(n+1))$, $k = 1, 2, \dots, n$, besitzt. Betrachten Sie dazu zunächst den Fall $a = 0$ und die Vektoren

$$v_k = ((c/b)^{\ell/2} \sin(k\pi\ell/(n+1)))_{\ell=1,\dots,n}.$$

Projekt 28.6.1 In MATLAB kann die Singulärwertzerlegung einer Matrix A mit dem Befehl `svd` berechnet werden. Für ein durch die Datei `bild.jpg` definiertes Bild, kann eine Kompression der Graustufendarstellung mit den in Abb. 28.1 gezeigten Zeilen definiert werden. Wählen Sie als Bild beispielsweise den Ausschnitt aus Albrecht Dürers Bild *Melancolia I*, der das magische Quadrat zeigt. Erklären Sie die einzelnen Zeilen des Programms und erweitern Sie es um eine Berechnung des Approximationsfehlers $\|X - X_{\text{comp}}\|_{\mathcal{F}}$. Wie beurteilen Sie das Verhältnis von Qualitätsverlust zur Reduktion des Speicheraufwands für verschiedene Werte von k ? Testen Sie das Programm für ein weiteres Bild.

```

RGB = imread('bild.jpg');
G = rgb2gray(RGB);
D = double(G);
X = mat2gray(D);
figure(1);
subplot(1,2,1); imshow(X); title('Original');
[U,S,V] = svd(X);
for k = 5:5:size(U,1)
    X_comp = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
    subplot(1,2,2); imshow(X_comp);
    title('Komprimiert'); pause
end

```

Abb. 28.1 Bildkompression mittels Singulärwertzerlegung

Projekt 28.6.2 Das Einheitsquadrat $Q = [0, 1]^2 \subset \mathbb{R}^2$ lässt sich in MATLAB durch $\text{fill}(X, Y, 0)$ mit $X = [0, 1, 1, 0, 0]$ und $Y = [0, 0, 1, 1, 0]$ darstellen. Visualisieren Sie das Bild $A(Q)$ mit den linearen Abbildungen, die durch die Matrizen

$$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}, \quad \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \quad \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} c' & s' \\ s & c' \end{bmatrix}$$

mit geeigneten Zahlen $k, k_1, k_2 \in \mathbb{R}$, $c = \cos(\theta)$, $s = \sin(\theta)$ für $\theta \in [0, 2\pi]$. Bestimmen Sie mit dem MATLAB-Kommando $[V, D] = \text{eig}(A)$ die Eigenwerte und Eigenvektoren der Abbildungen und interpretieren Sie diese geometrisch.

28.7 Das Simplex-Verfahren

Aufgabe 28.7.1 Sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Zeigen Sie, dass die Menge $C = \{x \in \mathbb{R}^n : x \geq 0, Ax = b\}$ höchstens endlich viele Ecken besitzt.

Hinweis: Betrachten Sie die Nulleinträge von Elementen in C .

Aufgabe 28.7.2 Sei $f(x) = a^\top x + b$ und $C \subset \mathbb{R}^n$ eine konvexe, abgeschlossene und beschränkte Menge. Zeigen Sie, dass die Funktion f ihre Extremwerte in den Ecken von C annimmt, das heißt es existieren Eckpunkte $x_m, x_M \in C$ mit $f(x_m) = \min_{x \in C} f(x)$ und $f(x_M) = \max_{x \in C} f(x)$.

Aufgabe 28.7.3 Formulieren Sie das Ausgleichsproblem

$$\text{Minimiere } x \mapsto \|Ax - b\|_\infty$$

als lineares Programm in Normalform.

Aufgabe 28.7.4 Bestimmen Sie sämtliche Ecken der konvexen Mengen $B_1^2(0) = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$ und $B_1^\infty(0) = \{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$.

Aufgabe 28.7.5

(i) Bestimmen Sie sämtliche Minima der Funktion $f(x) = \sum_{i=1}^n |x - z_i|$ für die Fälle

$$z = [-1, 1]^\top, \quad z = [-1, 0, 1]^\top, \quad z = [0, 10]^\top, \quad z = [0, 1, 10]^\top.$$

Wie lassen sich Minima charakterisieren?

(ii) Seien $z_1, z_2, \dots, z_n \in \mathbb{R}$ mit $z_1 \leq z_2 \leq \dots \leq z_n$. Bestimmen Sie ein Minimum der Funktion

$$f(x) = \sum_{i=1}^n |x - z_i|.$$

Verwenden Sie dabei die formale notwendige Optimalitätsbedingung $f'(x^*) = 0$ mit der Ableitung $|\cdot|' = \text{sign}(\cdot)$.

Aufgabe 28.7.6 Seien $z_1, z_2, \dots, z_n \in \mathbb{R}$. Formulieren Sie die Minimierung der Funktion $f(y) = \sum_{i=1}^n |y - z_i|$ als lineares Programm.

Aufgabe 28.7.7 Seien $a \in \mathbb{R}^2$ und $c \in \mathbb{R}$. Konstruieren Sie durch geometrische Überlegungen die Minimierer der Abbildungen $x \mapsto \|x\|_p$ unter der Nebenbedingung $a^\top x = \alpha$ für $p = 1, 2, \infty$.

Aufgabe 28.7.8 Sei $a \in \mathbb{R}^n \setminus \{0\}$ und sei $C \subset \mathbb{R}^n$ nichtleer und strikt konvex, das heißt gilt $\theta x_1 + (1-\theta)x_2 \in \partial C$ für $x_1, x_2 \in C$, so folgt $\theta = 1$ oder $\theta = 0$. Zeigen Sie, dass die Minimierung der Funktion $f(x) = a \cdot x$ unter der Nebenbedingung $x \in C$ eine eindeutige Lösung besitzt.

Aufgabe 28.7.9 Seien $A = [4, 2, 1]$, $b = 4$ und $c = [1, 1, 1]^\top$.

- (i) Bestimmen Sie die Ecken der Menge $\{x \in \mathbb{R}^3 : x \geq 0, Ax = b\}$ und untersuchen Sie, ob diese entartet sind.
- (ii) Führen Sie das Simplex-Verfahren zur Minimierung von $f(x) = c^\top x$ unter der Nebenbedingung $Ax = b$ und $x \geq 0$ mit der Startecke $x^0 = [0, 0, 4]^\top$ durch.

Aufgabe 28.7.10 Konstruieren Sie Matrizen $A \in \mathbb{R}^{3 \times 2}$ und Vektoren $b \in \mathbb{R}^2$, für die die Menge $M = \{x \in \mathbb{R}^3 : Ax = b, x \geq 0\}$ leer, unbeschränkt sowie beschränkt und nichtleer ist.

Projekt 28.7.1 Eine Firma stellt m verschiedene Produkte her, für deren Fertigung n Maschinen benötigt werden. Die j -te Maschine hat eine maximale monatliche Laufzeit von ℓ_j Stunden. Das k -te Produkt generiert pro Mengeneinheit einen Ertrag von e_k Euro und belegt die j -te Maschine mit t_{jk} Stunden pro Mengeneinheit. Der monatliche Gesamtertrag soll ohne Überschreitung der Maximallaufzeiten optimiert werden.

- (i) Formulieren Sie den beschriebenen Sachverhalt als Maximierungsproblem mit Nebenbedingungen in der Form

$$\text{Maximiere } f(x) = c \cdot x \text{ unter den Bedingungen } Ax \leq b, x \geq 0$$

wobei $x = (x_1, x_2, \dots, x_m)$ die monatlichen Mengeneinheiten der verschiedenen Produkte seien und die Ungleichungen komponentenweise zu verstehen sind.

- (ii) Verwenden Sie die MATLAB-Routine `linprog`, um das Problem für die Daten $m = 2, n = 3, e_1 = 200, e_2 = 600$, und $t_{11} = 1, t_{21} = 1, t_{31} = 0, t_{12} = 3, t_{22} = 1, t_{32} = 2$ sowie $\ell_1 = 150, \ell_2 = 180, \ell_3 = 140$ zu lösen. Wie groß ist der optimale monatliche Ertrag?

Projekt 28.7.2 Sind $a \in \mathbb{R}^3$ ein Vektor mit positiven Komponenten und $c \in \mathbb{R}$ eine positive Zahl, so wird durch $\{x \in \mathbb{R}^3 : x \geq 0, a^\top x \leq \alpha\}$ ein Tetraeder definiert. Es soll der Mittelpunkt m und der Radius $r > 0$ einer im Tetraeder enthaltenen Kugel maximalen Volumens bestimmt werden. Formulieren Sie das Problem als lineares Programm und lösen Sie es mit der MATLAB-Routine `linprog`. Bestimmen Sie dann die Lösung für den Fall $a = [1, 2, 3]^\top$ und $\alpha = 4$. Sie können Ihre Lösung mit den in Abb. 28.2 gezeigten MATLAB-Kommandos visualisieren, wobei $Z \in \mathbb{R}^{4 \times 3}$ eine Matrix ist, die die Koordinaten der Ecken des Tetraeders enthält.

```
[x_1,x_2,x_3] = sphere;
surf([m_1+r*x_1,m_2+r*x_2,m_3+r*x_3]); hold on;
tetramesh([1,2,3,4],z); hold off;
```

Abb. 28.2 Visualisierung einer Sphäre

Hinweis: Der Abstand eines Punktes $m \in \mathbb{R}^3$ zu der durch einen Vektor $v \in \mathbb{R}^3$ mit $\|v\|_2 = 1$ und eine Zahl $\gamma \in \mathbb{R}$ definierten Ebene $\{x \in \mathbb{R}^3 : v^\top x = \gamma\}$ ist gegeben durch $|v^\top m - \gamma|$.

28.8 Eigenwertaufgaben

Aufgabe 28.8.1

- (i) Bestimmen Sie die Gerschgorin-Kreise der Matrix

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}.$$

- (ii) Sei $A \in \mathbb{R}^{n \times n}$ strikt diagonaldominant und symmetrisch. Geben Sie eine obere Schranke der Konditionszahl $\text{cond}_2(A)$ an.

Aufgabe 28.8.2 Zeigen Sie, dass das charakteristische Polynom $p(\lambda) = \det(A - \lambda I_n)$ der $n \times n$ -Matrix

$$A = \begin{bmatrix} 0 & & & -a_0 \\ 1 & 0 & & -a_1 \\ \ddots & \ddots & \ddots & \vdots \\ & 1 & 0 & -a_{n-2} \\ & & 1 & -a_{n-1} \end{bmatrix}$$

gegeben ist durch $p(\lambda) = (-1)^n(\lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_1\lambda + a_0)$.

Aufgabe 28.8.3

- (i) Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch mit Eigenwerten $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ und sei $v_1 \in \mathbb{R}^n \setminus \{0\}$ ein Eigenvektor zum Eigenwert λ_1 . Zeigen Sie, dass

$$\lambda_2 = \max_{\substack{x \in \mathbb{R}^n \setminus \{0\} \\ x \cdot v_1 = 0}} \frac{x^\top A x}{\|x\|_2^2}.$$

- (ii) Zeigen Sie, dass der Vektor $x^* \in \mathbb{R}^n \setminus \{0\}$ genau dann ein Eigenvektor der Matrix $A \in \mathbb{R}^{n \times n}$ ist, wenn $\nabla r(x^*) = 0$ gilt mit der Funktion

$$r : \mathbb{R}^n \setminus \{0\} \rightarrow \mathbb{R}, \quad x \mapsto \frac{x^\top A x}{\|x\|_2^2}.$$

Aufgabe 28.8.4

- (i) Zeigen Sie, dass die Potenzmethode auch dann konvergiert, wenn die Iterierten bezüglich einer anderen Norm normiert werden.

- (ii) Führen Sie 5 Schritte der Potenzmethode für die Matrizen

$$A = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & -1 \\ 0 & -1 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} -6 & -22 & 59 \\ -4 & -6 & 22 \\ -2 & -4 & 13 \end{bmatrix}$$

mit dem Anfangsvektor $x_0 = [1, 1, 1]^\top / 2$ durch und beobachten Sie die Größen $\|\tilde{x}_k\|_2$ und $x_k^\top A x_k$.

- Aufgabe 28.8.5** Bestimmen Sie die k -te Iterierte der Potenzmethode für die Matrix

$$A = \begin{bmatrix} 0 & 2 & & \\ & \ddots & \ddots & \\ & & \ddots & 2 \\ 2 & & & 0 \end{bmatrix}$$

mit den Startvektoren $x_0 = [1, 0, \dots, 0]^\top$ sowie $x_0 = [1, 1, \dots, 1]^\top$ und diskutieren Sie die Gültigkeit der Voraussetzungen des Konvergenzresultats.

- Aufgabe 28.8.6** Sei $A \in \mathbb{R}^{n \times n}$.

- (i) Zeigen Sie, dass eine Householder-Matrix $\tilde{H} \in \mathbb{R}^{(n-1) \times (n-1)}$ existiert, sodass für $B = HAH^\top$ mit $H = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H} \end{bmatrix}$ die Eigenschaft $b_{i1} = 0$ für $i > 2$ gilt.
- (ii) Folgern Sie, dass sich A mit $n-2$ Ähnlichkeitstransformationen in eine Matrix $\hat{A} \in \mathbb{R}^{n \times n}$ mit $\hat{a}_{ij} = 0$ für $i > j+1$ überführen lässt. Diskutieren Sie den erforderlichen numerischen Aufwand.
- (iii) Zeigen Sie, dass die Eigenschaft $a_{ij} = 0$ für $i > j+1$ im QR-Verfahren erhalten bleibt.

- Aufgabe 28.8.7** Führen Sie einen Schritt des QR-Verfahrens für die Matrix

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 3 & 5 \\ 0 & 1 & 2 \end{bmatrix}$$

durch, bestimmen Sie die Eigenwerte von A mit Hilfe des charakteristischen Polynoms und vergleichen Sie die Ergebnisse.

Aufgabe 28.8.8

- (i) Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix und $G_{pq} \in \mathbb{R}^{n \times n}$ eine Givens-Rotation. Zeigen Sie für die Einträge der Matrix $B = G_{pq}^{-1}AG_{pq}$, dass

$$\begin{aligned} b_{pp} &= c^2 a_{pp} + 2csa_{pq} + s^2 a_{qq}, \\ b_{qq} &= s^2 a_{pp} - 2csa_{pq} + c^2 a_{qq}, \\ b_{pq} &= b_{qp} = cs(a_{qq} - a_{pp}) + (c^2 - s^2)a_{pq}, \\ b_{ip} &= ca_{ip} + sa_{iq}, \quad i \in \{1, 2, \dots, n\} \setminus \{p, q\}, \\ b_{iq} &= -sa_{ip} + ca_{iq}, \quad i \in \{1, 2, \dots, n\} \setminus \{p, q\}, \\ b_{ij} &= a_{ij}, \quad i, j \notin \{p, q\}. \end{aligned}$$

- (ii) Folgern Sie $b_{pq} = 0$, sofern $a_{pq} \neq 0$ gilt und G_{pq} definiert ist durch $c = \sqrt{(1+D)/2}$ und $s = -\text{sign}(a_{pq})\sqrt{(1-D)/2}$ mit

$$D = \frac{a_{pp} - a_{qq}}{\left((a_{pp} - a_{qq})^2 + 4a_{pq}^2\right)^{1/2}}.$$

Aufgabe 28.8.9 Sei $\|A\|_F = (\sum_{i,j=1}^n a_{ij}^2)^{1/2}$ die Frobenius-Norm.

- (i) Zeigen Sie, dass $\|A\|_F^2 = \text{tr}(A^\top A)$ sowie $\text{tr}(AB) = \text{tr}(BA)$ für alle $A, B \in \mathbb{R}^{n \times n}$ gilt und folgern Sie $\|Q^{-1}BQ\|_F = \|B\|_F$ für $B \in \mathbb{R}^{n \times n}$, $Q \in O(n)$.
(ii) Zeigen Sie, dass $\|A\|_2 \leq \|A\|_F$ für alle $A \in \mathbb{R}^{n \times n}$ gilt.

Aufgabe 28.8.10 Konstruieren Sie eine symmetrische Matrix $A_k \in \mathbb{R}^{3 \times 3}$ mit einem Eintrag $(A_k)_{ij} = 0$, sodass für die nächste Iterierte A_{k+1} im Jacobi-Verfahren $(A_{k+1})_{ij} \neq 0$ gilt.

Projekt 28.8.1 Implementieren Sie die von-Mises-Potenzmethode, um den kleinsten und größten Eigenwert sowie die zugehörigen Eigenvektoren der $n \times n$ -Matrix

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

für $n = 4, 16, 64, 256$ zu approximieren. Benutzen Sie für die inverse Iteration das MATLAB-Kommando $x = B \backslash c$ zur Lösung eines linearen Gleichungssystems $Bx = c$. Verwenden Sie unterschiedliche Startvektoren und ein geeignetes Abbruchkriterium für die Iteration und bestimmen Sie die Fehler der Approximationen mit Hilfe der exakten Werte $\lambda_{\min} = 2 - 2 \cos(\pi/(n+1))$ und $\lambda_{\max} = 2 - 2 \cos(n\pi/(n+1))$.

Projekt 28.8.2

- (i) Verwenden Sie die MATLAB-Routine $[Q, R] = \text{qr}(A)$, um das QR -Verfahren zu implementieren und beenden Sie die Iteration, falls $\|A_k - A_{k+1}\|_2 / \|A_k\|_2 \leq 10^{-5}$ gilt. Was wären andere sinnvolle Abbruchkriterien? Approximieren Sie mit Ihrem Programm die Eigenwerte der Matrizen $A \in \mathbb{R}^{n \times n}$, $n = 4, 10, 20$ und $B, B^\top \in \mathbb{R}^{3 \times 3}$ definiert durch

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & -10 & 29 \\ -2 & -4 & 18 \\ -1 & -3 & 11 \end{bmatrix}$$

und diskutieren Sie die Voraussetzungen des Satzes über die Konvergenz des Verfahrens anhand dieser Beispiele.

- (ii) Implementieren Sie das Jacobi-Verfahren mit dem Abbruchkriterium $\mathcal{N}(A_k) \leq 10^{-4}$ in MATLAB und testen Sie es für die Matrix $A \in \mathbb{R}^{n \times n}$ definiert durch

$$a_{ij} = \sin(|i - j|\pi/n) - 2\delta_{ij}$$

für $i, j = 1, 2, \dots, n$ mit $n = 2, 4, 8, 16$. Modifizieren Sie das Programm, um eine Implementation des zyklischen Jacobi-Verfahrens zu erhalten, das heißt auf die Suche des größten Eintrags wird verzichtet und alle Einträge werden sukzessive behandelt. Beobachten Sie grafisch die Größe der Einträge der Iterierten mit Hilfe der MATLAB-Kommandos $[X, Y] = \text{meshgrid}(1:n, 1:n)$, $\text{surf}(X, Y, A)$ und $\text{view}(-270, 90)$. Betrachten Sie die Anzahl der benötigten Iterationsschritte in Abhängigkeit von n .

28.9 Iterative Lösungsmethoden

Aufgabe 28.9.1 Konstruieren Sie eine Matrix $M \in \mathbb{R}^{2 \times 2}$, die bezüglich einer Operatornorm eine Kontraktion ist und bezüglich einer anderen nicht.

Aufgabe 28.9.2 Zeigen Sie, dass für die Iterierten der Fixpunktiteration $x^{k+1} = \Phi(x^k)$ mit der Kontraktion $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ die Fehlerabschätzung

$$\|x^k - x^*\| \leq \frac{q}{1-q} \|x^k - x^{k-1}\|$$

gilt. Inwiefern ist diese Abschätzung für praktische Zwecke relevant?

Aufgabe 28.9.3

- (i) Sei $T \in \mathbb{R}^{n \times n}$ eine reguläre Matrix und $\|\cdot\|$ eine Norm auf \mathbb{R}^n . Zeigen Sie, dass durch $\|x\|_T = \|Tx\|$ für $x \in \mathbb{R}^n$ eine weitere Norm auf \mathbb{R}^n definiert wird.
- (ii) Sei $R \in \mathbb{R}^{n \times n}$ und $D \in \mathbb{R}^{n \times n}$ eine invertierbare Diagonalmatrix. Zeigen Sie, dass für $T = D^{-1}RD$ und $i, j = 1, 2, \dots, n$ gilt

$$t_{ij} = \frac{d_{jj}}{d_{ii}} r_{ij}.$$

Aufgabe 28.9.4

- (i) Sei $q \in (0, 1)$. Zeigen Sie, dass ein $j_0 \in \mathbb{N}$ existiert, sodass für alle $j \geq j_0$ die Ungleichung $q(1-q)^j \leq 1/(ej)$ gilt.
- (ii) Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit mit Eigenwerten $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$ und sei $0 < \omega \leq 1/\lambda_1$. Zeigen Sie, dass mit $M = (I_n - \omega A)$ die Abschätzung $\|AM^j\|_2 \leq \lambda_1/j$ für $j \geq j_0$ mit einem geeigneten $j_0 \in \mathbb{N}$ gilt.

Aufgabe 28.9.5

- (i) Seien $A_1, A_2 \in \mathbb{R}^{n \times n}$ definiert durch

$$A_1 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}.$$

Untersuchen Sie die Matrizen im Hinblick auf Diagonaldominanz und Irreduzibilität.

- (ii) Zeigen Sie, dass im Fall der Matrix A_2 für die Iterationsmatrix M^J des Jacobi-Verfahrens die Abschätzung $\rho(M^J) \leq 1/2$ gilt.

Aufgabe 28.9.6 Führen Sie 5 Schritte des Richardson-, Jacobi- und Gauß-Seidel-Verfahrens für

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

mit $\omega = 1$ und $\omega = 1/10$ sowie $x^0 = [1, 1, 1]^\top$ durch. Vergleichen Sie die Iterierten mit der exakten Lösung des Gleichungssystems.

Aufgabe 28.9.7 Zeigen Sie, dass $A \in \mathbb{R}^{n \times n}$ genau dann irreduzibel ist, wenn für alle $i, j \in \{1, 2, \dots, n\}$ eine Folge $i_1, i_2, \dots, i_\ell \in \{1, 2, \dots, n\}$ existiert mit $i_1 = i$ und $i_\ell = j$ sowie $a_{i_k i_{k+1}} \neq 0$ für $k = 1, 2, \dots, \ell - 1$.

Aufgabe 28.9.8 Zeigen Sie, dass $A \in \mathbb{R}^{n \times n}$ genau dann reduzibel ist, wenn eine Permutationsmatrix $P \in \{0, 1\}^{n \times n}$ existiert, sodass

$$PAP^\top = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}$$

mit geeigneten Matrizen B_{11} , B_{12} und B_{22} gilt.

Aufgabe 28.9.9 Für die Matrix $A \in \mathbb{R}^{n \times n}$ gelte $\rho(I_n - A) < 1$. Zeigen Sie, dass A invertierbar ist und dass die Inverse A^{-1} gegeben ist durch die konvergente Reihe

$$A^{-1} = \sum_{i=0}^{\infty} (I_n - A)^i.$$

Hinweis: Betrachten Sie die Matrix $B = I_n - A$ und argumentieren Sie wie bei der Bestimmung des Werts der geometrischen Reihe.

Aufgabe 28.9.10 Zeigen Sie, dass das durch $(D + U)x^{k+1} = -Lx^k + b$ definierte Iterationsverfahren im Fall einer irreduziblen und diagonaldominanten Matrix $A = U + D + L \in \mathbb{R}^{n \times n}$ für jeden Startwert $x^0 \in \mathbb{R}^n$ konvergiert.

Projekt 28.9.1 Verwenden Sie die äquivalenten Darstellungen

$$\begin{aligned} x_i^{k+1} &= a_{ii}^{-1} \left(b_i - \sum_{j \neq i} a_{ij} x_j^k \right), \\ x_i^{k+1} &= a_{ii}^{-1} \left(b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k \right) \end{aligned}$$

des Jacobi- und Gauß-Seidel-Verfahrens, um diese in C zu implementieren. Testen Sie Ihre Programme für das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

und dem Startvektor $x^0 = [1, 1, \dots, 1]^\top \in \mathbb{R}^n$ für $n = 10, 20, 40$. Beenden Sie die Iteration, wenn $\|x^k - x^{k+1}\|_2 \leq \delta$ mit $\delta = 10^{-5}$ gilt. Kommentieren Sie die Abhängigkeit der Iterationszahlen von der Dimension n des Gleichungssystems.

Projekt 28.9.2 Implementieren Sie das Richardson-Verfahren in MATLAB und testen Sie es für das lineare Gleichungssystem $Ax = b$ mit

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad b = \frac{1}{n^2} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

und einem mit dem MATLAB-Kommando `randn(n, 1)` zufällig generierten Startvektor $x^0 \in \mathbb{R}^n$ für $n = 10, 20, 40, 80$ sowie den Parametern $\omega = 1, 1/10, 1/n$. Visualisieren Sie die Iterierten mittels `plot([0:1/n:1], [0, x'])` und beobachten Sie das Verhalten dieser Kurven für mehrere verschiedene Anfangswerte. Versuchen Sie die Iteration in verschiedene Phasen einzuteilen.

29.1 Allgemeine Konditionszahl und Maschinenzahlen

Aufgabe 29.1.1 Zeigen Sie, dass sich jede Zahl $x \in \mathbb{R} \setminus \{0\}$ bezüglich einer Basis $b \geq 2$ in der Form

$$x = \pm b^e \sum_{k=1}^{\infty} d_k b^{-k}$$

mit $d_1, d_2, \dots \in \{0, 1, \dots, b-1\}$ und $e \in \mathbb{Z}$ darstellen lässt, wobei $d_1 \neq 0$ gewählt werden kann.

Aufgabe 29.1.2

- Berechnen Sie die Anzahl der Gleitkommazahlen sowie die positiven Extrema g_{\min} und g_{\max} für die IEEE-Formate *single* und *double precision*.
- Bestimmen Sie $\text{rd}(\pi)$ für $b = 2, p = 5$ und $b = 10, p = 4$.
- Wie lässt sich das Auftreten von *Overflow* bei der Berechnung von $(a^2 + b^2)^{1/2}$ vermeiden, wenn $\max\{|a|, |b|\} > g_{\max}^{1/2}$ und $|a|, |b| \leq g_{\max}/2$ gilt?

Aufgabe 29.1.3

- Stellen Sie die Zahlen 142, 237 und 1111 für die Basen $b = 2, 4$ und 10 mit der Präzision $p = 10$ und den Exponentenschränken $e_{\min} = -10$ sowie $e_{\max} = 10$ als normalisierte Gleitkommazahlen dar.
- Bestimmen Sie die 25. Nachkommastelle von $1/7$.
- Wieso ist die Zahl $1/10$ im Binärsystem nur durch eine unendliche Reihe darstellbar?

Aufgabe 29.1.4

- Sei $\phi = \phi_1 \circ \dots \circ \phi_J$, wobei die Teilaufgaben ϕ_1, \dots, ϕ_J gut konditioniert seien. Zeigen Sie, dass ϕ gut konditioniert ist.
- Sei $g \in C^1(\mathbb{R})$. Diskutieren Sie die Konditionierung der Nullstellenbestimmung von g und illustrieren Sie die Ergebnisse grafisch.

Aufgabe 29.1.5 Es bezeichne $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $\phi(p, q) = (x_1, x_2)$, die Aufgabe der Bestimmung der Nullstellen x_1, x_2 des quadratischen Polynoms $x^2 + px + q$. Bestimmen Sie eine Teilmenge $W \subset \mathbb{R}^2$, auf der ϕ wohldefiniert ist, berechnen Sie für $(p, q) \in W$ die relative Konditionszahl $\kappa_\phi(p, q)$ und diskutieren Sie, für welche Paare (p, q) die Aufgabe gut konditioniert ist.

Aufgabe 29.1.6 Identifizieren Sie mögliche Probleme bei der Auswertung der pq -Formel $x_{1,2} = -p/2 \pm (p^2/4 - q)^{1/2}$ zur Bestimmung der Nullstellen des quadratischen Polynoms $x^2 + px + q$. Konstruieren Sie einen stabilen Algorithmus, indem Sie die Beziehung $x_1 x_2 = q$ ausnutzen.

Aufgabe 29.1.7

- (i) Zeigen Sie, dass die Menge der regulären $n \times n$ -Matrizen eine offene Teilmenge des $\mathbb{R}^{n \times n}$ definiert.
- (ii) Zeigen Sie, dass für $E \in \mathbb{R}^{n \times n}$ und hinreichend kleine Zahlen $h \in \mathbb{R}$ die Matrix $I_n + hE$ regulär ist mit

$$(I_n + hE)^{-1} = \sum_{k=0}^{\infty} (-1)^k h^k E^k.$$

Aufgabe 29.1.8

- (i) Beweisen Sie, dass die harmonische Reihe $\sum_{k=1}^{\infty} 1/k$ in Gleitkommaarithmetik konvergiert.
- (ii) Zeigen Sie, dass die Gleitkommaaddition $+_G$ nicht assoziativ ist.

Aufgabe 29.1.9

- (i) Zeigen Sie, dass die Aufgabe $\phi(x) = (1/x) - (1/(x+1))$ für große Zahlen $x \in \mathbb{R}$ gut konditioniert ist.
- (ii) Zeigen Sie, dass das Verfahren $\tilde{\phi}(x) = (1/x) - (1/(x+1))$ instabil ist.
- (iii) Zeigen Sie, dass das Verfahren $\tilde{\phi}(x) = 1/(x(x+1))$ stabil ist.

Hinweis: Identifizieren Sie die dominierenden Terme des Ausdrucks

$$\tilde{\phi}(\tilde{x}) = \left(\frac{1+\varepsilon_2}{x(1+\varepsilon_1)} - \frac{1+\varepsilon_4}{(x(1+\varepsilon_1)+1)(1+\varepsilon_3)} \right) (1 + \varepsilon_5)$$

und betrachten Sie den Quotienten $|\tilde{\phi}(\tilde{x}) - \phi(x)|/|\phi(x)|$. Verwenden Sie dabei Approximationen $1/(1 + \varepsilon) \approx 1 - \varepsilon$ und $1/(1 + \varepsilon + 1/x) \approx 1 - \varepsilon - 1/x$.

Aufgabe 29.1.10 Verwenden Sie das Gaußsche Eliminationsverfahren ohne beziehungsweise mit Pivotsuche zur Lösung des Gleichungssystems

$$\begin{bmatrix} 0.1 \cdot 10^{-3} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Verwenden Sie dabei Dezimalzahlen mit Präzision $p = 3, 4, 5$, das heißt arbeiten Sie unter Verwendung geeigneter Rundung mit Zahlen der Form $\pm 0.d_1d_2\dots d_p \cdot 10^e$ mit $e \in \mathbb{Z}$ und $d_1, d_2, \dots, d_p \in \{0, 1, \dots, 9\}$.

Projekt 29.1.1 Zur Bestimmung der Rundungsgenauigkeit eines Rechners sei $x = 1$ und es werde x solange durch $x/2$ ersetzt, wie der Ausdruck $1 + x > 1$ vom Rechner als wahr ausgewertet wird. Bestimmen Sie experimentell in C den Wert von x für den dieses Vorgehen abbricht. Definieren Sie dazu x als Variable vom Typ float beziehungsweise double.

Projekt 29.1.2 Wir betrachten die numerische Bestimmung der Eulerschen Zahl e , die sich durch die Grenzwerte

$$e = \lim_{n \rightarrow \infty} (1 + 1/n)^n, \quad e = \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{1}{k!}$$

charakterisieren lässt. Verwenden Sie ausschließlich arithmetische Grundoperationen und endliche Approximationen der obigen Grenzwerte mit $n = 10^j$, $j = 1, 2, \dots, 15$, um e zu approximieren. Bestimmen Sie jeweils die Approximationsfehler mit Hilfe der Näherung $e \approx 2,718281828459045$ und stellen Sie diesen mit 15 Nachkommastellen in einer Tabelle dar. Beurteilen Sie Ihre Ergebnisse.

Projekt 29.1.3 Die Lösung eines linearen Gleichungssystems $Ax = b$ ist nach der Cramerschen Regel gegeben durch $x_i = \det A_i / \det A$, $i = 1, 2, \dots, n$, wobei $A_i \in \mathbb{R}^{n \times n}$ aus A entsteht, indem die i -te Spalte von A durch den Vektor b ersetzt wird. In MATLAB lässt sich A_i mit den Kommandos `A_i=A` und `A_i(:, i)=b;` erzeugen. Implementieren Sie die Cramersche Regel in MATLAB und testen Sie Ihr Programm für das Gleichungssystem $Ax = b$ mit

$$A = \begin{bmatrix} 0.2161 & 0.1441 \\ 1.2969 & 0.8648 \end{bmatrix}, \quad b = \begin{bmatrix} 0.1440 \\ 0.8642 \end{bmatrix}.$$

Die exakte Lösung ist gegeben durch $x = [2, -2]^\top$. Bestimmen Sie für die numerische Lösung \tilde{x} den Vorwärtsfehler $\|x - \tilde{x}\|_\infty / \|x\|_\infty$ sowie den Rückwärtsfehler $\|A\tilde{x} - b\|_\infty / \|b\|_\infty$. Betrachten Sie die Konditionszahl von A und vergleichen Sie die Fehler mit denen der durch das Gaußsche Eliminationsverfahren mit Pivotsuche berechneten numerischen Lösung \hat{x} , die Sie in MATLAB mit `x=A\b` bestimmen können.

29.2 Polynominterpolation

Aufgabe 29.2.1

- (i) Sei $f \in C^2([a, b])$ mit der Eigenschaft $f(a) = f(b)$ und $f'(a) = f'(b) = 0$. Geben Sie eine optimale untere Schranke für die Anzahl der Nullstellen von f'' an.
- (ii) Für Stützstellen $x_0 < x_1 < \dots < x_n$ sei $w(x) = \prod_{j=0}^n (x - x_j)$ das Stützstellenpolynom und L_i , $i = 0, 1, \dots, n$, das i -te Lagrange-Basispolynom. Zeigen Sie, dass gilt

$$L_i(x) = \frac{w(x)}{(x - x_i)w'(x)}.$$

Aufgabe 29.2.2 Es seien $a \leq x_0 < x_1 < \dots < x_n \leq b$ gegebene Stützstellen und (v_0, v_1, \dots, v_n) Polynome vom maximalen Grad n .

- (i) Zeigen Sie, dass die durch $V_{ij} = v_i(x_j)$, $i, j = 0, 1, \dots, n$, definierte Matrix $V \in \mathbb{R}^{(n+1) \times (n+1)}$ genau dann regulär ist, wenn aus $\sum_{i=0}^n \alpha_i v_i(x) = 0$ für alle $x \in [a, b]$ folgt, dass $\alpha_i = 0$ für $i = 0, 1, \dots, n$ gilt.
- (ii) Zeigen Sie, dass im Fall der Monome $v_i(x) = x^i$, $i = 0, 1, \dots, n$, gilt

$$\det V = \prod_{0 \leq i < j \leq n} (x_j - x_i).$$

Aufgabe 29.2.3 Sei $f(x) = \sin(\pi x)$ für $x \in [0, 1]$, $x_0 = 0$ sowie $x_i = i/n$, $i = 0, 1, \dots, n$ sofern $n > 0$ gilt. Berechnen und skizzieren Sie das Interpolationspolynom von f für $n = 0, 1, \dots, 4$.

Aufgabe 29.2.4 Seien $f(x) = \sin(\pi x)$ für $x \in [0, 1]$, $x_0 = 0$ und $x_i = i/n$ für $i = 1, 2, \dots, n$ sofern $n > 0$ gilt. Berechnen und skizzieren Sie die Hermite-Interpolationspolynome für $n = 0, 1, 2$ und $\ell_i = \ell$, $i = 0, 1, \dots, n$, mit $\ell = 0, 1, 2$.

Aufgabe 29.2.5

- (i) Für $x \in [-5, 5]$ sei $f(x) = (1 + x^2)^{-1} = \arctan'(x)$. Verwenden Sie die Identitäten

$$\cos(\arctan(x)) = \frac{1}{(1 + x^2)^{1/2}}, \quad \sin(\arctan(x)) = \frac{x}{(1 + x^2)^{1/2}},$$

$$\sin(x)\sin(y) - \cos(x)\cos(y) = -\cos(x + y), \quad \sin(x)\cos(x) = \frac{1}{2}\sin(2x),$$

um zu beweisen oder für $n = 0, 1, 2, 3$ zu verifizieren, dass

$$f^{(n)}(x) = \frac{n!(-1)^{n/2}}{(1 + x^2)^{(n+1)/2}} \times \begin{cases} \cos((n+1)\arctan(x)), & n \text{ gerade}, \\ \sin((n+1)\arctan(x)), & n \text{ ungerade}. \end{cases}$$

- (ii) Folgern Sie, dass $\|f^{(2n)}\|_\infty = (2n)!$ und dass die Lagrange-Interpolationspolynome von $\tilde{f}(x) = f(5x)$ im Intervall $[-1, 1]$ nicht notwendigerweise uniform für $n \rightarrow \infty$ gegen \tilde{f} konvergieren.

Aufgabe 29.2.6 Konstruieren Sie Stützstellen $a \leq x_0 < x_1 < \dots < x_n \leq b$ im Intervall $[a, b]$, sodass für die Lagrange-Interpolation jeder Funktion $f \in C^{n+1}([a, b])$ gilt

$$\|f - p\|_{C^0([a,b])} \leq 2^{-n} \left(\frac{b-a}{2}\right)^{n+1} \frac{\|f^{(n+1)}\|_{C^0([a,b])}}{(n+1)!}.$$

Aufgabe 29.2.7 Beweisen Sie folgende Eigenschaften der für $t \in [-1, 1]$ durch $T_n(t) = \cos(n \arccos t)$ definierten Funktionen:

- (i) Es gilt $|T_n(t)| \leq 1$ für alle $t \in [-1, 1]$.
- (ii) Mit $T_0(t) = 1$ und $T_1(t) = t$ gilt

$$T_{n+1}(t) = 2t T_n(t) - T_{n-1}(t)$$

für alle $t \in [-1, 1]$. Insbesondere gilt $T_n \in \mathcal{P}_n|_{[-1,1]}$ und für $n \geq 1$ folgt $T_n(t) = 2^{n-1}t^n + q_{n-1}$ mit $q_{n-1} \in \mathcal{P}_{n-1}|_{[-1,1]}$.

- (iii) Für $n \geq 1$ hat T_n die Nullstellen $t_j = \cos((j+1/2)\pi/n)$, $j = 0, 1, \dots, n-1$, und die $n+1$ Extremstellen $s_j = \cos(j\pi/n)$, $j = 0, 1, \dots, n$.

Aufgabe 29.2.8

- (i) Geben Sie ein ausschließlich auf arithmetischen Grundoperationen basierendes Verfahren mit möglichst wenigen Operationen zur Auswertung des Polynoms $(x+3)^{16}$ an.
- (ii) Vergleichen Sie den Aufwand der direkten Auswertung des Polynoms $p(x) = a_0 + a_1x_1 + \dots + a_nx^n$ mit dem unter Verwendung der äquivalenten Darstellung

$$p(x) = a_0 + x(a_1 + x(a_2 + \dots x(a_{n-2} + x(a_{n-1} + xa_n)) \dots)).$$

Aufgabe 29.2.9 Für $n+1$ Stützstellen und -werte $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ und $0 \leq j \leq n$ sowie $0 \leq i \leq n-j$ sei $p_{i,j} \in \mathcal{P}_j$ festgelegt durch $p_{i,j}(x_k) = y_k$, $k = i, i+1, \dots, i+j$. Die Zahlen $y_{i,j}$ seien definiert durch $y_{i,0} = y_i$, $i = 0, 1, \dots, n$, und

$$y_{i,j} = \frac{y_{i+1,j-1} - y_{i,j-1}}{x_{i+j} - x_i}$$

für $1 \leq j \leq n$ und $0 \leq i \leq n-j$.

- (i) Zeigen Sie, dass $p_{i,j}(x) = y_{i,j}x^j + r_{i,j}(x)$ mit einem Polynom $r_{i,j} \in \mathcal{P}_{j-1}$ für $j \geq 1$ und $i = 0, 1, \dots, n-j$ gilt.
- (ii) Zeigen Sie, dass für $q_j(x) = p_{0,j}(x) - p_{0,j-1}(x)$, wobei $p_{0,-1} = 0$ sei, die Darstellung $q_j(x) = y_{0,j} \prod_{i=0}^{j-1} (x - x_i)$ gilt.
- (iii) Folgern Sie, dass $p_{0,n}(x) = \sum_{j=0}^n y_{0,j} \prod_{i=0}^{j-1} (x - x_i)$ gilt.

Aufgabe 29.2.10 Seien $x_0 < x_1 < \dots < x_n$ und $\ell \in \mathbb{N}$. Für $x \in \mathbb{R}$ und $0 \leq j \leq n$ definiere

$$H_{j,\ell}(x) = \frac{(x - x_j)^\ell}{\ell!} \prod_{\substack{i=0 \\ i \neq j}}^n \left(\frac{x - x_i}{x_j - x_i} \right)^{\ell+1}.$$

Zeigen Sie, dass für die Ableitungen von $H_{j,\ell}$ die Identitäten $\frac{d^k}{dx^k} H_{j,\ell}(x_m) = \delta_{k\ell} \delta_{jm}$ für $0 \leq k \leq \ell$ und $0 \leq m \leq n$ gelten.

Projekt 29.2.1 Implementieren Sie das Neville-Schema in nichtrekursiver Form und verwenden Sie es, um das Interpolationspolynom der Funktion $f(x) = (1 + 25x^2)^{-1}$ bezüglich äquidistanter Stützstellen $-1 = x_0 < x_1 < \dots < x_n = 1$ sowie Tschebyscheff-Knoten $-1 \leq t_0 < t_1 < \dots < t_n \leq 1$ an den Punkten $x_a = \pi/8$ und $x_b = \pi/4$ für $n = 1, 2, 4, 8, 16, 32$ auszuwerten. Kommentieren Sie Ihre Beobachtungen.

Projekt 29.2.2

- (i) Schreiben Sie ein MATLAB-Programm zur Bestimmung der Koeffizienten eines Interpolationspolynoms bezüglich der Newton-Basis für gegebene Stützstellen $x_0 < x_1 < \dots < x_n$ und zugehörige -werte y_0, \dots, y_n .
- (ii) Testen Sie Ihr Programm für die Funktionen $f(x) = \sin(\pi x)$, $g(x) = (1 + 25x^2)^{-1}$ und $h(x) = |x|$ im Intervall $[-1, 1]$ bei Verwendung von äquidistanten Stützstellen und Tschebyscheff-Knoten. Werten Sie die Interpolationspolynome an den Punkten $z_j = -1 + 2j/100$, $j = 0, 1, \dots, 100$ mit dem Horner-Schema aus und plotten Sie damit die Interpolationspolynome für $n = 1, 2, 4, 8$.

29.3 Interpolation mit Splines

Aufgabe 29.3.1

- (i) Seien $0 \leq a < b$ und $x \mapsto g(x)$ die lineare Funktion, die die Funktion $f(x) = x^{1/2}$ an den Stützstellen a und b interpoliert. Zeigen Sie, dass für den Fehler $e = \max_{x \in [a,b]} |g(x) - f(x)|$ die Abschätzungen $e \leq (b-a)^2 a^{-3/2}/8$ im Fall $a > 0$ und $e \leq b^{1/2}/4$ im Fall $a = 0$ gelten.
- (ii) Für $n \in \mathbb{N}$ und $x_i = i/n$, $i = 0, 1, \dots, n$, sei $f_n \in S^{1,0}(\mathcal{T}_n)$ die interpolierende Spline-Funktion von $f(x) = x^{1/2}$ im Intervall $[0, 1]$. Zeigen Sie, dass $\max_{x \in [0,1]} |f_n(x) - f(x)| \leq n^{-1/2}/4$ gilt.
- (iii) In welchen Bereichen ist die Fehlerabschätzung suboptimal?

Aufgabe 29.3.2 Für die durch die Punkte $x_i = (i/n)^4$, $i = 0, 1, \dots, n$, definierte Partitionierung von $[0, 1]$ sei $f_n \in S^{1,0}(\mathcal{T}_n)$ die interpolierende Spline-Funktion von $f(x) = x^{1/2}$. Zeigen Sie, dass $\max_{x \in [0,1]} |f_n(x) - f(x)| \leq cn^{-2}$ mit einer von n unabhängigen Konstanten $c > 0$ gilt. Skizzieren Sie f_n für $n = 2, 4, 8$.

Aufgabe 29.3.3

- (i) Zeigen Sie, dass es zu jedem Intervall $[a_0, a_1] \subset \mathbb{R}$ eindeutig bestimmte Polynome $q_{0,0}, q_{0,1}, q_{1,0}, q_{1,1} \in \mathcal{P}_3$ gibt, sodass $q_{j,k}^{(\ell)}(a_m) = \delta_{jm}\delta_{kl}$ für $j, k, \ell, m = 0, 1$ gilt. Zeichnen Sie die Polynome für das Intervall $[0, 1]$.
- (ii) Folgern Sie, dass auf jeder Partitionierung \mathcal{T}_n mit Gitterpunkten $x_0 < x_1 < \dots < x_n$ zu gegebenen Werten y_0, y_1, \dots, y_n und r_0, r_1, \dots, r_n ein eindeutig definierter Spline $s \in S^{3,1}(\mathcal{T}_n)$ mit $s(x_i) = y_i$ und $s'(x_i) = r_i$, $i = 0, 1, \dots, n$, existiert und geben Sie eine Darstellung an.

Aufgabe 29.3.4 Zeigen Sie, dass die kubische Spline-Interpolationsaufgabe mit natürlichen Randbedingungen eindeutig lösbar ist, indem Sie den linearen Teilraum $S_{\text{nat}}^{3,2}(\mathcal{T}_n) = \{s \in S^{3,2}(\mathcal{T}_n) : s''(a) = s''(b) = 0\}$ betrachten.

Aufgabe 29.3.5 Es sei \mathcal{T}_n eine Partitionierung $a = x_0 < x_1 < \dots < x_n = b$ und $s \in S^{3,2}(\mathcal{T}_n)$ der interpolierende kubische Spline der Funktionswerte $y_0 = 1$ und $y_i = 0$, $i = 1, 2, \dots, n$ mit natürlichen Randbedingungen. Zeigen Sie, dass s auf jedem Intervall $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$, nur endlich viele Nullstellen besitzt und geben Sie eine möglichst genaue obere Abschätzung an. Skizzieren Sie die Funktion s .

Aufgabe 29.3.6 Bestimmen Sie explizit die interpolierenden kubischen Splines mit natürlichen sowie Hermite-Randbedingungen $s'(-1) = 0$, $s'(1) = 3$, für die Stützstellen $x_i = -1 + i/2$ und Stützwerte $y_i = (-1)^i$, $i = 0, 1, 2, \dots, 4$, und zeichnen Sie diese.

Aufgabe 29.3.7 Es sei \mathcal{T}_n eine Partitionierung des Intervalls $[a, b]$ und es seien $s \in S^{1,0}(\mathcal{T}_n)$ und $g \in C^1([a, b])$, sodass $s(x_i) = g(x_i)$ für $i = 0, 1, \dots, n$ gilt. Beweisen Sie die Ungleichung

$$\sum_{i=1}^n \int_{x_{i-1}}^{x_i} |s'|^2 dx \leq \int_a^b |g'|^2 dx.$$

Aufgabe 29.3.8 Die Funktionen $B_m : \mathbb{R} \rightarrow \mathbb{R}$, $m \in \mathbb{N}$, seien durch die Rekursion

$$B_{m+1}(x) = \int_{x-1/2}^{x+1/2} B_m(t) dt$$

mit der Initialisierung $B_0(x) = 1$ für $|x| \leq 1/2$ und $B_0(x) = 0$ für $|x| > 1/2$ definiert.

- (i) Zeigen Sie, dass B_m nichtnegativ ist und $B_m(x) = 0$ für $|x| > (m + 1)/2$ gilt.
- (ii) Zeigen Sie, dass mit der durch die Punkte $x_i = i - (m + 1)/2$, $i = 0, \dots, m + 1 = n$, definierten Partitionierung \mathcal{T}_{m+1} des Intervalls $[-(m + 1)/2, (m + 1)/2]$ für jedes $m \in \mathbb{N}$ eine Spline-Funktion $B_m \in S^{m,m-1}(\mathcal{T}_{m+1})$ definiert wird.
- (iii) Bestimmen Sie die Funktionen B_1 , B_2 und B_3 explizit und skizzieren Sie diese.

Aufgabe 29.3.9 Für $n \in \mathbb{N}$ und $i = 0, 1, \dots, n$ sei die Funktion $B_{i,n} : \mathbb{R} \rightarrow \mathbb{R}$ definiert durch

$$B_{i,n}(x) = \binom{n}{i} x^i (1-x)^{n-i}.$$

- (i) Zeigen Sie, dass die Funktionen $(B_{0,n}, B_{1,n}, \dots, B_{n,n})$ eine Basis des Polynomraums \mathcal{P}_n definieren.
- (ii) Beweisen Sie die Formel $B_{i,n}(x) = (1-x)B_{i,n-1}(x) + xB_{i-1,n-1}(x)$.

Aufgabe 29.3.10

- (i) Es seien $P_0, P_1, \dots, P_n \in \mathbb{R}^m$. Zeigen Sie, dass die Abbildung $z : [0, 1] \rightarrow \mathbb{R}^m$,

$$z(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i,$$

- die Eigenschaften $z(0) = P_0$, $z(1) = P_n$ sowie $z'(0) = n(P_1 - P_0)$, $z'(1) = n(P_n - P_{n-1})$ besitzt.
- (ii) Konstruieren Sie Punkte $P_0, P_1, P_2, P_3 \in \mathbb{R}^2$, sodass der Graph der Abbildung z möglichst gut den Viertelkreis $\{(x, y) \in \mathbb{R}^2 : y = (1-x^2)^{1/2}, 0 \leq x \leq 1\}$ approximiert.

Projekt 29.3.1 Der MATLAB-Befehl `plot(X, Y, 'r-*')` stellt einen durch die Vektoren X und Y definierten Polygonzug grafisch dar. Ist $X = [x_0, x_1, \dots, x_n]^\top$ und $Y = [f(x_0), f(x_1), \dots, f(x_n)]^\top$, so wird eine stetige, stückweise lineare Interpolation der Funktion f dargestellt. Die Darstellung des Graphen kann mit dem optionalen Argument `r-*` in Farbe, Liniendarstellung und Markierung verändert werden. Weitere nützliche Befehle sind:

```
hold on, hold off, clf, axis, xlabel, ylabel, legend
```

- (i) Illustrieren Sie grafisch die stückweise lineare Approximation der Funktion $f(x) = x^{1/2}$ auf dem Intervall $[0, 1]$ mit den Gitterpunkten

$$(a) \quad x_i = i/n, \quad (b) \quad x_i = (i/n)^4$$

für $i = 0, 1, \dots, n$ und $n = 2, 4, 8, 16$, indem Sie diese mit der Darstellung von f auf einem sehr feinen Gitter vergleichen.

- (ii) Schreiben Sie eine Routine zur Berechnung eines interpolierenden kubischen Splines mit natürlichen Randbedingungen. Testen Sie die Routine mit den Partitionierungen aus (i) für die Funktion $f(x) = \sin(2\pi x)$. Erzeugen Sie jeweils aussagekräftige Grafiken und speichern Sie diese mit Hilfe des Kommandos `print -djpeg file.jpg` ab. Kommentieren Sie die Ergebnisse.

Projekt 29.3.2

- (i) Recherchieren Sie den Begriff der *Bézier-Kurve* und erläutern Sie ihn in 5 bis 10 Zeilen.
- (ii) Starten Sie das Zeichenprogramm `xfig` unter Unix und zeichnen Sie zwei identische schwarze Ellipsen in Rechtecke mit Seitenlängen $\ell_1 = 5.0\text{ cm}$ und $\ell_2 = 10.0\text{ cm}$.
- (iii) Je ein Viertel der beiden Ellipsen soll mit einer durch die Drawing-Funktionen `Approximated Spline`, `Interpolated Spline`, `Polyline` sowie `Arc` erzeugten, gegebenenfalls zusammengesetzten Kurve approximiert werden. Dabei sollen für jedes der beiden Kreissegmente maximal 3 beziehungsweise 5 Interpolations- oder Kontrollpunkte verwendet werden. Bei zusammengesetzten Kurven zählt eine zweimal verwendete Position als ein Punkt. Verwenden Sie der obigen Reihenfolge entsprechend die Farben rot, blau, grün und gelb. Bereits gesetzte Punkte können mit der `Editing`-Funktion `Move Points` verschoben werden. Am linken unteren Bildschirm können Sie mit Hilfe der Einstellungen `Zoom`, `Grid Mode` und `Point Position` die Darstellung vergrößern, ein Hilfsgitter einblenden und die möglichen Positionen der Punkte verändern.
- (iv) Mit welcher Funktion lässt sich die beste Approximation erzielen? Definieren Sie einen geeigneten Abstandsbegriff für die Kurven und messen Sie von Hand die entsprechenden Fehler.
- (v) Exportieren Sie Ihre Grafik als pdf-Datei.

29.4 Diskrete Fourier-Transformation

Aufgabe 29.4.1

- (i) Seien $n \in \mathbb{N}$ und $\ell \in \mathbb{Z}$. Zeigen Sie, dass $\sum_{k=0}^{n-1} e^{i\ell k 2\pi/n} = n$ gilt, falls n Teiler von ℓ ist, und $\sum_{k=0}^{n-1} e^{i\ell k 2\pi/n} = 0$ andernfalls gilt.
- (ii) Folgern Sie, dass die Fourier-Basis $(\omega^0, \omega^1, \dots, \omega^{n-1}) \subset \mathbb{C}^n$ definiert durch $\omega^k = [\omega_n^{0k}, \omega_n^{1k}, \dots, \omega_n^{(n-1)k}]^\top$, $k = 0, 1, \dots, n-1$, mit der n -ten Einheitswurzel $\omega_n = e^{i2\pi/n}$ die Eigenschaft $\omega^k \cdot \omega^\ell = n\delta_{k\ell}$ besitzt.

Aufgabe 29.4.2

- (i) Es seien die Stützstellen $z_0, z_1, \dots, z_{n-1} \in \mathbb{C}$ paarweise verschieden und die Werte $y_0, y_1, \dots, y_{n-1} \in \mathbb{C}$ beliebig. Zeigen Sie, dass ein eindeutig bestimmtes Polynom $p(z) = \beta_0 + \beta_1 z + \dots + \beta_{n-1} z^{n-1}$ mit komplexen Koeffizienten β_i , $i = 0, 1, \dots, n-1$, existiert, sodass $p(z_j) = y_j$ für $j = 0, 1, \dots, n-1$ gilt.
- (ii) Folgern Sie die eindeutige Lösbarkeit der komplexen trigonometrischen Interpolationsaufgabe.

Aufgabe 29.4.3 Seien $w_0, w_1, \dots, w_{n-1} \in \mathbb{C}$ und $n = 2m$. Konstruieren Sie Zahlen $y_0, y_1, \dots, y_{n-1} \in \mathbb{C}$, sodass mit den Koeffizienten $\beta_0, \beta_1, \dots, \beta_{n-1} \in \mathbb{C}$ der Lösung der zugehörigen komplexen trigonometrischen Interpolationsaufgabe und der Funktion

$$q(x) = \sum_{k=-m}^{m-1} \beta_{k+m} e^{ikx}$$

die Interpolationseigenschaft $q(x_j) = w_j$ für $j = 0, 1, \dots, n-1$ und $x_j = 2\pi j/n$ erfüllt ist.

Aufgabe 29.4.4 Berechnen Sie ohne Verwendung von Matrix-Vektor-Multiplikationen die Fourier-Synthese $y = T_8\beta$ des Vektors

$$\beta = [0, \sqrt{2}, 1, \sqrt{2}, 0, -\sqrt{2}, -1, -\sqrt{2}]^\top.$$

Aufgabe 29.4.5

- (i) Zeigen Sie, dass auf dem Raum der stetigen, komplexwertigen Funktionen $C^0([0, 2\pi]; \mathbb{C})$ durch

$$\langle v, w \rangle = \int_0^{2\pi} v(x) \overline{w(x)} dx$$

ein Skalarprodukt definiert wird.

- (ii) Zeigen Sie, dass die Funktionen $(\varphi_k : k \in \mathbb{Z})$ definiert durch $\varphi_k(x) = e^{ikx}$, $k \in \mathbb{Z}$, $x \in [0, 2\pi]$, ein Orthogonalsystem definieren, das heißt es gilt $\langle \varphi_k, \varphi_\ell \rangle = \delta_{k\ell}$ für alle $k, \ell \in \mathbb{Z}$ mit $k \neq \ell$.

- (iii) Zeigen Sie, dass die Orthogonalität des Systems $(\varphi_k : k \in \mathbb{Z})$ erhalten bleibt, wenn das Integral durch eine Riemannsche Summe approximiert wird, das heißt bezüglich

$$\langle v, w \rangle_n = \frac{2\pi}{n} \sum_{j=0}^{n-1} v(x_j) \overline{w(x_j)}$$

mit $x_j = 2\pi j/n, j = 0, 1, \dots, n-1$.

- Aufgabe 29.4.6** Sei $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ ein Skalarprodukt auf dem reellen, n -dimensionalen Vektorraum V und $(v_0, v_1, \dots, v_{n-1})$ eine Orthonormalbasis von V . Zeigen Sie, dass für jeden Vektor $w \in V$ gilt

$$w = \sum_{j=0}^{n-1} \langle w, v_j \rangle v_j.$$

- Aufgabe 29.4.7** Zu gegebenen $y_0, y_1, \dots, y_{n-1} \in \mathbb{R}$ seien T und p die Lösungen der reellen beziehungsweise komplexen trigonometrischen Interpolationsaufgabe. Zeigen Sie, dass $T(x_j) = p(x_j)$ für $x_j = 2\pi j/n, j = 0, 1, \dots, n-1$, aber im Allgemeinen $T \neq p$ gilt.

Aufgabe 29.4.8

- (i) Zeigen Sie, dass die Lösung der reellen trigonometrischen Interpolationsaufgabe durch die Koeffizienten

$$a_k = \frac{2}{n} \sum_{j=0}^{n-1} y_j \cos(kx_j), \quad b_\ell = \frac{2}{n} \sum_{j=0}^{n-1} y_j \sin(\ell x_j)$$

für $k = 0, 1, \dots, m$ und $\ell = 1, 2, \dots, m-1$ mit $x_j = 2\pi j/n, j = 0, 1, \dots, n-1$, und $n = 2m$ gegeben ist.

- (ii) Folgern Sie, dass die Vektoren

$$f^k = (\cos(kx_j))_{j=0,\dots,n-1}, \quad g^\ell = (\sin(\ell x_j))_{j=0,\dots,n-1}$$

für $k = 0, 1, \dots, m$ und $\ell = 1, 2, \dots, m-1$ eine Orthogonalbasis des \mathbb{R}^n definieren.

- Aufgabe 29.4.9** Es seien $n, m \in \mathbb{N}$ mit $n = 2m$, $A, B \in \mathbb{R}^{n \times n}$ und $C = AB$. Für $i, j \in \{1, 2\}$ seien $A_{ij}, B_{ij}, C_{ij} \in \mathbb{R}^{m \times m}$, die Unterblöcke von A, B und C , sodass

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

- (i) Zeigen Sie, dass die Berechnung von C mit dem Standardverfahren zur Berechnung des Produkts von Matrizen auf $\mathcal{O}(n^{\log_2 8})$ Multiplikationen führt.
(ii) Zeigen Sie, dass mit

$$\begin{aligned} M_1 &= (A_{11} + A_{22})(B_{11} + B_{22}), & M_2 &= (A_{21} + A_{22})B_{11}, \\ M_3 &= A_{11}(B_{12} - B_{22}), & M_4 &= A_{22}(B_{21} - B_{11}), \\ M_5 &= (A_{11} + A_{12})B_{22}, & M_6 &= (A_{21} - A_{11})(B_{11} + B_{12}), \\ M_7 &= (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

gilt

$$\begin{aligned} C_{11} &= M_1 + M_4 - M_5 + M_7, & C_{12} &= M_3 + M_5, \\ C_{21} &= M_2 + M_4, & C_{22} &= M_1 - M_2 + M_3 + M_6. \end{aligned}$$

- (iii) Sei $n = 2^k$ für ein $k \in \mathbb{N}$. Konstruieren Sie ein rekursives Verfahren zur Berechnung von AB , das $\mathcal{O}(7^k) = \mathcal{O}(n^{\log_2 7})$ Multiplikationen verwendet.

Aufgabe 29.4.10

- (i) Für $a_\ell, b_\ell \in \mathbb{R}$, $\ell = 0, 1, \dots, m$, sei

$$T(x) = \frac{a_0}{2} + \sum_{\ell=1}^m (a_\ell \sin(\ell x) + b_\ell \cos(\ell x)).$$

Konstruieren Sie $\delta_k \in \mathbb{C}$, $k = 0, 1, \dots, 2m$, sodass mit

$$q(x) = \sum_{k=-m}^m \delta_{k+m} e^{ikx}$$

gilt $T(x) = q(x)$ für alle $x \in [0, 2\pi]$.

- (ii) Zeigen Sie, dass die Funktion q genau dann reellwertig ist, wenn $\delta_{m-k} = \overline{\delta}_{m+k}$ für $k = 0, 1, \dots, m$ gilt.

Projekt 29.4.1 Implementieren Sie die komplexe Fourier-Synthese als rekursive Funktion und verwenden Sie Ihre Routine, um die Fourier-Transformation der Vektoren $y \in \mathbb{C}^n$ definiert durch $y_j = f_r(2\pi j/n)$, $j = 0, 1, \dots, n-1$, $r = 1, 2, 3$, mit $f_1(x) = \sin(5x) + (1/2) \cos(x)$ sowie

$$f_2(x) = \begin{cases} 1, & x \in [\pi - 1/4, \pi + 1/4], \\ 0, & x \notin [\pi - 1/4, \pi + 1/4], \end{cases} \quad f_3(x) = \begin{cases} 1, & x \in [0, \pi), \\ -1, & x \notin [\pi, 2\pi), \end{cases}$$

mit $n = 2^s$, $s = 1, 2, \dots, 5$, zu berechnen. Stellen Sie die zugehörigen komplexen trigonometrischen Polynome grafisch dar. Nutzen Sie bei der Erstellung Ihres Programms die MATLAB-Realisierung komplexer Zahlen.

Projekt 29.4.2 Die Funktion $f : [0, 2\pi] \rightarrow \mathbb{R}$ sei definiert durch

$$f(x) = \begin{cases} x, & x \in [0, \pi], \\ 2\pi - x, & x \in [\pi, 2\pi]. \end{cases}$$

Verwenden Sie die MATLAB-Routine `fft`, um für $n = 2^s$, $s = 1, 2, \dots, 5$, komplexe Koeffizienten $(\beta_k)_{k=0,1,\dots,n-1}$ und $(\delta_k)_{k=0,1,\dots,n-1}$ zu berechnen, sodass für die Funktionen

$$p(x) = \sum_{k=0}^{n-1} \beta_k e^{ikx}, \quad q(x) = \sum_{k=-n/2}^{n/2-1} \delta_{k+n/2} e^{ikx}$$

die Interpolationseigenschaft $p(x_j) = f(x_j)$ beziehungsweise $q(x_j) = f(x_j)$ für $j = 0, 1, \dots, n-1$ und mit $x_j = 2\pi j/n$ erfüllt ist. Plotten Sie jeweils den Real- und Imaginärteil der Funktionen p und q und diskutieren Sie Ihre Ergebnisse.

29.5 Numerische Integration

Aufgabe 29.5.1 Verwenden Sie die Darstellung des Fehlers der Lagrange-Interpolation

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x - x_j),$$

um für die Trapez- beziehungsweise Simpson-Regel zu beweisen, dass

$$\begin{aligned}|I(f) - Q_{\text{Trap}}(f)| &\leq \frac{(b-a)^3}{12} \|f''\|_{C^0([a,b])}, \\ |I(f) - Q_{\text{Sim}}(f)| &\leq \frac{(b-a)^5}{2880} \|f^{(4)}\|_{C^0([a,b])}.\end{aligned}$$

Aufgabe 29.5.2 Sei $Q : C^0([a, b]) \rightarrow \mathbb{R}$ eine Quadraturformel mit $n + 1$ Gewichten und Quadraturpunkten $(x_i, w_i)_{i=0, \dots, n}$, die exakt vom Grad n ist.

- (i) Zeigen Sie, dass

$$w_i = \int_a^b L_i(x) dx$$

für $i = 0, 1, \dots, n$ mit den durch die Stützstellen $(x_i)_{i=0, \dots, n}$ definierten Lagrange-Basispolynomen $(L_i)_{i=0, \dots, n}$.

- (ii) Zeigen Sie, dass im Fall der Exaktheit vom Grad $2n$ gilt, dass $w_i > 0$ für $i = 0, 1, \dots, n$.

Aufgabe 29.5.3 Die Quadraturformel $Q : C^0([a, b]) \rightarrow \mathbb{R}$ sei exakt vom Grad $2q$ und die zugehörigen Gewichte $(w_i)_{i=0, \dots, n}$ und Knoten $(x_i)_{i=0, \dots, n}$ seien symmetrisch bezüglich dem Intervallmittelpunkt $(a+b)/2$ angeordnet. Zeigen Sie, dass Q exakt vom Grad $2q+1$ ist.

Aufgabe 29.5.4

- (i) Es sei $\omega \in C^0(a, b)$ eine Funktion, die uneigentlich Riemann-integrierbar und außerhalb endlich vieler Punkte positiv ist. Zeigen Sie, dass durch

$$\langle f, g \rangle_\omega = \int_a^b f(x)g(x)\omega(x) dx$$

ein Skalarprodukt auf $C^0([a, b])$ definiert wird.

- (ii) Zeigen Sie, dass die Polynome $(P_n)_{n \in \mathbb{N}}$ definiert durch die Ableitungen

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n]$$

bezüglich des durch die Gewichtsfunktion $\omega(x) = 1$ für $x \in [-1, 1]$ definierten Skalarprodukts orthogonal sind, das heißt für $j \neq k$ gilt $\langle P_j, P_k \rangle_\omega = 0$.

Aufgabe 29.5.5 Es sei $(f, g) \mapsto \langle f, g \rangle$ ein Skalarprodukt auf dem Raum $C^0([a, b])$. Zeigen Sie, dass mit den Initialisierungen $p_0(x) = 1$ und $p_1(x) = x - \beta_0$ sowie der Rekursionsvorschrift

$$p_{j+1}(x) = (x - \beta_j) p_j(x) - \gamma_j p_{j-1}(x)$$

mit den Koeffizienten $\beta_j = \langle x p_j, p_j \rangle / \langle p_j, p_j \rangle$ und $\gamma_j = \langle p_j, p_j \rangle / \langle p_{j-1}, p_{j-1} \rangle$ eine Folge von paarweise orthogonalen Polynomen $p_j \in \mathcal{P}_j$ definiert wird.

Aufgabe 29.5.6

- (i) Zeigen Sie, dass die Funktion $\omega(x) = (1 - x^2)^{-1/2}$ auf dem Intervall $(-1, 1)$ uneigentlich Riemann-integrierbar ist.
- (ii) Zeigen Sie, dass die Tschebyscheff-Polynome $T_n(t) = \cos(n \arccos(t))$, $n \in \mathbb{N}_0$, orthogonal bezüglich des durch die Gewichtsfunktion $\omega(x) = (1 - x^2)^{-1/2}$ definierten Skalarprodukts sind.

Aufgabe 29.5.7 Bestimmen Sie $n+1$ Quadraturpunkte und -gewichte im Intervall $[-1, 1]$, sodass die dadurch definierte Quadraturformel exakt vom Grad $2n + 1$ für $n = 0, 1, 2$ ist. Verwenden Sie die Formeln, um die Funktion $x \mapsto x^5$ im Intervall $[-1, 1]$ approximativ zu integrieren.

Aufgabe 29.5.8 Es sei $\omega : (a, b) \rightarrow \mathbb{R}$ eine Gewichtsfunktion. Konstruieren mit Hilfe des Gram–Schmidt-Verfahrens Polynome $(\pi_j)_{j=0, \dots, n}$ derart, dass $\pi_j \in \mathcal{P}_j$ für $j = 0, 1, \dots, n$, $\langle \pi_j, \pi_k \rangle_\omega = \delta_{jk}$ für alle $0 \leq j, k \leq n$ mit $j \neq k$, $\langle \pi_j, p \rangle_\omega = 0$ für alle $p \in \mathcal{P}_{j-1}$ und $j = 1, 2, \dots, n$ gilt und die Polynome eine Basis von \mathcal{P}_n bilden.

Aufgabe 29.5.9 Es sei $f \in C^0([a, b])$ und für eine Zerlegungsfeinheit $h = (b - a)/N$ sei $T(h)$ der Wert der summierten Trapezregel, das heißt

$$T(h) = \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{N-1} f(a + ih) + f(b) \right].$$

Zeigen Sie, dass die Extrapolation $T^*(h) = (T(h) - 2^\gamma T(h/2))/(1 - 2^\gamma)$ der Werte $T(h)$ und $T(h/2)$ mit einem geeigneten Parameter γ auf die summierte Simpson-Regel führt.

Aufgabe 29.5.10 Für $f \in C^\infty([a, b])$ und $h > 0$ sei $T(h) \in \mathbb{R}$ der Wert einer summierten Quadraturformel für die Zerlegungsfeinheit $h > 0$ mit Fehlerordnung $\mathcal{O}(h^\gamma)$. Konstruieren Sie mit Hilfe der Werte $T(h)$, $T(h/2)$ und $T(h/4)$ eine Zahl $T^*(h)$, die das Integral von f mit einem Fehler der Ordnung $\mathcal{O}(h^{3\gamma})$ approximiert.

Projekt 29.5.1 Verwenden Sie die summierten Trapez- und Simpson-Regeln sowie eine summierte Gaußsche 3-Punkt-Quadraturformel, um die Integrale im Intervall $[0, 1]$ der Funktionen

$$f(x) = \sin(\pi x)e^x, \quad g(x) = x^{1/3}$$

mit Schrittweiten $h = 2^{-\ell}$, $\ell = 1, 2, \dots, 10$, zu approximieren. Berechnen Sie jeweils den Fehler e_h und bestimmen Sie eine experimentelle Konvergenzrate γ aus dem Ansatz $e_h \approx c_1 h^\gamma$ und der daraus folgenden Formel

$$\gamma \approx \frac{\log(e_h/e_H)}{\log(h/H)}$$

für zwei aufeinanderfolgende Schrittweiten $h, H > 0$. Vergleichen Sie die experimentellen Konvergenzraten mit den theoretischen Konvergenzraten der Verfahren und kommentieren Sie Ihre Ergebnisse. Stellen Sie die Paare (h, e_h) für die verschiedenen Quadraturformeln vergleichend als Polygonzüge grafisch in logarithmischer Achenskalierung mit Hilfe des MATLAB-Befehls `loglog` dar.

Projekt 29.5.2

(i) Aus der Taylor-Formel ergibt sich, dass die Quotienten

$$d_h^+ f(x) = \frac{f(x+h) - f(x)}{h}, \quad \hat{d}_h f(x) = \frac{f(x-h) - f(x+h)}{2h}$$

für eine gegebene Schrittweite $h > 0$ Approximationen von $f'(x)$ mit der Fehlerordnung $\mathcal{O}(h)$ beziehungsweise $\mathcal{O}(h^2)$ definieren. Überprüfen Sie diese Eigenschaft experimentell am Beispiel $f(x) = \tan(x)$ für $x = 1/2$ mit den Schrittweiten $h = 2^{-\ell}$, $\ell = 1, 2, \dots, 15$.

(ii) Konstruieren Sie durch Extrapolation einen Quotienten $\hat{d}_h^* f(x)$, der die Ableitung $f'(x)$ bis auf einen Fehler der Ordnung $\mathcal{O}(h^4)$ approximiert und wiederholen Sie die Rechnungen. Welche Vor- und Nachteile besitzt die Approximation der Ableitung mittels Extrapolation?

29.6 Nichtlineare Probleme

Aufgabe 29.6.1

- (i) Berechnen Sie drei Schritte des Newton-Verfahrens für die Funktion $f(x) = \arctan(x)$ mit den Startwerten $x_0 = 1, 3/2, 2$.
- (ii) Wiederholen Sie die Rechnungen für das gedämpfte Newton-Verfahren $x_{k+1} = x_k - \omega f(x_k)/f'(x_k)$ mit den Dämpfungsparametern $\omega = 1/2, 3/4$.

Aufgabe 29.6.2 Sei $f \in C^1(\mathbb{R})$ konvex, das heißt für alle $x, y \in \mathbb{R}$ und $t \in [0, 1]$ gilt $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$, sowie streng monoton wachsend und sei $x^* \in \mathbb{R}$ mit $f(x^*) = 0$. Zeigen Sie, dass das Newton-Verfahren für jeden Startwert $x_0 \in \mathbb{R}$ konvergiert.

Aufgabe 29.6.3 Formulieren Sie hinreichende Bedingungen für die globale Konvergenz des gedämpften Newton-Verfahrens, indem Sie es als Fixpunktiteration mit der Abbildung $\Phi(x) = x - \omega Df(x)^{-1}f(x)$ betrachten.

Aufgabe 29.6.4

- (i) Seien $a, b \in \mathbb{R}$ mit $a < b$. Konstruieren Sie Punkte $c, d \in (a, b)$ mit $c < d$, sodass die Intervalle (a, d) und (c, b) dieselbe Länge haben.
- (ii) Formulieren Sie auf Basis der vorigen Konstruktion ein Intervallverkleinerungsverfahren für Minimalstellen, in dem nur eine Funktionsauswertung je Iterationsschritt notwendig ist und die Intervalllänge stets um denselben Faktor verringert wird.

Aufgabe 29.6.5

- (i) Seien $g \in C^1(\mathbb{R}^n)$, $x \in \mathbb{R}^n$ und $\sigma \in (0, 1)$. Zeigen Sie, dass eine Zahl $\alpha > 0$ existiert, sodass mit $d = -\nabla g(x) \neq 0$ gilt

$$g(x + \alpha d) < g(x) - \sigma \alpha \|d\|^2.$$

- (ii) Zeigen Sie, dass im Fall $g \in C^2(\mathbb{R}^n)$ die gleiche Aussage mit $\sigma = 1$ gilt.

Aufgabe 29.6.6 Das Verfahren von Heron approximiert die Quadratwurzel $a^{1/2}$ einer Zahl $a \geq 0$ durch die Iteration $x_{k+1} = \Phi(x_k)$ mit der Funktion $\Phi(x) = (x + a/x)/2$.

- (i) Zeigen Sie, dass Φ eine Kontraktion im Intervall $((a/2)^{1/2}, \infty)$ ist.
- (ii) Zeigen Sie, dass das Verfahren von Heron mit dem Newton-Verfahren für die Funktion $x \mapsto x^2 - a$ übereinstimmt und untersuchen Sie hinreichende Bedingungen für die lokale, quadratische Konvergenz des Verfahrens.
- (iii) Zeigen Sie, dass sich das Verfahren von Heron als Abstiegsverfahren für die Funktion $g(x) = x + a/x$ interpretieren lässt.

Aufgabe 29.6.7

- (i) Mit $f_0 = f_1 = 1$ sei die Folge der Fibonacci-Zahlen definiert durch $f_k = f_{k-1} + f_{k-2}$ für alle $k \geq 2$ und es sei α die positive Lösung der Gleichung $x^2 = 1 + x$. Zeigen Sie, dass $\alpha^{k-1} \leq f_k \leq \alpha^k$ für alle $k \geq 0$ gilt.
- (ii) Es sei $(e_k)_{k \in \mathbb{N}_0}$ eine Folge positiver reeller Zahlen, sodass $e_0, e_1 < 1$ und $e_{k+2} \leq e_{k+1}e_k$ für alle $k \geq 0$ gilt. Zeigen Sie, dass die Folge $(e_k)_{k \in \mathbb{N}_0}$ von einer Folge $(\delta_k)_{k \geq 0}$ dominiert wird, die von der Ordnung α gegen Null konvergiert, das heißt es gilt $e_k \leq \delta_k$ für alle $k \in \mathbb{N}$ und es existiert ein $q \in \mathbb{R}$ mit

$$\limsup_{k \rightarrow \infty} \delta_{k+1}/\delta_k^\alpha = q.$$

Aufgabe 29.6.8

- (i) Diskutieren Sie die Wohlgestelltheit des Sekanten-Verfahrens.
- (ii) Zeigen Sie, dass für die Approximationsfehler $e_k = x^* - x_k$ der Iterierten des Sekanten-Verfahrens die Relation

$$\frac{e_{k+1}}{e_k e_{k-1}} = \frac{g(x_k) - g(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

mit der Funktion $g(x) = -f(x)/(x - x^*)$ gilt, sofern beide Seiten wohldefiniert sind.

- (iii) Unter welchen Bedingungen ist die rechte Seite in der Identität für $e_{k+1}/(e_k e_{k-1})$ beschränkt und was lässt sich über die Konvergenz des Verfahrens folgern?

Aufgabe 29.6.9 Zeigen Sie, dass das Polynom $p(x) = x^3 - 2x^2 - 1$ genau eine Nullstelle $x^* \geq 2$ besitzt und rechtfertigen Sie damit die Fixpunktgleichung $\Phi(x^*) = x^*$ mit $\Phi(x) = 2 + 1/x^2$. Beweisen Sie, dass Φ eine Kontraktion auf $[2, \infty) \subset \mathbb{R}$ ist, und berechnen Sie drei Schritte der Fixpunktiteration. Welche Genauigkeit liegt nach 3 Schritten vor? Wie viele Schritte benötigt man, um eine Genauigkeit von 10^{-6} zu erreichen?

Aufgabe 29.6.10 Für gegebene $g \in C^2(\mathbb{R}^n)$, $x_k \in \mathbb{R}^n$ und $H_k \in \mathbb{R}^{n \times n}$ definiere $q_k : \mathbb{R}^n \rightarrow \mathbb{R}$, $d \mapsto g(x_k) + \nabla g(x_k) \cdot d + (1/2)d^\top H_k d$.

- (i) Geben Sie hinreichende Bedingungen für die Existenz einer eindeutigen Minimalstelle $d_k \in \mathbb{R}^n$ von q_k an.
- (ii) Zeigen Sie, dass die Iteration $x_{k+1} = x_k + d_k$ dem Newton-Verfahren und dem Abstiegsverfahren mit fester Schrittweite $\alpha_k = \alpha$ entspricht, sofern $H_k = D^2g(x_k)$ beziehungsweise $H_k = \alpha I$ verwendet wird.
- (iii) Interpretieren Sie die Iteration geometrisch.

Projekt 29.6.1

- (i) Untersuchen Sie experimentell die Konvergenz des Verfahrens von Heron zur Berechnung einer Quadratwurzel, das heißt der Iterationsvorschrift $x_{k+1} = (x_k + a/x_k)/2$, für $a = 3/2$ und verschiedene Startwerte $x_0 \in \mathbb{R}$.
- (ii) Wiederholen Sie 10^8 mal die Ausführung der Befehle `sqrt(a)` und `(a^0.5)` oder `pow(a, 0.5)` mit $a = 3/2$ und diskutieren Sie Gründe für mögliche Unterschiede der Laufzeiten.
- (iii) Für eine holomorphe Funktion $f : \mathbb{C} \rightarrow \mathbb{C}$ mit Nullstellen $z_1, z_2, \dots, z_n \in \mathbb{C}$ kann die komplexe Ebene in Einzugsbereiche $E_j \subset \mathbb{C}$, die für $j = 1, 2, \dots, n$, durch

$$E_j = \{z \in \mathbb{C} : \text{Newton-Verfahren mit Startwert } z \text{ konvergiert gegen } z_j\}$$

definiert sind, sowie die Restmenge $X = \mathbb{C} \setminus \cup_{j=1}^n E_j$, partitioniert werden. Betrachten Sie die Funktion $f(z) = z^3 - 1$ und verwenden Sie als Startwerte Gitterpunkte $z_\ell = x_\ell + iy_\ell$ im Bereich $[-1, 1]^2 \subset \mathbb{C}^2$, die im Abstand $h = 1/200$ angeordnet sind. Markieren Sie die Punkte unterschiedlich entsprechend der Zugehörigkeit zum Einzugsbereich einer Nullstelle und stellen Sie diese grafisch dar. Verwenden Sie dazu die in Abb. 29.1 gezeigten MATLAB-Befehle mit einer geeignet definierten Matrix C.

Projekt 29.6.2

- (i) Implementieren Sie das Newton- und das Sekanten-Verfahren zur Nullstellensuche einer Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ in MATLAB und testen Sie es mit der Funktion $f(x) = \exp(x) + x^2 - 2$, dem Startwert $x_0 \in \{-1, 0, 1\}$ und dem Abbruchkriterium $|x_{k+1} - x_k| \leq 10^{-12}$. Beenden Sie das Newton-Verfahren bei Nichterreichen des Abbruchkriteriums mit 100 Iterationen. Vergleichen Sie die Iterationszahlen sowie die Anzahl der von Schritt zu Schritt beibehaltenen Nachkommastellen.
- (ii) Realisieren Sie die Nullstellenbestimmung von f durch ein Abstiegsverfahren für die Funktion $g(x) = |f(x)|^2$ und vergleichen Sie die Konvergenzgeschwindigkeit mit der des Newton-Verfahrens.
- (iii) Verwenden Sie das Newton-Verfahren, um eine Nullstelle der Abbildung

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad (x_1, x_2, x_3) \mapsto (x_1^2 + x_2^2 - e, 3x_2 + 4x_3 - \sqrt{5}, x_1^2 - \pi/4)$$

zu approximieren. Wie lässt sich die Lösbarkeit beurteilen und ein sinnvoller Startwert konstruieren?

```
[X, Y] = meshgrid(-1:h:1, -1:h:1);
scatter(X(:, ), Y(:, ), 15, C(:, ));
```

Abb. 29.1 Darstellung von unterschiedlich gefärbten Punkten

29.7 Methode der konjugierten Gradienten

Aufgabe 29.7.1 Für $A \in \mathbb{R}^{n \times n}$ und $x, y \in \mathbb{R}^n$ seien $\|x\|_A = (x \cdot (Ax))^{1/2}$ und $\langle x, y \rangle_A = (Ax) \cdot y$. Zeigen Sie, dass $(x, y) \mapsto \langle x, y \rangle_A$ genau dann ein Skalarprodukt definiert, das die Norm $\|\cdot\|_A$ induziert, wenn A symmetrisch und positiv definit ist.

Aufgabe 29.7.2 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit mit Eigenwerten $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Zeigen Sie, dass für alle $x \in \mathbb{R}^n \setminus \{0\}$ die Ungleichung

$$\frac{(x \cdot Ax)(x \cdot A^{-1}x)}{\|x\|^4} \leq \frac{(\lambda_1^{-1} + \lambda_n)^2}{4\lambda_1\lambda_n}$$

gilt. Betrachten Sie dazu zunächst den Fall $\lambda_1\lambda_n = 1$, verwenden Sie die Diagonalisierung $A = Q^\top D Q$ und benutzen Sie die elementare Ungleichung $ab \leq (a+b)^2/4$.

Aufgabe 29.7.3 Sei $b \in \mathbb{R}^n$, sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und sei $\phi(x) = (A^{-1}(b - Ax)) \cdot (b - Ax)$ für alle $x \in \mathbb{R}^n$. Für eine Approximation $\tilde{x} \in \mathbb{R}^n$ wird beim Abstiegsverfahren die Suchrichtung $\tilde{d} = -\nabla\phi(\tilde{x})$ verwendet.

- (i) Zeigen Sie, dass $\tilde{d} = b - A\tilde{x}$ gilt und bestimmen Sie die Minimalstelle $\tilde{\alpha}$ der Funktion $t \mapsto \phi(\tilde{x} + t\tilde{d})$.
- (ii) Zeigen Sie, dass mit dem optimalen $\tilde{\alpha}$ und $\tilde{x}^{\text{neu}} = \tilde{x} + \tilde{\alpha}\tilde{d}$ gilt

$$\|\tilde{x}^{\text{neu}} - x^*\|_A^2 = \|\tilde{x} - x^*\|_A^2 \left(1 - \frac{\|\tilde{d}\|^4}{(\tilde{d} \cdot A\tilde{d})(\tilde{d} \cdot A^{-1}\tilde{d})}\right).$$

(iii) Sei $\kappa = \text{cond}_2(A) = \lambda_{\min}^{-1}\lambda_{\max}$ die Konditionszahl von A . Verwenden Sie ohne Beweis die für alle $x \in \mathbb{R}^n \setminus \{0\}$ gültige Abschätzung

$$\frac{(x \cdot Ax)(x \cdot A^{-1}x)}{\|x\|^4} \leq \frac{(\lambda_{\min}^{-1} + \lambda_{\max})^2}{4\lambda_{\min}\lambda_{\max}},$$

um zu beweisen, dass

$$\|\tilde{x}^{\text{neu}} - x^*\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|\tilde{x} - x^*\|_A.$$

Aufgabe 29.7.4

- (i) Zeigen Sie, dass sich die Funktion $T_k(t) = \cos(k \arccos t)$, $t \in [-1, 1]$, eindeutig als Polynom auf \mathbb{R} fortsetzen lässt und für $|t| \geq 1$ gilt

$$T_k(t) = \frac{1}{2} \left(t + (t^2 - 1)^{1/2} \right)^k + \frac{1}{2} \left(t - (t^2 - 1)^{1/2} \right)^k.$$

- (ii) Zeigen Sie, dass für alle $s > 1$ gilt

$$\frac{1}{2} \left(\frac{s^{1/2} + 1}{s^{1/2} - 1} \right)^k \leq T_k \left(\frac{s+1}{s-1} \right) \leq \left(\frac{s^{1/2} + 1}{s^{1/2} - 1} \right)^k.$$

Aufgabe 29.7.5 Seien $0 < a < b$ und $k \geq 0$. Zeigen Sie, dass das Problem

$$\min \left\{ \max_{t \in [a,b]} |p(t)| : p \in \mathcal{P}_k, p(0) = 1 \right\}$$

die eindeutige Lösung

$$q(t) = T_k \left(\frac{a+b-2t}{b-a} \right) / T_k \left(\frac{a+b}{b-a} \right)$$

besitzt, wobei T_k das k -te Tschebyscheff-Polynom sei.

Hinweis: Nehmen Sie an, dass die Aussage falsch ist und betrachten Sie die Nullstellen und Extremwerte der Differenz $r = q - p$ für ein geeignetes Polynom $p \in \mathcal{P}_k$.

Aufgabe 29.7.6 Verwenden Sie das CG-Verfahren, um eine Lösung des linearen Gleichungssystems $Ax = b$ definiert durch

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

zu bestimmen. Starten Sie mit $x_0 = [1, 0, 1, 0]^\top$, berechnen Sie den Krylov-Raum $\mathcal{K}_2(A, r_0) = \text{span}\{r_0, Ar_0\}$ und vergleichen Sie diesen mit dem Raum $\text{span}\{d_0, d_1\}$.

Aufgabe 29.7.7

- (i) Leiten Sie aus bekannten Aussagen ab, dass für die mit dem Abstiegs- und dem CG-Verfahren berechneten Approximationslösungen $(x_k)_{k=0,1,\dots}$ jeweils eine Abschätzung

$$\|x^* - x_k\|_A \leq cq^k \|x^* - x_0\|_A$$

mit $c = 1$, $q = 1 - 2 \text{cond}_2(A)^{-1} + 2\xi$ und $c = 2$, $q = 1 - 2 \text{cond}_2(A)^{-1/2} + 2\xi$ mit Zahlen $0 \leq \xi \leq \text{cond}_2(A)^{-2}$ beziehungsweise $0 \leq \zeta \leq \text{cond}_2(A)^{-1}$ gilt.

- (ii) Zeigen Sie, dass $\log(1+s) \approx s$ für $|s| \ll 1$ gilt.
 (iii) Für $\varepsilon_{\text{stop}} > 0$ sei $M_\varepsilon = |\log(\varepsilon_{\text{stop}})|$. Folgern Sie, dass mit dem Abstiegs- und dem CG-Verfahren etwa $M_\varepsilon \text{cond}(A)$ beziehungsweise $M_\varepsilon \text{cond}(A)^{1/2}$ viele Iterationen benötigt werden, um das Abbruchkriterium $\|x^* - x_k\|_A \leq \varepsilon_{\text{stop}}$ zu erfüllen, wenn $\text{cond}_2(A) \gg 1$ gilt.

Aufgabe 29.7.8 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Für A -konjugierte Vektoren $d_0, d_1, \dots, d_{k-1} \in \mathbb{R}^n \setminus \{0\}$ und $b \in \mathbb{R}^n$ sei $f : \mathbb{R}^k \rightarrow \mathbb{R}$ definiert durch

$$f(\alpha_0, \alpha_1, \dots, \alpha_{k-1}) = \frac{1}{2} \left\| b - A \left(x_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) \right\|_{A^{-1}}^2.$$

Berechnen Sie $\nabla f(\alpha_0, \alpha_1, \dots, \alpha_{k-1})$.

Aufgabe 29.7.9 Modifizieren Sie das Gram–Schmidtsche Orthogonalisierungsverfahren, um für eine gegebene symmetrische und positiv definite Matrix $A \in \mathbb{R}^{n \times n}$ eine Familie $(d_i : i = 0, 1, \dots, n-1)$ nichtverschwindender A -konjugierter Vektoren zu bestimmen.

Aufgabe 29.7.10 Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Für $x \in \mathbb{R}^n$ sei $\phi(x) = \|b - Ax\|_{A^{-1}}^2/2$ und $x^* \in \mathbb{R}^n$ erfülle $Ax^* = b$.

- (i) Beweisen Sie $\phi(x) - \phi(x^*) = \|x - x^*\|_A^2/2$ und $\nabla \phi(x) = -(b - Ax)$.
- (ii) Zeigen Sie, dass $d = -\nabla \phi(x)$ orthogonal zur Niveaumenge $N_a \phi = \{y \in \mathbb{R}^n : \phi(y) = a\}$ zum Niveau $a = \phi(x)$ im Punkt x ist, das heißt für jede C^1 -Kurve $c : (-\varepsilon, \varepsilon) \rightarrow \mathbb{R}^n$ mit $c(t) \in N_a \phi$ für alle $t \in (-\varepsilon, \varepsilon)$ und $c(0) = x$ gilt $c'(0) \cdot d = 0$.

Projekt 29.7.1 Implementieren Sie das CG- und das Abstiegsverfahren zur approximativen Lösung des Systems $Ax = b$. Vergleichen Sie die Anzahl der benötigten Iterations schritte der beiden Verfahren am Beispiel

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad b = h^2 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n$$

mit $n = 10^s$, $s = 1, 2, \dots, 5$, und $h = 2/(n+1)$. Wählen Sie das Abbruchkriterium $\|b - Ax_k\| \leq h$ und als Startwert $x_0 = [0, 0, \dots, 0]^\top \in \mathbb{R}^n$. Berechnen Sie für jeweils zwei aufeinanderfolgende Residuen den Quotienten der Normen, geben Sie diese in einer Tabelle aus und kommentieren Sie Ihre Ergebnisse. Visualisieren Sie die numerische Lösung grafisch mittels `plot([-1:h:1], [0, x', 0])`. Die Kurve sollte eine Funktion $u : [-1, 1] \rightarrow \mathbb{R}$ approximieren, für die $-u'' = 1$ und $u(-1) = u(1) = 0$ gilt.

Projekt 29.7.2 Für $n \geq 1$ ist die Hilbert-Matrix $H \in \mathbb{R}^{n \times n}$ definiert durch die Einträge $h_{ij} = 1/(i+j-1)$. Die Matrix H ist symmetrisch und positiv definit aber schlecht konditioniert. In MATLAB kann sie mit dem Befehl `hilb(n)` generiert werden.

- (i) Verwenden Sie die MATLAB-Routine `cond`, um die Konditionszahl der Hilbert-Matrix für $n = 10^s$, $s = 1, 2, \dots, 3$, approximativ zu bestimmen und experimentell zu überprüfen, dass $\text{cond}(H) = \mathcal{O}((1 + \sqrt{2})^{4n} / \sqrt{n})$ gilt.
- (ii) Implementieren Sie das CG-Verfahren und verwenden Sie es, um die Gleichungssysteme $Hx = b$ mit $b_i = \sum_{j=1}^n h_{ij}$, $i = 1, 2, \dots, n$, für $n = 10^s$, $s = 1, 2, \dots, 4$, mit dem Anfangsvektor $x = [0, 0, \dots, 0]^\top \in \mathbb{R}^n$ zu lösen und zu beurteilen, inwiefern die Konvergenzaussage für das CG-Verfahren scharf ist.

29.8 Dünnbesetzte Matrizen und Vorkonditionierung

Aufgabe 29.8.1 Zeigen Sie, dass wenn $A \in \mathbb{R}^{n \times n}$ eine Bandmatrix mit Bandweite $w \in \mathbb{N}$ ist, also $a_{ij} = 0$ für $|i - j| > w$ gilt, auch die Faktoren der LU- und Cholesky-Zerlegungen Bandmatrizen der Bandweite w sind, sofern diese existieren.

Aufgabe 29.8.2

- (i) Seien $A, B \in \mathbb{R}^{n \times n}$ dünnbesetzte Matrizen und $b \in \mathbb{R}^n$. Konstruieren Sie möglichst effiziente Algorithmen zur Berechnung von AB und Ab und bestimmen Sie deren Aufwand.
- (ii) Zeigen Sie, dass das Produkt zweier dünnbesetzter Matrizen im Allgemeinen nicht dünnbesetzt ist.

Aufgabe 29.8.3 Sei $A \in \mathbb{R}^{n \times n}$ mit $n = w^2$ für ein $w \in \mathbb{N}$ definiert durch

$$a_{ij} = \begin{cases} 8, & |i - j| = 0, \\ 1, & |i - j| \in \{1, w\}. \end{cases}$$

- (i) Zeigen Sie, dass A eine dünnbesetzte Bandmatrix ist.
- (ii) Zeigen Sie, dass A eine Cholesky-Zerlegung besitzt, deren Faktoren nicht dünnbesetzt sind.

Aufgabe 29.8.4 Stellen Sie die Matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 3 & 4 \\ 0 & 2 & 5 & 0 & 1 \\ 4 & 0 & 0 & 1 & 3 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 7 & 6 & 0 \end{bmatrix}$$

im Koordinaten- und CCS-Format dar und berechnen Sie Ax für $x = [1, 2, \dots, 5]^\top$ mit Hilfe der Koordinatenvektoren.

Aufgabe 29.8.5 Seien $A, C \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit.

- (i) Zeigen Sie, dass das Produkt CA im Allgemeinen weder symmetrisch noch positiv definit ist.
- (ii) Zeigen Sie, dass CA positiv definit bezüglich des Skalarprodukts $(x, y) \mapsto (Cx) \cdot y$ ist.

Aufgabe 29.8.6 Sei $A \in \mathbb{R}^{n \times n}$ durch $a_{ii} = 2$ für $i = 1, 2, \dots, n$ und $a_{ij} = -1$ für $i, j = 1, 2, \dots, n$ mit $|i - j| = 1$ definiert und sei $b = [1, 1, \dots, 1]^\top \in \mathbb{R}^n$. Führen Sie für $n = 5$ so viele Iterationen des Gauß-Seidel-Verfahrens durch, bis sich die ersten zwei Nachkommastellen der Einträge des Lösungsvektors nicht mehr ändern. Nutzen Sie die Dünnbesetzung der Matrix A aus, um die Matrix-Vektor-Multiplikationen möglichst effizient durchzuführen.

Aufgabe 29.8.7 Seien $A, C \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und sei $C = LL^\top$ die Cholesky-Zerlegung von C . Zeigen Sie, dass die Matrix $L^\top AL$ symmetrisch und positiv definit ist und formulieren Sie das CG-Verfahren für das Gleichungssystem $L^\top ALy = L^\top b$.

Aufgabe 29.8.8 Seien $A, C \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit und $b \in \mathbb{R}^n$. Sei ferner $C = LL^\top$ die Cholesky-Zerlegung von C . Zeigen Sie, dass das vorkonditionierte CG-Verfahren der Anwendung des CG-Verfahrens auf das Gleichungssystem $L^\top AL(L^{-1}x) = L^\top b$ entspricht.

Aufgabe 29.8.9

- (i) Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Bestimmen Sie eine Cholesky-Zerlegung $C_{SGS} = VV^\top$ der symmetrischen Gauß-Seidel-Vorkonditionierungsma-
trix $C_{SGS} = [(L + D)D^{-1}(D + L)]^{-1}$ mit der Zerlegung $A = L + D + L^\top$ von A in Diagonal- und unteren sowie oberen Anteil.
- (ii) Zeigen Sie, dass $A - C_{SGS}^{-1} = -LD^{-1}L^\top$ gilt.
- (iii) Berechnen Sie die Differenz $A - C_{SGS}^{-1}$ für

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

Aufgabe 29.8.10 Seien $A, M \in \mathbb{R}^{n \times n}$ regulär mit der Eigenschaft, dass $\|I - MA\| = \delta < 1$ bezüglich einer geeigneten Operatornorm auf $\mathbb{R}^{n \times n}$ gilt. Zeigen Sie, dass die Abschätzungen $\|MA\| \leq 1 + \delta$ und $\|(MA)^{-1}\| \leq 1/(1 - \delta)$ gelten und folgern Sie $\text{cond}(MA) \leq (1 + \delta)/(1 - \delta)$.

Projekt 29.8.1 Implementieren Sie das vorkonditionierte CG-Verfahren und testen Sie es für das Gleichungssystem $Ax = b$, wobei $A \in \mathbb{R}^{n \times n}$ mit $n = m^2$ und $T_m \in \mathbb{R}^{m \times m}$ definiert sei durch

$$A = \begin{bmatrix} T_m & -I_m & & \\ -I_m & \ddots & \ddots & \\ & \ddots & \ddots & -I_m \\ & & -I_m & T_m \end{bmatrix}, \quad T_m = \begin{bmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix},$$

```
U = zeros(m+2,m+2); U(2:m+1,2:m+1) = reshape(x,m,m)';
dx = 1/(m+1); mesh(0:dx:1,0:dx:1,U);
```

Abb. 29.2 Plotten einer durch eine Matrix U definierten Funktion

und $b \in \mathbb{R}^n$ gegeben sei durch $b = (m+1)^{-2}[1, 1, \dots, 1]^\top$. Verwenden Sie dabei die Vorkonditionierungen durch Zeilenäquilibrierung, unvollständige Cholesky-Zerlegungen verschiedener Bandweiten und die symmetrische Gauß-Seidel-Vorkonditionierung. Vergleichen Sie die Iterationszahlen für $m = 2^s \cdot 10$, $s = 0, 1, \dots, 4$, und den Abbruchparameter $\varepsilon_{\text{stop}} = (m+1)^{-2}/10$. Visualisieren Sie die Lösung $x \in \mathbb{R}^{m^2}$ des Gleichungssystems mittels der in Abb. 29.2 gezeigten Befehle. Es sollte eine glatte Funktion im Gebiet $(0, 1)^2$ dargestellt werden, die auf dem Rand verschwindet.

Projekt 29.8.2

- (i) Definieren Sie in MATLAB die Matrizen $A = \text{eye}(n)$ und $B = \text{speye}(n)$ und berechnen Sie $A*x$ sowie $B*x$ für $x = \text{ones}(n, 1)$. Messen Sie dabei die benötigte Zeit für die Dimensionen $n = 10^s$, $s = 1, 2, \dots, 5$. Erklären Sie mögliche Unterschiede.
- (ii) Konstruieren Sie mit Hilfe der MATLAB-Kommandos `sparse` und `spdiags` die Bandmatrix $A \in \mathbb{R}^{n \times n}$ mit $n = w^2$ für $w \in \mathbb{N}$ und $a_{ii} = 8$ sowie $a_{ij} = 1$ für $|i - j| \in \{1, w\}$. Überprüfen Sie die Besetzungsstruktur der Matrix mit Hilfe des Befehls `spy(A)` für verschiedene Zahlen w . Lösen Sie das Gleichungssystem $Ax = b$ mit $b = [1, 1, \dots, 1]^\top$ für $w = 10^2$ und wiederholen Sie dies, nachdem Sie den Befehl `A = full(A)` ausgeführt haben. Kommentieren Sie Ihre Beobachtungen.

29.9 Mehrdimensionale Approximation

Aufgabe 29.9.1 Zeigen Sie, dass das durch $z_0, z_1, \dots, z_d \in \mathbb{R}^d$ definierte Simplex $T = \text{conv}\{z_0, z_1, \dots, z_d\}$ genau dann nichtentartet ist, wenn die Vektoren $z_i - z_0$ für $i = 1, 2, \dots, d$ linear unabhängig sind und in diesem Fall ist das Volumen durch den Betrag von $\det[z_1 - z_0, z_2 - z_0, \dots, z_d - z_0]/d!$ gegeben.

Aufgabe 29.9.2 Zeigen Sie, dass mit dem Referenzdreieck $\hat{T} = \text{conv}\{0, e_1, e_2\}$ für $j, k \geq 0$ gilt

$$\int_{\hat{T}} s^j t^k \, d(s, t) = \frac{j!k!}{(j+k+2)!}$$

und folgern Sie die Exaktheit vom partiellen Grad 2 der durch

$$\hat{\xi} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix}^\top, \quad \hat{w} = \frac{1}{6}[1, 1, 1]^\top$$

definierten Quadraturformel.

Aufgabe 29.9.3 Für ein nichtentartetes Simplex $T \subset \mathbb{R}^d$ und Funktionen $f, g \in C^1(T)$ seien $\hat{f}, \hat{g} \in C^1(\hat{T})$ auf dem Referenzsimplex $\hat{T} \subset \mathbb{R}^d$ definiert durch $\hat{f} = f \circ \Phi_T$ und $\hat{g} = g \circ \Phi_T$ mit einem affin-linearen Diffeomorphismus $\Phi_T : \hat{T} \rightarrow T$. Zeigen Sie, dass

$$\int_T \nabla f \cdot \nabla g \, dx = \det \hat{D}\Phi_T \int_{\hat{T}} \hat{\nabla} \hat{f} \cdot (\hat{D}\Phi \hat{D}\Phi^\top)^{-1} \hat{\nabla} \hat{g} \, d\hat{x}.$$

Dabei seien $\hat{\nabla}$ und \hat{D} die Differenzialoperatoren bezüglich der Koordinaten in \hat{T} .

Aufgabe 29.9.4 Für $n \in \mathbb{N}$ seien $\omega^k = (e^{ijk2\pi/n})_{j=0,\dots,n-1} \in \mathbb{C}^n$ für $k = 0, 1, \dots, n$ und $T_n = (e^{ijk2\pi/n})_{j,k=0,\dots,n-1} \in \mathbb{C}^{n \times n}$.

- (i) Zeigen Sie, dass durch $E^{k\ell} = \omega^k(\omega^\ell)^\top$ für $k, \ell = 0, 1, \dots, n-1$ bezüglich des durch $E : F = \sum_{j_1, j_2=0}^{n-1} E_{j_1 j_2} \overline{F}_{j_1 j_2}$ definierten Matrizen-Skalarprodukts eine Orthogonalbasis definiert wird, indem Sie $E : F = \text{tr}(E \overline{F}^\top)$ nachweisen.
- (ii) Zeigen Sie, dass für jede Matrix $Y \in \mathbb{C}^{n \times n}$ und $B = (b_{k\ell})_{k,\ell=0,\dots,n-1} = T_n Y T_n$ gilt

$$Y = \sum_{k,\ell=0}^{n-1} b_{k\ell} E^{k\ell}.$$

Aufgabe 29.9.5 Für die Matrix $F = (f_{jk})_{j,k=0,1} \in \mathbb{R}^{2 \times 2}$ sei der Vektor $f \in \mathbb{R}^4$ definiert durch $f = [f_{00}, f_{01}, f_{10}, f_{11}]^\top \in \mathbb{R}^4$. Zeigen Sie, dass die zweidimensionale Fourier-Transformierte von F und die eindimensionale Fourier-Transformierte von f zu unterschiedlichen Resultaten führen.

Aufgabe 29.9.6 Für ein nichtentartetes Simplex $T = \text{conv}\{z_0, z_1, \dots, z_d\}$ und $i = 0, 1, \dots, d$ sei $\varphi_i : T \rightarrow \mathbb{R}$ die affin-lineare Funktion mit der Eigenschaft $\varphi_i(z_j) = \delta_{ij}$, $j = 0, 1, \dots, d$. Zeigen Sie, dass

$$\varphi_i(x) = \frac{\det \begin{bmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ z_0 & \dots & z_{i-1} & x & z_{i+1} & \dots & z_n \end{bmatrix}}{\det \begin{bmatrix} 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ z_0 & \dots & z_{i-1} & z_i & z_{i+1} & \dots & z_n \end{bmatrix}}.$$

Aufgabe 29.9.7 Es sei (\mathcal{T}_n) eine Folge von Triangulierungen des Gebiets $\Omega \subset \mathbb{R}^2$ mit maximalen Netzweiten $h_n > 0$, für die $h_n \rightarrow 0$ gilt. Zudem gelte für alle Innenwinkel α der Dreiecke in \mathcal{T}_n die Abschätzung $\alpha \geq \alpha_0 > 0$ mit einer von n unabhängigen Konstanten α_0 . Zeigen Sie, dass eine von n unabhängige Konstante $K \in \mathbb{N}$ existiert, sodass jedes Dreieck in \mathcal{T}_n höchstens K Nachbarn besitzt.

Aufgabe 29.9.8 Es seien \mathcal{T}_h eine Triangulierung von $\Omega \subset \mathbb{R}^d$, $f \in C^1(\overline{\Omega})$ und $\mathcal{I}_h f \in S^{1,0}(\mathcal{T}_h)$ der nodale Interpolant von f . Zeigen Sie, dass

$$\|f - \mathcal{I}_h f\|_{C^0(\overline{\Omega})} \leq h \|\nabla f\|_{C^0(\overline{\Omega})}.$$

Aufgabe 29.9.9 Es sei $Q : C^0([0, 1]) \rightarrow \mathbb{R}$ eine Quadraturformel mit nichtnegativen Gewichten und Punkten $(w_i, t_i)_{i=0, \dots, n}$, mit Exaktheitsgrad $k \geq 0$ und $Q^d : C^0([0, 1]^d) \rightarrow \mathbb{R}$ definiert durch

$$Q^d(f) = \sum_{i_1=0}^n \sum_{i_2=0}^n \cdots \sum_{i_d=0}^n w_{i_1} w_{i_2} \cdots w_{i_d} f(t_{i_1}, t_{i_2}, \dots, t_{i_d}).$$

Zeigen Sie für den Fall $d = 3$, dass

$$|I^d(f) - Q^d(f)| \leq \sum_{i=1}^d \sup_{\hat{x}_i \in [0, 1]^{d-1}} |If_{\hat{x}_i} - Qf_{\hat{x}_i}|,$$

wobei $f_{\hat{x}_i}$ für $\hat{x}_i = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \in [0, 1]^{d-1}$ die Abbildung

$$t \mapsto f(x_1, \dots, x_{i-1}, t, x_{i+1}, \dots, x_d)$$

bezeichne.

Aufgabe 29.9.10 Es sei $T = \text{conv}\{z_0, z_1, \dots, z_d\} \subset \mathbb{R}^d$, $d \in \{2, 3\}$, ein nichtentartetes Simplex und $\varphi_0 : T \rightarrow \mathbb{R}$ die der Ecke z_0 zugeordnete Hutfunktion. Weiter sei S_0 die dem Knoten z_0 gegenüberliegende Seite des Dreiecks oder Tetraeders und n_0 die äußere Einheitsnormale zu T auf S_0 . Zeigen Sie, dass

$$\nabla \varphi_0 = \frac{-|S_0|}{d|T|} n_0$$

mit dem Flächeninhalt beziehungsweise Volumen $|T|$ von T und der Länge beziehungsweise dem Flächeninhalt $|S_0|$ von S_0 gilt.

Projekt 29.9.1 Für $d \in \mathbb{N}$ und eine Funktion $f \in C^0([0, 1]^d)$ soll ihr Integral auf dem Würfel $[0, 1]^d$ numerisch bestimmt werden.

- (i) Schreiben Sie eine Routine, die die iterierte Trapezformel Q_{Trap}^d realisiert und testen Sie diese für $d = 5$ und die Funktionen

$$f_1(x) = \prod_{i=1}^d x_i^2, \quad f_2(x) = \sin(x_1 x_2 \dots x_d).$$

Überprüfen Sie die Konvergenzordnung im Fall von f_1 und bestimmen Sie den Aufwand der Berechnung.

- (ii) Für uniform und unabhängig verteilte Zufallsvariablen $\xi^1, \xi^2, \dots, \xi^N \in [0, 1]^d$ wird eine *Monte-Carlo-Quadraturformel* definiert durch

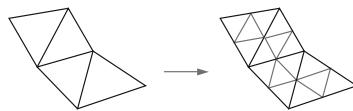
$$Q_{MC}^N(f) = \frac{1}{N} \sum_{i=1}^N f(\xi^i).$$

Man kann zeigen, dass der Erwartungswert von $|I^d(f) - Q_{MC}^N(f)|$ von der Ordnung $\mathcal{O}(N^{-1/2})$ ist. Überprüfen Sie mit den obigen Beispielen dieses Konvergenzverhalten und bestimmen Sie den Aufwand der Berechnung von $Q_{MC}^N(f)$. Realisierungen geeigneter Pseudo-Zufallsvariablen können Sie mit dem MATLAB-Befehl `rand(d, 1)` erzeugen.

- (iii) Diskutieren Sie, in welchen Situationen die Verwendung einer iterierten oder einer Monte-Carlo-Quadraturformel vorteilhaft ist.

Projekt 29.9.2 Ein gängiges Format zur Abspeicherung von Triangulierungen besteht aus einer Liste $Z \in \mathbb{R}^{N \times d}$ mit den Koordinaten der Knoten $z_1, z_2, \dots, z_N \in \mathbb{R}^d$, wodurch auch eine Numerierung der Knoten definiert wird, und einer Liste $T \in \mathbb{R}^{L \times (d+1)}$, die die Nummern der Knoten der einzelnen Dreiecke oder Tetraeder T_1, T_2, \dots, T_L enthält.

Abb. 29.3 Verfeinerung einer Triangulierung durch Bisektion der Kanten



- (i) Schreiben Sie ein Programm, das eine uniforme Verfeinerung einer gegebenen Triangulierung eines zweidimensionalen Gebiets im obigen Format durchführt. Dabei soll jedes Dreieck wie in Abb. 29.3 durch Bisektion seiner Seiten in vier kongruente Teildreiecke zerlegt werden. Testen Sie Ihre Routine an zwei einfachen Beispielen. Sie können Triangulierungen in MATLAB mit dem Kommando `trimesh(T, Z(:, 1), Z(:, 2))` visualisieren.
- (ii) Implementieren Sie für eine gegebene Triangulierung eines Gebiets $\Omega \subset \mathbb{R}^2$ eine zusammengesetzte Quadraturformel, die auf jedem Dreieck eine Gaußsche 5-Punkt-Quadraturformel verwendet. Überprüfen Sie experimentell die Exaktheit und Konvergenzeigenschaften der Formel anhand einer Folge von uniform verfeinerten Triangulierungen und der Funktion $f(x_1, x_2) = \sin(2\pi x_1) \sin(2\pi x_2)$ im Gebiet $\Omega = (-1, 1)^2 \setminus (0, 1)^2$.

30.1 Gewöhnliche Differenzialgleichungen

Aufgabe 30.1.1

(i) Zeigen Sie, dass die aus der formalen Äquivalenz

$$\frac{dy}{dt} = f(t)g(y) \iff \frac{1}{g(y)}dy = f(t)dt \iff \int \frac{1}{g(y)} = \int f(t)$$

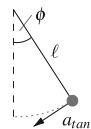
resultierende Funktion $y(t) = G^{-1}(F(t) + c)$ mit Stammfunktionen $G(y)$ von $1/g(y)$ und $F(t) + c$ von $f(t)$ die Differenzialgleichung $y' = f(t)g(y)$ löst und diskutieren Sie hinreichende Bedingungen für die Wohlgestelltheit dieser Darstellung.

- (ii) Wie lassen sich Anfangsbedingungen berücksichtigen und inwiefern gilt Eindeutigkeit der Lösung?
- (iii) Konstruieren Sie eine nichttriviale Lösung des Anfangswertproblems $y' = y^{2/3}$, $y(0) = 0$.

Aufgabe 30.1.2 Begründen Sie, dass das Zweikörperproblem für die Beschreibung der Flughöhe z eines Körpers nahe der Erdoberfläche durch die Gleichung $z'' = -g$ beschrieben wird, wobei $g \approx 9.812 \text{ m/s}^2$ die Erdbeschleunigung ist. Verwenden Sie dazu die Größen $m_{\text{Erde}} \approx 5.974 \cdot 10^{24} \text{ kg}$ und $r_{\text{Erde}} \approx 6.371 \cdot 10^6 \text{ m}$.

Aufgabe 30.1.3 Wir betrachten das in Abb. 30.1 skizzierte Fadenpendel der Länge $\ell > 0$, an dessen Ende ein Gewicht der Masse m angebracht sei. Bestimmen Sie die tangentiale Beschleunigung a_{\tan} , um zu zeigen, dass sich der Auslenkungswinkel $\phi : [0, T] \rightarrow \mathbb{R}$ bei Vernachlässigung von Reibungseffekten durch die Differenzialgleichung $\phi'' = -(g/\ell) \sin(\phi)$ mit der Erdbeschleunigung g beschreiben lässt. Vereinfachen Sie die Differenzialgleichung für den Fall kleiner Winkel und geben Sie die Lösung der resultierenden Gleichung an.

Abb. 30.1 Mathematische Beschreibung eines Fadenpendels



Aufgabe 30.1.4 Skizzieren Sie das Phasendiagramm des Räuber-Beute-Modells

$$y'_1 = \alpha(1 - y_2)y_1, \quad y'_2 = \beta(y_1 - 1)y_2,$$

im Bereich $[0, 5]^2$ für die Parameter $\alpha = 1$ und $\beta = 1$. Begründen Sie damit das Auftreten periodischer Lösungen sowie die Positivität von Lösungen für geeignete Anfangsdaten.

Aufgabe 30.1.5 Skizzieren Sie das Phasendiagramm für die Gleichung des ungedämpften Fadenpendels $\phi'' = -(g/\ell) \sin(\phi)$, indem Sie die Differenzialgleichung zunächst als System erster Ordnung schreiben. Zeichnen Sie verschiedene Lösungskurven in das Diagramm ein und interpretieren Sie diese physikalisch.

Aufgabe 30.1.6 Für eine natürliche Zahl $n \geq 2$ sei y eine Lösung der Differenzialgleichung $y' = f(t)y + g(t)y^n$. Zeigen Sie, dass die Funktion $z = y^{1-n}$ eine Differenzialgleichung erfüllt, die sich mit der Methode der Variation der Konstanten lösen lässt.

Aufgabe 30.1.7 Es sei \hat{y} eine Lösung der Differenzialgleichung $y' = f(t)y + g(t)y^2 + h$. Zeigen Sie, dass mit jeder Lösung z der Differenzialgleichung $z' = -(f(t) + 2\hat{y}(t)g(t))z - g(t)$ und der Formel $z = 1/(y - \hat{y})$ weitere Lösungen der ersten Differenzialgleichungen gewonnen werden können. Inwiefern ist diese Beobachtung nützlich?

Aufgabe 30.1.8 Konstruieren Sie die Lösung des Anfangswertproblems $my'' + ry' + D(y - \ell) = 0$, $y(0) = \ell$, $y'(0) = v_0$, welches die Auslenkung eines Federpendels der Länge ℓ beschreibt. Verwenden Sie den Ansatz $y(t) = cz(t) + \ell$, wobei $z(t) = e^{\lambda t}$ für ein $\lambda \in \mathbb{C}$ sei. Diskutieren Sie qualitative Eigenschaften von Lösungen für verschiedene Verhältnisse von D und r .

Aufgabe 30.1.9 Sei $A \in \mathbb{R}^{n \times n}$ diagonalisierbar, das heißt es existieren eine Diagonalmatrix $D \in \mathbb{R}^{n \times n}$ und eine reguläre Matrix $J \in \mathbb{R}^{n \times n}$, sodass $A = J^{-1}DJ$ gilt. Bestimmen sie die Lösung des Systems von Differenzialgleichungen $y' = Ay$ mit Anfangsbedingung $y(0) = y_0$.

Aufgabe 30.1.10 Bestimmen Sie nichttriviale Lösungen der Differenzialgleichungen $y' = ty$, $y' = \sin(t)y$, und $y' = \cos(t)e^y$.

```

function test_ode
T = 1; y_0 = [1,2];
[t_vec,y_vec] = ode45(@f, [0,T],y_0);
plot(t_vec,y_vec(:,1),'-r'); hold on;
plot(t_vec,y_vec(:,2),'-b'); hold off;

function dy = f(t,y)
A = [-2,0;0,-5]; dy = A*y;

```

Abb. 30.2 Numerisches Lösen eines Anfangswertproblems mit MATLAB-Routinen

Projekt 30.1.1 In MATLAB lassen sich Differenzialgleichungen approximativ mit der Routine `ode45` lösen. Im Fall des Systems $y' = f(t, y)$ im Intervall $[0, T]$ mit Anfangsbedingung $y(0) = y_0$ ist dies für die Abbildung $f(t, y) = Ay$ im in Abb. 30.2 gezeigten MATLAB-Programm realisiert. Die Routine `ode45` liefert dabei eine Liste `t_vec` von Zeitpunkten $0 = t_0 < t_1 < \dots < t_N = T$ und eine Matrix `y_vec` mit zugehörigen Approximationen $\tilde{y}(t_i)$ der exakten Lösungswerte $y(t_i)$ zu den Zeitpunkten t_i , $i = 0, 1, \dots, N$. Modifizieren Sie das Programm `test_ode.m`, um folgende Anfangswertprobleme approximativ zu lösen und die Approximationslösungen grafisch darzustellen:

- (i) das Anfangswertproblem des Räuber-Beute-Modells

$$y'_1 = \alpha y_1(1 - y_2), \quad y'_2 = \beta y_2(y_1 - 1)$$

im Intervall $[0, T]$ mit $T = 10$ sowie $\alpha = 2$, $\beta = 1$ und den Anfangsbedingungen $y_1(0) = 3$ und $y_2(0) = 1$;

- (ii) das Anfangswertproblem des Federpendels

$$my'' + ry' + D(y - \ell) = 0$$

im Intervall $[0, T]$ mit $T = 1$ und $m = 1$, $D = 1$, $\ell = 1$ und verschiedenen Werten $r \in \{0, 1, 5\}$ sowie den Anfangsbedingungen $y(0) = \ell$ und $y'(0) = 1$;

- (iii) das Anfangswertproblem des ungedämpften Fadenpendels

$$y'' = -(g/\ell) \sin(y)$$

mit $g = 1$, $\ell = 1$ und den Anfangsbedingungen $y(0) = 0$ sowie $y'(0) \in \{1, 2, 4, 8\}$;

- (iv) das Anfangswertproblem

$$y'' - Ny' - (N + 1)y = 0$$

im Intervall $[0, 1]$ mit Anfangsbedingungen $y(0) = 1$, $y'(0) = -1$, dessen exakte Lösung durch $y(t) = e^{-t}$ gegeben ist, für $N = 1, 2, 10$ und kleine Störungen der Anfangsbedingung $y(0) = 1$.

```

function test_phase_diagram
a = 1; b = 4; dx = 1/4;
c = -3; d = 3; dy = 1/6;
[x,y] = meshgrid(a:dx:b,c:dy:d);
r = (x.^2+y.^2).^(1/2);
v = sin(r); w = cos(r);
quiver(x,y,v,w,'c'); hold on;
v0 = 1.5; w0 = 2;
streamline(x,y,v,w,v0,w0); hold off;

```

Abb. 30.3 Darstellung eines Phasendiagramms

Projekt 30.1.2 Ein Punktgitter (x_i, y_i) , $i = 1, 2, \dots, N$, auf einer rechteckigen Menge $[a, b] \times [c, d] \subset \mathbb{R}^2$ der Feinheiten $d_x, d_y > 0$ in x- beziehungsweise y-Richtung wird in MATLAB durch $[x, y] = \text{meshgrid}(a:dx:b, c:dy:d)$ erzeugt. Dabei sind x und y Matrizen, die die x- und y-Koordinaten der Gitterpunkte enthalten. Ein diskretes Vektorfeld, das durch Matrizen v und w definiert wird, indem jedem Gitterpunkt (x_i, y_i) der Vektor (v_i, w_i) zugeordnet wird, lässt sich mittels `quiver(x, y, v, w)` visualisieren. Eine Integralkurve des diskreten Vektorfelds beginnend in einem Punkt (v_0, w_0) wird mittels `streamline(x, y, v, w, v0, w0)` dargestellt. Das in Abb. 30.3 gezeigte MATLAB-Programm realisiert dies für ein einfaches Beispiel. Modifizieren Sie das Programm, um die Phasendiagramme der folgenden Differenzialgleichungen und jeweils zwei zugehörige Integralkurven darzustellen:

- (i) das Anfangswertproblem des Räuber-Beute-Modells

$$y'_1 = \alpha(1 - y_2)y_1, \quad y'_2 = \beta(y_1 - 1)y_2$$

mit $\alpha = \beta = 1$;

- (ii) das Anfangswertproblem des Federpendels

$$my'' + ry' + D(y - \ell) = 0$$

mit $m = r = D = \ell = 1$;

- (iii) das Anfangswertproblem des ungedämpften Fadenpendels

$$y'' = -(g/\ell) \sin(y)$$

mit $g = 1, \ell = 1$.

30.2 Existenz, Eindeutigkeit und Stabilität

Aufgabe 30.2.1 Seien $L, T > 0$. Zeigen Sie, dass der Raum $C^0([0, T])$ bezüglich der Norm $\|u\|_L = \sup_{t \in [0, T]} e^{-2Lt} |u(t)|$ vollständig ist.

Aufgabe 30.2.2 Lösen Sie das Anfangswertproblem $y' = y^3$, $y(0) = y_0$, skizzieren Sie die Lösung und diskutieren Sie die Anwendbarkeit des Satzes von Picard–Lindelöf.

Aufgabe 30.2.3 Konstruieren Sie unendlich viele Lösungen des Anfangswertproblems $y' = y^{1/3}$, $y(0) = 0$, skizzieren Sie einige und diskutieren Sie die Anwendbarkeit des Satzes von Picard–Lindelöf.

Aufgabe 30.2.4 Bestimmen und skizzieren Sie die Iterierten y^k , $k = 0, 1, \dots, 4$, der Banachschen Fixpunktiteration

$$y^{k+1}(t) = y_0 + \int_0^t f(s, y(s)) \, ds, \quad y(0) = y_0$$

unter Verwendung der Startfunktion $y^0(t) = y_0$ für die Fälle $f(t, y) = ay$ und $y_0 = 1$ sowie $f(t, y) = 1 + y^2$ und $y_0 = 0$.

Aufgabe 30.2.5 Die Funktion $y : [0, T] \rightarrow \mathbb{R}$ sei eine Lösung des Anfangswertproblems $y' = f(y)$, $y(0) = y_0$. Zeigen Sie, dass y eindeutig ist, sofern $f \in C^1(\mathbb{R})$ gilt.

Aufgabe 30.2.6 Sei $f \in C^m([0, T] \times \mathbb{R})$ und $y \in C^1([0, T])$ eine Lösung der Differenzialgleichung $y' = f(t, y)$. Zeigen Sie, dass $y \in C^{m+1}([0, T])$ gilt.

Aufgabe 30.2.7 Konstruieren Sie eine autonome Differenzialgleichung $y' = f(y)$, für die eine Lösung $y \in C^1([0, T])$ mit der Eigenschaft $y \notin C^2([0, T])$ existiert.

Aufgabe 30.2.8 Wir betrachten die Differenzialgleichung $y'' + t^{-1}y' + 4ty = 0$ mit Anfangsdaten $y'(0) = 0$ und $y(0) = y_0$. Verwenden Sie den Potenzreihenansatz $y(t) = \sum_{n=0}^{\infty} (a_n/n!)t^n$, um die Lösung der Gleichung als Reihe darzustellen. Diskutieren Sie die Konvergenz dieser Reihe.

Aufgabe 30.2.9 Für eine stetige Abbildung $A : [0, T] \rightarrow \mathbb{R}^{n \times n}$ betrachten wir das System von Differenzialgleichungen $y' = A(t)y$.

- (i) Modifizieren Sie den Beweis des Satzes von Picard–Lindelöf, um die Existenz einer eindeutigen Lösung mit der Anfangsbedingung $y(0) = y_0$ für $y_0 \in \mathbb{R}^n$ zu zeigen.
- (ii) Zeigen Sie, dass die Menge L aller Lösungen des Systems $y' = A(t)y$ einen Vektorraum definiert.

- (iii) Betrachten Sie die Abbildung $E_0 : L \rightarrow \mathbb{R}^n$, $y \mapsto y(0)$, und folgern Sie, dass $\dim L = n$ gilt.

Aufgabe 30.2.10 Es sei $g : \mathbb{R}^n \rightarrow \mathbb{R}$ eine stetig differenzierbare, nichtnegative Abbildung und es sei $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ definiert durch $f = \nabla g$. Zeigen Sie, dass jede Lösung $y : [0, T] \rightarrow \mathbb{R}^n$ des Anfangswertproblems $y' = f(y)$, $y(0) = y_0$, die Identität

$$\int_0^t |y'(s)|^2 ds + g(y(t)) = g(y_0)$$

für jedes $t \in [0, T]$ erfüllt.

Projekt 30.2.1 Die Flughöhe eines Körpers im Schwerefeld der Erde unter Berücksichtigung von Reibungskräften wird bei großen Geschwindigkeiten durch die Gleichung

$$my''(t) + \eta \operatorname{sign}(y'(t))|y'(t)|^2 = -mg$$

beschrieben, wobei $\eta \geq 0$ ein Reibungskoeffizient ist, der beispielsweise von der Form des Körpers abhängt. Bestimmen Sie experimentell mit der MATLAB-Routine `ode45` Werte für η zur Beschreibung des freien und gebremsten Falls eines Fallschirmspringers, sodass der freie Fall aus 4 km Höhe bis zur Höhe von 1 km etwa 60 s und der anschließende Fallschirmflug bis zur Landung etwa 180 s dauern. Simulieren Sie mit den so gefundenen Parametern verschiedene Absprunghöhen und Höhen für das Auslösen des Fallschirms. Welche maximalen Durchschnittsgeschwindigkeiten beobachten Sie für den freien Fall und den Fallschirmflug?

Projekt 30.2.2 Verwenden Sie die MATLAB-Routine `ode45`, um das Zweikörperproblem

$$\begin{aligned} m_1 y_1'' &= \gamma \frac{m_1 m_2}{\|y_1 - y_2\|^2} \frac{y_2 - y_1}{\|y_1 - y_2\|}, \\ m_2 y_2'' &= \gamma \frac{m_1 m_2}{\|y_1 - y_2\|^2} \frac{y_1 - y_2}{\|y_1 - y_2\|} \end{aligned}$$

für verschiedene Anfangsdaten und Massenverhältnisse $m_1/m_2 \in \{1, 2, 10\}$ näherungsweise zu lösen und darzustellen. Konstruieren Sie sowohl Anfangsdaten, die zur Existenz einer für alle positiven Zeiten wohldefinierten Lösung führen, als auch Anfangsdaten, für die die Lösung nur in einem endlichen Intervall existiert.

30.3 Einschrittverfahren

Aufgabe 30.3.1 Es seien $y \in C^2(\mathbb{R}_{\geq 0})$ und $\tau > 0$. Für $k \in \mathbb{N}_0$ definiere $t_k = k\tau$ und setze $y^k = y(t_k)$. Zeigen Sie, dass für die Größen

$$d_t^- y^k = \frac{y^k - y^{k-1}}{\tau}, \quad d_t^+ y^k = \frac{y^{k+1} - y^k}{\tau},$$

$k = 1, 2, \dots, K-1$, die Abschätzungen

$$|d_t^\pm y^k - y'(t_k)| \leq \frac{\tau}{2} \sup_{t \in t_k \pm [0, \tau]} |y''(t)|$$

gelten. Welche Abschätzung lässt sich für die Differenz $|\hat{d}_t y^k - y'(t_k)|$ mit der Größe

$$\hat{d}_t y^k = \frac{y^{k+1} - y^{k-1}}{2\tau},$$

$k = 1, 2, \dots, K-1$, beweisen?

Aufgabe 30.3.2 Seien $(y_\ell)_{\ell=0, \dots, K}$ eine nichtnegative Zahlenfolge und $\alpha, \beta \geq 0$, sodass für $\ell = 0, 1, \dots, K$ die Abschätzung

$$y_\ell \leq \alpha + \sum_{k=0}^{\ell-1} \beta y_k$$

gilt. Zeigen Sie, dass $y_\ell \leq \alpha(1 + \beta)^\ell \leq \alpha \exp(K\beta)$ für $\ell = 0, 1, \dots, K$ gilt. Folgern Sie die diskrete Version des Lemmas von Gronwall.

Aufgabe 30.3.3 Verwenden Sie das explizite und implizite Euler-Verfahren für die Differenzialgleichung $y'(t) = 2\alpha t y(t)$ mit Schrittweiten $\tau = 1/2^\ell$, $\ell = 1, 2, 3$, sowie dem Anfangswert $y_0 = 1$ und $\alpha = \pm 3$, um die Approximationslösungen beider Verfahren zum Zeitpunkt $T = 1$ zu bestimmen und vergleichen Sie diese mit der exakten Lösung. Kommentieren Sie Ihre Ergebnisse.

Aufgabe 30.3.4 Für eine Inkrementfunktion Φ und $z_k \in \mathbb{R}$ seien $z : [t_k, t_{k+1}] \rightarrow \mathbb{R}$ die Lösung des Anfangswertproblems $z'(t) = f(t, z(t))$, $z(t_k) = z_k$, und $z_{k+1} = z_k + \tau \Phi(t_k, z_k, z_{k+1}, \tau)$. Damit seien die Konsistenzgrößen C und \tilde{C} definiert durch

$$C(t_k, z_k, \tau) = \frac{z(t_{k+1}) - z_k}{\tau} - \Phi(t_k, z_k, z_{k+1}, \tau),$$

$$\tilde{C}(t_k, z_k, \tau) = \frac{z(t_{k+1}) - z_k}{\tau} - \Phi(t_k, z_k, z(t_{k+1}), \tau).$$

Die Inkrementfunktion Φ sei uniform Lipschitz-stetig im dritten Argument mit Lipschitz-Konstante L . Zeigen Sie, dass für $\tau \leq 1/(2L)$ die Äquivalenz

$$c^{-1} |\tilde{C}(t_k, z_k, \tau)| \leq |C(t_k, z_k, \tau)| \leq c |\tilde{C}(t_k, z_k, \tau)|$$

gilt. Geben Sie dabei die nur von L abhängige Konstante c explizit an.

Aufgabe 30.3.5 Sei f eine Lipschitz-stetige Funktion. Zeigen Sie, dass das implizite Euler-Verfahren

$$y_{k+1} = y_k + \tau f(t_{k+1}, y_{k+1})$$

konsistent von der Ordnung $p = 1$ ist, das heißt dass $|C(t_k, z_k, \tau)| \leq c\tau$ mit einer geeigneten Konstanten $c \geq 0$ gilt.

Aufgabe 30.3.6 Sei $f \in C^2([0, T] \times \mathbb{R})$. Zeigen Sie, dass das Verfahren

$$y_{k+1} = y_k + \tau [f(t_k, y_k) + \frac{\tau}{2} (\partial_t f(t_k, y_k) + \partial_y f(t_k, y_k) f(t_k, y_k))]$$

konsistent von der Ordnung $p = 2$ ist.

Aufgabe 30.3.7 Vereinfachen Sie die allgemeine Fehlerabschätzung für Einschrittverfahren für die Spezialfälle des expliziten und impliziten Euler-Verfahrens im Fall der Differenzialgleichung $y' = -\lambda y$ in $(0, T)$, $y(0) = y_0$ mit $\lambda > 0$.

Aufgabe 30.3.8 Geben Sie eine Formel für die durch das implizite und explizite Euler-Verfahren definierten Approximationslösungen $(y_k)_{k=0,1,\dots}$ des Anfangswertproblems $y' = \lambda y$, $y(0) = y_0$ an. Skizzieren Sie die exakten und die numerischen Lösungen für drei verschiedene Werte von λ und verschiedene Zeitschrittweiten. Diskutieren Sie im Fall $\lambda < 0$ die Beschränktheit der Approximationen und der exakten Lösung.

Aufgabe 30.3.9 Bestimmen Sie Zahlen $a, b, c, d \in \mathbb{R}$, für die das durch die Inkrementfunktion

$$\Phi(t_k, y_k, \tau) = af(t_k, y_k) + bf(t_k + c\tau, y_k + \tau df(t_k, y_k))$$

definierte explizite Einschrittverfahren die Konsistenzordnung $p = 2$ besitzt.

Hinweis: Begründen und verwenden Sie die Approximation $f(t + c\tau, y + d\tau f(t, y)) = f(t, y) + \partial_t f(t, y)c\tau + \partial_y f(t, y)d\tau f(t, y) + \mathcal{O}(\tau^2)$ und differenzieren Sie die Differenzialgleichung.

Aufgabe 30.3.10 Verwenden Sie den Satz über implizite Funktionen, um die Existenz einer eindeutigen Lösung y_{k+1} der Gleichung

$$y_{k+1} = y_k + \tau \Phi(t_k, y_k, y_{k+1}, \tau)$$

unter geeigneten Voraussetzungen an die Funktion Φ und die Schrittweite τ sicherzustellen.

Projekt 30.3.1 Das in Abb. 30.4 gezeigte MATLAB-Programm realisiert das durch die Inkrementfunktion $\Phi(t_k, y_k, \tau) = f(t_k + \tau/2, y_k + \tau f(t_k, y_k)/2)$ definierte explizite Euler–Collatz-Verfahren für die Federpendelgleichung

$$y'' + ry' + D(y - \ell) = 0$$

mit den Anfangsdaten $y(0) = y_0$ und $y'(0) = v_0$.

- (i) Untersuchen Sie experimentell die Abhängigkeit der Approximationslösungen von den Parametern r und D .
- (ii) Verwenden Sie die exakte Lösung $y(t) = (v_0/\omega)e^{-rt/2} \sin(\omega t)$ mit $\omega = (D - r^2/4)^{1/2}$ des Anfangswertproblems für den Spezialfall $r = 1/10$, $D = 1$, $y_0 = \ell = 0$, $v_0 = 1$ und bestimmen Sie den Approximationfehler $|y_K - y(t_K)|$ für die Schrittweiten $\tau = 2^{-s}$, $s = 1, 2, \dots, 7$, zum Zeitpunkt $t_K = 100$.
- (iii) Modifizieren Sie das Programm, um das explizite und implizite Euler-Verfahren sowie das Verfahren von Heun zu realisieren. Vergleichen Sie das qualitative Verhalten der verschiedenen Approximationslösungen für den Zeithorizont $T = 1000$.

```

function federpendel
T = 10; y_0 = 0; v_0 = 1;
s = 5; tau = 2^(-s); K = floor(T/tau);
y = zeros(K+1,2);
y(1,:) = [y_0,v_0];
for k = 1:K
    y(k+1,:) = y(k,:)+tau*Phi((k-1)*tau,y(k,:),tau);
    plot(tau*(0:k),y(1:k+1,1),'r');
    axis([0,T,-5,5]); drawnow;
end
D = 1; r = 1/10; omega = sqrt(D-r^2/4); t = K*tau;
% y_ex = ...
% abs(y_ex-y(K+1))

function val = Phi(t,y,tau)
val = (f(t,y)+f(t+tau,y+tau*f(t,y)))/2;

function vec = f(t,y)
r = 1/10; D = 1; ell = 0;
vec = [y(2),-r*y(2)-D*(y(1)-ell)];

```

Abb. 30.4 Numerische Lösung des Anfangswertproblems für das Federpendel

```

function raeuber_beute
T = 10; tau = 1/100; K = floor(T/tau);
alpha = 2; beta = 1;
y = zeros(K+1,2);
y(1,:) = [3,1];
for k = 1:K
    y(k+1,1) = y(k,1)+tau*alpha*y(k,1)*(1-y(k,2));
    y(k+1,2) = y(k,2)+tau*beta*y(k,2)*(y(k,1)-1);
    plot(tau*(0:k),y(1:k+1,1),'b'); hold on;
    plot(tau*(0:k),y(1:k+1,2),'r'); hold off;
    axis([0,T,0,4]); drawnow;
end

```

Abb. 30.5 Numerische Lösung des Anfangswertproblems für das Räuber-Beute-Modell

Projekt 30.3.2 Das in Abb. 30.5 gezeigte MATLAB-Programm berechnet eine Approximationslösung des Räuber-Beute-Modells.

- (i) Kommentieren Sie jede Zeile des Programms und identifizieren Sie das realisierte numerische Verfahren.
- (ii) Testen Sie verschiedene Schrittweiten und beobachten Sie das qualitative Verhalten der Approximationslösungen. Für welche Schrittweiten ergeben sich sinnvolle Resultate?
- (iii) Modifizieren Sie eine Zeile des Programms, um ein implizites Verfahren zu erhalten. Wie ändert sich das qualitative Verhalten der numerischen Lösungen?

30.4 Runge–Kutta-Verfahren

Aufgabe 30.4.1 Bestimmen Sie die Iterierten $(y_k)_{k=0,1,\dots}$ des expliziten Euler Verfahrens, des Euler–Collatz-Verfahrens und des klassischen Runge–Kutta-Verfahrens bei der Approximation des Anfangswertproblems $y' = \lambda y$, $y(0) = y_0$, indem Sie jeweils einen Ausdruck $g(\tau\lambda)$ konstruieren, sodass $y_{k+1} = g(\tau\lambda)y_k$, $k = 0, 1, 2, \dots$, gilt. Bestimmen Sie die Konvergenzordnung der Approximationsfehler $|y(t_k) - y_k|$, indem Sie die Identität $y(t_{k+1}) = e^{\tau\lambda}y(t_k)$ verwenden und für die drei Verfahren jeweils die Differenz $e^{\tau\lambda} - g(\tau\lambda)$ betrachten.

Aufgabe 30.4.2 Leiten Sie hinreichende Bedingungen für die Konsistenz dritter Ordnung eines Runge–Kutta-Verfahrens für den Fall autonomer Differenzialgleichungen her.

Aufgabe 30.4.3 Zeigen Sie durch Konstruktion polynomieller Lösungen geeigneter Anfangswertprobleme, dass die Bedingungen $\sum_{\ell=1}^m \gamma_\ell = 1$, $\sum_{\ell=1}^m \gamma_\ell \alpha_\ell = 1/2$ und $\sum_{\ell=1}^m \sum_{j=1}^m \gamma_\ell \beta_{\ell j} = 1/2$ notwendig für die Konsistenzordnung $p = 2$ eines Runge–Kutta-Verfahrens sind.

Aufgabe 30.4.4 Bestimmen Sie ein zweistufiges Runge–Kutta-Verfahren der Konsistenzordnung $p = 4$, das auf der Gaußschen Quadraturformel mit den Quadraturpunkten $x_0, x_1 = 1/2 \pm 1/(2\sqrt{3})$ und zugehörigen Gewichten $w_0 = w_1 = 1/2$ basiert.

Aufgabe 30.4.5 Bestimmen Sie das Butcher-Tableau des durch die Inkrementfunktion

$$\begin{aligned}\Phi(t, y, \tau) &= \frac{1}{6}(\eta_1 + 4\eta_2 + \eta_3), \\ \eta_1 &= f(t, y), \quad \eta_2 = f(t + \tau/2, y + \tau\eta_1/2), \\ \eta_3 &= f(t + \tau, y + \tau(-\eta_1 + 2\eta_2))\end{aligned}$$

definierten Runge–Kutta-Verfahrens und zeigen Sie, dass es die Konsistenzordnung $p = 3$ besitzt.

Aufgabe 30.4.6 Welche Quadraturformeln liegen dem klassischen Runge–Kutta-Verfahren, der 3/8–Regel und dem Radau-3-Verfahren zugrunde und welche Exaktheitsgrade besitzen diese?

Aufgabe 30.4.7 Das autonome System $z' = F(z)$, $z(0) = z_0$, sei die durch Einführung der Hilfsvariablen w mit $w' = 1$ und $w(0) = 0$ äquivalente Formulierung der Differenzialgleichung $y' = f(t, y)$, $y(0) = y_0$. Zeigen Sie, dass Runge–Kutta-Verfahren in beiden Fällen identische Approximationen von y liefern.

Aufgabe 30.4.8 Konstruieren Sie auf Basis der Simpson-Regel ein Runge–Kutta-Verfahren der Konsistenzordnung $p = 4$.

Aufgabe 30.4.9 Zeigen Sie für den Fall autonomer Differenzialgleichungen, dass das klassische Runge–Kutta-Verfahren definiert durch $\alpha = [0, 1/2, 1/2, 1]^\top$, $\gamma = [1/6, 1/3, 1/3, 1/6]^\top$ und $\beta \in \mathbb{R}^{4 \times 4}$ mit den nichttrivialen Einträgen $\beta_{21} = 1/2$, $\beta_{32} = 1/2$, $\beta_{43} = 1$ konsistent von der Ordnung $p = 4$ ist.

Aufgabe 30.4.10

- (i) Sei $A \in \mathbb{R}^{m \times m}$, sodass $\|A\| < 1$ bezüglich einer Operatornorm gilt. Zeigen Sie, dass die Matrix $I_m - A$ invertierbar ist mit $(I_m - A)^{-1} = \sum_{n=0}^{\infty} A^n$.
- (ii) Formulieren Sie das Newton-Verfahren zur Lösung der Fixpunktgleichung $\eta = \Psi(\eta)$ für die Bestimmung eines Koeffizientenvektors $\eta \in \mathbb{R}^m$ in einem Runge–Kutta-Verfahren und diskutieren Sie dessen Wohlgestelltheit.

Projekt 30.4.1 Das in Abb. 30.6 gezeigte MATLAB-Programm realisiert ein explizites Runge–Kutta-Verfahren zur Lösung einer skalaren Differenzialgleichung $y' = f(t, y)$, $y(0) = y_0$.

- (i) Dokumentieren Sie jede Zeile des Programms.
- (ii) Überprüfen Sie, dass die exakte Lösung für den Fall $f(t, y) = -2y + 5 \cos(t)$ und $y_0 = 2$ gegeben ist durch $y(t) = 2 \cos(t) + \sin(t)$. Bestimmen Sie für die Schrittweiten $\tau = 2^{-s}$, $s = 0, 1, \dots, 5$, den Approximationsfehler $|y(T) - y_K|$ mit $T = t_K = 10$.

```

function runge_kutta_expl
T = 10; s = 2; tau = 2^(-s); K = floor(T/tau);
y = zeros(K+1,1); y(1) = 2;
for k = 1:K
    y(k+1) = y(k)+tau*Phi((k-1)*tau,y(k),tau);
end
plot(tau*(0:K),y(1:K+1),'b-o'); hold on;

function val = Phi(t,y,tau)
m = 2; alpha = [0,1/2]; beta = [0,0;1/2,0]; gamma = [0,1];
eta = zeros(m,1);
val = 0;
for ell = 1:m
    dy = 0;
    for j = 1:ell-1
        dy = dy+beta(ell,j)*eta(j);
    end
    eta(ell) = f(t+tau*alpha(ell),y+tau*dy);
    val = val+gamma(ell)*eta(ell);
end

function val = f(t,y)
val = -2*y+5*cos(t);

```

Abb. 30.6 Realisierung eines expliziten Runge–Kutta-Verfahrens

- (iii) Modifizieren Sie das Programm, um das explizite Euler-Verfahren, das Euler–Collatz-Verfahren, das klassische Runge–Kutta-Verfahren und die 3/8-Regel zu realisieren.
- (iv) Bestimmen Sie für alle Verfahren die Approximationsfehler $|y(T) - y_K|$ zum Zeitpunkt $T = 10$ mit den Schrittweiten $\tau = 2^{-s}$, $s = 0, 1, \dots, 5$. Stellen Sie diese vergleichend als Polygonzüge in einer Grafik mit logarithmischen Achsenkalierungen dar, was in MATLAB mit dem Kommando `loglog` realisiert werden kann.

Projekt 30.4.2 Schreiben Sie zwei MATLAB-Routinen zur numerischen Approximation gewöhnlicher Differenzialgleichungen mit allgemeinen impliziten Runge–Kutta-Verfahren. Verwenden Sie dazu einerseits eine Fixpunktiteration und andererseits das Newton-Verfahren mit einem sinnvollen Abbruchkriterium. Untersuchen Sie die jeweiligen Iterationszahlen in den Zeitschritten für das Radau-3-Verfahren am Beispiel $y' = (1 + y^2)^{1/2}$, $y(0) = 0$, im Intervall $[0, T]$ mit $T = 4$, dessen exakte Lösung durch $y(t) = \sinh(t)$ gegeben ist.

30.5 Mehrschrittverfahren

Aufgabe 30.5.1 Für eine Schrittweite $\tau > 0$ und Zeitschritte $t_k = k\tau, k \in \mathbb{N}_0$, seien Werte $w_k \in \mathbb{R}$ gegeben.

- (i) Konstruieren Sie das durch die drei Stützstellen $(t_{k+\ell}, w_{k+\ell})_{\ell=0,1,2}$ definierte Interpolationspolynom $q \in \mathcal{P}_2$ und integrieren Sie dieses über das Intervall $[t_{k+2}, t_{k+3}]$, um Koeffizienten $(\beta_\ell)_{\ell=0,1,2}$ zu erhalten, sodass

$$\int_{t_{k+2}}^{t_{k+3}} q(t) dt = \tau \sum_{\ell=0}^2 \beta_\ell w_{k+\ell}.$$

- (ii) Konstruieren Sie das durch die drei Stützstellen $(t_{k+\ell}, w_{k+\ell})_{\ell=0,1,2}$ definierte Interpolationspolynom $q \in \mathcal{P}_2$ und integrieren Sie dieses über das Intervall $[t_{k+1}, t_{k+2}]$, um Koeffizienten $(\beta_\ell)_{\ell=0,1,2}$ zu erhalten, sodass

$$\int_{t_{k+1}}^{t_{k+2}} q(t) dt = \tau \sum_{\ell=0}^2 \beta_\ell w_{k+\ell}.$$

Aufgabe 30.5.2 Bestimmen Sie die maximale Zahl $p \in \mathbb{N}$, sodass die Identitäten

$$\sum_{\ell=0}^m \alpha_\ell = 0, \quad \sum_{\ell=0}^m (\alpha_\ell \ell^q - \beta_\ell q \ell^{q-1}) = 0, \quad q = 1, 2, \dots, p,$$

für das Adams–Bashforth- und das Adams–Moulton-Verfahren mit $m = 3$ beziehungsweise $m = 2$ gelten.

Aufgabe 30.5.3 Zeigen Sie, dass das Adams–Moulton-Verfahren unter der Bedingung $\tau \|\beta\|_1 L < 1$ wohldefiniert ist, wobei L die uniforme Lipschitz-Konstante der zur Differenzialgleichung gehörenden Funktion f sei.

Aufgabe 30.5.4 Bestimmen Sie die Konsistenzordnung des *leap-frog*-Verfahrens einerseits direkt mit einer Fehlerabschätzung für die Approximation der Zeitableitung mittels $y'(t_k) \approx (y(t_{k+1}) - y(t_{k-1}))/2\tau$ und andererseits durch Überprüfen des allgemeinen Konsistenzkritieriums für Mehrschrittverfahren.

Aufgabe 30.5.5 Für eine Schrittweite $\tau > 0$ und Zeitschritte $t_k = k\tau, k \in \mathbb{N}_0$, seien Werte $w_k \in \mathbb{R}$ gegeben. Bestimmen Sie die Ableitung $p'(t_{k+m})$ des Interpolationspolynoms $p \in \mathcal{P}_m$ für die Stützpaare $(t_{k+\ell}, w_{k+\ell})_{\ell=0,\dots,m}$ mit $m = 1, 2, 3$. Diskutieren Sie, wie damit ein Mehrschrittverfahren konstruiert werden kann.

Aufgabe 30.5.6 Konstruieren Sie ein Mehrschrittverfahren, indem Sie das Integral in der Darstellung

$$y(t_{k+2}) = y(t_k) + \int_{t_k}^{t_{k+2}} f(s, y(s)) \, ds$$

mit der Simpson-Regel approximieren und bestimmen Sie die Konsistenzordnung des so erhaltenen Verfahrens.

Aufgabe 30.5.7 Zeigen Sie durch Konstruktion geeigneter Anfangswertprobleme, dass das hinreichende Konsistenzkriterium für lineare Mehrschrittverfahren

$$\sum_{\ell=0}^m \alpha_\ell = 0, \quad \sum_{\ell=0}^m (\alpha_\ell \ell^q - \beta_\ell q \ell^{q-1}) = 0, \quad q = 1, 2, \dots, p,$$

notwendig ist.

Aufgabe 30.5.8 Zeigen Sie, dass es für jedes $m \geq 1$ genau ein lineares m -Mehrschrittverfahren der Konsistenzordnung $2m$ und keins der Konsistenzordnung $2m + 1$ gibt. Verwenden Sie dazu die Normierung $\beta_0 = 1$ und formulieren Sie das allgemeine Konsistenzkriterium als lineares Gleichungssystem $A[\hat{\alpha}, \hat{\beta}]^\top = b$ mit $\hat{\alpha} = [\alpha_1, \dots, \alpha_m]^\top$ sowie $\hat{\beta} = [\beta_1, \dots, \beta_m]^\top$. Benutzen Sie den Hauptsatz der Algebra zur Untersuchung der Matrix A^\top .

Aufgabe 30.5.9 Es seien ein lineares, explizites Mehrschrittverfahren definiert durch $(\hat{\alpha}_\ell, \hat{\beta}_\ell)_{\ell=0, \dots, m}$ und ein lineares, implizites Mehrschrittverfahren definiert durch $(\alpha_\ell, \beta_\ell)_{\ell=0, \dots, m}$. Die Approximation y_{k+m} sei definiert durch $y_{k+m} = y_{k+m}^{(v)}$, wobei $y_{k+m}^{(v)}$ durch die Iterationsvorschrift

$$\tilde{y}_{k+m}^{(i+1)} = - \sum_{\ell=0}^{m-1} \alpha_\ell y_{k+\ell} + \tau \sum_{\ell=0}^{m-1} \beta_\ell f(t_{k+\ell}, y_{k+\ell}) + \tau \beta_m f(t_{k+m}, y_{k+m}^{(i)})$$

mit der Initialisierung $y_{k+m}^{(0)} = \tilde{y}_{k+m}$ für

$$\tilde{y}_{k+m} = - \sum_{\ell=0}^{m-1} \hat{\alpha}_\ell y_{k+\ell} + \tau \sum_{\ell=0}^{m-1} \hat{\beta}_\ell f(t_{k+\ell}, y_{k+\ell})$$

berechnet werde. Zeigen Sie, dass dadurch ein explizites Mehrschrittverfahren der Konsistenzordnung $p = \min\{p_{\text{expl}} + v, p_{\text{impl}}\}$ definiert wird, wobei p_{expl} und p_{impl} die Konsistenzordnungen des expliziten beziehungsweise des impliziten Verfahrens bezeichnen.

Aufgabe 30.5.10 Untersuchen Sie, für welche Werte $z = \tau\lambda \in \mathbb{C}$ man mit den durch

α_2	α_1	α_0	β_2	β_1	β_0
1	-1	0	0	3/2	-1/2
1	-1	0	5/12	8/12	-1/12
1	-4/3	1/3	2/3	0	0

definierten Zweischrittverfahren beschränkte Approximationen des Anfangswertproblems $y' = \lambda y$ in $(0, \infty)$, $y(0) = 1$ erhält. Schreiben Sie dazu das Verfahren in der Form $Y_{k+1} = BY_k$ und untersuchen Sie die Matrix $B \in \mathbb{R}^{2 \times 2}$.

Projekt 30.5.1 Wir betrachten das Anfangswertproblem $y' = f(t, y)$ für $t \in (0, T]$, $y(0) = y_0$, mit $f(t, y) = (1 + y^2)^{1/2}$, $y_0 = 0$ und $T = 1$. Die exakte Lösung ist gegeben durch $y(t) = \sinh(t)$.

- (i) Implementieren Sie das Adams–Bashforth–Verfahren.
- (ii) Verwenden Sie eine Fixpunktiteration mit einem geeigneten Abbruchkriterium, um das Adams–Moulton–Verfahren zu realisieren.
- (iii) Programmieren Sie das Adams–Bashforth–Moulton–Verfahren.
- (iv) Vergleichen Sie die Fehler $|y(T) - y_K|$ zum finalen Zeitpunkt $t_K = T$ der drei Verfahren für $m = 2, 3, 4$ und Schrittweiten $\tau = 2^{-\ell}$, $\ell = 2, 3, \dots, 6$, in drei Tabellen. Als Startwerte können Sie die Funktionswerte der exakten Lösung verwenden.

Projekt 30.5.2 Schreiben Sie ein kurzes Programm zur algorithmischen Bestimmung der Konsistenzordnung eines gegebenen Mehrschrittverfahrens. Testen Sie es für die Adams–Verfahren mit $m = 1, 2, 3, 4$ Schritten sowie für das durch $m = 6$ und

$$[\alpha_6, \alpha_5, \dots, \alpha_0] = \frac{1}{147}[147, -360, 450, -400, 225, -72, 10],$$

$$[\beta_6, \beta_5, \dots, \beta_0] = \frac{1}{147}[60, 0, 0, 0, 0, 0, 0]$$

definierte Mehrschrittverfahren.

30.6 Konvergenz von Mehrschrittverfahren

Aufgabe 30.6.1 Es sei $A \in \mathbb{R}^{m \times m}$ die diagonalisierbare Begleitmatrix der durch $(\alpha_\ell)_{\ell=0,\dots,m}$ definierten Differenzengleichung mit linear unabhängigen Eigenvektoren v_1, v_2, \dots, v_m . Zeigen Sie, dass die Folge $(y_k)_{k \geq 0}$ genau dann eine Lösung der homogenen Differenzengleichung ist, wenn für die Vektoren $Y_k = [y_k, y_{k+1}, \dots, y_{k+m-1}]^\top$ gilt $Y_k = \sum_{j=1}^m \lambda_j^k \gamma_j v_j$, $k \geq 0$ mit geeigneten Zahlen $\gamma_j \in \mathbb{R}$ und den Nullstellen λ_j des Polynoms $q(\lambda) = \lambda^m + \alpha_{m-1}\lambda^{m-1} + \dots + \lambda\alpha_1 + \alpha_0$.

Aufgabe 30.6.2 Untersuchen Sie die Nullstabilität und Konsistenz des Mehrschrittverfahrens $y_{k+2} - 4y_{k+1} + 3y_k = -2\tau f(t_k, y_k)$.

Aufgabe 30.6.3 Für $(\alpha_\ell)_{\ell=0,\dots,m}$ mit $\alpha_m = 1$ betrachten wir die lineare homogene Differenzengleichung

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = 0.$$

- (i) Zeigen Sie, dass zu m Startwerten $y_0, y_1, \dots, y_{m-1} \in \mathbb{R}$ genau eine Folge $(y_k)_{k \geq 0}$ existiert, welche die homogene Differenzengleichung löst.
- (ii) Zeigen Sie, dass die homogene Differenzengleichung m linear unabhängige Lösungen $(y_k)_{k \geq 0}$ besitzt.

Aufgabe 30.6.4 Es sei $\lambda \in \mathbb{C}$ eine s -fache Nullstelle des Polynoms $q(z) = z^m + \alpha_{m-1}z^{m-1} + \dots + \alpha_1z + \alpha_0$ und $(y_k)_{k \geq 0}$ definiert durch $y_k = k^r \lambda^k$ mit $r \in \mathbb{N}$, $r < s$. Ferner sei für $f \in C^1(\mathbb{R})$ und $x \in \mathbb{R}$ die Funktion $Af \in C^0(\mathbb{R})$ definiert durch $Af(x) = xf'(x)$.

- (i) Beweisen Sie die Identität

$$\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = \lambda^k \sum_{v=0}^r \binom{r}{v} k^v \sum_{\ell=0}^m \alpha_\ell \ell^{r-v} \lambda^\ell = \lambda^k \sum_{v=0}^r \binom{r}{v} k^v A^{r-v} q(\lambda).$$

- (ii) Sei x_0 eine $(r+1)$ -fache Nullstelle von $f \in C^r(\mathbb{R})$, das heißt es gelte $f(x_0) = f'(x_0) = \dots = f^{(r)}(x_0) = 0$. Zeigen Sie, dass $A^i f(x_0) = 0$ für $i = 0, 1, \dots, r$ gilt.
- (iii) Folgern Sie, dass $(y_k)_{k \geq 0}$ eine Lösung der linearen homogenen Differenzengleichung $\sum_{\ell=0}^m \alpha_\ell y_{k+\ell} = 0$ ist und diskutieren Sie die Beschränktheit dieser Folge.

Aufgabe 30.6.5 Sei $R \in \mathbb{C}^{m \times m}$ regulär und $\|\cdot\|$ eine Norm auf \mathbb{C}^m . Zeigen Sie, dass durch $A \mapsto \sup_{x \in \mathbb{R}^m \setminus \{0\}} \|RAx\|/\|x\|$ eine Operatornorm auf $\mathbb{R}^{m \times m}$ definiert wird.

Aufgabe 30.6.6 Untersuchen Sie die Nullstabilität der Fibonacci-Folge $y_{k+2} = y_{k+1} + y_k$ und der Tschebyscheff-Rekursion $T_{k+2}(x) = 2xT_{k+1}(x) - T_k(x)$.

Aufgabe 30.6.7 Die Jordansche Normalform der Begleitmatrix $A \in \mathbb{R}^{m \times m}$ einer Differenzengleichung sei reell und die Dahlquistsche Wurzelbedingung verletzt. Zeigen Sie, dass dann $\rho(A) > 1$ gilt.

Aufgabe 30.6.8 Spezifizieren Sie die Konstanten C_0, C_1, C_2 in der allgemeinen Konvergenzaussage für Mehrschrittverfahren und diskutieren Sie, in welchen Situationen die Fehlerabschätzung von praktischem Nutzen ist.

Aufgabe 30.6.9 Sei $f \in C^1([0, T] \times \mathbb{R})$ mit $|\partial_z f(t, z)| \leq C$ für alle $(t, z) \in [0, T] \times \mathbb{R}$. Zeigen Sie, dass die Adams–Moulton-, Adams–Bashforth-, und Adams–Bashforth–Moulton–Verfahren die Bedingungen der allgemeinen Konvergenzaussage für Mehrschrittverfahren erfüllen.

Aufgabe 30.6.10 Es sei $J = T^{-1}AT$ die Jordansche Normalform der Matrix $A \in \mathbb{R}^{m \times m}$ mit Jordan-Blöcken J_i , $i = 1, 2, \dots, r$. Für $\varepsilon \geq 0$ sei $D \in \mathbb{R}^{m \times m}$ die Diagonalmatrix mit Einträgen $d_{kk} = \varepsilon^{k-1}$ für $k = 1, 2, \dots, m$. Zeigen Sie, dass die Matrix $\tilde{J} = D^{-1}JD$ durch die Blöcke

$$\tilde{J}_i = \begin{bmatrix} \lambda_i & \varepsilon & & \\ & \ddots & \ddots & \\ & & \ddots & \varepsilon \\ & & & \lambda_i \end{bmatrix},$$

$i = 1, 2, \dots, r$, gegeben ist.

Projekt 30.6.1 Formulieren Sie Algorithmen zur systematischen experimentellen Analyse der Nullstabilität einer Differenzengleichung einerseits durch Testen zufällig gewählter Startwerte und andererseits durch Lösen eines geeigneten Eigenwertproblems. Diskutieren Sie die Zuverlässigkeit der so ermittelten Beurteilung und testen Sie Ihre Algorithmen mit den Koeffizienten

$$[\alpha_2, \alpha_1, \alpha_0] = [1, 4, -5],$$

$$[\alpha_2, \alpha_1, \alpha_0] = [1, -4, 3],$$

$$[\alpha_2, \alpha_1, \alpha_0] = [1, 0, -1],$$

$$[\alpha_4, \alpha_3, \alpha_2, \alpha_1, \alpha_0] = [1, -48/25, 36/25, -16/25, 3/25].$$

Projekt 30.6.2 Die BDF-Verfahren (*Backward Differentiation Formulas*) sind für $m \geq 1$ gegeben durch

$$\sum_{\ell=0}^m \hat{\alpha}_\ell y_{k+\ell} = \tau f(t_{k+m}, y_{k+m})$$

mit den Koeffizienten $\hat{\alpha}_m = \sum_{j=1}^m 1/j$ und

$$\hat{\alpha}_\ell = (-1)^{m-\ell} \sum_{j=m-\ell}^m \frac{1}{j} \binom{j}{m-\ell},$$

$\ell = 0, 1, \dots, m-1$. Verwenden Sie die BDF-Verfahren mit $m = 1, 2, \dots, 7$ zur numerischen Approximation des Anfangswertproblems $y' = f(t, y)$ in $(0, T]$, $y(0) = y_0$, mit $f(t, y) = -2y + 5 \cos(t)$, $y_0 = 1$ und $T = 1$, dessen exakte Lösung gegeben ist durch $y(t) = 2 \cos(t) + \sin(t)$. Bestimmen Sie die experimentellen Konvergenzraten zum Zeitpunkt $T = 1$ mit geeigneten Folgen von Zeitschrittweiten und dem Ansatz $e_\tau \approx c\tau^\gamma$, sodass für zwei verschiedene Schrittweiten folgt

$$\gamma \approx \log(e_\tau/e_{\tau'}) / \log(\tau/\tau').$$

30.7 Steife Differenzialgleichungen

Aufgabe 30.7.1 Sei $A \in \mathbb{R}^{2 \times 2}$ mit Eigenwerten $\lambda_1, \lambda_2 \in \mathbb{C}$. Zeichnen Sie die Phasendiagramme der Differenzialgleichung $z' = Az$ in einer Umgebung des Ursprungs für vier typische Situationen charakterisiert durch

- (i) $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$, (ii) $\lambda_1, \lambda_2 \in \mathbb{R}_{<0}$, (iii) $\lambda_1, \lambda_2 \in \mathbb{R}$, $\lambda_1 \lambda_2 < 0$, (iv) $\lambda_1 = \bar{\lambda}_2$.

Aufgabe 30.7.2 Ein numerisches Verfahren führe für jede Schrittweite $\tau > 0$ zu beschränkten Approximationen der skalaren Differenzialgleichung $y' = \lambda y$, sofern $\operatorname{Re}(\lambda) \leq 0$ gilt. Weiter sei $A \in \mathbb{R}^{n \times n}$ komplex diagonalisierbar und die Eigenwerte von A haben ausschließlich negative Realteile. Zeigen Sie, dass das Verfahren A -stabil ist.

Aufgabe 30.7.3

- (i) Zeigen Sie, dass die Anwendung eines linearen Mehrschrittverfahrens auf die Differenzialgleichung $y' = \lambda y$ auf eine homogene Differenzengleichung führt.
- (ii) Definieren Sie den Begriff der A -Stabilität für lineare Mehrschrittverfahren, sodass er im Fall des impliziten Euler-Verfahrens konsistent mit der Definition für Einschrittverfahren ist.
- (iii) Untersuchen Sie die A -Stabilität der durch $m = 2$ und

$$[\alpha_2, \alpha_1, \alpha_0] = [1, -4/3, 1/3], \quad [\beta_2, \beta_1, \beta_0] = [2/3, 0, 0]$$

beziehungsweise $m = 3$ und

$$[\alpha_3, \alpha_2, \alpha_1, \alpha_0] = [1, -18/11, 9/11, -2/11], \quad [\beta_3, \beta_2, \beta_1, \beta_0] = [6/11, 0, 0, 0]$$

definierten Verfahren.

Aufgabe 30.7.4 Sei $A \in \mathbb{R}^{n \times n}$ negativ definit, das heißt es existiere ein $\alpha > 0$, sodass $z^\top A z \leq -\alpha \|z\|^2$ für alle $z \in \mathbb{R}^n$. Zeigen Sie, dass die Lösung des Anfangswertproblems $y' = Ay$ für jeden Anfangswert $y_0 \in \mathbb{R}^n$ exponentiell schnell gegen 0 konvergiert.

Aufgabe 30.7.5 Seien $\alpha \in \mathbb{R}^m$, $\beta \in \mathbb{R}^{m \times m}$ und $\gamma \in \mathbb{R}^m$ die Koeffizienten eines Runge-Kutta-Verfahrens. Zeigen Sie, dass die zugehörige Stabilitätsfunktion eine polynomiale oder rationale Funktion ist.

Aufgabe 30.7.6 Untersuchen Sie das durch das Butcher-Tableau

	1/3	5/12	-1/12
	1	3/4	1/4
		3/4	1/4

definierte Runge-Kutta-Verfahren auf A - und L -Stabilität.

Aufgabe 30.7.7 Die Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ sei einseitig Lipschitz-stetig, das heißt für alle $z, w \in \mathbb{R}^n$ gelte

$$\langle f(z) - f(w), z - w \rangle \leq L \|z - w\|^2.$$

Zeigen Sie, dass die Differenzialgleichung $y' = f(y)$ für jeden Anfangswert y_0 höchstens eine Lösung besitzt und diskutieren Sie die Wohlgestelltheit der Differenzialgleichungen $y' = -y^3$ sowie $y' = y^3$.

Aufgabe 30.7.8 Sei $G \in C^1(\mathbb{R}^n)$. Zeigen Sie, dass G genau dann konvex ist, wenn

$$\nabla G(z) \cdot (w - z) + G(z) \leq G(w)$$

für alle $z, w \in \mathbb{R}^n$ gilt.

Aufgabe 30.7.9 Der Satz von Peano besagt, dass jedes Anfangswertproblem $y' = f(y)$, $y(0) = y_0$, mit einer stetigen Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ eine Lösung in einem Intervall $(0, \varepsilon)$ besitzt und $\varepsilon > 0$ beliebig gewählt werden kann, sofern die Lösung beschränkt bleibt. Zeigen Sie, dass das Anfangswertproblem $y' = -\nabla G(y)$ mit einer koerziven Funktion $G \in C^1(\mathbb{R}^n)$ für jeden Anfangswert $y_0 \in \mathbb{R}^n$ eine auf ganz $\mathbb{R}_{\geq 0}$ definierte Lösung besitzt und diskutieren Sie die Anwendbarkeit auf die Differenzialgleichung $y' = -y^3$.

Aufgabe 30.7.10

- (i) Sei $G : \mathbb{R} \rightarrow \mathbb{R}$ definiert durch $G(z) = (1 - z^2)^2$. Skizzieren Sie die Funktion G und zeigen Sie, dass G μ -konvex ist.
- (ii) Es gelte $G(x) \geq -c_1 + c_2|x|^p$ mit $p \geq 1$. Zeigen Sie, dass G koerziv ist und, sofern G zudem stetig ist, ein Minimum besitzt.

Projekt 30.7.1 Wir betrachten das Anfangswertproblem $y' = -\alpha(y - \cos(t))$, $y(0) = 0$ im Intervall $[0, T]$ mit $T = 1$ und $\alpha = 50$.

- (i) Überprüfen Sie, dass die Lösung des Problems gegeben ist durch

$$y(t) = \frac{\alpha}{1 + \alpha^2} (\sin(t) + \alpha \cos(t) - \alpha e^{-\alpha t}).$$

- (ii) Lösen Sie das Problem approximativ mit dem expliziten und impliziten Euler-Verfahren, dem Trapez-Verfahren sowie dem klassischen Runge–Kutta-Verfahren mit den Schrittweiten $\tau = 2^{-\ell}/10$, $\ell = 0, 1, 2, 3$. Stellen Sie die Fehler zum Zeitpunkt T vergleichend in einer Tabelle dar.
- (iii) Stellen Sie die Approximationen für einige Schrittweiten und die exakte Lösung vergleichend in einer Grafik dar.

Projekt 30.7.2 Wir betrachten das Anfangswertproblem $y' = -\alpha y^3$, $y(0) = 1$, im Intervall $[0, T]$ mit $T = 1$ und $\alpha = 200$.

- (i) Zeigen Sie, dass das Anfangswertproblem einen Gradientenfluss für eine geeignete Funktion G definiert und bestimmen Sie die exakte Lösung.
- (ii) Testen Sie das explizite und implizite Euler-Verfahren zur approximativen Lösung des Problems. Verwenden Sie dabei das Newton-Verfahren, um nichtlineare Gleichungen approximativ zu lösen. Dokumentieren Sie Ihre Beobachtungen.
- (iii) Testen Sie das semiimplizite Euler-Verfahren

$$y_{k+1} = y_k - \tau \alpha y_k^2 y_{k+1}$$

sowie das linearisierte implizite Euler-Verfahren

$$y_{k+1} = y_k - \tau \alpha (f(y_k) + f'(y_k)(y_{k+1} - y_k)),$$

wobei $f(y) = y^3$ sei, und dokumentieren Sie Ihre Beobachtungen.

- (iv) Bestimmen Sie experimentell für jedes der obigen Verfahren Schrittweiten, für die die Folge $(G(y_k))_{k=0,\dots,K}$ monoton fallend ist.

30.8 Schrittweitensteuerung

Aufgabe 30.8.1 Es sei $\hat{y}_\tau : [0, T] \rightarrow \mathbb{R}$ der affin-lineare Interpolant der mit dem impliziten Euler-Verfahren berechneten Approximationen des Anfangswertproblems $y' = f(y)$, $y(0) = y_0$, und es gelte $y \in C^1([0, T])$. Zeigen Sie, dass $\hat{y}_\tau \rightarrow y$ für $\tau_{\max} = \max_{k=1,\dots,K} \tau_k \rightarrow 0$ gleichmäßig in $[0, T]$ konvergiert.

Aufgabe 30.8.2 Seien $\hat{y}_\tau, \bar{y}_\tau : [0, T] \rightarrow \mathbb{R}$ die Interpolanten einer Folge $(y_k)_{k=0,\dots,K}$ mit maximaler Schrittweite $\tau = \max_{k=1,\dots,K} \tau_k$. Zeigen Sie, dass für $k = 1, 2, \dots, K$ gilt

$$\sup_{t \in [t_{k-1}, t_k]} |\hat{y}_\tau(t) - \bar{y}_\tau(t)| \leq \tau \sup_{t \in [t_{k-1}, t_k]} |\hat{y}'_\tau(t)|.$$

Aufgabe 30.8.3 Sei $\bar{y}_\tau, \hat{y}_\tau : [0, T] \rightarrow \mathbb{R}$ der stückweise konstante beziehungsweise stückweise affine Interpolant der Folge $(y_k)_{k=0,\dots,K}$ zur uniformen Schrittweite $\tau > 0$. Zeigen Sie, dass für jede Funktion $v \in C^1([0, T])$ die Identitäten

$$\int_0^T v' \bar{y}_\tau dt = - \sum_{k=0}^{K-1} (y_{k+1} - y_k) v(t_k) + y_K v(T) - y_0 v(0)$$

und

$$\int_0^T v' \hat{y}_\tau dt = - \int_0^T v \hat{y}'_\tau dt + y_K v(T) - y_0 v(0)$$

gelten.

Aufgabe 30.8.4 Sei $y \in C^0([0, T])$. Bestimmen Sie Bedingungen unter denen für einen gegebenen Parameter $\delta > 0$ eine Zahl $\tau > 0$ existiert, sodass

$$|y(t + \tau) - y(t)| \leq \delta$$

für alle $t \in [0, T - \tau]$ gilt. Zeigen Sie mit einem Beispiel, dass dies ohne Zusatzvoraussetzungen an y im Allgemeinen nicht gilt.

Aufgabe 30.8.5 Leiten Sie eine a-posteriori Fehlerabschätzung für das explizite Euler-Verfahren her.

Aufgabe 30.8.6 Zeigen Sie für den Fall einer autonomen Differenzialgleichung $y' = f(y)$ mit einer Lipschitz-stetigen Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$, dass das auf der a-posteriori Fehlerabschätzung basierende adaptive Verfahren stets terminiert, das heißt die finale Zeit erreicht wird.

Aufgabe 30.8.7 Es gelte $\hat{y}'_\tau = f(\hat{y}_\tau) + R_\tau$ und $y' = f(y)$ im Intervall $(0, T)$ sowie $\hat{y}_\tau(0) = y(0)$. Zeigen Sie, dass für den Fehler $e(t) = y(t) - \hat{y}_\tau(t)$ gilt

$$\sup_{t \in [0, T]} |e(t)| \leq \max_{t \in [0, T]} |R_\tau(t)| \exp(LT),$$

und vergleichen Sie diese Abschätzung mit anderen a-posteriori Fehlerabschätzungen.

Aufgabe 30.8.8 Es sei ein numerisches Verfahren der Konsistenzordnung p gegeben. Konstruieren Sie durch Extrapolation von Approximationen zu den Schrittweiten τ , $\tau/2$ und $\tau/4$ ein Verfahren der Konsistenzordnung $3p$. Diskutieren Sie den Gesamtaufwand des so erhaltenen Verfahrens im Vergleich zur Verwendung des Ausgangsverfahrens mit der Schrittweite τ^3 .

Aufgabe 30.8.9 Bestimmen Sie das Butcher-Tableau des durch Extrapolation des impliziten Euler-Verfahrens mit Schrittweiten τ und $\tau/2$ erhaltenen Verfahrens und diskutieren Sie dessen Konsistenzordnung.

Aufgabe 30.8.10 Seien $f \in C^1([0, T])$ und $g \in C^0([0, T])$ mit $f, g \geq 0$ sowie $c_0 \geq 0$, sodass $f'(t) \leq c_0 + (g(t)f(t))^{1/2}$ für alle $t \in [0, T]$ gilt.

- (i) Zeigen Sie, dass für alle $a, b \in \mathbb{R}$ und $\gamma > 0$ gilt $ab \leq \gamma a^2/2 + b^2/(2\gamma)$.
- (ii) Zeigen Sie mit dem Gronwall-Lemma, dass für jedes $\delta > 0$ gilt

$$\max_{t \in [0, T]} f(t) \leq (f(0) + c_0 T + (\delta T/2) \max_{t \in [0, T]} g(t)) \exp(T/(2\delta)).$$

- (iii) Beweisen Sie ohne Verwendung des Gronwall-Lemmas, dass

$$\max_{t \in [0, T]} f(t) \leq 2f(0) + 2c_0 T + T^2 \max_{t \in [0, T]} g(t).$$

- (iv) Diskutieren Sie vergleichend Vor- und Nachteile der Abschätzungen aus (ii) und (iii).

Projekt 30.8.1 Implementieren Sie den adaptiven Algorithmus zur Schrittweitensteuerung und testen Sie ihn mit dem impliziten Euler-Verfahren für die Anfangswertprobleme

$$y'(t) = -(y(t) - 100 \cos(t)), \quad t \in (0, 1], \quad y(0) = 0,$$

und

$$y''(t) = 20(1 - y(t)^2)y'(t) - y(t), \quad t \in [0, 100], \quad y(0) = 1/10, \quad y'(0) = 0.$$

Verwenden Sie unterschiedliche Parameter $\delta > 0$ für die Bedingung $|y_{k+1} - y_k| \leq \delta$ und stellen Sie die variablen Schrittweiten als Funktion der Zeit dar. Vergleichen Sie den Aufwand und die Genauigkeit des adaptiven Verfahrens zur Berechnung der Approximationen auf einem uniformen Gitter. Verwenden Sie dabei, dass die Lösung des ersten Anfangswertproblems gegeben ist durch $y(t) = 50(\sin(t) + \cos(t) - e^{-t})$.

Projekt 30.8.2 Implementieren Sie die Extrapolation des Trapezverfahrens mit Schrittweiten τ und $\tau/2$ und überprüfen Sie die verbesserte Konsistenzordnung am Beispiel des Anfangswertproblems $y'(t) = -y(t) + \cos(t)$, $y(0) = 0$ im Intervall $[0, T]$ mit $T = 1$, dessen exakte Lösung durch $y(t) = (\sin(t) + \cos(t) - e^{-t})/2$ gegeben ist. Stellen Sie die Approximationen $(y_k^\tau)_{k=0,\dots,K}$, $(y_k^{\tau/2})_{k=0,\dots,2K}$ und $(\tilde{y}_k)_{k=0,\dots,K}$ vergleichend grafisch dar.

30.9 Symplektische, Schieß- und dG-Verfahren

Aufgabe 30.9.1 Formulieren Sie die kinetische Energie $mv^2/2$ und die potenzielle Energie mgh in geeigneten Polarkoordinaten, um eine Hamilton-Funktion für das Fadenpendel herzuleiten.

Aufgabe 30.9.2 Zeigen Sie, dass das Newtonsche Trägheitsgesetz als Hamiltonsches System interpretiert werden kann. Nehmen Sie dazu an, dass die wirkende Kraft als negativer Gradient eines Potenzials gegeben ist.

Aufgabe 30.9.3 Mit einer Funktion $V \in C^1(\mathbb{R})$ sei ein Hamiltonsches System gegeben durch die Funktion $H : \mathbb{R}^{N \times 3} \times \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}$,

$$H(q, p) = \sum_{i=1}^N \frac{\|p_i\|^2}{2m_i} + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N V(\|q_i - q_j\|).$$

Zeigen Sie, dass der Gesamtimpuls P und der Gesamtdrehimpuls L , die mit dem dreidimensionalen Kreuzprodukt definiert sind durch

$$P = \sum_{i=1}^N p_i, \quad L = \sum_{i=1}^N q_i \times p_i,$$

des Systems erhalten bleiben.

Aufgabe 30.9.4 Sei $J \in \mathbb{R}^{2n \times 2n}$ definiert durch

$$J = \begin{bmatrix} & I_n \\ -I_n & \end{bmatrix}.$$

- (i) Zeigen Sie, dass $\omega : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \rightarrow \mathbb{R}$, $\omega(z_1, z_2) = z_1^\top J z_2$ eine schiefsymmetrische Bilinearform definiert.
- (ii) Sei P ein Parallelogramm in \mathbb{R}^2 , das durch die Vektoren z_1 und z_2 aufgespannt wird. Zeigen Sie, dass der Flächeninhalt von P gegeben ist durch $|\omega(z_1, z_2)|$. Wie ist das Vorzeichen von $\omega(z_1, z_2)$ zu interpretieren?
- (iii) Konstruieren Sie eine nichtlineare Abbildung $\Psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, die symplektisch ist.

Aufgabe 30.9.5 Bestimmen Sie sämtliche symplektische Matrizen $A \in \mathbb{R}^{2 \times 2}$.

Aufgabe 30.9.6 Zur Beschreibung der Bahn eines Planeten der Masse m im Gravitationsfeld einer ruhenden Sonne der Masse $M \gg m$ verwenden wir die Hamilton-Funktion

$$H(q, p) = \frac{\|p\|^2}{2m} - \gamma \frac{mM}{\|q\|}.$$

- (i) Die Bewegung des Körpers finde in einer Ebene statt und werde durch die Funktion $q : [0, T] \rightarrow \mathbb{R}^2$ beschrieben. Ferner sei $p = mq'$. Verwenden Sie Polarkoordinaten (r, ϕ) um zu zeigen, dass

$$H(q, p) = \frac{m}{2}((r')^2 + (r\phi')^2) - \frac{\gamma m M}{r}.$$

- (ii) Verwenden Sie die Konstanz des Drehimpulses $\hat{L} = q \times p$, dessen Länge durch $L = mr^2\phi'$ gegeben ist, und der Gesamtenergie $H(q(t), p(t)) = H_0$, um zu zeigen, dass für den Radius als Funktion des Winkels gilt

$$\left(\frac{dr}{d\phi}\right)^2 = \frac{2mr^4}{L^2} \left(H_0 + \frac{\gamma M m}{r} - \frac{L^2}{2mr^2}\right).$$

- (iii) Beweisen Sie, dass sich jede Ellipse $\{(x, y) \in \mathbb{R}^2 : (x/a)^2 + (y/b)^2 = c^2\}$ in Polarkoordinaten bezüglich eines Brennpunkts darstellen lässt durch $r(\phi) = s/(1 + \varepsilon \cos(\phi))$, $\phi \in [0, 2\pi]$, und zeigen Sie, dass

$$\left(\frac{dr}{d\phi}\right)^2 = \frac{r^4}{s^2} \left(\varepsilon^2 - 1 + \frac{2s}{r} - \frac{s^2}{r^2}\right).$$

- (iv) Folgern Sie, dass die Laufbahn des Planeten durch eine Ellipse beschrieben wird.

Aufgabe 30.9.7 Zeigen Sie, dass das Mittelpunktverfahren symplektisch ist, aber im Fall der Hamilton-Funktion

$$H(q, p) = \sum_{i=1}^N \frac{\|p_i\|^2}{2m_i} + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N V(\|q_i - q_j\|)$$

die Lösung nichtlinearer Gleichungssysteme erfordert.

Aufgabe 30.9.8 Zeigen Sie, dass das implizite Euler-Verfahren nicht symplektisch ist.

Aufgabe 30.9.9

- (i) Zeigen Sie, dass das Verfahren

$$\begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} q_k \\ p_k \end{bmatrix} + \tau \begin{bmatrix} \partial_p H(q_{k+1}, p_k) \\ -\partial_q H(q_{k+1}, p_k) \end{bmatrix}$$

symplektisch ist.

- (ii) Welche Nachteile ergeben sich im Vergleich zum partionierten Euler-Verfahren, bei dem auf der rechten Seite die Ausdrücke $\partial_p H(q_k, p_{k+1})$ und $-\partial_q H(q_k, p_{k+1})$ verwendet werden.

Aufgabe 30.9.10 Zeigen Sie, dass das diskontinuierliche Galerkin-Verfahren für $\ell = 1$ auf eine Variante des Mittelpunktverfahrens führt.

Projekt 30.9.1 Eine Kugel der Masse $m = 10 \text{ g}$ soll vertikal so in die Höhe geschossen werden, dass sie nach genau 10 Sekunden wieder den Erdboden erreicht. Unter Berücksichtigung des Luftwiderstands ist damit eine Anfangsgeschwindigkeit s gesucht, sodass für die Lösung des Anfangswertproblems

$$my'' + \eta \operatorname{sign}(y')|y'|^2 = -mg, \quad t \in (0, 10], \quad y(0) = 0, \quad y'(0) = v_0$$

gilt $y(10) = 0$. Dabei sei $g = 9.81 \text{ m/s}^2$ und $\eta = 2 \cdot 10^{-4} \text{ kg/m}$. Verwenden Sie Bisektions-Verfahren und das Newton-Verfahren, um die Aufgabe approximativ zu lösen. Zur Definition eines geeigneten Startwerts, können Sie das Problem zunächst unter Vernachlässigung von Reibungseffekten lösen. Testen Sie andere Reibungskoeffizienten und diskutieren Sie das Konvergenzverhalten der Verfahren. Überprüfen Sie die Plausibilität Ihrer Ergebnisse.

Projekt 30.9.2 Verwenden Sie das explizite und implizite Euler-Verfahren, das Mittelpunktverfahren sowie das partitionierte Euler-Verfahren, um das Fadenpendel mittels der Hamilton-Funktion

$$H(\phi, \psi) = \frac{1}{2}\psi^2 - \cos(\phi)$$

im Zeitintervall $[0, T]$ mit $T = 10$ zu simulieren. Stellen Sie die Trajektorien $t \mapsto (\phi(t), \psi(t))$ im Phasendiagramm dar und plotten Sie die Gesamtenergie $t \mapsto H(\phi(t), \psi(t))$ sowie die kinetische und potenzielle Energie vergleichend für die Verfahren und verschiedene Schrittweiten. Lösen Sie nichtlineare Gleichungssysteme mit dem Newton-Verfahren.

Teil V
Anhänge

31.1 Skalarprodukt von Vektoren

Auf dem Vektorraum \mathbb{R}^n wird durch die Abbildung

$$\cdot : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}, \quad (v, w) \mapsto v \cdot w = v^\top w = \sum_{i=1}^n v_i w_i$$

eine bilinare Abbildung definiert, die als *Skalarprodukt* bezeichnet wird. Die Euklidische Länge eines Vektors ist damit gegeben durch

$$\|v\|_2 = (v \cdot v)^{1/2} = \left(\sum_{i=1}^n v_i^2 \right)^{1/2}.$$

Zwei Vektoren $v, w \in \mathbb{R}^n$ spannen eine Ebene auf und mit dem Winkel α zwischen diesen Vektoren innerhalb der Ebene gilt

$$v \cdot w = \cos(\alpha) \|v\|_2 \|w\|_2.$$

Zwei Vektoren $v, w \in \mathbb{R}^n$ heißen *orthogonal*, bezeichnet durch $v \perp w$, falls $v \cdot w = 0$ gilt.

31.2 Determinante quadratischer Matrizen

Im Fall $n = 2$ wird ein orientierter Flächeninhalt des von zwei Vektoren $v, w \in \mathbb{R}^2$ aufgespannten Parallelogramms definiert durch

$$\det[v, w] = v_1 w_2 - v_2 w_1.$$

Allgemeiner ist das orientierte Volumen eines von Vektoren $v_1, v_2, \dots, v_n \in \mathbb{R}^n$ aufgespannten Parallelepipeds durch die *Determinante* $\det V$ der Matrix V , deren Spalten die Vektoren v_1, v_2, \dots, v_n sind, gegeben. Das Vorzeichen der Determinante definiert eine Äquivalenzrelation auf der Menge der Basen des \mathbb{R}^n und erlaubt so die Definition einer positiven und negativen Orientierung. Der Wert einer Determinante lässt sich rekursiv mit dem Laplaceschen Entwicklungssatz berechnen, der besagt, dass

$$\det V = \sum_{j=1}^n (-1)^{i+j} \det \hat{V}_{ij}$$

gilt, wobei $\hat{V}_{ij} \in \mathbb{R}^{(n-1) \times (n-1)}$ aus V durch Streichen der i -ten Zeile und j -ten Spalte entsteht und für jede reelle Zahl $s \in \mathbb{R}$ die Identität $\det s = s$ gilt. Für Dreiecksmatrizen $R \in \mathbb{R}^{n \times n}$, das heißt es gilt $r_{ij} = 0$ für alle $i > j$ oder für alle $i < j$, ist $\det R = r_{11}r_{22} \dots r_{nn}$.

31.3 Bild und Kern linearer Abbildungen

Für eine Matrix $A \in \mathbb{R}^{m \times n}$ beziehungsweise die damit identifizierte lineare Abbildung sind ihr *Bild* und *Kern* definiert durch

$$\begin{aligned}\text{Im } A &= \{w \in \mathbb{R}^m : \exists v \in \mathbb{R}^n, w = Av\}, \\ \ker A &= \{v \in \mathbb{R}^n : Av = 0\}.\end{aligned}$$

Damit gelten die Identitäten

$$\mathbb{R}^m = \text{Im } A + \ker A^\top, \quad \mathbb{R}^n = \text{Im } A^\top + \ker A,$$

wobei die Zerlegungen sogar orthogonal sind, das heißt für $w = Av \in \text{Im } A$ und $u \in \ker A^\top$ gilt

$$w \cdot u = w^\top u = (Av)^\top u = (v^\top A^\top)u = v^\top (A^\top u) = 0.$$

Damit ist $\text{Im } A$ das orthogonale Komplement von $\ker A^\top$, das heißt $\text{Im } A = (\ker A^\top)^\perp$. Der Rang einer Matrix A ist die Dimension des Bildes der induzierten linearen Abbildung, das heißt

$$\text{rank } A = \dim \text{Im } A.$$

Der *Rang* einer Matrix entspricht der Anzahl linear unabhängiger Spaltenvektoren. Durch Elementarumformungen erhält man, dass der Rang einer Matrix mit dem Rang der transponierten Matrix übereinstimmt. Aus der Orthogonalität der obigen Zerlegungen ergeben

sich damit die Formeln

$$m = \operatorname{rank} A + \dim \ker A^\top, \quad n = \operatorname{rank} A + \dim \ker A,$$

insbesondere gilt $\operatorname{rank} A = \operatorname{rank} A^\top$. Für einen Endomorphismus beziehungsweise eine quadratische Matrix $A \in \mathbb{R}^{n \times n}$ folgt, dass er genau dann bijektiv ist, wenn er surjektiv, das heißt $\operatorname{Im} A = \mathbb{R}^n$, oder injektiv, das heißt $\ker A = \{0\}$, ist. In diesem Fall ist A regulär beziehungsweise invertierbar und es gilt $\det A \neq 0$.

31.4 Eigenwerte und Diagonalisierbarkeit

Charakteristische Informationen über eine Matrix A und die zugehörige lineare Abbildung sind in den *Eigenwerten* enthalten, die die Nullstellen des *charakteristischen Polynoms* n -ten Grads

$$p_A(t) = \det(A - tI_n)$$

sind. Eine Zahl $\lambda \in \mathbb{R}$ ist ein Eigenwert von A genau dann, wenn ein zugehöriger Eigenvektor $v \in \mathbb{R}^n \setminus \{0\}$ mit $Av = \lambda v$ existiert. Die Menge der Eigenwerte wird auch als *Spektrum* bezeichnet. Jede Dreiecksmatrix $R \in \mathbb{R}^{n \times n}$ besitzt n Eigenwerte, die durch die Diagonaleinträge von R gegeben sind. Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt *diagonalisierbar*, wenn eine invertierbare Matrix $V \in \mathbb{R}^{n \times n}$ und eine Diagonalmatrix $D \in \mathbb{R}^{n \times n}$ existieren, sodass $V^{-1}AV = D$ gilt. In diesem Fall haben A und D dieselben Eigenwerte und sind durch die Diagonaleinträge von D gegeben. Ferner sind dann die Spaltenvektoren von V zugehörige Eigenvektoren, denn es gilt

$$[Av_1, \dots, Av_n] = A[v_1, \dots, v_n] = AV = VD = [v_1, \dots, v_n]D = [\lambda_1 v_1, \dots, \lambda_n v_n].$$

Damit folgt, dass A genau dann diagonalisierbar ist, wenn es eine Basis bestehend aus Eigenvektoren von A gibt. Ein Beispiel für eine nicht diagonalisierbare Matrix ist

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

denn das charakteristische Polynom von A besitzt nur die Nullstelle $\lambda = 0$ und wäre A diagonalisierbar, so gäbe es eine invertierbare Matrix $V \in \mathbb{R}^{2 \times 2}$ mit $V^{-1}AV = 0$, was $A = 0$ zur Folge hätte. Symmetrische Matrizen sind stets diagonalisierbar und es existiert eine Orthonormalbasis bestehend aus Eigenvektoren, das heißt es existieren linear unabhängige Eigenvektoren v_1, \dots, v_n mit $\|v_j\|_2 = 1$ sowie $v_j \cdot v_k = 0$ für $1 \leq j, k \leq n$ mit $j \neq k$.

31.5 Jordansche Normalform

Das charakteristische Polynom einer Matrix $A \in \mathbb{R}^{n \times n}$ besitzt stets n komplexe Nullstellen, allerdings ist auch die durch A definierte Abbildung $A : \mathbb{C}^n \rightarrow \mathbb{C}^n$, $z \mapsto Az$ im Allgemeinen nicht diagonalisierbar. Jede Matrix $A \in \mathbb{R}^{n \times n}$ ist jedoch komplex trigonalisierbar, das heißt es existieren eine invertierbare Matrix $T \in \mathbb{C}^{n \times n}$ sowie eine obere Dreiecksmatrix $J \in \mathbb{C}^{n \times n}$, deren Diagonaleinträge die komplexen Eigenwerte von A sind, sodass $A = T^{-1}JT$ gilt. Die Existenz der *Jordanschen Normalform* besagt, dass J so gewählt werden kann, dass

$$J = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{bmatrix}$$

mit Blockmatrizen $J_i \in \mathbb{R}^{s_i \times s_i}$, $i = 1, 2, \dots, r$, den sogenannten Jordan-Kästchen, die mit Eigenwerten λ_{ℓ_i} , $i = 1, 2, \dots, r$, durch

$$J_i = \begin{bmatrix} \lambda_{\ell_i} & 1 & & \\ & \lambda_{\ell_i} & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_{\ell_i} \end{bmatrix}$$

gegeben sind, gilt. Dabei tritt jeder Eigenwert λ entsprechend seiner geometrischen Vielfachheit, das heißt der Dimension von $\ker(A - \lambda I_n)$, in mehreren Jordan-Kästchen auf. Die Summe der Größen der Jordan-Kästchen eines Eigenwerts λ entspricht seiner algebraischen Vielfachheit, das heißt die Vielfachheit der Nullstelle λ des charakteristischen Polynoms $p_A(t)$.

32.1 Stetige und differenzierbare Funktionen

Der *Zwischenwertsatz* garantiert für jede stetige Funktion $f \in C^0([a, b])$ mit der Eigenschaft $f(a)f(b) \leq 0$ die Existenz eines $\xi \in [a, b]$, sodass $f(\xi) = 0$ gilt. Der *Satz von Bolzano–Weierstraß* besagt, dass jede Funktion $f \in C^0([a, b])$ ihr Maximum und Minimum annimmt, das heißt es existieren $\xi_{\max}, \xi_{\min} \in [a, b]$ mit $f(\xi_{\max}) \geq f(x) \geq f(\xi_{\min})$ für alle $x \in [a, b]$. Eine Funktion $f : [a, b] \rightarrow \mathbb{R}$ heißt differenzierbar in $x_0 \in [a, b]$, falls eine Zahl $L \in \mathbb{R}$, eine Zahl $\delta > 0$ und eine Funktion $\varphi : [0, \delta) \rightarrow \mathbb{R}$ existieren, sodass

$$f(x) = f(x_0) + L(x - x_0) + \varphi(|x - x_0|)$$

für alle $x \in [a, b]$ mit $|x - x_0| < \delta$ sowie $\lim_{s \rightarrow 0} \varphi(s)/|s| \rightarrow 0$ gilt. In diesem Fall heißt L Ableitung von f bei x_0 und wir schreiben $f'(x) = L$. Ist f in jedem Punkt $x_0 \in [a, b]$ differenzierbar, und ist die so definierte Funktion $x_0 \mapsto f'(x_0)$ stetig, so heißt f stetig differenzierbar und wir schreiben $f \in C^1([a, b])$. Induktiv lassen sich so k -mal stetig differenzierbare Funktionen $f \in C^k([a, b])$ definieren. Für jedes $k \in \mathbb{N}_0$ ist die Menge $C^k([a, b])$ ein Vektorraum, auf dem durch

$$\|f\|_{C^k([a,b])} = \max_{i=0,\dots,k} \sup_{x \in [a,b]} |f^{(i)}(x)|$$

die sogenannte Supremumsnorm definiert wird. Ist eine Funktion f beziehungsweise eine ihrer Ableitungen bis zur Ordnung k nur stetig im offenen Intervall (a, b) , so schreiben wir $f \in C^k(a, b)$. In diesem Fall kann f oder eine Ableitung von f unbeschränkt und die Norm $\|f\|_{C^k([a,b])}$ nicht definiert sein.

32.2 Mittelwertsatz und Taylor-Polynome

Ist $f \in C^1([a, b])$, so besagt der Mittelwertsatz beziehungsweise im Spezialfall $f(a) = f(b)$ der *Satz von Rolle*, dass ein $\xi \in (a, b)$ existiert mit

$$\frac{f(a) - f(b)}{a - b} = f'(\xi).$$

Nach dem *Fundamentalsatz der Differenzial- und Integralrechnung* gilt die Identität

$$\int_a^b f'(x) dx = f(b) - f(a),$$

und mit dem *Mittelwertsatz* folgt

$$f(b) - f(a) = \int_a^b f'(x) dx = f'(\xi)(b - a)$$

für ein $\xi \in (a, b)$. Allgemeiner zeigt man, dass für eine Funktion $f \in C^{n+1}([a, b])$ und $x_0 \in [a, b]$ die *Taylor-Formel*

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n + R_{n+1}(x_0) \\ &= \sum_{j=0}^n \frac{1}{j!} f^{(j)}(x_0)(x - x_0)^j + R_{k+1}(x_0), \end{aligned}$$

mit einem *Restglied* $R_{k+1}(x_0)$, das die *Lagrange-Darstellung*

$$R_{k+1}(x_0) = \frac{1}{k!} \int_{x_0}^x (x - t)^k f^{(k+1)}(t) dt = \frac{1}{(k+1)!} f^{(k+1)}(\xi)(x - x_0)^{k+1}$$

mit einer Zahl $\xi \in [x_0, x]$ erfüllt, gilt. Die Taylor-Formel definiert somit ein approximierendes Polynom $T_{k,x_0}f$ vom Grad k mit der Eigenschaft

$$\|f - T_{k,x_0}f\|_{C^0([a,b])} \leq \frac{\|f^{(k+1)}\|_{C^0([a,b])}}{(k+1)!} (b-a)^{k+1}.$$

32.3 Landau-Symbole

Die Approximationseigenschaft des Taylor-Polynoms lässt sich für eine Funktion $f \in C^{k+1}([a, b])$ kürzer schreiben als

$$f(x) - T_{k,x_0}f(x) = \mathcal{O}(|x - x_0|^{k+1}), \quad x \rightarrow x_0.$$

Dabei steht das sogenannte *Landau-Symbol* $\mathcal{O}(|x - x_0|^{k+1})$ für einen Ausdruck $\varphi(|x - x_0|^{k+1})$ mit einer Funktion $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, für die Zahlen $\delta > 0$ und $c \geq 0$ existieren, sodass $\varphi(s)/|s|^{k+1} \leq c$ für alle $s \in \mathbb{R}$ mit $|s| \leq \delta$ gilt. Für alle $x_0 \in [a, b]$ und $x \in [a, b]$ mit $|x - x_0| < \delta$ gilt also

$$|f(x) - T_{k,x_0} f(x)| \leq c|x - x_0|^{k+1}.$$

Etwas allgemeiner gilt für $f \in C^k([a, b])$ die Eigenschaft

$$f(x) - T_{k,x_0} f(x) = o(|x - x_0|^k), \quad x \rightarrow x_0,$$

wobei das Landau-Symbol $o(|x - x_0|^k)$ einen Ausdruck $\varphi(|x - x_0|^k)$ mit einer Funktion $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ repräsentiert, die die Eigenschaft $\lim_{s \rightarrow 0} \varphi(s)/|s|^k \rightarrow 0$ besitzt. Basierend auf der Taylor-Formel lässt sich der Weierstraßsche Approximationssatz beweisen, welcher besagt, dass jede Funktion $f \in C^0([a, b])$ gleichmäßig durch Polynome approximiert werden kann. Im Unterschied zur beispielsweise bei Aufwandsbetrachtungen von Algorithmen verwendeten Notation $\mathcal{O}(n^p)$ wird hier der Grenzwert $s \rightarrow 0$ betrachtet. Die wichtigsten Fälle der Landau-Symbole lassen sich folgendermaßen zusammenfassen:

$$\begin{aligned} g(n) = \mathcal{O}(n^p), n \rightarrow \infty &\iff \exists c \geq 0 \forall n \in \mathbb{N}, |g(n)| \leq cn^p, \\ \psi(s) = \mathcal{O}(s^p), s \rightarrow 0 &\iff \exists c \geq 0 \limsup_{s \rightarrow 0} |\psi(s)|/|s|^p \leq c, \\ \psi(s) = o(s^p), s \rightarrow 0 &\iff \lim_{s \rightarrow 0} |\psi(s)|/|s|^p = 0. \end{aligned}$$

In der Regel ist aus dem Kontext ersichtlich, welcher Grenzwert gemeint ist, sodass auf den Zusatz $n \rightarrow \infty$, $s \rightarrow 0$ oder $x \rightarrow x_0$ häufig verzichtet wird.

32.4 Fundamentalsatz der Algebra

Ein Punkt $x_0 \in [a, b]$ wird als ℓ -fache Nullstelle einer Funktion $f \in C^r([a, b])$ bezeichnet, falls $r \geq \ell$ und $f^{(j)}(x_0) = 0$ für $j = 0, 1, \dots, \ell$ gelten. Im Fall eines Polynoms p vom Grad $k \geq \ell$ folgt, dass

$$p(x) = (x - x_0)^\ell r(x)$$

mit einem Polynom r vom Grad $k - \ell$ gilt. Identifiziert man ein Polynom p vom Grad k mit einer Abbildung $f_p : \mathbb{C} \rightarrow \mathbb{C}$, indem man p in kanonischer Weise auf die komplexe Zahlebene fortsetzt, so gilt nach dem *Fundamentalsatz der Algebra*, dass die Funktion f stets k Nullstellen $z_1, z_2, \dots, z_k \in \mathbb{C}$ besitzt, die im Allgemeinen nicht paarweise verschieden sind. Sind Polynome p und q gegeben, so existieren Polynome s und r , sodass

$$p(x) = s(x)q(x) + r(x)$$

für alle $x \in \mathbb{R}$ gilt. Mit der Bedingung, dass der Grad des Rests r echt kleiner als der von s ist, sind s und r eindeutig bestimmt, sofern p oder q nicht identisch Null ist.

32.5 Mehrdimensionale Differentialrechnung

Eine auf einer offenen Menge $U \subset \mathbb{R}^n$ definierte stetige Abbildung $f : U \rightarrow \mathbb{R}^m$ heißt (total) differenzierbar im Punkt $x_0 \in U$, falls eine lineare Abbildung $L : \mathbb{R}^n \rightarrow \mathbb{R}^m$ existiert, sodass

$$f(x) - f(x_0) = L(x - x_0) + o(\|x - x_0\|_2)$$

gilt. In diesem Fall wird das Differential $Df(x_0)$ von f am Punkt x_0 als die lineare Abbildung L definiert und mit der darstellenden, sogenannten *Jacobi-* oder Funktionalmatrix identifiziert. Diese Matrix wird ebenfalls mit $Df(x_0) \in \mathbb{R}^{m \times n}$ bezeichnet und ihre Einträge sind für $i = 1, 2, \dots, m$ und $j = 1, 2, \dots, n$ durch die partiellen Ableitungen

$$\partial_j f_i(x_0) = \frac{\partial f_i}{\partial x_j}(x_0) = \lim_{x \rightarrow x_0} \frac{f_i(x_0 + he_j) - f(x_0)}{h}$$

gegeben. Ist f in jedem Punkt $x_0 \in U$ differenzierbar, so schreiben wir $f \in C^1(U; \mathbb{R}^m)$. Im Fall $m = 1$, das heißt $f : U \rightarrow \mathbb{R}$, wird der *Gradient* von f definiert durch $\nabla f(x) = (Df(x))^\top \in \mathbb{R}^n$. Mit dieser Definition gilt

$$Df(x)[s] = \nabla f(x) \cdot s$$

für alle $s \in \mathbb{R}^n$. Sind alle partiellen Ableitungen von ∇f stetig differenzierbar, so entspricht die symmetrische *Hesse-Matrix* $D^2 f$ der Funktionalmatrix von ∇f . Die mehrdimensionale Taylor-Formel impliziert, dass für $f \in C^2(U)$ und $x, x_0 \in U$ ein $\xi \in U$ existiert, sodass

$$f(x) = f(x_0) + \nabla f(x) \cdot (x - x_0) + \frac{1}{2} D^2 f(\xi)[x - x_0, x - x_0].$$

Dabei steht $D^2 f(\xi)[d, d]$ für den Ausdruck $d^\top D^2 f(\xi)d$. Eine notwendige Bedingung für ein Extremum einer Funktion $f \in C^1(U)$ an der Stelle $x_0 \in U$ ist, dass $\nabla f(x_0) = 0$ gilt. Ist zusätzlich $f \in C^2(U)$ erfüllt und $D^2 f(x_0)$ positiv definit, so folgt, dass x_0 ein lokales isoliertes Minimum ist. Im Fall einer konvexen Funktion ist x_0 sogar ein globales, eindeutiges Minimum.

33.1 Struktur

Die Programmiersprache C ist eine Compiler-basierte Sprache, das heißt mit einem Texteditor wie *emacs* oder *kate* erstellte Programme werden mit Hilfe des Compilers in einen maschinenlesbaren Code übersetzt. Damit dies fehlerfrei funktioniert, müssen Programme nach einem vorgegebenen Rahmen geschrieben werden. Ein C-Programm beginnt mit der Einbindung benötigter vordefinierter Routinen, die in Bibliotheken bereitgestellt werden, wie mathematischer Funktionen oder Ein- und Ausgabefunktionen. Es folgen optional selbstdefinierte Funktionen und am Ende steht das mit `main()` beginnende Hauptprogramm. Im Hauptprogramm stehen Variablendefinitionen und Kommandos wie beispielsweise der Aufruf einer Funktion. Das linksseitig stehende Programm in Abb. 33.1 zeigt ein einfaches Beispiel, in dem in einer Unterroutine das Quadrat einer Zahl berechnet wird. Diese wird vom Hauptprogramm aus mit einem Argument aufgerufen. Ist das Programm als Textdatei unter dem Namen `comp_square.c` abgespeichert, so kann es im selben Verzeichnis mit dem Unix-Kommando

```
gcc -lm comp_square.c -o comp_square.out  
kompiliert werden. Mit dem Unix-Kommando  
./comp_square.out  
wird das Programm gestartet.
```

33.2 Bibliotheken

Die Bibliothek `stdio.h` stellt Routinen zur Ein- und Ausgabe zur Verfügung, während in der Bibliothek `math.h` Implementationen elementarer mathematischer Funktionen realisiert sind. Die arithmetischen Grundoperationen `+`, `-`, `*`, `/` können ohne die Einbindung von Bibliotheken verwendet werden. Zur Darstellung von Gleitkommazahlen und ganzzahligen Maschinenzahlen werden die Kommandos `printf("text %f\n", x)`

```
// comp_square.c
#include <stdio.h>
#include <math.h>
double square(double x) {
    return pow(x, 2.0);
}
main() {
    double x, y;
    x = 3.8;
    y = square(x);
    printf("square is %f\n", y);
}

// simple_loop.c
#include <stdio.h>
main() {
    int i;
    for (i=0; i<5; i=i+1) {
        printf("%d\n", i);
    }
    printf("\n");
    if (i==5){
        printf("i is 5 \n");
    }
}
```

Abb. 33.1 Elementare C-Programme**Tab. 33.1** Ein- und Ausgabe, Kommentare sowie elementare mathematische Funktionen

printf, scanf	Aus- und Eingabe von Text und Variablen
/*...*/, //	Kommentare
cos, sin, tan	Trigonometrische Funktionen
exp, log, log10	Exponentialfunktion und Logarithmen
pow, sqrt	Potenz und Quadratwurzel
floor, ceil, fabs	Runden auf ganze Zahlen und Betragsfunktion

und `printf("text %d\n", i)` verwendet, wobei `\n` einen Zeilenumbruch bewirkt. Das Einlesen von Werten für die Variablen erfolgt mit `scanf ("%lf", &x)` beziehungsweise `scanf ("%d", &i)`. Weitere Befehle sind in Tab. 33.1 aufgeführt.

33.3 Typen

Jede Variable muss in C deklariert werden, das heißt ihr Typ wie *integer* oder *double* muss vor ihrer Verwendung in einem Unterprogramm oder im Hauptprogramm festgelegt werden, s. Tab. 33.2. Auch die Verwendung von Zahlen und arithmetischen Operationen ist mit Typen verbunden, beispielsweise wird 2 als Variable vom Typ *integer* interpretiert, während 2. als Variable vom Typ *double* verwendet wird. Die Operation 2/3 wird in C als binäre Operation des höherwertigen Variablentyps ausgeführt, das heißt

$$2/3 = 0, \quad 2./3. \approx 0.\overline{6}, \quad 2/3. \approx 0.\overline{6}, \quad 2./3 \approx 0.\overline{6}.$$

Variablen können beispielsweise mit `x = (double)a` umgewandelt werden. Arrays fester Größe können über `double x[n]` oder `double A[m][n]` initialisiert werden.

Tab. 33.2 Variablentypen in C

int	Ganzzahlige Maschinenzahlen
float, double	Gleitkommazahlen einfacher und doppelter Genauigkeit

```
if (condA) statementA else if (condB) statementB else statementC
while (cond) statement
for (init; cond; step) statement
```

Abb. 33.2 Kontrollstrukturen in C

Die Indizierung der Einträge von Arrays beginnt mit 0. Eine falsche Indizierung von Arrays führt im Allgemeinen nicht zu einer Fehlermeldung und muss vom Programmierer ausgeschlossen werden. Bei der Deklaration einer Variablen kann bereits ein Wert zugewiesen werden, sofern es sich nicht um ein Array handelt, dessen Größe durch eine Variable definiert ist.

33.4 Kontrollanweisungen

In C können Fallunterscheidungen und Schleifen in den in Abb. 33.2 dargestellten Formaten realisiert werden. Dabei steht `cond` für eine logische Bedingung, die über eine logische Operation wie `a < b` definiert sein kann, während `statement` für eine Liste von Kommandos steht, die von geschwungenen Klammern eingefasst sind. Die Ausdrücke `init` und `step` stehen für eine Initialisierung wie `i=0` und eine Anweisung der Art `i=i+1`, deren Ausführung so lange wiederholt wird, wie die Bedingung `cond` beispielsweise `i < 5` als wahr ausgewertet wird. Dabei wird zunächst `init` ausgeführt, dann `cond` überprüft, anschließend der Kommandoblock `statement` abgearbeitet und schließlich `step` ausgeführt, bevor wiederum die Bedingung `cond` ausgewertet wird und dieser Vorgang wiederholt wird, bis `cond` falsch ist. Gelegentlich ist auch die Verwendung der `do while`-Schleife sinnvoll, bei der die Bedingung im Unterschied zur `while`-Schleife nach statt vor der Ausführung der Kommandos überprüft wird.

33.5 Logische Ausdrücke und Inkrementa

Zur Formulierung logischer Bedingungen stehen binäre Operationen zum Vergleich von Maschinenzahlen, die logischen Verknüpfungen *und*, *oder* sowie die Negation zur Verfügung, s. Tab. 33.3. Vergleiche stehen dabei in Klammern also beispielsweise `(a < b)`. Gleitkommazahlen werden nur bis auf Maschinengenauigkeit verglichen und aufgrund möglicher Störungen ist ein Test auf exakte Gleichheit zweier Gleitkommazahlen wenig

Tab. 33.3 Logische Operationen sowie Inkrement- und Dekrementfunktionen

<code>==, !=, >, >=, <, <=</code>	Logischer Vergleich von Maschinenzahlen
<code>&&, , !</code>	Logische Verknüpfungen <i>und</i> , <i>oder</i> sowie Negation
<code>i++, ++i, i-, -i</code>	Prä- und Postinkrement sowie -decrement
<code>b+=x</code>	Kurzform für <code>b=b+x</code>

```
// static_array.c           // functions.c
#include <stdio.h>          #include <stdio.h>
void mod_vector(double* vec) {    void fun_1(double z) {
    vec[0] = 2.0;            z = z+1.0;
    vec[1] = 1.0;            }
}                                void fun_2(double* z) {
main() {                          *z = *z+1.0;
    int n = 2;                }
    double x[n];
    x[0] = 1.0;
    x[1] = 2.0;
    mod_vector(x);
    printf("x[0] = %f\n", x[0]);
    printf("x[1] = %f\n", x[1]);
}
}                                main() {
                                double x = 1.0;
                                fun_1(x);
                                printf("x = %f\n", x);
                                fun_2(&x);
                                printf("x = %f\n", x);
}
```

Abb. 33.3 Übergabe von Arrays sowie einfachen Variablen und Pointern an Funktionen

sinnvoll. Das Kommando `i=i+1` kann in C durch `i++` oder `++i` ersetzt werden und in arithmetischen oder logischen Ausdrücken verwendet werden. Im Fall `i++` wird die Variable zunächst erhöht und anschließend der Ausdruck ausgewertet, während bei Verwendung von `++i` zunächst die Variable erhöht wird. Die logischen Ausdrücke (`i==i+1`) or (`++i==1`) führen also zu unterschiedlichen Ergebnissen.

33.6 Funktionen

Funktionen können entweder eine oder keine Variable zurückgeben. Wird ein Wert zurückgegeben, so steht der Typ des Funktionswerts vor dem Funktionsnamen, andernfalls wird `void` verwendet. Auf den Funktionsnamen folgt in Klammern eine Liste von Argumenten. Argumente einfachen Typs wie `double` und `int` werden mittels *call by value* behandelt, das heißt sie werden in eine lokale Variable kopiert. Die entsprechende Variable im Hauptprogramm bleibt dabei unverändert. Arrays sind nicht als Rückgabewert einer Funktion zugelassen. Gegebenenfalls werden daher Arrays an Funktionen als Argumente mittels *call by reference* übergeben und dabei als globale Variablen von dem Unterprogramm verändert. In dem in Abb. 33.3 links gezeigten Programm wird das Array `x` an die Funktion `mod_vector` übergeben und dort unter dem Namen `vec` verwendet. Nach dem Durchlauf der Funktion sind die Werte des Arrays `x` verändert. Wichtig ist hierbei die Verwendung des Sterns bei der Deklarierung des Arguments der Funktion.

33.7 Pointer

Ein Pointer ist eine Variable, die die Adresse eines Abschnitts im Speicher enthält. Der Pointer erlaubt es, den Inhalt des entsprechenden Speicherabschnitts auszulesen oder zu verändern. Die Größe des Speicherabschnitts hängt davon ab, ob dort eine Gleitkommazahl oder ganzzählige Maschinenzahl abgelegt werden soll. Ist `ptr` ein Pointer, so ist der

Tab. 33.4 Kommandos zur Initialisierung und De-Initialisierung dynamischer Arrays

<code>malloc</code>	Reservieren von Speicherplatz
<code>realloc</code>	Neues Reservieren von Speicherplatz
<code>free</code>	Freigeben von Speicherplatz
<code>NULL</code>	Wert für Pointer ohne reservierten Speicherplatz

Wert der Variable, die unter der in `ptr` enthaltenen Adresse zu finden ist, gegeben durch `*ptr`. Umgekehrt wird für eine gewöhnliche Variable `var` durch `&var` ein Pointer definiert, der die Adresse des entsprechenden Speicherplatzes enthält. Initialisiert wird ein Pointer beispielsweise mittels `double* ptr`. Übergibt man die Adresse einer Variable, das heißt den entsprechenden Pointer, an eine Funktion, dann wird diese Variable mittels *call by reference* behandelt, sodass der Inhalt der Variablen mit globalen Auswirkungen manipuliert werden kann. Die oben beschriebene Übergabe von Arrays an Funktionen folgt gerade diesem Prinzip. In dem in Abb. 33.3 rechts gezeigten Programm verändert die Funktion `fun_1` den Wert der Variablen `x` des Hauptprogramms nicht, während die Funktion `fun_2` deren Wert erhöht. Ein in einer Funktion definierter Pointer kann als Rückgabewert der Funktion verwendet werden. In diesem Fall muss die Funktion mittels `double*` deklariert werden.

33.8 Dynamische Arrays

Die oben beschriebene Verwendung von Arrays stößt an Grenzen, wenn die Dimension der Arrays erst im Laufe des Programmdurchlaufs bestimmt wird. In diesem Fall können dynamische Arrays verwendet werden, für deren Deklaration und Manipulation von der Bibliothek `stdlib.h` Routinen bereitgestellt werden. Ein Array kann als Kette von Pointern angesehen werden und es muss Speicherplatz für die entsprechenden Variablen reserviert werden. Ein Array kann damit einem Pointer `ptr` auf eine entsprechende Variable zugewiesen werden und mit `ptr[j]` erhält man die entsprechenden Variablen. Reservierter Speicher sollte stets wieder freigegeben werden. Tabelle 33.4 bietet eine Übersicht der wichtigsten Kommandos. In Abb. 33.4 ist beispielhaft eine Implementation der Ein- und Ausgabe von Vektoren variabler Länge gezeigt.

33.9 Arbeiten mit Matrizen

Eine Matrix $A \in \mathbb{R}^{m \times n}$ kann mit einem Vektor $\hat{A} \in \mathbb{R}^{mn}$ identifiziert werden, indem man die Spalten von A untereinander in einen Vektor schreibt. Es gilt, bei Numerierung der Einträge mit den Indizes $i = 0, 1, \dots, m - 1$ und $j = 0, 1, \dots, n - 1$, dass

$$A_{ij} = \hat{A}_{i+jm}.$$

```
// dynamic_vectors.c
#include <stdio.h>
#include <stdlib.h>
double* scan_vector(int n) {
    int i = 0;
    double* vec = malloc(n*sizeof(double));
    for (i=0; i<n; ++i) {
        printf("x[%d] = ", i);
        scanf("%lf", &vec[i]);
    }
    return vec;
}
void print_vector(double* vec, int n) {
    int i = 0;
    for (i=0; i<n; ++i) {
        printf("x[%d] = %f\n", i, vec[i]);
    }
}
main() {
    double* x = NULL;
    int dim = 0;
    printf("dim = ");
    scanf("%d", &dim);
    x = scan_vector(dim);
    print_vector(x, dim);
    free(x);
    x = NULL;
}
```

Abb. 33.4 Ein- und Ausgabe von Vektoren beliebiger Länge

```
// matrix.c
#include <stdio.h>
const int m = 3;
const int n = 2;
void mod_matrix(double mat[m][n]) {
    mat[0][0] = 7.0;
    mat[2][1] = 8.0;
}
main() {
    double A[m][n];
    A[0][0] = 1.0; A[0][1] = 2.0;
    A[1][0] = 3.0; A[1][1] = 4.0;
    A[2][0] = 5.0; A[2][1] = 6.0;
    mod_matrix(A);
    printf("A[0][0] = %f\n", A[0][0]);
    printf("A[2][1] = %f\n", A[2][1]);
}
```

Abb. 33.5 Verwendung zweidimensionaler Arrays

Mit dieser Identifikation lassen sich Matrizen in C wie Vektoren behandeln. Gelegentlich ist es übersichtlicher, Matrizen als zweidimensionale Arrays zu behandeln. Ist die Größe bekannt, so kann man sie als Arrays verwenden wie beispielsweise in dem in Abb. 33.5 gezeigten Programm.

33.10 Zeitmessung, Speicherung und Pakete

Die Bibliothek `time.h` stellt den Typen `clock_t`, das Kommando `clock()`, und die Konstante `CLOCKS_PER_SEC` zur Verfügung, mit denen Laufzeitmessungen durchgeführt werden können. Deren Verwendung ist in Abb. 33.6 illustriert.

Zum Speichern von Daten können Befehle aus der Bibliothek `stdio.h` verwendet werden. Mit dem Variablentypen `FILE` und dem Kommando `fopen` kann ein Pointer auf eine Datei definiert werden, in die dann mit `fprintf` geschrieben werden. Ist das Schreiben beendet, so muss die Datei mit `fclose` geschlossen werden. Das in Abb. 33.6 rechts gezeigte Beispielprogramm speichert einen Vektor in der Datei `var.dat` ab. Diese kann von MATLAB mit dem Kommando `load var.dat` ausgelesen werden und weist die Werte des Vektors der Variablen `var` zu.

Die Pakete BLAS und LAPACK stellen Implementationen numerischer Verfahren beispielsweise zur Lösung linearer Gleichungssysteme und Eigenwertaufgaben bereit.

```
// runtime.c // save_data.c
#include <stdio.h> #include <stdio.h>
#include <time.h> main(){
main(){ FILE* file;
int i;
double t, diff, x = 0.33;
clock_t t1, t2;
t1 = clock();
for (i=0; i<10000000; i++){
    x*x;
}
t2 = clock();
diff = (double)(t2-t1);
t = diff/CLOCKS_PER_SEC;
printf("runtime = %fs\n", t); }
} // save_data.c
// save_data.c
#include <stdio.h>
main(){
FILE* file;
int i;
double x[3] = {0.0,1.0,2.0};
file = fopen("var.dat","w");
if (file==NULL){
    printf("file error");
}
for (i=0; i<3; i++){
    fprintf(file,"%f\n",x[i]);
}
fclose(file);
}
```

Abb. 33.6 Laufzeitmessung und Speichern von Daten

34.1 Aufbau

MATLAB steht für *Matrix Laboratory* und ist ein kommerzielles Programm Paket, welches Implementierungen einer Vielzahl numerischer Verfahren bereitstellt und es erlaubt, eigene Programme zu erstellen. Es ist eine Interpreter-Sprache, das heißt Programme sind Folgen von Kommandos, die ohne Kompilierung abgearbeitet werden. Die Benutzeroberfläche besteht im Wesentlichen aus dem *Command Window*, in dem Befehle eingegeben, und einem Editor, in dem Programme erstellt werden können. Diese werden im Format `prog.m` abgespeichert, und können dann in einer Kommandozeile oder von anderen Programmen aus über den Befehl `prog` gestartet werden. Ein Kommando wird mit einem Semikolon abgeschlossen. Geschieht dies nicht, so wird das Resultat der Operation angezeigt. Variablen werden standardmäßig als Typ *double* definiert, sie können jedoch problemlos wie Variablen vom Typ *integer* verwendet werden, beispielsweise bei der Indizierung von Arrays. In der Regel werden Variablen von MATLAB als Matrizen behandelt.

34.2 Listen und Arrays

Zentrale Objekte in MATLAB sind Matrizen beziehungsweise Arrays und Listen. Diese werden mit Hilfe eckiger Klammern definiert. Einträge einer Zeile werden durch Kommas und verschiedene Zeilen durch Semikolons getrennt. Auf die Einträge eines Arrays wird beginnend mit dem Index 1 zugegriffen. Über Indexlisten können Teilmatrizen wie $A_{IJ} = (a_{ij})_{i \in I, j \in J}$ extrahiert werden; statt Indexlisten können dabei auch boolesche Listen verwendet werden. Tabelle 34.1 zeigt einige wichtige Operationen.

Tab. 34.1 Erstellung von Listen und Arrays

<code>[a,b,...;x,y,...]</code>	Definition eines Arrays (Kommas optional)
<code>[a,b,...],[x;y;...]</code>	Definition eines Zeilen- oder Spaltenvektors
<code>A(i,j), I(j)</code>	Zugriff auf die Einträge eines Arrays
<code>a:b, a:step:b</code>	Liste von a bis b mit Schrittweite 1 oder $step$
<code>A(i,:), A(:,j)</code>	i -te Zeile und j -te Spalte von A
<code>A(I,J)</code>	Teilmatrix definiert durch Listen I und J
<code>ones(n,m)</code>	Array mit Einträgen 1
<code>zeros(n,m)</code>	Array mit Einträgen 0
<code>accumarray(I,X)</code>	Konstruktion eines Arrays durch Summierung

34.3 Matrixoperationen

Die grundlegenden Matrixoperationen sind in MATLAB definiert und lassen sich in kanonischer Weise verwenden, wobei die Wohlgestelltheit der Operation sichergestellt werden sollte. Matrixfaktorisierungen und Approximationen von Eigenvektoren und -werten stehen ebenfalls zur Verfügung. Einige Standardroutinen sind in Tab. 34.2 aufgeführt.

Tab. 34.2 Elementare Matrixoperationen

<code>A'</code>	Transponierte Matrix
<code>A+B, A-B, A*B</code>	Addition, Subtraktion und Produkt von Matrizen
<code>inv(A), det(A)</code>	Inverse und Determinante einer Matrix
<code>x = A\b</code>	Lösung des linearen Gleichungssystems $Ax = b$
<code>eye(n)</code>	Einheitsmatrix der Dimension n
<code>A.*B, A./B</code>	Komponentenweise Multiplikation und Division
<code>diag(A)</code>	Extraktion der Diagonalelemente
<code>[L,U] = lu(A)</code>	LU -Faktorisierung einer Matrix
<code>L = chol(A)</code>	Cholesky-Faktorisierung einer Matrix
<code>[Q,R] = qr(A)</code>	QR -Faktorisierung einer Matrix
<code>[V,D] = eig(A)</code>	Approximation von Eigenvektoren und -werten

Tab. 34.3 Manipulation von Arrays

<code>A(:)</code>	Umordnung eines Arrays in einen Spaltenvektor
<code>reshape(A,m,n)</code>	Umordnung eines Arrays als $m \times n$ Array
<code>repmat(A,m,n)</code>	wiederholte Anordnung eines Arrays
<code>unique(A)</code>	Extraktion der Elemente eines Arrays
<code>setdiff(A,B)</code>	Komplement von A und B
<code>sort(A)</code>	Sortierung der Einträge eines Arrays
<code>sum(A,1), sum(A,2)</code>	Spalten- und zeilenweise Summenbildung
<code>max(A), min(A)</code>	Spaltenweise Extremwerte eines Arrays
<code>size(A), length(I)</code>	Dimensionen eines Arrays und Länge einer Liste

Tab. 34.4 Elementare analytische Funktionen

<code>sqrt(x), x^y</code>	Quadratwurzel und Potenzen
<code>exp(x), ln(x)</code>	Exponentielle Funktion und Logarithmus
<code>sin(x), cos(x), pi</code>	Trigonometrische Funktionen und Konstante π
<code>norm(x,p)</code>	p -Norm eines Vektors

34.4 Manipulation von Arrays

Verschiedene mengentheoretische Operationen und Umsortierungen von Arrays sind in Routinen verfügbar. Diese erlauben meist weitere Argumente und Ausgabewerte, mit denen die Ausführung spezifiziert werden kann wie beispielsweise die Bildung des zeilen- oder spaltenweisen Maximums. Tabelle 34.3 zeigt einige nützliche Befehle.

34.5 Elementare Funktionen

Numerische Realisierungen einiger Funktionen sind unter ihren jeweiligen Namen verfügbar. Sie können auf Arrays angewendet werden, was in der Regel die komponentenweise Ausführung realisiert. Bei Ausnahmen wie A^n wird die komponentenweise Ausführung durch $A.^n$ erzeugt. Eine kurze Übersicht findet sich in Tab. 34.4.

34.6 Schleifen und Kontrollanweisungen

Schleifen lassen sich über Listen oder Kontrollanweisungen in naheliegender Weise realisieren. Der Vergleich von Variablen kann auf Arrays angewendet werden. Tabelle 34.5 zeigt einige wichtige Kommandos.

Tab. 34.5 Logische Operationen und Schleifen

<code>a==b, a~=b</code>	Logischer Test auf Gleich- oder Ungleichheit
<code>a<b, a<=b</code>	Logischer Vergleich zweier Zahlen
<code>E&&F, E F</code>	Logisches <i>und</i> beziehungsweise <i>oder</i>
<code>while E ... end</code>	<i>while</i> -Schleife mit booleschem Ausdruck <i>E</i>
<code>for i = I ... end</code>	<i>for</i> -Schleife über Einträge der Liste <i>I</i>
<code>if E ... end</code>	Fallunterscheidung
<code>tic ... toc</code>	Messen der CPU-Zeit

34.7 Text- und Grafikausgabe

Wird ein Programm über eine Kommandozeile gestartet, so können Zwischenergebnisse im Kommandofenster ausgegeben werden. Funktionen oder andere Objekte können in Grafikfenstern sogenannten *figures* dargestellt werden. Eine Auswahl entsprechender MATLAB-Kommandos findet sich in Tab. 34.6.

34.8 Erstellen neuer Funktionen

Neue Funktionen mit mehreren Ein- und Ausgabewerten lassen sich mit dem in Abb. 34.1 gezeigten Rahmen definieren. Dabei ist das abschließende `end` optional. Funktionen sollten als Datei mit dem Namen der Funktion also beispielsweise `new_function.m` abgespeichert werden. Eine Datei kann mehrere Funktionsdefinitionen enthalten, jedoch kann nur die erste von außen über den Dateinamen aufgerufen werden. Dabei muss man sich im Verzeichnis der Datei befinden oder der Pfad muss als Suchpfad eingerichtet worden sein.

Tab. 34.6 Darstellung von Objekten

<code>disp(A)</code>	Anzeigen der Variablen <i>A</i>
<code>plot(X,Y,'-*')</code>	Polygonzug durch Punkte $(X(k), Y(k))$ in \mathbb{R}^2
<code>hold on, hold off</code>	Darstellung mehrerer Objekte in einer Grafik
<code>mesh(X,Y,Z)</code>	Darstellung eines zweidimensionalen Graphen
<code>meshgrid</code>	Erzeugung eines Gitters
<code>axis([x1,x2,...])</code>	Begrenzung des dargestellten Bereichs
<code>xlabel, ylabel</code>	Beschriftung der Achsen
<code>legend</code>	Einfügen einer Legende
<code>figure(k)</code>	Wahl eines Grafikfensters
<code>subplot(n,m,j)</code>	Darstellung mehrerer Plots in einem Fenster
<code>quiver, quiver3</code>	Visualisierung von Vektorfeldern
<code>trisurf</code>	Graph einer Funktion auf einem Dreiecksgitter
<code>tetramesh</code>	Darstellung einer Zerlegung in Tetraeder

Abb. 34.1 Rahmen für eine neu erstellte Funktion

```
function [y1,y2,...] = new_function(x1,x2,...)
...
end
```

34.9 Verschiedene Befehle

Neben einigen Unix-Befehlen wie `cd` und `ls` sind verschiedene Befehle zur Verwaltung der verwendeten Dateien und Verzeichnisse sowie Variablen verfügbar, die in Tab. 34.7 dargestellt sind.

34.10 Dünnbesetzte Matrizen

Bei Matrizen mit vielen verschwindenden Einträgen lässt sich der Aufwand der Lösung zugehöriger linearer Gleichungssysteme reduzieren, sofern die Matrizen über den Matrixtyp *sparse* definiert werden. Für Indexlisten $I \subset \{1, 2, \dots, m\}$ und $J \subset \{1, 2, \dots, n\}$ sowie einen Vektor X gleicher Länge wird eine Matrix $A \in \mathbb{R}^{m \times n}$ definiert durch

$$a_{ij} = \sum_{k : I(k)=i, J(k)=j} X(k),$$

das heißt bei mehrfach auftretenden Indexpaaren werden die zugehörigen Einträge summiert. Der Zugriff auf einzelne Einträge einer dünnbesetzten Matrix ist im Allgemeinen ineffizient. Einige wichtige Kommandos sind in Tab. 34.8 aufgeführt.

Tab. 34.7 Verschiedene Befehle

<code>whos, clear</code>	Anzeigen und Löschen aller Variablen
<code>clc, clf</code>	Löschen des Kommando- oder Grafikfensters
<code>addpath</code>	Hinzufügen eines Suchpfads für Funktionen
<code>save, load</code>	Laden und Speichern von Variablen
<code>Ctrl-C</code>	Abbruch eines Programms
<code>fopen</code>	Öffnen einer Datei
<code>printf</code>	Formatierte Ausgabe
<code>strcat</code>	Zusammenfügen von Zeichenketten

Tab. 34.8 Erzeugung dünnbesetzter Matrizen

<code>sparse(I, J, X, m, n)</code>	Zusammensetzung einer dünnbesetzten Matrix
<code>speye(n)</code>	Einheitsmatrix als dünnbesetzte Matrix

```

>> A = [2,1;1,2]; b = [1;1];
>> x = A\b;
>> x = [pi/2,0,1];
>> sin(x)

x =
0.3333
0.3333
>> x'
ans =
0.3333    0.3333
>>
>>

```

Abb. 34.2 Ausführung von Befehlen im Kommandofenster

34.11 Beispiele

In Abb. 34.2 ist die Eingabe verschiedener Befehle im Kommandofenster von MATLAB dargestellt. Die Berechnung der Determinante einer Matrix nach dem Laplaceschen Entwicklungssatz führt auf eine Rekursion, deren MATLAB-Realisierung in Abb. 34.3 gezeigt ist. Eine Implementation des Bisektionsverfahrens und dessen Anwendung auf eine Funktion $f(x)$ ist ebenfalls in Abb. 34.3 gezeigt.

Die grafische Darstellung verschiedener Funktionen in einem Grafikfenster wird durch das in Abb. 34.4 gezeigte Programm `several_plots.m` illustriert. Die daneben gezeigte Funktion `plot_bubble.m` wertet eine auf \mathbb{R}^2 definierte Funktion $f(x)$ aus und stellt sie grafisch dar. Die durch die Funktionen erzeugten Grafiken sind in Abb. 34.5 gezeigt.

```

function val = laplace(A)
n = size(A,1);
val = 0;
if n == 1
    val = A(1,1);
else
    for j = 1:n
        I = 2:n;
        J = [1:j-1,j+1:n];
        val = val+(-1)^(1+j)...
            *A(1,j)...
            *laplace(A(I,J));
    end
end

function x = bisect(a,b)
x = a; z = b;
tol = 1e-4;
while z-x > tol
    c = (x+z)/2;
    if f(x)*f(c)<0
        z = c;
    else
        x = c;
    end
end

function y = f(x)
y = x^3+cos((pi/2)*x);

```

Abb. 34.3 Berechnung der Determinante nach Laplace (links) und Realisierung des Bisektionsverfahrens (rechts)

```
function several_plots
dx = .1;
X = 0:dx:pi;
Y1 = sin(X); plot(X,Y1,'-r');
hold on;
Y2 = cos(X); plot(X,Y2,':k');
hold off;
legend('sin','cos');
disp('press key'); pause; clf
Z1 = exp(X); plot(X,Z1,'-+');
hold on;
Z2 = log(X); plot(X,Z2,'-*');
hold off;
legend('exp','log');
```

```
function plot_bubble
dx = .1;
dy = .1;
[X,Y] = ...
meshgrid(-2:dx:2,-2:dy:2);
W = f([X(:),Y(:)]);
Z = reshape(W,size(X));
mesh(X,Y,Z);

function y = f(x)
y = zeros(size(x,1),1);
r = sum(x.^2,2).^(1/2);
I = r<1;
y(I) = exp(-1./(1-r(I).^2));
```

Abb. 34.4 Darstellung eindimensionaler Funktionen (links) und einer auf \mathbb{R}^2 definierten Funktion (rechts)

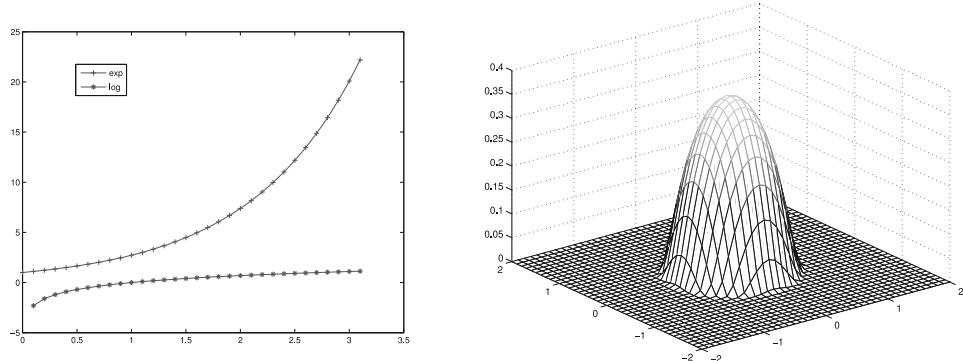


Abb. 34.5 Grafische Ausgaben der Funktionen `several_plots.m` (links) und `plot_bubble.m` (rechts)

34.12 Freie Alternativen

OCTAVE und SCILAB sind frei verfügbare Programm pakete, die mit MATLAB weitestgehend kompatibel sind. SCILAB ist sehr einfach zu installieren, jedoch sind die grafischen Möglichkeiten eingeschränkt und aufwendige Programme sind in der Regel recht langsam. Einige in der Syntax von MATLAB abweichende Befehle sind:

```
eye(n,n), sparse([I,J],X), function ... endfunction
```

Funktionen müssen mittels `execute` eingebunden werden. OCTAVE ist nahezu vollständig kompatibel zu MATLAB. Die Installation ist jedoch etwas aufwendiger.

35.1 LU-Zerlegung und Lösen von Dreieckssystemen

Die Berechnung der LU -Zerlegung ist in MATLAB vorimplementiert und die in Abb. 35.1 gezeigten Kommandos liefern dasselbe Ergebnis wie das Programm `lu_solution.m`. Direkter hätte dies auch mit dem Befehl `x = A\b` geschehen können.

Das in Abb. 35.2 dargestellte MATLAB-Programm `lu_solution.m` berechnet die LU -Zerlegung einer gegebenen Matrix mit dem Algorithmus von Crout, der auf den Identitäten

$$u_{ik} = a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} u_{jk}, \quad \ell_{ki} = \left(a_{ki} - \sum_{j=1}^{i-1} \ell_{kj} u_{ji} \right) / u_{ii}$$

basiert. Dabei wird die gegebene Matrix A mit den Einträgen der Faktoren L und U überschrieben, was aufgrund der Normalisierung von L , das heißt der Bedingung $\ell_{ii} = 1$, $i = 1, 2, \dots, n$, möglich ist. Mit Hilfe der Zerlegung wird dann das Gleichungssystem $Ax = b$ durch explizites Auflösen zweier Gleichungssysteme mit Dreiecksmatrizen gelöst, das heißt

$$Ax = b \iff Ly = b, \quad Ux = y.$$

Bei der dabei verwendeten Vorwärtssubstitution wird wiederum ausgenutzt, dass die Matrix L normalisiert ist. In den Unterroutinen werden nur der obere Dreiecksanteil beziehungsweise der strikte untere Dreiecksanteil der übergebenen Matrix verwendet.

```
>> A = [2, -1, 0; -1, 2, -1; 0, -1, 2]; b = [1; 1; 1];
>> [L, U] = lu(A);
>> y = L\b; x = U\y;
```

Abb. 35.1 Numerische Lösung eines Gleichungssystems mit Hilfe einer von MATLAB berechneten LU -Zerlegung

```

function lu_solution
n = 3;
A = [2,-1,0;-1,2,-1;0,-1,2];
b = [1;1;1];
A = lu_crout(n,A);
y = solve_lower_normalized(n,A,b);
x = solve_upper(n,A,y);
disp(x);

function A = lu_crout(n,A)
for i = 1:n
    for k = i:n
        sum = 0;
        for j = 1:i-1
            sum = sum+A(i,j)*A(j,k);
        end
        A(i,k) = A(i,k)-sum;
    end
    for k = i+1:n
        sum = 0;
        for j=1:i-1
            sum = sum+A(k,j)*A(j,i);
        end
        A(k,i) = (A(k,i)-sum)/A(i,i);
    end
end

function y = solve_lower_normalized(n,L,b)
y = zeros(n,1);
for j = 1:n
    sum = 0;
    for k = 1:j-1
        sum = sum+L(j,k)*y(k);
    end
    y(j) = b(j)-sum;
end

function x = solve_upper(n,U,y)
x = zeros(n,1);
for j = n:-1:1
    sum = 0;
    for k = j+1:n
        sum = sum+U(j,k)*x(k);
    end
    x(j) = (y(j)-sum)/U(j,j);
end

```

Abb. 35.2 Lösung eines linearen Gleichungssystems mit Hilfe einer explizit berechneten LU-Zerlegung in MATLAB (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

```

// lu_solution.c
#include <stdio.h>
const int n = 3;
void lu_crout(double A[n][n]){
    int i, j, k; double sum;
    for (i=0; i<n; i++){
        for (k=i; k<n; k++){
            sum = 0.0;
            for (j=0; j<=i-1; j++) {sum = sum+A[i][j]*A[j][k];}
            A[i][k] = A[i][k]-sum;
        }
        for (k=i+1; k<n; k++){
            sum = 0.0;
            for (j=0; j<=i-1; j++) {sum = sum+A[k][j]*A[j][i];}
            A[k][i] = (A[k][i]-sum)/A[i][i];
        }
    }
}
void solve_lower_normalized(double L[n][n], double b[n],
                           double y[n]){
    int j, k; double sum;
    for (j=0; j<n; j++){
        sum = 0.0;
        for (k=0; k<=j-1; k++) {sum = sum+L[j][k]*y[k];}
        y[j] = b[j]-sum;
    }
}
void solve_upper(double U[n][n], double y[n], double x[n]){
    int j, k; double sum;
    for (j=n-1; j>=0; j--){
        sum = 0.0;
        for (k=j+1; k<n; k++) {sum = sum+U[j][k]*x[k];}
        x[j] = (y[j]-sum)/U[j][j];
    }
}
main(){
    double A[n][n], b[n], x[n], y[n]; int i;
    A[0][0] = 2.0; A[0][1] = -1.0; A[0][2] = 0.0;
    A[1][0] = -1.0; A[1][1] = 2.0; A[1][2] = -1.0;
    A[2][0] = 0.0; A[2][1] = -1.0; A[2][2] = 2.0;
    b[0] = 1.0; b[1] = 1.0; b[2] = 1.0;
    lu_crout(A);
    solve_lower_normalized(A,b,y);
    solve_upper(A,y,x);
    for (i=0; i<n; i++){
        printf("x[%d] = %f\n", i, x[i]);
    }
}

```

Abb. 35.3 Berechnung der LU-Zerlegung und anschließendes Lösen eines Gleichungssystems in C (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

Eine Realisierung in der Programmiersprache C ist in Abb. 35.3 gezeigt. Hier ist zu beachten, dass die Indizierung der Arrays bei Null beginnt. Die Kompilierung und Ausführung des Programms erfolgt mit den folgenden Unix-Kommandos:

```
gcc -lm lu_solution.c -o lu_solution.out
./lu_solution.out
```

35.2 Polynominterpolation und Neville-Schema

Das Neville-Schema erlaubt die Auswertung eines durch Stützstellen x_0, x_1, \dots, x_n und -werte y_0, y_1, \dots, y_n definierten Interpolationspolynoms p an einer Stelle z über die Formel

$$p_{i,j}(z) = \frac{(z - x_i)p_{i+1,j-1}(z) - (z - x_{i+j})p_{i,j-1}(z)}{x_{i+j} - x_i}$$

für $j = 1, 2, \dots, n$ und $i = 0, 1, \dots, n - j$ mit der Initialisierung $p_{i,0}(z) = y_i$ für $i = 0, 1, \dots, n$. Es gilt dann $p(z) = p_{0,n}(z)$. In Abb. 35.5 ist das MATLAB-Programm neville_scheme.m gezeigt, das die Werte des Interpolationspolynoms an den Punkten z_k , $k = 0, 1, \dots, N$, mit dem Neville-Schema berechnet und anschließend das Interpolationspolynom approximativ durch einen Polygonzug durch die Punkte $(z_k, p(z_k))$, $k = 0, 1, \dots, N$ grafisch darstellt. Die Berechnung erfolgt einerseits rekursiv mit der Unterroutine neville_recursive und andererseits durch sukzessives Auswerten der obigen Formel in der Unterroutine neville_forward. Bei Zugriffen auf Arrays werden dabei die Indizes stets um den Wert 1 erhöht, da der Index 0 in MATLAB nicht zulässig ist. Das Endergebnis ist also durch den Eintrag $P(1, n+1)$ gegeben. Statt der Verwendung des Arrays P könnte in der Unterroutine auch die lokale Variable y in jedem Schritt der Schleife über Variable j überschrieben werden, um Speicherplatz zu sparen.

In MATLAB stehen verschiedene Interpolationsmethoden zur Verfügung. Die Funktionswerte eines kubischen Spline-Interpolanten können beispielsweise mit den in Abb. 35.4 gezeigten Kommandos berechnet werden.

Eine zum MATLAB-Programm neville_scheme.m analoge Realisierung in C ist in Abb. 35.6 gezeigt, dessen Kompilierung und Ausführung mit den Unix-Kommandos

```
gcc -lm neville_scheme.c -o neville_scheme.out
./neville_scheme.out
```

erfolgt. Bei der Implementation wurden Listen der Stützstellen und -werte als statische, das heißt vor dem Zeitpunkt des Kompilierens bekannte, Arrays implementiert, während die Liste der Punkte, an denen das Polynom ausgewertet wird, und die zugehörigen Funktionswerte als dynamische Arrays definiert wurden. Im C-Programm ist ein Überschreiben des Arrays y in der Unterroutine neville_forward nicht möglich, da diese Variable als globale Variable vorliegt.

```
>> x = [-1,-1/3,1/3,1]; y = [-1,1,-1,1];
>> N = 100; z = -1+2*[0:N]/N;
>> w = interp1(x,y,z,'spline');
>> plot(z,w);
```

Abb. 35.4 Interpolation von Stützpaaren $(x_j, y_j)_{j=0,\dots,n}$ und Auswertung an den Stellen $(z_k)_{k=0,\dots,N}$ sowie grafische Darstellung mit MATLAB-Kommandos

```
function neville_scheme
n = 3;
x = [-1,-1/3,1/3,1];
y = [-1,1,-1,1];
N = 20; z = zeros(N+1,1);
w_rec = zeros(N+1,1);
w_for = zeros(N+1,1);
for k = 0:N
    z(k+1) = -1+2*k/N;
    w_rec(k+1) = neville_recursive(z(k+1),x,y,0,n);
    w_for(k+1) = neville_forward(z(k+1),x,y,n);
end
plot(z,w_rec,'b-o'); hold on;
plot(z,w_for,'r-x'); hold off;

function val = neville_recursive(z,x,y,i,j)
if j == 0
    val = y(i+1);
else
    val = ((z-x(i+1))*neville_recursive(z,x,y,i+1,j-1)...
        -(z-x(i+j+1))*neville_recursive(z,x,y,i,j-1))/...
        (x(i+j+1)-x(i+1));
end

function val = neville_forward(z,x,y,n)
P = zeros(n+1,n+1);
for i = 0:n
    P(i+1,1) = y(i+1);
end
for j = 1:n
    for i = 0:n-j
        P(i+1,j+1) = ((z-x(i+1))*P(i+2,j)...
            -(z-x(i+j+1))*P(i+1,j))/(x(i+j+1)-x(i+1));
    end
end
val = P(1,n+1);
```

Abb. 35.5 Rekursive und direkte Realisierung des Neville-Schemas zur Auswertung des Lagrange-Interpolationspolynoms durch die Stützpaare (x_i, y_i) , $i = 0, 1, \dots, n$, in MATLAB (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

```

// neville_scheme.c
#include <stdio.h>
#include <stdlib.h>
const int n = 3;
double neville_recursive(double z, double x[n+1], double y[n+1],
    int i, int j){
    if (j==0){
        return y[i];
    }
    else{
        return ((z-x[i])*neville_recursive(z,x,y,i+1,j-1)
            -(z-x[i+j])*neville_recursive(z,x,y,i,j-1))/(x[i+j]-x[i]);
    }
}
double neville_forward(double z, double x[n+1], double y[n+1]){
    double P[n+1][n+1]; int i, j;
    for (i=0; i<=n; i++){
        P[i][0] = y[i];
    }
    for (j=1; j<=n; j++){
        for (i=0; i<=n-j; i++){
            P[i][j] = ((z-x[i])*P[i+1][j-1]-(z-x[i+j])*P[i][j-1])/
                (x[i+j]-x[i]);
        }
    }
    return P[0][n];
}
main(){
    double x[n+1], y[n+1]; int k, N = 20;
    double *z = malloc((N+1)*sizeof(double));
    double *w_rec = malloc((N+1)*sizeof(double));
    double *w_for = malloc((N+1)*sizeof(double));
    x[0] = -1.0; x[1] = -1.0/3; x[2] = 1.0/3; x[3] = 1.0;
    y[0] = -1.0; y[1] = 1.0; y[2] = -1.0; y[3] = 1.0;
    for (k=0; k<=N; k++){
        z[k] = -1.0+2.0*(double)k/N;
        w_rec[k] = neville_recursive(z[k],x,y,0,n);
        w_for[k] = neville_forward(z[k],x,y);
        printf("w_rec = %f, w_for = %f \n",w_rec[k],w_for[k]);
    }
    free(z); free(w_rec); free(w_for);
}

```

Abb. 35.6 Rekursive und direkte Implementation des Neville-Schemas zur Auswertung eines Interpolationspolynoms an verschiedenen Stellen in C (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

35.3 Numerische Lösung gewöhnlicher Differenzialgleichungen

Das implizite Euler-Verfahren approximiert die Lösung eines Anfangswertproblems $y' = f(t, y)$, $y(0) = y_0$, durch die rekursiv definierte Folge

$$y_{k+1} = y_k + \tau f(t_{k+1}, y_{k+1}) = y_k + \tau \Phi(t_k, y_k, y_{k+1}, \tau).$$

Dies erfordert im Allgemeinen die Lösung eines nichtlinearen Gleichungssystems in jedem Zeitschritt, was unter geeigneten Bedingungen mit der Fixpunktiteration

$$z_{i+1} = \Psi(z_i) = y_k + \tau \Phi(t_k, y_k, z_i, \tau)$$

oder einem Newton-Verfahren für die Gleichung

$$F(z) = z - y_k - \tau \Phi(t_k, y_k, z, \tau) = 0,$$

das heißt der Iteration

$$z_{i+1} = z_i - F(z_i)/F'(z_i),$$

erfolgen kann. Als Startwert z_0 wird dabei jeweils die Lösung aus dem vorhergegangenen Zeitschritt verwendet. Beide Zugänge sind in dem in Abb. 35.8 gezeigten MATLAB-Programm realisiert. Um die mit 1 beginnende Indizierung von Arrays in MATLAB zu berücksichtigen, wurde im Programm eine Routine `inc` definiert, die eine gegebene Zahl um den Wert 1 erhöht. Damit kann die Iterationsvorschrift sehr direkt aus dem theoretischen Algorithmus übertragen werden.

Verschiedene Verfahren zur numerischen Lösung von Differenzialgleichungen sind in MATLAB-Routinen bereits vorimplementiert, wie beispielsweise in der Routine `ode45`, welche eine Liste von Zeitpunkten und zugehörigen Approximationen zurückgibt. Die in Abb. 35.7 gezeigten Zeilen geben ein Beispiel für die Verwendung dieser Routine. Andere MATLAB-Routinen zur Lösung gewöhnlicher Differenzialgleichungen mit unterschiedlichen Exaktheits-, Aufwands- und Stabilitätseigenschaften sind die Routinen `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t`, `ode23tb`.

```
>> T = 10; y_0 = 1;
>> f = @(t,y)cos(2*t)*y^2;
>> [t_list,y_list] = ode45(f,[0,T],y_0);
>> plot(t_list,y_list)
```

Abb. 35.7 Numerische Lösung eines Anfangswertproblems mit Hilfe einer MATLAB-Routine

```
function implicit_euler
y_0 = 1; T = 10;
tau = 1/100; K = floor(T/tau);
y(inc(0)) = y_0;
for k = 0:K-1
    t_k = k*tau;
    y(inc(k+1)) = fixed_point_iteration(t_k,y(inc(k)),tau);
    % y(inc(k+1)) = newton_iteration(t_k,y(inc(k)),tau);
end
plot(tau*(0:K),y);

function z = fixed_point_iteration(t,y_old,tau)
z = y_old; diff = 1; eps_stop = tau/10;
while diff > eps_stop
    z_new = y_old+tau*Phi(t,y_old,z,tau);
    diff = abs(z_new-z);
    z = z_new;
end

function z = newton_iteration(t,y_old,tau)
z = y_old; diff = 1; eps_stop = tau/10;
while diff > eps_stop
    F = z-y_old-tau*Phi(t,y_old,z,tau);
    dF = 1-tau*dPhi_y(t,y_old,z,tau);
    z_new = z-F/dF;
    diff = abs(z_new-z);
    z = z_new;
end

function val = Phi(t,y_old,y_new,tau)
val = f(t+tau,y_new);

function val = dPhi_y(t,y_old,y_new,tau)
val = df_y(t+tau,y_new);

function val = f(t,y)
val = cos(2*t)*y^2;

function val = df_y(t,y)
val = cos(2*t)*2*y;

function val = inc(k)
val = k+1;
```

Abb. 35.8 Zwei Realisierungen des impliziten Euler-Verfahrens zur numerischen Lösung einer gewöhnlichen Differenzialgleichung in MATLAB; die Lösung der nichtlinearen Gleichung in jedem Zeitschritt erfolgt über eine Fixpunkt- oder Newton-Iteration (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

```

// implicit_euler.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double f(double t, double y){
    return cos(2.0*t)*pow(y,2.0);
}
double Phi(double t, double y_old, double y_new, double tau){
    return f(t+tau,y_new);
}
void save_solution(double *y, int K){
    FILE* file;
    int k;
    file = fopen("sol.dat", "w");
    if (file==NULL){
        printf("file error");
    }
    for (k=0; k<=K; k++){
        fprintf(file,"%f\n",y[k]);
    }
    fclose(file);
}
main(){
    double y_0 = 1.0, T = 10.0, tau = 1.0/100.0, t_k;
    double z, z_new, diff, eps_stop = tau/10;
    int k, K = floor(T/tau);
    double *y = malloc((K+1)*sizeof(double));
    y[0] = y_0;
    for (k=0; k<K; k++){
        t_k = k*tau;
        z = y[k];
        diff = 1.0;
        while (diff>eps_stop){
            z_new = y[k]+tau*Phi(t_k,y[k],z,tau);
            diff = fabs(z_new-z);
            z = z_new;
        }
        y[k+1] = z;
    }
    save_solution(y,K);
}

```

Abb. 35.9 Implementation des impliziten Euler-Verfahrens in C; die nichtlinearen Gleichungen werden mit einer Fixpunktiteration gelöst (Programm verfügbar unter <http://www.springer.com/978-3-662-48202-5>)

Eine C-Realisierung des impliziten Euler-Verfahrens ist in Abb. 35.9 gezeigt. Die Komplilierung und Ausführung des Programms geschieht mit folgenden Unix-Kommandos:

```

gcc -lm implicit_euler.c -o implicit_euler.out
./implicit_euler.out

```

Anhang A – Weiterführende Themen

Einige wichtige Themen und Konzepte konnten nicht in dieses Buch aufgenommen werden. Diese eignen sich als Vortragsthemen für ein Proseminar, das sich einer Numerik-Vorlesung anschließt.

Numerische lineare Algebra

- Konvergenz des QR -Verfahrens für Eigenwertprobleme
- SOR-Verfahren zur iterativen Lösung linearer Gleichungssysteme
- Stabilitätseigenschaften der Gauß-Elimination
- Störungsresultate für Eigenwerte symmetrischer Matrizen
- Lanczos-Verfahren zur Eigenwertbestimmung
- Aspekte der praktischen Umsetzung des Simplex-Algorithmus

Numerische Analysis

- Lebesgue-Konstante bei numerischer Interpolation
- Fehlerabschätzungen für die Spline-Interpolation
- GMRES-Verfahren und Arnoldi-Prozess
- Euler–Maclaurinsche Formel und Romberg–Quadratur
- Levenberg–Marquardt-Verfahren
- CAD-Methoden

Numerik gewöhnlicher Differenzialgleichungen

- Splitting-Methoden und exponentielle Integratoren
- Kollokations-, Gauß- und Radau-Verfahren
- Analyse von Extrapolationsverfahren
- Diskussion spezieller Runge–Kutta–Verfahren
- Dahlquistsche Grenztheoreme
- Fehlerkonstanten bei Mehrschrittverfahren
- Störmer–Verlet–Verfahren für Hamiltonsche Systeme
- Lagrange–Formulierungen und variationelle Integratoren
- Algebro–Differenzialgleichungen

Anhang B – Literaturhinweise

Bei der Ausarbeitung des dargestellten Materials wurden die folgenden Lehrbücher und Vorlesungsskripte zur Numerik herangezogen:

1. W. Dahmen und A. Reusken, *Numerik für Ingenieure und Naturwissenschaftler*, Springer-Lehrbuch, Springer, 2008.
2. O. Deiser, C. Lasser, E. Vogt, und D. Werner, *12×12 Schlüsselkonzepte zur Mathematik*, Spektrum Akademischer Verlag, 2011.
3. R. W. Freund und R. H. W. Hoppe, *Stoer/Bulirsch: Numerische Mathematik 1*, Springer-Lehrbuch, Springer, 2007.
4. G. Hämmerlin und K.-H. Hoffmann, *Numerische Mathematik*, Springer-Lehrbuch, Springer, 1994.
5. M. Hanke-Bourgeois, *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*, Vieweg+Teubner, 2009.
6. H. Harbrecht, *Einführung in die Numerik und Numerik der Differentialgleichungen*, Vorlesungsskripten, Universität Basel, 2014/2015. Verfügbar unter: <http://jones.math.unibas.ch/~harbrech>
7. R. Plato, *Numerische Mathematik kompakt*, Springer-Vieweg, 2010.
8. D. Praetorius, *Numerische Mathematik*, Vorlesungsskript, TU Wien, 2006.
9. R. Rannacher, *Einführung in die Numerische Mathematik*, Vorlesungsskript, Universität Heidelberg, 2006. Verfügbar unter <http://numerik.uni-hd.de/~lehre/notes/>
10. R. Schaback und H. Wendland, *Numerische Mathematik*, Springer-Lehrbuch, Springer, 2005.
11. E. Süli und D. F. Mayers, *An introduction to numerical analysis*, Cambridge University Press, Cambridge, 2003.

Themen der numerischen linearen Algebra, insbesondere der effizienten Lösung großer linearer Gleichungssysteme finden sich in den folgenden Büchern:

- G. H. Golub und C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996.

- W. Hackbusch, *Iterative Lösung großer schwachbesetzter Gleichungssysteme*, Leitfäden der Angewandten Mathematik und Mechanik, B. G. Teubner, Stuttgart, 1991.
- A. Meister, *Numerik linearer Gleichungssysteme*, Friedr. Vieweg & Sohn, Braunschweig, 1999.
- Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003. Verfügbar unter <http://www-users.cs.umn.edu/~saad/books.html>
- G. Strang, *Linear algebra and its applications*, Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1980.
- L. N. Trefethen und D. Bau, III, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

Detaillierte Stabilitätsanalysen und weiterführende Themen der numerischen Analysis finden sich in den folgenden Büchern:

- N. J. Higham, *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.
- A. R. Krommer und C. W. Überhuber, *Computational integration*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.
- J. M. Ortega und W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.
- M. L. Overton, *Numerical computing with IEEE floating point arithmetic*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

Die numerische Behandlung von Optimierungsproblemen unter Nebenbedingungen wird in den folgenden Büchern behandelt:

- C. Geiger und C. Kanzow, *Theorie und Numerik restringierter Optimierungsaufgaben*, Springer-Lehrbuch, Springer, 2002.
- M. Ulbrich und S. Ulbrich, *Nichtlineare Optimierung*, Mathematik Komapkt, Birkhäuser, 2012.

Die Theorie und Numerik gewöhnlicher Differentialgleichungen ist Gegenstand der folgenden Bücher:

- J. C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons, Ltd. Chichester, 2008.
- P. Deuflhard und F. Bornemann, *Numerische Mathematik II. Integration gewöhnlicher Differentialgleichungen*, Walter de Gruyter & Co., Berlin, 1994.
- E. Hairer, S. P. Norsett, und G. Wanner, *Solving ordinary differential equations I. Non-stiff problems*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1993.

- E. Hairer und G. Wanner, *Solving ordinary differential equations II. Stiff and differential-algebraic problems*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1996.
- A. Iserles, *A first course in the numerical analysis of differential equations*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 1996.
- G. Teschl, *Ordinary differential equations and dynamical systems*, Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 2012. Verfügbar unter <http://www.mat.univie.ac.at/~gerald/ftp/book-ode/>
- W. Walter, *Gewöhnliche Differentialgleichungen*, Springer-Lehrbuch, Springer-Verlag, Berlin, 1993.

Einführungen in MATLAB und C bieten die folgenden Bücher:

- D. J. Higham und N. J. Higham, *MATLAB guide*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
- R. Kirsch und U. Schmitt, *Programmieren in C: Eine mathematikorientierte Einführung*, Springer-Lehrbuch, Springer, 2007.
- A. Quarteroni und F. Saleri, *Scientific computing with MATLAB and Octave*, Texts in Computational Science and Engineering, Springer-Verlag, Berlin, 2006.
- C. W. Überhuber, S. Katzenbeisser, und D. Praetorius, *MATLAB 7. Eine Einführung*, Springer Verlag Wien, 2004.

Die benötigten Resultate der Analysis und linearen Algebra können in den folgenden Büchern nachgelesen werden:

- G. Fischer, *Lineare Algebra: Eine Einführung für Studienanfänger*, Grundkurs Mathematik, Springer Spektrum, 2013.
- O. Forster, *Analysis 1: Differential- und Integralrechnung einer Veränderlichen*, Grundkurs Mathematik, Springer Spektrum, 2013.
- O. Forster, *Analysis 2: Differentialrechnung im \mathbb{R}^n , gewöhnliche Differentialgleichungen*, Grundkurs Mathematik, Springer Spektrum, 2013.

Anhang C – Notation

Zahlen, Vektoren und Matrizen

\mathbb{Z}	ganze Zahlen
\mathbb{N}, \mathbb{N}_0	positive und nichtnegative ganze Zahlen
\mathbb{R}, \mathbb{C}	reelle und komplexe Zahlen
$\mathbb{R}_{\geq 0}$	nichtnegative reelle Zahlen
$[s, t], (s, t)$	abgeschlossenes und offenes Intervall
\mathbb{R}^d	n -dimensionaler Euklidischer Raum
$\mathbb{R}^{n \times m}$	Menge der $n \times m$ -Matrizen
$B_r(x), K_r(x)$	offene und abgeschlossene Kugel mit Radius r um x
$A \subset B$	A ist Teilmenge von B oder $A = B$
$x = (x_i), A = (a_{ij})$	Spaltenvektor und Matrix
x^\top, A^\top	Transposition eines Vektors oder einer Matrix
$\ \cdot\ $	Norm eines Vektors oder Operatornorm einer Matrix
$x \cdot y = x^\top y$	Skalarprodukt der Vektoren $x, y \in \mathbb{R}^n$
$x \times y$	Kreuzprodukt der Vektoren $x, y \in \mathbb{R}^3$
$x \perp y$	x ist orthogonal zu y
I_n	$n \times n$ Einheitsmatrix
$O(n)$	orthogonale Gruppe
$[x, y]^\top, (x, y)$	Vektor mit Einträgen x und y
$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}$	Matrix mit Einträgen x_1, x_2, y_1, y_2
$i = \sqrt{-1}$	imaginäre Einheit

Verschiedene Symbole

$o(s^p)$, $\mathcal{O}(s^p)$, $\mathcal{O}(n^p)$	Landau-Symbole
$\lfloor r \rfloor$	maximale Zahl $k \in \mathbb{Z}$ mit $k \leq r$
$ A $	Kardinalität einer endlichen Menge
C, \tilde{C}	Konsistenzterme
$N_p(1)$	Niveaumenge der p -Norm
$a \ll b$	a ist wesentlich kleiner als b
$a \approx b$	a und b sind nahezu gleich groß
$a \sim b$	a ist proportional zu b

Lineare Abbildungen

$\ker A$	Kern der linearen Abbildung A
$\text{Im } A$	Bild der linearen Abbildung A
$\text{rank } A$	Rang der linearen Abbildung A
$\dim W$	Dimension des Vektorraums W
$\det A$	Determinante der Matrix A
$\text{tr } A$	Spur der Matrix A

Differentialoperatoren

$\partial_i, \partial_{x_i}, \frac{\partial}{\partial x_i}$	partielle Ableitung bezüglich des i -ten Arguments
∇f	Gradient der Funktion f
$\text{div } F$	Divergenz des Vektorfelds F
$Df, D^2 f$	Jacobi- und Hesse-Matrix einer Funktion f
y_t	Zeitableitung der Funktion y

Funktionenräume

$C^k([a, b])$	k -fach stetig differenzierbare Funktionen auf $[a, b]$
\mathcal{P}_m	Polynome vom Grad m
$S^{m,k}(\mathcal{T}_h)$	stückweise vom Grad m polynomische, k -fach stetig differenzierbare Funktionen

Sachverzeichnis

A

Abstiegsverfahren, 127
Adams–Bashforth-Verfahren, 200
Adams–Moulton-Verfahren, 200
adaptiver Algorithmus, 230
 A -konjugiert, 132
Algorithmus, 6
Anfangsbedingung, 161
Anfangswertproblem, 161
Approximation, 3
approximativer Lösen, 3
Armijo-Bedingung, 127
 A -stabil, 218
asymptotisch, 7
asymptotischer Bereich, 122
Aufwand, 7
Ausgleichsproblem, 31
Auslöschung, 7

B

Backward-Differentiation-Formulas, 200
Banachscher Fixpunktsatz, 63
Bandmatrix, 141
Bandweite, 71
Begleitmatrix, 210
Bild, 336
Bisektionsverfahren, 123
Butcher-Tableau, 189

C

CG -Verfahren, 135
charakteristisches Polynom, 337
Cholesky-Zerlegung, 19
Cholesky-Zerlegung, unvollständig, 145

D

Dahlquiastsche Wurzelbedingung, 211

Datenfehler, 3
Determinante, 336
diagonaldominant, 67
diagonalisierbar, 337
Differenzengleichung, 210
Differenzenquotient, 177
Differenzialgleichung, autonom, 164
diskontinuierliches Galerkin-Verfahren, 240
diskrete Suche, 126
Diskretisierungsfehler, 180
dividierte Differenzen, 87
double precision, 77
Dreiecksmatrix, 15
dünnbesetzt, 141, 355

E

Ecke, 44
Eigenwert, 337
Eigenwertaufgabe, 49
Einheitswurzel, 103
Einschrittverfahren, 178
Elemente, 150
Eliminationsverfahren, 23
Euklidische Norm, 9
Euler-Verfahren, 178
Euler-Verfahren, partitioniert, 236
Exaktheit, 192
Exaktheitsgrad, 110
experimentelle Konvergenzordnung, 118
explizites Verfahren, 178
Extrapolation, 118

F

Fill-In, 145, 146
Fluss, 234
Fourier-Basis, 103

Fourier-Synthese, 104
 Fourier-Transformation, 104
 Frobenius-Norm, 11
 Fundamentalsatz, 340, 341

G

Gauß-Quadratur, 114, 156
 Gaußsche Normalengleichung, 31
 Gauß-Seidel-Verfahren, 67
 Gerschgorin-Kreise, 49
 Gewichtsfunktion, 115
 gewöhnliche Differenzialgleichung, 161
 Gitter, 149
 Givens-Rotation, 58
 Gleitkommazahl, 77
 globale Konvergenz, 64, 121
 Gradient, 342
 Gradientenfluss, 222
 Gradientenverfahren, 127

H

Hermite-Interpolation, 90
 Hesse-Matrix, 342
 Horner-Schema, 87
 Householder-Transformation, 34
 Hutfunktion, 93, 154

I

implizites Verfahren, 178
 Indexmenge, 44
 Inkrementfunktion, 178
 Integralkurve, 165
 Interpolant, 155, 227
 Interpolation, trigonometrisch, 101
 Interpolationsaufgabe, 95
 Interpolationspolynom, 84
 Intervallverkleinerung, 126
 inverse Iteration, 56
 irreduzibel, 68
 iteratives Verfahren, 65

J

Jacobi-Matrix, 342
 Jacobi-Verfahren, 60, 67
 Jordan-Normalform, 338

K

Keplersche Fassregel, 112
 Kern, 336
 Knoten, 83, 150

koerziv, 222
 Konditionierung, 5, 75
 Konditionszahl, 12, 75
 konjugierte Vektoren, 132
 Konsistenz, 179, 201
 Konsistenzordnung, 180
 Kontraktion, 63
 Kontrollverfahren, 230
 Konvergenzordnung, 114, 122, 182
 konkav, 222
 Krylov-Raum, 134

L

Lagrange-Darstellung, 340
 Lagrange-Interpolation, 83
 Lagrange-Polynom, 84
 Landau-Notation, 7
 Landau-Symbol, 341
 leapfrog-Verfahren, 200
 Lemma von Gronwall, diskret, 181
 Lemma von Gronwall, kontinuierlich, 172
 lineares Programm, 43
 lokale Konvergenz, 121
 L-stabil, 221
 LU-Zerlegung, 16

M

Maschinengenauigkeit, 78
 Maschinenzahl, 3, 77
 mathematische Aufgabe, 3
 Mehrkörperproblem, 165, 233
 Mehrschrittverfahren, 200
 Methode der kleinsten Quadrate, 31
 Minimierungsproblem, 121
 Mittelpunktregel, 112
 Mittelpunktverfahren, 179
 Mittelwertsatz, 340
 Modellfehler, 3
 Moore-Penrose Inverse, 41

N

Netzweite, 150
 Neville-Schema, 86, 362
 Newton-Basis, 87
 Newton-Cotes-Formel, 111
 Newtonsches Abkühlungsgesetz, 161
 Newton-Verfahren, 124
 nodale Basis, 155
 Norm, 9

Normalform, 43

normalisierte Dreiecksmatrix, 15

normalisierte LU-Zerlegung, 16

Not-a-Number (NaN), 78

Nullstabilität, 211

Nullstellensuche, 121

O

Operatornorm, 10

Ordnung, 7

Ordnung einer Differenzialgleichung, 163

orthogonal, 335

orthogonale Matrix, 33

Orthogonalpolynom, 115

Overflow, 78

P

partieller Grad, 151

Permutationsmatrix, 26

Phasendiagramm, 165

Pivotsuche, 26

Polygonzugverfahren, 178

positiv definit, 18

positiv semidefinit, 18

Potenzmethode, 52

Prädiktor-Korrektor-Verfahren, 204

Pseudoinverse, 41

Q

QR-Verfahren, 56

QR-Zerlegung, 35

Quadraturformel, 109, 152, 156

Quadraturformel, summiert, 112

R

Randwertproblem, 239

Rang, 336

Räuber-Beute-Modell, 162

Rayleigh-Quotienten, 50

Rechenaufwand, 3

reduzibel, 68

regula-falsi-Verfahren, 124

relativer Fehler, 5

Residuum, 31, 133

Restglied, 340

Richardson-Verfahren, 66

Rückwärtssubstitution, 15

Rundung, 78

Rundungsfehler, 3

Runge–Kutta-Verfahren, 189

S

Satz von Bolzano–Weierstraß, 339

Satz von Peano, 172

Satz von Picard–Lindelöf, 170

Satz von Rolle, 340

Schießverfahren, 239

schnelle Fourier-Transformation, 105

Schrittweite, 177

Schrittweitensteuerung, 230

Sekantenverfahren, 123

Separation der Variablen, 166

Simplex, 150

Simplex-Verfahren, 47

Simpson-Regel, 112

single precision, 77

Singulärwertzerlegung, 40

Skalarprodukt, 335

Spaltensummennorm, 11

Spektralnorm, 11

Spektralradius, 11, 65

Spektrum, 337

Spline, 93, 154

Stabilität, 6, 79, 173

Stabilitätsfunktion, 219

steife Differenzialgleichung, 218

Stützstelle, 83

Stützwerte, 83

symplektische Abbildung, 234

symplektisches Verfahren, 236

T

Taylor-Formel, 340

Tensorgitter, 149

totaler Grad, 151

Trapezregel, 112

Trapezverfahren, 190

Triangulierung, 150

Tschebyscheff-Knoten, 88

Tschebyscheff-Polynom, 88

U

unbedingt stabil, 218

Underflow, 78

uniforme Triangulierung, 150

uniformes Gitter, 149

V

Vandermonde-Matrix, 84

Variablen, 161

Variation der Konstanten, [166](#)

Verfahren, [6](#)

Vorkonditionierungsmatrix, [142](#)

W

Wärmeleitungsgleichung, [223](#)

Z

Zeilenäquilibrierung, [143](#)

Zeilensummennorm, [11](#)

Zeitschritte, [177](#)

zulässige Menge, [43](#)

Zweikörperproblem, [165](#)

Zwischenwertsatz, [339](#)

Zyklen, [47](#)