Skriptum zur Vorlesung
Numerik (Lehramt)

Walter Zulehner
Institut für Numerische Mathematik
Johannes Kepler Universität Linz

Sommersemester 2012

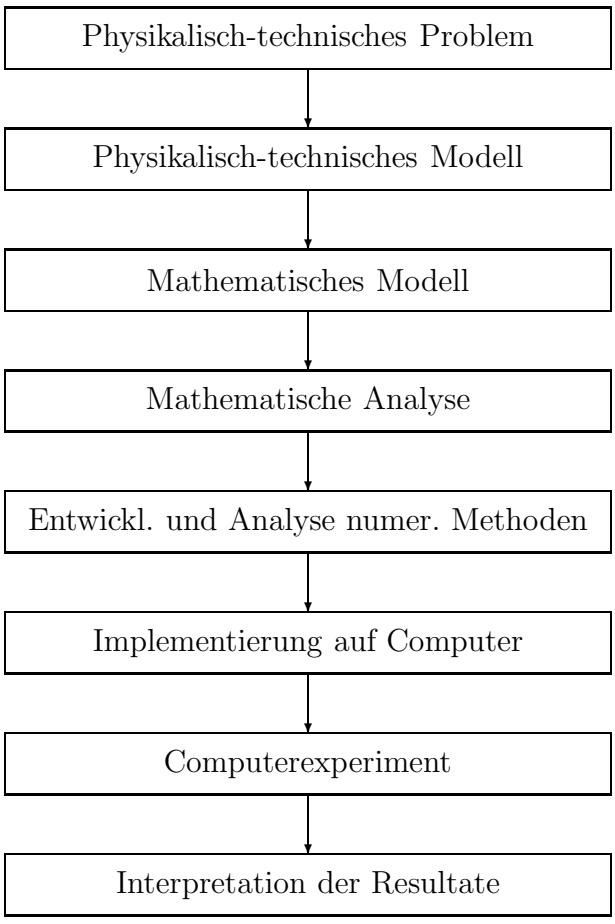
Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Der Problemlösungsprozess

Der Prozess des Lösens eines Anwendungsproblems mit numerischen Methoden lässt sich grob in folgende Stufen einteilen:



Das physikalisch-technische Modell berücksichtigt z.B. die relevanten Bilanzgleichungen, Materialgesetze, Minimalprinzipien, u.ä.

Dabei ist zu beachten, dass beim Übergang von einer realen Problemstellung auf ein Modell, also beim Modellieren, zwangsweise eine Reihe von Vereinfachungen zu treffen sind. Hier passiert bereits ein erster Fehler (**Modellfehler**) durch die gerechtfertigten oder ungerechtfertigten Modellannahmen.

Der Übergang vom physikalisch-technischen Modell zum mathematischen Modell ist fließend. Die getroffene Unterscheidung soll nur den unterschiedlichen Grad der Mathematisierung des Modells zum Ausdruck bringen. Am Ende steht im Idealfall ein wohldefiniertes mathematisches Problem, wie z.B. ein Anfangsrandwertproblem für eine partielle Differentialgleichung mit vorgegebenen Eigenschaften.

Der zentrale Begriff der mathematischen Analyse ist das korrekt gestellte Problem, d.h., eine Lösung des Problems existiert, sie ist eindeutig und sie hängt stetig von den Daten ab. Die in vielen Fällen als selbstverständlich geltende Existenz und Eindeutigkeit einer Lösung des realen Problems bedeutet keineswegs, dass auch ein Modell diese Eigenschaften besitzen muss. Gelingt der Nachweis der Existenz und Eindeutigkeit einer Lösung, so ist eine Mindestanforderung an ein „sinnvolles“ Modell erfüllt. Die stetige Abhängigkeit von den Daten sichert die Richtigkeit einer Lösung auch bei kleinen Störungen in den Daten. Solche **Datenfehler** können z.B. durch Messfehler oder durch Fehler aufgrund einer vorangegangenen Rechnung entstanden sein.

Eine numerische Methode muss natürlich vor allem konstruktiv sein und in vernünftiger Zeit zu einem Ergebnis führen. Das hat häufig zur Folge, dass man sich mit einer Näherung der Lösung begnügen muss. Der in Kauf genommene Fehler wird **Verfahrensfehler** genannt.

Die Realisierung einer numerischen Methode auf dem Computer führt zu einer weiteren Verfälschung der Ergebnisse. Zahlen lassen sich nicht exakt darstellen und die Operationen können nicht exakt durchgeführt werden. Fehler dieser Art werden als **Rundungsfehler** bezeichnet.

Steht schließlich ein Berechnungsprogramm zur Verfügung, so ist es ein (weiteres) Werkzeug, das zur Analyse oder Verbesserung von Produkten oder Prozessen eingesetzt werden kann. Durch Variation der Daten lassen sich Experimente am Computer durchführen, die gewonnenen Erkenntnisse ergänzen (gelegentlich ersetzen sogar) Kenntnisse aus realen Experimenten.

Schließlich erhält man konkrete Zahlen als Resultat eines Programms, deren Aussagekraft in Beziehung zu den gemachten Fehlern bewertet werden muss. Dies kann zu Modellmodifikationen, anderen numerischen Methoden oder deren besserer Implementierung auf einem Computer führen.

1.2 Ein Beispiel

Problemstellung:

Es soll die zeitliche Entwicklung der Abkühlung eines Metallstabes bei gegebenem Anfangszustand vorhergesagt werden.

Modellierung:

Für die Modellierung wird angenommen, dass der Stab ein homogenes eindimensionales Kontinuum der Länge L ist, und dass der Wärmestrom nur in axialer Richtung zu berücksichtigen ist. Der Stab habe konstante Materialeigenschaften und keine inneren Wärmequellen. Es wird angenommen, dass Wärmetransport nur durch Wärmeleitung erfolgt.

Der Stab sei durch ein Intervall $[a, b]$ mit $L = b-a$ auf der x-Achse dargestellt. $|S|$ sei die konstante Querschnittsfläche. Der Stab besitze eine bekannte Temperaturverteilung $T_A(x)$ im Anfangszeitpunkt t_A. Die Temperaturverteilung $T(x, t)$ bis zu einem Endzeitpunkt t_E ist gesucht.

In einem beliebigen Abschnitt $[x_1, x_2]$ mit $\Delta x = x_2 - x_1$ wird die Wärmemenge für ein beliebiges Zeitintervall $[t_1, t_2]$ mit $\Delta t = t_2 - t_1$ bilanziert:

Der Unterschied der Wärmemenge zu den Zeitpunkten t_1 und t_2, gegeben durch

$$\int_{x_1}^{x_2} \rho\, c\, T(x, t_2)\, |S|\; dx - \int_{x_1}^{x_2} \rho\, c\, T(x, t_1)\, |S|\; dx$$

mit der Dichte ρ und der spezifischen Wärmekapazität c, muss gleich der Wärmemenge sein, die durch die Querschnitte bei x_1 und x_2 im Zeitintervall $[t_1, t_2]$ einfließt.

Die Gültigkeit des Fourierschen Gesetzes wird vorausgesetzt:

$$\dot{q}_n = -\lambda \frac{\partial T}{\partial n},$$

$\dot{q}_n$ bezeichnet die Wärmestromdichte im betrachteten Querschnitt mit Richtung n, λ ist die Wärmeleitzahl, $\partial T/\partial n$ ist die Richtungsableitung der Temperatur in Richtung n.

Daraus ergibt sich für die einfließende Wärmemenge durch den Querschnitt bei x_2:

$$\int_{t_1}^{t_2} \lambda \frac{\partial T}{\partial x}(x_2, t)\, |S|\; dt$$

bzw. durch den Querschnitt bei x_1:

$$-\int_{t_1}^{t_2} \lambda \frac{\partial T}{\partial x}(x_1, t)\, |S|\; dt.$$

Es wird vorausgesetzt, dass keine Wärmemenge durch den Mantel fließt und dass keine zusätzlichen Wärmequellen im Stab vorhanden sind.

Somit erhält man das folgende erste Modell:
Gesucht ist die Temperaturverteilung $T(x,t)$, sodass

$$\int_{x_1}^{x_2} \rho\, c\, (T(x,t_2) - T(x,t_1))\ |S|\ dx = \int_{t_1}^{t_2} \lambda\, \left(\frac{\partial T}{\partial x}(x_2,t) - \frac{\partial T}{\partial x}(x_1,t) \right) |S|\ dt \qquad (1.1)$$

für alle $x_1, x_2 \in (a,b)$, $t_1, t_2 \in (t_A, t_E)$. Zusätzlich werden folgende Randbedingungen

$$\begin{aligned} T(a,t) &= T_a(t) \quad \text{für alle } t \in (t_A, t_E) \\ T(b,t) &= T_b(t) \quad \text{für alle } t \in (t_A, t_E) \end{aligned}$$

mit gegebenen (Umgebungs-)Temperaturen $T_a(t)$ und $T_b(t)$ und die Anfangsbedingung

$$T(x,t_A) = T_A(x), \quad \text{für alle } x \in [a,b]$$

mit gegebener Anfangstemperatur $T_A(x)$ vorausgesetzt.

Damit diese Gleichungen Sinn machen, muss offensichtlich $T(x,t)$ bezüglich x und t stetig auf $\bar{Q} = [a,b] \times [t_A, t_E]$ sein, kurz $T \in C^{0,0}(\bar{Q})$, und zusätzlich bezüglich x auf $Q = (a,b) \times (t_A, t_E)$ stetig differenzierbar sein, kurz $T \in C^{1,0}(Q)$, insgesamt also $T \in C^{0,0}(\bar{Q}) \cap C^{1,0}(Q)$. Man spricht von einem Modell in integraler Form.

Wählt man für beliebige Werte $x \in (a,b)$ und $t \in (t_A, t_E)$ und hinreichend kleine Werte $\Delta x > 0$ und $\Delta t > 0$ speziell

$$x_1 = x - \frac{\Delta x}{2}, \quad x_2 = x + \frac{\Delta x}{2}, \quad t_1 = t - \frac{\Delta t}{2}, \quad t_2 = t + \frac{\Delta t}{2},$$

so erhält man aus (1.1) nach Division mit $\Delta x\, \Delta t$ und dem Grenzübergang $\Delta x \to 0$, $\Delta t \to 0$ das folgende zweite Modell:
Gesucht ist die Temperaturverteilung $T(x,t)$, sodass

$$\rho\, c\, \frac{\partial T}{\partial t}(x,t) = \lambda\, \frac{\partial^2 T}{\partial x^2}(x,t) \quad \text{für alle } (x,t) \in Q.$$

Zusätzlich werden wie vorhin folgende Randbedingungen

$$\begin{aligned} T(a,t) &= T_a(t), \quad \text{für alle } t \in (t_A, t_E) \\ T(b,t) &= T_b(t), \quad \text{für alle } t \in (t_A, t_E) \end{aligned}$$

mit gegebenen (Umgebungs-)Temperaturen $T_a(t)$ und $T_b(t)$ und die Anfangsbedingung

$$T(x,t_A) = T_A(x), \quad \text{für alle } x \in [a,b]$$

mit gegebener Anfangstemperatur $T_A(x)$ vorausgesetzt.

Damit diese Gleichungen Sinn machen, muss offensichtlich $T(x,t)$ bezüglich x und t stetig auf $\bar{Q} = [a,b] \times [t_A, t_E]$, bezüglich x auf $Q = (a,b) \times (t_A, r_E)$ zweimal stetig differenzierbar und bezüglich t auf $Q = (a,b) \times (t_A, r_E)$ einmal stetig differenzierbar sein, kurz $T \in C^{0,0}(\bar{Q}) \cap C^{2,1}(Q)$. Man spricht bei diesem Modell in differentieller Form von einem Anfangsrandwertproblem für eine partielle Differentialgleichung.

Mathematische Analyse:

In Räumen geeignet oft differenzierbarer Funktionen lässt sich die Korrektgestelltheit des Problems bei hinreichend glatten Daten nachweisen.

Im Weiteren gelten ohne Beschränkung der Allgemeinheit folgende Setzungen: $a = 0$, $b = L$, $t_A = 0$. Zusätzlich wird ab nun mit a die Temperaturleitzahl bezeichnet: $a = \lambda/(\rho\, c)$.

Numerische Methode:

In einem ersten Schritt wird das im Ort kontinuierliche (unendlichdimensionale) Problem durch ein im Ort diskretes (endlichdimensionales) Problem approximiert. Man spricht von Semidiskretisierung bezüglich der Ortsvariablen:

Das kontinuierliche Intervall $[0, L]$ wird durch eine endliche Punktmenge (Gitterpunkte), z.B. $x_i = i\, h$, $i = 0, 1, \ldots, N, N+1$ mit $h = L/(N+1)$ ersetzt.

Die 2. Ortsableitung in einem Gitterpunkt x_i wird durch Differenzenquotienten approximiert:

$$\begin{aligned}
\frac{\partial^2 T}{\partial x^2}(x_i, t) &\approx \frac{1}{h}\left[\frac{\partial T}{\partial x}(x_i + \frac{1}{2}h, t) - \frac{\partial T}{\partial x}(x_i - \frac{1}{2}h, t)\right] \\
&\approx \frac{1}{h}\left[\frac{T(x_{i+1}, t) - T(x_i, t)}{h} - \frac{T(x_i, t) - T(x_{i-1}, t)}{h}\right] \\
&= \frac{1}{h^2}\left[T(x_{i-1}, t) - 2T(x_i, t) + T(x_{i+1}, t)\right].
\end{aligned}$$

Ersetzt man in der ursprünglichen Differentialgleichung die 2. Ortsableitung durch die obige Differenzenapproximation, so entsteht folgendes System gewöhnlicher Differentialgleichungen für die Näherungen $T_i(t)$ von $T(x_i, t)$:

$$\frac{dT_i}{dt}(t) = \frac{a}{h^2}\left[T_{i-1}(t) - 2T_i(t) + T_{i+1}(t)\right] \quad \text{für } i = 1, 2, \ldots, N$$

mit den Randbedingungen (Dirichlet-Randbedingungen)

$$T_0(t) = T_a(t), \quad T_{N+1}(t) = T_b(t)$$

und der Anfangsbedingung

$$T_i(0) = T_A(x_i) \quad \text{für } i = 0, 1, \ldots, N+1.$$

Bemerkung: Die hier vorgestellte Methode der Diskretisierung bezüglich der Ortvariablen ist eine so genannte Finite Differenzen Methode (FDM). Sie ist nur eine von vielen Möglichkeiten, eine endliche Zahl von Ortsfreiheitsgraden zu erhalten. Andere Techniken sind u.a. Finite Elemente Methoden (FEM), Randelementemethoden (BEM) und Finite Volumen Methoden (FVM).

Im nächsten Schritt wird die Zeit diskretisiert:

Das interessierende Zeitintervall $[0, t_E]$ wird durch eine endliche Zahl von Zeitpunkten $t_j = j\,\tau$, $j = 0, 1, \ldots, m$ mit $\tau = t_E/m$ ersetzt.

Für die Approximation der Zeitableitung wird der so genannte Vorwärtsdifferenzenquotient verwendet:

$$\frac{dT_i}{dt}(t_j) \approx \frac{T_i(t_{j+1}) - T_i(t_j)}{\tau}.$$

Ersetzt man im obigen System gewöhnlicher Differentialgleichungen die 1. Zeitableitung durch diesen Differenzenquotienten, so erhält man für die Näherungen T_i^j von $T(x_i, t_j)$:

$$\frac{T_i^{j+1} - T_i^j}{\tau} = \frac{a}{h^2}\left(T_{i-1}^j - 2T_i^j + T_{i+1}^j\right) \quad \text{für } i = 1, 2, \ldots, N,\ j = 0, 1, \ldots, m.$$

Also:

$$T_i^{j+1} = T_i^j + r\left(T_{i-1}^j - 2T_i^j + T_{i+1}^j\right) \quad \text{für } i = 1, 2, \ldots, N,\ j = 0, 1, \ldots, m.$$

mit $r = a\tau/h^2$.

Zusammen mit den Randbedingungen

$$T_0^j = T_a(t_j) \quad T_{N+1}^j = T_b(t_j) \quad \text{für } j = 1, 2, \ldots, m$$

und der Anfangsbedingung

$$T_i^0 = T_A(x_i) \quad \text{für } i = 0, 1, \ldots, N+1$$

kann die obige Differenzengleichung dazu benutzt werden, um Näherungen T_i^j für die Temperatur im Punkt x_i zur Zeit t_j schichtweise in der Zeit zu berechnen.

Die vorgestellte numerische Methode lässt sich leicht implementieren.

Computerexperiment:

Für einen Kupferstab ($\rho = 8930$ kg m^{-3}, $c = 394$ J kg^{-1} K^{-1}, $\lambda_K = 385$ W m^{-1} K^{-1}, mantelisoliert, wärmequellenfrei) der Länge $L = 1$ m, der an beiden Randpunkten die fixierten Temperaturwerte $T_a = 20°$, $T_b = 40°$ und zum Zeitpunkt $t_A = 0$ die Temperaturverteilung

$$T_A(x) = T_a + (T_b - T_a)\left[\frac{x}{L} + \sin\left(\frac{\pi x}{L}\right)\right]$$

besitzt, soll die Temperatur nach zwei Stunden ($t_E = 7220$ s) ermittelt werden.

Abbildung 1.1 zeigt neben der Anfangstemperaturverteilung das Ergebnis der numerischen Methode zum Zeitpunkt $t = 5220$ s für $h = 10$ cm und $\tau = 1$ min. Der Versuch, das Ergebnis durch kleinere Schrittweiten $h = 1$ cm und $\tau = 1$ s scheitert bereits nach 30 Sekunden, wie Abbildung 1.2 belegt. Hingegen erhält man mit der geringfügigen Modifikation $h = 10$ cm und $\tau = 50$ s ein sinnvolles Resultat, nämlich den erwarteten (annähernd) linearen Temperaturverlauf zwischen den beiden vorgegebenen Randtemperaturen, siehe

Abbildung 1.3. Hätte man das Ergebnis nach der scheinbar besseren (weil kleineren) Wahl $h = 1$ cm und $\tau = 1$ s beurteilt, wäre man zu einer völlig falschen Interpretation der Resultate gekommen. Offensichtlich dominiert bei dieser Parameterwahl der numerische Verfahrensfehler, sodass keine sinnvollen Rückschlüsse auf das physikalisch richtige Ergebnis gezogen werden können. Die Vorlesung soll unter anderem dazubeitragen, solche möglichen Falschinterpretationen zu vermeiden. Dazu ist es notwendig, die numerischen Methoden besser zu verstehen.

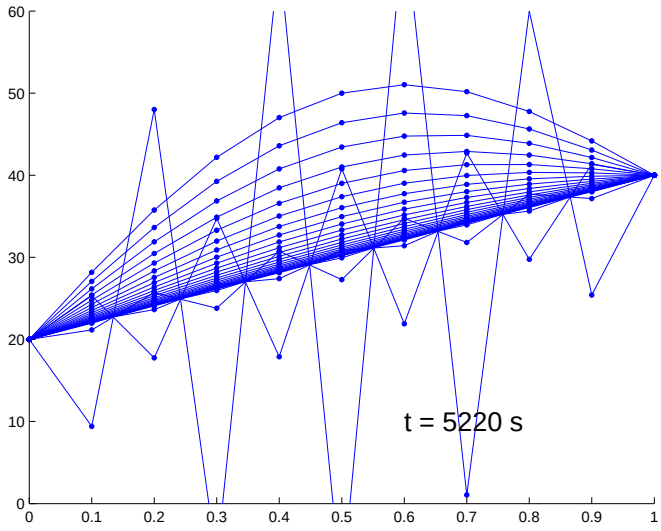


Abbildung 1.1: $h = 10$ cm, $\tau = 60$ s

1.3 Problemstellungen

Einige wichtige Modellgleichungen aus den verschiedensten technischen Bereichen sind die Wärmeleitgleichung, die Naviersche Gleichungen (in der Festigkeitslehre), die Navier-Stokes-Gleichungen (in der Strömungsmechanik), die Maxwell-Gleichungen (Elektromagnetismus), Dynamische Systeme (Automatisierungstechnik), u.s.w.

Solche und andere Problemstellungen können grob unterschieden werden in:

- Ortskontinuierliche und ortsdiskrete Probleme.
- Stationäre und instationäre Probleme.

Die wichtigste Unterscheidung bezüglich der Komplexität von Problemen ist die Einteilung in

- lineare und nichtlineare Probleme.

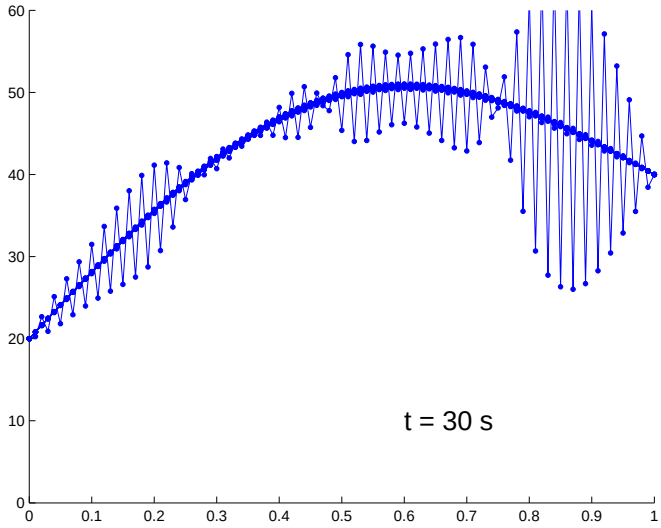


Abbildung 1.2: $h = 1$ cm, $\tau = 1$ s

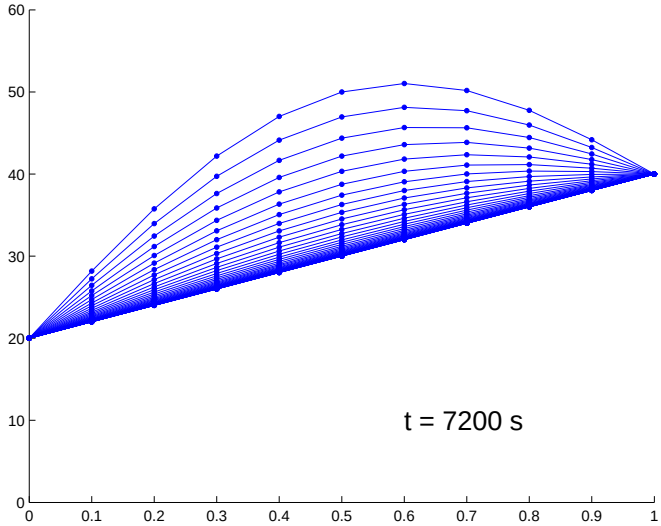


Abbildung 1.3: $h = 10$ cm, $\tau = 50$ s

Nach einem einführenden Kapitel über Besonderheiten des Numerischen Rechnens werden folgende typische Problemstellungen näher untersucht:

- Lineare Gleichungssysteme (Sie treten bei linearen stationären Problemen auf, sie

sind aber auch ein wichtiger Baustein bei den anderen Problemstellungen.)

- Nichtlineare Gleichungen (zur Beschreibung nichtlinearer stationärer Probleme)
- Gewöhnliche Differentialgleichungen (zur Beschreibung instationärer Probleme)

Kapitel 2

Besonderheiten des Numerischen Rechnens

2.1 Zahlendarstellung

Jede reelle Zahl $x \neq 0$ lässt sich folgendermaßen darstellen:

$$x = \pm \left(\alpha_m 10^m + \alpha_{m-1} 10^{m-1} + \alpha_{m-2} 10^{m-2} + \ldots\right)$$

mit $m \in \mathbb{Z}$, $\alpha_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $\alpha_m \neq 0$.

Schreibweise: $x = \pm \alpha_m \ldots \alpha_1 \alpha_0, \alpha_{-1} \alpha_{-2} \ldots$ für $m \geq 0$, $x = \pm 0, 0 \ldots 0 \alpha_m \alpha_{m-1} \alpha_{m-2} \ldots$ für $m < 0$ mit $|m - 1|$ Nullen zwischen Komma und α_m.

Diese Darstellung lässt sich nicht nur für die Zahl 10 sondern allgemein für eine beliebige positive ganze Zahl $B \neq 1$ durchführen:

$$x = \pm \left(\alpha_m B^m + \alpha_{m-1} B^{m-1} + \alpha_{m-2} B^{m-2} + \ldots\right)$$

mit $m \in \mathbb{Z}$, $\alpha_i \in \{0, 1, \ldots, B - 1\}$, $\alpha_m \neq 0$.

Man nennt B die Basis des Zahlensystems, für die Zahlen $0, 1, \ldots, B-1$ werden spezielle Symbole, die Ziffern des Zahlensystems, verwendet.

Neben dem Dezimalsystem ($B = 10$, Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) ist im Zusammenhang mit Computern vor allem das Dualsystem ($B = 2$, Ziffern 0, 1) in Verwendung.

Am Computer wird für reelle Zahlen die normalisierte Gleitkommadarstellung verwendet:

$$x = \pm 0, \alpha_1 \alpha_2 \ldots \alpha_t \cdot B^e \tag{2.1}$$

mit $\alpha_i \in \{0, 1, \ldots, B - 1\}$ und $\alpha_1 \neq 0$. Dabei ist $B \in \mathbb{N} - \{1\}$ die Basis des verwendeten Zahlensystems. $m = 0, \alpha_1 \alpha_2 \ldots \alpha_t$ heißt die Mantisse, t die Mantissenlänge. Der Exponent e ist eine ganze Zahl zwischen vorgegebenen Schranken: $U \leq e \leq O$.

Beispiel: Für den Datentyp `float` (einfache Genauigkeit) in C wird meistens eine Zahlendarstellung mit $B = 2$, $t = 24$, $U = -125$ und $O = 128$ verwendet. Das Rechnen mit doppelter Genauigkeit (Datentyp `double`) basiert auf der Setzung $B = 2$, $t = 53$, $U = -1021$ und $O = 1024$. 24 (53) Dualstellen entsprechen etwa 7 (15) Dezimalstellen.

Das Überschreiten bzw. Unterschreiten des Exponentenbereiches wird als overflow bzw. underflow bezeichnet. Während manchmal ein underflow nicht angezeigt wird und der Computer 0 oder die kleinste positive oder negative darstellbare Zahl verwendet, führt ein overflow im Allgemeinen zum Programmabbruch. Wir werden im Weiteren das Auftreten von underflows oder overflows nicht berücksichtigen.

Die Zahl 0 und alle Zahlen der Form (2.1) bilden die Menge $\mathcal{M}$ aller Maschinenzahlen. Nachdem $\mathcal{M}$ nur endlich viele Zahlen enthält, kann natürlich nicht jede reelle Zahl am Rechner exakt dargestellt werden. Man muss runden, d.h., jede reelle Zahl x wird durch eine geeignete Maschinenzahl $rd(x)$ approximiert.

Die bestmögliche Approximation $\mathrm{rd}(x)$ einer reellen Zahl x erfüllt die Bedingung

$$|x - \mathrm{rd}(x)| \leq |x - y| \quad \text{für alle Maschinenzahlen } y. \tag{2.2}$$

Diese Forderung lässt sich leicht realisieren: Für

$$x = \pm 0, \alpha_1 \alpha_2 \ldots \alpha_t \alpha_{t+1} \ldots \cdot B^e$$

mit $\alpha_1 \neq 0$ rundet man auf, falls $\alpha_{t+1} \geq B/2$, bzw. rundet man ab, falls $\alpha_{t+1} < B/2$.

Bezeichnung: Sei $\bar{x}$ eine Approximation einer reellen Zahl x. Dann heißt

$$\Delta x = \bar{x} - x$$

absoluter Fehler und, falls $x \neq 0$,

$$\varepsilon_x = \frac{\bar{x} - x}{x}$$

relativer Fehler. Natürlich gilt:

$$\Delta x = \varepsilon_x x \quad \text{und} \quad \bar{x} = x(1 + \varepsilon_x).$$

Wir werden auch $|\bar{x} - x|$ als absoluten Fehler bzw. $|(\bar{x} - x)/x|$ als relativen Fehler bezeichnen.

Falls für den absoluten Fehler gilt:

$$|\bar{x} - x| \leq \frac{1}{2} B^s,$$

so heißt die Ziffer mit dem Stellenwert B^s gültig. Die gültigen Ziffern ohne die führenden Nullen heißen signifikante Ziffern. Die gültigen Ziffern beschreiben den absoluten Fehler, die Anzahl der signifikanten Ziffern den relativen Fehler.

Der relative Fehler ist im Allgemeinen aussagekräftiger, weil er den Fehler relativ zum exakten Wert misst. Allerdings gibt es Schwierigkeiten in der Gegend von 0.

Es lässt sich leicht zeigen, dass für das Runden nach (2.2) gilt:

$$\frac{|\,\mathrm{rd}(x) - x|}{|x|} \leq \mathrm{eps} \tag{2.3}$$

mit $\mathrm{eps} = \frac{1}{2} B^{1-t}$, der so genannten **Maschinengenauigkeit**, oder anders ausgedrückt:

$$\mathrm{rd}(x) = x(1 + \varepsilon) \quad \text{mit } |\varepsilon| \leq \mathrm{eps}.$$

Beweis. Der Einfachheit halber gelte $x > 0$. Sei $x = mB^e$ eine reelle Zahl mit normalisierter Mantisse m, also $B^{-1} \leq |m| < 1$, und sei $\mathrm{rd}(x)$ die gerundete normalisierte Gleitkommazahl mit Mantissenlänge t. Man sieht sofort, dass gilt:

$$|\,\mathrm{rd}(x) - x| \leq \frac{1}{2} B^{-t} B^e \quad \text{und} \quad |x| \geq B^{-1} B^e.$$

Also folgt durch Division:

$$\frac{|\,\mathrm{rd}(x) - x|}{|x|} \leq \frac{1}{2} B^{-t} B^e B^1 B^{-e} = \frac{1}{2} B^{1-t} = \mathrm{eps}.$$

□

Der relative Fehler, der beim Runden entsteht, ist also durch die Maschinengenauigkeit eps nach oben beschränkt.

Beispiel: Für das Rechnen mit einfacher Genauigkeit und der obigen Form des Rundens erhält man $\mathrm{eps} = 2^{-24} \approx 10^{-7}$, mit doppelter Genauigkeit $\mathrm{eps} = 2^{-53} \approx 10^{-16}$.

Eine andere Möglichkeit des Rundens ist das Abschneiden der Mantisse nach t Ziffern, d.h., man rundet immer ab. In diesem Fall gilt die Abschätzung (2.3) für die Maschinengenauigkeit $\mathrm{eps} = B^{1-t}$.

2.2 Gleitkommaarithmetik

Die Addition zweier Maschinenzahlen x, y muss nicht wieder eine Maschinenzahl ergeben. Man fordert nun, dass die auf der Maschine verfügbare „Addition“, die so genannte Gleitkommaaddition, die im Weiteren mit dem Symbol $+^*$ bezeichnet wird, so genau ist, dass gilt:

$$x +^* y = \mathrm{rd}(x + y).$$

Diese Forderung lässt sich leicht konstruktiv realisieren: Gegeben seien zwei Maschinenzahlen $x = m_1 \cdot 10^{e_1}$ und $y = m_2 \cdot 10^{e_2}$. Der Einfachheit halber sei angenommen, dass $x \geq y > 0$. (Die anderen Fälle lassen sich analog behandeln.) Die Addition lässt sich in 4 Teilschritten durchführen:

1. Exponentenangleich: $d = e_1 - e_2$
2. Mantissenverschiebung: $m_2' = m_2 \cdot 10^{-d}$, falls $d \leq t$ bzw. $m_2' = 0$ sonst.
3. Mantissenaddition: $m = m_1 + m_2'$
4. Runden

Die Genauigkeit dieser Gleitkommaaddition lässt sich folgendermaßen abschätzen (für $x + y \neq 0$):

$$\frac{|(x +^* y) - (x + y)|}{|x + y|} = \frac{|\operatorname{rd}(x + y) - (x + y)|}{|x + y|} \leq \text{eps},$$

oder anders ausgedrückt:

$$x +^* y = (x + y)(1 + \varepsilon) \quad \text{mit } |\varepsilon| \leq \text{eps}.$$

Der relative Fehler der Gleitkommaaddition ist also durch eps beschränkt.

Die Gleitkommaaddition erfüllt nicht alle Rechengesetze der „normalen“ Addition. So ist z.B. das Assoziativgesetz im Allgemeinen nicht mehr erfüllt:

$$x +^* (y +^* z) \neq (x +^* y) +^* z,$$

d.h., die Reihenfolge der Summanden beeinflusst das Resultat.

Auch alle anderen arithmetischen Operationen (Subtraktion, Multiplikation und Division) und weitere einfache binäre Operationen sind am Computer durch entsprechende Gleitkommaoperationen realisiert. Sei $\circ$ eine dieser Operationen und bezeichne $\circ^*$ die dazugehörige Gleitkommaoperation. Dann fordern wir:

$$x \circ^* y = \operatorname{rd}(x \circ y),$$

woraus wie oben für die Genauigkeit folgt:

$$\frac{|(x \circ^* y) - (x \circ y)|}{|x \circ y|} \leq \text{eps}. \tag{2.4}$$

Am Computer stehen auch einfache Funktionen $f(x)$ wie $\sin x$, $\sqrt{x}$, $\log x$, $\ldots$, zur Verfügung. Es wird im Weiteren davon ausgegangen, dass die Realisierungen $f^*(x)$ dieser Funktionen die Abschätzung

$$\frac{|f^*(x) - f(x)|}{|f(x)|} \leq \text{eps} \tag{2.5}$$

erfüllen.

Man nennt die am Rechner realisierten Operationen mit den Abschätzungen (2.4), (2.5) **elementare Operationen**.

Bemerkung: Die in diesem und in dem vorigen Abschnitt gemachten Annahmen über die Genauigkeit der Zahlendarstellung und der am Rechner implementierten Operationen sind eine Idealisierung der realen Situation, die allerdings für die Zwecke einer Fehleranalyse zumindest größenordnungsmäßig zutreffende Aussagen erlauben.

2.3 Rechengeschwindigkeit

Eine Gleitkommaoperation (z.B.: $z = x + y$, aber auch eine zusammengesetzte Operation, z.B. $z = a * x + y$) wird als ein FLOP (floating point operation) bezeichnet. Für numerische Zwecke wird die Rechengeschwindigkeit durch die Anzahl der FLOPs, die eine Rechenanlage pro Sekunde durchführt, angegeben (kurz: FLOPS, floating point operations per second). Gebräuchlichere Einheiten sind 1 MFLOPS (MegaFLOPS) = 10^6 FLOPS und 1 GFLOPS (GigaFLOPS) = 1000 MFLOPS.

Die Angabe der Anzahl der MFLOPS charakterisiert die Rechengeschwindigkeit eines skalaren Rechners sehr gut. PCs erreichen FLOP-Raten der Größenordnung 10^2 MFLOPS bis 1 GFLOPS.

Auf einem Vektorrechner werden gewisse Operationenfolgen (vektorisierbare Operationen) wesentlich schneller durchgeführt.

Beispiel: (Pipeline-Prinzip bei der Addition) Unter der vereinfachenden Annahme, dass die Addition in 4 Segmente aufgeteilt wird (siehe vorher) und der Rechner für die Durchführung jedes Segments die gleiche Zeiteinheit (einen Takt) benötigt, dauert eine Addition 4 Zeiteinheiten. Verlässt ein Operandenpaar das erste Segment, kann ein zweites Operandenpaar bereits in das erste Segment nachrücken, u.s.w. Für eine Schleife

```
for (i = 1; i <= n; i++)
  z(i) = x(i) + y(i);
```

gilt daher: Die erste Addition benötigt 4 Zeiteinheiten, dann wird in jeder weiteren Zeiteinheit eine weitere Addition fertig.

Die Beschleunigung wird allerdings nur dann voll wirksam, wenn es keine Datenabhängigkeiten der einzelnen Operationen gibt.

Man unterscheidet auf einem Vektorrechner zwischen der skalaren Leistung und der Spitzenleistung für vektorisierbare Operationen vom obigen Typ. Vektorrechner erreichen eine Spitzenleistung in der Größenordnung von GFLOPS. Je nach Anteil der vektorisierbaren Operationen (Vektorisierungsgrad) liegt die tatsächliche Rechengeschwindigkeit zwischen diesen Werten.

Ein Parallelrechner besteht aus mehreren Prozessoren. Zur Bewertung der Leistung eines Parallelrechners kommt es auf die Anzahl der Prozessoren, die Leistung der einzelnen Prozessoren und auf die Kommunikationsleistung zwischen den Prozessoren an. Die tatsächliche Rechengeschwindigkeit ergibt sich dann vor allem aus dem Anteil der parallel ausführbaren Programmteile (Parallelisierungsgrad), dem Kommunikationsaufwand und der Synchronität des Rechenablaufs.

2.4 Fehleranalyse

Man unterscheidet grob drei Fehlerarten:

- Verfahrensfehler,
- Datenfehler und
- Rundungsfehler.

Eine Behandlung von Verfahrensfehlern ist nur für jedes Verfahren individuell möglich und erfolgt daher in den einzelnen Kapiteln. Zur Illustration wird hier nur ein einfaches Beispiel eines Verfahrensfehlers diskutiert:

Beispiel: Wir betrachten das Problem der Berechnung der ersten Abnleitung einer Funktion f an einer Stelle x, also das Problem

$$y = f'(x).$$

Mit Hilfe eines zentralen Differenzenquotienten lässt sich diese Ableitung näherungsweise berechnen:

$$y_h = \frac{1}{2h}[f(x+h) - f(x-h)].$$

Damit ergibt sich ein Verfahrensfehler

$$\Delta y^V = y_h - y = \frac{1}{2h}[f(x+h) - f(x-h)] - f'(x),$$

der sich mit Hilfe einer Taylor-Reihe für kleine Schrittweiten h leicht abschätzen lässt:

$$\begin{aligned}
\Delta y^V = y_h - y &= \frac{1}{2h}[f(x+h) - f(x-h)] - f'(x) \\
&= \frac{1}{2h}\left[f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + O(h^4)\right. \\
&\qquad \left. -f(x) + f'(x)h - \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + O(h^4)\right] - f'(x) \\
&= \frac{1}{6}f'''(x)h^2 + O(h^3).
\end{aligned}$$

Im Folgenden werden die Auswirkungen von Daten- und Rundungsfehlern (Fehlerfortpflanzung) näher diskutiert.

2.4.1 Datenfehleranalyse

Eine mathematische Problemstellung lässt sich (zumindest formal) in folgende Form bringen: Die gesuchten Größen $y \in \mathbb{R}^m$ sollen aus gegebenen Größen (den Daten) $x \in \mathbb{R}^n$ über eine gegebene Vorschrift $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$ berechnet werden, kurz:

$$y = \varphi(x).$$

Wir untersuchen nun, wie sich eventuell vorhandene Störungen in den Daten auf die gesuchten Größen auswirken: Sei x der Vektor der exakten Daten und $\bar{x} = x + \Delta x$ ein Vektor von verfälschten Daten. Entsprechend bezeichnet $y = \varphi(x)$ das exakte Ergebnis und $\bar{y} = y + \Delta y = \varphi(\bar{x})$ das Ergebnis der verfälschten Daten.

Wir nehmen im Weiteren an, dass φ stetig differenzierbar ist. Dann gilt nach dem Satz von Taylor für den absoluten Fehler:

$$\Delta y_i = \varphi_i(x + \Delta x) - \varphi_i(x) \approx \sum_{j=1}^{n} \frac{\partial \varphi_i}{\partial x_j}(x)\, \Delta x_j \tag{2.6}$$

bzw. für den relativen Fehler, falls $y_i \neq 0$:

$$\varepsilon_{y_i} = \frac{\Delta y_i}{y_i} \approx \sum_{j=1}^{n} \frac{x_j}{\varphi_i(x)} \frac{\partial \varphi_i}{\partial x_j}(x)\, \varepsilon_{x_i}. \tag{2.7}$$

Die Verstärkungsfaktoren $k_{ij} = (x_j/\varphi_i(x))\partial\varphi_i/\partial x_j(x)$ heißen die Konditionszahlen des Problems.

Diese Formeln belegen zwei wichtige Erkenntnisse:

- Die Auswirkung der Störung einer einzelnen Komponente der Daten lässt sich (näherungsweise) durch Multiplikation mit der entsprechenden Ableitung bzw. der dazugehörigen Konditionszahl beschreiben.
- Der Effekt der Störung mehrerer Daten ist näherungsweise gleich der Summe der Effekte der einzelnen Störungen.

Ein Problem heißt **gut konditioniert** (oder auch: **datenstabil**), wenn kleine Fehler der Daten nur kleine Störungen der Lösung bewirken. Andernfalls heißt das Problem **schlecht konditioniert** (**dateninstabil**).

Nach der Fehlerformel (2.7) ist ein Problem dann gut (bzw. schlecht) konditioniert, wenn die Konditionszahlen des Problems klein (bzw. groß) im Vergleich mit 1 sind.

Bezeichnung: Die Fehlerformel (2.6) lässt sich in Matrix-Vektor-Schreibweise folgendermaßen kurz darstellen:

$$\Delta y \approx \varphi'(x)\, \Delta x$$

mit

$$\Delta y = \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_m \end{pmatrix},\ \Delta x = \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{pmatrix},\ \varphi'(x) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1}(x) & \frac{\partial \varphi_1}{\partial x_2}(x) & \cdots & \frac{\partial \varphi_1}{\partial x_n}(x) \\ \frac{\partial \varphi_2}{\partial x_1}(x) & \frac{\partial \varphi_2}{\partial x_2}(x) & \cdots & \frac{\partial \varphi_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varphi_m}{\partial x_1}(x) & \frac{\partial \varphi_m}{\partial x_2}(x) & \cdots & \frac{\partial \varphi_m}{\partial x_n}(x) \end{pmatrix}.$$

Beispiel: Für die Addition zweier Zahlen:

$$y = \varphi(x_1, x_2) = x_1 + x_2$$

folgt nach Formel (2.7):

$$\varepsilon_{x_1+x_2} = \frac{x_1}{x_1 + x_2}\,\varepsilon_{x_1} + \frac{x_2}{x_1 + x_2}\,\varepsilon_{x_2}.$$

Die Konditionszahlen sind besonders groß, wenn gilt: $x_1 \approx -x_2$. Diesen Effekt der großen Verstärkung von Datenfehlern bei der Addition zweier Zahlen mit $x_1 \approx -x_2$ nennt man Auslöschung.

Analog gilt für die Subtraktion:

$$\varepsilon_{x_1-x_2} = \frac{x_1}{x_1 - x_2}\,\varepsilon_{x_1} - \frac{x_2}{x_1 - x_2}\,\varepsilon_{x_2}.$$

Der kritische Fall der Auslöschung tritt hier für $x_1 \approx x_2$ ein.

In diesen Spezialfällen sind die Fehlerformeln sogar exakt. Es entsteht kein Fehler bei der Verwendung der Taylor-Formel, da höhere als erste Ableitungen von φ verschwinden.

Beispiel: Für die Multiplikation zweier Zahlen

$$y = \varphi(x_1, x_2) = x_1 \cdot x_2$$

folgt nach Formel (2.7):

$$\varepsilon_{x_1 \cdot x_2} \approx \varepsilon_{x_1} + \varepsilon_{x_2}.$$

Die Konditionszahlen sind hier 1, das Problem ist gut konditioniert.

Analog gilt für die Division:

$$\varepsilon_{x_1/x_2} \approx \varepsilon_{x_1} - \varepsilon_{x_2}.$$

Die Division zweier Zahlen ist ebenfalls ein gut konditioniertes Problem.

Beispiel: Wir betrachten das Problem der Berechnung einer Näherung der ersten Ableitung $f'(x)$ einer Funktion f an einer Stelle x mit Hilfe des zentralen Differenzenquotienten:

$$y_h = \frac{1}{2h}[f(x + h) - f(x - h)]$$

Fehler in h können vermieden werden und werden nicht weiter diskutiert.

- Zunächst werden nur die Auswirkungen von Fehlern in x für eine fixe Funktion f im interessanten Fall $h \ll x$ diskutiert:

$$y_h = \varphi(x) \equiv \frac{1}{2h}[f(x + h) - f(x - h)].$$

Somit folgt für den absoluten Datenfehler

$$\Delta y_h \approx \varphi'(x)\Delta x = \frac{1}{2h}[f'(x+h) - f'(x-h)]\Delta x \approx f''(x)\Delta x.$$

und für den relativen Datenfehler

$$\varepsilon_{y_h} \approx \frac{x\, f''(x)}{f'(x)}\, \varepsilon_x.$$

Das Problem ist also für den Fall $h \ll x$ bezüglich Störungen in x sicherlich gut konditioniert, falls $f'(x)$ von Null verschieden ist und $|x\, f''(x)|$ nicht wesentlich größer als $|f'(x)|$ ist.

- Wenn davon ausgegangen werden muss, dass es auch zu Fehlern bei der Auswertung der Funktion f kommt, ist auch die Funktion f zu den Daten zu zählen und die Auswirkungen solcher Datenstörungen sind zu diskutieren.

  Angenommen, der relative Fehler bei der Berechnung von $f(z)$ überschreitet eine Schranke ε_f nicht, d.h.:

  $$|\Delta f(z)| \leq |f(z)|\, \varepsilon_f.$$

  Will man nur die Auswirkungen von Störungen bei der Berechnung von $f_+ = f(x+h)$ und $f_- = f(x-h)$ diskutieren, lautet das Problem

  $$y_h = \varphi(f_+, f_-) = \frac{1}{2h}(f_+ - f_-)$$

  und man erhält für den dadurch verursachten absoluten Datenfehler:

  $$\Delta y_h = \frac{\partial \varphi(f_+, f_-)}{\partial f_+}\Delta f_+ + \frac{\partial \varphi(f_+, f_-)}{\partial f_-}\Delta f_- = \frac{1}{2h}[\Delta f_+ - \Delta f_-],$$

  also

  $$|\Delta y_h| \leq \frac{1}{2h}\,[|\Delta f_+| + |\Delta f_-|] \leq \frac{1}{2h}(|f(x+h)| + |f(x-h)|)\, \varepsilon_f \approx \frac{|f(x)|}{h}\, \varepsilon_f,$$

  und für den relativen Datenfehler:

  $$|\varepsilon_{y_h}| \leq \frac{|f(x+h)| + |f(x-h)|}{|f(x+h) - f(x-h)|}\, \varepsilon_f \approx \frac{|f(x)|}{|f'(x)|h}\, \varepsilon_f.$$

  Das Problem ist also für den Fall kleiner Schrittweiten h bezüglich Störungen in f im Allgemeinen schlecht konditioniert.

2.4.2 Rundungsfehleranalyse

Ein Algorithmus zur Lösung des Problems

$$y = \varphi(x)$$

lässt sich in elementare Operationen (Operationen, die am Rechner zur Verfügung stehen und eine relative Genauigkeit von eps besitzen) zerlegen:

$$x = x^{(0)} \mapsto x^{(1)} \mapsto x^{(2)} \mapsto \ldots \mapsto x^{(r)} \mapsto x^{(r+1)} = y.$$

Die Abbildung, die das Zwischenergebnis $x^{(s)}$ auf y abbildet, wird mit $\psi^{(s)}$ bezeichnet und heißt Restabbildung, wobei $s = 1, 2, \ldots, r$.

Beispiel: Ein möglicher Algorithmus zur Berechnung von

$$y = \frac{1}{2h}[f(x+h) - f(x-h)]$$

ist durch folgende Einzelschritte gegeben:

Algorithmus:

0. $x^{(0)} = x$
1. $x^{(1)} = f(x^{(0)} - h)$
2. $x^{(2)} = f(x^{(0)} + h)$
3. $x^{(3)} = x^{(2)} - x^{(1)}$
4. $x^{(4)} = x^{(3)}/(2h)$

Die dazugehörigen Restabbildungen lauten:

$$\begin{aligned} \psi^{(1)}(x^{(1)}) &= [f(x+h) - x^{(1)}]/(2h), \\ \psi^{(2)}(x^{(2)}) &= [x^{(2)} - f(x-h)]/(2h), \\ \psi^{(3)}(x^{(3)}) &= x^{(3)}/(2h). \end{aligned}$$

Sowohl bei der Dateneingabe als auch bei der Ausführung einer elementaren Operation entstehen Rundungsfehler. Dadurch erhält man anstelle von y nur eine Nährung $\bar{y}$.

Nach der Fehlerformel (2.6) verfälschen Datenfehler $\Delta x^{(0)}$ das Endergebnis näherungsweise um den Beitrag $\varphi'(x)\,\Delta x^{(0)}$.

Bei der Berechnung des Zwischenergebnisses $x^{(s)}$ für $s = 1, 2, \ldots, r$ entsteht ein neuer absoluter Fehler $\Delta x^{(s)}$, der laut (2.6) zu einer Verfälschung des Endergebnisses um näherungsweise $(\psi^{(s)})'(x^{(s)})\,\Delta x^{(s)}$ führt, wenn man annimmt, dass alle anderen Operationen exakt ausgeführt werden.

Schließlich entsteht noch ein weiterer Fehler $\Delta x^{(r+1)}$ bei der letzten elementaren Operation.

In erster Ordnung ist es gerechtfertigt, den Gesamteinfluss der einzelnen Rundungsfehler durch die Addition der oben beschriebenen Einzeleffekte zu erfassen. Somit erhält man für den gesamten Rundungsfehler Δy^R näherungsweise:

$$\Delta y^R = \bar{y} - y \approx \varphi'(x)\,\Delta x^{(0)} + \sum_{s=1}^{r} (\psi^{(s)})'(x^{(s)})\,\Delta x^{(s)} + \Delta x^{(r+1)}.$$

Der Gesamtrundungsfehler setzt sich aus zwei Anteilen zusammen, dem unvermeidbaren Fehler

$$\Delta y^0 = \varphi'(x)\,\Delta x^{(0)},$$

der sich aus der Datenfehlerfortpflanzung ergibt und der unabhängig vom gewählten Algorithmus ist, und dem restlichen Rundungsfehler

$$\Delta y^r = \sum_{s=1}^{r} (\psi^{(s)})'(x^{(s)})\,\Delta x^{(s)} + \Delta x^{(r+1)},$$

der vom gewählten Algorithmus abhängt.

Ein Algorithmus heißt **numerisch stabil**, wenn der restliche Rundungsfehler Δy^r den unvermeidbaren Fehler Δy^0 nicht dominiert.

Die Fehler $\Delta x^{(s)}$ für $s = 0, 1, \ldots, r, r+1$ lassen sich mit Hilfe der Maschinengenauigkeit abschätzen, siehe die Abschnitte über die Größe des Rundungsfehlers bei der Eingabe und bei elementaren Operationen:

$$|\Delta x^{(s)}| \leq \text{eps}\,|x^{(s)}| \quad \text{für } s = 0, 1, \ldots, r+1.$$

Daraus ergeben sich folgende Abschätzungen

$$|\Delta y^0| \leq \text{eps}\,|\varphi'(x)| \cdot |x| \quad \text{und} \quad |\Delta y^r| \leq \text{eps} \left[\sum_{s=1}^{r} |(\psi^{(s)})'(x^{(s)})|\,|x^{(s)}| + |y| \right].$$

Zur Beurteilung der numerischen Stabilität eines Algorithmus vergleicht man im Allgemeinen diese oberen Schranken für den unvermeidbaren Fehler bzw. den restlichen Rundungsfehler.

Beispiel: Für den obigen Algorithmus zur Berechnung von

$$y_h = \frac{1}{2h}\,[f(x+h) - f(x-h)]$$

erhält man im Fall $h \ll x$ für den unvermeidbaren Fehler

$$|\Delta y_h^0| \quad \lesssim \quad \text{eps}\,|f''(x)||x|$$

und für den restlichen Rundungsfehler

$$\begin{aligned} |\Delta y_h^r| &\lesssim \text{eps}\left(\frac{1}{2h}|f(x)| + \frac{1}{2h}|f(x)| + |f'(x)| + |f'(x)|\right) \\ &= \text{eps}\left(\frac{1}{h}|f(x)| + 2\,|f'(x)|\right) \approx \frac{\text{eps}}{h}\,|f(x)|. \end{aligned}$$

Falls $f(x)$ von Null verschieden ist und h hinreichend klein ist, ist der Algorithmus numerisch instabil. In diesem Fall erhält man für den gesamten Rundungsgfehler $\Delta y_h^R = \bar{y}_h - y_h$:

$$|\Delta y_h^R| \leq |\Delta y^0| + |\Delta y^r| \lesssim \text{eps}\,|f''(x)||x| + \frac{\text{eps}}{h}\,|f(x)| \approx \frac{\text{eps}}{h}\,|f(x)|.$$

Auch ohne detaillierte Fehleranalyse sieht man, dass es im letzten Schritt des Algorithmus zur Auslöschung kommt. Die Subtraktion selbst ist harmlos. Aber die an sich kleinen Rundungsfehler, die in den ersten 4 Schritten des Algorithmus entstehen, werden stark verstärkt.

Sieht man von eventuell vorhandenen zusätzlichen Datenfehlern ab, setzt sich der Gesamtfehler $\Delta y^G = \bar{y}_h - y$, der bei der Verwendung des zentralen Differenzenquotienten zur Approximation der ersten Ableitung $f'(x)$ entsteht, aus dem Verfahrensfehler $\Delta y^V = y_h - y$ und dem Rundungsfehler $\Delta y^R = \bar{y}_h - y_h$ zusammen:

$$\Delta y^G = \bar{y}_h - y = \bar{y}_h - y_h + y_h - y = \Delta y^R + \Delta y^V$$

mit

$$\Delta y^V \approx \frac{h^2}{6} f'''(x).$$

und

$$|\Delta y^R| \lesssim \frac{\text{eps}}{h}\,|f(x)|$$

zusammen. Also

$$|\Delta y^G| \lesssim \frac{h^2}{6}\,|f'''(x)| + \frac{\text{eps}}{h}\,|f(x)|.$$

Der Verfahrensfehler wird umso kleiner, je kleiner die Schrittweite h ist, der Rundungsfehler steigt hingegen mit kleiner werdender Schrittweite (Auslöschung). Die Abbildung 2.1 zeigt dieses gegenläufige Verhalten am Beispiel

$$f(x) = \sin x.$$

Für den Differenzenquotienten gilt hier

$$\frac{1}{2h}[f(x+h) - f(x-h)] = \cos x\,\frac{\sin h}{h}.$$

Der letzte Ausdruck ermöglicht für dieses Beispiel die Vermeidung der Auslöschung und damit eine (fast) rundungsfehlerfreie Auswertung des Differenzenquotienten. Wie Abbildung 2.1 zeigt, fällt der Verfahrensfehler proportional zu h^2, während der Rundungsfehler proportional zu $1/h$ steigt.

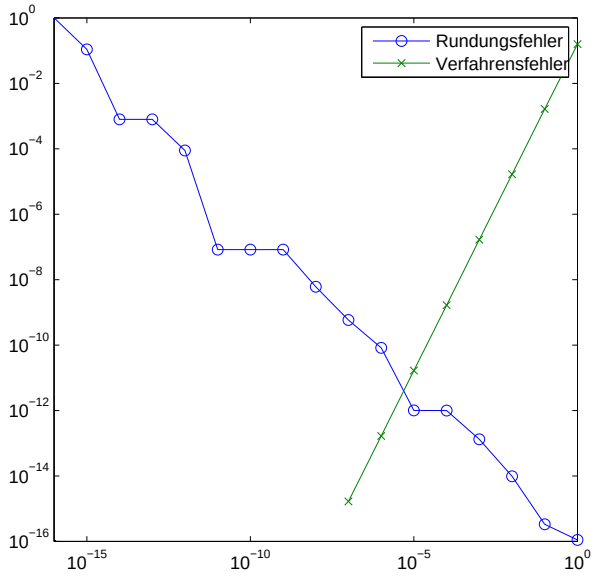


Abbildung 2.1: Numerische Differentiation: Verfahren- und Rundungsfehler

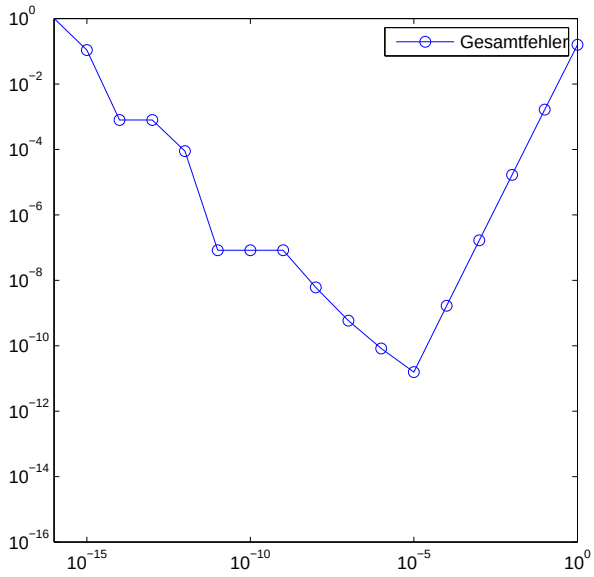


Abbildung 2.2: Numerische Differentiation: Gesamtfehler

Die optimale Schrittweite ist von der Größenordnung $\text{eps}^{1/3}$, siehe auch Abbildung 2.2.

Oft muss bei der Auswertung von f von einem wesentlich größeren relativen Fehler $\varepsilon_f \gg \text{eps}$ ausgegangen werden. Dann erhält man völlig analog die Abschätzung

$$|\Delta y^G| \lesssim \frac{h^2}{6} |f'''(x)| + \frac{\varepsilon_f}{h} |f(x)|.$$

Die optimale Schrittweite ist diesmal von der Größenordnung ${\varepsilon_f}^{1/3}$.

Kapitel 3

Direkte Verfahren zur Lösung linearer Gleichungssysteme

Lineare Gleichungssysteme entstehen z.B. bei der **Diskretisierung** von linearen partiellen Differentialgleichungen, die bei der Beschreibung zahlreicher physikalisch-technischer Probleme auftreten. Aber auch die Lösung nichtlinearer Probleme wird häufig auf die Lösung einer Folge von linearen Gleichungssystemen zurückgeführt (**Linearisierung**).

3.1 Ein Beispiel

In der Einleitung wurde das Problem der Berechnung der Temperaturverteilung in einem Stab diskutiert, das im stationären Fall auf ein Randwertproblem der folgender Art führt:

Gesucht ist eine Funktion u mit

$$\begin{aligned} -u''(x) &= f(x), \qquad x \in (0,1) \\ u(0) &= u(1) = 0, \end{aligned}$$

wobei f eine vorgegebene Funktion ist.

Durch Diskretisierung (Finite Differenzen Methode) entstand folgendes lineare Gleichungssystem:

$$-\frac{1}{h^2}\left(u_{i-1} - 2u_i + u_{i+1}\right) = f(x_i) \quad \text{für } i = 1, 2, \ldots, N$$

mit den Randbedingungen

$$u_0 = u_{N+1} = 0$$

für die Unbekannten u_i, $i = 1, 2, \ldots, N$, die als Näherungen für die exakte Lösung $u(x_i)$ in den Punkten x_i interpretiert werden können.

Man erhält also ein lineares Gleichungssystem

$$K_h \underline{u}_h = \underline{f}_h$$

mit der Matrix

$$K_h = (K_{ij})_{i,j=1,2,\ldots,N} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

und den Vektoren

$$\underline{u}_h = (u_i)_{i=1,2,\ldots,N}, \quad \underline{f}_h = (f_i)_{i=1,2,\ldots,N} \text{ mit } f_i = f(x_i).$$

3.2 Datenstabilität

Wir betrachten nun folgende allgemeine Problemstellung: Gegeben ist eine Matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ und ein Vektor $b = (b_1, b_2, \ldots, b_n)^T \in \mathbb{R}^n$. Gesucht ist ein Vektor $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ mit

$$Ax = b.$$

Im Sinne der allgemeinen Diskussion in Kapitel 2 ist das Problem, aus gegebenen Daten A und b die gesuchte Lösung x auszurechnen, rein formal durch

$$x = A^{-1}b = \varphi(A, b)$$

gegeben. Verfälschte Daten $\bar{b} = b+\Delta b$ und $\bar{A} = A+\Delta A$ führen auf ein verfälschtes Ergebnis $\bar{x} = x + \Delta x$

Die in Kapitel 2 vorgestellte Methode der Beurteilung der Datenstabilität würde es erforderlich machen, für jede der n Komponenten der Lösung und jede der $n^2 + n$ Komponenten der Daten eine Konditionszahl zu berechnen und abzuschätzen, eine völlig unübersichtliche Situation.

Besser ist es, mit Hilfe von Normen die Größe eines Vektors oder einer Matrix auf jeweils nur eine Zahl zu komprimieren.

So lässt sich statt der n komponentenweise gebildeten relativen Fehler $\varepsilon_{x_j} = (\bar{x}_j - x_j)/x_j$ die Abweichung im Ergebnis auch durch eine einzige Zahl (relativ gut) messen: $\varepsilon_x = \|\bar{x} - x\|_2/\|x\|_2$. Dabei bezeichnet $\|x\|_2$ die euklidische Länge eines Vektors x:

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Analog misst man den relativen Fehler der rechten Seite b: $\varepsilon_b = \|\bar{b} - b\|_2/\|b\|_2$.

Anstelle der euklidischen Norm kann auch eine andere Vektornorm verwendet werden.

Beispiel: Will man vor allem den komponentenweisen maximalen Fehler erfassen, bietet sich die Maximumnorm an:

$$\|x\|_\infty = \max_{i=1,2,\ldots,n} |x_i|.$$

Beispiel: Lassen sich die Komponenten eines Vektors z.B. als einzelne Massendefekte interpretieren, so ist man gelegentlich am Gesamtmassendefekt interessiert. Das führt auf die Verwendung der Betragssummennorm:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

Beispiel: Alle oben genannten Normen sind Spezialfälle der l_p-Norm

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

für alle $p \in [1, \infty)$. Der Fall $p = \infty$ ist als Grenzfall $\|x\|_\infty = \lim_{p\to\infty} \|x\|_p$ zu verstehen.

Beispiel: Die euklidische Norm eines Vektors lässt sich mit Hilfe des euklidischen Skalarprodukts darstellen:

$$\|x\|_2 = \sqrt{(x,x)_2} \quad \text{mit} \quad (x,y)_2 = \sum_i x_i y_i.$$

Für jede symmetrische und positiv definite Matrix A lässt sich durch

$$(x,y)_A = (Ax,y)_2 = \sum_{i,j} a_{ij} x_i y_j$$

ein Skalarprodukt und damit eine Norm definieren:

$$\|x\|_A = \sqrt{(x,x)_A}.$$

Alle diese Vektornormen in $\mathbb{R}^n$ erfüllen die drei für eine Norm charakteristischen Eigenschaften:

1. Definitheit: $\|v\| \geq 0$ und $\|v\| = 0$ nur, wenn $v = 0$,
2. Homogenität: $\|\lambda v\| = |\lambda| \|v\|$ für alle reellen Zahlen λ,
3. Dreiecksungleichung: $\|v + w\| \leq \|v\| + \|w\|$.

Auch die Größe von Matrizen lässt sich durch Normen (Matrixnormen) messen. Wenn Matrix- und Vektornormen gemeinsam verwendet werden, fordert man gewisse Verträglichkeitsbedingungen, vor allem soll folgende Eigenschaft gelten:

$$\|Ax\| \leq \|A\| \, \|x\| \quad \text{für alle } x \in \mathbb{R}^n.$$

Man sieht sofort, dass diese Eigenschaft für folgende Definition von $\|A\|$ erfüllt ist:

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Man nennt diese Norm die der entsprechenden Vektornorm zugeordnete Matrixnorm.

Tatsächlich stellt sich heraus, dass durch diese Definition eine Matrixnorm entsteht, dass also die für eine Norm charakteristischen Eigenschaften (Definitheit, Homogenität und Dreiecksungleichung) erfüllt sind.

Darüber hinaus gelten zusätzlich noch die folgenden Rechenregeln für jede Vektornorm und die zugeordnete Matrixnorm:

1. Die Matrixnorm ist passend zur Vektornorm, d.h.:

$$\|Ax\| \leq \|A\| \, \|x\| \quad \text{für alle } A \in \mathbb{R}^{n\times n} \text{ und alle } x \in \mathbb{R}^n.$$

2. Die Matrixnorm ist submultiplikativ, d.h.:

$$\|AB\| \leq \|A\| \, \|B\| \quad \text{für alle } A, B \in \mathbb{R}^{n\times n}.$$

3. Für die Einheitsmatrix gilt: $\|I\| = 1$.

Beispiele:

1. Die der euklidischen Norm $\|x\|_2$ zugeordnete Matrixnorm lässt sich folgendermaßen darstellen:
$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)},$$
wobei $\lambda_{\max}(B)$ den größten Eigenwert einer Matrix B bezeichnet. Sie heißt Spektralnorm.

2. Die der Maximumsnorm $\|x\|_\infty$ zugeordnete Matrixnorm lässt sich folgendermaßen darstellen:
$$\|A\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{ij}|.$$
Sie heißt Zeilenbetragssummennorm.

3. Die der Betragssummennorm $\|x\|_1$ zugeordnete Matrixnorm lässt sich folgendermaßen darstellen:
$$\|A\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{ij}|.$$
Sie heißt Spaltenbetragssummennorm.

Beispiel: Die Frobenius–Norm ist eine Matrixnorm, gegeben durch:

$$\|A\|_F = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$

Sie entspricht der euklidischen Norm, wenn man die Matrix $A \in \mathbb{R}^{n\times n}$ als Vektor in $\mathbb{R}^{n^2}$ interpretiert. Offensichtlich gilt:

$$\|I\|_F = \sqrt{n},$$

sie kann also nicht eine einer Vektornorm zugeordnete Matrixnorm sein. Sie ist aber trotzdem eine submultiplikative und zur euklidischen Norm passende Matrixnorm und wesentlich einfacher berechenbar als die Spektralnorm.

Mit Hilfe einer dieser Matrixnormen lässt sich der relative Fehler in der Matrix A durch die Größe $\varepsilon_A = \|\bar{A} - A\|/\|A\|$ messen.

Mit diesen Vorbereitungen lässt sich nun die Datenstabilität eines linearen Gleichungssystems leicht untersuchen. Zuerst wird der Spezialfall $\bar{A} = A$ betrachtet:

Satz 3.1. *Seien $A \in \mathbb{R}^{n\times n}$ eine reguläre Matrix, $b \in \mathbb{R}^n$, $\Delta b \in \mathbb{R}^n$ und $\bar{b} = b + \Delta b$. $x \in \mathbb{R}^n$ erfülle das Gleichungssystem $Ax = b$, $\bar{x} = x + \Delta x$ erfülle das Gleichungssystem $A\bar{x} = \bar{b}$. Dann gilt für jede Vektornorm und eine dazu passende Matrixnorm:*

$$\varepsilon_x \leq \kappa(A)\, \varepsilon_b$$

mit $\varepsilon_x = \|\Delta x\|/\|x\|$, $\varepsilon_b = \|\Delta b\|/\|b\|$ und $\kappa(A) = \|A\|\, \|A^{-1}\|$.

Beweis. Durch Subtraktion von $x = A^{-1}b$ und $\bar{x} = x + \Delta x = A^{-1}\,\bar{b} = A^{-1}\,(b + \Delta b)$ erhält man $\Delta x = A^{-1}\,\Delta b$. Also

$$\|\Delta x\| \leq \|A^{-1}\|\, \|\Delta b\| = \|A^{-1}\|\, \|b\|\, \frac{\|\Delta b\|}{\|b\|}.$$

Mit $\|b\| = \|A\,x\| \leq \|A\|\, \|x\|$ folgt

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

woraus nach Division mit $\|x\|$ die Behauptung folgt. □

Die Zahl $\kappa(A)$ heißt die Konditionszahl der Matrix A. Sie ist für die Größe der Auswirkung von Datenfehlern in b auf die Lösung x verantwortlich.

Berücksichtigt man auch Störungen in A, so erhält man:

Satz 3.2. *Für jede Vektornorm und jede dazu passende und submultiplikative Matrixnorm gelten folgende Aussagen:*

1. *Falls $A \in \mathbb{R}^{n\times n}$ regulär ist und $\Delta A \in \mathbb{R}^{n\times n}$ die Abschätzung $\|\Delta A\| < 1/\|A^{-1}\|$ erfüllt, dann ist auch $\bar{A} = A + \Delta A$ regulär.*

2. *Seien zusätzlich $b \in \mathbb{R}^n$, $\Delta b \in \mathbb{R}^n$ und $\bar{b} = b + \Delta b$. $x \in \mathbb{R}^n$ erfülle das Gleichungssystem $Ax = b$, $\bar{x} = x + \Delta x$ erfülle das Gleichungssystem $\bar{A}\bar{x} = \bar{b}$. Dann gilt:*

$$\varepsilon_x \leq \frac{\kappa(A)}{1 - \kappa(A)\, \varepsilon_A}\, (\varepsilon_A + \varepsilon_b)$$

mit $\varepsilon_A = \|\Delta A\|/\|A\|$.

Für kleine Fehler ε_A gilt also näherungsweise:

$$\varepsilon_x \leq \kappa(A)\,(\varepsilon_A + \varepsilon_b).$$

$\kappa(A)$ ist somit auch im allgemeinen Fall der Verstärkungsfaktor der Datenfehler.

Bemerkung: Für jede submultiplikative Matrixnorm gilt: $\|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\|$, also: $\kappa(A) \geq \|I\|$, wobei I die Einheitsmatrix bezeichnet. Für Matrizennormen, die einer Vektornorm zugeordnet sind, gilt $\|I\| = 1$ und somit $\kappa(A) \geq 1$.

Bemerkung:

1. Die Konditionszahl einer Matrix A bezüglich der Spektralnorm lässt sich mit Hilfe der so genannten Singulärwerte von A darstellen:

   Die Eigenwerten von $A^T A$ sind immer reell und größer oder gleich 0. Die nichtnegativen Quadratwurzel aus diesen Eigenwerten heißen die Singulärwerte von A und werden im Weiteren mit

   $$0 \leq \sigma_1 \leq \sigma_{\leq} \ldots \leq \sigma_n$$

   bezeichnet. Offensichtlich gilt: $\|A\|_2 = \sigma_n$.

   Im Falle einer regulären Matrix A gilt: $\sigma_1 > 0$.

   Für die Konditionszahl $\kappa(A)$ benötigt man noch die Wurzeln der Eigenwerte von $(A^{-1})^T A^{-1} = (AA^T)^{-1}$. Die Eigenwerte von $A^T A$ und AA^T stimmen überein. Somit erhält man für die Singulärwerte von A^{-1}:

   $$\frac{1}{\sigma_n} \leq \frac{1}{\sigma_{n-1}} \leq \ldots \leq \frac{1}{\sigma_1}.$$

   Also: $\|A^{-1}\|_2 = 1/\sigma_1$. Zusammenfassend folgt somit:

   $$\kappa(A) = \frac{\sigma_n}{\sigma_1}.$$

   Die Konditionszahl einer Matrix ist also gleich dem Verhältnis des größten zum kleinsten Singulärwert der Matrix.

2. Im Spezialfall $AA^T = A^T A$ (A nennt man dann eine normale Matrix, symmetrische Matrizen sind Beispiele von normalen Matrizen) sind die Singulärwerte gleich dem Betrag der Eigenwerte der Matrix A: $\sigma_i = |\lambda_i|$ mit

   $$|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|,$$

   woraus für die Konditionszahl folgt:

   $$\kappa(A) = \frac{|\lambda_n|}{|\lambda_1|}.$$

   Die Konditionszahl einer normalen Matrix ist also gleich dem Betrag des Verhältnisses des betragsgrößten zum betragskleinsten Eigenwert der Matrix.

3. Ist A symmetrisch und positiv definit, so sind alle Eigenwerte positiv und es gilt

$$\kappa(A) = \frac{\lambda_n}{\lambda_1}.$$

Beispiel: Für die Matrix K_h, die bei der diskutierten Diskretisierung entsteht, lassen sich die Eigenwerte explizit berechnen:

$$\lambda_k = \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2}\,h\right) = \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2(N+1)}\right) \quad \text{für } k = 1, 2, \ldots, N.$$

Für den kleinsten und größten Eigenwert folgt:

$$\lambda_1 = \frac{4}{h^2}\sin^2\left(\frac{\pi}{2}\,h\right) \approx \frac{4}{h^2}\left(\frac{\pi}{2}\,h\right)^2 = \pi^2$$

und

$$\lambda_N = \frac{4}{h^2}\sin^2\left(\frac{N}{N+1}\frac{\pi}{2}\right) \approx \frac{4}{h^2}.$$

Somit erhält man für die Konditionszahl:

$$\kappa(K_h) = \frac{\lambda_N}{\lambda_1} \approx \frac{4}{\pi^2}\,\frac{1}{h^2} \gg 1.$$

Dieses Ergebnis ist typisch für viele Matrizen K_h, die durch Diskretisierung von Differentialgleichungsproblemen 2. Ordnung entstehen: Die Konditionszahl ist von der Größenordnung $1/h^2$:

$$\kappa(K_h) = O\left(\frac{1}{h^2}\right).$$

Mit jeder Norm misst man die Größe eines Vektors oder einer Matrix anders. Allerdings gilt z.B.

$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\,\|x\|_\infty.$$

Ist also ein Vektor klein bezüglich der euklidischen Norm, so ist er auch klein bezüglich der Maximumnorm, und umgekehrt.

Bemerkung: Zwei Normen $\|.\|_\alpha$ und $\|.\|_\beta$ mit der Eigenschaft, dass Konstanten $c > 0$ und $C > 0$ existieren, sodass

$$c\,\|x\|_\alpha \le \|x\|_\beta \le C\,\|x\|_\alpha,$$

für alle Vektoren x gilt, heißen äquivalent. Es gilt: Alle Normen in endlichdimensionalen Räumen wie $\mathbb{R}^n$ und $\mathbb{R}^{n\times n}$ sind äquivalent. So gesehen, misst man mit jeder Norm in endlichdimensionalen Räumen etwa das gleiche. Allerdings können sich die Konstanten c und C in Abhängigkeit der Raumdimension n ändern. Für große n misst man u.U. doch wieder erheblich anders. Welche Norm wann am besten ist, hängt vom Zweck ab.

3.3 Das Gaußsche Eliminationsverfahren

Das klassische Verfahren zur Lösung eines linearen Gleichungssystems

$$Ax = b$$

oder ausführlicher

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \dots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \dots & + & a_{2n}\,x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \dots & + & a_{nn}\,x_n & = & b_n \end{array}$$

ist das Gaußsche Eliminationsverfahren.

Im Folgenden werden gelegentlich auch die Bezeichnungen $A^{(0)} = (a_{ij}^{(0)})$ für A und $b^{(0)} = (b_i^{(0)})$ für b verwendet.

Im ersten Schritt des Gaußschen Eliminationsverfahrens wird die Variable x_1 aus der 2. bis zur n-ten Gleichung eliminiert, indem ein jeweils geeignetes Vielfaches der 1. Gleichung von den restlichen Gleichungen abgezogen wird.

Dadurch entsteht ein neues Gleichungssystem mit unveränderter Lösung:

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

mit

$$A^{(1)} = \left(\begin{array}{c|ccc} u_{11} & u_{12} & \cdots & u_{1n} \\ \hline 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{array}\right), \qquad b^{(1)} = \left(\begin{array}{c} c_1 \\ \hline b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{array}\right),$$

wobei

$$\begin{aligned} u_{1j} &= a_{1j}, \quad j = 1, 2, \dots, n, \\ l_{i1} &= \frac{a_{i1}}{u_{11}}, \quad i = 2, \dots, n, \\ a_{ij}^{(1)} &= a_{ij} - l_{i1}\,u_{1j}, \quad i, j = 2, \dots, n, \end{aligned}$$

und

$$\begin{aligned} c_1 &= b_1^{(0)}, \\ b_i^{(1)} &= b_i^{(0)} - l_{i1}\,c_1, \quad i = 2, \dots, n. \end{aligned}$$

Im zweiten Schritt des Gaußschen Eliminationsverfahrens wird die Variable x_2 aus der 3. bis zur n-ten Gleichung analog eliminiert: Es entsteht das neue Gleichungssystem

$$A^{(2)}x = b^{(2)}$$

mit

$$A^{(2)} = \left(\begin{array}{cc|ccc} u_{11} & \dots & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & \cdots & u_{2n} \\ \hline \vdots & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{array}\right), \qquad b^{(2)} = \left(\begin{array}{c} c_1 \\ c_2 \\ \hline b_3^{(2)} \\ \vdots \\ b_n^{(2)} \end{array}\right),$$

wobei

$$\begin{aligned} u_{2j} &= a_{2j}^{(1)}, \quad j = 2, 3, \dots, n, \\ l_{i2} &= \frac{a_{i2}^{(1)}}{u_{22}}, \quad i = 3, \dots, n, \\ a_{ij}^{(2)} &= a_{ij}^{(1)} - l_{i2}\, u_{2j}, \quad i, j = 3, \dots, n \end{aligned}$$

und

$$\begin{aligned} c_2 &= b_2^{(1)}, \\ b_i^{(2)} &= b_i^{(1)} - l_{i2}\, c_2, \quad i = 3, \dots, n. \end{aligned}$$

Nach insgesamt $n-1$ Schritten erhält man schließlich ein Gleichungssystem der Form

$$Ux = c \tag{3.1}$$

mit

$$U = \begin{pmatrix} u_{11} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{pmatrix}, \qquad c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}.$$

U heißt rechte obere Dreiecksmatrix. Man nennt das Gleichungssystem (3.1) ein gestaffeltes System. Es lässt sich leicht durch Rückwärtseinsetzen (backward substitution) von unten nach oben auflösen:

$$\begin{aligned} x_n &= \frac{1}{u_{nn}} c_n, \\ x_i &= \frac{1}{u_{ii}} \left(c_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \quad i = n-1, n-2, \dots, 1. \end{aligned}$$

Das Gaußsche Eliminationsverfahren ist genau dann **durchführbar**, wenn

$$u_{kk} = a_{kk}^{(k-1)} \neq 0, \quad k = 1, 2, \dots, n.$$

Der **Aufwand** zur Berechnung der rechten oberen Dreiecksmatrix U beträgt ungefähr

$$(n-1)^2 + (n-2)^2 + \cdots + 2^2 + 1^2 \approx \frac{n^3}{3}$$

Operationen der Form $z = x - a \cdot y$.

Zur Berechnung von c benötigt man ungefähr

$$(n-1) + (n-2) + \cdots + 2 + 1 \approx \frac{n^2}{2}$$

Operationen.

Zur Berechnung von x durch Auflösung des gestaffelten Systems benötigt man ebenfalls ungefähr

$$1 + 2 + \cdots + (n-2) + (n-1) \approx \frac{n^2}{2}$$

Operationen.

Die **Abspeicherung** des Zwischenergebnisses nach $k-1$ Schritten lässt sich in der Form

$$\left(\begin{array}{ccc|ccc} u_{11} & \cdots & \cdots & \cdots & \cdots & u_{1n} \\ l_{21} & \ddots & & & & \vdots \\ \vdots & \ddots & u_{k-1,k-1} & \cdots & \cdots & u_{k-1,n} \\ \hline \vdots & & l_{k,k-1} & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & & \vdots & \vdots & & \vdots \\ l_{n1} & \cdots & l_{n,k-1} & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{array}\right), \qquad \left(\begin{array}{c} c_1 \\ \vdots \\ c_{k-1} \\ \hline b_k^{(k-1)} \\ \vdots \\ b_n^{(k-1)} \end{array}\right),$$

in ungefähr n^2 Speicherplätzen realisieren, falls A und b überschrieben werden dürfen.

3.4 Dreieckszerlegungen

Die einzelnen Schritte des Gaußschen Eliminationsverfahrens lassen sich auch folgendermaßen interpretieren:

Die Operationen, die im 1. Schritt auszuführen sind, sind äquivalent zu den Formeln

$$\begin{aligned} a_{1j} &= 1 \cdot u_{1j}, \quad j = 1, \ldots, n, \\ a_{i1} &= l_{i1} \cdot u_{11}, \quad i = 2, \ldots, n, \\ a_{ij} &= l_{i1} \cdot u_{1j} + a_{ij}^{(1)}, \quad i, j = 2, \ldots, n. \end{aligned}$$

Die Operationen, die im 2. Schritt auszuführen sind, sind äquivalent zu den Formeln

$$\begin{aligned} a_{2j}^{(1)} &= 1 \cdot u_{2j}, \quad j = 2, \ldots, n, \\ a_{i2}^{(1)} &= l_{i2} \cdot u_{22}, \quad i = 3, \ldots, n, \\ a_{ij}^{(1)} &= l_{i2} \cdot u_{2j} + a_{ij}^{(2)}, \quad i, j = 3, \ldots, n. \end{aligned}$$

Aus den Formeln des 1. Schrittes folgt dann:

$$\begin{aligned} a_{2j} &= l_{21} \cdot u_{1j} + 1 \cdot u_{2j}, \quad j = 2, \ldots, n, \\ a_{i2} &= l_{i1} \cdot u_{12} + l_{i2} \cdot u_{22}, \quad i = 3, \ldots, n, \\ a_{ij} &= l_{i1} \cdot u_{1j} + l_{i2} \cdot u_{2j} + a_{ij}^{(2)}, \quad i, j = 3, \ldots, n, \end{aligned}$$

u.s.w.

Man erkennt, dass

$$A = L\,U$$

gilt, wobei

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} u_{11} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{pmatrix}.$$

L heißt linke untere Dreiecksmatrix. Man spricht von einer Dreieckszerlegung von A.

Die Operationen, die im 1. Schritt für die rechte Seite auszuführen sind, sind äquivalent zu den Formeln

$$\begin{aligned} b_1 &= 1 \cdot c_1, \\ b_i &= l_{i1} \cdot c_1 + b_i^{(1)}, \quad i = 2, \ldots, n. \end{aligned}$$

Die Operationen, die im 2. Schritt für die rechte Seite auszuführen sind, sind äquivalent zu den Formeln

$$\begin{aligned} b_2^{(1)} &= 1 \cdot c_2, \\ b_i^{(1)} &= l_{i2} \cdot c_2 + b_i^{(2)}, \quad i = 3, \ldots, n. \end{aligned}$$

Aus den Formeln des 1. Schrittes folgt dann:

$$\begin{aligned} b_2 &= l_{21} \cdot c_1 + 1 \cdot c_2, \\ b_i &= l_{i1} \cdot c_1 + l_{i2} \cdot c_2 + b_i^{(2)}, \quad i = 3, \ldots, n, \end{aligned}$$

u.s.w.

Man erkennt, dass

$$Lc = b$$

gilt, wobei $c = (c_1, c_2, \ldots, c_n)^T$.

Schließlich ist im letzten Teil das System

$$Ux = c$$

zu lösen.

Zusammenfassung: In der ersten Phase wird eine Dreieckszerlegeung $A = LU$ durchgeführt. Daraus lässt sich dann leicht die Lösung des linearen Gleichungssystems $Ax = b$ bestimmen. Denn mit der Bezeichnung $c = Ux$ ist die Auflösung des Gleichungssystems

$$Ax = L\,Ux = b$$

gleichbedeutend mit der sukzessiven Auflösung der beiden Gleichungssysteme

$$Lc = b \quad \text{und} \quad Ux = c.$$

Da L und U Dreiecksmatrizen sind, nennt man diese Systeme gestaffelte Systeme. Das erste gestaffelte System löst man leicht von oben nach unten auf, das zweite gestaffelte System von unten nach oben.

Bemerkung: Nach den Überlegungen zum Aufwand des Gaußschen Eliminationsverfahrens benötigt man zur Dreieckszerlegung von A etwa $n^3/3$ Operationen, zur Auflösung der beiden gestaffelten Systeme je $n^2/2$ Operationen.

3.5 Rundungsfehleranalyse für das Gaußsche Eliminationsverfahren

Bei der Rückwärtsanalyse für das Gaußsche Eliminationsverfahren zur Lösung des linearen Gleichungssystems

$$Ax = b$$

wird versucht, die tatsächlich berechnete und durch Rundungsfehler verfälschte Näherung $\bar{x}$ als exaktes Ergebnis künstlich gestörter Daten $\bar{A}$, $\bar{b}$ darzustellen.

Beispiel: Die Lösung einer einzelnen linearen Gleichung

$$a \cdot x = b$$

erfolgt durch den Algorithmus

$$\bar{x} = b /^* a.$$

Auf Grund der Genauigkeit der Gleitkommadivision weiß man, dass

$$\bar{x} = b /^* a = (b/a)(1 + \varepsilon) \quad \text{mit } |\varepsilon| \leq \text{eps}.$$

Also

$$\bar{a} \cdot \bar{x} = b \quad \text{mit } \bar{a} = \frac{1}{1+\varepsilon} a.$$

Die Größe der Datenstörung in a lässt sich folgendermaßen abschätzen:

$$|\Delta a| = |\bar{a} - a| = |\varepsilon|\,|\bar{a}| \leq \text{eps}\,|\bar{a}| = \text{eps}\,|\bar{l}|\,|\bar{u}|$$

mit $\bar{l} = 1$ und $\bar{u} = \bar{a}$, der trivialen Dreieckszerlegung von a. Die tatsächlich berechnete Näherung lässt sich also als exaktes Ergebnis gestörter Daten interpretieren, durch die obige Abschätzung kennt man die Größenordnung der notwendigen Datenstörung in a.

Der folgende Satz enthält das Ergebnis der Rückwärtsanalyse des Rundungsfehlereinflusses für allgemeine lineare Gleichungssysteme. Wie im vorigen Beispiel wird angenommen, dass die Daten A und b selbst keine Rundungsfehler aufweisen:

Satz 3.3 (Sautter)**.** *Die mit Hilfe des Gaußschen Eliminationsverfahrens berechnete Näherung $\bar{x}$ des linearen Gleichungssystems $Ax = b$ ist exakte Lösung eines Gleichungssystems*

$$\bar{A}\bar{x} = b \quad mit \quad |\Delta A| = |\bar{A} - A| \leq \frac{2(n+1) \cdot eps}{1 - n \cdot eps} |\bar{L}|\,|\bar{U}|,$$

falls $n \cdot eps \leq 1/2$. $\bar{L}$ und $\bar{U}$ bezeichnen die tatsächlich berechneten Dreiecksmatrizen.

Bezeichnung: $|M|$ bezeichnet jene Matrix, die aus M entsteht, indem man komponentenweise den Betrag bildet. $\|M\|$, die Norm einer Matrix, ist eine Zahl.

In der Terminologie des 2. Kapitels bedeutet das, dass der restliche Rundungsfehler beim Gaußschen Eliminationsverfahren einem künstlichen Datenfehler $\varepsilon_A^{(r)} = \|\Delta A\|/\|A\|$ entspricht. Somit erhält man (in erster Ordnung) folgende Abschätzung für den restlichen Rundungsfehler $\varepsilon_x^{(r)}$:

$$\varepsilon_x^{(r)} \leq \kappa(A)\varepsilon_A^{(r)}$$

mit

$$\varepsilon_A^{(r)} = \frac{\|\Delta A\|}{\|A\|} \leq \frac{2(n+1) \cdot \mathrm{eps}}{1 - n \cdot \mathrm{eps}} \frac{\||\bar{L}| \cdot |\bar{U}|\|}{\|A\|}.$$

Zur Beurteilung der numerischen Stabilität des Gaußschen Eliminationsverfahrens ist dieser restliche Rundungsfehler mit dem unvermeidbaren Rundungsfehler $\varepsilon_x^{(0)}$, verursacht durch die Rundungsfehler $\varepsilon_A^{(0)}$ und $\varepsilon_b^{(0)}$ bei der Dateneingabe, zu vergleichen. Es gilt (in erster Ordnung)

$$\varepsilon_x^{(0)} \leq \kappa(A)(\varepsilon_A^{(0)} + \varepsilon_b^{(0)}) \leq 2\,\mathrm{eps}\,\kappa(A).$$

Durch Vergleich der Abschätzungen des unvermeidbaren Rundungsfehlers und des restlichen Rundungsfehlers erkennt man, dass das Gaußsche Eliminationsverfahren (für nicht zu große n) dann numerisch stabil ist, wenn die Komponenten von $\bar{L}$ und $\bar{U}$ im Vergleich zu den Komponenten von A nicht zu groß sind.

Durch geeignete Zeilen- und Spaltenvertauschungen lässt sich stets garantieren, dass die Komponenten von $\bar{L}$ kleiner oder gleich 1 bleiben:

Im k-ten Schritt des Gaußschen Eliminationsverfahren wird die k-te Zeile der Matrix

$$A^{(k-1)} = \left(\begin{array}{ccc|ccc} u_{11} & \cdots & \cdots & \cdots & \cdots & u_{1n} \\ 0 & \ddots & & & & \vdots \\ \vdots & \ddots & u_{k-1,k-1} & \cdots & \cdots & u_{k-1,n} \\ \hline \vdots & & 0 & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{array}\right)$$

zur Elimination der Variable x_k aus den restlichen Gleichungen verwendet. Das Element $a_{kk}^{(k-1)}$ heißt Pivotelement. Aus dem Pivotelement und den restlichen Komponenten der k-ten Spalte werden die Komponenten von L gebildet. Je kleiner das Pivotelement ist, desto größer können die Komponenten von L werden.

Es kann allerdings jedes beliebige Element $a_{ij}^{(k-1)}$ mit $i, j = k, \ldots, n$ als Pivotelement verwendet werden, wenn man vor dem nächsten Eliminationsschritt zuerst die i-te mit der k-ten Zeile und die j-te mit der k-ten Spalte vertauscht. Die Vertauschung zweier Zeilen ändert nicht die Lösung eines Gleichungssystem. Die Vertauschung zweier Spalten bedeutet nur eine Umbenennung der Unbekannten.

Man unterscheidet vor allem zwei Strategien zur Wahl des Pivotelements:

1. Totalpivotsuche: Es wird das betragsgrößte Element unter allen möglichen Elementen $a_{ij}^{(k-1)}$, $i, j = k, \ldots, n$ der Restmatrix als nächstes Pivotelement verwendet. Bezeichnet man mit $\tilde{a}_{ij}^{(k-1)}$ die Elemente der Restmatrix nach der Zeilen- und Spaltenvertauschung, so bedeutet offensichtlich diese Wahl des Pivotelements:
$$|\tilde{a}_{kk}^{(k-1)}| = \max_{i,j=k,\ldots,n} |a_{ij}^{(k-1)}|.$$

2. Spaltenpivotsuche: Es wird das betragsgrößte Element unter allen möglichen Elementen $a_{ik}^{(k-1)}$, $i = k, \ldots, n$ der k-ten Spalte als nächstes Pivotelement verwendet. Also
$$|\tilde{a}_{kk}^{(k-1)}| = \max_{i=k,\ldots,n} |a_{ik}^{(k-1)}|.$$

In beiden Fällen gilt dann
$$|l_{ik}| = \frac{|\tilde{a}_{ik}^{(k-1)}|}{|\tilde{a}_{kk}^{(k-1)}|} \leq 1.$$

Man hat also das Wachstum der Elemente von L durch Total- oder Spaltenpivotsuche unter Kontrolle.

Nicht so einfach ist die Auswirkung der Pivotsuche auf das Wachstum der Elemente von U. Für den k-ten Eliminationsschritt gilt wegen $|l_{ik}| \leq 1$ jedenfalls die Abschätzung
$$|\tilde{a}_{ij}^{(k)}| = |\tilde{a}_{ij}^{(k-1)} - l_{ik}\tilde{a}_{kj}^{(k-1)}| \leq |\tilde{a}_{ij}^{(k-1)}| + |\tilde{a}_{kj}^{(k-1)}|,$$

also
$$\max |\tilde{a}_{ij}^{(k)}| \leq 2 \max |\tilde{a}_{ij}^{(k-1)}|.$$

Es kommt also maximal zu einer Verdoppelung der Einträge pro Eliminationsschritt. Somit gilt zumindest folgende Abschätzung:
$$\max |u_{ij}| \leq 2^{n-1} \max |a_{ij}|. \tag{3.2}$$

Bei Verwendung der Totalpivotsuche vermutet man, dass
$$\max |u_{ij}| \leq n \max |a_{ij}|,$$

dass also die Elemente von U nur moderat anwachsen. Diese Behauptung konnte bisher nicht bewiesen werden. Alle praktischen Rechnungen bestätigten jedoch diese Vermutung. Daher gilt das Gaußsche Eliminationsverfahren mit Totalpivotsuche als numerisch stabil.

Bei Verwendung der Spaltenpivotsuche kann man durch konkrete Beispiele zeigen, dass die pessimistische Abschätzung (3.2) im allgemeinen Fall nicht verbessert werden kann. Das Gaußsche Eliminationsverfahren mit Spaltenpivotsuche ist also nicht immer numerisch stabil.

Für reguläre Matrizen ist das Gaußsche Eliminationsverfahren für beide Varianten der Pivotsuche stets **durchführbar**, d.h., man erhält stets ein von Null verschiedenes Pivotelement. Die Durchführung des Verfahrens ohne Pivotsuche ist für reguläre Matrizen nicht in allen Fällen gesichert.

In der Praxis begnügt man sich aus Aufwandsgründen meistens mit Spaltenpivotsuche. Es ist allerdings empfehlenswert, vor Anwendung der Spaltenpivotsuche das Problem zu skalieren.

3.6 Spezielle Gleichungssysteme

In Anwendungsproblemen treten häufig lineare Gleichungssysteme mit Matrizen spezieller Struktur auf, die eine effizientere Lösung erlauben.

Eine häufig auftretende Eigenschaft von Matrizen, die durch Diskretisierung von Differentialgleichungsproblemen entstehen, ist ihre Dünnbesetztheit, d.h., viele Einträge einer solchen Matrix sind 0. Durch geeignete Nummerierung kann man oft erreichen, dass vor allem Diagonalen, die von der Hauptdiagonale genügend weit entfernt sind, nur Nulleinträge besitzen. Solche Matrizen nennt man **Bandmatrizen**.

Ein weiterer wichtiger Fall sind lineare Gleichungssysteme mit **symmetrischen positiv definiten Matrizen**. Solche Systeme erhält man vor allem bei der Diskretisierung elliptischer Differentialgleichungsprobleme, die gewisse Symmetrieeigenschaften aufweisen. Lassen sich z.B. die Gleichungen aus einem Variationsprinzip (wie z.B. das Prinzip der virtuellen Arbeit) ableiten und werden sie geeignet diskretisiert, so erhält man Gleichungssysteme mit symmetrischen positiv definiten Matrizen.

Im Folgenden werden spezielle direkte Verfahren, das sind Verfahren, die abgesehen von Rundungsfehlern in endlich vielen Schritten die exakte Lösung eines Gleichungssystems liefern, für diese beiden Klassen von Matrizen diskutiert.

3.6.1 Dreieckszerlegungen für Bandmatrizen

Eine Bandmatrix ist eine Matrix, bei der bis auf ein Band von Diagonalen rund um die Hauptdiagonale alle Matrixelement gleich 0 sind. Genauer gesagt, unterscheidet man zwischen oberer und unterer Bandbreite:

Definition 3.1. *Eine Matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ besitzt eine untere Bandbreite p bzw. eine obere Bandbreite q, falls $a_{ij} = 0$ für alle i, j mit $i > j + p$ bzw. mit $j > i + q$.*

Eine Bandmatrix hat also folgende Gestalt:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1,q+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{p+1,1} & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & a_{n-q,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,n-p} & \cdots & a_{nn} \end{pmatrix}.$$

Beim Gaußschen Eliminationsverfahren ohne Pivotsuche überträgt sich diese Gestalt auf die Dreiecksmatrizen. L besitzt die gleiche untere Bandbreite wie A, U besitzt die gleiche obere Bandbreite wie A:

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ l_{p+1,1} & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{n,n-p} & \cdots & l_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & \cdots & u_{1,q+1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & u_{n-q,n} \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & u_{nn} \end{pmatrix}.$$

Dadurch lässt sich Aufwand sparen. Zur Durchführung des Gaußschen Eliminationsverfahrens für Matrizen mit unterer Bandbreite p und oberer Bandbreite q benötigt man ungefähr $p\,q\,n$ Operationen, falls $1 \ll p, q \ll n$.

Der Speicherbedarf zur Durchführung des Gaußschen Eliminationsverfahrens beträgt $(p + q + 1)n$ Speicherplätze.

Bemerkung: Elemente außerhalb des Bandes von insgesamt $p + q + 1$ Diagonalen rund um die Hauptdiagonale sind gleich 0 und bleiben auch während des Gaußschen Eliminationsverfahrens gleich 0. Dies gilt nicht für eventuell vorhandene Nulleinträge der Matrix A innerhalb des Bandes. Solche Nulleinträge bleiben im Allgemeinen während des Verfahrens nicht erhalten (fill in).

Matrizen mit $p = q = 1$ nennt man Tridiagonalmatrizen. Die Durchführung des Verfahrens erfordert nur $8n - 7$ Gleitkommaoperationen, der Rechenaufwand ist also proportional zu n, der Anzahl der Unbekannten. Zur Abspeicherung der Daten benötigt man $4n - 2$ Speicherplätze, die zur Durchführung des gesamten Algorithmus ausreichen, wenn die ursprünglichen Daten überschrieben werden dürfen. Der Algorithmus ist asymptotisch optimal.

3.6.2 Die Cholesky-Zerlegung für symmetrische positiv definite Matrizen

Für symmetrische positiv definite Matrizen A, also für Matrizen mit

$$A^T = A$$

und

$$x^T Ax > 0 \quad \text{für alle } x \in \mathbb{R}^n \text{ mit } x \neq 0,$$

lässt sich eine Dreieckszerlegung $A = LU$ mit $U = L^T$ durchführen:

$$A = LL^T.$$

Man spricht von einer Cholesky-Zerlegung. Im Gegensatz zum Gaußschen Eliminationsverfahren sind die Diagonalelemente von L nicht gleich 1, sondern ergeben sich im Laufe der Rechnung.

Der Aufwand des Cholesky-Verfahrens entspricht dem Aufwand des Gaußschen Eliminationsverfahrens, allerdings müssen alle Operationen aufgrund der Symmetrie nur für Indizes i, j mit $i \geq j$ durchgeführt werden. Man benötigt damit nur etwa die Hälfte der Operationen für das Gaußsche Eliminationsverfahren, also etwa $n^3/6$ Operationen.

3.7 Ergänzungen zu direkten Verfahren

Mit den bisher diskutierten Verfahren lassen sich auch weitere Problemstellungen lösen:

3.7.1 Gleichungssysteme mit mehreren rechten Seiten

Zur Lösung mehrerer Gleichungssysteme der Form

$$Ax_l = b_l, \qquad l = 1, \ldots, r$$

mit gleicher Matrix aber verschiedenen rechten Seiten, benötigt man nur einmal eine Dreieckszerlegung. Nur die gestaffelten Systeme müssen mehrmals gelöst werden. Damit ergibt sich ein Aufwand von $n^3/3 + rn^2$ Operationen.

Beispiel: Berechnung der Inversen einer regulären Matrix. Bezeichnet man die unbekannten Spaltenvektoren der Inversen A^{-1} mit x_l, so entspricht die Berechnung der Inversen der Lösung der Systeme

$$Ax_l = e_l, \qquad l = 1, \ldots, n.$$

Dabei bezeichnet e_l den l-ten natürlichen Einheitsvektor, dessen l-te Komponente gleich 1 ist, während alle anderen gleich 0 sind. Nach den obigen Aufwandsüberlegungen kostet diese Berechnung der Inversen insgesamt $n^3/3 + nn^2 = 4n^3/3$ Operationen. Eine etwas genauere Zählung ergibt tatsächlich nur n^3 Operationen.

Man könnte die Lösung eines linearen Gleichungssystems $Ax = b$ auch nach der Formel $x = A^{-1}b$ über die Berechnung der Inversen durchführen. Die obige Analyse zeigt allerdings, dass dazu wesentlich mehr Operationen benötigt werden als beim Gaußschen Eliminationsverfahren.

3.7.2 Berechnung der Determinante einer Matrix

Bei Vorliegen einer Dreieckszerlegung $A = LU$ lässt sich die Determinante von A leicht bestimmen:

$$\det A = \det L \cdot \det U = l_{11} \cdot l_{22} \cdots l_{nn} \cdot u_{11} \cdot u_{22} \cdots u_{nn}.$$

Dabei wurde verwendet, dass die Determinante einer Dreiecksmatrix gleich dem Produkt der Diagonalelemente ist.

Kapitel 4

Iterative Verfahren zur Lösung linearer Gleichungssysteme

Als Motivation für typische Gleichungssysteme, die mit iterativen Verfahren gelöst werden, wird im Folgenden die Finite Differenzen Methode für ein zweidimensionales Wärmeleitproblem näher diskutiert.

4.1 Ein Beispiel

Sei $\Omega = (0,1)\times(0,1)$ (das Einheitsquadrat in der Ebene), Γ bezeichnet den Rand der Menge Ω. Für vorgegebene Funktionen $f(x,y)$ in Ω ist eine Funktion $u(x,y)$ auf $\overline{\Omega}$ gesucht, die folgende Bedingungen erfüllt:

$$\begin{aligned} -u_{xx} - u_{yy} &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{auf } \Gamma. \end{aligned}$$

Man spricht von einem Randwertproblem. Diese Gleichungen beschreiben eine Vielzahl von physikalisch-technischen Problemstellungen, z.B. die Temperaturverteilung bei vorgegebenen Wärmequellen.

Diskretisiert man das obige Randwertproblem auf dem Gitter $\Omega_h = \{(x_i, y_j) : i,j = 1,2,\ldots,N\}$ mit $h = 1/(N+1)$, $x_i = i \cdot h$, $y_j = j \cdot h$ unter Verwendung von zentralen Differenzenquotienten für die zweiten Ableitungen

$$\begin{aligned} u_{xx} &\approx \frac{1}{h^2}\left(u_{i-1,j} - 2u_{ij} + u_{i+1,j}\right), \\ u_{yy} &\approx \frac{1}{h^2}\left(u_{i,j-1} - 2u_{ij} + u_{i,j+1}\right), \end{aligned}$$

so entsteht in jedem Gitterpunkt (x_i, y_j) eine Differenzengleichung

$$\frac{1}{h^2}(-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}) = f_{ij},$$

wobei $f_{ij} = f(x_i, y_j)$. u_{ij} bezeichnet dabei die Näherung für $u(x_i, y_j)$.

Ordnet man die Unbekannten u_{ij} und die Differenzengleichungen zeilenweise von links nach rechts und die Zeilen von unten nach oben, so entsteht unter Verwendung der Randbedingungen

$$u_{0j} = 0, \quad u_{N+1,j} = 0, \quad u_{i0} = 0, \quad u_{i,N+1} = 0$$

das lineare Gleichungssystem

$$K_h \underline{u}_h = \underline{f}_h$$

mit

$$K_h = \frac{1}{h^2} \left(\begin{array}{cccc|cccc|cccc|cccc} 4 & -1 & & & -1 & & & & & & & & & & & \\ -1 & \ddots & \ddots & & & \ddots & & & & & & & & & & \\ & \ddots & \ddots & -1 & & & \ddots & & & & & & & & & \\ & & -1 & 4 & & & & -1 & & & & & & & & \\ \hline -1 & & & & 4 & -1 & & & \ddots & & & & & & & \\ & \ddots & & & -1 & \ddots & \ddots & & & \ddots & & & & & & \\ & & \ddots & & & \ddots & \ddots & -1 & & & \ddots & & & & & \\ & & & -1 & & & -1 & 4 & & & & \ddots & & & & \\ \hline & & & & \ddots & & & & \ddots & \ddots & & & -1 & & & \\ & & & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & & \\ & & & & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & \\ & & & & & & & \ddots & & & \ddots & \ddots & & & & -1 \\ \hline & & & & & & & & -1 & & & & 4 & -1 & & \\ & & & & & & & & & \ddots & & & -1 & \ddots & \ddots & \\ & & & & & & & & & & \ddots & & & \ddots & \ddots & -1 \\ & & & & & & & & & & & -1 & & & -1 & 4 \end{array} \right)$$

und

$$\underline{u}_h = \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{N1} \\ \hline u_{12} \\ u_{22} \\ \vdots \\ u_{N2} \\ \hline \vdots \\ \vdots \\ \vdots \\ \hline u_{1N} \\ u_{2N} \\ \vdots \\ u_{NN} \end{pmatrix}, \qquad \underline{f}_h = \begin{pmatrix} f_{11} \\ f_{21} \\ \vdots \\ f_{N1} \\ \hline f_{12} \\ f_{22} \\ \vdots \\ f_{N2} \\ \hline \vdots \\ \vdots \\ \vdots \\ \hline f_{1N} \\ f_{2N} \\ \vdots \\ f_{NN} \end{pmatrix}.$$

Dabei sind die Vektoren $\underline{u}_h$ und $\underline{f}_h$ in N Blöcke von jeweils N Komponenten unterteilt. Die Matrix K_h besteht aus N^2 Teilmatrizen, die jeweils $N \times N$-Matrizen sind.

Das entstehende Gleichungssystem ist für große N sehr groß ($n = N^2$ Unbekannte und Gleichungen) und dünnbesetzt (maximal 5 Nichtnulleinträge pro Zeile). Als Bandmatrix besitzt es die untere und obere Bandbreite $p = q = N = \sqrt{n}$.

4.1.1 Typische Eigenschaften von Diskretisierungsmatrizen

Bei der Diskretisierung entstehen Matrizen mit folgenden typische Eigenschaften:

1. Die Dimension n_h der Matrix K_h ist sehr groß, da man vor allem an feinen Zerlegungen interessiert ist. Je feiner die Zerlegung ist, desto kleiner ist im Allgemeinen der Diskretisierungsfehler. Die Dimension n_h der Matrix K_h ist also (sehr) groß:
$$n_h = N^d = O(\frac{1}{h^d}).$$
2. Die meisten Einträge der Matrix sind 0, die Matrix ist dünn besetzt. Die Gesamtanzahl der Nichtnulleinträge ist typisch von der Größenordnung $O(n_h) = O(1/h^d)$.
3. Bei geeigneter Nummerierung der Knoten entsteht eine Bandmatrix mit Bandbreite
$$p_h = q_h = N^{d-1} = n_h^{\frac{d-1}{d}} = O(\frac{1}{h^{d-1}}).$$
4. Zumindest bei der konventionellen Diskretisierung von Randwertproblemen 2. Ordnung gilt für die Konditionszahl unabhängig von der Raumdimension:
$$\kappa(K_h) = O(\frac{1}{h^2}).$$

5. Im diskutierten Beispiel (aber nicht im Allgemeinen) ist die Matrix K_h symmetrisch: $K_h^T = K_h$.

6. Im diskutierten Beispiel (aber nicht im Allgemeinen) ist die Matrix K_h positiv definit: $\underline{v}_h^T K_h \underline{v}_h > 0$ für alle $\underline{v}_h \neq 0$.

Daraus ergeben sich bei Verwendung des Gaußschen Eliminationsverfahrens (für Bandmatrizen) folgende Aufwandsbetrachtungen:

Speicherbedarf:

$$n_h(p_h + q_h + 1) = O(N^d N^{d-1}) = O(N^{2d-1}) = O(n_h^{\frac{2d-1}{d}}).$$

Anzahl der Operationen:

$$p_h q_h n_h = O(N^d N^{d-1} N^{d-1}) = O(N^{3d-1}) = O(n_h^{\frac{3d-1}{d}}).$$

Also:

	$d=1$	$d=2$	$d=3$
Speicherbedarf	n_h	$n_h^{3/2}$	$n_h^{5/3}$
Anzahl der Operationen	n_h	n_h^2	$n_h^{7/3}$

Man sieht, dass der Aufwand an Speicher und Rechenzeit nur bei eindimensionalen Problemen proportional zur Anzahl der Unbekannten steigt. In diesem Fall nennt man das Verfahren (quasi-)optimal. Für zwei- oder dreidimensionale Probleme ist das Gaußsche Eliminationsverfahren nicht mehr optimal. Im Folgenden werden als Alternative zu einem direkten Verfahren, wie dem Gaußschen Eliminationsverfahren, iterative Verfahren konstruiert und deren Effizienz zur Lösung von linearen Gleichungssystemen mit großen dünnbesetzten Matrizen untersucht. Ziel ist es jedenfalls, (im obigen Sinn) optimale Verfahren zu konstruieren.

4.2 Konstruktion von Iterationsverfahren

Viele Iterationsverfahren zur Lösung eines linearen Gleichungssystems

$$Ax = b \tag{4.1}$$

beruhen auf einer Umformung von (4.1) als Fixpunktgleichung der Form

$$x = Mx + Nb. \tag{4.2}$$

Daraus gewinnt man ein Iterationsverfahren (Fixpunktiteration), indem man eine bekannte Näherung $x^{(k)}$ der exakten Lösung x^* von (4.1) in dieser Gleichung auf der rechten Seite einsetzt, um durch die linke Seite eine neue Näherung $x^{(k+1)}$ zu erhalten:

$$x^{(k+1)} = Mx^{(k)} + Nb.$$

Man startet die Iteration mit einem beliebigen Startwert $x^{(0)}$.

Beispiel: Das **Jacobi-Verfahren** beruht auf folgenden Umformungen der i-ten Zeile des Gleichungssystems (4.1):

$$\begin{aligned}\sum_{j=1}^{n} a_{ij}x_j = b_i \quad &\Longleftrightarrow \quad a_{ii}x_i + \sum_{j\neq i} a_{ij}x_j = b_i \\ &\Longleftrightarrow \quad x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_j\right),\end{aligned}$$

unter der Voraussetzung, dass $a_{ii} \neq 0$. Das Iterationsverfahren lautet also:

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

Mit der Bezeichnung

$$A = D - E - F,$$

wobei

$$D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}, \; E = -\begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \; F = -\begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$

lässt sich das Jacobi-Verfahren auch folgendermaßen darstellen:

$$Dx^{(k+1)} - (E + F)x^{(k)} = b,$$

also

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b.$$

Das Verfahren hängt nicht von der Nummerierung der Variablen ab.

Beispiel: Zum Zeitpunkt der Berechnung von $x_i^{(k+1)}$ stehen die Komponenten $x_j^{(k+1)}$ mit $j < i$ bereits zur Verfügung und könnten statt den entsprechenden Komponenten der alten Näherung verwendet werden. Diese Idee führt auf das Iterationsverfahren

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

das man **Gauß–Seidel–Verfahren** nennt. Mit den eben eingeführten Bezeichnungen lässt sich das Verfahren auch folgendermaßen darstellen

$$Dx^{(k+1)} - Ex^{(k+1)} - Fx^{(k)} = b,$$

also

$$x^{(k+1)} = (D - E)^{-1}Fx^{(k)} + (D - E)^{-1}b.$$

Das Verfahren hängt von der Nummerierung der Variablen ab.

Jacobi–Verfahren und Gauß–Seidel–Verfahren sind Beispiele für das folgende allgemeine Konstruktionsprinzip, basierend auf einer additiven Aufspaltung von A: Sei

$$A = W - R$$

mit W regulär. Dann erhält man aus $Ax = b$ das System

$$Wx - Rx = b,$$

also

$$x = W^{-1}Rx + W^{-1}b,$$

das auf das Iterationsverfahren

$$x^{(k+1)} = W^{-1}Rx^{(k)} + W^{-1}b = Mx^{(k)} + Nb \tag{4.3}$$

mit

$$M = W^{-1}R = W^{-1}(W - A) = I - W^{-1}A \quad \text{und} \quad N = W^{-1}.$$

führt.

Das Jacobi–Verfahren entspricht der Setzung $W = D$, das Gauß–Seidel–Verfahren der Setzung $W = D - E$.

Im Folgenden wird eine andere Interpretation für (4.3) entwickelt:

Angenommen, eine Näherung $x^{(k)}$ der exakten Lösung x^* ist bekannt. Man wäre in einem Schritt am Ziel, wenn man jenen Zuwachs $p^{(k,*)}$ kennen würde, für den gilt:

$$x^{(k)} + p^{(k,*)} = x^*.$$

Durch Multiplikation mit A erhält man daraus die Gleichung

$$Ax^{(k)} + Ap^{(k,*)} = Ax^* = b.$$

Also sollte man den idealen Zuwachs aus der Gleichung

$$Ap^{(k,*)} = r^{(k)} \tag{4.4}$$

mit $r^{(k)} = b - Ax^{(k)}$, dem Residuum, berechnen.

Nun möchte man allerdings bei einem iterativen Verfahren gerade die Auflösung einer Gleichung mit der Matrix A vermeiden. Daher ersetzt man in der Gleichung (4.4) die Matrix A durch eine Näherung W und erhält damit das Iterationsverfahren

$$x^{(k+1)} = x^{(k)} + p^{(k)} \quad \text{mit} \quad Wp^{(k)} = r^{(k)}. \tag{4.5}$$

Dieses Verfahren ist ident mit (4.3).

Es sind also zwei Forderungen an W zu stellen:

1. W soll A möglichst gut approximieren.

2. Gleichungssysteme der Form $Wp = r$ sollen leicht lösbar sein.

Die Setzung $W = A$ ist zwar optimal bezüglich der ersten Forderung, aber unbrauchbar wegen der zweiten Forderung. Annähernd umgekehrt ist die Situation beim Jacobi–Verfahren mit $W = D$.

In diesem Sinne lässt sich W als Approximation von A und $N = W^{-1}$ als approximative Inverse von A interpretieren.

Wird in (4.5) ein zusätzlicher Skalierungsfaktor $\tau > 0$ eingeführt, so erhält man die folgende Variante:

$$x^{(k+1)} = x^{(k)} + \tau\, p^{(k)} \quad \text{mit} \quad Wp^{(k)} = r^{(k)} \tag{4.6}$$

oder kürzer

$$x^{(k+1)} = x^{(k)} + \tau\, W^{-1}(b - Ax^{(k)}). \tag{4.7}$$

Für $\tau < 1$ spricht man von einem gedämpften Verfahren, für $\tau > 1$ von einem extrapolierten Verfahren, für $\tau = 1$ erhält man das ursprüngliche (ungedämpfte) Verfahren. Meist wird jedoch (etwas ungenau) in allen Fällen von einem gedämpften Verfahren gesprochen.

Die Durchführung des Verfahrens (4.6) kann man in folgende Schritte trennen:

1. Berechne $r^{(k)} = b - Ax^{(k)}$.

2. Löse $Wp^{(k)} = r^{(k)}$.

3. Setze $x^{(k+1)} = x^{(k)} + \tau\, p^{(k)}$.

Beispiel: Das gedämpfte Jacobi–Verfahren lässt sich folgendermaßen darstellen:

$$x^{(k+1)} = x^{(k)} + \tau\, D^{-1}(b - Ax^{(k)}).$$

Beispiel: Das einfachste (gedämpfte) Iterationsverfahren, das **Richardson-Verfahren**, entsteht für die Setzung $W = I$:

$$x^{(k+1)} = x^{(k)} + \tau\,(b - Ax^{(k)}).$$

In gewisser Weise ist es der Prototyp aller bisheriger Iterationsverfahren: Wendet man das Richardson–Verfahren nicht direkt auf das System $Ax = b$ sondern auf das äquivalente System

$$\hat{A}x = \hat{b} \quad \text{mit} \quad \hat{A} = W^{-1}A,\ \hat{b} = W^{-1}b,$$

an, so erhält man (4.7).

Anwendung auf das Beispiel aus Abschnitt 4.1

Das Jacobi-Verfahren lautet:

$$u_{i,j}^{(k+1)} = \frac{1}{4}\left(u_{i-1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)} + h^2 f_{i,j}\right)$$

Das Gauß-Seidel-Verfahren lautet:

$$u_{i,j}^{(k+1)} = \frac{1}{4}\left(u_{i-1,j}^{(k+1)} + u_{i,j-1}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)} + h^2 f_{i,j}\right)$$

Das Richardson-Verfahren lautet:

$$\begin{aligned} u_{i,j}^{(k+1)} &= u_{i,j}^{(k)} + \tau\left[f_{i,j} - \frac{1}{h^2}\left(-u_{i-1,j}^{(k)} - u_{i,j-1}^{(k)} + 4u_{i,j}^{(k)} - u_{i+1,j}^{(k)} - u_{i,j+1}^{(k)}\right)\right] \\ &= \left(1 - \frac{4\tau}{h^2}\right) u_{i,j}^{(k)} + \frac{\tau}{h^2}\left(u_{i-1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)} + h^2 f_{i,j}\right) \end{aligned}$$

Für die Parameterwahl $\tau = h^2/4$ erhält man das Jacobi-Verfahren.

4.3 Konvergenzanalyse

Die Vorteile von iterativen Verfahren gegenüber direkten Verfahren sind:

- der geringe Speicherbedarf;
- der geringe Aufwand pro Iteration;
- der unbedeutendere Rundungsfehlereinfluss.

Diesen Vorteilen steht der Nachteil

- eines Verfahrensfehlers (durch Abbruch der Iteration)

gegenüber.

Um einen kleinen Verfahrensfehler mit relativ geringem Aufwand zu erreichen, muss das Iterationsverfahren schnell konvergieren. Daher wird nun die Konvergenz von Iterationsverfahren diskutiert.

Ein Iterationsverfahren der Form (4.7) lässt sich auch folgendermaßen schreiben:

$$x^{(k+1)} = Mx^{(k)} + Nb \quad \text{mit} \quad M = I - \tau\, W^{-1}A,\ N = \tau\, W^{-1}.$$

Für die exakte Lösung x^* gilt: $Ax^* = b$, also

$$x^* = x^* + \tau\, W^{-1}\left(b - Ax^*\right) = \left(I - \tau\, W^{-1}A\right) x^* + \tau\, W^{-1}b = Mx^* + Nb.$$

Durch Subtraktion folgt für den Fehler $e^{(l)} = x^{(l)} - x^*$:

$$e^{(k+1)} = Me^{(k)}.$$

Die Matrix M, die so genannte Iterationsmatrix, steuert also das Fehlerverhalten.

Für eine beliebige Vektornorm $\|.\|$ bzw. der dazugehörigen Matrixnorm gilt nun:

Satz 4.1. *Falls $\|M\| = q < 1$, konvergiert das Iterationsverfahren für beliebige Startwerte $x^{(0)}$ und es gilt*

1. $\|e^{(k+1)}\| \leq q\,\|e^{(k)}\|$ *(q–lineare Konvergenz),*
2. $\|e^{(k)}\| \leq C\,q^k$ *(r–lineare Konvergenz).*

Beweis. Aus $e^{(k+1)} = Me^{(k)}$ folgt

$$\|e^{(k+1)}\| \leq \|M\|\,\|e^{(k)}\| = q\,\|e^{(k)}\|.$$

Durch mehrmalige Anwendung dieser Ungleichung erhält man

$$\|e^{(k)}\| \leq q\,\|e^{(k-1)}\| \leq q^2\,\|e^{(k-2)}\| \leq \cdots \leq q^k\,\|e^{(0)}\| = C\,q^k.$$

□

Bemerkung: Nach diesem Satz ist die Bedingung $\|M\| < 1$ hinreichend für die Konvergenz des Iterationsverfahrens für alle Startwerte. Notwendig und hinreichend für die Konvergenz des Iterationsverfahrens für alle Startwerte ist die (im Allgemeinen) schwächere Bedingung

$$\rho(M) < 1,$$

wobei $\rho(M) = \max\{|\lambda| : \lambda \text{ ist Eigenwert von } M\}$ den Spektralradius von M bezeichnet. Unter dieser schwächeren Bedingung lässt sich nur die r–lineare Konvergenz zeigen.

Aufwand eines Iterationsverfahrens:

Wie viele Iterationen werden nun benötigt, um einen Anfangsfehler um einen Faktor ε zu verkleinern? Gesucht ist also k mit

$$\|e^{(k)}\| \leq \varepsilon\|e^{(0)}\|.$$

Wegen $\|e^{(k)}\| \leq q^k\|e^{(0)}\|$, ist diese Bedingung sicher erfüllt, wenn k so groß ist, dass

$$q^k \leq \varepsilon.$$

Das ist gleichbedeutend mit der Forderung

$$\ln q^k \leq \ln \varepsilon,$$

also

$$k \ln q \leq \ln \varepsilon,$$

und somit

$$k \geq \frac{\ln \varepsilon}{\ln q} = \frac{-\ln \varepsilon}{-\ln q}.$$

Man benötigt also rund

$$k = \frac{-\ln \varepsilon}{-\ln q}$$

Iterationen.

Anwendung auf das Richardson-Verfahren für symmetrische Matrizen A

Voraussetzung: Sei A eine symmetrische Matrix, d.h.:

$$(Ax, y) = (Ay, x) \quad \text{für alle } x, y \in \mathbb{R}^n,$$

wobei (x, y) ein Skalarprodukt in $\mathbb{R}^n$ bezeichnet (z.B. das euklidische Skalarprodukt $(x, y)_2 = y^T x$).

Das Richardson-Verfahren lautet:

$$x^{(k+1)} = (I - \tau A)x^{(k)} + \tau b$$

mit $\tau > 0$, also

$$x^{(k+1)} = Mx^{(k)} + \tau b$$

mit der Iterationsmatrix

$$M = I - \tau A.$$

Da M symmetrisch ist, gilt

$$\|M\| = \sup_{0 \neq x \in \mathbb{R}^n} \frac{\|Mx\|}{\|x\|} = \sup_{0 \neq x \in \mathbb{R}^n} \frac{|(Mx, x)|}{(x, x)} = \rho(M),$$

wobei $\rho(M)$ den sogenannten Spektralradius von M bezeichnet:

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

Im Speziellen gilt:

$$\|M\| = \rho(M) = \rho(I - \tau\ A) \ = \ \max\{|1 - \tau\, \lambda| \ : \ \lambda \text{ ist Eigenwert von } A\}.$$

Um die Bedingung $\|M\| < 1$ zu erfüllen, ist es offensichtlich notwendig, dass alle Eigenwerte von A positiv sind, dass also A positiv definit ist:

$$(Ax, x) > 0 \quad \text{für alle } x \in \mathbb{R}^n \text{ mit } x \neq 0.$$

Wie man leicht sieht, gilt für symmetrische und positiv definite Matrizen A

$$\|M\| = \max\{|1 - \tau\, \lambda_{\min}(A)|, |1 - \tau\, \lambda_{\max}(A)|) = q(\tau) = \begin{cases} 1 - \tau\, \lambda_{\min}(A) & \text{falls } 0 \leq \tau < \tau_* \\ \tau\, \lambda_{\max}(A) - 1 & \text{falls } \tau \geq \tau_* \end{cases}$$

mit

$$\tau_* = \frac{2}{\lambda_{\min}(A) + \lambda_{\max}(A)}.$$

Die Bedingung $\|M\| < 1$ ist genau dann erfüllt, wenn

$$0 < \tau < \frac{2}{\lambda_{\max}(A)}.$$

Das Richardson-Verfahren konvergiert also, wenn der Dämpfungsparameter τ hinreichend klein gewählt wird. Um sicher zu sein, wie klein man den Dämpfungsparameter wählen muss, sollte man (eine gute obere Schranke für) den größten Eigenwert von A kennen.

Zu kleine Parameter verschlechtern allerdings die Konvergenz. Eine in einem gewissen Sinn optimale Wahl für den Parameter τ erhält man durch die Forderung, dass $q(\tau)$ möglichst klein sein soll. Das ist offensichtlich für $\tau = \tau_*$ der Fall. Für diese Wahl erhält man für den dazugehörigen Konvergenzfaktor q_*:

$$\begin{aligned} q_* &= q(\tau_*) = 1 - \tau_* \lambda_{\min}(A) = 1 - \frac{2\lambda_{\min}(A)}{\lambda_{\min}(A) + \lambda_{\max}(A)} \\ &= \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} = \frac{\lambda_{\max}(A)/\lambda_{\min}(A) - 1}{\lambda_{\max}(A)/\lambda_{\min}(A) + 1} = \frac{\kappa(A) - 1}{\kappa(A) + 1}. \end{aligned}$$

Für die Bestimmung des optimalen Dämpfungsparameters sollte man also (gute Näherungen für) den kleinsten und den größten Eigenwert von A kennen.

Es gilt

$$\frac{1}{-\ln q_*} = \frac{1}{-\ln\left[1 - \frac{2}{\kappa(A)+1}\right]} \leq \frac{1}{\frac{2}{\kappa(A)+1}} = \frac{\kappa(A)+1}{2}$$

für große Konditionszahlen $\kappa(A)$. Daraus erhält man für die Anzahl der benötigten Iterationen

$$k \leq \frac{-\ln \varepsilon}{2}(\kappa(A) + 1) = O(\kappa(A)).$$

Die Gesamtzahl der Operationen für dünnbesetzte Matrizen mit $O(n)$ Nichtnulleinträgen ist dann von der Größenordnung $O(\kappa(A)\, n)$.

Beispiel: Für die diskutierten FD-Diskretisierungen gilt: K_h ist symmetrisch und positiv definit bezüglich des euklidischen Skalarprodukts und es gilt

$$\kappa(K_h) = O(\frac{1}{h^2}) \quad \text{und} \quad n_h = O(\frac{1}{h^d}).$$

Nach der obigen Analyse ist daher die Gesamtzahl der Operationen von der Ordnung $O(\kappa(K_h)\, n_h) = O(1/h^{d+2}) = O(n_h^{1+2/d})$. Für $d = 2$ benötigt man also $O(n_h^2)$ Operationen, das ist die gleiche Größenordnung des Aufwands wie beim Gaußschen Eliminationsverfahren. Für $d = 3$ ergeben sich $O(n_h^{5/3})$ Operationen, also (asymptotisch) weniger Operationen als beim Gaußschen Eliminationsverfahren.

Bisher wurde die Analyse nur für das Euklische Skalarprodukt angewendet. Dann ist allerdings das Abbruchkriterium

$$\|e^{(k)}\|_2 \leq \varepsilon \, \|e^{(0)}\|_2$$

nicht konstruktiv, da die euklidische Norm des Fehlers ohne Kenntnis der exakten Lösung nicht berechenbar ist.

Angenommen A is symmetrisch und positiv definit bezüglich des euklidischen Skalarprodukts. Ein anderes Skalarprodukt, das zu den gleichen Aussagen führt, ist das Energie-Skalarprodukt $(x,y)_A$ mit der dazugehörigen Norm $\|.\|_A$, der Energie-Norm. Man beachte, dass Matrix $M = I - \tau A$ auch bezüglich des Skalarprodukts $(x,y)_A$ symmetrisch ist:

$$(Mx,y)_A = (AMx,y)_2 = (A(I-\tau A)x,y)_2 = (A(I-\tau A)y,x)_2 = (AMy,x)_2 = (My,x)_A.$$

Daher gelten alle bisherigen Aussagen auch in dieser Norm. Aussagen über die Konvergenz des Richardson–Verfahren in der Energienorm werden im nächsten Abschnitt benötigt. Diese Norm liefert ebenfalls kein konstruktives Abbruchkriterium.

Eine weitere interessante Norm ist die Residuumsnorm $\|.\|_{A^TA}$, die für jede reguläre Matrix A eingeführt werden kann. Der Fehler in dieser Norm ist ohne Kenntnis der exakten Lösung berechenbar:

$$\|e^k\|^2_{A^TA} = (A^TAe^k, e^k)_2 = (Ae^k, Ae^k)_2 = (Ax^k - b, Ax^k - b)_2 = (r^k, r^k)_2 = \|r^k\|_2^2.$$

Für den Fall A symmetrisch und positiv definit stimmt die Residuumsnorm natürlich mit $\|.\|_{A^2}$ überein. Wie oben lässt sich leicht zeigen, dass M auch bezüglich des Skalarprodukts $(x,y)_{A^2}$ symmetrisch ist, woraus wieder alle früher gezeigten Aussagen folgen.

Die drei diskutierten Normen für den Fall A symmetrisch und positiv definit sind die Spezialfälle $s = 0$, $s = 1$ und $s = 2$ aus der Familie von Normen $\|.\|_{(s)} = \|.\|_{A^s}$, für die gleichen Konvergenzaussagen folgen.

Die Aussagen über die Konvergenz des Richardson-Verfahrens lassen sich auf allgemeinere präkonditionierte (und gedämpfte) Richardson-Verfahren der Form

$$x^{(k+1)} = (I - \tau W^{-1}A)x^{(k)} + \tau W^{-1}b$$

mit $\tau > 0$ übertragen:

Satz 4.2. *Seien A und W symmetrische und positiv definite Matrizen bezüglich des euklidischen Skalarprodukts. Dann gilt:*

1. *Das präkonditionierte Richardson-Verfahren konvergiert, falls*

$$0 < \tau < \frac{2}{\lambda_{max}(W^{-1}A)}.$$

2. *Für die (optimale) Parameterwahl*

$$\tau_* = \frac{2}{\lambda_{min}(W^{-1}A) + \lambda_{max}(W^{-1}A)}$$

gelten die Fehlerabschätzungen

$$\|e^{(k+1)}\|_{(s)} \le q_* \|e^{(k)}\|_{(s)} \quad \textit{und} \quad \|e^{(k)}\|_{(s)} \le q_*^k \|e^{(0)}\|_{(s)}$$

mit

$$q_* = \frac{\kappa(W^{-1}A) - 1}{\kappa(W^{-1}A) + 1}$$

und den folgende Normen für $s = 0, 1, 2$*:*

$$\|e\|_{(0)} = (We, e)_2^{1/2}, \quad \|e\|_{(1)} = (Ae, e)_2^{1/2}, \quad \|e\|_{(2)} = (W^{-1}Ae, Ae)_2^{1/2}.$$

3. *Für den Fehler* $e^{(k)} = x^{(k)} - x^*$ *gilt:*

$$\|e^{(k)}\|_{(2)}^2 = (W^{-1}r^{(k)}, r^{(k)})_2 = (p^{(k)}, r^{(k)})_2.$$

Beweis. Es gilt: M ist symmetrisch in den entsprechenden Skalarprodukten, z.B.:

$$\begin{aligned}(Mx, y)_W &= (WMx, y)_2 = (W(I - \tau W^{-1}A)x, y)_2 = ((W - \tau A)x, y)_2 \\ &= (W(I - \tau W^{-1}A)y, x)_2 = ((W - \tau A)y, x)_2 = (WMy, x)_2 = (My, x)_W.\end{aligned}$$

Alles Weitere folgt aus der Analyse von vorhin. □

Nach diesem Satz ist es also zweckmäßig, die Matrix W so zu wählen, dass die Konditionszahl von $W^{-1}A$ möglichst klein ist. Man nennt diese Strategie Präkonditionierung und nennt in diesem Zusammenhang W eine Präkonditionierungsmatrix oder einen Präkonditionierer.

Der Idealfall $W = A$ führt auf die kleinstmögliche Konditionszahl 1, ist aber unrealistisch, da das Verfahren die Lösung einer Gleichung der Form $Wp = r$ erfordert. Die bisherigen klassischen Iterationsverfahren, auf die der letzte Satz anwendbar ist (Jacobi-Verfahren, Gauß-Seidel-Verfahren) führen zu keiner wesentlichen Verkleinerung der Konditionszahl von $W^{-1}A$.

Gute Präkonditioner kann man z.B. durch additive Aufspaltungen der Form

$$A = LU - R$$

gewinnen, wobei L eine linke untere Dreiecksmatrix und U eine rechte obere Dreiecksmatrix ist. Falls $R = 0$ gewählt wird, spricht man von vollständiger Dreieckszerlegung (Gaußsche Elimination), die Berechnung von L und U und die Lösung von $Wp = r$ sind in diesem Fall allerdings zu aufwendig. Es gibt Bedingungen an R, die eine wesentlich effizientere Zerlegung erlauben.

Es gibt Verallgemeinerungen des Jacobi– und des Gauß–Seidel–Verfahrens, so genannte additive und multiplikative Schwarz-Methoden, die zu einer erheblichen Verkleinerung der Konditionszahl führen können.

4.4 Das Gradientenverfahren und das cg-Verfahren

Der für Anwendungsprobleme wichtige Spezialfall eines Gleichungssystems

$$Ax = b \tag{4.8}$$

mit einer symmetrischen und positiv definiten Matrix A ermöglicht die Durchführung eines besonders schnellen Iterationsverfahrens.

Für symmetrische und positiv definite Matrizen ist das Problem (4.8) zum Minimierungsproblem

$$J(x) = \frac{1}{2}(Ax, x)_2 - (b, x)_2 \longrightarrow \min!$$

äquivalent, da Gradient und Hessematrix von J in einem Punkt x durch

$$\nabla J(x) = Ax - b, \quad \nabla^2 J(x) = A$$

gegeben sind.

Die Richtung des steilsten Abstiegs von J in einem Punkt x ist durch $-\nabla J(x) = b - Ax = r$, also dem Residuum gegeben.

Ausgehend von einer Näherung $x^{(0)}$ ist es naheliegend, zuerst in dieser Richtung nach der nächsten Näherung zu suchen:

$$x^{(1)} = x^{(0)} + \alpha^{(0)} r^{(0)},$$

wobei die Zahl $\alpha^{(0)}$ so gewählt wird, dass $J(x^{(1)})$ minimal wird.

Für eine allgemeine Suchrichtung $p^{(0)}$ gilt:

$$\begin{aligned} J(x^{(0)} + \alpha\, p^{(0)}) &= \frac{1}{2}(A(x^{(0)} + \alpha\, p^{(0)}), x^{(0)} + \alpha\, p^{(0)}) - (b, x^{(0)} + \alpha\, p^{(0)})_2 \\ &= \frac{1}{2}(Ax^{(0)}, x^{(0)})_2 - (b, x^{(0)})_2 + \frac{\alpha}{2}\,(Ap^{(0)}, x^{(0)})_2 + \frac{\alpha}{2}\,(Ax^{(0)}, p^{(0)})_2 \\ &\quad - \alpha\,(b, p^{(0)})_2 + \frac{\alpha^2}{2}\,(Ap^{(0)}, p^{(0)})_2 \\ &= \frac{\alpha^2}{2}\,(Ap^{(0)}, p^{(0)})_2 - \alpha\,(r^{(0)}, p^{(0)})_2 + \frac{1}{2}(Ax^{(0)}, x^{(0)})_2 - (b, x^{(0)})_2. \end{aligned}$$

$J(x^{(0)} + \alpha p^{(0)})$ ist genau dann minimal, wenn

$$\frac{d}{d\alpha} J(x^{(0)} + \alpha\, p^{(0)})_2 \Big|_{\alpha=\alpha^{(0)}} = 0,$$

d.h.:

$$\alpha^{(0)}\,(Ap^{(0)}, p^{(0)})_2 - (r^{(0)}, p^{(0)})_2 = 0. \tag{4.9}$$

Das führt auf die Setzung

$$\alpha^{(0)} = \frac{(r^{(0)}, p^{(0)})_2}{(Ap^{(0)}, p^{(0)})_2}.$$

und man erhält die nächste Näherung

$$x^{(1)} = x^{(0)} + \alpha^{(0)} r^{(0)}.$$

Die Richtung des steilsten Abstiegs von J im Punkt $x^{(1)}$ lässt sich leicht berechnen:

$$r^{(1)} = b - Ax^{(1)} = b - Ax^{(0)} - \alpha^{(0)} Ar^{(0)} = r^{(0)} - \alpha^{(0)} Ar^{(0)}.$$

Setzt man diese Strategie fort, erhält man das so genannte Gradientenverfahren:

Für einen gegebenen Startwert $x^{(0)}$ setzt man $r^{(0)} = b - Ax^{(0)}$, dann führt man für $k = 0, 1, \ldots$ die folgende Iteration durch:

$$\begin{aligned} p^{(k)} &= r^{(k)} \\ x^{(k+1)} &= x^{(k)} + \alpha^{(k)} p^{(k)} \quad \text{mit} \quad \alpha^{(k)} = \frac{(r^{(k)}, p^{(k)})}{(Ap^{(k)}, p^{(k)})}, \\ r^{(k+1)} &= r^{(k)} - \alpha^{(k)} Ar^{(k)}. \end{aligned}$$

Konvergenzeigenschaften des Gradientenverfahrens

Ein Schritt des Gradientenverfahrens entspricht einem Schritt des Richardson-Verfahrens mit der Parameterwahl $\tau = \alpha^{(k)}$. Die Konvergenz lässt sich besonders einfach mit Hilfe der Norm $\|x\|_A = \sqrt{(Ax, x)_2}$ untersuchen: Es gilt folgender Zusammenhang zwischen dem Fehler in der Energienorm und dem Energiefunktional:

$$\begin{aligned} \|x - x^*\|_A^2 &= (A(x - x^*), x - x^*)_2 \\ &= (Ax, x)_2 - (Ax, x^*)_2 - (Ax^*, x)_2 + (Ax^*, x^*)_2 \\ &= (Ax, x)_2 - 2\,(Ax^*, x)_2 + (Ax^*, x^*)_2 \\ &= [(Ax, x)_2 - 2\,(b, x)_2] - [(Ax^*, x^*)_2 - 2\,(b, x^*)_2] = 2\,[J(x) - J(x^*)], \end{aligned}$$

Daraus folgt:

$$\begin{aligned} \|x^{(1)} - x^*\|_A^2 &= 2\,[J(x^{(1)}) - J(x^*)] \le 2\,[J(x^{(0)} + \tau_* r^{(0)}) - J(x^*)] \\ &= \|(x^{(0)} + \tau_* r^{(0)}) - x^*\|_A^2 \\ &\le q_*^2 \, \|x^{(0)} - x^*\|_A^2. \end{aligned}$$

Man beachte, dass $x^{(0)} + \tau_* r^{(0)}$ der nächste Iterationspunkt des Richardson-Verfahrens mit optimaler Parameterwahl ist. Die letzte Abschätzung folgt dann aus Satz 4.2.

Ein Schritt des Gradientenverfahrens ist also nicht schlechter als ein Schritt des Richardson-Verfahrens mit optimaler Parameterwahl. Während man allerdings für die Durchführung des Richardson-Verfahrens mit optimaler Parameterwahl den kleinsten und größten Eigenwert (oder zumindest sehr gute Näherungen dafür) kennen muss, ist die Durchführung des Gradientenverfahrens ohne Kenntnis von Eigenwerten möglich.

Damit ist folgender Satz bewiesen:

Satz 4.3. *Sei A eine symmetrische und positiv definite Matrix mit kleinstem Eigenwert $\lambda_{min}(A)$ und größtem Eigenwert $\lambda_{max}(A)$. Dann konvergiert das Gradientenverfahren und es gelten die Fehlerabschätzungen*

$$\|x^{(k+1)} - x^*\|_A \leq q \, \|x^{(k)} - x^*\|_A \quad \textit{und} \quad \|x^{(k)} - x^*\|_A \leq q^k \, \|x^{(0)} - x^*\|_A$$

mit

$$q = \frac{\lambda_{max}(A) - \lambda_{min}(A)}{\lambda_{max}(A) + \lambda_{min}(A)} = \frac{\kappa(A) - 1}{\kappa(A) + 1}.$$

Daraus ergibt sich für die Anzahl der Iterationen bei großer Konditionszahl die gleiche Abschätzung wie beim Richardson-Verfahren mit optimaler Parameterwahl:

$$k \approx \frac{-\ln \varepsilon}{-\ln q} \approx (-\ln \varepsilon) \frac{\kappa(A) + 1}{2} = O(\kappa(A))$$

Aus (4.9) folgt leicht, dass zwei aufeinanderfolgende Suchrichtungen des Gradientenverfahrens orthogonal sind: Für $p^{(k+1)} = r^{(k+1)}$ gilt

$$(p^{(k+1)}, p^{(k)})_2 = 0.$$

Der erste Schritt des cg-Verfahrens stimmt mit dem ersten Schritt des Gradientenverfahrens überein:

$$x^{(1)} = x^{(0)} + \alpha^{(0)} p^{(0)}$$

mit $p^{(0)} = r^{(0)}$. Wie vorhin erhält man für das nächste Residuum:

$$r^{(1)} = r^{(0)} - \alpha^{(0)} \, A p^{(0)}.$$

Die beste Wahl für die nächste Suchrichtung wäre $p^{(1,*)} = x^* - x^{(1)}$, die natürlich nicht verfügbar ist. Es gilt:

$$(A p^{(1,*)}, p^{(0)})_2 = (r^{(1)}, p^{(0)})_2 = 0.$$

Diese Eigenschaft motiviert die folgende Bedingung, die für die tatsächliche nächste Suchrichtung $p^{(1)}$ gefordert wird:

$$(A p^{(1)}, p^{(0)}) = 0. \tag{4.10}$$

Zwei Richtungen, die (4.10) erfüllen, heißen konjugierte Richtungen oder A–orthogonal.

Das Residuum $r^{(1)}$ ist orthogonal, aber im Allgemeinen nicht A–orthogonal zu $p^{(0)}$. Aber die Modifikation

$$p^{(1)} = r^{(1)} + \beta^{(0)} \, p^{(0)}$$

erfüllt (4.10), falls

$$(A(r^{(1)} + \beta^{(0)} \, p^{(0)}), p^{(0)})_2 = 0,$$

also

$$\beta^{(0)} = -\frac{(r^{(1)}, A p^{(0)})_2}{(A p^{(0)}, p^{(0)})_2}.$$

Setzt man diese Strategie analog fort, so erhält man das so genannte cg–Verfahren:

Mit der Anfangssetzung $r^{(0)} = b - Ax^{(0)}$ lautet der allgemeine Schritt des cg-Verfahrens für $k = 0, 1, 2 \ldots$:

$$\begin{aligned} p^{(k)} &= \begin{cases} r^{(0)} & \text{für } k = 0 \\ r^{(k)} + \beta^{(k-1)} p^{(k-1)} \quad \text{mit} \quad \beta^{(k-1)} = -\dfrac{(r^{(k)}, Ap^{(k-1)})}{(Ap^{(k-1)}, p^{(k-1)})} & \text{für } k \geq 1 \end{cases} \\ x^{(k+1)} &= x^{(k)} + \alpha^{(k)} p^{(k)} \quad \text{mit} \quad \alpha^{(k)} = \frac{(r^{(k)}, p^{(k)})}{(Ap^{(k)}, p^{(k)})}, \\ r^{(k+1)} &= r^{(k)} - \alpha^{(k)} Ap^{(k)}. \end{aligned}$$

Konvergenzeigenschaften des cg-Verfahrens:

Es lässt sich leicht zeigen, dass sowohl für das Gradienten-Verfahren als auch für das cg-Verfahren gilt:

$$x^{(k)} \in x^{(0)} + K_k(A, r^{(0)}),$$

wobei $K_k(M, v)$ den sogenannten k-ten Krylov-Unterraum bezeichnet:

$$K_k(M.v) = \text{span}(v, Mv, M^2 v, \ldots, M^{k-1} v).$$

Für das cg-Verfahren lässt sich zeigen, dass $x^{(k)}$ die beste Wahl in $x^{(0)} + K_k(A, r^{(0)})$ ist:

$$J(x^{(k)}) = \min_{y \in x^{(0)} + K_k(A, r^{(0)})} J(y).$$

(Das cg-Verfahren bezeichnet man auch als ein Krylov-Unterraum-Verfahren.) Daraus folgt einerseits: Bei exakter Rechnung führt das cg-Verfahren nach spätestens n Schritten zur exakten Lösung des Gleichungssystems. In diesem Sinne ist es ein direktes Verfahren.

Wichtiger allerdings für die Lösung großer Gleichungssysteme mit dünnbesetzten Matrizen ist das cg-Verfahren als Iterationsverfahren, da es bereits nach wesentlich weniger als n Schritten eine gute Näherung der exakten Lösung liefern kann. Es gilt nämlich:

Satz 4.4. *Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische, positiv definite Matrix. Dann konvergiert das cg-Verfahren und es gilt die Fehlerabschätzung*

$$\|x^{(k)} - x^*\|_A \leq 2 \frac{q^k}{1 + q^{2k}} \|x^{(0)} - x^*\|_A \leq 2\, q^k \, \|x^{(0)} - x^*\|_A$$

mit

$$q = \frac{\sqrt{\lambda_{max}(A)} - \sqrt{\lambda_{min}(A)}}{\sqrt{\lambda_{max}(A)} + \sqrt{\lambda_{min}(A)}} = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}.$$

Um den Anfangsfehler $\|x^{(0)} - x^*\|_A$ um einen Faktor ε zu reduzieren, muss nach diesem Satz gelten:

$$2\, q^k \leq \varepsilon.$$

Also genügen für große Konditionszahlen

$$\begin{aligned} k &\approx \frac{-\ln(\varepsilon/2)}{-\ln q} = \frac{-\ln(\varepsilon/2)}{-\ln\left[1-\frac{2}{\sqrt{\kappa(A)}+1}\right]} \approx \frac{-\ln(\varepsilon/2)}{\frac{2}{\sqrt{\kappa(A)}+1}} = (-\ln(\varepsilon/2))\frac{\sqrt{\kappa(A)}+1}{2} \\ &= O(\sqrt{\kappa(A)}) \end{aligned}$$

Iterationen.

Beispiel: Für die diskutierten FD-Diskretisierungen erhält man wegen

$$\kappa(K_h) = O(\frac{1}{h^2}) \quad \text{und} \quad n_h = O(\frac{1}{h^d})$$

eine Gesamtzahl von $O(\sqrt{\kappa(K_h)}\, n_h) = O(1/h^{d+1}) = O(n_h^{(1+1/d)})$ Operationen. Für $d = 2$ benötigt man $O(n_h^{3/2})$ Operationen. Für $d = 3$ ergeben sich $O(n_h^{4/3})$ Operationen, also in beiden Fällen (asymptotisch) weniger Operationen als beim Gaußschen Eliminationsverfahren.

Bemerkung: Das cg-Verfahren kann noch weiter verbessert werden, wenn man es nicht auf das ursprüngliche Gleichungssystem $Ax = b$ sondern auf das vorkonditioniertes Problem

$$W^{-1}Ax = W^{-1}b$$

anwendet, wobei die symmetrische und positiv definite Matrix W so zu wählen ist, dass $\kappa(W^{-1}A)$ möglichst klein ist.

Bemerkung: Es gibt Varianten des cg-Verfahrens auch für symmetrische, nicht positiv definite Matrizen und für nichtsymmetrische Matrizen. Allgemein spricht man von Krylov-Unterraum Methoden.

Bemerkung: Zu den effizientesten Verfahren für diskretisierte Gleichungen gehören die so genannten Mehrgitterverfahren, für die sich für viele Anwendungen zeigen lässt, dass sie einen Konvergenzfaktor $q < 1$ unabhängig von h besitzen. Damit erhält man für die Anzahl k der notwendigen Iterationen, um einen Anfangsfehler um einen Faktor ε zu reduzieren:

$$k \approx \frac{-\ln \varepsilon}{-\ln q} = O(1).$$

Somit benötigt man nur $O(k\, n_h) = O(n_h)$ Operationen für die Durchführung des Mehrgitterverfahrens, es ist also ein optimales Verfahren.

Kapitel 5

Nichtlineare Gleichungssysteme

Lineare Probleme sind oft Idealisierungen eigentlich nichtlinearer Probleme.

Beispiel: Die Wärmeleitgleichung ist nur linear, wenn man annimmt, dass die Materialeigenschaften, wie die Wärmeleitfähigkeit, unabhängig von der Temperatur sind. Tatsächlich ist in vielen Situationen diese Temperaturabhängigkeit zu berücksichtigen. Dann besitzt die stationäre Wärmeleitgleichung die Form

$$-\left(\lambda(u(x))u'(x)\right)' = f(x).$$

Eine mögliche Diskretisierung führt auf die Differenzengleichungen

$$\frac{1}{h}\left[\lambda\left(\frac{u_i+u_{i+1}}{2}\right)\frac{u_{i+1}-u_i}{h} - \lambda\left(\frac{u_{i-1}+u_i}{2}\right)\frac{u_i-u_{i-1}}{h}\right] = f(x_i)$$

für $i = 1, 2, \ldots, N$, oder kurz

$$K_h(\underline{u}_h) = \underline{f}_h.$$

Durch Diskretisierung solcher nichtlinearer Differentialgleichungsprobleme (mit entsprechenden Randbedingungen) erhält man dann nichtlineare Gleichungssysteme, die im Allgemeinen von folgender Form sind:

Gesucht sind n Zahlen $x_1, x_2, \ldots, x_n \in \mathbb{R}$ mit

$$\begin{aligned} F_1(x_1, x_2, \ldots, x_n) &= 0, \\ F_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\vdots \\ F_n(x_1, x_2, \ldots, x_n) &= 0, \end{aligned}$$

wobei die Funktionen $F_i \colon \mathbb{R}^n \to \mathbb{R}$ für $i = 1, 2, \ldots, n$ vorgegeben sind, oder kurz

$$F(x) = 0$$

mit $x = (x_1, x_2, \ldots, x_n)^T$, $F\colon \mathbb{R}^n \to \mathbb{R}^n$, $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T$. Man spricht von einem nichtlinearen Gleichungssystem. Etwas genauer müsste man eigentlich von einem nicht notwendig linearen Gleichungssystem sprechen, da der Spezialfall eines linearen Gleichungssystems durch die Setzung $F(x) = Ax - b$ nicht ausgeschlossen ist.

Bis auf vereinzelte Spezialfälle stehen keine direkten Verfahren zur Lösung nichtlinearer Gleichungen zur Verfügung. Es kommen daher im Allgemeinen nur Iterationsverfahren zur Berechnung einer Näherungslösung eines nichtlinearen Gleichungssystems in Frage.

Wie im linearen Fall erhält man durch Umwandlung eines nichtlinearen Gleichungssystems

$$F(x) = 0$$

in Fixpunktform

$$x = G(x)$$

ein Iterationsverfahren der Form

$$x^{(k+1)} = G(x^{(k)}), \quad k = 1, 2, \ldots, \tag{5.1}$$

für einen geeignet gewählten Startwert $x^{(0)}$. Ein Iterationsverfahren der Form (5.1) heißt ein stationäres Einschrittverfahren. (Ein Verfahren heißt Einschrittverfahren, wenn zur Berechnung der nächsten Näherung nur der aktuelle Näherungswert verwendet wird. Ein Verfahren heißt stationär, wenn die Berechnungsformel unabhängig vom Iterationsindex ist.) Man spricht auch von einer Fixpunktiteration oder einem Verfahren der sukzessiven Approximation.

Sei $W(x)$ für alle x eine reguläre Matrix, und sei $\tau > 0$. Eine mögliche Fixpunktform für $F(x) = 0$ ist durch

$$x = x - \tau\, W(x)^{-1} F(x)$$

gegeben (präkonditioniertes Richardson-Verfahren). Das entsprechende Iterationsverfahren lautet dann:

$$x^{(k+1)} = x^k - \tau\, W(x^k)^{-1} F(x^k).$$

Beispiel: Gesucht ist die Lösung der Gleichung

$$F(x) = e^{\frac{x}{2}} + x - 2 = 0.$$

Für $\tau = 1$ und $W(x) = 1$ entsteht die Iteration:

$$x^{(k+1)} = G(x^{(k)}) = 2 - e^{\frac{x^{(k)}}{2}}.$$

Für den Startwert $x^{(0)} = 1$ erhält man

$$\begin{aligned} x^{(0)} &= 1.000, \\ x^{(1)} &= 0.351, \\ x^{(2)} &= 0.808, \\ x^{(3)} &= 0.502, \\ x^{(4)} &= 0.715, \\ x^{(5)} &= 0.571, \\ x^{(6)} &= \underline{0.6}70, \\ x^{(7)} &= \underline{0.6}02, \\ x^{(8)} &= \underline{0.6}48, \\ x^{(9)} &= \underline{0.6}16, \\ x^{(10)} &= \underline{0.6}38, \\ x^{(11)} &= \underline{0.62}4, \\ x^{(12)} &= \underline{0.63}4, \end{aligned}$$

also eine sehr langsam gegen die exakte Lösung $x^* = 0.629\,846\,115\,690\,812\,2$ konvergente Folge von Näherungen (eine richtige Dezimalstelle in 5 Iterationen).

Wichtige Aussagen über die Konvergenz von stationären Einschrittverfahren liefert der Banachsche Fixpunktsatz:

Satz 5.1 (Banach)**.** *Sei D eine abgeschlossene Menge und sei $G\colon D \to \mathbb{R}^n$ eine Abbildung mit $G(D) \subset D$ und*

$$\|G(y) - G(x)\| \leq q\,\|y - x\| \quad \textit{für alle } x, y \in D$$

mit $q < 1$. Dann existiert ein eindeutiger Fixpunkt in D, die Folge

$$x^{(k+1)} = G(x^{(k)})$$

konvergiert gegen diesen Fixpunkt für alle Startwerte $x^{(0)} \in D$ und es gelten die folgenden Fehlerabschätzungen:

1. $\|e^{(k+1)}\| \leq q \cdot \|e^{(k)}\|$ *(q–lineare Konvergenz),*
2. $\|e^{(k)}\| \leq Cq^k$ *(r–lineare Konvergenz)*

für eine Konstante C.

Bemerkung: Sei x^* ein Fixpunkt von G im Inneren des Definitionsbereiches und sei G differenzierbar in x^*. Dann folgt aus der Bedingung $\|G'(x^*)\| < 1$ die q–lineare Konvergenz, aus der Bedingung $\rho(G'(x^*)) < 1$ zumindest die r–lineare Konvergenz für die Startwerte, die hinreichend nahe bei x^* liegen (Satz von Ostrowski). Im Spezialfall $n = 1$ erhält man lokal monotone Konvergenz für $0 < G'(x^*) < 1$ bzw. alternierende Konvergenz für $-1 < G'(x^*) < 0$.

Beispiel: Durch Taylor-Entwicklung sieht man sofort, dass für die Iterationsfolge

$$x^{(k+1)} = G(x^{(k)})$$

mit

$$G(x) = x - \tau \frac{F(x)}{W(x)}$$

zur Lösung der Gleichung

$$F(x) = e^{\frac{x}{2}} + x - 2 = 0$$

gilt:

$$x^{(k+1)} - x^* = G(x^{(k)}) - G(x^*) \approx G'(x^*)(x^{(k)} - x^*).$$

wobei

$$G'(x^*) = 1 - \tau \frac{W'(x^*)F(x^*) - W(x^*)F'(x^*)}{W^2(x^*)} = 1 - \tau \frac{F'(x^*)}{W(x^*)} = 1 - \tau \frac{\frac{1}{2}e^{\frac{x^*}{2}} + 1}{W(x^*)} \approx 1 - \tau \frac{1,93}{W(x^*)}.$$

Für $\tau = 1$ und $W(x) = 1$ folgt:

$$G'(x^*) \approx -0,93$$

Das Verfahren konvergiert also q–linear:

$$|e^{(k+1)}| \leq q\,|e^{(k)}|$$

mit $q \approx |G'(x^*)| \approx 0,93$. Man erhält in diesem Fall alternierende Konvergenz.
Für $\tau = 1$ und $W(x) = 4$ folgt:

$$G'(x^*) \approx +0,52$$

Man erhält in diesem Fall monotone Konvergenz.

Für die Wahl $\tau = 1$ und $W(x) = F'(x)$ folgt $G'(x^*) = 0$. Also ist mit einer besonders schnellen Konvergenz zu rechnen. Diese Wahl führt auf das Newton-Verfahren, das im Folgenden näher diskutiert wird.

5.1 Das Newton-Verfahren

Das wichtigste Iterationsverfahren zur Lösung nichtlinearer Gleichungssysteme ist das Newton-Verfahren, das im Folgenden neu hergeleitet wird.

Angenommen, eine Näherung $x^{(k)}$ der exakten Lösung x^* des nichtlinearen Gleichungssystems

$$F(x) = 0$$

ist bekannt. Man wäre in einem Schritt am Ziel, wenn man jenen Zuwachs $p^{(k,*)}$ kennen würde, für den gilt:

$$x^{(k)} + p^{(k,*)} = x^*.$$

Daraus erhält man die Bedingung

$$F(x^{(k)} + p^{(k,*)}) = 0. \tag{5.2}$$

Nun wird die linke Seite durch Taylor-Entwicklung linearisiert:

$$F(x^{(k)} + p^{(k,*)}) \approx F(x^{(k)}) + F'(x^{(k)})p^{(k,*)}.$$

wobei $F'(x)$ die Jacobi-Matrix von F an der Stelle x bezeichnet:

$$F'(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1}(x) & \frac{\partial F_1}{\partial x_2}(x) & \cdots & \frac{\partial F_1}{\partial x_n}(x) \\ \frac{\partial F_2}{\partial x_1}(x) & \frac{\partial F_2}{\partial x_2}(x) & \cdots & \frac{\partial F_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(x) & \frac{\partial F_n}{\partial x_2}(x) & \cdots & \frac{\partial F_n}{\partial x_n}(x) \end{pmatrix}.$$

Aus der idealen Bedingungsgleichung (5.2), die allerdings nicht konstruktiv ist, wird durch diese Linearisierung die Bedingung

$$F(x^{(k)}) + F'(x^{(k)})p^{(k)} = 0.$$

Man erhält somit die Iterationsvorschrift:

$$x^{(k+1)} = x^{(k)} + p^{(k)} \quad \text{mit} \quad F'(x^{(k)})p^{(k)} = -F(x^{(k)}). \tag{5.3}$$

Dieses Iterationsverfahren heißt Newton-Verfahren.

Geometrische Interpretation: Offensichtlich gilt:

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

also

$$L_k(x^{(k+1)}) = 0.$$

mit

$$L_k(x) = F(x^{(k)}) + F'(x^{(k)})(x - x^{(k)}).$$

Die (affin) lineare Funktion $L_k(x)$ besitzt an der Stelle $x^{(k)}$ den gleichen Funktionswert und die gleiche erste Ableitung wie $F(x)$.

$$F(x) \approx F(x^{(k)}) + F'(x^{(k)})(x - x^{(k)}) = L_k(x).$$

Die entscheidende Idee des Newton-Verfahrens ist also die Approximation der nichtlinearen Funktion $F(x)$ durch die (affin) lineare Funktion $L_k(x)$ in jedem Iterationsschritt (Linearisierung), deren Nullstelle dann die nächste Näherung für x^* bestimmt.

Im Fall $n = 1$ erhält man also den nächsten Iterationspunkt als Schnittpunkt der Tangente an den Graphen von F im Punkt $x^{(k)}$ mit der x-Achse.

Durch Elimination von $p^{(k)}$ lässt sich das Newton–Verfahren auch folgendermaßen darstellen:

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

Das Newton–Verfahren ist also ein stationäres Einschrittverfahren

$$x^{(k+1)} = G(x^{(k)})$$

mit

$$G(x) = x - F'(x)^{-1} F(x).$$

Für den Spezialfall $n = 1$ erhält man die einfachere Darstellung:

$$x^{(k+1)} = x^{(k)} - \frac{F(x^{(k)})}{F'(x^{(k)})}.$$

Man beachte allerdings, dass zur Durchführung des Verfahrens nicht die Inverse der Jacobi-Matrix $F'(x)$ gebildet wird, sondern der Zuwachs $p^{(k)}$ durch Lösen des linearen Gleichungssystems $F'(x^{(k)})p^{(k)} = -F(x^{(k)})$ bestimmt wird:

Die Durchführung des Verfahrens (5.3) kann man in folgende Schritte trennen:

1. Berechne $r^{(k)} = -F(x^{(k)})$, $W^{(k)} = F'(x^{(k)})$.
2. Löse $W^{(k)} p^{(k)} = r^{(k)}$.
3. Setze $x^{(k+1)} = x^{(k)} + p^{(k)}$.

Konvergenzeigenschaften des Newton-Verfahrens

Satz 5.2. *Sei $F\colon \mathbb{R}^n \to \mathbb{R}^n$ zweimal differenzierbar. $x^* \in \mathbb{R}^n$ sei Nullstelle von F, also $F(x^*) = 0$, und $F'(x^*)$ sei regulär. Dann konvergiert das Newton-Verfahren für Startwerte $x^{(0)}$, die hinreichend nahe bei x^* liegen und es gelten die folgenden Fehlerabschätzungen:*

1. $\|e^{(k+1)}\| \leq C \cdot \|e^{(k)}\|^2$ *(q–quadratische Konvergenz),*
2. $\|e^{(k)}\| \leq q^{2^k - 1} \|e^{(0)}\|$ *(r–quadratische Konvergenz).*

für Konstanten $C > 0$ und $q < 1$.

Beweis. (Für den Fall $n = 1$) Das Newton-Verfahren lässt sich folgendermaßen darstellen:

$$x^{(k+1)} = G(x^{(k)}) \quad \text{mit} \quad G(x) = x - \frac{F(x)}{F'(x)}.$$

Für x^* gilt natürlich $x^* = G(x^*)$.

Es gilt:

$$G'(x) = 1 - \frac{F'(x)^2 - F(x)F''(x)}{F'(x)^2},$$

also

$$G'(x^*) = 0.$$

Für den Fehler $e^{(k)} = x^{(k)} - x^*$ erhält man damit:

$$\begin{aligned} e^{(k+1)} &= x^{(k+1)} - x^* = G(x^{(k)}) - G(x^*) \\ &\approx G'(x^*)(x^{(k)} - x^*) + \frac{1}{2}G''(x^*)(x^{(k)} - x^*)^2 = \frac{1}{2}G''(x^*)\left(e^{(k)}\right)^2 \end{aligned}$$

Also folgt (korrekterweise erst durch Restgliedabschätzung) die erste Fehlerabschätzung:

$$|e^{(k+1)}| \leq C\,|e^{(k)}|^2.$$

Daraus ergibt sich leicht die zweite Abschätzung:

$$\begin{aligned} |e^{(k)}| &\leq C\,|e^{(k-1)}|^2 \leq C\,C^2\,|e^{(k-2)}|^4 \leq C\,C^2\,C^4\,|e^{(k-3)}|^8 \leq \cdots \\ &\leq C\,C^2\,C^4 \cdots C^{2^{k-1}}\,|e^{(0)}|^{2^k} = C^{1+2+4+\cdots+2^{k-1}}\,|e^{(0)}|^{2^k} \\ &= C^{2^k-1}\,|e^{(0)}|^{2^k} = \left(C|e^{(0)}|\right)^{2^k-1}|e^{(0)}| = q^{2^k-1}\,|e^{(0)}| \end{aligned}$$

□

Bemerkung: Die Bedingung, dass $F'(x^*)$ regulär ist, reduziert sich für $n = 1$ auf die Forderung $F'(x^*) \neq 0$. Es wird also vorausgesetzt, dass die Tangente an den Graphen von F in der Nullstelle x^* nicht waagrecht ist. Solche Nullstellen heißen einfache Nullstellen. Im Falle $F'(x^*) = 0$ spricht man von mehrfachen Nullstellen.

Zur Illustration dieser Begriffe betrachte man folgendes Zahlenbeispiel:

1. Für die Fehler einer r–linear konvergenten Folge $\|e^{(k)}\| \leq q^k\,\|e^{(0)}\|$ mit $\|e^{(0)}\| \leq 1$ and $q = 0.1$ erhält man:

$$\begin{aligned} \|e^{(0)}\| &\leq 1.000\ 000 \\ \|e^{(1)}\| &\leq 0.100\ 000 \\ \|e^{(2)}\| &\leq 0.010\ 000 \\ \|e^{(3)}\| &\leq 0.001\ 000 \\ \|e^{(4)}\| &\leq 0.000\ 100 \\ \|e^{(5)}\| &\leq 0.000\ 010 \end{aligned}$$

   Bei einer r–linear konvergenten Folge nimmt also die Anzahl der „richtigen“ Stellen linear zu, hier eine Stelle pro Iteration.

2. Für die Fehler einer r–quadratisch konvergenten Folge $\|e^{(k)}\| \leq q^{2^k-1}\|e^{(0)}\|$ mit $\|e^{(0)}\| \leq 1$ and $q = 0.1$ erhält man: Für $q = 0.1$ erhält man:

$$\begin{aligned} \|e^{(0)}\| &\leq 1.000\ 000\ 000 \\ \|e^{(1)}\| &\leq 0.100\ 000\ 000 \\ \|e^{(2)}\| &\leq 0.001\ 000\ 000 \\ \|e^{(3)}\| &\leq 0.000\ 000\ 100 \end{aligned}$$

Bei einer r–quadratisch konvergenten Folge verdoppelt sich die Anzahl der „richtigen“ Stellen in jedem Schritt. Die Anzahl der „richtigen“ Stellen nehmen also exponentiell zu.

Beispiel: Das Newton-Verfahren für die Gleichung

$$F(x) = e^{\frac{x}{2}} + x - 2 = 0$$

lautet

$$x^{(k+1)} = x^{(k)} - \frac{e^{\frac{x^{(k)}}{2}} + x^{(k)} - 2}{\frac{1}{2}e^{\frac{x^{(k)}}{2}} + 1}.$$

Für den Startwert $x^{(0)} = 1$ erhält man

$$\begin{aligned} x^{(0)} &= 1., \\ x^{(1)} &= \underline{0.6}44, \\ x^{(2)} &= \underline{0.629\ 8}67, \\ x^{(3)} &= \underline{0.629\ 846\ 115\ 7}38, \\ x^{(4)} &= \underline{0.629\ 846\ 115\ 690\ 812}, \end{aligned}$$

also eine sehr schnell gegen die exakte Lösung $x^* = 0.629\ 846\ 115\ 690\ 812\ \ldots$ konvergente Folge von Näherungen.

Beispiel: Die Berechnung der positiven Wurzel aus einer positiven Zahl a, also die Lösung der nichtlinearen Gleichung

$$x^2 - a = 0$$

ist ein weiteres einfaches Beispiel für das Newton-Verfahren:

$$x^{(k+1)} = x^{(k)} - \frac{\left(x^{(k)}\right)^2 - a}{2x^{(k)}} = \frac{1}{2}\left(x^{(k)} + \frac{a}{x^{(k)}}\right).$$

Jede positive (Maschinen-)Zahl a kann man in der Form $a = m \cdot 2^e$ mit einer geraden Zahl e und $m \in [0.5, 2)$ darstellen. Dann gilt: $\sqrt{a} = \sqrt{m} \cdot 2^{e/2}$. Verwendet man zur Berechnung von $\sqrt{m}$ das Newton-Verfahren mit Startwert $m^{(0)} = 1$ für $m \in [0.5, 1]$ und Startwert $m^{(0)} = 1.5$ für $m \in [1, 2)$, so benötigt man nur 5 Iterationen, um den relativen Fehler unterhalb der Maschinengenauigkeit eps $= 2^{-53}$ zu halten. Die Wurzelfunktion ist (etwa) auf diese Weise am Computer implementiert.

5.2 Varianten des Newton-Verfahrens

Der Satz 5.2 liefert nur die lokale Konvergenz, d.h., der Startwert muss hinreichend nahe bei einer einfachen Nullstelle sein.

Beispiel: Das Newton-Verfahren zur Lösung der Gleichung

$$\arctan x = 0$$

konvergiert nur für Startwerte $x^{(0)}$ mit $|x^{(0)}| < x_{krit}$ mit $x_{krit} \approx 1.4$.

Zusammenfassend lässt sich also sagen: Der große Vorteil des Newton-Verfahrens ist die schnelle Konvergenz. Diesem Vorteil stehen als Nachteile die nur lokale Konvergenz und der hohe Aufwand zur Durchführung des Newton-Verfahrens gegenüber.

In den nächsten beiden Abschnitten werden Modifikationen des Newton-Verfahrens diskutiert, die die oben genannten Nachteile abschwächen, ohne dabei den Vorteil der schnellen Konvergenz vollständig zu verlieren.

Zunächst wird kurz eine Strategie vorgestellt, die im Allgemeinen die Menge der Startwerte, für die das Iterationsverfahren konvergiert, vergrößert.

5.2.1 Das gedämpfte Newton-Verfahren

Das Newton-Verfahren lässt sich auch als ein Liniensuchverfahren interpretieren. Ausgehend von der aktuellen Näherung $x^{(k)}$ berechnet man eine so genannte Suchrichtung $p^{(k)}$ als Lösung der Gleichung

$$F'(x^{(k)})p^{(k)} = -F(x^{(k)}),$$

und wählt als nächste Näherung jenen Punkt, den man von $x^{(k)}$ aus in Richtung $p^{(k)}$ mit Schrittweite 1 erreicht. Beim so genannten gedämpften Newton-Verfahren verallgemeinert man nun dieses Vorgehen, indem man für den nächsten Iterationspunkt auch Schrittweiten $\alpha^{(k)} \in (0, 1]$ zulässt:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}.$$

Dabei wird der Parameter $\alpha^{(k)}$ so gewählt, dass

$$\|F(x^{(k+1)})\|_2^2 < \|F(x^{(k)})\|_2^2. \tag{5.4}$$

Man erhofft sich dadurch, dass der nächste Iterationspunkt „näher“ bei der Nullstelle liegt, da für die Nullstelle x^* natürlich gilt: $\|F(x^*)\|_2^2 = 0$.

Eine praktische Durchführung wäre z.B. das sukzessive Durchprobieren der Werte

$$1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8}, \ldots,$$

für $\alpha^{(k)}$, bis das Kriterium (5.4) schließlich erfüllt ist.

Der folgende Satz zeigt die grundsätzliche Durchführbarkeit dieser Strategie:

Satz 5.3. *Sei $F\colon \mathbb{R}^n \to \mathbb{R}^n$ differenzierbar. Für $x^{(k)} \in \mathbb{R}^n$ gelte: $F(x^{(k)}) \neq 0$ und $F'(x^{(k)})$ sei regulär. Dann gilt für $p^{(k)} = -F'(x^{(k)})^{-1}F(x^{(k)})$: Es gibt eine Zahl $\alpha_0 > 0$ mit*

$$\|F(x^{(k)} + \alpha p^{(k)})\|^2 < \|F(x^{(k)})\|^2$$

für alle $\alpha \in (0, \alpha_0)$.

Beweis. Durch Taylor-Entwicklung erhält man für $f(x) = \|F(x)\|^2$:

$$f(x^{(k)} + \alpha p^{(k)}) \approx f(x^{(k)}) + \alpha f'(x^{(k)})p^{(k)}.$$

Die Ableitung von $f(x) = \|F(x)\|_2^2 = F(x)^T F(x)$ ist durch

$$f'(x) = 2F(x)^T F'(x)$$

gegeben. Somit folgt:

$$f'(x^{(k)})p^{(k)} = -2F(x^{(k)})^T F'(x^{(k)})F'(x^{(k)})^{-1}F(x^{(k)}) = -2F(x^{(k)})^T F(x^{(k)}) = -2f(x^{(k)}) < 0.$$

Also gilt

$$f(x^{(k)} + \alpha p^{(k)}) \approx (1 - 2\alpha)f(x^{(k)}) < f(x^{(k)})$$

für hinreichend kleine Werte von α. □

Dieser Satz zeigt also, dass durch sukzessive Verkleinerung von α (z.B. durch Halbierung) das Kriterium (5.4) immer erfüllt werden kann.

Bemerkung: Das Kriterium (5.4) ist etwas zu schwach, um unter geeigneten Voraussetzungen die Konvergenz des gedämpften Newton-Verfahrens nachweisen zu können. Man verwendet stattdessen das (durch die Taylor-Entwicklung im Beweis des Satzes 5.3 motivierte) Kriterium

$$f(x^{(k+1)}) < (1 - 2\mu\alpha)f(x^{(k)})$$

für die Wahl der nächsten Schrittweite $\alpha^{(k)}$, wobei der Parameter $\mu \in (0, 1)$ vorzugeben ist, z.B. $\mu = 0.1$.

5.2.2 Inexakte Newton-Verfahren

Für den Fall, dass zwar die Berechnung der Jacobi-Matrix $F'(x^{(k)})$ wenig Rechenzeit kostet, dafür aber die Lösung des linearen Gleichungssystems

$$F'(x^{(k)})p^{(k)} = -F(x^{(k)})$$

zu aufwendig ist, bieten sich iterative Verfahren zur näherungsweisen Lösung des linearen Gleichungssystems an.

Wichtig dabei ist die richtige Wahl eines Abbruchkriteriums für diese (innere) Iteration. Wird zu früh abgebrochen, kostet zwar eine inexakte Newton-Iteration wenig, es ist

jedoch zu befürchten, dass durch die relativ große Abweichung von der exakten Newton-Iteration die schnelle Konvergenz des Verfahrens verloren geht. Andererseits kostet eine zu genaue Berechnung der nächsten Näherung zuviel Rechenzeit. Es lässt sich zeigen, dass ein Abbruchkriterium der Form

$$\|r^{(k)}\| \leq \eta_k \, \|F(x^{(k)})\| \quad \text{für } r^{(k)} = F(x^{(k)}) + F'(x^{(k)})p^{(k)}$$

mit $\eta_k \leq \eta < 1$ die lokale Konvergenz eines inexakten Newton-Verfahrens garantiert.

Für den Fall, dass die Berechnung der Jacobi-Matrix $F'(x^{(k)})$ explizit nicht möglich oder zu aufwendig ist, bieten sich Differenzenquotienten zur Approximation der Jacobi–Matrix an.

Eine besonders effiziente Strategie, die der Einfachheit halber zunächst nur für den Fall $n = 1$ diskutiert wird, führt auf das so genannte Sekantenverfahren:

Man ersetzt im Newton-Verfahren die Ableitung $F'(x^{(k+1)})$ durch die Approximation

$$A^{(k+1)} = \frac{F(x^{(k+1)}) - F(x^{(k)})}{x^{(k+1)} - x^{(k)}}.$$

Das bedeutet geometrisch, dass man statt der Tangente im Punkt $x^{(k+1)}$ die Sekante in den Punkten $x^{(k+1)}$ und $x^{(k)}$ zur Linearisierung verwendet. Der Schnittpunkt dieser Sekante mit der x-Achse bestimmt die nächste Näherung.

Als Iterationsvorschrift erhält man

$$\begin{aligned} x^{(k+2)} &= x^{(k+1)} - \frac{F(x^{(k+1)})}{A^{(k+1)}} = x^{(k+1)} - \frac{F(x^{(k+1)})}{\dfrac{F(x^{(k+1)}) - F(x^{(k)})}{x^{(k+1)} - x^{(k)}}} \\ &= \frac{x^{(k)}F(x^{(k+1)}) - x^{(k+1)}F(x^{(k)})}{F(x^{(k+1)}) - F(x^{(k)})}. \end{aligned}$$

Eine Verallgemeinerung dieses Verfahrens für den allgemeinen Fall $n \geq 1$ ist das so genannte Broyden-Rang-1-Verfahren, bei dem eine Approximation $A^{(k+1)}$ der Jacobi-Matrix gesucht wird, die die so genannte Sekantenbedingung

$$A^{(k+1)}s^{(k)} = y^{(k)} \quad \text{mit} \quad s^{(k)} = x^{(k+1)} - x^{(k)}, \quad y^{(k)} = F(x^{(k+1)}) - F(x^{(k)})$$

erfüllt. Diese Bedingung wird durch folgende Setzung erfüllt:

$$A^{(k+1)} = A^{(k)} + u^{(k)}{v^{(k)}}^T$$

mit

$$u^{(k)} = \frac{1}{{s^{(k)}}^T s^{(k)}} \left(y^{(k)} - A^{(k)}s^{(k)}\right), \quad v^{(k)} = s^{(k)}.$$

Ein Iterationsschritt des Broyden-Rang-1-Verfahrens besteht also aus folgenden Teilschritten:

1. Löse die Gleichung $A^{(k)}p^{(k)} = -F(x^{(k)})$.
2. Berechne $x^{(k+1)} = x^{(k)} + p^{(k)}$.
3. Berechne $A^{(k+1)} = A^{(k)} + u^{(k)}{v^{(k)}}^T$.

Beim Broyden-Rang-1-Verfahren benötigt man zur Berechnung einer Approximation der Jacobi-Matrix keine zusätzlichen Funktionsauswertungen. $A^{(k+1)}$ erhält man aus $A^{(k)}$ und den Funktionswerten $F(x^{k+1})$ und $F(x^{(k)})$, die man für die Berechnung von $x^{(k+1)}$ und $x^{(k)}$ ohnehin benötigt, durch einige wenige zusätzliche arithmetische Operationen pro Matrixelement.

Auch die Lösung des linearen Gleichungssystems, das in jedem Iterationsschritt anfällt, lässt sich schneller als beim gewöhnlichen Newton-Verfahren durchführen. So erhält man z.B. eine LU–Zerlegung von $A^{(k+1)}$ auf Grund der speziellen Struktur aus der LU–Zerlegung von $A^{(k)}$ in $O(n^2)$ Operationen. Im Gegensatz dazu würde der Aufwand zur Lösung des linearen Gleichungssystems im Allgemeinen proportional zu n^3 steigen.

Das Broyden-Rang-1-Verfahren konvergiert trotz der Abweichungen vom Newton-Verfahren immer noch lokal und q–superlinear, d.h.

$$\lim_{k\to\infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} = 0.$$

Das ist umso überraschender, als die Folge der Approximationen $A^{(k)}$ im Allgemeinen nicht gegen die exakte Jacobi-Matrix $F'(x^*)$ konvergiert.

Bemerkung: Bei der praktischen Durchführung des Broyden-Rang-1-Verfahrens wird nach allen K Schritten ein so genannter „restart“ mit der exakten Jacobi-Matrix durchgeführt. Dadurch verhindert man zu schlechte Approximationen $A^{(k)}$ der Jacobi-Matrix.

Kapitel 6

Anfangswertprobleme gewöhnlicher Differentialgleichungen

In diesem Kapitel werden sogenannte Anfangswertprobleme für gewöhnliche Differentialgleichungen diskutiert.

Beispiel: Die Temperaturverteilung in einem homogenen Stab wird durch die instationäre Wärmeleitgleichung

$$\frac{\partial u}{\partial t}(x,t) - a\,\frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t) \quad \text{für } x \in (0,L),\ t \in (0,t_E)$$

beschrieben. Wählt man die Randbedingungen

$$\begin{aligned} u(0,t) &= g_a(t) \quad \text{für } t \in (0,t_E) \\ u(L,t) &= g_b(t) \quad \text{für } t \in (0,t_E) \end{aligned}$$

und die Anfangsbedingung

$$u(x,0) = u_A(x) \quad \text{für } x \in [0,L],$$

so erhält man ein sogenanntes Anfangsrandwertproblem.

Führt man wie im ersten Kapitel beschrieben eine Ortsdiskretisierung mit der Finiten-Differenzen-Methode durch, so erhält man für die Näherungen $u_i(t)$ der exakten Lösung $u(x_i,t)$ folgendes System von gewöhnlichen Differentialgleichungen:

$$\frac{du_i}{dt}(t) - \frac{a}{h^2}(u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)) = f(x_i,t) \quad \text{für alle } i = 1,2,\ldots,N.$$

Gemeinsam mit den Randbedingungen $u_0(t) = g_a(t)$ und $u_{N+1}(t) = g_b(t)$ und der Anfangsbedingung $u_i(0) = u_A(x_i)$ für $i = 0,1,\ldots,N+1$ lässt sich dieses System auch kompakt in folgender Form schreiben:

$$\begin{aligned} \underline{u}_h'(t) + K_h\underline{u}_h(t) &= \underline{f}_h(t), \quad \text{für } t \in (0,t_E) \\ \underline{u}_h(0) &= \underline{u}_{0h} \end{aligned}$$

mit $\underline{u}_h(t) = (u_1(t), u_2(t), \ldots, u_N(t))^T$, $u_{0h} = (u_A(x_1), u_A(x_2), \ldots, u_A(x_N))^T$ und

$$K_h = \frac{a}{h^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}, \quad \underline{f}_h = \begin{pmatrix} f(x_1,t) + \frac{a}{h^2} g_a(t) \\ f(x_2,t) \\ \vdots \\ f(x_{N-1},t) \\ f(x_N,t) + \frac{a}{h^2} g_b(t) \end{pmatrix}.$$

Diese Problemstellung hat folgende allgemeine Form:

Gesucht ist eine Funktion $u(t)$, die das folgende Anfangswertproblem erfüllt:

$$\begin{aligned} u'(t) &= f(t, u(t)), \quad t \in (0, T) \\ u(0) &= u_0. \end{aligned} \tag{6.1}$$

Beispiel: Bei der Diskretisierung der instationären Wärmeleitgleichung mit der Finiten-Differenzen-Methode erhält man diese Form mit den Setzungen

$$u(t) = \underline{u}_h(t), \quad f(t,u) = \underline{f}_h - K_h \underline{u}_h, \quad u_0 = \underline{u}_{0h}.$$

Auch andere Problemstellungen lassen sich auf die obige Standardform bringen.

Beispiel: Die Bewegungsgleichung einer Punktmasse mit Masse m und Position $x(t)$ lautet nach dem Newtonschen Gesetz:

$$mx''(t) = F(t, x(t), x'(t)),$$

wobei $F(t, x, x')$ die angreifende Kraft in Abhängigkeit der Zeit t, des Ortes x und der Geschwindigkeit x' beschreibt. Die dazugehörigen Anfangsbedingungen schreiben Startwerte für den Ort und die Geschwindigkeit der Punktmasse vor:

$$x(0) = x_0, \quad x'(t) = v_0.$$

Mit den Setzungen

$$u_1(t) = x(t), \quad u_2(t) = x'(t)$$

lässt sich dieses Anfangswertproblem einer Differentialgleichung zweiter Ordnung auf die Standardform (6.1) bringen:

$$\begin{aligned} u_1'(t) &= u_2(t), \\ u_2'(t) &= \frac{1}{m} F(t, u_1(t), u_2(t)) \end{aligned}$$

mit der Anfangsbedingung

$$\begin{aligned} u_1(0) &= x_0, \\ u_2(0) &= v_0. \end{aligned}$$

Bemerkung: Auf analoge Weise lässt sich jede Differentialgleichung höherer als erster Ordnung als ein System von Differentialgleichungen erster Ordnung schreiben. Die Beschränkung der Diskussion auf Systeme erster Ordnung ist also keine Einschränkung der Allgemeinheit.

Spezialfälle:

- Die rechte Seite der Differentialgleichung hängt nicht explizit von u ab:

$$\begin{aligned} u'(t) &= f(t), \quad t \in (0,T), \\ u(0) &= u_0. \end{aligned}$$

  Die Lösung lässt sich dann in folgender Weise darstellen:

$$u(t) = u_0 + \int_0^t f(s)\, ds.$$

  Die Berechnung der Lösung des Anfangswertproblems reduziert sich also in diesem Fall auf die Integration der rechten Seite $f(t)$.

- Die rechte Seite der Differentialgleichung hängt nicht explizit von t ab:

$$\begin{aligned} u'(t) &= f(u(t)), \quad t \in (0,T), \\ u(0) &= u_0. \end{aligned}$$

  Man nennt die Differentialgleichung in diesem Fall autonom.

  Dieser Fall ist nicht wirklich ein Spezialfall, denn jede Differentialgleichung

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

  lässt sich auch als autonomes Differentialgleichungssystem schreiben. Mit der Setzung

$$u_1(t) = t, \quad u_2(t) = u(t)$$

  erhält man nämlich

$$\begin{aligned} u_1'(t) &= 1, \\ u_2'(t) &= f(u_1(t), u_2(t)). \end{aligned}$$

6.1 Quadraturformeln

siehe:
W. Zulehner, Numerische Mathematik. Eine Einführung anhand von Differentialgleichungsproblemen. Band 2: Instationäre Probleme, Mathematik Kompakt. Basel: Birkhäuser, Seiten 31 - 39, 2011.

6.2 Die Eulersche Polygonzugmethode

Das Intervall $[0,T]$ wird durch eine Folge von Gitterpunkten

$$0 = t_0 < t_1 < \ldots < t_m = T$$

z.B. mit $t_j = j \cdot \tau$ für $j = 0, 1, \ldots, m$ und einer vorgegebenen Schrittweite $\tau = T/m$ diskretisiert.

Motivation durch Taylor-Reihenentwicklung:

Durch Taylor–Entwicklung an der Stelle $t_0 = 0$ erhält man

$$u(t) \approx u(t_0) + u'(t_1)(t - t_0) = u_0 + f(t_0, u_0)(t - t_0) \equiv u_\tau(t) \quad \text{für } t \in [t_0, t_1].$$

Dadurch erhält man an der Stelle t_1 die Näherung

$$u_1 = u_0 + \tau\, f(t_0, u_0).$$

Durch Taylor–Entwicklung an der Stelle t_1 erhält man

$$\begin{aligned} u(t) &\approx u(t_1) + u'(t_1)(t - t_1) \\ &= u(t_1) + f(t_1, u(t_1))(t - t_1) \approx u_1 + f(t_1, u_1)(t - t_1) \equiv u_\tau(t) \quad \text{für } t \in [t_1, t_2]. \end{aligned}$$

Dadurch erhält man an der Stelle t_2 die Näherung

$$u_2 = u_1 + \tau\, f(t_1, u_1).$$

Setzt man den Approximationsprozess analog fort, so erhält man einen durch $u_\tau(t)$ beschriebenen Polygonzug als Approximation der exakten Lösung, der die Näherungen an den Gitterpunkten, gegeben durch

$$u_{j+1} = u_j + \tau\, f(t_j, u_j) \quad j = 0, 1, \ldots, m - 1$$

verbindet.

Motivation als FDM:

Ersetzt man in der Differentialgleichung an der Stelle t_j die Ableitung durch den Vorwärtsdifferenzenquotienten:

$$u'(t_j) \approx \frac{1}{\tau}(u(t_{j+1}) - u(t_j)),$$

so erhält man das Eulersche Polygonzugverfahren als Finite Differenzen Methode:

$$\frac{1}{\tau}(u_{j+1} - u_j) = f(t_j, u_j) \quad j = 0, 1, \ldots, m - 1.$$

Motivation durch Quadraturformeln:

Durch Integration der Differentialgleichung

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

über dem Intervall $[t, t + \tau]$ erhält man die Integralbeziehung:

$$u(t + \tau) = u(t) + \int_t^{t+\tau} f(s, u(s))\, ds.$$

Wird das Integral durch die linksseitige Rechtecksregel approximiert, also

$$\int_t^{t+\tau} f(s,u(s))\,ds \approx \tau\, f(t,u(t)),$$

so entsteht

$$u(t+\tau) \approx u(t) + \tau\, f(t,u(t)). \tag{6.2}$$

Das motiviert die Formel

$$u_{j+1} = u_j + \tau\, f(t_j,u_j), \quad j = 0,1,\ldots,m-1 \tag{6.3}$$

zur sukzessiven Berechnung von Näherungen $u_j = u_\tau(t_j)$ der exakten Werte $u(t_j)$.

6.3 Die klassische Konvergenzanalyse

Ziel des Verfahrens ist es natürlich, durch Wahl einer entsprechend kleinen Schrittweite eine kleine Abweichung von der exakten Lösung zu erreichen. Es sollte also gelten:

$$e_\tau(t) \to 0 \quad \text{für } \tau \to 0 \tag{6.4}$$

für alle $t \in [0,T]$, wobei $e_\tau(t)$ den so genannten **globalen (Diskretisierungs-)Fehler** bezeichnet:

$$e_\tau(t) = u(t) - u_\tau(t).$$

Im Allgemeinen beschränkt man sich darauf, den globalen Fehler an den Gitterpunkten zu betrachten und spricht von einem **konvergenten Verfahren**, falls das Verfahren die Eigenschaft (6.4) (in einem geeigneten Sinn) erfüllt.

Der globale Fehler setzt sich aus einzelnen Beiträgen zusammen, die als Fortpflanzung von lokalen Fehlern $d_\tau(t_j)$, $j = 0,1,\ldots$ durch das Verfahren interpretiert werden können.

Dabei beschreibt der lokale Fehler an der Stelle t_{j+1}, für die Euler-Methode gegeben durch

$$d_\tau(t_{j+1}) = u(t_{j+1}) - u_\tau(t_{j+1}) = u(t_{j+1}) - \Big(u(t_j) + \tau\, f(t_j,u(t_j))\Big),$$

den Unterschied zwischen exakter Lösung der Differentialgleichung und der Näherungslösung nach einem Schritt des Verfahrens, jeweils vom Startpunkt $(t_j,u(t_j))$ aus betrachtet.

Als $d_\tau(t_0)$ definiert man einen eventuell vorhandenen Fehler bei der Eingabe des Startwertes:

$$d_\tau(t_0) = u(0) - u_\tau(0).$$

Für den Konsistenzfehler des als FDM interpretierten Eulerschen Polygonzugverfahrens gilt folgender Zusammenhang mit den lokalen Fehlern:

$$\psi_\tau(t_{j+1}) = \frac{u(t_{j+1}) - u(t_j)}{\tau} - f(t_j,u(t_j)) = \frac{1}{\tau} d_\tau(t_{j+1}).$$

Die Untersuchung der Größe der lokalen Fehler nennt man eine Konsistenzanalyse. Man spricht von einem **konsistenten Verfahren**, falls der Konsistenzfehler für $\tau \to 0$ verschwindet, falls also (in einem geeigneten Sinn):

$$d_\tau = o(\tau).$$

Um die Fortpflanzung der lokalen Fehler abschätzen zu können, muss die Differenz $w_j - v_j$ in Abhängigkeit der Startdifferenz $w_{j_0} - v_{j_0}$ für $j \geq j_0$ untersucht werden, wobei die Folgen v_j und w_j durch das Näherungsverfahren, hier das Euler-Verfahren, erzeugt werden:

$$\begin{aligned} v_{j+1} &= v_j + \tau\, f(t_j, v_j), \\ w_{j+1} &= w_j + \tau\, f(t_j, w_j), \end{aligned}$$

ausgehend von Anfangssetzungen v_{j_0} und w_{j_0} an der Stelle t_{j_0}. Diese Untersuchung nennt man eine Stabilitätsanalyse.

Lässt sich für ein Verfahren eine Abschätzung der Form

$$\|w_j - v_j\| \leq C \|w_{j_0} - v_{j_0}\|$$

mit einer von τ unabhängige Konstante C nachweisen, dann heißt das Verfahren **stabil**.

Um also die Konvergenz eines Verfahrens nachzuweisen, muss die Konsistenz und die Stabilität des Verfahrens untersucht werden.

Für das Euler-Verfahren sind die Konsistenz- und Stabilitätsanalyse leicht durchzuführen:

Konsistenzanalyse:

Durch Taylor-Entwicklung erhält man:

$$\begin{aligned} d_\tau(t+\tau) &= u(t+\tau) - u(t) - \tau\, f(t, u(t)) \\ &= u(t) + u'(t)\tau + \frac{1}{2}u''(t)\tau^2 + \cdots - u(t) - \tau\, f(t, u(t)) \\ &= \Big[u'(t) - f(t, u(t))\Big]\,\tau + \frac{1}{2}u''(t)\tau^2 + \cdots = \frac{1}{2}u''(t)\tau^2 + \cdots = O(\tau^2) \end{aligned}$$

Es gibt also eine Konstante K mit

$$\|d_\tau\|_{L^\infty(0,T)} \leq K\,\tau^2$$

oder kurz

$$d_\tau = O(\tau^2).$$

Der lokale Fehler konvergiert also mindestens so schnell wie τ^2 gegen 0, falls τ gegen 0 konvergiert.

Falls der lokale Fehler eines Verfahrens mindestens so schnell wie $K\tau^{p+1}$ mit $p > 0$ für $\tau \to 0$ gegen 0 konvergiert, oder kurz

$$d_\tau = O(\tau^{p+1}),$$

so nennt man das Verfahren **konsistent** mit **Konsistenzordnung** p.

Das Euler-Verfahren ist nach den obigen Überlegungen konsistent mit Konsistenzordnung 1.

Stabilitätsanalyse:

Es gelte folgende Voraussetzung für die rechte Seite der Differentialgleichung: Es gibt eine Konstante $L \geq 0$ mit

$$\|f(t,w) - f(t,v)\| \leq L \, \|w - v\| \quad \text{für alle } t, v, w. \tag{6.5}$$

Man nennt diese Bedingung eine Lipschitz-Bedingung.

Dann gilt:

$$\begin{aligned} \|w_{j+1} - v_{j+1}\| &= \| \, [w_j + \tau \, f(t_j, w_j)] - [v_j + \tau \, f(t_j, v_j)] \, \| \\ &\leq \|w_j - v_j\| + \tau \, \|f(t_j, w_j) - f(t_j, v_j)\| \leq (1 + \tau \, L) \, \|w_j - v_j\|. \end{aligned}$$

Durch wiederholte Anwendung dieser Abschätzung erhält man:

$$\|w_j - v_j\| \leq (1 + \tau \, L)^{j - j_0} \, \|w_{j_0} - v_{j_0}\| \leq \left(e^{\tau \, L}\right)^{j - j_0} \, \|w_{j_0} - v_{j_0}\| = e^{(t_j - t_{j_0}) L} \, \|w_{j_0} - v_{j_0}\|.$$

Es gibt also eine von τ unabhängige Konstante $C = e^{(t_j - t_{j_0}) L} \leq e^{TL}$ mit

$$\|w_j - v_j\| \leq C \, \|w_{j_0} - v_{j_0}\|.$$

Es wurde also unter der Voraussetzung (6.5) nachgewiesen, dass das Euler-Verfahren stabil ist.

Nach diesen Vorbereitungen lässt sich nun die Konvergenz des Verfahrens untersuchen:

Konvergenzanalyse

Der globale Fehler setzt sich aus der Fortpflanzung der einzelnen lokalen Fehlern zusammen. Für ein stabiles und konsistentes Verfahren der Konsistenzordnung $p > 0$ gilt (für $e_\tau(t_0) = 0$):

$$\|e_\tau(t_j)\| \leq \sum_{k=1}^{j} C \|d_\tau(t_k)\| \leq \sum_{k=1}^{j} C \, K \, \tau^{p+1} = C \, K \, \tau^p \sum_{k=1}^{j} \tau \leq C \, K \, t_j \, \tau^p = C' \, \tau^p$$

mit $C' = C \, K \, t_j \leq C \, K \, T$, also:

$$e_\tau = O(\tau^p).$$

Das Verfahren ist daher konvergent. Etwas genauer spricht man von einem konvergenten Verfahren mit **Konvergenzordnung p**.

Bemerkung: Kurz zusammengefasst lässt sich die obige Konvergenzanalyse auf die Formel

$$\text{Konsistenz} \; + \; \text{Stabilität} \; = \; \text{Konvergenz}$$

bringen.

Im Speziellen gilt natürlich, dass das Euler-Verfahren ein konvergentes Verfahren mit Konvergenzordnung 1 ist.

6.4 Die expliziten Runge-Kutta-Formeln

Verwendet man zur Approximation des Integrals in der Integralbeziehung

$$u(t+\tau) = u(t) + \int_t^{t+\tau} f(\sigma, u(\sigma))\, d\sigma \tag{6.6}$$

genauere Näherungsformeln (Quadraturformeln), so erhält man genauere Verfahren.

So kann man z.B. die so genannte Mittelpunktsregel zur Approximation des Integrals verwenden:

$$\int_t^{t+\tau} f(\sigma, u(\sigma))\, d\sigma \approx \tau\, f\left(t + \frac{\tau}{2}, u\left(t + \frac{\tau}{2}\right)\right).$$

Die in der Approximation auftretende Größe $u(t+\tau/2)$ ist allerdings nicht verfügbar und muss aus der Integralbeziehung

$$u\left(t + \frac{\tau}{2}\right) = u(t) + \int_t^{t+\tau/2} f(\sigma, u(\sigma))\, d\sigma$$

ebenfalls durch Verwendung einer Quadraturformel, z.B. der linksseitigen Rechtecksregel (Euler-Methode), approximiert werden:

$$u\left(t + \frac{\tau}{2}\right) \approx u(t) + \frac{\tau}{2} f(t, u(t)).$$

Damit erhält man insgesamt das so genannte verbesserte Euler-Verfahren:

$$u_{j+1} = u_j + \tau\, f\left(t_j + \frac{\tau}{2}, g_2\right)$$

mit

$$\begin{aligned} g_1 &= u_j, \\ g_2 &= u_j + \frac{\tau}{2} f(t_j, g_1). \end{aligned}$$

Zur Untersuchung der Konsistenz muss wie beim Euler-Verfahren der lokale Fehler analysiert werden:

$$\begin{aligned} d_\tau(t+\tau) &= u(t+\tau) - u_\tau(t+\tau) \\ &= u(t+\tau) - u(t) - \tau\, f\left(t + \frac{\tau}{2}, u(t) + \frac{\tau}{2} f(t, u(t))\right) \\ &= u(t) + u'(t)\tau + \frac{1}{2}u''(t)\tau^2 + O(\tau^3) - u(t) \\ &\quad - \tau\left[f(t, u(t)) + f_t(t, u(t))\frac{\tau}{2} + f_u(t, u(t))\frac{\tau}{2} f(t, u(t)) + O(\tau^2)\right] \\ &= [u'(t) - f(t, u(t))]\tau + [u''(t) - f_t(t, u(t)) - f_u(t, u(t)) f(t, u(t))]\frac{\tau^2}{2} + O(\tau^3). \end{aligned}$$

Aus

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

folgt durch Differentiation

$$u''(t) = f_t(t, u(t)) + f_u(t, u(t))u'(t) = f_t(t, u(t)) + f_u(t, u(t))f(t, u(t)).$$

Also erhält man insgesamt für den lokalen Fehler:

$$d_\tau(t + \tau) = O(\tau^3),$$

d.h.: die Konsistenzordnung des verbesserten Euler-Verfahrens ist 2.

Die obige Konstruktion von Verfahren lässt sich leicht verallgemeinern. Ausgangspunkt ist eine s-stufige Quadraturformel:

$$\begin{aligned}\int_t^{t+\tau} & f(s, u(s))\, ds \\ &\approx \tau \Big[b_1\, f(t, u(t)) + b_2\, f(t + c_2\,\tau, u(t + c_2\,\tau)) + \cdots + b_s\, f(t + c_s\,\tau, u(t + c_s\,\tau))\Big].\end{aligned}$$

Die Zahlen b_i, $i = 1, 2, \ldots, s$ nennt man die Gewichte, die Punkte $t + c_i\,\tau$, $i = 1, 2, \ldots, s$ mit $c_1 = 0$ die Stützpunkte der Quadraturformel.

Anstelle der unbekannten Funktionswerte $u(t + c_i\,\tau)$ werden Näherungen

$$g_i \approx u(t + c_i\,\tau)$$

rekursiv durch Anwendung von $(i-1)$-stufigen Quadraturformeln auf

$$u(t + c_i\,\tau) = u(t) + \int_t^{t+c_i\,\tau} f(s, u(s))\, ds$$

berechnet. Dann erhält man:

$$\begin{aligned} g_1 &= u_j, \\ g_2 &= u_j + \tau\, a_{21}\, f(t_j, g_1), \\ g_3 &= u_j + \tau \Big[a_{31}\, f(t_j, g_1) + a_{32}\, f(t_j + c_2\,\tau, g_2)\Big], \\ &\vdots \\ g_s &= u_j + \tau \Big[a_{s1}\, f(t_j, g_1) + a_{s2}\, f(t_j + c_2\,\tau, g_2) + \cdots + a_{s,s-1}\, f(t_j + c_{s-1}\,\tau, g_{s-1})\Big]. \end{aligned} \tag{6.7}$$

Die nächste Näherung hat dann die Form:

$$u_{j+1} = u_j + \tau \Big[b_1\, f(t_j, g_1) + b_2\, f(t_j + c_2\,\tau, g_2) + \cdots + b_s\, f(t_j + c_s\,\tau, g_s)\Big]. \tag{6.8}$$

Man nennt die Methode (6.7), (6.8) eine s-stufige explizite Runge-Kutta-Formel. Zur Beschreibung der Methode genügt es, das folgende Tableau anzugeben:

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} \\ \hline & b_1 & b_2 & \dots & b_{s-1} \quad b_s \end{array}$$

oder in kompakter Form:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}.$$

Beispiele: Die Euler-Methode ist eine 1-stufige Runge-Kutta-Formel mit Tableau

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

und Konsistenzordnung 1.

Die verbesserte Euler-Methode ist eine 2-stufige Runge-Kutta-Formel mit Tableau

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

und Konsistenzordnung 2.

Durch geeignete Wahl der Koeffizienten des Tableaus erreicht man entsprechend hohe Konsistenzordungen. Bezeichnet man mit $s(p)$ die minimale Stufenzahl, die notwendig ist, um die Ordnung p zu erreichen, so gilt:

p	1	2	3	4	5	6	7	8
$s(p)$	1	2	3	4	6	7	9	11

Nach dieser Tabelle lässt sich mit einer 4-stufigen Runge-Kutta-Formel die Konsistenzordnung 4 erreichen. Der bekannteste Vertreter aus dieser Klasse ist die „klassische“ Runge-Kutta-Formel mit Tableau

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

Unter der Voraussetzung (6.5) lässt sich für die gesamte Klasse der expliziten Runge-Kutta-Formeln die Stabilität zeigen.

6.5 Steife Differentialgleichungen und A-Stabilität

Für das Beispiel der semidiskretisierten Wärmeleitgleichung erhält man für die kleinstmögliche Lipschitz-Konstante der rechten Seite:

$$\|f(t,w) - f(t,v)\| = \|K_h(\underline{w}_h - \underline{v}_h)\| \leq L \, \|\underline{w}_h - \underline{v}_h\|$$

in der Spektralnorm:

$$L = \|K_h\|_2 = \lambda_{\max}(K_h) \approx \frac{4a}{h^2}.$$

Also folgt:

$$L = O\left(\frac{1}{h^2}\right) \gg 1.$$

Damit werden Stabilitätsschranken der Form $C = e^{tL}$ völlig unbrauchbar.

Das Stabilitätsverhalten des kontinuierlichen Problems bezüglich der Anfangswerte ist wesentlich gutartiger: Seien $\underline{v}_0$ und $\underline{w}_0$ unterschiedliche Startwerte der Differentialgleichung

$$\underline{u}_h'(t) = \underline{f}_h(t) - K_h \underline{u}_h.$$

Dann erhält man für $\underline{z}_h(t) = \underline{w}_h(t) - \underline{v}_h(t)$ das Anfangswertproblem

$$\begin{aligned} \underline{z}_h'(t) &= -K_h \, \underline{z}_h(t), \\ \underline{z}_h(0) &= \underline{w}_{h0} - \underline{v}_{h0}. \end{aligned}$$

Daraus folgt:

$$(\underline{z}_h'(t), \underline{z}_h(t))_2 = -(K_h \underline{z}_h(t), \underline{z}_h(t))_2 \leq 0.$$

Wegen

$$(\underline{z}_h'(t), \underline{z}_h(t))_2 = \frac{1}{2}\frac{d}{dt}(\underline{z}_h(t), \underline{z}_h(t))_2 = \frac{1}{2}\frac{d}{dt}\|\underline{z}_h(t)\|_2^2 = \|\underline{z}_h(t)\|_2 \frac{d}{dt}\|\underline{z}_h(t)\|_2$$

folgt

$$\frac{d}{dt}\|\underline{z}_h(t)\|_2 \leq 0.$$

Das bedeutet

$$\|\underline{w}_h(t) - \underline{v}_h(t)\|_2 \leq \|\underline{w}_{h0} - \underline{v}_{h0}\|_2.$$

Allgemeiner erhält man für Differentialgleichungssysteme

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

mit der Eigenschaft

$$(f(t,w) - f(t,v), w - v) \leq 0$$

auf gleiche Weise die Abschätzung

$$\|w(t) - v(t)\| \leq \|w(0) - v(0)\| \quad \text{für alle } t \in [0,T]$$

für zwei Lösungen $w(t)$ und $v(t)$. Man nennt solche Differentialgleichungen dissipativ.

Diese Stabilitätseigenschaft wird nun auch für das Näherungsverfahren gefordert. Es soll also gelten

$$\|w_{j+1} - v_{j+1}\| \leq \|w_j - v_j\| \quad \text{für alle } v_j, w_j.$$

Man nennt solche Näherungsverfahren kontraktiv.

Für kontraktive Verfahren erhält man für die Stabilitätskonstante $C = 1$ und es gelten entsprechenden Konvergenzaussagen.

Wir nun untersuchen, unter welchen Bedingungen ein Differentialgleichungssystem dissipativ und eine Runge-Kutta-Methode kontraktiv ist.

Zunächst wird eine lineare Differentialgleichung der Form

$$u'(t) = \lambda\, u(t) \tag{6.9}$$

mit $\lambda \in \mathbb{C}$ betrachtet.

Für die Runge-Kutta-Methode, angewendet auf (6.9), erhält man in diesem Spezialfall:

$$\begin{aligned} g &= u_j e + \tau\lambda A g, \\ u_{j+1} &= u_j + \tau\lambda b^T g \end{aligned}$$

mit $g = (g_1, g_2, \ldots, g_s)^T$ und $e = (1, 1, \ldots, 1)^T$, also

$$u_{j+1} = R(\tau\lambda)\, u_j$$

mit

$$R(z) = 1 + z b^T (I - zA)^{-1} e.$$

$R(z)$ heißt die Stabilitätsfunktion der Runge-Kutta-Methode.

Beispiele: Explizite Euler-Methode:

$$R(z) = 1 + z$$

Verbesserte Euler-Methode:

$$R(z) = 1 + z + \frac{1}{2}z^2.$$

Klassische Runge-Kutta-Methode:

$$R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4.$$

Bemerkung: Für die exakte Lösung der Differentialgleichung (6.9) erhält man

$$u(t) = u_0 e^{\lambda t}$$

und es gilt:

$$u(t_{j+1}) = e^{\lambda\tau} u(t_j).$$

Die Genauigkeit einer Runge-Kutta-Methode entspricht der Genauigkeit der Approximation der Exponentialfunktion e^z durch die Stabilitätsfunktion $R(z)$ in einer Umgebung von $z = 0$.

Die Differentialgleichung (6.9) ist genau dann dissipativ, wenn

$$\operatorname{Re}\lambda \leq 0.$$

Die entsprechende Eigenschaft der Kontraktivität für die Runge-Kutta-Methode führt auf die Bedingung

$$|R(\tau\lambda)| \leq 1.$$

Mit Hilfe der Stabilitätsfunktion lässt sich der so genannte Stabilitätsbereich einer Runge-Kutta-Methode definieren:

$$S = \{z \in \mathbb{C} : |R(z)| \leq 1\}.$$

Mit dieser Notation lautet die obige Bedingung an die Schrittweite τ (Kontraktivität):

$$\tau\lambda \in S.$$

Die Aussagen lassen sich sofort auf lineare Systeme

$$u'(t) = Ju(t) \tag{6.10}$$

übertragen, für den Fall, dass J eine konstante und normale Matrix ist:

$$J^T J = JJ^T.$$

Für normale Matrizen gilt:

$$J = UDU^H \quad \text{mit } D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

wobei U eine unitäre Matrix ist, $\lambda_i \in \mathbb{C}$ sind die Eigenwerte von J, die Spaltenvektoren u_i von U sind die dazugehörigen Eigenvektoren.

Das System

$$u'(t) = Ju(t)$$

ist genau dann dissipativ, wenn

$$\operatorname{Re}\lambda \leq 0 \quad \text{für alle } \lambda \in \sigma(J).$$

Beweis. Sei $v \in \mathbb{R}^n$. Dann gilt:

$$(Jv, v) = \operatorname{Re}(Jv, v) = \operatorname{Re}(UDU^H v, v) = \operatorname{Re}(DU^H v, U^H v) = \sum_i \operatorname{Re}\lambda_i |w_i|^2$$

für $w = U^H v$. □

Eine Runge-Kutta-Methode für (6.10) ist genau dann kontraktiv, falls

$$\|R(\tau J)\| \leq 1.$$

Beweis.

$$\|w_{j+1} - v_{j+1}\| \;=\; \|R(\tau J)(w_j - v_j)\| \le \|(w_j - v_j)\| \quad \text{für alle } v_j,\ w_j.$$

□

Es gilt:

$$\|R(\tau J)\| = \|R(\tau U D U^H)\| = \|U R(\tau D) U^H\| = \|R(\tau D)\| = \max_{\lambda \in \sigma(J)} |R(\tau\lambda)|$$

Also ist eine Runge-Kutta-Methode ist genau dann kontraktiv, falls

$$|R(\tau\lambda)| \le 1 \quad \text{für alle } \lambda \in \sigma(J).$$

Beispiel: Das eindimensionale Modellproblem der semidiskretisierten Wärmeleitgleichung führt auf ein lineares System der Form

$$\underline{u}_h'(t) = J\underline{u}_h(t) + \underline{f}_h(t) \quad \text{mit} \quad J = -K_h.$$

Da K_h symmetrisch und positiv definit ist, besitzt die Matrix J nur reelle und negative Eigenwerte λ.

Für die Eulersche Polygonzugmethode erhält man als Stabilitätsbereich die Kreisscheibe mit Mittelpunkt -1 und Radius 1:

$$S = \{z \in \mathbb{C} : |z - (-1)| \le 1\}.$$

Die Bedingung $\tau\,\lambda \in S$ führt daher auf

$$\tau \le \frac{2}{|\lambda|},$$

die garantiert, dass das Verfahren bei Anwendung auf

$$u'(t) = \lambda u(t)$$

kontraktiv ist.

Die Eigenwerte von K_h sind durch

$$\lambda_k = \frac{4a}{h^2} \sin^2\left(\frac{k\,\pi}{2(N+1)}\right), \quad k = 1, 2, \ldots, N,$$

gegeben, die Eigenwerte von J sind dann durch $-\lambda_k$ für $k = 1, 2, \ldots, N$ gegeben. Die einschränkendste Bedingung ergibt sich für den betragsgrößten negativen Eigenwert:

$$|\lambda_{\max}(J)| = \lambda_N = \frac{4a}{h^2} \sin^2\left(\frac{N\pi}{2(N+1)}\right) = \frac{4a}{h^2} \cos^2\left(\frac{\pi}{2(N+1)}\right) \le \frac{4a}{h^2}.$$

Die Stabilitätsbedingung

$$\tau \leq \frac{2}{4a/h^2} = \frac{h^2}{2a}$$

oder, dazu äquivalent

$$\frac{a\,\tau}{h^2} \leq \frac{1}{2},$$

stellt daher sicher, dass das Verfahren für alle Eigenwerte kontraktiv ist.

Das bedeutet, dass die Zeitschrittweite τ für kleine Ortsschrittweiten h besonders klein gewählt werden muss. Das kann zu unverhältnismäßig vielen Zeitschritten und somit zu einem sehr hohen Rechenaufwand führen, um ein vorgegebenes Zeitintervall $[0, T]$ zu überstreichen.

Diese Stabilitätsbedingung erklärt das Verhalten der numerischen Experimente im ersten Kapitel: Die Stabilitätsbedingung ist in den Fällen $h = 10$ cm, $\tau = 1$ min und $h = 1$ cm, $\tau = 1$ s nicht erfüllt, im Fall $h = 10$ cm, $\tau = 50$ s hingegen erfüllt.

Dieses Anfangswertproblem ist ein Beispiel für eine steife Differentialgleichung: Die Existenz von Eigenwerten mit großem negativem Realteil sind die Ursache, dass nur bei sehr kleinen Schrittweiten die Eulersche Polygonzugmethode sinnvolle (d.h. beschränkte) Näherungen erzeugt.

Es wäre wünschenswert, wenn für alle Werte von λ, die zu beschränkten exakten Lösungen führen, auch die Näherungsmethode beschränkte (stabile) Näherungen erzeugt. Das führt auf den Begriff der A-Stabilität:

Definition 6.1. *Eine Runge-Kutta-Methode heißt A-stabil, falls*

$$|R(z)| \leq 1 \quad \textit{für alle } \lambda \in \mathbb{C} \textit{ mit } \operatorname{Re}\lambda \leq 0$$

oder kurz:

$$\mathbb{C}^- \subset S$$

mit $\mathbb{C}^- = \{z \in \mathbb{C} : \operatorname{Re}\lambda \leq 0\}$.

Die Stabilitätsfunktion einer expliziten s-stufigen Runge-Kutta-Methode ist ein Polynom vom Grad $\leq s$, also kann keine dieser Methoden A-stabil sein.

6.6 Die impliziten Runge-Kutta-Formeln

Man erhält die Eulersche Polygonzugmethode (oder auch explizite Euler-Methode), indem man in (6.6) die linksseitige Rechtecksregel verwendet. Bei Verwendung der rechtsseitigen Rechtecksregel

$$\int_t^{t+\tau} f(s, u(s))\, ds \approx \tau\, f(t+\tau, u(t+\tau))$$

entsteht die so genannte implizite Euler-Methode:

$$u_{j+1} = u_j + \tau\, f(t_j + \tau, u_{j+1}).$$

Zur tatsächlichen Berechnung von u_{j+1} muss also eine im Allgemeinen nichtlineare Gleichung gelöst werden.

Verwendet man zur Verbesserung der Genauigkeit statt der Rechtecksregel die Mittelpunktsregel:

$$\int_t^{t+\tau} f(s,u(s))\,ds \approx \tau\, f(t+\frac{\tau}{2}, u(t+\frac{\tau}{2}))$$

und approximiert die benötigte Lösung $u(t+\tau/2)$ durch eine Näherung g_1 mit Hilfe der impliziten Euler-Methode, so erhält man die so genannte implizite Mittelpunktsregel:

$$\begin{aligned} g_1 &= u_j + \frac{\tau}{2} f(t_j + \frac{\tau}{2}, g_1), \\ u_{j+1} &= u_j + \tau\, f(t_j + \frac{\tau}{2}, g_1). \end{aligned}$$

Diese beiden Verfahren sind Beispiele von so genannten impliziten Runge-Kutta-Formeln.

Die allgemeine Form einer s-stufigen Runge-Kutta-Formel lässt sich folgendermaßen schreiben:

$$\begin{aligned} g_1 &= u_j + \tau\,[a_{11} f(t+c_1\tau, g_1) + a_{12} f(t+c_2\tau, g_2) + \cdots + a_{1s} f(t+c_s\tau, g_s)]\,, \\ g_2 &= u_j + \tau\,[a_{21} f(t+c_1\tau, g_1) + a_{22} f(t+c_2\tau, g_2) + \cdots + a_{2s} f(t+c_s\tau, g_s)]\,, \\ &\vdots \\ g_s &= u_j + \tau\,[a_{s1} f(t+c_1\tau, g_1) + a_{s2} f(t+c_2\tau, g_2) + \cdots + a_{ss} f(t+c_s\tau, g_s)] \end{aligned} \tag{6.11}$$

und

$$u_{j+1} = u_j + \tau\,[b_1 f(t+c_1\tau, g_1) + b_2 f(t+c_2\tau, g_2) + \cdots + b_s f(t+c_s\tau, g_s)] \tag{6.12}$$

Das Verfahren ist durch folgendes Tableau eindeutig beschrieben:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & & & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

oder in kompakter Form:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}. \tag{6.13}$$

Definition 6.2. *Eine durch das Tableau (6.13) beschriebene Runge-Kutta-Formel heißt*

1. *explizit, falls A eine echte linke untere Dreiecksmatrix ist,*
2. *implizit, falls sie nicht explizit ist.*

Diese Definition einer expliziten Formel ist geringfügig allgemeiner als bisher. Bisher wurde stets angenommen, dass $c_1 = 0$.

Beispiele:

1. Die implizite Euler-Methode ist eine 1-stufige implizite Runge-Kutta-Formel mit Tableau:
$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

2. Die implizite Mittelpunktsregel ist eine 1-stufige implizite Runge-Kutta-Formel mit Tableau:
$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}.$$

3. Eine weitere mögliche Quadraturformel für (6.6) ist durch
$$\int_t^{t+\tau} f(s, u(s))\, ds \approx \tau \Big[(1-\theta)\, f(t, u(t)) + \theta\, f(t+\tau, u(t+\tau))\Big]$$
gegeben. Sie führt auf das Verfahren:
$$u_{j+1} = u_j + \tau \Big[(1-\theta)\, f(t_j, u_j) + \theta\, f(t_j+\tau, u_{j+1})\Big],$$
also
$$\begin{aligned} g_1 &= u_j, \\ g_2 &= u_j + \tau \Big[(1-\theta)\, f(t_j, g_1) + \theta\, f(t_j+\tau, g_2)\Big], \\ u_{j+1} &= u_j + \tau \Big[(1-\theta)\, f(t_j, g_1) + \theta\, f(t_j+\tau, g_2)\Big]. \end{aligned}$$
Die Methode heißt θ-Methode und ist eine 2-stufige Runge-Kutta-Formel mit Tableau
$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1-\theta & \theta \\ \hline & 1-\theta & \theta \end{array}.$$
Die Klasse enthält als wichtige Spezialfälle das explizite Euler-Verfahren ($\theta = 0$), das implizite Euler-Verfahren ($\theta = 1$) und die implizite Trapezregel ($\theta = 1/2$).

Durchführung impliziter Runge-Kutta-Formeln:

Um die nächste Näherung u_{j+1} mit Hilfe einer impliziten Runge-Kutta-Formel aus (6.12) zu berechnen, müssen zuerst $g_1, g_2, \ldots, g_s$ aus dem nichtlinearen Gleichungssystem (6.11) näherungsweise berechnet werden.

Die Gleichungen liegen in Fixpunktform vor. Daher könnte man eine einfache Fixpunktiteration durchführen, deren Konvergenz für die Startwerte $g_i = u$ bei hinreichend kleiner Schrittweite τ nachgewiesen werden kann. Bessere Startwerte lassen sich durch explizite Runge-Kutta-Formeln berechnen.

Für steife Differentialgleichungen ist ein vereinfachtes Newton-Verfahren zur Berechnung von $g_1, g_2, \ldots, g_s$ aus (6.11) vorzuziehen, wobei es genügt, die Jacobi-Matrix nur einmal und zwar im Startpunkt $g_i = u$ auszuwerten, und dann unverändert als Approximation der Jacobi-Matrix in den weiteren Iterationspunkten zu verwenden.

Beispiel: Die θ-Methode für das Modellproblem der semidiskretisierten Wärmeleitgleichung

$$\begin{aligned} \underline{u}_h'(t) &= \underline{f}_h(t) - K_h \underline{u}_h(t), \\ \underline{u}_h(0) &= \underline{u}_{0,h} \end{aligned}$$

lautet

$$\underline{u}_h^{j+1} = \underline{u}_h^j + \tau \left\{ (1-\theta) \left[\underline{f}_h(t_j) - K_h \underline{u}_h^j \right] + \theta \left[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1} \right] \right\}.$$

Die Näherung $\underline{u}_h^{j+1}$ erhält man also durch Lösen des linearen Gleichungssystems

$$[I + \tau\, \theta\, K_h]\, \underline{u}_h^{j+1} = [I - \tau(1-\theta)\, K_h]\, \underline{u}_h^j + \tau \left[(1-\theta)\, \underline{f}_h(t_j) + \theta\, \underline{f}_h(t_{j+1}) \right].$$

Im Falle $\theta = 1/2$ (implizite Trapezregel) heißt diese Methode auch das Crank-Nicolson-Verfahren.

Konsistenzordnung der impliziten Runge-Kutta-Formeln

Wie im expliziten Fall lassen sich für implizite Runge-Kutta-Formeln der globale, der lokale Fehler und der Konsistenzfehler einführen.

Die Konsistenzordnung lässt sich wie im expliziten Fall durch Taylor-Entwicklungen bestimmen.

Beispiele:

1. Die θ-Methode besitzt im Allgemeinen die Konsistenzordnung 1. Für $\theta = 1/2$ erhält man die Konsistenzordnung 2.

2. Für die 1-stufige implizite Mittelpunktsregel erhält man die Konsistenzordnung 2.

Man sieht an diesen Beispielen bereits, dass mit einer s-stufigen impliziten Runge-Kutta-Formel höhere Konsistenzordnungen als mit expliziten Formeln erreichbar sind.

Es lässt sich zeigen, dass mit einer s-stufigen Runge-Kutta-Formel maximal die Konsistenzordnung $p = 2s$ erreicht werden kann. Diese Methoden mit maximaler Konsistenzordnung heißen Runge-Kutta-Formeln vom Gauß-Typ. Sie beruhen auf den Gaußschen Quadraturformeln, die maximalen Genauigkeitsgrad besitzen. Die Mittelpunktsregel ist eine Gaußsche Quadraturformel mit einem Stützpunkt.

Stabilität von impliziten Runge-Kutta-Formeln

Zur Beurteilung der A-Stabilität benötigt man wieder die Stabilitätsfunktion.

Beispiele:

1. Für die impliziten Euler-Methode erhält man

$$R(z) = \frac{1}{1-z}.$$

   Der Stabilitätsbereich ist das Komplement der offenen Kreisscheibe mit Mittelpunkt 1 und Radius 1:

$$S = \{z \in \mathbb{C} : |z-1| \geq 1\}.$$

   Das Verfahren ist also A-stabil.

2. Für die θ-Methode erhält man

$$R(z) = \frac{1+(1-\theta)z}{1-\theta z}.$$

   Die Methode ist für $\theta \geq 1/2$ A-stabil.

3. Für die implizite Mittelpunktsregel erhält man:

$$G(z) = \frac{1+z/2}{1-z/2}.$$

   Der Stabilitätsbereich S ist gleich $\mathbb{C}_-$, das Verfahren ist also A-stabil.

Es gilt zusammenfassend:

Satz 6.1. *A-stabile Runge-Kutta-Methoden für dissipative lineare Systeme von Differentialgleichungen mit konstanter Koeffizientenmatrix J, die normal ist, sind für alle Schrittweiten kontraktiv.*

Beweis. Für die Eigenwerte λ von J gilt: $\lambda \in \mathbb{C}^-$. Daher folgt für alle $\tau > 0$: $\tau\,\lambda \in \mathbb{C}^- \subset S$. □

Bemerkung: Die Aussage des letzten Satzes gilt auch ohne die Einschränkung auf normale Matrizen.

Für nichtlineare Systeme reicht der Begriff der A-Stabilität nicht für die Kontraktivität aus.

Definition 6.3. *Eine Runge-Kutta-Methode heißt B-stabil, falls für alle Anfangswertprobleme mit*

$$(f(t,w) - f(t,v), w - v) \leq 0$$

für die Näherungen gilt:

$$\|w_{j+1} - v_{j+1}\| \leq \|w_j - v_j\|$$

für alle Schrittweiten $\tau > 0$.

Offensichtlich ist also eine B-stabile Methode für alle Schrittweiten $\tau > 0$ kontraktiv.

Bemerkung: Eine gute Implementierung einer Runge-Kutta-Formel erfordert eine effiziente Schrittweitensteuerung. Im Prinzip versucht man, die Schrittweite in jedem Schritt so zu wählen, dass der jeweils neue lokale Fehler einen vorgegebenen Toleranzwert nicht überschreitet.

6.7 Anfangswertprobleme 2. Ordnung

Das am Beginn des Kapitels diskutierte Anfangsrandwertproblem für die Wärmeleitgleichung ist ein typisches Beispiel eines sogenannten parabolischen Problem. Ein Beispiel einer weiteren wichtigen Klasse von Anfangsrandwertproblemen wird nun kurz diskutiert:

Beispiel: Die Schwingung einer Saite wird durch die Wellengleichung

$$\frac{\partial^2 u}{\partial t^2}(x,t) - c^2 \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t) \quad \text{für } x \in (0,L),\ t \in (0,t_E)$$

beschrieben. Wählt man die Randbedingungen

$$\begin{aligned} u(0,t) &= g_a(t) \quad \text{für } t \in (0,t_E) \\ u(L,t) &= g_b(t) \quad \text{für } t \in (0,t_E) \end{aligned}$$

und die Anfangsbedingungen

$$\begin{aligned} u(x,0) &= u_A(x) \quad \text{für } x \in [0,L], \\ \frac{\partial u}{\partial t}(x,0) &= v_A(x) \quad \text{für } x \in [0,L], \end{aligned}$$

so erhält man ein sogenanntes hyperbolisches Anfangsrandwertproblem.

Durch Ortsdiskretisierung erhält man analog wie bei der Wärmeleitgleichung erhält man für die Näherungen $u_i(t)$ der exakten Lösung $u(x_i,t)$ folgendes Anfangswertproblem:

$$\begin{aligned} \underline{u}_h''(t) + K_h \underline{u}_h(t) &= \underline{f}_h(t), \quad \text{für } t \in (0,t_E) \\ \underline{u}_h(0) &= \underline{u}_{0h} \\ \underline{u}_h'(0) &= \underline{v}_{0h} \end{aligned}$$

mit $\underline{u}_h(t) = (u_1(t), u_2(t), \ldots, u_N(t))^T$, $u_{0h} = (u_A(x_1), u_A(x_2), \ldots, u_A(x_N))^T$, $v_{0h} = (v_A(x_1), v_A(x_2), \ldots, v_A(x_N))^T$ und

$$K_h = \frac{c^2}{h^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}, \quad \underline{f}_h = \begin{pmatrix} f(x_1,t) + \dfrac{c^2}{h^2} g_a(t) \\ f(x_2,t) \\ \vdots \\ f(x_{N-1},t) \\ f(x_N,t) + \dfrac{c^2}{h^2} g_b(t) \end{pmatrix}.$$

Setzt man

$$\underline{v}_h(t) = \underline{u}_h'(t),$$

so lässt sich das obige System 2. Ordnung als ein System 1. Ordnung schreiben:

$$\begin{pmatrix} \underline{u}_h(t) \\ \underline{v}_h(t) \end{pmatrix}' = \begin{pmatrix} 0 & I \\ -K_h & 0 \end{pmatrix} \begin{pmatrix} \underline{u}_h(t) \\ \underline{v}_h(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \underline{f}_h(t) \end{pmatrix}$$

mit der Anfangsbedingung

$$\begin{pmatrix} \underline{u}_h(0) \\ \underline{v}_h(0) \end{pmatrix} = \begin{pmatrix} \underline{u}_{0h} \\ \underline{v}_{0h} \end{pmatrix}.$$

Damit lassen sich alle Methoden bzw. Analysen entsprechend übertragen.

Beispiel: Für die θ-Methode erhält man:

$$\begin{aligned} \underline{u}_h^{j+1} &= \underline{u}_h^j + \tau\,[(1-\theta)\underline{v}_h^j + \theta \underline{v}_h^{j+1}], \\ \underline{v}_h^{j+1} &= \underline{v}_h^j + \tau \left\{ (1-\theta)[\underline{f}_h(t_j) - K_h \underline{u}_h^j] + \theta[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1}] \right\}. \end{aligned}$$

Eliminiert man aus der ersten Zeile $\underline{v}_h^{j+1}$ mit Hilfe der zweiten Zeile, so erhält man

$$\begin{aligned} \underline{u}_h^{j+1} &= \underline{u}_h^j + \tau\, \underline{v}_h^j + \theta\tau^2 \left\{ (1-\theta)[\underline{f}_h(t_j) - K_h \underline{u}_h^j] + \theta[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1}] \right\}, \\ \underline{v}_h^{j+1} &= \underline{v}_h^j + \tau \left\{ (1-\theta)[\underline{f}_h(t_j) - K_h \underline{u}_h^j] + \theta[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1}] \right\}. \end{aligned}$$

Also

$$\begin{aligned} \left[I + \theta\tau^2 K_h\right] \underline{u}_h^{j+1} &= \underline{u}_h^j + \tau\, \underline{v}_h^j + \theta\tau^2 \left\{ (1-\theta)[\underline{f}_h(t_j) - K_h \underline{u}_h^j] + \theta\, \underline{f}_h(t_{j+1}) \right\}, \\ \underline{v}_h^{j+1} &= \underline{v}_h^j + \tau \left\{ (1-\theta)[\underline{f}_h(t_j) - K_h \underline{u}_h^j] + \theta[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1}] \right\}. \end{aligned}$$

Zur Beurteilung der Kontraktivität sind die Eigenwerte der Matrix

$$J = \begin{pmatrix} 0 & I \\ -K_h & 0 \end{pmatrix}$$

von entscheidender Bedeutung. Es gilt: Eine Zahl μ ist Eigenwert von J, falls es einen Vektor $(u,v)^T \neq 0$ gibt mit

$$\begin{aligned} v &= \mu\, u, \\ -K_h u &= \mu\, v, \end{aligned}$$

also

$$K_h u_h = -\mu^2\, u.$$

Daher erhält man aus den reellen und positiven Eigenwerten λ von K_h die entsprechenden Eigenwerte μ von J durch

$$\mu = \pm\, i\, \sqrt{\lambda}.$$

Alle Eigenwerte sind rein imaginär, also sind alle skalaren Probleme $w'(t) = \lambda w(t)$ dissipativ. Es lässt sich ein Skalarprodukt konstruieren, sodass J bezüglich dieses Skalarprodukts symmetrisch ist, daher ist auch das System $w'(t) = Jw(t)$ dissipativ.

Natürlich gilt dann wieder, dass alle A-stabilen Runge-Kutta-Methoden für beliebige Schrittweiten τ kontraktiv sind.

Anders als beim parabolischen Fall lässt sich hier zeigen, dass das explizite Euler-Verfahren nie kontraktiv sein kann: Die Bedingung

$$\tau\mu \in S = \{z \in \mathbb{C} : |z+1| \leq 1\}$$

lässt sich durch keine positive Schrittweite erfüllen.

Wendet man für die erste Differentialgleichung

$$\underline{u}_h'(t) = \underline{v}_h(t)$$

und die zweite Differentialgleichung

$$\underline{v}_h'(t) = \underline{f}_h(t) - K_h \underline{u}(t)$$

unterschiedliche Runge-Kutta-Methoden an, so spricht man von partitionierten Runge-Kutta-Methoden. Kombiniert man beispielsweise die explizite Euler-Methode für die erste Gleichung

$$\underline{u}_h^{j+1} = \underline{u}_h^j + \tau\, \underline{v}_h^j$$

mit der impliziten Runge-Kutta für die zweite Gleichung

$$\underline{u}_h^{j+1} = \underline{u}_h^j + \tau\, \left[\underline{f}_h(t_{j+1}) - K_h \underline{u}_h^{j+1}\right],$$

so sieht man sofort, dass die kombinierte Methode nicht die Lösung eines Gleichungssystems erfordert, also explizit ist.

Es lässt sich für dieses explizite Verfahren zeigen, das es unter der Bedingung

$$\tau < \frac{2}{\sqrt{\lambda}}$$

kontraktiv ist. Diese Schrittweitenbeschränkung in τ ist weniger restriktiv als im parabolischen Fall. Wegen $\lambda_{\max}(K_h) \leq 4c^2/h^2$ garantiert also die Bedingung

$$\tau < \frac{h}{c}$$

bzw.

$$\frac{c\tau}{h} < 1$$

die Kontraktivität des Verfahrens.