Wissenschaftliches Rechnen

Steffen Börm

Stand 19. Februar 2016

Inhaltsverzeichnis

1 Einleitung

Unter „Wissenschaftlichem Rechnen“ versteht man den Einsatz von Computern für die Lösung wissenschaftlicher Probleme. In der Regel sind dabei eine Reihe von Teilproblemen zu behandeln:

1. Die mathematische **Modellierung** überführt das Problem in die Form mathematischer Gleichungen, aus deren Analyse sich die Lösung ergibt.

2. Häufig wird durch eine **Diskretisierung** eine kontinuierliche Formulierung durch eine angenähert, die durch *endlich viele* Größen beschrieben werden kann und deshalb für einen Computer handhabbar ist.

3. Anschließend werden **Lösungsverfahren** angewendet, um aus dem mathematischen Modell die für die konkrete Fragestellung wichtigen Größen zu gewinnen.

4. Häufig ist auch eine **Optimierung** erforderlich, beispielsweise um während der Modellierung bestimmte Parameter des Modells so zu wählen, dass reale Experimente möglichst gut erfasst werden, oder um herauszufinden, wie sich ein simulierter Prozess so steuern lässt, dass er bestimmten Zielvorgaben entspricht.

Die Vorlesung konzentriert sich auf die ersten drei Gebiete und zielt dabei darauf, einen Überblick zu geben, statt die mathematische Analyse der einzelnen verwendeten Techniken detailliert vorzustellen.

Interessierte Leserinnen und Lesern können in den Vorlesungsskripten

- „Numerik von Differentialgleichungen“,
- „Finite Elemente“,
- „Iterative Verfahren für große Gleichungssysteme“ und
- „Numerik nicht-lokaler Operatoren“

eine Darstellung der theoretischen Grundlagen der verwendeten Verfahren (und einiger weiterer) finden.

Ein typisches Anwendungsgebiet des Wissenschaftlichen Rechnens ist die Simulation physikalischer Phänomene, beispielsweise um die Bewegungen eines Körpers in einem Gravitationsfeld oder das Strömen von Luft um eine Tragfläche vorhersagen zu können. Derartige Simulationen beruhen auf Naturgesetzen, die die in unzähligen Experimenten gesammelten Erfahrungen in kompakter Form beschreiben, häufig in Gestalt weniger mathematischer Gleichungen.

Indem wir diese Gleichungen auf die konkrete Fragestellung anwenden, entsteht ein Gleichungssystem, das die für uns interessanten Größen zueinander in eine Beziehung bringt. Damit stehen wir vor der Aufgabe, die Lösungen dieses Systems zu untersuchen.

Häufig entstehen dabei *Differentialgleichungen*, also Gleichungen, die die Veränderung bestimmter Größen zu den Größen selber in Verbindung setzt. Ein einfaches Beispiel ist das physikalische Modell eines Federpendels, bei dem die Auslenkung des Pendels eine Kraft hervorruft, die die Geschwindigkeit des Pendels beeinflusst.

Da die Analyse derartiger Differentialgleichungen nur im Ausnahmefall per Hand erfolgen kann, wird man sie meistens einem Computer übertragen. Computer arbeiten der Reihe nach einzelne Befehle ab, die die Werte einzelner Speicherzellen verändern, deshalb sind sie in der Regel nicht dazu geeignet, die kontinuierliche Bewegung eines Teilchens oder die kontinuierliche Verteilung von Wärme in einem erhitzten Körper zu beschreiben. Dieses Problem lässt sich lösen, indem man das Kontinuum durch diskrete (also voneinander getrennte) Punkte ersetzt, beispielsweise die kontinuierliche Zeit durch diskrete Zeitpunkte oder die kontinuierliche Wärmeverteilung durch die Wärme in diskreten Punkten des Körpers. Im Rahmen dieser *Diskretisierung* wird das ursprüngliche Modell durch eine Näherung ersetzt, die meistens zu einer etwas anderen Lösung führt.

Gute Diskretisierungsverfahren sind allerdings *konvergent*, ihre Näherungslösungen streben also gegen die des ursprünglichen Modells, wenn wir die Abstände zwischen den Punkten verkleinern. Allerdings bedeutet beispielsweise eine Halbierung des Abstands zwischen Zeitpunkten im typischen Fall auch eine Verdoppelung der Anzahl der Zeitpunkte. Genaue Simulationen erfordern deshalb in der Regel sehr viel Zeit, sehr viel Speicher, oder sogar beides. Also ist es von großer Bedeutung, für die Lösung der diskreten Modelle möglichst effiziente Algorithmen einzusetzen, die auch Tausende von Zeitschritten und Millionen von Unbekannten in vertretbarer Zeit behandeln können.

Danksagung

Ich bedanke mich bei Dirk Boysen, Philip Freese, Knut Reimer, Sven Marquardt, Hendrik Felix Pohl, Christina Börst, Daniel Hans, Jens Liebenau, Bennet Carstensen und Torsten Knauf für Hinweise auf Fehler in früheren Fassungen dieses Skripts und für Hilfe bei seiner Verbesserung.

2 Bewegung und Kraft

In diesem Kapitel untersuchen wir besonders einfache mechanische Systeme: Punktmassen, die sich unter Einfluss von Kräften bewegen. In der Realität treten solche Punktmassen zwar genau genommen nicht auf, sie eignen sich aber trotzdem sehr gut für die Beschreibung vieler Phänomene. Insbesondere lassen sich realistische Massen als Ansammlung vieler kleiner Punktmassen annähern, so dass sich durch Übergang zu einem Grenzwert allgemeinere Modelle gewinnen lassen.

2.1 Newton-Axiome

Die Grundlage des in diesem Kapitel behandelten Modells bilden die auf Isaac Newton zurückgehenden Axiome der Festkörpermechanik [7]. Wir formulieren sie hier in moderner Notation für eine abstrakte Punktmasse, also für einen Körper ohne räumliche Ausdehnung, der sich zu einem Zeitpunkt $t \in \mathbb{R}$ in einem Punkt $x(t) \in \mathbb{R}^3$ befindet.

Die *Geschwindigkeit*[1] der Punktmasse zu einem Zeitpunkt $t \in \mathbb{R}$ bezeichnen wir mit $v(t)$. Nach den Newtonschen Regeln ist die Geschwindigkeit gerade die Ableitung des Orts nach der Zeit, also

$$v(t) = x'(t) \qquad \text{für alle } t \in \mathbb{R}. \tag{2.1}$$

Falls die Geschwindigkeit konstant ist, falls also $v(t) = v_0$ für ein $v_0 \in \mathbb{R}^d$ gilt, folgt unmittelbar

$$x(t) = x(0) + v_0 t \qquad \text{für alle } t \in \mathbb{R},$$

die Punktmasse bewegt sich also entlang einer Geraden und legt in gleich großen Zeitintervallen gleich große Strecken zurück.

Um realistische Aufgabenstellungen behandeln zu können, müssen wir zulassen, dass sich die Geschwindigkeit ändern darf. Dazu führen wir zu jedem Zeitpunkt $t \in \mathbb{R}$ die *Beschleunigung*[2] $a(t)$ ein, die die Veränderung der Geschwindigkeit mit der Zeit misst und als deren Ableitung durch

$$a(t) = v'(t) \qquad \text{für alle } t \in \mathbb{R} \tag{2.2}$$

gegeben ist.

Eine Beschleunigung entsteht durch eine *Kraft*[3], die auf die Punktmasse wirkt. Dabei sind Kraft und Beschleunigung zueinander proportional, der Faktor hängt von der

[1] Im Englischen *velocity*.
[2] Im Englischen *acceleration*.
[3] Im Englischen *force*.

Trägheit der Masse ab: Bei einer Verdoppelung der Masse muss auch die Kraft verdoppelt werden, um dieselbe Beschleunigung zu erreichen. Die zu einem Zeitpunkt t wirkende Kraft bezeichnen wir mit $f(t)$ und die Masse mit m, so dass sich die Beziehung durch die kurze Formel

$$f(t) = ma(t) \qquad \text{für alle } t \in \mathbb{R} \tag{2.3}$$

ausdrücken lässt.

Insgesamt wird die Bewegung der Punktmasse also durch das System

$$x'(t) = v(t), \qquad v'(t) = a(t), \qquad a(t) = \frac{1}{m} f(t) \qquad \text{für alle } t \in \mathbb{R}$$

beschrieben. Falls uns die Kraft zu jedem Zeitpunkt bekannt ist, lassen sich Geschwindigkeit und Position mit dem Hauptsatz der Integral- und Differentialgleichung ermitteln.

Erinnerung 2.1 (Hauptsatz Integral- und Differentialrechnung) *Für jede stetig differenzierbare Funktion $g : [a, b] \to \mathbb{R}$ gilt*

$$g(b) - g(a) = \int_a^b g'(s)\, ds. \tag{2.4}$$

Falls uns nämlich die Geschwindigkeit zu einem bestimmten Zeitpunkt, beispielsweise $t = 0$, bekannt ist, folgt direkt

$$\begin{aligned} v(t) &\overset{(2.4)}{=} v(0) + \int_0^t v'(s)\, ds \overset{(2.2)}{=} v(0) + \int_0^t a(s)\, ds \\ &\overset{(2.3)}{=} v(0) + \frac{1}{m} \int_0^t f(s)\, ds \qquad \text{für alle } t \in \mathbb{R}. \end{aligned} \tag{2.5}$$

Entsprechend können wir $x(t)$ berechnen, indem wir diese Gleichung mit (2.1) kombinieren und den Hauptsatz ein zweites Mal anwenden.

2.2 Ballistik

Als erstes Beispiel einer Anwendung der Newtonschen Gesetze untersuchen wir die Bewegung eines geworfenen Körpers, der der Schwerkraft der Erde unterliegt. Dabei gehen wir davon aus, dass der Körper nur eine relativ zum Durchmesser der Erde geringe Entfernung zurücklegt, so dass wir die auf ihn wirkende Gravitationskraft als konstant annehmen dürfen.

Wenn wir die erste Komponente der auftretenden Vektoren jeweils als horizontale Entfernung und die zweite als (vertikale) Höhe über dem als flach angenommenen Boden interpretieren, können wir die Gravitationskraft durch die Gleichung

$$f(t) = f_0 := \begin{pmatrix} 0 \\ -gm \end{pmatrix} \qquad \text{für alle } t \in \mathbb{R} \tag{2.6}$$

ausdrücken, wobei g die Erdbeschleunigung bezeichnet.

Zur Abkürzung schreiben wir die Anfangsposition als $x_0 := x(0)$ und die Anfangsgeschwindigkeit als $v_0 := v(0)$. Damit erhalten wir

$$v(t) \overset{(2.5)}{=} v_0 + \frac{1}{m}\int_0^t f_0\,ds = v_0 - \int_0^t \begin{pmatrix}0\\ g\end{pmatrix} ds = \begin{pmatrix}v_{0,1}\\ v_{0,2} - tg\end{pmatrix} \qquad \text{für alle } t \in \mathbb{R}. \quad (2.7)$$

Dieser Gleichung können wir schon einige nützliche Aussagen entnehmen. Beispielsweise ist die horizontale Komponente der Geschwindigkeit konstant, während die vertikale Komponente linear mit der Zeit abnimmt. Zu dem Zeitpunkt

$$t_v = v_{0,2}/g$$

ist die vertikale Komponente gleich null, zu diesem Zeitpunkt erreicht also die Punktmasse ihre maximale Höhe, den Scheitelpunkt ihrer Flugbahn.

Mit (2.1) und dem Hauptsatz erhalten wir aus (2.7) die Gleichung

$$x(t) \overset{(2.1)}{=} x_0 + \int_0^t v(s)\,ds = \begin{pmatrix}x_{0,1} + tv_{0,1}\\ x_{0,2} + tv_{0,2} - t^2 g/2\end{pmatrix} \qquad \text{für alle } t \in \mathbb{R},$$

die die Flugbahn vollständig beschreibt. Dieser Gleichung können wir beispielsweise entnehmen, dass die Punktmasse ihre Ausgangshöhe wieder erreicht, falls

$$x_{0,2} + tv_{0,2} - t^2 g/2 = x_{0,2} \quad \Longleftrightarrow \quad tv_{0,2} - t^2 g/2 = 0 \quad \Longleftrightarrow \quad t(2v_{0,2} - tg) = 0$$

gilt. Neben der trivialen Lösung $t = 0$ besitzt diese Gleichung auch die Lösung $t = 2v_{0,2}/g = 2t_v$, bis zum Erreichen des Bodens vergeht also doppelt soviel Zeit wie bis zum Erreichen des Scheitelpunkts.

Übungsaufgabe 2.2 (Reichweite) *Nehmen wir an, dass die Ausgangshöhe $x_{0,2}$ die Höhe eines völlig flachen Erdbodens ist und wir unseren Flugkörper "nach oben" starten, dass also $v_{0,2} \geq 0$ gilt.*

Berechnen Sie, welche Entfernung der Flugkörper zurückgelegt hat, wenn er den Boden wieder erreicht, wenn also $x_2(t) = x_{0,2}$ gilt.

Finden Sie eine Anfangsgeschwindigkeit v_0 mit $\|v_0\|_2 = 1$ derart, dass die Reichweite maximal wird. Dabei bezeichnet $\|v_0\|_2 = \sqrt{v_{0,1}^2 + v_{0,2}^2}$ die Länge des Vektors v_0.

Interpretieren Sie das Ergebnis geometrisch: In welchem Winkel zum Erdboden muss der Flugkörper starten, um die maximale Reichweite zu erzielen?

Leider beschreibt die Gleichung (2.6) die Realität nur näherungsweise: In der Regel wird die *Reibung* des Flugkörpers an der ihn umgebenden Luft dazu führen, dass er abgebremst wird. Für nicht zu hohe Geschwindigkeiten lässt sich dieser Effekt modellieren, indem wir die Gleichung (2.6) um die durch die Reibung ausgeübte Kraft ergänzen und so zu

$$f(t) = f_0 - \beta v(t) \qquad \text{für alle } t \in \mathbb{R} \quad (2.8)$$

gelangen. Hier ist $\beta \in \mathbb{R}_{\geq 0}$ ein Koeffizient, der den Luftwiderstand beschreibt. Der im Vergleich zu (2.6) hinzugekommene Term $-\beta v(t)$ beschreibt, dass die Luftreibung eine Kraft verursacht, die in der der Geschwindigkeit entgegengesetzten Richtung wirkt und zu ihr proportional ist.

Mit diesem zusätzlichen Term verändert sich der Charakter unseres mathematischen Modells: Die Kraft ist nicht länger konstant, sondern hängt von der aktuellen Geschwindigkeit ab, denn es gilt

$$v'(t) = a(t) = \frac{1}{m} f(t) = \frac{1}{m}(f_0 - \beta v(t)) \qquad \text{für alle } t \in \mathbb{R}. \tag{2.9}$$

Da die Ableitung $v'(t)$ von $v(t)$ abhängt, lässt sich diese Gleichung nicht mehr einfach mit dem Hauptsatz auflösen, wir haben es mit einer *gewöhnlichen Differentialgleichung* zu tun, für die andere Lösungstechniken zum Einsatz kommen müssen.

Übungsaufgabe 2.3 (Analytische Lösung) *Im vorliegenden einfachen Fall können wir den Ansatz*

$$v(t) = c_1 + c_2 \exp(c_3 t) \qquad \textit{für alle } t \in \mathbb{R}$$

verwenden, um $c_1, c_2 \in \mathbb{R}^2$ und $c_3 \in \mathbb{R}$ so zu bestimmen, dass die Differentialgleichung (2.9) mit der Anfangsbedingung $v(0) = v_0$ *gilt.*

Dieser Ansatz funktioniert nur im Fall $\beta \neq 0$, allerdings kann man sich mit Hilfe der Taylor-Entwicklung der Exponentialfunktion überlegen, dass wir für $\beta \to 0$ wieder die bereits bekannte Lösung erhalten.

Da das finden einer geschlossenen Formel für die Lösung einer gewöhnlichen Differentialgleichung häufig schwierig oder sogar unmöglich ist, kommen in der Praxis *numerische Näherungsverfahren* zum Einsatz, die eine Approximation der Lösung ermitteln.

Ein besonders einfacher Ansatz besteht darin, die Ableitung durch einen Differenzenquotienten anzunähern, in dem lediglich Werte der Funktion auftreten, aber nicht mehr ihre Ableitung. Die Ableitung $g'(t)$ einer stetig differenzierbaren Funktion g wird beispielsweise durch den *Vorwärtsdifferenzenquotienten*

$$d_{+,\delta} g(t) := \frac{g(t+\delta) - g(t)}{\delta}, \tag{2.10a}$$

den *Rückwärtsdifferenzenquotienten*

$$d_{-,\delta} g(t) := \frac{g(t) - g(t-\delta)}{\delta} \tag{2.10b}$$

oder den *zentralen Differenzenquotienten*

$$d_{z,\delta} g(t) := \frac{g(t+\delta) - g(t-\delta)}{2\delta} \tag{2.10c}$$

angenähert, sofern die Schrittweite $\delta \in \mathbb{R}_{>0}$ klein genug gewählt ist.

Natürlich stellt sich die Frage, ob diese Differenzenquotienten sich überhaupt als Näherung der Ableitung eignen. Für die Antwort benötigen wir als Hilfsmittel den *Zwischenwertsatz* und den *Satz von Taylor*.

Erinnerung 2.4 (Zwischenwertsatz) *Sei $g : [a, b] \to \mathbb{R}$ eine stetige Funktion. Für jeden zwischen $g(a)$ und $g(b)$ liegenden Wert y (also $y \in [g(a), g(b)]$ falls $g(a) \leq g(b)$ oder $y \in [g(b), g(a)]$ ansonsten) existiert ein $x \in [a, b]$ mit $g(x) = y$.*

Erinnerung 2.5 (Satz von Taylor) *Sei $g : [a, b] \to \mathbb{R}$ eine k-mal stetig differenzierbare Funktion. Dann existieren $\eta_+, \eta_- \in [a, b]$ mit*

$$g(b) = \sum_{\nu=0}^{k-1} \frac{(b-a)^\nu}{\nu!} g^{(\nu)}(a) + \frac{(b-a)^k}{k!} g^{(k)}(\eta_+),$$
$$g(a) = \sum_{\nu=0}^{k-1} \frac{(a-b)^\nu}{\nu!} g^{(\nu)}(b) + \frac{(a-b)^k}{k!} g^{(k)}(\eta_-).$$

Diese Gleichungen können verwendet werden, um eine Funktion in der Nähe eines Punkts durch ein Polynom anzunähern.

Diese beiden Hilfsmittel ermöglichen es uns, relativ konkrete Gleichungen für den bei der Approximation der Ableitung durch Differenzenquotienten auftretenden Fehler anzugeben.

Lemma 2.6 (Differenzenquotienten) *Seien $t \in \mathbb{R}$ und $\delta \in \mathbb{R}_{>0}$ gegeben.*
Falls $g : [t, t + \delta] \to \mathbb{R}$ zweimal stetig differenzierbar ist, gilt

$$d_{+,\delta} g(t) = g'(t) + \frac{\delta}{2} g''(\eta_+) \tag{2.11a}$$

mit einem $\eta_+ \in [t, t + \delta]$.
Falls $g : [t - \delta, t] \to \mathbb{R}$ zweimal stetig differenzierbar ist, gilt

$$d_{-,\delta} g(t) = g'(t) + \frac{\delta}{2} g''(\eta_-) \tag{2.11b}$$

mit einem $\eta_- \in [t - \delta, t]$.
Falls $g : [t - \delta, t + \delta] \to \mathbb{R}$ dreimal stetig differenzierbar ist, gilt

$$d_{z,\delta} g(t) = g'(t) + \frac{\delta^2}{6} g'''(\eta_z) \tag{2.11c}$$

mit einem $\eta_z \in [t - \delta, t + \delta]$.

Beweis. Aus der zweifachen stetigen Differenzierbarkeit folgt mit dem Satz von Taylor die Existenz von $\eta_+ \in [t, t + \delta]$ und $\eta_- \in [t - \delta, t]$ mit

$$g(t + \delta) = g(t) + \delta g'(t) + \frac{\delta^2}{2} g''(\eta_+),$$
$$g(t - \delta) = g(t) - \delta g'(t) + \frac{\delta^2}{2} g''(\eta_-).$$

Für den ersten Differenzenquotienten folgt unmittelbar

$$\frac{g(t+\delta)-g(t)}{\delta} = \frac{\delta g'(t) + \frac{\delta^2}{2} g''(\eta_+)}{\delta} = g'(t) + \frac{\delta}{2} g''(\eta_+)$$

und für den zweiten

$$\frac{g(t)-g(t-\delta)}{\delta} = \frac{\delta g'(t) - \frac{\delta^2}{2} g''(\eta_-)}{\delta} = g'(t) - \frac{\delta}{2} g''(\eta_-).$$

Damit ist der Beweis für die ersten beiden Fälle auch schon abgeschlossen.

Im Fall des dritten Differenzenquotienten folgt mit dem Satz von Taylor die Existenz von (neuen) $\eta_+ \in [t, t+\delta]$ und $\eta_- \in [t-\delta, t]$ mit

$$\begin{aligned} g(t+\delta) &= g(t) + \delta g'(t) + \frac{\delta^2}{2} g''(t) + \frac{\delta^3}{6} g'''(\eta_+), \\ g(t-\delta) &= g(t) - \delta g'(t) + \frac{\delta^2}{2} g''(t) - \frac{\delta^3}{6} g'''(\eta_-), \end{aligned}$$

so dass wir für den dritten Quotienten

$$\frac{g(t+\delta)-g(t-\delta)}{2\delta} = \frac{2\delta g'(t) + \frac{\delta^3}{6}(g'''(\eta_+) + g'''(\eta_-))}{2\delta} = g'(t) + \frac{\delta^2}{6} \frac{g'''(\eta_+) + g'''(\eta_-)}{2}$$

erhalten. Der rechte Quotient ist das arithmetische Mittel der Zahlen $g'''(\eta_+)$ und $g'''(\eta_-)$, muss also insbesondere zwischen diesen Zahlen liegen. Damit lässt sich der *Zwischenwertsatz* anwenden, um ein $\eta_z \in [\eta_-, \eta_+] \subseteq [t-\delta, t+\delta]$ zu finden, das

$$\frac{g'''(\eta_+) + g'''(\eta_-)}{2} = g'''(\eta_z)$$

erfüllt, so dass wir insgesamt

$$\frac{g(t+\delta)-g(t-\delta)}{2\delta} = g'(t) + \frac{\delta^2}{6} g'''(\eta_z)$$

erhalten und fertig sind. ∎

Wir dürfen festhalten, dass die Differenzenquotienten wie gewünscht die Ableitung approximieren.

Wenden wir uns nun wieder der Differentialgleichung (2.9) zu. Wir ersetzen die Ableitung $v'(t)$ durch den Vorwärtsdifferenzenquotienten (2.10a) und erhalten

$$\begin{aligned} \frac{v(t+\delta)-v(t)}{\delta} &\approx \frac{1}{m}(f_0 - \beta v(t)), \\ v(t+\delta) - v(t) &\approx \frac{\delta}{m}(f_0 - \beta v(t)), \\ v(t+\delta) &\approx v(t) + \frac{\delta}{m}(f_0 - \beta v(t)). \end{aligned} \tag{2.12}$$

Diese „Gleichung“ können wir verwenden, um zu der zu einem Zeitpunkt t gegebenen Geschwindigkeit $v(t)$ eine Näherung der Geschwindigkeit zu einem Zeitpunkt $v(t+\delta)$ zu berechnen.

Aufgrund der Abschätzung (2.11a) dürfen wir erwarten, dass die Näherung wie δ gegen die exakte Lösung strebt. Für eine hohe Genauigkeit ist also eine kleine Schrittweite erforderlich.

Diese Voraussetzung können wir einfach erfüllen: Um die Lösung auf einem Intervall zu approximieren, ersetzen wir es durch diskrete Zeitpunkte t_i mit dem Abstand δ, indem wir

$$t_i := \delta i \qquad \text{für alle } i \in \{0, \dots, n\}$$

definieren. Auf jedem Teilintervall $[t_i, t_{i+1}]$ können wir unsere Approximation wegen $t_{i+1} - t_i = \delta$ anwenden und uns so der Lösung nähern. Insbesondere können wir eine Lösung in jedem beliebigen Punkt $t \in \mathbb{R}_{>0}$ berechnen, indem wir das Intervall $[0, t]$ in $k \in \mathbb{N}$ Teilintervalle zerlegen und $\delta = t/k$ setzen.

Unsere Aufgabe besteht nun darin, Näherungen $\tilde{v}(t_i)$ der Geschwindigkeit $v(t_i)$ für alle $i \in \{0, \dots, n\}$ zu berechnen. Für den Zeitpunkt $t_0 = 0$ können wir den gegebenen Anfangswert $v(t_0) = v(0) = v_0$ verwenden. Mit der Formel (2.12) können wir dann eine Näherung für den Zeitpunkt $t_1 = t_0 + \delta$ gewinnen, aus der sich wiederum eine für $t_2 = t_1 + \delta$ berechnen lässt.

Insgesamt gelangen wir zu dem *expliziten Euler-Verfahren* [3], das sich für unsere Anwendung wie folgt zusammenfassen lässt:

$$\tilde{v}(t_0) = v_0, \tag{2.13a}$$

$$\tilde{v}(t_{i+1}) = \tilde{v}(t_i) + \frac{\delta}{m}(f_0 - \beta\tilde{v}(t_i)) \qquad \text{für alle } i \in \{0, \dots, n-1\}. \tag{2.13b}$$

Es lässt sich offenbar sehr einfach implementieren, erfordert allerdings auch einen hohen Rechenaufwand, um eine brauchbare Genauigkeit zu erreichen: Für eine Halbierung des Fehlers muss die Anzahl der Zeitschritte verdoppelt werden.

Um die Flugbahn beschreiben zu können, brauchen wir natürlich nicht nur die Geschwindigkeit, sondern auch die Position des Flugkörpers. Indem wir den Differenzenquotienten auf (2.1) anwenden, erhalten wir

$$\frac{x(t+\delta) - x(t)}{\delta} \approx x'(t) = v(t),$$
$$x(t+\delta) \approx x(t) + \delta v(t)$$

und können die mit dem Euler-Verfahren berechneten Näherungswerte einsetzen, um zu

$$\tilde{x}(t_0) = x_0, \tag{2.14a}$$

$$\tilde{v}(t_0) = v_0, \tag{2.14b}$$

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_i) + \delta\tilde{v}(t_i), \tag{2.14c}$$

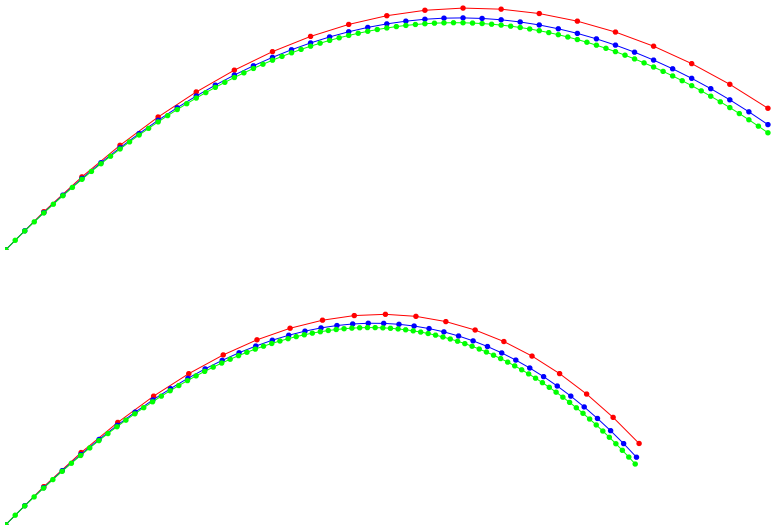

Abbildung 2.1: Simulation der Flugbahn ohne (oben) und mit (unten) Reibung per Euler-Verfahren mit 20, 40 und 80 Schritten

$$\tilde{v}(t_{i+1}) = \tilde{v}(t_i) + \frac{\delta}{m}(f_0 - \beta\tilde{v}(t_i)) \qquad \text{für alle } i \in \{0, \dots, n-1\} \tag{2.14d}$$

zu gelangen. Dieser Algorithmus berechnet gleichzeitig Näherungen für die Positionen und Geschwindigkeiten des Flugkörpers zu allen Zeitpunkten t_i. Ein Beispiel findet sich in Abbildung 2.1.

Übungsaufgabe 2.7 (Implizites Euler-Verfahren) *Wenn es ein* explizites *Euler-Verfahren gibt, liegt es nahe, nach einem* impliziten *Euler-Verfahren zu fragen. Es ergibt sich, wenn man für die Näherung der Ableitung den Rückwärtsdifferenzenquotienten (2.10b) verwendet, also*

$$\frac{v(t+\delta) - v(t)}{\delta} \approx v'(t+\delta) = \frac{1}{m}(f_0 - \beta v(t+\delta)).$$

Leiten Sie daraus einen Algorithmus her, der ausgehend von $\tilde{v}(t_0) = v_0$ eine Folge von Näherungen $\tilde{v}(t_i)$ berechnet.

Gibt es Schrittweiten δ, für die er nicht durchführbar ist?

2.3 Zeitschrittverfahren

Da uns gewöhnliche Differentialgleichungen noch häufiger begegnen werden, bietet es sich an, die zu ihrer Behandlung geeigneten Algorithmen so zu formulieren, dass sie sich möglichst allgemein anwenden lassen.

Definition 2.8 (Anfangswertproblem) *Sei $d \in \mathbb{N}$, sei $g : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ eine stetige Abbildung und $y_0 \in \mathbb{R}^d$.*

Eine stetig differenzierbare Funktion $y : \mathbb{R} \to \mathbb{R}^d$, die

$$y(0) = y_0, \qquad y'(t) = g(t, y(t)) \qquad \text{für alle } t \in \mathbb{R} \tag{2.15}$$

erfüllt, nennen wir eine Lösung des durch die rechte Seite *g und den* Anfangswert *y_0 gegebenen* Anfangswertproblems.

Beispielsweise können wir (2.9) als Anfangswertproblem identifizieren:

Beispiel 2.9 (Geschwindigkeit) *Wenn wir*

$$y_0 := v_0, \qquad y(t) := v(t), \qquad g(t, z) := \frac{1}{m}(f_0 - \beta z) \qquad \text{für alle } t \in \mathbb{R},\ z \in \mathbb{R}^2$$

definieren, folgt

$$\begin{aligned} &y(0) = v(0) = v_0 = y_0, \\ &y'(t) = v'(t) = \frac{1}{m}(f_0 - \beta v(t)) = g(t, v(t)) = g(t, y(t)) \qquad \text{für alle } t \in \mathbb{R}, \end{aligned}$$

also ist die Geschwindigkeit v gerade die Lösung des Anfangswertproblems mit der rechten Seite g und dem Anfangswert y_0.

Wir können allerdings auch die gleichzeitige Berechnung von Position und Geschwindigkeit in diese Form bringen, indem wir beide zu einem Vektor zusammenfassen:

Beispiel 2.10 (Position und Geschwindigkeit) *Wir setzen*

$$y_0 := \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}, \quad y(t) := \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}, \quad g(t, z) := \begin{pmatrix} z_2 \\ (f_0 - \beta z_2)/m \end{pmatrix} \qquad \begin{array}{r} \text{für alle } t \in \mathbb{R}, \\ z \in \mathbb{R}^2 \times \mathbb{R}^2 \end{array}$$

und stellen fest, dass

$$y'(t) = \begin{pmatrix} x'(t) \\ v'(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ (f_0 - \beta v(t))/m \end{pmatrix} = g(t, y(t)) \qquad \text{für alle } t \in \mathbb{R}$$

gelten muss. Also ist der Vektor aus Position und Geschwindigkeit gerade die Lösung des Anfangswertproblems mit der rechten Seite g und dem Anfangswert y_0.

Diese Abstraktion der Aufgabenstellung ist sehr nützlich, weil sie es uns ermöglicht, unsere Lösungsverfahren in einer Form zu formulieren, die sich auf *sämtliche* Anfangswertprobleme anwenden lässt.

Definition 2.11 (Explizites Euler-Verfahren) *Sei ein Anfangswertproblem mit einer rechten Seite g und einem Anfangswert y_0 gegeben. Sei $\delta \in \mathbb{R}_{>0}$. Die durch*

$$t_i := i\delta, \tag{2.16a}$$
$$\tilde{y}(t_0) := y_0, \tag{2.16b}$$
$$\tilde{y}(t_{i+1}) := \tilde{y}(t_i) + \delta g(t_i, \tilde{y}(t_i)) \qquad \text{für alle } i \in \mathbb{N}_0 \tag{2.16c}$$

gegebene Folge $\tilde{y}(t_0), \tilde{y}(t_1), \tilde{y}(t_2), \ldots$ nennen wir die durch das explizite Euler-Verfahren *[3] mit der Schrittweite δ berechnete* Näherungslösung *des Anfangswertproblems.*

Indem wir (2.16) mit den Beispielen 2.9 und 2.10 vergleichen, stellen wir fest, dass die uns bereits bekannten Gleichungen (2.13) und (2.14) in der Tat Umsetzungen des expliziten Euler-Verfahrens sind.

Unser Ziel ist es, *effizientere* Algorithmen für die Behandlung von Anfangswertproblemen zu entwickeln. Sowohl der Vorwärtsdifferenzenquotient (2.10a) als auch der Rückwärtsdifferenzenquotient (2.10b) weisen gemäß Lemma 2.6 einen Fehler der Größenordnung δ auf, so dass für eine Halbierung des Fehlers eine Verdoppelung der Schritte erforderlich ist, also auch eine Verdoppelung des Rechenaufwands.

Attraktiver ist der zentrale Differenzenquotient (2.10c), da sich bei ihm der Fehler wie δ^2 verhält: Eine Verdoppelung der Schritte führt zu einer Viertelung des Fehlers.

Da wir von einem Zeitpunkt t_i zu dem nachfolgenden Zeitpunkt t_{i+1} gelangen wollen, wenden wir (2.10c) auf die halbe Schrittweite $\delta/2$ und den Punkt $t_{i+1/2} := t_i + \delta/2$ zwischen beiden Zeitpunkten an und erhalten

$$\frac{y(t_{i+1}) - y(t_i)}{\delta} = \frac{y(t_{i+1/2} + \delta/2) - y(t_{i+1/2} - \delta/2)}{2\ \delta/2} \approx y'(t_{i+1/2}) = g(t_{i+1/2}, y(t_{i+1/2})),$$
$$y(t_{i+1}) \approx y(t_i) + \delta g(t_{i+1/2}, y(t_{i+1/2})).$$

Der einzige Unterschied zu dem Euler-Verfahren besteht also darin, dass wir g im Zwischenpunkt $t_{i+1/2}$ statt in t_i auswerten.

Leider steht uns $y(t_{i+1/2})$ nicht zur Verfügung, so dass wir aus dieser Formel nicht unmittelbar ein praktisch durchführbares Verfahren gewinnen können. Allerdings können wir den Wert durch eine Näherung ersetzen: Wie im Euler-Verfahren verwenden wir

$$y(t_{i+1/2}) = y(t_i + \delta/2) \approx y(t_i) + \frac{\delta}{2} g(t_i, y(t_i))$$

und gelangen so zu

$$\begin{aligned} y(t_{i+1}) &\approx y(t_i) + \delta g(t_i + \delta/2, y(t_i + \delta/2)) \\ &\approx y(t_i) + \delta g\left(t_i + \delta/2, y(t_i) + \frac{\delta}{2} g(t_i, y(t_i))\right). \end{aligned}$$

Dieser Ansatz ist offenbar deutlich aufwendiger als das Euler-Verfahren, beispielsweise sind zwei Auswertungen der rechten Seite g erforderlich, aber es lässt sich beweisen, dass er auch deutlich weniger Schritte benötigt, um eine gewisse Genauigkeit zu erreichen,

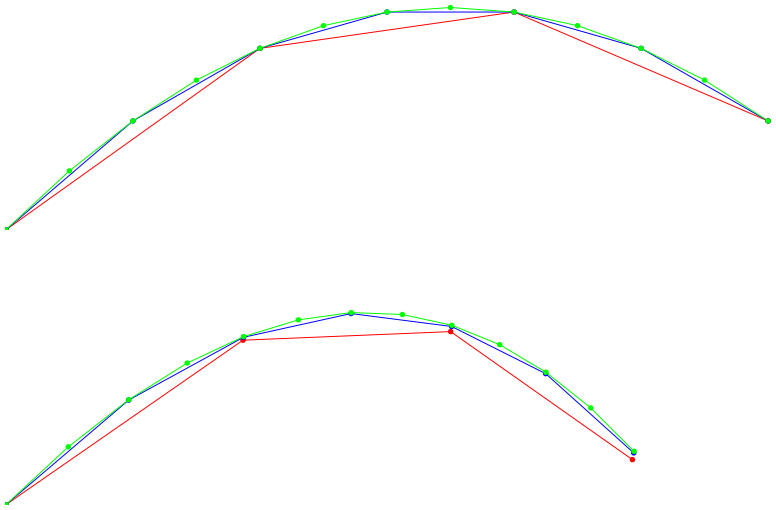

Abbildung 2.2: Simulation der Flugbahn ohne (oben) und mit (unten) Reibung per Heun-Verfahren mit 3, 6 und 12 Schritten

da der Fehler wie δ^2 statt wie δ fällt: Für eine gegebene Genauigkeit von ϵ benötigt das Euler-Verfahren $\mathcal{O}(1/\epsilon)$ Schritte, während das neue Verfahren mit $\mathcal{O}(1/\sqrt{\epsilon})$ Schritten auskommt.

Definition 2.12 (Heun-Verfahren) *Sei ein Anfangswertproblem mit einer rechten Seite g und einem Anfangswert y_0 gegeben. Sei $\delta \in \mathbb{R}_{>0}$. Die durch*

$$t_i := i\delta, \tag{2.17a}$$

$$\tilde{y}(t_0) := y_0, \tag{2.17b}$$

$$\tilde{y}(t_{i+1/2}) := \tilde{y}(t_i) + \frac{\delta}{2} g(t_i, \tilde{y}(t_i)), \tag{2.17c}$$

$$\tilde{y}(t_{i+1}) := \tilde{y}(t_i) + \delta g(t_i + \delta/2, \tilde{y}(t_{i+1/2})) \qquad \text{für alle } i \in \mathbb{N}_0 \tag{2.17d}$$

gegebene Folge $\tilde{y}(t_0), \tilde{y}(t_1), \tilde{y}(t_2), \ldots$ nennen wir die durch das Heun-Verfahren *[5] mit der Schrittweite δ berechnete* Näherungslösung *des Anfangswertproblems.*

Ein Beispiel für die Anwendung des Heun-Verfahrens auf die Flugbahnsimulation ist in Abbildung 2.2 zu finden. Schon mit lediglich drei Schritten erreicht das Heun-Verfahren eine sehr gute Genauigkeit, sechs Schritte des Heun-Verfahrens führen zu einer Genauigkeit, für die das Euler-Verfahren ungefähr achtzig Schritte benötigt. Wenn man davon ausgeht, dass ein Heun-Schritt ungefähr den doppelten Rechenaufwand eines Euler-Schritts erfordert, entspricht das einer Einsparung von 85% der Rechenzeit.

2.4 Federn

Als nächstes Beispiel untersuchen wir ein physikalisches System, in dem die Veränderung der Geschwindigkeit von der Position des beweglichen Körpers abhängt, nicht von der Geschwindigkeit selbst: Das Federpendel.

Es besteht aus einem Körper, der an einem festen Punkt mittels einer Feder befestigt ist, die eine Kraft ausübt, die von seiner Entfernung von der „Aufhängung“ bestimmt wird. Um unsere Formeln übersichtlich zu halten verwenden wir den Nullpunkt als Aufhängung.

Ein besonders einfacher und trotzdem in vielen Fällen realistischer Zusammenhang ist das *Federgesetz von Hooke* [6], das davon ausgeht, dass die wirkende Kraft proportional zu der Auslenkung ist und in Richtung des Nullpunkts wirkt. Diese Regel lässt sich durch die Formel

$$f(t) = -cx(t) \qquad \text{für alle } t \in \mathbb{R} \tag{2.18}$$

ausdrücken, die die Kraft $f(t)$ mit der Auslenkung $x(t)$ in Beziehung setzt. Der Proportionalitätsfaktor c wird als *Federkonstante* bezeichnet und beschreibt, wieviel Widerstand die Feder leistet, wenn sie auseinander gezogen wird.

In Kombination mit (2.3) und (2.2) erhalten wir

$$v'(t) = a(t) = \frac{1}{m} f(t) = -\frac{c}{m} x(t) \qquad \text{für alle } t \in \mathbb{R}.$$

Zusammen mit (2.1) und den Anfangsbedingungen $x(0) = x_0$ und $v(0) = v_0$ ergibt sich das Anfangswertproblem

$$x(0) = x_0, \qquad v(0) = v_0, \tag{2.19a}$$

$$x'(t) = v(t), \qquad v'(t) = -\frac{c}{m} x(t) \qquad \text{für alle } t \in \mathbb{R}, \tag{2.19b}$$

das die Bewegung des Körpers beschreibt. Indem wir wieder Position und Geschwindigkeit in Vektoren zusammenfassen, also

$$y_0 := \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}, \qquad y(t) := \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \qquad \text{für alle } t \in \mathbb{R}$$

setzen, gelangen wir zu

$$y'(t) = \begin{pmatrix} x'(t) \\ v'(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -c/m\ x(t) \end{pmatrix} = g(t, y(t)) \qquad \text{für alle } t \in \mathbb{R},$$

mit der Funktion

$$g(t, z) := \begin{pmatrix} z_2 \\ -c/m\ z_1 \end{pmatrix} \qquad \text{für alle } t \in \mathbb{R},\ z \in \mathbb{R}^2,$$

wir haben es also in der Tat mit einem Anfangswertproblem gemäß der Definition 2.8 zu tun.

Da wir das Euler- und das Heun-Verfahren für allgemeine Anfangswertprobleme formuliert haben, können wir beide unmittelbar auf (2.19) anwenden, um die Bewegung des Pendels zu simulieren.

Wir können uns allerdings auch die Eigenschaften des Gleichungssystems zunutze machen, um einen schnelleren Algorithmus zu erhalten: Wie schon im Fall des Heun-Verfahrens wenden wir den zentralen Differenzenquotienten mit der Schrittweite $\delta/2$ an, um die Ableitungen in (2.19) zu approximieren, und gelangen so zu

$$\frac{x(t+\delta)-x(t)}{\delta} \approx v(t+\delta/2), \qquad \frac{v(t+\delta)-v(t)}{\delta} \approx -\frac{c}{m}x(t+\delta/2) \qquad \text{für alle } t \in \mathbb{R}.$$

Für die Berechnung von $x(t+\delta)$ benötigen wir $v(t+\delta/2)$, während wir für die Berechnung von $v(t+\delta)$ gerade $x(t+\delta/2)$ brauchen. Die nötigen Werte sind also jeweils gerade um $\delta/2$ gegeneinander versetzt.

Diese Eigenschaft machen wir uns zunutze, indem wir beschließen, dass wir die Position weiterhin in den Zeitpunkten $t_i = i\delta$ für $i \in \mathbb{N}_0$ berechnen wollen, die Geschwindigkeit dagegen in den Zeitpunkten $t_{i+1/2} = (i+1/2)\delta$. Die dafür benötigte erste Näherung der Geschwindigkeit in $t_{1/2}$ können wir mit dem Euler-Verfahren gemäß

$$\tilde{v}(t_{1/2}) = v_0 - \frac{\delta c}{2m}x_0$$

gewinnen und gelangen so zu dem folgenden effizienten Verfahren:

$$\begin{aligned}
t_i &:= i\delta, \quad t_{i+1/2} := (i+1/2)\delta, \\
\tilde{x}(t_0) &:= x_0, & \tilde{v}(t_{1/2}) &:= v_0 - \frac{\delta c}{2m}x_0, \\
\tilde{x}(t_{i+1}) &:= \tilde{x}(t_i) + \delta\tilde{v}(t_{i+1/2}), \\
\tilde{v}(t_{i+3/2}) &:= \tilde{v}(t_{i+1/2}) - \frac{\delta c}{m}\tilde{x}(t_{i+1}) & &\text{für alle } i \in \mathbb{N}_0
\end{aligned}$$

Es lässt sich offenbar auch für allgemeinere Anfangswertprobleme formulieren, bei denen zwei Variablen voneinander abhängen:

Definition 2.13 (Leapfrog-Verfahren) *Wir betrachten ein Anfangswertproblem, das in der Form*

$$\begin{aligned}
y(0) &= y_0, & z(0) &= z_0, \\
y'(t) &= f(t, z(t)), & z'(t) &= g(t, y(t)) & &\textit{für alle } t \in \mathbb{R}
\end{aligned}$$

gegeben ist.

Sei $\delta \in \mathbb{R}_{>0}$. Die durch

$$t_i := i\delta, \quad t_{i+1/2} := (i+1/2)\delta, \tag{2.20a}$$

$$\tilde{y}(t_0) := y_0, \quad \tilde{z}(t_{1/2}) := z_0 + (\delta/2)\, g(t_0, \tilde{y}(t_0)), \tag{2.20b}$$

$$\tilde{y}(t_{i+1}) := \tilde{y}(t_i) + \delta f(t_{i+1/2}, \tilde{z}(t_{i+1/2})), \tag{2.20c}$$

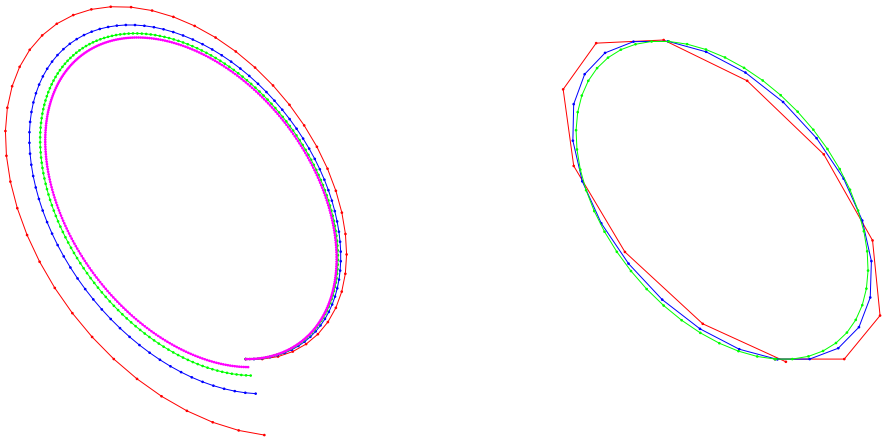

Abbildung 2.3: Simulation des Federpendels per Euler-Verfahren mit 50, 100 und 200 Schritten (links) und per Leapfrog-Verfahren mit 12, 24 und 48 Schritten (rechts). Die exakte Lösung ist eine geschlossene Ellipse.

$$\tilde{z}(t_{i+3/2}) := \tilde{z}(t_{i+1/2}) + \delta g(t_{i+1}, \tilde{y}(t_{i+1})) \qquad \textit{für alle } i \in \mathbb{N}_0 \qquad (2.20\text{d})$$

definierten Folgen $\tilde{y}(t_0), \tilde{y}(t_1), \ldots$ *und* $\tilde{z}(t_{1/2}), \tilde{z}(t_{3/2}), \ldots$ *nennen wir die durch das* Leapfrog-Verfahren *[7] mit der Schrittweite* δ *berechneten Näherungslösungen des Anfangswertproblems (2.19).*

Der Name des Verfahrens ist dadurch motiviert, dass Position und Geschwindigkeit jeweils abwechseln einen Zeitpunkt überspringen. Das Kinderspiel „Bockspringen“, bei dem die Spieler demselben Muster folgen, wird im Englischen als *leapfrog* bezeichnet.

Wenn wir uns die pro Zeitschritt auszuführenden Berechnungen anschauen, erkennen wir, dass der Rechenaufwand dem des Euler-Verfahrens entspricht. Da das Leapfrog-Verfahren aber auf dem *zentralen* Differenzenquotienten beruht, konvergiert es wie δ^2, also mit derselben Geschwindigkeit wie das Heun-Verfahren und damit deutlich schneller als der Euler-Algorithmus.

Übungsaufgabe 2.14 (Exakte Lösung) *Auch für das Anfangswertproblem (2.19) lässt sich eine exakte Lösung angeben: Ausgehend von dem Ansatz*

$$x(t) = c_1 \sin(c_3 t) + c_2 \cos(c_3 t) \qquad \textit{für alle } t \in \mathbb{R}$$

lassen sich $c_1, c_2 \in \mathbb{R}^2$ *und* $c_3 \in \mathbb{R}$ *so bestimmen, dass alle Gleichungen erfüllt sind.*

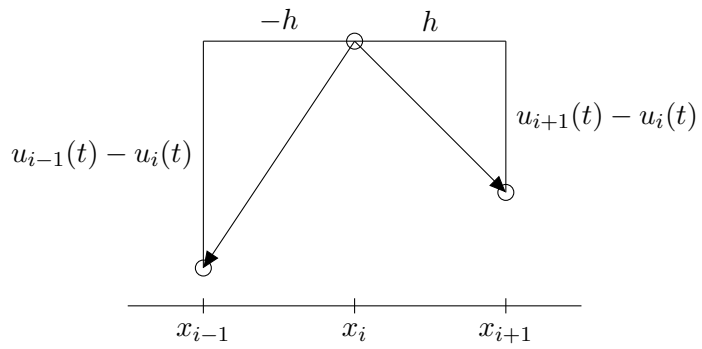

Abbildung 2.4: Auf die i-te Punktmasse wirkende Kräfte.

2.5 Wellen

Bisher haben wir uns lediglich mit Systemen beschäftigt, in denen sich nur ein einziger Körper bewegte. Nun untersuchen wir ein System mit sehr vielen beweglichen Körpern: Wir approximieren eine zwischen zwei Punkten eingespannte Saite durch n Punktmassen, die miteinander durch Federn gekoppelt sind.

Der Einfachheit halber gehen wir davon aus, dass diese Massen im Ruhezustand äquidistant, also in regelmäßigen Abständen, auf dem Intervall $[0, 1]$ verteilt liegen. Die Positionen sind dann durch

$$h := \frac{1}{n+1}, \qquad x_i := hi \qquad \text{für alle } i \in \{0, \dots, n+1\}$$

gegeben, wobei die Punkte $x_0 = 0$ und $x_{n+1} = 1$ fest eingespannt sind, sich also nicht bewegen können.

Wir interessieren uns nur für die in einem Zeitpunkt $t \in \mathbb{R}$ vorliegende Auslenkung nach oben (oder bei negativem Vorzeichen nach unten), die wir mit $u_i(t) \in \mathbb{R}$ bezeichnen.

Die i-te Punktemasse soll durch Federn mit den benachbarten Massen $i-1$ und $i+1$ verbunden sein. Nach (2.18) übt dann der linke Nachbar eine Kraft von

$$c_i \begin{pmatrix} x_{i-1} - x_i \\ u_{i-1}(t) - u_i(t) \end{pmatrix} = c_i \begin{pmatrix} -h \\ u_{i-1}(t) - u_i(t) \end{pmatrix}$$

aus, während der rechte Nachbar für eine Kraft von

$$c_i \begin{pmatrix} x_{i+1} - x_i \\ u_{i+1}(t) - u_i(t) \end{pmatrix} = c_i \begin{pmatrix} h \\ u_{i+1}(t) - u_i(t) \end{pmatrix}$$

verantwortlich ist. Hierbei bezeichnet c_i jeweils die Federkonstante der beiden mit der i-ten Masse verbundenen Federn.

Die Kräfte summieren sich, so dass insgesamt auf die i-te Masse die Kraft

$$f_i(t) = c_i \begin{pmatrix} 0 \\ u_{i+1}(t) - 2u_i(t) + u_{i-1}(t) \end{pmatrix} \qquad \text{für alle } t \in \mathbb{R},\ i \in \{1, \dots, n\} \tag{2.21}$$

wirkt. Da keine Kraft in horizontaler Richtung wirkt, erfolgt auch keine Beschleunigung in dieser Richtung, so dass die horizontalen Abstände der Punkte immer unverändert bleiben und deshalb von uns nicht weiter betrachtet werden müssen.

Die von einer Feder ausgeübte Kraft hängt von der *relativen* Längenveränderung der Feder ab: Eine einen Meter lange Feder um einen Zentimeter auszulenken ist leichter, als dasselbe bei einer nur einen Zentimeter langen Feder zu tun, denn im ersten Fall geht es um eine Auslenkung von einem Prozent der Federlänge, im zweiten Fall um hundert Prozent. Wenn wir mit c die Federkonstante einer Feder der Länge 1 bezeichnen, erhalten wir für Federn der Länge h eine Federkonstante von

$$c_i = \frac{c}{h} \qquad \text{für alle } i \in \{1, \dots, n\}.$$

Um die Kraft mit (2.3) in eine Beschleunigung umrechnen zu können, müssen wir die Masse der einzelnen Punkte kennen. Für unser Modell genügt es, die Gesamtmasse m der Saite gleichmäßig auf alle Punkte zu verteilen, wobei der linke und rechte Randpunkt nur „halb“ gezählt werden, so dass sich

$$m_i = \frac{m}{n+1} = mh \qquad \text{für alle } i \in \{1, \dots, n\}$$

ergibt. Indem wir (2.21) in (2.3) einsetzen, erhalten wir für die (vertikale) Beschleunigung des i-ten Punkts die Gleichung

$$a_i(t) = \frac{c}{mh^2}(u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)) \qquad \text{für alle } t \in \mathbb{R},\ i \in \{1, \dots, n\}.$$

Wenn wir die (ebenfalls vertikale) Geschwindigkeit des i-ten Punkts mit $v_i(t)$ bezeichnen, ergibt sich mit (2.1) und (2.2) schließlich das System

$$u_i'(t) = v_i(t), \quad v_i'(t) = \frac{c}{m}\frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{h^2} \qquad \text{für alle } t \in \mathbb{R},\ i \in \{1, \dots, n\}. \tag{2.22}$$

Es bietet sich an, die Auslenkungen und Geschwindigkeiten zu Vektoren

$$u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{pmatrix}, \qquad v(t) = \begin{pmatrix} v_1(t) \\ \vdots \\ v_n(t) \end{pmatrix} \qquad \text{für alle } t \in \mathbb{R}$$

zusammenzufassen und die Interaktion der einzelnen Punkte durch die Matrix

$$L := \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}$$

auszudrücken, so dass wir kurz

$$u'(t) = v(t), \qquad v'(t) = -\frac{c}{m}Lu(t) \qquad \text{für alle } t \in \mathbb{R} \tag{2.23}$$

schreiben können. In dieser Form erinnert uns das System sehr an das zu dem Federpendel gehörende System (2.19b), so dass es sich anbietet, demselben Ansatz zu benutzen.

Indem wir das Leapfrog-Verfahren mit den Anfangswerten $u(0) = u_0$ und $v(0) = v_0$ auf (2.23) anwenden, erhalten wir

$$\begin{aligned} \tilde{u}(t_0) &:= u_0, & \tilde{v}(t_{1/2}) &= v_0 - \frac{\delta c}{2m} L\tilde{u}(t_0), \\ \tilde{u}(t_{i+1}) &:= \tilde{u}(t_i) + \delta\tilde{v}(t_{i+1/2}), & & \\ \tilde{v}(t_{i+3/2}) &:= \tilde{v}(t_{i+1/2}) - \frac{\delta c}{m} L\tilde{u}(t_{i+1}) & & \text{für alle } i \in \mathbb{N}_0. \end{aligned}$$

Bei der Implementierung dieses Ansatzes können wir davon profitieren, dass $\tilde{v}(t_{i+1/2})$ nur einmal für die Berechnung von $\tilde{u}(t_{i+1})$ und $\tilde{u}(t_{i+1})$ wiederum nur einmal für die Berechung von $\tilde{v}(t_{i+3/2})$ verwendet wird, so dass wir die zu dem jeweils vorangehenden Zeitpunkt gehörenden Vektoren mit den aktuellen überschreiben können und keinen zusätzlichen Speicherplatz benötigen.

Bemerkung 2.15 (Schwachbesetzte Matrizen) *Bei der Multiplikation der Matrix L mit den Vektoren $\tilde{u}(t_i)$ sollten wir ausnutzen, dass fast alle Koeffizienten dieser Matrizen gleich null sind und deshalb keinen Beitrag zu dem Ergebnis leisten. Derartige Matrizen nennt man* schwachbesetzt.

Um effizient mit ihnen zu arbeiten, empfiehlt es sich dringend, nur für die von null verschiedenen Koeffizienten Arbeit zu leisten, beispielsweise indem wir für die Auswertung von $L\tilde{u}(t_i)$ auf die Formel (2.22) zurückgreifen. Mit dieser Vorgehensweise genügen $\mathcal{O}(n)$ Operationen für die Durchführung eines Zeitschritts.

Übungsaufgabe 2.16 (Wellengleichung) *Sei $g : [t-h, t+h] \to \mathbb{R}$ eine viermal stetig differenzierbare Funktion. Beweisen Sie, dass*

$$\frac{g(t+h) - 2g(t) + g(t-h)}{h^2} = g''(t) + \frac{h^2}{12} g^{(4)}(\eta)$$

für ein $\eta \in [t-h, t+h]$ gilt.

Indem wir in einem festen Punkt $x \in (0,1)$ in (2.22) die Ortsschrittweite h gegen null gehen lassen, erhalten wir die Wellengleichung

$$\frac{\partial u}{\partial t}(x,t) = v(x,t), \qquad \frac{\partial v}{\partial t}(x,t) = \frac{c}{m}\frac{\partial^2 u}{\partial x^2}(x,t) \qquad \text{für alle } x \in (0,1),\ t \in \mathbb{R}.$$

Die Lösungen dieser Differentialgleichung können deshalb als Grenzwert der Lösungen der Gleichungen (2.23) für $h \to 0$ interpretiert werden. Die Matrix $L \in \mathbb{R}^{n\times n}$ spielt dann die Rolle einer Näherung der zweiten Ableitung nach x.

Unsere Methode lässt sich von einer eingespannten Saite auch auf eine eingespannte Membran übertragen, also von einer eindimensionalen auf eine zweidimensionale Geometrie. Im Fall der schwingenden Saite haben wir die Saite durch eine Reihe von Punktmassen angenähert, die sich horizontal an den Positionen x_i befinden und nach oben oder

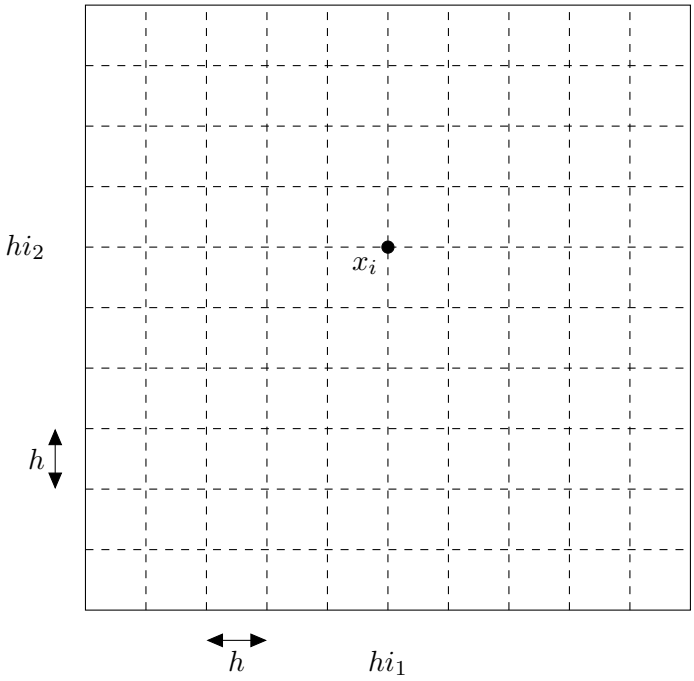

Abbildung 2.5: Zweidimensionales Punktgitter für die Simulation einer sich auf einer Membran ausbreitenden Welle. Verwendet wird $n = 9$, markiert ist der Punkt x_i mit $i = (5, 6)$.

unten um $u_i(t)$ ausgelenkt werden. Um eine zweidimensionale Membran zu beschreiben, können wir analog vorgehen, indem wir die Punkte x_i nun auf der Grundfläche der Membran anordnen und wieder Auslenkungen $u_i(t)$ betrachten.

Der Einfachheit halber betrachten wir hier nur den Fall einer quadratischen Grundfläche $[0, 1] \times [0, 1]$, auf der wir wie gehabt Punktmassen verteilen. Aufgrund der zweidimensionalen Geometrie empfiehlt es sich, auch die Punktmassen mit einer zweidimensionalen Indexmenge zu versehen:

$$h := \frac{1}{n+1}, \qquad x_i := \begin{pmatrix} hi_1 \\ hi_2 \end{pmatrix} \qquad \text{für alle } i = (i_1, i_2) \text{ mit } i_1, i_2 \in \{0, \ldots, n+1\}.$$

Jeder Index i ist nun also ein Paar (i_1, i_2) von Zahlen, die seine Position innerhalb der zweidimensionalen Grundfläche beschreiben. Die Auslenkung des i-ten Punkts bezeichnen wir wieder mit $u_i(t)$.

In unserer eindimensionalen Konstruktion war der i-te Punkt mit seinem linken und rechten Nachbarn durch Federn verbunden. In der zweidimensionalen Konstruktion betrachten wir stattdessen östliche, westliche, südliche und nördliche Nachbarn. Die Randpunkte mit $i_1 = 0$, $i_1 = n + 1$, $i_2 = 0$ oder $i_2 = n + 1$ werden wieder festgehalten,

beweglich sind nur die Punkte zu den Indizes in der Menge

$$\mathcal{I} := \{1, \dots, n\} \times \{1, \dots, n\}.$$

Der Punkt $i = (i_1, i_2) \in \mathcal{I}$ ist mit seinem östlichen Nachbarn $(i_1 - 1, i_2)$, seinem westlichen Nachbarn $(i_1 + 1, i_2)$, seinem südlichen Nachbarn $(i_1, i_2 - 1)$ und seinem nördlichen Nachbarn $(i_1, i_2 + 1)$ verbunden. Die Kräfte sind dabei durch

$$c_i \begin{pmatrix} -h \\ 0 \\ u_{i_1-1,i_2}(t) - u_{i_1,i_2}(t) \end{pmatrix}, \qquad c_i \begin{pmatrix} h \\ 0 \\ u_{i_1+1,i_2}(t) - u_{i_1,i_2}(t) \end{pmatrix},$$
$$c_i \begin{pmatrix} 0 \\ -h \\ u_{i_1,i_2-1}(t) - u_{i_1,i_2}(t) \end{pmatrix}, \qquad c_i \begin{pmatrix} 0 \\ h \\ u_{i_1,i_2+1}(t) - u_{i_1,i_2}(t) \end{pmatrix}$$

gegeben und addieren sich zu der Gesamtkraft

$$f_i(t) = c_i \begin{pmatrix} 0 \\ 0 \\ u_{i_1+1,i_2}(t) + u_{i_1-1,i_2}(t) + u_{i_1,i_2+1} + u_{i_1,i_2-1} - 4u_{i_1,i_2} \end{pmatrix}$$
$$\text{für alle } t \in \mathbb{R},\ i \in \mathcal{I}$$

Wie im eindimensionalen Fall wirken keine Kräfte innerhalb der Grundfläche, so dass Auslenkungen ausschließlich nach oben und unten erfolgen. Analog zu (2.22) erhalten wir

$$u_i'(t) = v_i(t), \quad v_i'(t) = \frac{c}{m} \frac{u_{i_1+1,i_2}(t) + u_{i_1-1,i_2}(t) + u_{i_1,i_2+1}(t) + u_{i_1,i_2-1}(t) - 4u_{i_1,i_2}(t)}{h^2}$$
$$\text{für alle } t \in \mathbb{R},\ i = (i_1, i_2) \in \mathcal{I}.$$

Da die Indizes i nun nicht mehr natürliche Zahlen, sondern Elemente der allgemeineren Indexmenge $\mathcal{I}$ sind, müssen wir auch Vektoren und Matrizen auf dieser allgemeineren Indexmenge verwenden: Ein Vektor $w \in \mathbb{R}^{\mathcal{I}}$ ordnet jedem Index $i \in \mathcal{I}$ eine Zahl $w_i \in \mathbb{R}$ zu, und eine Matrix $A \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ ordnet jedem Zeilenindex $i \in \mathcal{I}$ und jedem Spaltenindex $j \in \mathcal{I}$ eine Zahl a_{ij} zu. Wir definieren für alle $t \in \mathbb{R}$ die Vektoren $u(t), v(t) \in \mathbb{R}^{\mathcal{I}}$ durch

$$(u(t))_i = u_i(t), \qquad (v(t))_i = v_i(t) \qquad \text{für alle } t \in \mathbb{R},\ i \in \mathcal{I}$$

und führen die Matrix $L \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ durch

$$l_{ij} = \begin{cases} 4/h^2 & \text{falls } i_1 = j_1,\ i_2 = j_2, \\ -1/h^2 & \text{falls } |i_1 - j_1| = 1,\ i_2 = j_2, \\ -1/h^2 & \text{falls } i_1 = j_1,\ |i_2 - j_2| = 1, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i = (i_1, i_2), j = (j_1, j_2) \in \mathcal{I}$$

ein, um zu der kurzen Schreibweise

$$u'(t) = v(t), \qquad v'(t) = -\frac{c}{m} Lu(t) \qquad \text{für alle } t \in \mathbb{R} \tag{2.24}$$

zu gelangen, die das zweidimensionale Gegenstück des Systems (2.23) darstellt.

Wie im eindimensionalen Fall können wir mit dem Leapfrog-Verfahren eine Näherungslösung gewinnen, und wie im eindimensionalen Fall ist es wichtig, bei der Implementierung darauf zu achten, dass nur die von null verschiedenen Einträge der Matrix L bei der Berechnung des Produkt $Lu(t)$ berücksichtigt werden, um den Rechenaufwand möglichst gering zu halten.

Bemerkung 2.17 (Wellengleichung) *Indem wir auf die Übungsaufgabe 2.16 zurückgreifen können wir zeigen, dass sich aus (2.24) für $h \to 0$ die zweidimensionale Wellengleichung*

$$\frac{\partial u}{\partial t}(x,t) = v(x,t), \quad \frac{\partial v}{\partial t}(x,t) = \frac{c}{m}\left(\frac{\partial^2 u}{\partial x_1^2}(x,t) + \frac{\partial^2 u}{\partial x_2^2}(x,t)\right) \quad \textit{für alle } x \in (0,1)^2,\ t \in \mathbb{R}$$

ergibt. Die zweite Ableitung in x-Richtung wird also durch die Summe der partiellen Ableitungen in x_1- und x_2-Richtung ersetzt.

3 Nicht-lokale Kraftfelder

Bei dem von uns bereits als Beispiel untersuchten Federpendel hängt die von der Feder ausgeübte Kraft von der aktuellen Auslenkung des Pendels ab. Das Federgesetz stellt abstrakt gesehen eine Abbildung dar, die jedem Ort, an dem sich die pendelnde Masse gerade aufhält, die Kraft zuordnet, die von der Feder ausgeübt wird. Derartige Abbildungen bezeichnet man als *Kraftfelder*.

Kraftfelder werden beispielsweise verwendet, um in der Physik Phänomene wie die (nicht-relativistische) Gravitation oder die elektrostatische Anziehung oder Abstößung zu beschreiben.

Das von mehreren Massen hervorgerufene Gravitationsfeld ergibt sich als Summe der Felder der einzelnen Massen und lässt sich gut mit den schon diskutierten Zeitschrittverfahren behandeln. Wesentlich schwieriger wird es, wenn sehr viele Massen miteinander wechselwirken und die naive Berechnung des Kraftfelds einen zu hohen Rechenaufwand erfordern würde. In diesem Kapitel werden wir Techniken behandeln, mit denen sich auch das Gravitationsfeld von Millionen von Massen effizient berechnen lässt.

3.1 Gravitation

Wir untersuchen die Gravitationskraft, die von einer Masse m_1 an einem Punkt x_1 auf eine zweite Masse m_2 an einem Punkt x_2 ausgeübt wird. Wir lassen dabei relativistische Effekte außer Acht und verwenden das auf Newton [7] zurückgehende Gravitationsgesetz, das die Kraft als

$$f = \gamma m_1 m_2 \frac{x_1 - x_2}{\|x_1 - x_2\|_2^3} \tag{3.1}$$

ansetzt. Hier ist $\gamma \in \mathbb{R}_{>0}$ eine astrophysikalische Konstante, während $\|x_1 - x_2\|_2$ den euklidischen Abstand der Punkte x_1 und x_2 angibt. Er wird durch die *euklidische Norm*

$$\|\cdot\|_2 : \mathbb{R}^d \to \mathbb{R}^d, \qquad z \mapsto \left(\sum_{i=1}^{d} |z_i|^2 \right)^{1/2}, \tag{3.2}$$

gemessen, die jedem Vektor im d-dimensionalen Raum seine Länge zuordnet.

Übungsaufgabe 3.1 (Erdbeschleunigung) *Wir können eine Beziehung zu der in (2.6) verwendeten Erdbeschleunigung herstellen: Die Gravitationskonstante γ beträgt ungefähr $6,674 \times 10^{-11}\ Nm^2/kg^2$, die Masse der Erde ungefähr $5,974 \times 10^{24}\ kg$.*

Wir interessieren uns für die von der Erde auf eine Masse $m_2 = 1kg$ ausgeübte Gravitationskraft. Der Einfachheit halber wählen wir den Erdmittelpunkt als Nullpunkt, setzen also $x_1 = 0\,km$ und $m_1 = 5,974 \times 10^{24}\ kg$.

Berechnen Sie die Taylor-Entwicklung erster Ordnung der in (3.1) gegebenen Gravitationskraft als Funktion von x_2. Als Entwicklungspunkt soll dabei ein Punkt mit dem Abstand $\hat{x}_2 = 6\,350\,km$ zum Erdmittelpunkt dienen, das entspricht ungefähr dem Erdradius.

Damit erhalten wir eine Näherung der Kraft in der Nähe der Erdoberfläche. Vergleichen Sie Ihr Ergebnis mit (2.6), geben Sie den daraus folgenden Näherungswert für g an, und schätzen Sie ab, wie genau die Näherung für $x_2 \in [6\,300\,km, 6\,400\,km]$ ist.

Wir interessieren uns für das *Mehrkörperproblem*, also für die Bewegung mehrerer Körper, die einander mittels der Gravitationskraft beeinflussen. Bei der Simulation $n \in \mathbb{N}$ Körpern bezeichnen wir die Position des i-ten Körpers zu einem Zeitpunkt $t \in \mathbb{R}$ mit $x_i(t)$ und seine Masse mit m_i. Auf diesen Körper wird durch den j-ten Körper (mit $j \neq i$) gemäß (3.1) die Kraft

$$f_{ij}(t) = \gamma m_i m_j \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|_2^3} \qquad \text{für alle } t \in \mathbb{R}$$

ausgeübt. Die von *sämtlichen* Körpern auf den i-ten Körper ausgeübten Kräfte summieren sich auf, so dass insgesamt die Kraft

$$f_i(t) = \gamma m_i \sum_{\substack{j=1 \\ j \neq i}}^{n} m_j \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|_2^3} \qquad \text{für alle } t \in \mathbb{R} \tag{3.3}$$

wirkt. Nun können wir wie bisher vorgehen: Durch die Kraft wird der i-te Körper beschleunigt, seine Beschleunigung ergibt sich dank (2.3) als

$$a_i(t) = \frac{1}{m_i} f_i(t) = \gamma \sum_{\substack{j=1 \\ j \neq i}}^{n} m_j \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|_2^3} \qquad \text{für alle } t \in \mathbb{R}.$$

Für die Geschwindigkeit $v_i(t)$ und die Position dieses Körpers gelten gemäß (2.2) und (2.1) gerade

$$v_i'(t) = a_i(t), \qquad x_i'(t) = v_i(t) \qquad \text{für alle } t \in \mathbb{R},$$

so dass wir wieder in der Lage sind, die Bewegung der Körper mit Hilfe eines Zeitschrittverfahrens zu simulieren. Da die Kräfte, also auch die Beschleunigung, nur von der Position der Körper abhängen, lässt sich beispielsweise das Leapfrog-Verfahren unmittelbar anwenden.

Eine Besonderheit ist dabei allerdings zu beachten: Falls zwei Körper aufeinandertreffen, falls also $x_i(t) = x_j(t)$ für $i \neq j$ gelten sollte, lässt sich die Kraft nicht mehr berechnen, da im Nenner der Formel (3.3) eine Null auftritt. In dieser Situation verliert unser Modell seine Gültigkeit und müsste um Formeln erweitert werden, die den Zusammenstoß zweier Körper beschreiben.

3.2 Ersatzmassen

Die Berechnung einer einzelnen Kraft mit Hilfe der Gleichung (3.3) erfordert es, insgesamt $n-1$ Summanden zu berechnen. Für Systeme mit nur wenigen Körpern stellt das keine große Herausforderung dar, allerdings ändert sich die Situation grundlegend, wenn sehr viele Wechselwirkungen zwischen Körpern zu berechnen sind.

Die Anzahl der Sonnen in der Milchstraße wird beispielsweise auf mehr als 100 Milliarden (also 10^{11}) geschätzt. Wollten wir sie simulieren, müssten wir also für jede einzelne Sonne ungefähr 100 Milliarden Summanden berechnen. Die Durchführung eines einzelnen Zeitschritts erfordert die Bestimmung der Kräfte für alle Sonnen, so dass ungefähr 10^{22} Terme auszuwerten wären. Ein aktueller Prozessor wird nicht mehr als eine Milliarde von Termen pro Sekunde ausrechnen können, so dass 10^{13} Sekunden für einen einzigen Zeitschritt nötig wären, also mehr als 300 000 Jahre. Auf Probleme dieser Größenordnung lassen sich also die von uns bisher behandelten Verfahren nicht mehr sinnvoll anwenden.

Da wir ohnehin über das Zeitschrittverfahren nur eine Näherungslösung berechnen, ist es allerdings akzeptabel, auch die Auswertung der Kräfte nur näherungsweise durchzuführen, solange wir sicherstellen können, dass wir dabei jede beliebige Genauigkeit erreichen.

Ein in der Praxis sehr erfolgreicher Ansatz besteht darin, mehrere Massen zu einer einzigen „Ersatzmasse“ zusammenzufassen, die in hinreichend großer Entfernung im Wesentlichen denselben Effekt wie die Einzelmassen ausübt.

Für die Herleitung der Theorie verzichten wir dabei zunächst auf die Zeitabhängigkeit, gehen also davon aus, dass n Punktmassen gegeben sind, von denen die i-te an der Position $x_i \in \mathbb{R}^3$ eine Masse von $m_i \in \mathbb{R}_{\geq 0}$ aufweist. Wir bezeichnen die Indexmenge wieder mit

$$\mathcal{I} := \{1, \dots, n\}$$

und wählen eine Teilmenge

$$\hat{s} \subseteq \mathcal{I}$$

von Indizes (der Buchstabe s soll hier für das englische Wort *sources* stehen), die zu denjenigen Massen gehören, die wir durch eine „Ersatzmasse“ ersetzen wollen.

Die Position dieser Ersatzmasse bezeichnen wir mit $x_s \in \mathbb{R}^3$, ihre Masse mit $m_s \in \mathbb{R}_{\geq 0}$. Wir wollen sie so wählen, dass

$$\sum_{j \in \hat{s}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} \approx m_s \frac{x_s - x_i}{\|x_s - x_i\|_2^3}$$

gilt, dass also die Ersatzmasse auf x_i ungefähr dieselbe Kraft ausübt wie *sämtliche* durch $\hat{s}$ gegebene Massen zusammen.

Um herauszufinden, unter welchen Bedingungen wir eine gute Näherung erhalten, müssen wir auf grundlegende Regeln für das Rechnen mit Normen zurückgreifen können.

Erinnerung 3.2 (Dreiecksungleichung) *Es gilt*

$$\|x + y\|_2 \leq \|x\|_2 + \|y\|_2 \qquad \text{für alle } x, y \in \mathbb{R}^d.$$

Da Integrale eng mit Summen verwandt sind, gilt auch

$$\left\| \int_a^b g(t)\,ds \right\|_2 \leq \int_a^b \|g(t)\|\,ds \qquad \text{für alle } g \in C([a,b],\mathbb{R}^d).$$

In unserem Beweis werden wir die Ableitung der Norm berechnen müssen. Dafür ist es hilfreich, das *Skalarprodukt* einzuführen.

Erinnerung 3.3 (Skalarprodukt) *Das* euklidische Skalarprodukt *ist durch*

$$\langle x,y\rangle_2 := \sum_{i=1}^n x_i y_i \qquad \text{für alle } x,y \in \mathbb{R}^n$$

gegeben. Es erfüllt die Gleichung

$$\|x\|_2^2 = \langle x,x\rangle_2 \qquad \text{für alle } x \in \mathbb{R}^n.$$

Mit Hilfe der Produkt-, Ketten- und Potenzregel folgen daraus die Gleichungen

$$\frac{\partial}{\partial t}\|x+ty\|_2^2 = 2\langle x+ty,y\rangle_2 \qquad \text{für alle } x,y \in \mathbb{R}^n,\ t \in \mathbb{R},$$
$$\frac{\partial}{\partial t}\|x+ty\|_2^\alpha = \alpha\langle x+ty,y\rangle_2\|x+ty\|_2^{\alpha-2}, \quad \text{für alle } x,y \in \mathbb{R}^n,\ t \in \mathbb{R} \text{ mit } x+ty \neq 0.$$

Darüber hinaus steht uns die Cauchy-Schwarz-Ungleichung

$$|\langle x,y\rangle| \leq \|x\|_2\|y\|_2 \qquad \text{für alle } x,y \in \mathbb{R}^d \tag{3.4}$$

zur Verfügung, in der Gleichheit genau dann gilt, wenn x und y linear abhängig sind.

Mit diesen Hilfsmitteln können wir die folgende Abschätzung für den durch den Wechsel zu einer Ersatzmasse eingeführten Fehler gewinnen:

Satz 3.4 (Ersatzmasse) *Seien $x_i, x_j, x_s \in \mathbb{R}^3$ gegeben, und sei*

$$\delta := \min\{\|x_j + t(x_s - x_j) - x_i\|_2 \ :\ t \in [0,1]\} \tag{3.5}$$

der minimale Abstand zwischen der Strecke von x_j zu x_s und dem Punkt x_i.

Falls $\delta > 0$ gilt, haben wir

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} \right\|_2 \leq 4\frac{\|x_s - x_j\|_2}{\delta^3}. \tag{3.6}$$

Beweis. Es gelte $\delta > 0$.

Für die Untersuchung von Differenzen ist häufig der Hauptsatz der Integral- und Differentialrechnung (siehe Erinnerung 2.1) hilfreich, so auch in diesem Fall. Um ihn anwenden zu können, definieren wir die Funktion

$$g : [0,1] \to \mathbb{R}^3, \qquad t \mapsto \frac{x_j + t(x_s - x_j) - x_i}{\|x_j + t(x_s - x_j) - x_i\|_2^3},$$

die offenbar

$$g(0) = \frac{x_j - x_i}{\|x_j - x_i\|_2^3}, \qquad g(1) = \frac{x_s - x_i}{\|x_s - x_i\|_2^3}$$

erfüllt, so dass wir

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} \right\|_2 = \|g(0) - g(1)\|_2 = \|g(1) - g(0)\|_2$$

erhalten. Mit dem Hauptsatz und der Dreiecksungleichung folgt

$$\|g(1) - g(0)\|_2 = \left\| \int_0^1 g'(t)\, dt \right\|_2 \leq \int_0^1 \|g'(t)\|_2\, dt.$$

Um bei der Berechnung der Ableitung g' etwas Zeit zu sparen kürzen wir $z := x_j - x_i$ und $d := x_s - x_j$ ab und schreiben

$$g(t) = \frac{z + td}{\|z + td\|_2^3} \qquad \text{für alle } t \in [0, 1].$$

Mit der Produktregel und Erinnerung 3.3 erhalten wir

$$g'(t) = d\frac{1}{\|z + td\|_2^3} - 3(z + td)\frac{\langle z + td, d\rangle}{\|z + td\|_2^5} \qquad \text{für alle } t \in [0, 1],$$

und mit der Dreiecksungleichung sowie der Cauchy-Schwarz-Ungleichung folgt

$$\|g'(t)\|_2 \leq \frac{\|d\|_2}{\|z + td\|_2^3} + 3\frac{\|z + td\|^2\|d\|_2}{\|z + td\|_2^5} = 4\frac{\|d\|_2}{\|z + td\|_2^3} \qquad \text{für alle } t \in [0, 1].$$

Nach Voraussetzung gilt $\|z + td\|_2 \geq \delta > 0$, so dass wir zu

$$\int_0^1 \|g'(t)\|_2\, dt \leq \int_0^1 4\frac{\|d\|_2}{\|z + td\|_2^3}\, dt \leq 4\int_0^1 \frac{\|d\|_2}{\delta^3}\, dt = 4\frac{\|x_s - x_j\|_2}{\delta^3}$$

gelangen und der Beweis vollständig ist. ∎

Damit die von der Ersatzmasse x_s auf x_i ausgeübte Kraft der der ursprünglichen Masse x_j möglichst nahe kommt, sollte also der Quotient aus $\|x_j - x_s\|_2$ und δ möglichst klein sein. Da die Größe δ etwas unhandlich ist, werden wir sie nun durch eine untere Abschätzung ersetzen, die sich praktisch konstruieren lässt.

Dazu ersetzen wir die Punktwolke $\{x_j \,:\, j \in \hat{s}\}$ durch einen achsenparallelen Quader

$$s = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3],$$

der sie vollständig enthält, der also

$$x_j \in s \qquad \text{für alle } j \in \hat{s} \tag{3.7}$$

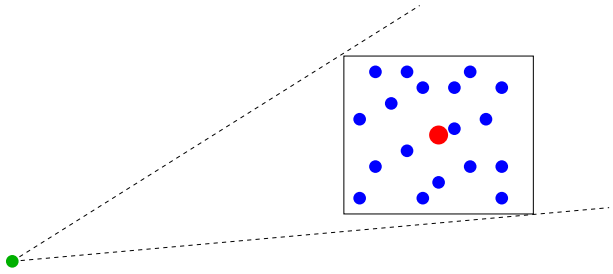

Abbildung 3.1: Eine Ersatzmasse (rot) kann ungefähr dieselbe Kraft wie viele Einzelmassen (blau) auf einen hinreichend weit entfernten Punkt (grün) ausüben.

erfüllt. Ein derartiger Quader lässt sich beispielsweise konstruieren, indem man die Minima und Maxima der Koordinaten aller Punkte berechnet.

Quader haben den Vorteil, *konvexe* Mengen zu sein, dass bedeutet, dass die Verbindungsstrecke zweier beliebiger Punkte innerhalb des Quaders vollständig im Quader enthalten ist. Präzise können wir diese Eigenschaft als

$$ty + (1-t)z \in s \qquad \text{für alle } y, z \in s,\ t \in [0,1] \tag{3.8}$$

formulieren und mit

$$a_\iota = ta_\iota + (1-t)a_\iota \leq ty_\iota + (1-t)z_\iota \leq tb_\iota + (1-t)b_\iota = b_\iota \quad \text{für alle } y, z \in s,\ t \in [0,1]$$

für alle Koordinatenrichtungen $\iota \in \{1,2,3\}$ nachprüfen. Diese Eigenschaft ist nützlich für die Abschätzung der in (3.5) definierten Zahl δ: Falls $x_j, x_s \in s$ gilt, folgt

$$y := x_j + t(x_s - x_j) = (1-t)x_j + tx_s \in s,$$

so dass wir δ durch den Abstand

$$\operatorname{dist}(x_i, s) := \min\{\|x_i - y\|_2 \ :\ y \in s\}$$

des Punkts x_i von dem Quader s nach unten abschätzen können. Damit ist der Nenner der Fehlerschranke (3.6) auf eine besser handhabbare Form gebracht.

Übungsaufgabe 3.5 (Abstand) *Die Verwendung eines achsenparallelen Quaders für unsere Zwecke bietet den großen Vorteil, dass sich der Abstand eines Punkts zu diesem Quader einfach berechnen lässt: Wir definieren durch*

$$\operatorname{dist}(y,[a,b]) := \begin{cases} a-y & \text{falls } y < a, \\ y-b & \text{falls } b < y, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } y, a, b \in \mathbb{R} \text{ mit } a < b$$

den eindimensionalen Abstand von y zu dem Intervall $[a,b]$. Beweisen Sie

$$\operatorname{dist}(y,s) = \left(\sum_{\iota=1}^{3} \operatorname{dist}(y_\iota, [a_\iota, b_\iota])^2\right)^{1/2} \qquad \text{für alle } y \in \mathbb{R}^3.$$

Um auch den Zähler der Fehlerschranke (3.6) zu vereinfachen legen wir fest, dass die Ersatzmasse im Mittelpunkt des Quaders s untergebracht werden soll, wir setzen also

$$x_s := \begin{pmatrix} (b_1 + a_1)/2 \\ (b_2 + a_2)/2 \\ (b_3 + a_3)/2 \end{pmatrix}.$$

Es gilt

$$\frac{a_\iota - b_\iota}{2} = a_\iota - \frac{b_\iota + a_\iota}{2} \leq y_\iota - \frac{b_\iota + a_\iota}{2} \leq b_\iota - \frac{b_\iota + a_\iota}{2} \leq \frac{b_\iota - a_\iota}{2} \qquad \text{für alle } y \in s$$

und alle Koordinatenrichtungen $\iota \in \{1, 2, 3\}$, so dass wir

$$\|y - x_s\|_2 = \left(\sum_{\iota=1}^{3} (y_\iota - x_{s,\iota})^2 \right)^{1/2} \leq \left(\sum_{\iota=1}^{3} \frac{(b_\iota - a_\iota)^2}{4} \right)^{1/2} = \operatorname{diam}(s)/2 \quad \text{für alle } y \in s$$

erhalten, wobei

$$\operatorname{diam}(s) := \left(\sum_{\iota=1}^{3} (b_\iota - a_\iota)^2 \right)^{1/2}$$

den *Durchmesser* des Quaders s bezeichnet. Insgesamt können wir die Abschätzung (3.6) in die einfachere Form

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} \right\|_2 \leq 4 \frac{\operatorname{diam}(s)/2}{\operatorname{dist}(x_i, s)^3}$$

bringen, die sich algorithmisch ausnutzen lässt.

Um den Fehler zu steuern wählen wir eine Zahl $\eta > 0$ und werden einen Algorithmus entwickeln, der sicherstellt, dass nur solche Quader s verwendet werden, die die *Zulässigkeitsbedingung*

$$\operatorname{diam}(s) \leq 2\eta \operatorname{dist}(x_i, s) \tag{3.9}$$

erfüllen. Aus ihr folgt unmittelbar

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} \right\|_2 \leq \frac{4\eta}{\operatorname{dist}(x_i, s)^2}, \tag{3.10}$$

wir können den Fehler also beliebig reduzieren, indem wir η klein genug wählen. Für kleine Werte von η können wir uns diese Zahl als den Winkel vorstellen, unter dem die durch $\hat{s}$ beschriebene Punktwolke von x_i aus zu sehen ist.

Wir sind daran interessiert, *sämtliche* zu den Indizes $j \in \hat{s}$ gehörenden Massen zu ersetzen. Falls η in (3.9) hinreichend klein ist, erhalten wir mit (3.10) die Näherung

$$\sum_{j \in \hat{s}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} \approx \sum_{j \in \hat{s}} m_j \frac{x_s - x_i}{\|x_s - x_i\|_2^3} = m_s \frac{x_s - x_i}{\|x_s - x_i\|_2^3},$$

indem wir einfach

$$m_s := \sum_{j \in \hat{s}} m_j$$

verwenden, also die Ersatzmasse gerade als Summe aller Einzelmassen wählen.

In der Praxis ist es sehr unwahrscheinlich, dass $\hat{s} = \mathcal{I} \setminus \{i\}$ die Bedingung (3.9) erfüllt, so dass wir die Ersatzmasse nicht für die gesamte Summe in (3.3) anwenden können. Wir können allerdings versuchen, die Summe in Teilsummen zu zerlegen, für die sich Ersatzmassen einsetzen lassen. Es empfiehlt sich, diese Zerlegung so zu organisieren, dass sich die einmal berechneten Ersatzmassen nicht nur für ein $i \in \mathcal{I}$, sondern für möglichst viele benutzen lassen.

Dazu verfolgen wir einen geometrischen Ansatz: Unser Ziel ist es, für eine Menge $\hat{s} \subseteq \mathcal{I}$ die Summe

$$\sum_{j \in \hat{s}} \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$$

zu approximieren. Falls (3.9) gilt, können wir $\hat{s}$ durch die Ersatzmasse an der Position x_s ersetzen und den Fehler mit Satz 3.4 und (3.10) abschätzen.

Anderenfalls wählen wir eine Koordinatenrichtung und zerlegen den Quader s entlang dieser Richtung in zwei gleich große Teilquader. Beispielsweise erhalten wir für die erste Koordinatenrichtung

$$\begin{aligned} s_1 &:= [a_1, c_1] \times [a_2, b_2] \times [a_3, b_3], \\ s_2 &:= [c_1, b_1] \times [a_2, b_2] \times [a_3, b_3], \end{aligned}$$

wobei $c_1 := (b_1 + a_1)/2$ der Mittelpunkt des Intervalls $[a_1, b_1]$ ist. Um die Eigenschaft (3.7) sicherzustellen definieren wir

$$\hat{s}_1 := \{j \in \hat{s} \ : \ x_j \in s_1\}, \qquad \hat{s}_2 := \hat{s} \setminus \hat{s}_1.$$

Nun gilt (3.7) sowohl für den Quader s_1 mit der Indexmenge $\hat{s}_1$ als auch für den Quader s_2 mit der Indexmenge $\hat{s}_2$. Nach Konstruktion haben wir auch

$$\sum_{j \in \hat{s}} \frac{x_j - x_i}{\|x_j - x_i\|_2^3} = \sum_{j \in \hat{s}_1} \frac{x_j - x_i}{\|x_j - x_i\|_2^3} + \sum_{j \in \hat{s}_2} \frac{x_j - x_i}{\|x_j - x_i\|_2^3},$$

so dass wir eine Rekursion verwenden können, um die beiden Teilsummen für die Indexmengen $\hat{s}_1$ und $\hat{s}_2$ zu approximieren.

3.3 Clusterbaum

Wir haben gesehen, dass wir die Auswertung der auf eine Masse wirkenden Kräfte effizient approximieren können, indem wir hinreichend weit entfernte Massen durch wenige Ersatzmassen ersetzen, die annähernd dasselbe Kraftfeld hervorrufen.

Unsere Aufgabe besteht nun darin, eine effiziente und praktisch durchführbare Methode zu entwickeln, mit der sich diese Ersatzmassen finden lassen. Dazu definieren wir

zunächst formal eine Hierarchie von Teilgebieten, die für die Approximation in Frage kommen. Aus dieser Hierarchie werden wir anschließend diejenigen Gebiete auswählen, die die Zulässigkeitsbedingung (3.9) erfüllen.

Definition 3.6 (Clusterbaum) *Sei $\mathcal{I}$ eine Indexmenge. Sei $\mathcal{T}$ ein Baum, und sei zu jedem seiner Knoten $s \in \mathcal{T}$ eine Beschriftung $\hat{s}$ gegeben. Wir nennen $\mathcal{T}$ einen* Clusterbaum für die Indexmenge $\mathcal{I}$*, falls die folgenden Bedingungen gelten:*

1. *Die Wurzel $r \in \mathcal{T}$ des Baums erfüllt $\hat{r} = \mathcal{I}$.*
2. *Falls ein Knoten $s \in \mathcal{T}$ Söhne besitzt, gilt*

$$\hat{s} = \bigcup_{s' \in \operatorname{sons}(s)} \hat{s}'.$$

3. *Falls ein Knoten $s \in \mathcal{T}$ Söhne besitzt, gilt auch*

$$s_1 \neq s_2 \Rightarrow \hat{s}_1 \cap \hat{s}_2 = \emptyset \qquad \text{für alle } s_1, s_2 \in \operatorname{sons}(s).$$

Einen Clusterbaum für die Indexmenge $\mathcal{I}$ bezeichnen wir mit $\mathcal{T}_\mathcal{I}$, seine Knoten nennen wir Cluster*.*

Ein Clusterbaum repräsentiert eine hierarchische Zerlegung einer Indexmenge $\mathcal{I}$ in disjunkte Teilmengen: Die erste Bedingung stellt sicher, dass wir die gesamte Indexmenge zerlegen können. Die zweite sorgt dafür, dass bei der Zerlegung kein Index verloren geht. Die dritte besagt, dass auch kein Index doppelt auftreten soll.

Falls beispielsweise ein Cluster $s \in \mathcal{T}_\mathcal{I}$ gegeben ist, für den wir keine Ersatzmasse verwenden dürfen, können wir zu seinen Söhnen übergehen und erhalten mit der zweiten und dritten Bedingung die Gleichung

$$\sum_{j \in \hat{s}} \frac{x_j - x_i}{\|x_j - x_i\|_2^3} = \sum_{s' \in \operatorname{sons}(s)} \sum_{j \in \hat{s}'} \frac{x_j - x_i}{\|x_j - x_i\|_2^3},$$

so dass wir rekursiv die Cluster $s' \in \operatorname{sons}(s)$ überprüfen können.

Für diese Überprüfung benötigen wir allerdings in der Regel auch Informationen über die Geometrie, beispielsweise um die Zulässigkeitsbedingung (3.9) anwenden zu können. Diese geometrischen Informationen können wir dem Clusterbaum hinzufügen.

Definition 3.7 (Geometrischer Clusterbaum) *Sei $\mathcal{T}_\mathcal{I}$ ein Clusterbaum für eine Indexmenge $\mathcal{I}$, und sei $d \in \mathbb{N}$. Wir nennen $\mathcal{T}_\mathcal{I}$ einen* geometrischen Clusterbaum*, falls die folgenden Bedingungen gelten:*

1. *Jeder Cluster $s \in \mathcal{T}_\mathcal{I}$ ist eine Teilmenge des Raums $\mathbb{R}^d$.*
2. *Es gilt*

$$s' \subseteq s \qquad \text{für alle } s \in \mathcal{T}_\mathcal{I},\ s' \in \operatorname{sons}(s).$$

3. Außerdem gilt

$$x_j \in s \qquad \text{für alle } s \in \mathcal{T}_\mathcal{I},\ j \in \hat{s}.$$

Die erste Bedingung stellt die Beziehung zu geometrischen Objekten her. Indem wir mit der zweiten Bedingung sicherstellen, dass die Söhne eines Clusters Teilmengen des Clusters sind, können wir bei der Überprüfung der Zulässigkeitsbedingung besonders einfache Algorithmen verwenden. Die dritte Bedingung kennen wir bereits aus dem vorigen Abschnitt, sie garantiert, dass wir die Zulässigkeitsbedingung nur für s statt für alle in s enthaltenen Punkte überprüfen müssen.

Eine praktische Konstruktion für geometrische Clusterbäume wurde bereits am Ende des vorangehenden Abschnitts behandelt: Die Cluster $s \in \mathcal{T}_\mathcal{I}$ sind achsenparallele Quader, die rekursiv in Teilquader zerlegt werden. Als Wurzel des Clusterbaums empfiehlt sich der Quader

$$r := [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$$

minimaler Größe, der alle Punkte enthält. Er ist durch

$$a_\iota := \min\{x_{j,\iota} \ :\ j \in \mathcal{I}\}, \qquad b_\iota := \max\{x_{j,\iota} \ :\ j \in \mathcal{I}\} \qquad \text{für alle } \iota \in \{1,2,3\}$$

gegeben und kann mit diesen Formeln auch praktisch konstruiert werden. Mit $\hat{r} := \mathcal{I}$ erfüllt der Wurzelcluster r damit die ersten Bedingungen der Definitionen 3.6 und 3.7.

Bei der Zerlegung von Clustern sollten wir dafür sorgen, dass ihre Durchmesser möglichst schnell abnehmen, damit die Zulässigkeitsbedingung (3.9) möglichst schnell erfüllt ist. Um einen Cluster $s \in \mathcal{T}_\mathcal{I}$ mit

$$s = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$$

zu zerlegen wählen wir deshalb diejenige Richtung, in der er die größte Ausdehnung aufweist, also $\iota \in \{1,2,3\}$ mit

$$b_k - a_k \leq b_\iota - a_\iota \qquad \text{für alle } k \in \{1,2,3\}.$$

Entlang dieser Richtung wird der Quader halbiert, wir berechnen also den Mittelpunkt

$$c_\iota := \frac{b_\iota + a_\iota}{2}$$

und definieren die Söhne des Clusters s durch

$$s_1 := \begin{cases} [a_1, c_1] \times [a_2, b_2] \times [a_3, b_3] & \text{falls } \iota = 1, \\ [a_1, b_1] \times [a_2, c_2] \times [a_3, b_3] & \text{falls } \iota = 2, \\ [a_1, b_1] \times [a_2, b_2] \times [a_3, c_3] & \text{ansonsten,} \end{cases} \tag{3.11a}$$

$$s_2 := \begin{cases} [c_1, b_1] \times [a_2, b_2] \times [a_3, b_3] & \text{falls } \iota = 1, \\ [a_1, b_1] \times [c_2, b_2] \times [a_3, b_3] & \text{falls } \iota = 2, \\ [a_1, b_1] \times [a_2, b_2] \times [c_3, b_3] & \text{ansonsten.} \end{cases} \tag{3.11b}$$

Damit ist die zweite Bedingung der Definition 3.7 erfüllt. Wenn wir die zu den Sohnclustern gehörenden Indexmengen durch

$$\hat{s}_1 := \{j \in \hat{s} \ : \ x_j \in s_1\}, \qquad \hat{s}_2 := \hat{s} \setminus \hat{s}_1$$

festlegen, erfüllen wir auch die dritte Bedingung dieser Definition und die zweite und dritte Bedingung der Definition 3.6.

Wir unterteilen so lange rekursiv, bis die Cluster nur noch sehr wenige Punkte enthalten und es ohne größeren Aufwand möglich ist, die von ihnen ausgeübten Kräfte direkt zu berechnen.

Wenn uns der Clusterbaum $\mathcal{T}_\mathcal{I}$ zur Verfügung steht, können wir daran gehen, die Ersatzmassen zu konstruieren. Den Mittelpunkt x_s eines Clusters $s \in \mathcal{T}_\mathcal{I}$ zu finden ist dabei sehr einfach, für die Berechnung der Masse m_s empfiehlt es sich, einen Trick zu verwenden: Wir haben bereits gesehen, dass

$$m_s = \sum_{j \in \hat{s}} m_j$$

eine sinnvolle Wahl der Ersatzmasse m_s ist. Statt die Summe direkt zu bestimmen können wir uns wieder die zweite und dritte Eigenschaft eines Clusterbaums zunutze machen: Falls $\operatorname{sons}(s) \neq \emptyset$ gilt, folgt

$$m_s = \sum_{j \in \hat{s}} m_j = \sum_{s' \in \operatorname{sons}(s)} \sum_{j \in \hat{s}'} m_j = \sum_{s' \in \operatorname{sons}(s)} m_{s'}.$$

Es genügt also, die *Ersatzmassen* der Söhne s' des Clusters s zu addieren, statt alle Punkte x_j einzeln zu betrachten.

Praktisch ausnutzen können wir diese Eigenschaft mit einer Rekursion, die den Clusterbaum von den Blättern in Richtung der Wurzel durchläuft und so sicherstellt, dass die Werte $m_{s'}$ für alle Söhne eines Clusters s berechnet werden, bevor wir den Wert m_s bestimmen. Dann lässt sich die oben beschriebene effizientere Summe einsetzen.

In Abbildung 3.2 ist der Algorithmus zusammengefasst, mit dem wir den Clusterbaum und die Ersatzmassen effizient konstruieren können. Dabei bezeichnet der Parameter $r_\mathcal{I} \geq 1$ die Anzahl der Massen, die ein Cluster maximal enthalten darf, der nicht weiter unterteilt wurde.

Bemerkung 3.8 (Indexmengen) *Die Indexmengen $\hat{s}$ lassen sich elegant durch Arrays darstellen, in denen die Indizes aufgelistet sind. Wenn eine Menge in Teilmengen $\hat{s}_1$ und $\hat{s}_2$ zerlegt werden soll, sortiert man das Array so um, dass die Elemente in $\hat{s}_1$ zuerst vorkommen und die von $\hat{s}_2$ zuletzt. Dann kann man die beiden Teilarrays für die Söhne weiterverwenden. Die Vorgehensweise lässt sich dabei analog zu der üblichen Implementierung des Quicksort-Algorithmus' gestalten.*

Sobald Clusterbaum und Ersatzmassen vorliegen, können wir daran gehen, die Kraftfelder auszuwerten. Auch dieser Aufgabe nähern wir uns mit einem rekursiven Algorithmus: Um für einen Punkt x_i und einen Cluster $s \in \mathcal{T}_\mathcal{I}$ die Summe

$$\sum_{j \in \hat{s} \setminus \{i\}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$$

```
procedure setup(var s);
begin
  if ŝ ≤ r_I then begin
    m_s ← Σ_{j∈ŝ} m_j;
    sons(s) ← ∅
  end
  else begin
    Finde ι ∈ {1,2,3} mit b_ι − a_ι maximal
    Zerlege s gemäß (3.11) in s_1 und s_2
    ŝ_1 ← {j ∈ ŝ : x_j ∈ s_1}; ŝ_2 ← ŝ \ ŝ_1;
    setup(s_1); setup(s_2);
    m_s := m_{s_1} + m_{s_2};
    sons(s) ← {s_1, s_2}
  end
end
```

Abbildung 3.2: Konstruktion des Clusterbaums und der Ersatzmassen

auszuwerten, prüfen wir zunächst, ob die Zulässigkeitsbedingung (3.9) erfüllt ist. Der Durchmesser der Cluster s kann dabei ebenfalls vorberechnet werden, der Abstand zwischen x_i und s lässt sich mit der in Übungsaufgabe 3.5 gegebenen Formel effizient ermitteln.

Falls die Bedingung erfüllt ist, ersetzen wir die Kräfte durch die durch die Ersatzmasse ausgeübte Kraft

$$\sum_{j\in\hat{s}\setminus\{i\}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} \approx m_s \frac{x_s - x_i}{\|x_s - x_i\|_2^3}$$

und sind fertig.

Ansonsten prüfen wir, ob s Söhne hat. Falls ja, verwenden wir wieder die zweite und dritte Eigenschaft des Clusterbaums, um mittels

$$\sum_{j\in\hat{s}\setminus\{i\}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} = \sum_{s'\in\mathrm{sons}(s)} \sum_{j\in\hat{s}'\setminus\{i\}} m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$$

die Summe des Vaterclusters durch Summen seiner Söhne zu ersetzen. Diese Summen berechnen wir rekursiv.

Falls schließlich s keine Söhne hat, dürfen wir davon ausgehen, dass der Cluster nur wenige Punkte enthält, so dass wir die Summe einfach direkt auswerten können.

Der resultierende Algorithmus für die approximative Auswertung der Kraft ist in Abbildung 3.3 zusammengefasst.

```
procedure eval(i, s, var f);
begin
  if diam(s) ≤ 2η dist(x_i, s) then
    f ← f + m_s (x_s − x_i)/‖x_s − x_i‖_2^3
  else if sons(s) = ∅ then
    f ← f + Σ_{j∈ŝ\{i}} m_j (x_j − x_i)/‖x_j − x_i‖_2^3
  else
    for s' ∈ sons(s) do
      eval(i, s', f)
end
```

Abbildung 3.3: Auswertung der approximierten Kraft

3.4 Rechenaufwand

Im Vergleich zu den bisher behandelten Algorithmen gestaltet sich die Analyse des in den Abbildungen 3.2 und 3.3 dargestellten Verfahrens für die näherungsweise Bestimmung des Gravitationsfelds etwas anspruchsvoller.

Wir werden deshalb zunächst einige vorbereitende Aussagen beweisen, bevor wir uns der eigentlichen Aufwandsabschätzung widmen. Die erste dieser Aussagen beschäftigt sich mit dem Verhalten des Durchmessers bei der von uns verwendeten Strategie für die Konstruktion des Clusterbaums. In Hinblick auf die Zulässigkeitsbedingung (3.9) ist es erstrebenswert, dass der Durchmesser schnell sinkt, und wir können beweisen, dass er das tatsächlich tut:

Lemma 3.9 (Durchmesser) *Seien $s \in \mathcal{T}_{\mathcal{I}}$, $s' \in \operatorname{sons}(s)$, $s'' \in \operatorname{sons}(s')$, $s''' \in \operatorname{sons}(s'')$ vier Cluster, die mit der vorgestellten Strategie konstruiert wurden. Dann gilt*

$$\operatorname{diam}(s''') \leq \frac{1}{2} \operatorname{diam}(s).$$

Beweis. Da s''' ein achsenparalleler Quader ist, finden wir $a_1, a_2, a_3, b_1, b_2, b_3 \in \mathbb{R}$ mit

$$s''' = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3].$$

Die Ausdehnung des Quaders in der ersten, zweiten und dritten Koordinatenrichtung bezeichnen wir mit

$$e_1 := b_1 - a_1, \qquad e_2 := b_2 - a_2, \qquad e_3 := b_3 - a_3.$$

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass s''' aus s'' entstanden ist, indem in der ersten Koordinatenrichtung geteilt wurde. Infolge unserer Strategie kann

das nur geschehen, wenn s'' seine größte Ausdehnung in der ersten Koordinatenrichtung aufweist, falls also $2e_1 \geq e_2$ und $2e_1 \geq e_3$ gelten.

Fall 1: s'' ist aus s' entstanden, indem ebenfalls in der ersten Koordinatenrichtung geteilt wurde.

Fall 1a: s' ist aus s entstanden, indem erneut in der ersten Koordinatenrichtung geteilt wurde. Dann erhalten wir für den Durchmesser von s die Abschätzung

$$\begin{aligned}\operatorname{diam}(s)^2 &= 64e_1^2 + e_2^2 + e_3^2 = 40e_1^2 + (12e_1^2 + e_2^2) + (12e_1^2 + e_3^2)\\ &\geq 40e_1^2 + (3e_2^2 + e_2^2) + (3e_3^2 + e_3^2) = 40e_1^2 + 4e_2^2 + 4e_3^2\\ &\geq 4e_1^2 + 4e_2^2 + 4e_3^2 = 4\operatorname{diam}(s''')^2.\end{aligned}$$

Fall 1b: s' ist aus s entstanden, indem in einer anderen Koordinatenrichtung unterteilt wurde. Ohne Beschränkung der Allgemeinheit dürfen wir annehmen, dass es die zweite war. Dann erhalten wir

$$\begin{aligned}\operatorname{diam}(s)^2 &= 16e_1^2 + 4e_2^2 + e_3^2 = 4e_1^2 + 4e_2^2 + (12e_1^2 + e_3^2)\\ &\geq 4e_1^2 + 4e_2^2 + (3e_3^2 + e_3^2) = 4e_1^2 + 4e_2^2 + 4e_3^2 = 4\operatorname{diam}(s''')^2.\end{aligned}$$

Fall 2: s'' ist aus s' entstanden, indem in einer anderen Koordinatenrichtung unterteilt wurde. Ohne Beschränkung der Allgemeinheit dürfen wir annehmen, das es die zweite war. Wegen unserer Strategie muss dann die Ausdehnung in dieser Richtung am größten gewesen sein, es müssen also $2e_2 \geq 2e_1$ sowie $2e_2 \geq e_3$ gelten.

Fall 2a: s' ist aus s entstanden, indem in der ersten Koordinatenrichtung unterteilt wurde. Dann gilt

$$\begin{aligned}\operatorname{diam}(s)^2 &= 16e_1^2 + 4e_2^2 + e_3^2 = 4e_1^2 + 4e_2^2 + (12e_1^2 + e_3^2)\\ &\geq 4e_1^2 + 4e_2^2 + (3e_2^2 + e_3^2) = 4e_1^2 + 4e_2^2 + 4e_3^2 = 4\operatorname{diam}(s''')^2.\end{aligned}$$

Fall 2b: s' ist aus s entstanden, indem in der zweiten Koordinatenrichtung unterteilt wurde. Das kann nur geschehen sein, weil in dieser Koordinatenrichtung Analog zu Fall 2a gilt

$$\begin{aligned}\operatorname{diam}(s)^2 &= 4e_1^2 + 16e_2^2 + e_3^2 = 4e_1^2 + 4e_2^2 + (12e_2^2 + e_3^2)\\ &\geq 4e_1^2 + 4e_2^2 + (3e_3^2 + e_3^2) = 4e_1^2 + 4e_2^2 + 4e_3^2 = 4\operatorname{diam}(s''')^2.\end{aligned}$$

Fall 2c: s' ist aus s entstanden, indem in der dritten Koordinatenrichtung unterteilt wurde. Dann gilt

$$\operatorname{diam}(s)^2 = 4e_1^2 + 4e_2^2 + 4e_3^2 = 4\operatorname{diam}(s''')^2.$$

Insgesamt haben wir in allen Fällen die Ungleichung $\operatorname{diam}(s)^2 \geq 4\operatorname{diam}(s''')^2$ bewiesen, aus der unmittelbar die Behauptung folgt. ∎

Um den Rechenaufwand und den Speicherbedarf unseres Verfahrens abschätzen zu können ist es von Interesse, die *Tiefe* des Clusterbaums abschätzen zu können. Dazu definieren wir zunächst die *Stufenzahl* induktiv als

$$\operatorname{level}(s) := \begin{cases}\operatorname{level}(s^+) + 1 & \text{falls } s \in \operatorname{sons}(s^+) \text{ für ein } s^+ \in \mathcal{T}_\mathcal{I},\\ 0 & \text{ansonsten, also falls } s \text{ die Wurzel ist,}\end{cases} \quad \text{für alle } s \in \mathcal{T}_\mathcal{I}.$$

Sie misst den Abstand eines Clusters von der Wurzel. Die Tiefe des Baums ist das Maximum der Stufenzahlen, also

$$\operatorname{depth}(\mathcal{T}_\mathcal{I}) := \max\{\operatorname{level}(s) \ : \ s \in \mathcal{T}_\mathcal{I}\}.$$

Wir können sie abschätzen, wenn wir wissen, wie nahe sich die Punkte x_j kommen können.

Lemma 3.10 (Baumtiefe) *Wir bezeichnen mit*

$$h := \min\{\|x_i - x_j\|_2 \ : \ i, j \in \mathcal{I}, \ i \neq j\}$$

den minimalen Abstand zwischen zwei Punkten unserer Menge. Dann gilt für den von uns konstruierten Clusterbaum

$$\operatorname{depth}(\mathcal{T}_\mathcal{I}) \leq 3 \log_2 \left(\frac{\operatorname{diam}(r)}{h} \right) + 3.$$

Beweis. Indem wir Lemma 3.9 wiederholt anwenden erhalten wir

$$\operatorname{diam}(s) \leq 2^{-\ell} \operatorname{diam}(r) \qquad \text{für alle } \ell \in \mathbb{N}_0, \ s \in \mathcal{T}_\mathcal{I}, \ \operatorname{level}(s) \geq 3\ell. \tag{3.12}$$

Indem wir diese Abschätzung auf einen geeigneten Cluster $s \in \mathcal{T}_\mathcal{I}$ anwenden erhalten wir unsere Abschätzung.

Wir wählen einen Cluster $s' \in \mathcal{T}_\mathcal{I}$ mit

$$\operatorname{level}(s') = \operatorname{depth}(\mathcal{T}_\mathcal{I}).$$

Falls $\operatorname{level}(s') = 0$ gilt, gilt unsere Aussage bereits.

Anderenfalls können wir ein $s \in \mathcal{T}_\mathcal{I}$ mit $s' \in \operatorname{sons}(s)$ finden. Nach unserer Konstruktion wird ein Cluster nur unterteilt, falls er mindestens zwei Punkte enthält. Da zwei Punkte mindestens einen Abstand von h zueinander aufweisen, folgt

$$\operatorname{diam}(s) \geq h.$$

Wir wählen $\ell \in \mathbb{N}_0$ mit $3\ell \leq \operatorname{level}(s) \leq 3\ell + 2$ und erhalten mit (3.12) die Abschätzung

$$\begin{aligned} h &\leq \operatorname{diam}(s) \leq 2^{-\ell} \operatorname{diam}(r), \\ \frac{h}{\operatorname{diam}(r)} &\leq 2^{-\ell}, \\ \frac{\operatorname{diam}(r)}{h} &\geq 2^{\ell}, \\ \log_2 \left(\frac{\operatorname{diam}(r)}{h} \right) &\geq \ell. \end{aligned}$$

Damit haben wir

$$\operatorname{depth}(\mathcal{T}_\mathcal{I}) = \operatorname{level}(s') = \operatorname{level}(s) + 1 \leq 3\ell + 3 \leq 3 \log_2 \left(\frac{\operatorname{diam}(r)}{h} \right) + 3$$

bewiesen, also die gewünschte Abschätzung. ∎

Näherungsweise verhält sich die Baumtiefe also proportional zu dem Logarithmus des Verhältnisses zwischen dem größten und dem kleinsten Abstand zweier Punkte.

Im nächsten Schritt untersuchen wir, wieviele Cluster *derselben Stufe* an der Berechnung der Kraft beteiligt sein können. Nach Konstruktion wird die Funktion „eval“ in Abbildung 3.3 nur dann für einen Cluster $s \in \mathcal{T}_\mathcal{I}$ aufgerufen, wenn es sich entweder um die Wurzel des Clusterbaums handelt oder der Vater $s^+ \in \mathcal{T}_\mathcal{I}$ des Clusters die Zulässigkeitsbedingung (3.9) verletzte.

Die Anzahl derartiger Cluster s^+ auf einer Stufe des Baums lässt sich durch eine Konstante beschränken:

Lemma 3.11 (Anzahl unzulässiger Cluster) *Sei $y \in \mathbb{R}^3$, sei $\eta \in \mathbb{R}_{>0}$. Sei $c \in \mathbb{R}_{>0}$ eine Konstante, die*

$$c \operatorname{diam}(s)^3 \leq |s| \qquad \text{für alle } s \in \mathcal{T}_\mathcal{I} \tag{3.13}$$

erfüllt, wobei $|s|$ das Volumen des Clusters angibt. Dann gilt

$$\#\left\{s \in \mathcal{T}_\mathcal{I} \;:\; \operatorname{level}(s) = \ell,\ \operatorname{diam}(s) > 2\eta \operatorname{dist}(y,s)\right\} \leq \frac{4}{3c}\pi \left(1 + \frac{1}{2\eta}\right)^3 \quad \text{für alle } \ell \in \mathbb{N}_0.$$

Beweis. Sei $\ell \in \mathbb{N}_0$. Wir kürzen die uns interessierende Menge mit

$$N := \left\{s \in \mathcal{T}_\mathcal{I} \;:\; \operatorname{level}(s) = \ell,\ \operatorname{diam}(s) > 2\eta \operatorname{dist}(y,s)\right\}$$

ab. Sei ein $s \in N$ gewählt. Dann gilt

$$\begin{aligned} 2\eta \operatorname{dist}(y,s) &< \operatorname{diam}(s), \text{ also} \\ \operatorname{dist}(y,s) &< \frac{1}{2\eta} \operatorname{diam}(s). \end{aligned}$$

Sei $z \in s$ ein Punkt, für den

$$\|z - y\|_2 = \operatorname{dist}(y,s)$$

gilt. Für jedes $x \in s$ gilt dann

$$\|x - y\|_2 = \|(x - z) + (z - y)\|_2 \leq \|x - z\|_2 + \|z - y\|_2 \leq \operatorname{diam}(s) + \operatorname{dist}(y,s),$$

so dass wir

$$\|x - y\|_2 \leq \operatorname{diam}(s) + \operatorname{dist}(y,s) < \left(1 + \frac{1}{2\eta}\right) \operatorname{diam}(s) \qquad \text{für alle } x \in s$$

erhalten. Damit ist jeder Punkt des Clusters s in der Kugel um y mit dem Radius $(1 + 1/(2\eta)) \operatorname{diam}(s)$ enthalten, wir haben also

$$s \subseteq B, \qquad B := \left\{x \in \mathbb{R}^3 \;:\; \|x - y\| \leq (1 + 1/(2\eta)) \operatorname{diam}(s)\right\}.$$

Die Kugel B hat das Volumen

$$|B| = \frac{4}{3}\pi \left(1 + \frac{1}{2\eta}\right)^3 \operatorname{diam}(s)^3,$$

und da die Cluster $s \in N$ nach unserer Konstruktion disjunkt (abgesehen von ihren Rändern) sind, muss

$$\sum_{s \in N} |s| = \left| \bigcup_{s \in N} s \right| \leq |B|$$

gelten. Infolge unserer Konstruktion weisen auch alle Cluster auf derselben Stufe dasselbe Volumen auf (sie sind bis auf Verschiebungen identisch), so dass wir einen festen Cluster $s_0 \in N$ wählen können und

$$|s_0| \# N = \sum_{s \in N} |s_0| = \sum_{s \in N} |s| \leq |B|$$

erhalten. Mit der Voraussetzung (3.13) folgt

$$c \operatorname{diam}(s_0)^3 \# N \leq |B| = \frac{4}{3}\pi \left(1 + \frac{1}{2\eta}\right)^3 \operatorname{diam}(s_0)^3,$$
$$\# N \leq \frac{4}{3c}\pi \left(1 + \frac{1}{2\eta}\right)^3,$$

also die zu zeigende Abschätzung. ∎

Da ein $s \in \mathcal{T}_{\mathcal{I}}$ nur im Algorithmus „eval“ auftritt, falls sein Vater s^+ nicht zulässig war, folgt aus Lemma 3.11, dass es höchstens

$$C_{\mathrm{sp}} := \frac{8}{3c}\pi \left(1 + \frac{1}{2\eta}\right)^3$$

solcher Cluster pro Baumstufe geben kann.

In Kombination mit Lemma 3.10 folgt, dass ein Aufruf der Funktion „eval“ nicht mehr als $\mathcal{O}(\log_2(\operatorname{diam}(r)/h))$ Operationen erfordert.

In Hinblick auf unsere Fehlerabschätzung sind wir vor allem an dem Fall $\eta \to 0$ interessiert, für den sich C_{sp} ungefähr wie η^{-3} verhält. Der Preis für eine Halbierung des Fehlers wäre also eine *Verachtfachung* des Rechenaufwands.

3.5 Verfahren höherer Ordnung

Die Abschätzung (3.10) besagt, dass wir bei einer Halbierung des Zulässigkeitsparameters η lediglich auf eine Halbierung des Approximationsfehlers hoffen dürfen. Die Aufwandsabschätzung des vorangehenden Abschnitts legt nahe, dass dieser Schritt den Rechenaufwand erheblich in die Höhe treiben dürfte, also eher unattraktiv ist.

Einen möglichen Ausweg bieten Verfahren höherer Ordnung, bei denen sich der Fehler proportional zu η^m für ein $m > 1$ verhält, denn bei ihnen dürfte sich eine ausreichende Genauigkeit auch dann noch erreichen lassen, wenn η relativ groß ist, wir also nur relativ wenige Cluster berücksichtigen müssen.

Für die Konstruktion derartiger Verfahren bietet es sich an, unseren bisherigen Ansatz etwas abstrakter zu fassen: Wir bezeichnen die entscheidende Funktion mit

$$g(x, y) := \frac{y - x}{\|y - x\|_2^3}$$

und stellen fest, dass unsere Approximation von der Form

$$\sum_{j \in \hat{s}} g(x_i, x_j) m_j \approx g(x_i, x_s) m_s = g(x_i, x_s) \sum_{j \in \hat{s}} m_j$$

ist. Um die entscheidende Eigenschaft etwas deutlicher hervortreten zu lassen bezeichnen wir mit $\mathbf{1}$ die Funktion, die konstant gleich eins ist, und schreiben

$$\sum_{j \in \hat{s}} g(x_i, x_j) m_j \approx g(x_i, x_s) \sum_{j \in \hat{s}} \mathbf{1}(x_j) m_s.$$

Letztendlich haben wir also lediglich die Funktion g durch die Approximation

$$\tilde{g}(x, y) := g(x, x_s)\mathbf{1}(y)$$

ersetzt, deren entscheidende Eigenschaft darin besteht, dass x und y voneinander getrennt sind. Diese Eigenschaft können wir verallgemeinern:

Definition 3.12 (Entartete Kernfunktion) *Seien $t, s \subseteq \mathbb{R}^d$. Eine Funktion $\tilde{g} : t \times s \to \mathbb{R}^d$ nennen wir* entartete Kernfunktion*, falls*

$$\tilde{g}(x, y) = \sum_{\nu \in K} a_\nu(x) b_\nu(y) \qquad \text{für alle } x \in t,\ y \in s \tag{3.14}$$

gilt, wobei K eine endliche Indexmenge ist und

$$a_\nu : t \to \mathbb{R}^d, \qquad b_\nu : s \to \mathbb{R} \qquad \text{für alle } \nu \in K$$

geeignete Funktionen sind.

Gehen wir also davon aus, dass uns zwei Gebiete $t \subseteq \mathbb{R}^d$ („t“ steht für *target*) und $s \subseteq \mathbb{R}^d$ („s“ steht wie bisher für *source*) zur Verfügung stehen, denen Indexmengen $\hat{t}, \hat{s} \subseteq \mathcal{I}$ mit

$$x_i \in t, \qquad x_j \in s \qquad \text{für alle } i \in \hat{t},\ j \in \hat{s}$$

zugeordnet sind. Sei $\tilde{g}$ eine entartete Kernfunktion der Form (3.14). Dann gilt

$$\sum_{j \in \hat{s}} \tilde{g}(x_i, x_j) m_j = \sum_{j \in \hat{s}} \sum_{\nu \in K} a_\nu(x_i) b_\nu(x_j) m_j = \sum_{\nu \in K} a_\nu(x_i) \sum_{j \in \hat{s}} b_\nu(x_j) m_j \quad \text{für alle } i \in \hat{t}.$$

Indem wir die zweite Summe als „verallgemeinerte Ersatzmasse“

$$m_{s,\nu} := \sum_{j \in \hat{s}} b_\nu(x_j) m_j \qquad \text{für alle } \nu \in K$$

definieren, gelangen wir zu

$$\sum_{j \in \hat{s}} \tilde{g}(x_i, x_j) m_j = \sum_{\nu \in K} a_\nu(x_i) m_{s,\nu} \qquad \text{für alle } i \in \hat{t}.$$

Sofern in (3.14) nur wenige Summanden auftreten, sofern also die Mächtigkeit $\#K$ der Indexmenge deutlich geringer als $\#\hat{s}$ ist, lässt sich die entartete Kernfunktion $\tilde{g}$ sehr effizient auswerten.

Im Vergleich zu zuvor ersetzen wir die eine einzelne Ersatzmasse durch mehrere, die wir hoffentlich so geschickt wählen können, dass sie zu einer wesentlich besseren Approximation des Kraftfelds führen.

Eine besonders elegante Konstruktion beruht auf der *Interpolation*. Dabei handelt es sich um eine Methode, mit der sich aus einigen wenigen Werten einer Funktion die restlichen Werte näherungsweise bestimmen lassen. In der Mehrzahl der Fälle werden dazu Polynome verwendet. Wir bezeichnen den Raum der Polynome m-ter Ordnung mit

$$\Pi_m := \{\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_m x^m \ : \ \alpha_0, \ldots, \alpha_m \in \mathbb{R}\}.$$

Wir gehen davon aus, dass paarweise verschiedene Punkte $\xi_0, \ldots, \xi_m \in [a, b]$ gegeben sind. Wir nennen ein $p \in \Pi_m$ ein *Interpolationspolynom* einer Funktion $f \in C[a, b]$ in den *Interpolationspunkten* $\xi_0, \ldots, \xi_m$, falls

$$p(\xi_i) = f(\xi_i) \qquad \text{für alle } i \in [0 : m] \tag{3.15}$$

gilt.

Erinnerung 3.13 (Lagrange-Polynome) *Es gibt genau ein $p \in \Pi_m$, das die Gleichungen (3.15) erfüllt, und es lässt sich mit den* Lagrange-Polynomen

$$\ell_i(x) := \prod_{\substack{k=0 \\ k \neq i}}^{m} \frac{x - \xi_k}{\xi_i - \xi_k} \qquad \textit{für alle } i \in [0 : m],\ x \in \mathbb{R}$$

in der Form

$$p = \sum_{i=0}^{m} f(\xi_i) \ell_i \tag{3.16}$$

darstellen.

Für den Fall $d = 1$ würde (3.16) bereits eine Konstruktion einer entarteten Kernfunktion nahelegen: Wir könnten g approximieren, indem wir die Funktion $y \mapsto g(x, y)$ für ein festes x durch das Interpolationspolynom ersetzen:

$$\tilde{g}(x, y) := \sum_{j=0}^{m} g(x, \xi_j) \ell_j(y).$$

Dadurch sind x und y getrennt und wir haben unser Ziel erreicht.

Für den dreidimensionalen Fall brauchen wir auch eine mehrdimensionale Interpolation auf dem Quader

$$s = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3].$$

Zur Vereinheitlichung unserer Konstruktion wählen wir zunächst paarweise verschiedene Interpolationspunkte $\xi_0, \ldots, \xi_m \in [-1, 1]$ für das *Referenzintervall* $[-1, 1]$.

Interpolationspunkte in den Intervallen $[a_1, b_1]$, $[a_2, b_2]$ sowie $[a_3, b_3]$ können wir dann mit einer einfachen bijektiven linearen Transformation definieren:

$$\xi_{\iota,i} := \frac{b_\iota + a_\iota}{2} + \frac{b_\iota - a_\iota}{2}\xi_i \qquad \text{für alle } \iota \in \{1,2,3\},\ i \in [0:m].$$

Die entsprechenden Lagrange-Polynome sind durch

$$\ell_{\iota,i}(x) := \prod_{\substack{k=0 \\ k \neq i}}^{m} \frac{x - \xi_{\iota,k}}{\xi_{\iota,i} - \xi_{\iota,k}} \qquad \text{für alle } \iota \in \{1,2,3\},\ i \in [0:m],\ x \in \mathbb{R}$$

gegeben. Wir definieren *Interpolationsoperatoren*

$$\mathfrak{I}_\iota : C[a_\iota, b_\iota] \to \Pi_m, \qquad f \mapsto \sum_{i=0}^{m} f(\xi_{\iota,i})\ell_{\iota,i},$$

die beliebigen stetigen Funktionen ihre Interpolationspolynome zuordnen. Diese Operatoren besitzen zwei für uns wichtige Eigenschaften: Es existiert eine *Stabilitätskonstante* $\Lambda \in \mathbb{R}_{\geq 1}$ derart, dass

$$\|\mathfrak{I}_\iota[f]\|_{\infty,[a_\iota,b_\iota]} \leq \Lambda \|f\|_{\infty,[a_\iota,b_\iota]} \qquad \text{für alle } f \in C[a_\iota, b_\iota],\ \iota \in \{1,2,3\}+ \tag{3.17a}$$

gilt, und es existiert eine *Approximationskonstante* $\Psi \in \mathbb{R}_{>0}$ mit

$$\|f - \mathfrak{I}_\iota[f]\|_{\infty,[a_\iota,b_\iota]} \leq \Psi (b_\iota - a_\iota)^{m+1} \frac{\|f^{(m+1)}\|_{\infty,[a_\iota,b_\iota]}}{(m+1)!} \qquad \text{für alle } f \in C^{m+1}[a_\iota, b_\iota],\ \iota \in \{1,2,3\}. \tag{3.17b}$$

Um nun auf dem Quader s definierte Funktionen interpolieren zu können, führen wir *partielle Interpolationsoperatoren* ein, die jeweils in einer Koordinatenrichtung interpolieren und die beiden anderen unverändert lassen:

$$\mathfrak{I}_{s,1}[f](x) := \sum_{i=0}^{m} f(\xi_{1,i}, x_2, x_3)\ell_{1,i}(x_1),$$
$$\mathfrak{I}_{s,2}[f](x) := \sum_{i=0}^{m} f(x_1, \xi_{2,i}, x_3)\ell_{2,i}(x_2),$$

$$\mathfrak{I}_{s,3}[f](x) := \sum_{i=0}^{m} f(x_1, x_2, \xi_{3,i})\ell_{3,i}(x_3) \qquad \text{für alle } f \in C(s),\ x \in s.$$

Indem wir diese drei Operatoren hintereinander anwenden, erhalten wir mit

$$\mathfrak{I}_s := \mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2} \circ \mathfrak{I}_{s,3}$$

einen Operator, der

$$\mathfrak{I}_s[f](x) = \sum_{i=0}^{m}\sum_{j=0}^{m}\sum_{k=0}^{m} f(\xi_{1,i}, \xi_{2,j}, \xi_{3,k})\ell_{1,i}(x_1)\ell_{2,j}(x_2)\ell_{3,k}(x_3) \quad \text{für alle } f \in C(s),\ x \in s$$

erfüllt. Mit den Abkürzungen

$$\begin{aligned} M &:= \{(i,j,k)\ :\ i,j,k \in [0:m]\}, \\ \xi_{s,\nu} &:= (\xi_{1,i}, \xi_{2,j}, \xi_{3,k}) && \text{für alle } \nu = (i,j,k) \in M, \\ \ell_{s,\nu}(x) &:= \ell_{1,i}(x_1)\ell_{2,j}(x_2)\ell_{3,k}(x_3) && \text{für alle } \nu = (i,j,k) \in M,\ x \in s \end{aligned}$$

erhalten wir die kompakte Schreibweise

$$\mathfrak{I}_s[f] = \sum_{\nu \in M} f(\xi_{s,\nu})\ell_{s,\nu} \qquad \text{für alle } f \in C(s),$$

die die gewünschte Verallgemeinerung der eindimensionalen Interpolation darstellt. Da die Polynome $\ell_{s,\nu}$ nun Tensorprodukte eindimensionaler Polynome sind, spricht man von *Tensorinterpolation*.

Die Untersuchung des Approximationsfehlers können wir auf die Untersuchung der partiellen Operatoren $\mathfrak{I}_{s,\iota}$ zurückführen.

Lemma 3.14 (Partielle Interpolation) *Sei $\iota \in \{1,2,3\}$. Es gelten*

$$\|\mathfrak{I}_{s,\iota}[f]\|_{\infty,s} \leq \Lambda \|f\|_{\infty,s} \qquad \textit{für alle } f \in C(s), \tag{3.18a}$$

$$\|f - \mathfrak{I}_{s,\iota}[f]\|_{\infty,s} \leq \Psi (b_\iota - a_\iota)^{m+1} \frac{\|\partial_\iota^{m+1} f\|_{\infty,s}}{(m+1)!} \qquad \textit{für alle } f \in C^{m+1}(s). \tag{3.18b}$$

Beweis. Wir beschränken uns auf den Fall $\iota = 1$, die beiden anderen Fälle lassen sich ähnlich behandeln.

Sei $x \in s$, und sei

$$\hat{f} : [a_\iota, b_\iota] \to \mathbb{R}, \qquad y \mapsto f(y, x_2, x_3).$$

Dann gilt

$$\mathfrak{I}_{s,\iota}[f](x) = \sum_{i=0}^{m} f(\xi_{1,i}, x_2, x_3)\ell_{1,i}(x_1) = \sum_{i=0}^{m} \hat{f}(\xi_{1,i})\ell_{1,i}(x_1) = \mathfrak{I}_1[\hat{f}](x_1),$$

und mit (3.17a) folgt

$$|\mathfrak{I}_{s,\iota}[f](x)| = |\mathfrak{I}_1[\hat{f}](x_1)| \leq \Lambda \|\hat{f}\|_{\infty,[a_1,b_1]} \leq \Lambda \|f\|_{\infty,s}.$$

Da x beliebig gewählt ist, haben wir (3.18a) bewiesen.

Analog erhalten wir mit (3.17b)

$$\begin{aligned} |f(x) - \mathfrak{I}_{s,\iota}[f](x)| = |\hat{f}(x_1) - \mathfrak{I}_1[\hat{f}](x_1)| &\leq \Psi(b_1 - a_1)^{m+1} \frac{\|\hat{f}^{(m+1)}\|_{\infty,[a_1,b_1]}}{(m+1)!} \\ &\leq \Psi(b_1 - a_1)^{m+1} \frac{\|\partial_1^{m+1} f\|_{\infty,s}}{(m+1)!}, \end{aligned}$$

womit auch (3.18b) bewiesen ist. ∎

Satz 3.15 (Tensorinterpolation) *Es gelten*

$$\begin{aligned} \|\mathfrak{I}_s[f]\|_{\infty,s} &\leq \Lambda^3 \|f\|_{\infty,s} && \text{für alle } f \in C(s), \\ \|f - \mathfrak{I}_s[f]\|_{\infty,s} &\leq \sum_{\iota=1}^{3} \Psi \Lambda^{\iota-1} (b_\iota - a_\iota)^{m+1} \frac{\|\partial_\iota^{m+1} f\|_{\infty,s}}{(m+1)!} && \text{für alle } f \in C^{m+1}(s). \end{aligned}$$

Beweis. Mit (3.18a) erhalten wir direkt

$$\begin{aligned} \|\mathfrak{I}_{s,1}[f]\|_{\infty,s} &\leq \Lambda \|f\|_{\infty,s}, \\ \|\mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2}[f]\|_{\infty,s} &\leq \Lambda \|\mathfrak{I}_{s,2}[f]\|_{\infty,s} \leq \Lambda^2 \|f\|_{\infty,s}, \\ \|\mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2} \circ \mathfrak{I}_{s,3}[f]\|_{\infty,s} &\leq \Lambda^2 \|\mathfrak{I}_{s,3}[f]\|_{\infty,s} \leq \Lambda^3 \|f\|_{\infty,s} && \text{für alle } f \in C(s). \end{aligned}$$

Mit diesen Abschätzungen, (3.18b) und der Dreiecksungleichung finden wir außerdem

$$\begin{aligned} \|f - \mathfrak{I}_s[f]\|_{\infty,s} &\leq \|f - \mathfrak{I}_{s,1}[f]\|_{\infty,s} \\ &\quad + \|\mathfrak{I}_{s,1}[f] - \mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2}[f]\|_{\infty,s} \\ &\quad + \|\mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2}[f] - \mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2} \circ \mathfrak{I}_{s,3}[f]\|_{\infty,s} \\ &\leq \|f - \mathfrak{I}_{s,1}[f]\|_{\infty,s} + \|\mathfrak{I}_{s,1}[f - \mathfrak{I}_{s,2}[f]]\|_{\infty,s} + \|\mathfrak{I}_{s,1} \circ \mathfrak{I}_{s,2}[f - \mathfrak{I}_{s,3}[f]]\|_{\infty,s} \\ &\leq \|f - \mathfrak{I}_{s,1}[f]\|_{\infty,s} + \Lambda \|f - \mathfrak{I}_{s,2}[f]\|_{\infty,s} + \Lambda^2 \|f - \mathfrak{I}_{s,3}[f]\|_{\infty,s} \\ &\leq \sum_{\iota=1}^{3} \Psi \Lambda^{\iota-1} (b_\iota - a_\iota)^{m+1} \frac{\|\partial_\iota^{m+1} f\|_{\infty,s}}{(m+1)!} \qquad \text{für alle } f \in C^{m+1}(s). \end{aligned}$$

∎

In unserem konkreten Fall lässt sich nachrechnen, dass die Ableitungen der Funktion

$$f : s \to \mathbb{R}^3, \qquad y \mapsto g(x, y),$$

Abschätzungen der Form

$$\|\partial_\iota^{m+1} f\|_{\infty,s} \leq \frac{C}{\operatorname{dist}(x,s)^2} \frac{(m+1)!}{\operatorname{dist}(x,s)^{m+1}} \qquad \text{für alle } \iota \in \{1,2,3\},\ x \in \mathbb{R}^3 \setminus s$$

erfüllen, so dass wir schließlich für die Approximation

$$\tilde{g}(x,y) := \mathfrak{I}_s[f](y) = \sum_{\nu \in M} g(x, \xi_{s,\nu}) \ell_{s,\nu}(y)$$

die Abschätzung

$$\|g(x,y) - \tilde{g}(x,y)\|_2 \leq \frac{\widetilde{C}}{\operatorname{dist}(x,s)^2} \left(\frac{\operatorname{diam}(s)}{\operatorname{dist}(x,s)} \right)^{m+1}$$

erhalten. Mit der Zulässigkeitsbedingung (3.9) ergibt sich

$$\|g(x,y) - \tilde{g}(x,y)\|_2 \leq \frac{\widetilde{C}}{\operatorname{dist}(x,s)^2} 2^{m+1} \eta^{m+1},$$

eine Verkleinerung des Zulässigkeitsparameters η führt also zu einer Reduktion des Fehlers um einen Faktor von ungefähr η^{m+1}. Damit kann unser neuer Ansatz *wesentlich* schneller als der ursprüngliche konvergieren, wenn wir m hinreichend hoch wählen.

3.6 Symmetrisches Verfahren

Wir können unseren Algorithmus erheblich beschleunigen, indem wir Ersatzmassen nicht nur für die „Quellen“ der Kräfte verwenden, sondern auch für ihre Ziele.

Sei $t \in \mathcal{T}_\mathcal{I}$ ein Cluster und x_t sein Mittelpunkt. Sei $j \in \mathcal{T}_\mathcal{I}$ ein beliebiger Index. Indem wir die Rollen von x_i und x_j in Satz 3.4 tauschen erhalten wir

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_j - x_t}{\|x_j - x_t\|_2^3} \right\|_2 \leq 4 \frac{\operatorname{diam}(t)/2}{\operatorname{dist}(x_j, t)^3} \qquad \text{für alle } i \in \hat{t}$$

also können wir mit der entsprechend angepassten Zulässigkeitsbedingung

$$\operatorname{diam}(t) \leq 2\eta \operatorname{dist}(x_j, t) \tag{3.19}$$

zu der Abschätzung

$$\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_j - x_t}{\|x_j - x_z\|_2^3} \right\|_2 \leq \frac{4\eta}{\operatorname{dist}(x_j, t)^2} \qquad \text{für alle } i \in \hat{t}$$

gelangen. Statt die von einer Masse im Punkt x_j verursachte Kraft in allen Punkten x_i mit $i \in \hat{t}$ individuell zu berechnen können wir sie auch lediglich in dem Mittelpunkt x_t bestimmen und erhalten für ein hinreichend kleines η immer noch eine gute Näherung.

Mit Hilfe der Dreiecksungleichung können wir diesen Ansatz noch einen Schritt weiter treiben: Wir fordern die *starke Zulässigkeitsbedingung*

$$\max\{\operatorname{diam}(t), \operatorname{diam}(s)\} \leq 2\eta \operatorname{dist}(t,s),$$

bei der der Abstand zwischen t und s durch

$$\operatorname{dist}(t,s) := \min\{\operatorname{dist}(x,s) \ : \ x \in t\}$$

$$= \min\{\operatorname{dist}(y,t) \ : \ y \in s\} = \min\{\|x-y\|_2 \ : \ x \in t,\ y \in s\}$$

gegeben ist. Sie impliziert offenbar (3.9) und (3.19), so dass wir mit den bereits bekannten Argumenten zu

$$\begin{aligned}
\left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_t}{\|x_s - x_t\|_2^3} \right\|_2 &= \left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} + \frac{x_s - x_i}{\|x_s - x_i\|_2^3} - \frac{x_s - x_t}{\|x_s - x_t\|_2^3} \right\|_2 \\
&\leq \left\| \frac{x_j - x_i}{\|x_j - x_i\|_2^3} - \frac{x_s - x_i}{\|x_s - x_i\|_2^3} \right\|_2 \\
&\quad + \left\| \frac{x_s - x_i}{\|x_s - x_i\|_2^3} - \frac{x_s - x_t}{\|x_s - x_t\|_2^3} \right\|_2 \\
&\leq \frac{4\eta}{\operatorname{dist}(x_i,s)^2} + \frac{4\eta}{\operatorname{dist}(x_j,t)^2} \leq \frac{8\eta}{\operatorname{dist}(t,s)^2}
\end{aligned}$$

gelangen. Falls η hinreichend klein ist, können wir also die Wirkung *aller* x_j mit $j \in \hat{s}$ auf *alle* x_i mit $i \in \hat{t}$ näherungsweise durch die Wirkung von x_s auf x_t beschreiben: Das Kraftfeld ist für alle $x \in t$ und $y \in s$ im Wesentlichen konstant. Dadurch fällt für ein zulässiges Paar $t, s \in \mathcal{T}_{\mathcal{I}}$ von Clustern nur noch eine einzige Auswertung des Kraftfelds an, wogegen es für unser ursprüngliches Verfahren $\#\hat{t}$ gewesen wären.

4 Elektromagnetismus und lineare Gleichungssysteme

Bisher entstanden die in den von uns betrachteten Systemen auftretenden Kräfte durch die Gravitation oder durch die mechanische Auslenkung einer Feder. In diesem Kapitel widmen wir uns Kräften, die durch elektrische und magnetische Felder hervorgerufen werden.

Die Kraftfelder lassen sich dabei häufig nicht mehr durch eine kurze Formel beschreiben, sondern als Lösungen von *partiellen Differentialgleichungen*, die wir geeignet diskretisieren, also in lineare Gleichungssysteme überführen, müssen. Diese Gleichungssysteme werden in der Regel sehr groß sein, so dass einfache Standardalgorithmen wie die Gauß-Elimination oder die LR-Zerlegung überfordert sind und wir auf alternative Verfahren zurückgreifen müssen.

4.1 Lorentz-Kraft

Ein elektrisches Feld wird üblicherweise als Kraftfeld

$$E : \Omega \to \mathbb{R}^3$$

beschrieben, wobei $\Omega \subseteq \mathbb{R}^3$ das Gebiet ist, auf dem die Kraft wirkt. Für jeden Punkt $x \in \Omega$ gibt $E(x)$ die Kraft an, die auf ein Partikel mit einer *Ladung* von eins wirkt. Analog zu der Gravitationskraft, die sich proportional zu der Masse eines Körpers verhält, ist die elektrische Kraft proportional zu seiner Ladung. Auf ein Partikel mit der Ladung $q \in \mathbb{R}$ an der Stelle $x \in \Omega$ wirkt deshalb eine Kraft von $qE(x)$.

Bei magnetischen Feldern wird ein anderer Zugang verfolgt: Magnetische Felder üben nur eine Kraft auf *bewegte* Partikel aus, und diese Kraft hängt von der Geschwindigkeit ab. Um diesen Zusammenhang kompakt darstellen zu können benötigen wir einige einfache Hilfsmittel der Vektorrechnung.

Das *Kreuzprodukt* zweier Vektoren ist durch

$$x \times y = \begin{pmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{pmatrix} \qquad \text{für alle } x, y \in \mathbb{R}^3 \tag{4.1}$$

definiert. Seine wichtigsten Eigenschaften können wir knapp zusammenfassen, indem wir auf das euklidische Skalarprodukt (siehe Erinnerung 3.3) zurückgreifen.

Mit Hilfe des Skalarprodukts lässt sich der *Winkel* zwischen Vektoren durch

$$\frac{\langle x, y\rangle_2}{\|x\|_2\|y\|_2} = \cos \angle(x, y) \qquad \text{für alle } x, y \in \mathbb{R}^n \setminus \{0\}$$

beschreiben. Insbesondere ist das Skalarprodukt zweier Vektoren gleich null, falls die Vektoren senkrecht aufeinander stehen, denn dann ist der Winkel $\pm\pi/2$ und sein Cosinus verschwindet.

Mit Hilfe des Skalarprodukts können wir eine Beziehung zwischen dem Kreuzprodukt und der *Determinante* herstellen.

Erinnerung 4.1 (Determinante) *Die* Determinante *zweier zweidimensionaler Vektoren ist durch*

$$\det(x,y) = x_1y_2 - x_2y_1 \qquad \text{für alle } x,y \in \mathbb{R}^2 \tag{4.2}$$

gegeben. Mit Hilfe des Cavalierischen Prinzips *kann man folgern, dass* $|\det(x,y)|$ *gerade die Fläche des von* x *und* y *aufgespannten Parallelogramms ist.*

Die Determinante dreier dreidimensionaler Vektoren ist durch den Laplaceschen Entwicklungssatz *als*

$$\det(x,y,z) = x_1 \det\begin{pmatrix} y_2 & z_2 \\ y_3 & z_3 \end{pmatrix} - x_2 \det\begin{pmatrix} y_1 & z_1 \\ y_3 & z_3 \end{pmatrix} + x_3 \det\begin{pmatrix} y_1 & z_1 \\ y_2 & z_2 \end{pmatrix} \tag{4.3}$$
$$\text{für alle } x,y,z \in \mathbb{R}^3$$

gegeben. Wieder mit Hilfe des Cavalierischen Prinzips folgt, dass $|\det(x,y,z)|$ *das Volumen des von* x*,* y *und* z *aufgespannten Parallelepipeds (oder Spats) ist.*

Die Determinante ist eine Multilinearform*, also linear in jedem Argument. Sie ist auch* alternierend*, sie ändert also ihr Vorzeichen, wenn zwei Argumente vertauscht werden.*

Aus dieser letzten Eigenschaft folgt, dass die Determinante gleich null ist, falls zwei ihrer Argumente gleich sind. Es folgt

$$\begin{aligned}\det(\alpha x + \beta y, x, y) &= \det(\alpha x, x, y) + \det(\beta y, x, y) \\ &= \alpha \det(x,x,y) + \beta \det(y,x,y) = 0 \quad \text{für alle } x,y \in \mathbb{R}^3,\ \alpha,\beta \in \mathbb{R},\end{aligned}$$

die Determinante linear abhängiger Vektoren ist also gleich null. Das entspricht durchaus der Anschauung: In diesem Fall ist das Parallelepiped zu einer zweidimensionalen Fläche entartet, deren Volumen gleich null ist.

Mit Hilfe des Skalarprodukts können wir eine Beziehung zwischen dem Kreuzprodukt und der Determinante herstellen.

Lemma 4.2 (Determinante) *Es gilt*

$$\langle x, y \times z\rangle_2 = \det(x,y,z) \qquad \text{für alle } x,y,z \in \mathbb{R}^3.$$

Insbesondere steht $y \times z$ *senkrecht auf* y *und* z*.*

Beweis. Mit der Definition des Kreuzprodukts erhalten wir

$$\langle x, y \times z\rangle_2 = x_1(y \times z)_1 + x_2(y \times z)_2 + x_3(y \times z)_3$$

$$
\begin{aligned}
&= x_1(y_2z_3 - y_3z_2) + x_2(y_3z_1 - y_1z_3) + x_3(y_1z_2 - y_2z_1) \\
&= x_1(y_2z_3 - y_3z_2) - x_2(y_1z_3 - y_3z_1) + x_3(y_1z_2 - y_2z_1) \\
&= x_1 \det \begin{pmatrix} y_2 & z_2 \\ y_3 & z_3 \end{pmatrix} - x_2 \det \begin{pmatrix} y_1 & z_1 \\ y_3 & z_3 \end{pmatrix} + x_3 \det \begin{pmatrix} y_1 & z_1 \\ y_2 & z_2 \end{pmatrix} \\
&= \det \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix} = \det(x, y, z),
\end{aligned}
$$

wobei wir in der vierten Zeile die Definition (4.2) der zweidimensionalen Determinante und im letzten Schritt den Laplaceschen Entwicklungssatz (4.3) verwendet haben. ∎

Für uns ist von entscheidender Bedeutung, dass das Kreuzprodukt zweier Vektoren senkrecht auf beiden Vektoren steht, denn das magnetische Feld hat gerade die Eigenschaft, dass es eine Kraft auf ein bewegtes geladenes Partikel ausübt, die senkrecht zu seiner Bewegungsrichtung wirkt, also senkrecht auf dem Geschwindigkeitsvektor steht.

Das magnetische Feld wird wie das elektrische durch eine Abbildung

$$B : \Omega \to \mathbb{R}^3$$

beschrieben, diese Abbildung wird allerdings anders als bisher interpretiert: Auf ein Partikel an der Position $x \in \Omega$ mit der Geschwindigkeit $v \in \mathbb{R}^3$ und der Ladung $q \in \mathbb{R}$ übt das magnetische Feld eine Kraft von $q(v \times B(x))$ aus. Die Kraft ist also senkrecht zu der Geschwindigkeit und dem Feld $B(x)$, und sie wirkt um so stärker, je höher die Geschwindigkeit oder die Ladung ist.

Definition 4.3 (Lorentz-Kraft) *Sei $\Omega \subseteq \mathbb{R}^3$ ein Gebiet, seien*

$$E : \Omega \to \mathbb{R}^3, \qquad B : \Omega \to \mathbb{R}^3$$

das elektrische und das magnetische Feld. Die auf ein Partikel am Ort $x \in \Omega$ mit Geschwindigkeit $v \in \mathbb{R}^3$ und Ladung $q \in \mathbb{R}$ wirkende Lorentz-Kraft *ist gegeben durch*

$$f := qE(x) + q(v \times B(x)) = q(E(x) + v \times B(x)). \tag{4.4}$$

Wenn und E und B gegeben sind, können wir mit Hilfe dieser Gleichung in der bereits bekannten Weise die Flugbahn eines geladenen Partikels durch eine gewöhnliche Differentialgleichung beschreiben.

4.2 Maxwell-Gleichungen

Das magnetische und das elektrische Feld, die gemeinsam die Bewegung geladener Partikel beschreiben, sind durch ein System gekoppelter Differentialgleichungen gegeben, die *Maxwell-Gleichungen*.

Diese Gleichungen stellen Bezüge zwischen *Wirbeln* und *Quellen* der Felder her. Beispielsweise verursacht ein bewegter Magnet einen Wirbel im elektrischen Feld, der sich

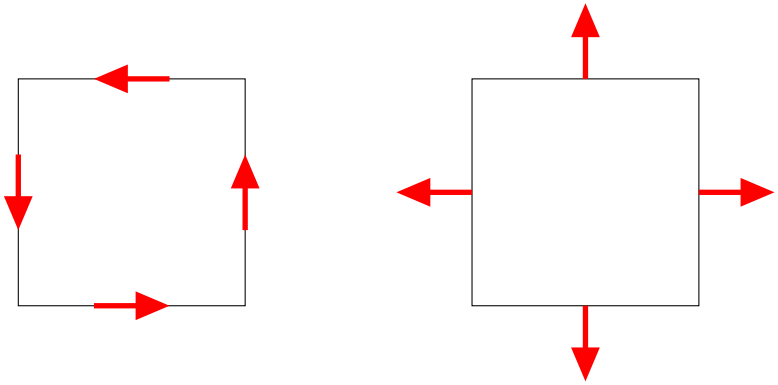

Abbildung 4.1: Mathematische Beschreibung von Wirbeln und Quellen.

bei der Konstruktion eines Dynamos verwenden lässt. Entsprechend verursacht ein fließender Strom einen Wirbel im magnetischen Feld, so dass wir einen Elektromagneten konstruieren können.

Um Wirbel und Quellen mathematisch präzise beschreiben zu können benötigen wir als Hilfsmittel aus der Analysis *Mittelwertsätze*.

Erinnerung 4.4 (Mittelwertsätze) *Sei $g : [a, b] \to \mathbb{R}$ eine stetig differenzierbare Funktion. Dann existiert ein $\eta \in [a, b]$ so, dass*

$$\frac{g(b) - g(a)}{b - a} = g'(\eta)$$

gilt. In mindestens einem Punkt zwischen a und b nimmt also die Ableitung exakt den Wert des Differenzenquotienten an.

Sei $g : [a, b] \to \mathbb{R}$ eine stetige Funktion. Dann existiert ein $\eta \in [a, b]$ so, dass

$$\frac{1}{b - a} \int_a^b g(x)\, dx = g(\eta)$$

gilt. In mindestens einem Punkt zwischen a und b nimmt also die Funktion ihren Integral-Mittelwert an.

Unser erstes Ziel ist es, Wirbel in einem Feld $u : \Omega \to \mathbb{R}^3$ um einen Punkt $x \in \Omega$ mathematisch präzise zu fassen.

Der Einfachheit halber beschränken wir uns auf den zweidimensionalen Fall und untersuchen nur Wirbel um den Nullpunkt $x = 0$. Falls ein Wirbel um x existiert, müsste ein Partikel, das sich auf einer Bahn um x bewegt, durch das Feld beschleunigt werden.

Wir verwenden als Bahn den Rand des Quadrats $[-h, h] \times [-h, h]$, den wir im mathematisch positiven Sinn, also gegen den Uhrzeigersinn, durchlaufen. Auf dem Wegstück von $[-h, -h]$ zu $[h, -h]$ bewegen wir uns in positiver x_1-Richtung, so dass nur die erste Komponente u_1 des Felds von Interesse ist. Auf dem Stück von $[h, -h]$ zu $[h, h]$ laufen

wir in positiver x_2-Richtung, so dass nur die zweite Komponente zählt. Auf dem Stück von $[h,h]$ zu $[-h,h]$ ist es wieder die erste Komponente, allerdings wegen der umgekehrten Richtung auch mit negativem Vorzeichen. Auf dem letzten Stück von $[-h,h]$ zu $[h,h]$ ist es entsprechend die zweite Komponente mit negativem Vorzeichen. Indem wir zu Integralen übergehen und durch die Fläche des Quadrats dividieren erhalten wir den Ausdruck

$$\frac{1}{4h^2}\left(\int_{-h}^{h} u_1(x_1,-h)\,dx_1 + \int_{-h}^{h} u_2(h,x_2)\,dx_2 - \int_{-h}^{h} u_1(x_1,h)\,dx_1 - \int_{-h}^{h} u_2(-h,x_2)\,dx_2\right).$$

Indem wir die x_1- und die x_2-Integrale zusammenfassen und die Mittelwertsätze anwenden erhalten wir

$$\begin{aligned}
\frac{1}{4h^2}\int_{-h}^{h} u_1(x_1,-h) - u_1(x_1,h)\,dx_1 &= \frac{2h}{4h^2}(u_1(\eta_1,-h) - u_1(\eta_1,h)) \\
&= \frac{u_1(\eta_1,-h) - u_1(\eta_1,h)}{2h} = -\partial_2 u_1(\eta_1,\eta_2), \\
\frac{1}{4h^2}\int_{-h}^{h} u_2(h,x_2) - u_2(-h,x_2)\,dx_2 &= \frac{2h}{4h^2}(u_2(h,\eta_3) - u_2(-h,\eta_3)) \\
&= \frac{u_2(h,\eta_3) - u_2(-h,\eta_3)}{2h} = \partial_1 u_2(\eta_4,\eta_3)
\end{aligned}$$

mit $\eta_1,\eta_2,\eta_3,\eta_4 \in [-h,h]$. Mit dem Grenzübergang $h \to 0$ ergibt sich die Größe

$$\lim_{h\to 0}\frac{1}{4h^2}\left(\int_{-h}^{h} u_1(x_1,-h)\,dx_1 + \int_{-h}^{h} u_2(h,x_2)\,dx_2 - \int_{-h}^{h} u_1(x_1,h)\,dx_1 - \int_{-h}^{h} u_2(-h,x_2)\,dx_2\right) = \partial_1 u_2(0,0) - \partial_2 u_1(0,0),$$

die angibt, wie stark das Feld lokal um den Nullpunkt „rotiert“.

Im dreidimensionalen Fall führen wir eine entsprechende Rechnung in jeder Koordinatenebene durch und erhalten die *Rotation* eines Vektorfelds.

Definition 4.5 (Rotation) *Sei $u : \Omega \to \mathbb{R}^3$ stetig differenzierbar. Wir bezeichnen mit*

$$\nabla \times u(x) := \begin{pmatrix} \partial_2 u_3(x) - \partial_3 u_2(x) \\ \partial_3 u_1(x) - \partial_1 u_3(x) \\ \partial_1 u_2(x) - \partial_2 u_1(x) \end{pmatrix} \qquad \text{für alle } x \in \Omega$$

die Rotation *des Felds u im Punkt x.*

Mit Hilfe der Rotation können wir Wirbel in unserem Feld beschreiben. Im nächsten Schritt wollen wir auch Quellen des Felds mathematisch präzise beschreiben. Falls in

einem Punkt $x \in \Omega$ etwas in das Gebiet „hineinfließt“, werden Partikel in benachbarten Punkten „nach außen“ gedrückt.

Mathematisch können wir diese Wirkung messen, indem wir wieder ein kleines Volumen um den Punkt herum legen und untersuchen, wie Partikel auf dem Rand aus dem Gebiet heraus bewegt werden. Wir beschränken uns wieder auf den Nullpunkt und das Quadrat $[-h,h] \times [-h,h]$. Auf der Seite $[-h,h] \times \{-h\}$ interessiert uns die Stärke des Felds in negativer x_2-Richtung, also aus dem Quadrat hinaus. Wir sollten also $-u_2$ betrachten. Auf der Seite $\{h\} \times [-h,h]$ ist die positive x_1-Richtung von Interesse, auf der Seite $[-h,h] \times \{h\}$ die positive x_2-Richtung und auf der Seite $\{-h\} \times [-h,h]$ die negative x_1-Richtung. Indem wir wieder durch die Fläche dividieren erhalten wir

$$\frac{1}{4h^2}\left(-\int_{-h}^{h} u_2(x_1,-h)\,dx_1 + \int_{-h}^{h} u_1(h,x_2)\,dx_2 + \int_{-h}^{h} u_2(x_1,h)\,dx_1 - \int_{-h}^{h} u_1(-h,x_2)\,dx_2\right).$$

Analog zu unserer vorangehenden Betrachtung wenden wir wieder die Mittelwertsätze an, um

$$\begin{aligned}
\frac{1}{4h^2}\int_{-h}^{h} u_2(x_1,h) - u_2(x_1,-h)\,dx_1 &= \frac{2h}{4h^2}(u_2(\eta_1,h) - u_2(\eta_1,-h)) \\
&= \frac{u_2(\eta_1,h) - u_2(\eta_1,-h)}{2h} = \partial_2 u_2(\eta_1,\eta_2), \\
\frac{1}{4h^2}\int_{-h}^{h} u_1(h,x_2) - u_1(-h,x_2)\,dx_2 &= \frac{2h}{4h^2}(u_1(h,\eta_3) - u_1(-h,\eta_3)) \\
&= \frac{u_1(h,\eta_3) - u_1(-h,\eta_3)}{2h} = \partial_1 u_1(\eta_4,\eta_3)
\end{aligned}$$

mit $\eta_1, \eta_2, \eta_3, \eta_4 \in [-h,h]$ zu erhalten. Mit dem Grenzübergang $h \to 0$ ergibt sich die Größe

$$\lim_{h\to 0} \frac{1}{4h^2}\left(-\int_{-h}^{h} u_2(x_1,-h)\,dx_1 + \int_{-h}^{h} u_1(h,x_2)\,dx_2 + \int_{-h}^{h} u_2(x_1,h)\,dx_1 - \int_{-h}^{h} u_1(-h,x_2)\,dx_2\right) = \partial_1 u_1(0) + \partial_2 u_2(0),$$

mit der wir beschreiben können, wieviel im Nullpunkt in das Gebiet „hineinfließt“.

Im dreidimensionalen Fall kommt lediglich ein weiterer Term für die dritte Koordinate hinzu und wir erhalten die *Divergenz* eines Vektorfelds.

Definition 4.6 (Divergenz) *Sei $u : \Omega \to \mathbb{R}^3$ stetig differenzierbar. Wir bezeichnen mit*

$$\nabla \cdot u(x) := \partial_1 u_1(x) + \partial_2 u_2(x) + \partial_3 u_3(x) \qquad \text{für alle } x \in \Omega$$

die Divergenz *des Felds u im Punkt x.*

Bemerkung 4.7 (Nabla-Notation) *Die Schreibweisen für Rotation und Divergenz folgen der* Nabla-Notation*: Wir definieren den „Vektor“*

$$\nabla := \begin{pmatrix} \partial_1 \\ \partial_2 \\ \partial_3 \end{pmatrix}$$

aus partiellen Ableitungen (das Symbol ∇ wird Nabla *genannt).*

Wenn wir formal das Kreuzprodukt mit einem Feld u berechnen erhalten wir

$$\nabla \times u = \begin{pmatrix} \partial_2 u_3 - \partial_3 u_2 \\ \partial_3 u_1 - \partial_1 u_3 \\ \partial_1 u_2 - \partial_2 u_1 \end{pmatrix},$$

also gerade die Rotation, während wir für das Skalarprodukt (das manchmal auch als $x \cdot y$ geschrieben wird) gerade

$$\nabla \cdot u = \partial_1 u_1 + \partial_2 u_2 + \partial_3 u_3$$

erhalten, also die Divergenz.

Mit Rotation und Divergenz steht uns nun alles zur Verfügung, was für die Beschreibung des elektrischen und des magnetischen Felds erforderlich ist. Da sich beide Felder mit der Zeit ändern können beschreiben wir sie durch Abbildungen

$$E : [a,b] \times \Omega \to \mathbb{R}^3, \qquad B : [a,b] \times \Omega \to \mathbb{R}^3,$$

bei denen die erste Komponente jeweils die Zeit angibt.

Faraday'sches Gesetz. Das *Faraday'sche Gesetz* besagt, dass eine Veränderung des magnetischen Felds einen Wirbel im elektrischen Feld hervorruft. Wir können es mathematisch präzise durch

$$\nabla \times E(t,x) = -\frac{\partial B}{\partial t}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega \tag{4.5}$$

beschreiben. Ein Beispiel für seine Anwendung ist der Dynamo: Ein sich bewegender Magnet ruft einen Wirbel im elektrischen Feld hervor, durch den Ladungsträger in Bewegung gesetzt werden, so dass ein Strom fließt.

Ampère'sches Gesetz. Das *Ampère'sche Gesetz* besagt, dass eine Veränderung des elektrischen Felds einen Wirbel im magnetischen Feld hervorruft. Es ist durch die Gleichung

$$\nabla \times \left(\frac{1}{\mu} B\right)(t,x) = \left(j(t,x) + \epsilon \frac{\partial E}{\partial t}(t,x)\right) \qquad \text{für alle } t \in [a,b],\ x \in \Omega \tag{4.6}$$

gegeben, in der j die *Stromdichte* beschreibt, $\mu \in \mathbb{R}_{>0}$ die *magnetische Permeabilität* und $\epsilon \in \mathbb{R}_{>0}$ die *Dielektrizität* des Materials.

Die Stromdichte ordnet jedem Punkt im Raum eine Richtung zu, in die in ihm ein Strom fließt, in der also Elektronen transportiert werden.

Falls ein Material nicht leitfähig ist, wirkt das elektrische Feld trotzdem auf die Elektronen innerhalb der einzelnen Atome, die sich so ausrichten, dass sie das Feld innerhalb des Materials abschwächen. Diesen Effekt beschreibt die Dielektrizität.

Ein Material kann auch auf das magnetische Feld reagieren, denn man kann sich anschaulich vorstellen, dass Elektronen in den einzelnen Atomen rotieren, so dass sie durch das magnetische Feld beeinflusst werden und auch selbst ein magnetisches Feld hervorrufen. Die Permeabilität beschreibt, ob dieser Effekt zu einer Verstärkung (beispielsweise bei ferromagnetischen Stoffen) oder Schwächung des Magnetfelds führt.

Gauß'sches Gesetz für elektrische Felder. Das *Gauß'sche Gesetz für elektrische Felder* beschreibt, dass stationäre Ladungsträger Quellen des elektrischen Felds sind. Es ist durch die Gleichung

$$\nabla \cdot (\epsilon E)(t,x) = \varrho(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega \tag{4.7}$$

gegeben, in der ϱ die *Ladungsdichte* beschreibt, also jedem Punkt des Raums zuordnet, wieviele Ladungsträger sich (im Mittel) in ihm aufhalten.

Gauß'sches Gesetz für magnetische Felder. Das *Gauß'sche Gesetz für magnetische Felder* schließlich besagt, dass magnetische Felder keine Quellen besitzen können, dass also

$$\nabla \cdot B(t,x) = 0 \qquad \text{für alle } t \in [a,b],\ x \in \Omega \tag{4.8}$$

gilt. Während es sehr wohl Körper gibt, die nur positiv oder nur negativ geladen sind, gibt es nach diesem Gesetz keine, die nur einen magnetischen Nord- oder Südpol aufweisen.

4.3 Elektromagnetische Wellen

Eine wichtige Eigenschaft elektromagnetischer Felder besteht darin, dass sich in ihnen Wellen ausbreiten können. Anders als bei den von uns bisher behandelten mechanischen Wellen ist bei elektromagnetischen Wellen kein Medium (kein „Netz aus Federn“) erforderlich, durch die die Wellen übertragen werden.

Um zu einer mathematischen Beschreibung dieses Phänomens zu gelangen, gehen wir von dem Ampére'schen Gesetz (4.6) aus. Wenn kein zusätzlicher Strom fließt nimmt es die kurze Form

$$\nabla \times B(t,x) = \mu\epsilon \frac{\partial E}{\partial t}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega$$

an. Wir differenzieren diese Gleichung nach der Zeit und erhalten

$$\nabla \times \frac{\partial B}{\partial t}(t,x) = \mu\epsilon \frac{\partial^2 E}{\partial t^2}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega.$$

Nach dem Faraday'schen Gesetz (4.5) ist die Zeitableitung des magnetischen Felds durch

$$-\nabla \times E(t,x) = \frac{\partial B}{\partial t}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega$$

gegeben, und durch Einsetzen erhalten wir

$$-\nabla \times \nabla \times E(t,x) = \mu\epsilon \frac{\partial^2 E}{\partial t^2}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega.$$

Den doppelten Rotationsoperator können wir etwas vereinfachen:

$$\begin{aligned}
\nabla \times \nabla \times E(t,x) &= \begin{pmatrix} \partial_2(\nabla \times E(t,x))_3 - \partial_3(\nabla \times E(t,x))_2 \\ \partial_3(\nabla \times E(t,x))_1 - \partial_1(\nabla \times E(t,x))_3 \\ \partial_1(\nabla \times E(t,x))_2 - \partial_2(\nabla \times E(t,x))_1 \end{pmatrix} \\
&= \begin{pmatrix} \partial_2(\partial_1 E_2(t,x) - \partial_2 E_1(t,x)) - \partial_3(\partial_3 E_1(t,x) - \partial_1 E_3(t,x)) \\ \partial_3(\partial_2 E_3(t,x) - \partial_3 E_2(t,x)) - \partial_1(\partial_1 E_2(t,x) - \partial_2 E_1(t,x)) \\ \partial_1(\partial_3 E_1(t,x) - \partial_1 E_3(t,x)) - \partial_2(\partial_2 E_3(t,x) - \partial_3 E_2(t,x)) \end{pmatrix} \\
&= \begin{pmatrix} \partial_1(\partial_2 E_2(t,x) + \partial_3 E_3(t,x)) - \partial_2^2 E_1(t,x) - \partial_3^2 E_1(t,x) \\ \partial_2(\partial_3 E_3(t,x) + \partial_1 E_1(t,x)) - \partial_3^2 E_2(t,x) - \partial_1^2 E_2(t,x) \\ \partial_3(\partial_1 E_1(t,x) + \partial_2 E_2(t,x)) - \partial_1^2 E_3(t,x) - \partial_2^2 E_3(t,x) \end{pmatrix} \\
&= \begin{pmatrix} \partial_1(\nabla \cdot E(t,x)) - \partial_1^2 E_1(t,x) - \partial_2^2 E_1(t,x) - \partial_3^2 E_1(t,x) \\ \partial_2(\nabla \cdot E(t,x)) - \partial_1^2 E_2(t,x) - \partial_2^2 E_2(t,x) - \partial_3^2 E_2(t,x) \\ \partial_3(\nabla \cdot E(t,x)) - \partial_1^2 E_3(t,x) - \partial_2^2 E_3(t,x) - \partial_3^2 E_3(t,x) \end{pmatrix}. \qquad (4.9)
\end{aligned}$$

Wenn wir davon ausgehen, dass keine elektrischen Ladungen vorhanden sind, folgt aus dem Gauß'schen Gesetz für elektrische Felder (4.7)

$$\nabla \cdot E(t,x) = 0 \qquad \text{für alle } t \in [a,b],\ x \in \Omega,$$

so dass sich unsere Gleichung auf

$$\partial_1^2 E(t,x) + \partial_2^2 E(t,x) + \partial_3^2 E(t,x) = \mu\epsilon \frac{\partial^2 E}{\partial t^2}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega \qquad (4.10)$$

reduziert. Diese Gleichung weist eine enge Verwandtschaft mit der bereits in Übungsaufgabe 2.16 angesprochenen Wellengleichung auf, allerdings treten im dreidimensionalen Fall auf der linken Seite die zweiten Ableitungen in allen drei Koordinatenrichtungen auf. Für das magnetische Feld lässt sich eine sehr ähnliche Gleichung herleiten, wenn man im letzten Schritt das Gauß'sche Gesetz für das magnetische Feld verwendet.

Anders als bei den bisher von uns betrachteten Gleichungen handelt es sich bei der *dreidimensionalen Wellengleichung* (4.10) um eine *partielle Differentialgleichung*,

in der sowohl Zeit- als auch Ortsableitungen auftreten. Die von uns bisher behandelten Lösungstechniken lassen sich auf derartige Gleichungen nicht ohne weiteres anwenden.

Wir können allerdings versuchen, die Gleichung in die Form einer gewöhnlichen Differentialgleichung zu bringen, schließlich tritt auf der rechten Seite bereits eine Zeitableitung auf. Allerdings stoßen wir dabei auf das Problem, dass der Zustand, indem sich das elektrische Feld befindet, aus einem unendlich-dimensionalen Raum stammt: In jedem der unendlich vielen Punkte $x \in \Omega$ kann $E(t,x)$ unabhängig von den umliegenden Punkten Werte annehmen. Wir können aber nicht unendliche viele Werte $E(t,x)$ abspeichern.

Eine Lösung bietet eine *Diskretisierung* der Gleichung: Wir wählen in geschickter Weise *endlich* viele Punkte aus, durch die sich das Feld $E(t,x)$ noch hinreichend gut approximieren lässt. Es gibt verschiedene Diskretisierungstechniken, für unsere Zwecke genügt zunächst das *Finite-Differenzen-Verfahren.*

Zur Vereinfachung gehen wir davon aus, dass Ω der Einheitswürfel

$$\Omega = \{x \in \mathbb{R}^3 \ : \ x_1, x_2, x_3 \in (0,1)\} = (0,1)^3$$

ist. Wir ersetzen die unendlich vielen Punkte in Ω durch endlich viele Punkte, indem wir ein $n \in \mathbb{N}$ wählen und

$$h := \frac{1}{n+1}, \qquad x_i := \begin{pmatrix} hi_1 \\ hi_2 \\ hi_3 \end{pmatrix} = hi \qquad \text{für alle } i \in \bar{\mathcal{I}} := \{0, \dots, n+1\}^3$$

setzen. Dann ist

$$\mathcal{I} := \{i \in \bar{\mathcal{I}} \ : \ x_i \in \Omega\} = \{1, \dots, n\}^3$$

gerade die Indexmenge der Punkte, die in Ω liegen. Statt auf dem Gebiet Ω rechnen wir auf dem *Gitter*

$$\Omega_h := \{x_i \ : \ i \in \mathcal{I}\},$$

das nur aus n^3 Punkten besteht, so dass wir eine *Gitterfunktion*

$$u_h : \Omega_h \to \mathbb{R}$$

durch n^3 Werte beschreiben können.

Beispielsweise könnten wir das elektrische Feld $E : \Omega \to \mathbb{R}^3$ durch die Gitterfunktion $E_h : \Omega_h \to \mathbb{R}^3$ ersetzen, die durch

$$E_h(t,x) = E(t,x) \qquad \text{für alle } t \in [a,b], \ x \in \Omega_h$$

definiert ist. Die $3n^3$ Werte, durch die $E_h(t,\cdot)$ beschrieben ist, könnten wir dann mit den uns bereits bekannten Techniken berechnen, beispielsweise mit dem effizienten Leapfrog-Verfahren.

Allerdings tritt dabei eine entscheidende Schwierigkeit auf: Auf der linken Seite der Wellengleichung (4.10) treten Ableitungen der Funktion E auf, und Ableitungen sind als Grenzwerte für gegen einen Punkt konvergierende Folgen definiert. Da die Punkte in

Ω_h jeweils einen Mindestabstand von h aufweisen, können wir Ableitungen von E_h nicht berechnen.

Wir können allerdings *Näherungen* der Ableitungen gewinnen, indem wir geeignete Differenzenquotienten verwenden und analog zu Lemma 2.6 vorgehen, um deren Genauigkeit zu analysieren.

Lemma 4.8 (Zweiter Differenzenquotient) *Seien $x \in \mathbb{R}$ und $h \in \mathbb{R}_{>0}$ gegeben. Falls $g : [x-h, x+h] \to \mathbb{R}$ viermal stetig differenzierbar ist, gilt*

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + \frac{h^2}{12} g^{(4)}(\eta) \tag{4.11}$$

mit einem $\eta \in [x-h, x+h]$.

Beweis. Wir verwenden die Taylor-Entwicklung (vgl. Erinnerung 2.5), um $\eta_+ \in [x, x+h]$ und $\eta_- \in [x-h, x]$ mit

$$\begin{aligned} g(x+h) &= g(x) + hg'(x) + \frac{h^2}{2} g''(x) + \frac{h^3}{6} g'''(x) + \frac{h^4}{24} g^{(4)}(\eta_+), \\ g(x-h) &= g(x) - hg'(x) + \frac{h^2}{2} g''(x) - \frac{h^3}{6} g'''(x) + \frac{h^4}{24} g^{(4)}(\eta_-) \end{aligned}$$

zu finden. Indem wir beide Gleichungen addieren und $2g(x)$ subtrahieren ergibt sich

$$\begin{aligned} g(x+h) + g(x-h) &= 2g(x) + h^2 g''(x) + \frac{h^4}{24}(g^{(4)}(\eta_+) + g^{(4)}(\eta_-)), \\ g(x+h) - 2g(x) + g(x-h) &= h^2 g''(x) + \frac{h^4}{12} \frac{g^{(4)}(\eta_+) + g^{(4)}(\eta_-)}{2}. \end{aligned}$$

Da $g^{(4)}$ eine stetige Funktion ist, finden wir mit Hilfe des Zwischenwertsatzes (vgl. Erinnerung 2.4) ein $\eta \in [\eta_-, \eta_+]$ mit

$$\frac{g^{(4)}(\eta_+) + g^{(4)}(\eta_-)}{2} = g^{(4)}(\eta),$$

so dass wir schließlich zu

$$\begin{aligned} g(x+h) - 2g(x) + g(x-h) &= h^2 g''(x) + \frac{h^4}{12} g^{(4)}(\eta), \\ \frac{g(x+h) - 2g(x) + g(x-h)}{h^2} &= g''(x) + \frac{h^2}{12} g^{(4)}(\eta) \end{aligned}$$

gelangen. Das ist die gewünschte Gleichung. ∎

In der Wellengleichung (4.10) treten partielle Ableitungen in den drei Koordinatenrichtungen auf, für die wir mit diesem Lemma die Gleichungen

$$\frac{E(x_1+h, x_2, x_3) - 2E(x) + E(x_1-h, x_2, x_3)}{h^2} = \partial_1^2 E(x) + \frac{h^2}{12} \partial_1^4 E(\eta_1, x_2, x_3),$$

$$\frac{E(x_1, x_2 + h, x_3) - 2E(x) + E(x_1, x_2 - h, x_3)}{h^2} = \partial_2^2 E(x) + \frac{h^2}{12}\partial_2^4 E(x_1, \eta_2, x_3),$$
$$\frac{E(x_1, x_2, x_3 + h) - 2E(x) + E(x_1, x_2, x_3 - h)}{h^2} = \partial_3^2 E(x) + \frac{h^2}{12}\partial_3^4 E(x_1, x_2, \eta_3)$$

mit $\eta_1 \in [x_1 - h, x_1 + h]$, $\eta_2 \in [x_2 - h, x_2 + h]$ und $\eta_3 \in [x_3 - h, x_3 + h]$ erhalten. Zur Abkürzung wurde hier die Zeitvariable weggelassen, da sie für die Approximation nicht von Bedeutung ist.

Falls wir davon ausgehen, dass wir ein hinreichend feines Punktegitter verwenden, dass also h^2 hinreichend klein ist, dürfen wir den letzten Term in jeder dieser Gleichungen vernachlässigen und gelangen zu

$$\begin{aligned}\frac{1}{h^2}(&E(x_1 + h, x_2, x_3) + E(x_1 - h, x_2, x_3)\\ &+ E(x_1, x_2 + h, x_3) + E(x_1, x_2 - h, x_3)\\ &+ E(x_1, x_2, x_3 + h) + E(x_1, x_2, x_3 - h) - 6E(x)) \approx \mu\epsilon\frac{\partial^2 E}{\partial t^2}(t, x).\end{aligned}$$

Für einen Gitterpunkt $x \in \Omega_h$ treten auf der linken Seite lediglich Punkte auf, die entweder ebenfalls Gitterpunkte sind oder auf dem Rand des Gebiets liegen. Wenn wir annehmen, dass Randwerte vorgegeben sind, brauchen wir uns um letztere nicht weiter zu kümmern, da es sich nicht um zu berechnende Unbekannte handelt. Also können wir E durch eine Gitterfunktion E_h ersetzen, die die gewöhnliche Differentialgleichung

$$\begin{aligned}\frac{\partial^2 E_h}{\partial t^2}(t, x) = \frac{1}{\mu\epsilon h^2}(&E_h(t, x_1 + h, x_2, x_3) + E_h(t, x_1 - h, x_2, x_3)\\ &+ E_h(t, x_1, x_2 + h, x_3) + E_h(t, x_1, x_2 - h, x_3)\\ &+ E_h(t, x_1, x_2, x_3 + h) + E_h(t, x_1, x_2, x_3 - h) - 6E_h(t, x))\\ &\text{für alle } t \in [a, b],\ x \in \Omega_h\end{aligned}$$

löst. Dieser Ansatz, also das Ersetzen von Differentialoperatoren durch Differenzenquotienten, trägt den Namen *Finite-Differenzen-Verfahren*.

Bisher traten bei unseren Gleichungen nur erste Ableitungen nach der Zeit auf, während in diesem Fall die zweite Ableitung zu approximieren ist. Diese Aufgabe können wir leicht lösen, indem wir die erste Ableitung als Hilfsvariable

$$V_h(t, x) := \frac{\partial E_h}{\partial t}(t, x) \qquad \text{für alle } t \in [a, b],\ x \in \Omega_h$$

einführen und so zu dem gekoppelten System

$$\begin{aligned}\frac{\partial V_h}{\partial t}(t, x) = \frac{1}{\mu\epsilon h^2}(&E_h(t, x_1 + h, x_2, x_3) + E_h(t, x_1 - h, x_2, x_3)\\ &+ E_h(t, x_1, x_2 + h, x_3) + E_h(t, x_1, x_2 - h, x_3)\\ &+ E_h(t, x_1, x_2, x_3 + h) + E_h(t, x_1, x_2, x_3 - h) - 6E_h(t, x))\end{aligned}$$

$$\frac{\partial E_h}{\partial t}(t,x) = V_h(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega_h$$

gelangen, das wir wie bisher mit einem Zeitschrittverfahren behandeln können, beispielsweise mit dem uns inzwischen wohlbekannten Leapfrog-Algorithmus.

Erfreulicherweise sind die drei Komponenten der Gitterfunktionen E_h nicht aneinander gekoppelt, so dass wir sie vollständig unabhängig voneinander berechnen können.

4.4 Kopplung des elektrischen und magnetischen Felds

Mit Hilfe der Wellengleichung (4.10) können wir das elektrische Feld direkt berechnen, weil wir das magnetische Feld mit Hilfe des Faraday'schen Gesetzes durch das elektrische Feld ersetzen konnten.

In der Praxis ist man häufig daran interessiert, beide Felder gleichzeitig zu berechnen. Das ist vor allem dann wichtig, wenn die Permeabilität μ nicht auf dem gesamten Gebiet konstant ist und sich im Ampére'sche Gesetz (4.6) nicht mehr ohne weiteres aus der Rotation herausziehen lässt, denn dann ist die Vereinfachung (4.9) nicht mehr möglich.

Wir können allerdings immer noch ein Finite-Differenzen-Verfahren entwickeln, indem wir direkt mit dem Ampére'schen Gesetz und dem Faraday'schen Gesetz arbeiten.

Wir beginnen mit dem Faraday'schen Gesetz

$$-\nabla \times E(t,x) = \frac{\partial B}{\partial t}(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega.$$

Da wir auf der Suche nach einem Finite-Differenzen-Verfahren sind, empfiehlt es sich, die in dem Rotationsoperator auftretenden Ableitungen durch Differenzenquotienten zu ersetzen. Aufgrund seiner höheren Genauigkeit ist dabei der zentrale Differenzenquotient (vgl. (2.10c)) besonders attraktiv, mit dem wir

$$\begin{aligned}
\frac{E_3(t,x_1,x_2+h/2,x_3) - E_3(t,x_1,x_2-h/2,x_3)}{h} &= \partial_2 E_3(t,x) + \frac{h^2}{6}\partial_2^3 E_3(t,\eta_1),\\
\frac{E_2(t,x_1,x_2,x_3+h/2) - E_2(t,x_1,x_2,x_3-h/2)}{h} &= \partial_3 E_2(t,x) + \frac{h^2}{6}\partial_3^3 E_2(t,\eta_2),\\
\frac{E_1(t,x_1,x_2,x_3+h/2) - E_1(t,x_1,x_2,x_3-h/2)}{h} &= \partial_3 E_1(t,x) + \frac{h^2}{6}\partial_3^3 E_1(t,\eta_3),\\
\frac{E_3(t,x_1+h/2,x_2,x_3) - E_3(t,x_1-h/2,x_2,x_3)}{h} &= \partial_1 E_3(t,x) + \frac{h^2}{6}\partial_1^3 E_3(t,\eta_4),\\
\frac{E_2(t,x_1+h/2,x_2,x_3) - E_2(t,x_1-h/2,x_2,x_2)}{h} &= \partial_1 E_2(t,x) + \frac{h^2}{6}\partial_1^3 E_2(t,\eta_5),\\
\frac{E_1(t,x_1,x_2+h/2,x_3) - E_1(t,x_1,x_2-h/2,x_3)}{h} &= \partial_2 E_1(t,x) + \frac{h^2}{6}\partial_2^3 E_1(t,\eta_6)
\end{aligned}$$

mit geeigneten $\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6 \in [x_1 - h/2, x_1 + h/2] \times [x_2 - h/2, x_2 + h/2] \times [x_3 - h/2, x_3 + h/2]$ erhalten.

Also können wir alle Komponenten der Rotation $\nabla \times E(t,x)$ approximieren und erhalten

$$\begin{aligned}\frac{\partial B_1}{\partial t}(t,x) \approx -\frac{1}{h}(&E_3(t,x_1,x_2+h/2,x_3) - E_3(t,x_1,x_2-h/2,x_3)\\ &- E_2(t,x_1,x_2,x_3+h/2) + E_2(t,x_1,x_2,x_3-h/2)), \qquad (4.12a)\\ \frac{\partial B_2}{\partial t}(t,x) \approx -\frac{1}{h}(&E_1(t,x_1,x_2,x_3+h/2) - E_1(t,x_1,x_2,x_3-h/2)\\ &- E_3(t,x_1+h/2,x_2,x_3) + E_3(t,x_1-h/2,x_2,x_3)), \qquad (4.12b)\\ \frac{\partial B_3}{\partial t}(t,x) \approx -\frac{1}{h}(&E_2(t,x_1+h/2,x_2,x_3) - E_2(t,x_1-h/2,x_2,x_3)\\ &- E_1(t,x_1,x_2+h/2,x_3) + E_1(t,x_1,x_2-h/2,x_3)). \qquad (4.12c)\end{aligned}$$

Aus dem Ampére'schen Gesetz würden wir eine vergleichbare Näherung für die Zeitableitung des elektrischen Felds erhalten, also insgesamt ein gekoppeltes System für beide Felder.

Da wir den zentralen Differenzenquotienten verwenden benötigen wir allerdings für die Aktualisierung des magnetischen Felds in den bisher verwendeten Gitterpunkten Werte des elektrischen Felds, die gerade *zwischen* diesen Gitterpunkten liegen. Entsprechendes gilt auch für die Aktualisierung des elektrischen Felds.

An dieser Stelle setzt das *Verfahren von Yee* an, das auf der Idee beruht, die einzelnen Gitter gerade so um $h/2$ gegeneinander zu verschieben, dass alle nötigen Werte zur Verfügung stehen. Wir bezeichnen mit

$$\begin{aligned}\bar{\omega} &:= \{ih \ : \ i \in \{0,\dots,n+1\}\},\\ \omega &:= \{ih \ : \ i \in \{1,\dots,n\}\},\\ \widehat{\omega} &:= \{(i+1/2)h \ : \ i \in \{0,\dots,n\}\}\end{aligned}$$

die für ein eindimensionales Gitter und für ein eindimensionales verschobenes Gitter benötigten Punkte. Wir verwenden

- die Werte von E_1 in den Punkten $\widehat{\omega} \times \bar{\omega} \times \bar{\omega}$,
- die Werte von E_2 in den Punkten $\bar{\omega} \times \widehat{\omega} \times \bar{\omega}$,
- die Werte von E_3 in den Punkten $\bar{\omega} \times \bar{\omega} \times \widehat{\omega}$,
- die Werte von B_1 in den Punkten $\bar{\omega} \times \widehat{\omega} \times \widehat{\omega}$,
- die Werte von B_2 in den Punkten $\widehat{\omega} \times \bar{\omega} \times \widehat{\omega}$,
- die Werte von B_3 in den Punkten $\widehat{\omega} \times \widehat{\omega} \times \bar{\omega}$.

Ein Blick auf (4.12) zeigt, dass für die Auswertung der Ableitung von B_1 in einem Gitterpunkt $x \in \omega \times \widehat{\omega} \times \widehat{\omega}$ gerade die Werte von E_2 in $\omega \times \widehat{\omega} \times \bar{\omega}$ und die Werte von E_3 in $\omega \times \bar{\omega} \times \widehat{\omega}$ benötigt werden, und genau diese Werte liegen uns vor. Entsprechendes erhalten wir für B_2 und B_3, so dass wir (4.12) einsetzen können.

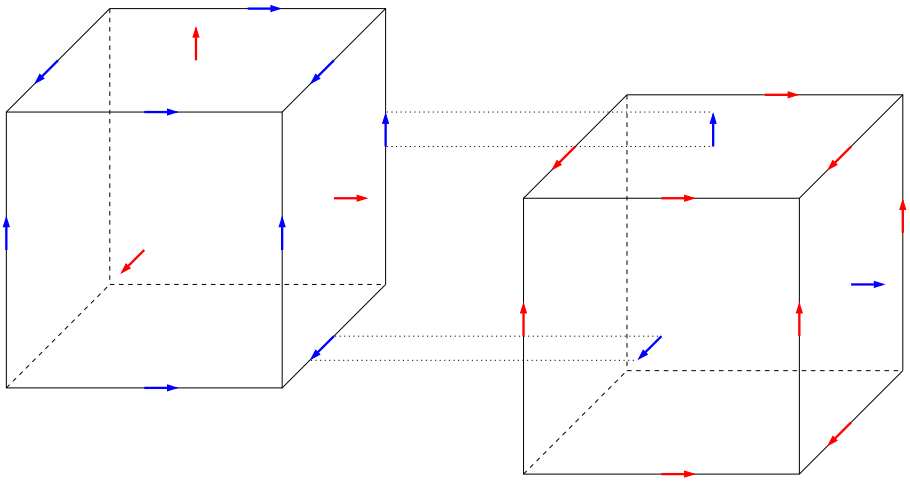

Abbildung 4.2: Komponenten des elektrischen (blau) und magnetischen (rot) Felds auf gegeneinander verschobenen Gittern im Yee-Verfahren

Wir müssen allerdings auch das Ampére'sche Gesetz untersuchen, das wir für die Berechnung des elektrischen Felds benötigen. Mit dem zentralen Differenzenquotienten erhalten wir

$$\begin{aligned}\frac{\partial E_1}{\partial t}(t,x) &\approx \frac{1}{\mu\epsilon h}(B_3(t,x_1,x_2+h/2,x_3) - B_3(t,x_1,x_2-h/2,x_3)\\ &\quad - B_2(t,x_1,x_2,x_3+h/2) + B_2(t,x_1,x_2,x_3-h/2)), \qquad (4.13a)\\ \frac{\partial E_2}{\partial t}(t,x) &\approx \frac{1}{\mu\epsilon h}(B_1(t,x_1,x_2,x_3+h/2) - B_1(t,x_1,x_2,x_3-h/2)\\ &\quad - B_3(t,x_1+h/2,x_2,x_3) + B_3(t,x_1-h/2,x_2,x_3)), \qquad (4.13b)\\ \frac{\partial E_3}{\partial t}(t,x) &\approx \frac{1}{\mu\epsilon h}(B_2(t,x_1+h/2,x_2,x_3) - B_2(t,x_1-h/2,x_2,x_3)\\ &\quad - B_1(t,x_1,x_2+h/2,x_3) + B_1(t,x_1,x_2-h/2,x_3)). \qquad (4.13c)\end{aligned}$$

Auch hier stellen wir fest, dass für die Auswertung der Ableitung von E_1 in einem Gitterpunkt $x \in \widehat{\omega} \times \omega \times \omega$ gerade die Werte von B_2 in $\widehat{\omega} \times \omega \times \widehat{\omega}$ und die Werte von B_3 in $\widehat{\omega} \times \widehat{\omega} \times \omega$ gebraucht werden, und diese Werte stehen uns zur Verfügung.

Bemerkung 4.9 (FDTD-Verfahren) *Die Yee-Diskretisierung ist ein Beispiel für ein FDTD-Verfahren (engl.* finite difference time domain*), also ein Verfahren, bei dem die Ortsableitungen mittels eines Finite-Differenzen-Verfahrens approximiert werden und die Lösung in Abhängigkeit von der Zeit bestimmt wird.*

Eine Alternative sind FDFD-Diskretisierungen (engl. finite difference frequency domain*), bei denen man die Zeitvariable mit einem Ansatz der Form*

$$E(t,x) = E_0(x)\cos(\omega t), \qquad B(t,x) = B_0(x)\sin(\omega t)$$

von den Ortsvariablen trennt und letztendlich eliminieren kann. Stattdessen ist dann gegebenenfalls der Frequenzparameter ω zu bestimmen.

4.5 Elektrostatik

In bestimmten Anwendungen ist man daran interessiert, Felder zu untersuchen, die konstant (oder zumindest „fast konstant“) sind. Als Beispiel behandeln wir hier den Fall der *elektrostatischen Felder*.

Wir gehen davon aus, dass das magnetische Feld B in der Zeit konstant ist, dass also

$$\frac{\partial B}{\partial t}(t,x) = 0 \qquad \text{für alle } t \in [a,b],\ x \in \Omega$$

gilt. Mit dem Faraday'schen Gesetz (4.5) folgt daraus

$$\nabla \times E(t,x) = 0 \qquad \text{für alle } t \in [a,b],\ x \in \Omega, \tag{4.14}$$

das elektrische Feld weist also keine Wirbel auf. Mit Hilfe eines Resultats der Analysis von Vektorfeldern können wir aus dieser Gleichung eine spezielle Darstellung des Felds E gewinnen.

Definition 4.10 (Gradient) *Sei $\varphi : \Omega \to \mathbb{R}$ eine stetig differenzierbare Funktion. Die Abbildung*

$$\nabla\varphi : \Omega \to \mathbb{R}^3, \qquad x \mapsto \nabla\varphi(x) := \begin{pmatrix} \partial_1\varphi(x) \\ \partial_2\varphi(x) \\ \partial_3\varphi(x) \end{pmatrix}$$

heißt der Gradient *der Funktion φ.*

Definition 4.11 (Einfach zusammenhängendes Gebiet) *Eine offene zusammenhängende Teilmenge $\Omega \subseteq \mathbb{R}^3$ nennen wir ein* Gebiet.

Eine stetige Abbildung $\gamma : [0,1] \to \Omega$ nennen wir einen Weg *in Ω und einen* geschlossenen Weg*, falls $\gamma(0) = \gamma(1)$ gilt.*

Ein geschlossener Weg γ heißt nullhomotop*, falls ein Punkt $x_0 \in \Omega$ und eine stetige Abbildung $h : [0,1] \times [0,1] \to \Omega$ mit*

$$h(0,t) = \gamma(t), \qquad h(1,t) = x_0 \qquad \textit{für alle } t \in [0,1]$$

existieren, falls sich also der Weg „stetig zu einem Punkt zusammenziehen lässt“.

Ein Gebiet Ω heißt einfach zusammenhängend*, falls jeder geschlossene Weg nullhomotop ist.*

Anschaulich sind Gebiete „ohne Löcher“ einfach zusammenhängend.

Erinnerung 4.12 (Potentialdarstellung) *Sei $\Omega \subseteq \mathbb{R}^3$ ein einfach zusammenhängendes Gebiet und $u : \Omega \to \mathbb{R}^3$ eine stetig differenzierbare Abbildung mit*

$$\nabla \times u(x) = 0 \qquad \text{für alle } x \in \Omega.$$

Dann existiert eine stetig differenzierbare Funktion $\varphi : \Omega \to \mathbb{R}$ mit

$$u(x) = \nabla\varphi(x) \qquad \text{für alle } x \in \Omega.$$

Diese Funktion φ wird gelegentlich als das Potential *des Vektorfelds u bezeichnet.*

Wir setzen im Folgenden voraus, dass Ω einfach zusammenhängend ist. Dann folgt aus (4.14) mit Erinnerung 4.12, dass sich das elektrische Feld durch ein Potential $\varphi : [a,b] \times \Omega \to \mathbb{R}$ in der Form

$$E(t,x) = \nabla\varphi(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega$$

darstellen lässt. Indem wir diese Gleichung in das Gauß'sche Gesetz (4.7) einsetzen erhalten wir

$$\nabla \cdot (\epsilon\nabla\varphi)(t,x) = \varrho(t,x) \qquad \text{für alle } t \in [a,b],\ x \in \Omega.$$

Wenn wir diese partielle Differentialgleichung lösen, erhalten wir eine Beschreibung des elektrischen Felds durch das Potential φ.

Die linke Seite der Gleichung können wir wieder etwas umformulieren, um sie auf eine uns bereits bekannte Form zu bringen:

$$\begin{aligned}\nabla \cdot (\epsilon\nabla\varphi)(t,x) &= \partial_1(\epsilon(\nabla\varphi)_1)(t,x) + \partial_2(\epsilon(\nabla\varphi)_2)(t,x) + \partial_3(\epsilon(\nabla\varphi)_3)(t,x)\\ &= \epsilon\left(\partial_1^2\varphi(t,x) + \partial_2^2\varphi(t,x) + \partial_3^2\varphi(t,x)\right).\end{aligned}$$

Eine Näherung der zweiten Ableitungen stellt uns Lemma 4.8 zur Verfügung, so dass es sich anbietet, wieder ein Finite-Differenzen-Verfahren einzusetzen.

Der Einfachheit halber beschränken wir wieder uns auf den Einheitswürfel $\Omega = (0,1)^3$ und gehen davon aus, dass die Ladungsdichte ϱ in der Zeit konstant ist. Dann können wir auch φ unabhängig von der Zeit wählen und müssen nur noch

$$-\epsilon\left(\partial_1^2\varphi(x) + \partial_2^2\varphi(x) + \partial_3^2\varphi(x)\right) = -\varrho(x) \qquad \text{für alle } x \in \Omega$$

lösen. Das Minuszeichen wurde auf beiden Seiten der Gleichung eingefügt, um bestimmte für die Lösungsalgorithmen wichtige Eigenschaften sicherzustellen.

Damit die Gleichung eindeutig lösbar wird, müssen wir Randwerte vorgeben. Der Einfachheit halber nehmen wir an, dass der Rand des Gebiets *supraleitend* ist, dass also das elektrische Feld keine Kraft in tangentialer Richtung ausübt, weil jegliche solche Kraft sofort dazu führen würde, dass sich Elektronen so verschieben, dass sie sie aufheben. Diese Randbedingung können wir einfach durch die Formel

$$\varphi(x) = 0 \qquad \text{für alle } x \in \partial\Omega$$

beschreiben, wobei

$$\partial\Omega := \{0,1\} \times [0,1] \times [0,1] \cup [0,1] \times \{0,1\} \times [0,1] \cup [0,1] \times [0,1] \times \{0,1\}$$

den Rand des Gebiets bezeichnet.

Insgesamt erhalten wir die *Potentialgleichung*

$$\begin{aligned} -\epsilon \left(\partial_1^2\varphi(x) + \partial_2^2\varphi(x) + \partial_3^2\varphi(x)\right) &= -\varrho(x) && \text{für alle } x \in \Omega, && (4.15a)\\ \varphi(x) &= 0 && \text{für alle } x \in \partial\Omega, && (4.15b) \end{aligned}$$

die wir nun mit Hilfe des Lemmas 4.8 diskretisieren werden.

Dazu wählen wir wieder ein $n \in \mathbb{N}$ und konstruieren durch

$$\begin{aligned} h &:= \frac{1}{n+1},\\ \bar{\Omega}_h &:= \{ih \ : \ i \in \{0,\dots,n+1\}\}^3,\\ \Omega_h &:= \{ih \ : \ i \in \{1,\dots,n\}\}^3,\\ \partial\Omega_h &:= \bar{\Omega}_h \setminus \Omega_h \end{aligned}$$

ein Gitter $\bar{\Omega}_h$ mit *inneren Punkten* Ω_h und *Randpunkten* $\partial\Omega_h$. Indem wir die zweiten Ableitungen in (4.15) durch Differenzenquotienten ersetzen erhalten wir

$$\begin{aligned} \frac{\epsilon}{h^2}(6\varphi(x) &- \varphi(x_1-h,x_2,x_3) - \varphi(x_1+h,x_2,x_3)\\ &- \varphi(x_1,x_2-h,x_3) - \varphi(x_1,x_2+h,x_3)\\ &- \varphi(x_1,x_2,x_3-h) - \varphi(x_1,x_2,x_3+h)) \approx \varrho(x) \qquad \text{für alle } x \in \Omega_h. \end{aligned}$$

Wie zuvor gehen wir über zu einer Gitterfunktion $\varphi_h : \bar{\Omega} \to \mathbb{R}$ mit

$$\begin{aligned} \frac{\epsilon}{h^2}(6\varphi_h(x) &- \varphi_h(x_1-h,x_2,x_3) - \varphi_h(x_1+h,x_2,x_3)\\ &- \varphi_h(x_1,x_2-h,x_3) - \varphi_h(x_1,x_2+h,x_3)\\ &- \varphi_h(x_1,x_2,x_3-h) - \varphi_h(x_1,x_2,x_3+h)) = \varrho(x) \quad \text{für alle } x \in \Omega_h \quad (4.16) \end{aligned}$$

und ergänzen die Randbedingung

$$\varphi_h(x) = 0 \qquad \text{für alle } x \in \partial\Omega_h. \qquad (4.17)$$

Aus einer etwas abstrakteren Perspektive betrachtet handelt es sich bei (4.16) um ein lineares Gleichungssystem mit den Unbekannten $(\varphi_h(x))_{x\in\Omega_h}$. Zur Abkürzung bezeichnen wir die Indexmenge mit $\mathcal{I} := \Omega_h$ und fassen die Unbekannten und die rechte Seite in Vektoren $\mathbf{u}, \mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ zusammen, die durch

$$u_x := \varphi_h(x), \qquad b_x := \varrho(x) \qquad \text{für alle } x \in \Omega_h$$

definiert sind. Dank der besonders einfachen Randbedingung (4.17) dürfen wir die Randwerte einfach wegfallen lassen und können so die Gleichung (4.16) mit Hilfe der durch

$$l_{xy} = \begin{cases} 6\epsilon/h^2 & \text{falls } x = y, \\ -\epsilon/h^2 & \text{falls } |x_1 - y_1| = h,\ x_2 = y_2,\ x_3 = y_3, \\ -\epsilon/h^2 & \text{falls } x_1 = y_1,\ |x_2 - y_2| = h,\ x_3 = y_3, \\ -\epsilon/h^2 & \text{falls } x_1 = y_1,\ x_2 = y_2,\ |x_3 - y_3| = h, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } x, y \in \Omega_h \tag{4.18}$$

gegebenen Matrix $\mathbf{L} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ in die kompakte Form

$$\mathbf{Lu} = \mathbf{b}$$

bringen. Dieses lineare Gleichungssystem müssen wir lösen, um das Potential und damit auch das elektrische Feld zu bestimmen.

4.6 Gradientenverfahren für lineare Gleichungssysteme

Aus der Schule kennt man beispielsweise das Gauß'sche Eliminationsverfahren für das Lösen eines linearen Gleichungssystems, bei dem man eine Unbekannte nach der anderen eliminiert, bis nur noch eine Gleichung bleibt, die man lösen kann. Anschließend lassen sich die eliminierten Unbekannten rekonstruieren. In modernen Implementierungen wird dieses Verfahren als Zerlegung der Systemmatrix in das Produkt zweier Dreiecksmatrizen umgesetzt, die aktuelle Computer sehr effizient durchführen können.

Allerdings erfordert diese Berechnung auch bei modernen Computern einen relativ hohen Rechenaufwand, insbesondere geht bei den meisten Verfahren die Anzahl der Unbekannten mindestens quadratisch in die Anzahl der Rechenoperationen ein. Da nach Lemma 4.8 die Genauigkeit unserer Approximation von $h^2 = 1/(n+1)^2$ abhängt, werden wir n relativ hoch wählen müssen, um eine vertretbare Qualität der Näherungslösung zu erreichen. Damit wird auch die Anzahl $\#\mathcal{I} = n^3$ der Unbekannten und Gleichungen sehr hoch sein, so dass die Gauß-Elimation nicht mehr praktikabel ist.

Glücklicherweise gibt es eine Reihe alternativer Lösungsverfahren, die wesentlich effizienter arbeiten, und eines dieser Verfahren soll an dieser Stelle eingeführt werden.

Es beruht auf zwei wesentlichen Eigenschaften der Matrix $\mathbf{L}$ aus (4.18): Sie ist *symmetrisch* und *positiv definit*.

Definition 4.13 (Transponierte Matrix) *Sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ eine Matrix. Die durch*

$$b_{ij} := a_{ji} \qquad \text{für alle } i \in \mathcal{J},\ j \in \mathcal{I}$$

definierte Matrix $\mathbf{B} \in \mathbb{R}^{\mathcal{J}\times\mathcal{I}}$ nennen wir die Transponierte *(oder* Adjungierte*) der Matrix $\mathbf{A}$ und schreiben sie als $\mathbf{A}^* := \mathbf{B}$.*

Transponierte Matrizen sind für uns von Bedeutung, weil sie in enger Beziehung zu dem euklidischen Skalarprodukt (vgl. Erinnerung 3.3) stehen:

Lemma 4.14 (Transponierte und Skalarprodukt) *Sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$. Dann gilt*

$$\langle \mathbf{x}, \mathbf{A}\mathbf{y}\rangle_2 = \langle \mathbf{A}^*\mathbf{x}, \mathbf{y}\rangle_2 \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}},\ \mathbf{y} \in \mathbb{R}^{\mathcal{J}}. \tag{4.19}$$

Beweis. Seien $\mathbf{x} \in \mathbb{R}^{\mathcal{I}}$ und $\mathbf{y} \in \mathbb{R}^{\mathcal{J}}$ gegeben. Analog zu Definition 4.13 setzen wir $\mathbf{B} := \mathbf{A}^*$. Dann gilt

$$\begin{aligned}\langle \mathbf{x}, \mathbf{A}\mathbf{y}\rangle_2 &= \sum_{i\in\mathcal{I}} x_i(\mathbf{A}\mathbf{y})_i = \sum_{i\in\mathcal{I}} x_i \sum_{j\in\mathcal{I}} a_{ij}y_j = \sum_{j\in\mathcal{I}} y_j \sum_{i\in\mathcal{I}} a_{ij}x_i \\ &= \sum_{j\in\mathcal{I}} y_j \sum_{i\in\mathcal{I}} b_{ji}x_i = \sum_{j\in\mathcal{I}} y_j(\mathbf{B}\mathbf{x})_j = \langle \mathbf{B}\mathbf{x}, \mathbf{y}\rangle_2 = \langle \mathbf{A}^*\mathbf{x}, \mathbf{y}\rangle_2.\end{aligned}$$

■

Von besonderem Interesse für uns sind in diesem Abschnitt Matrizen, die mit ihren Transponierten übereinstimmen.

Definition 4.15 (Symmetrische Matrix) *Sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine Matrix. Wir nennen sie* symmetrisch *(oder* selbstadjungiert*), falls $\mathbf{A} = \mathbf{A}^*$ gilt.*

Für eine symmetrische Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ erhalten wir mit der Symmetrie des euklidischen Skalarprodukts

$$\langle \mathbf{x}, \mathbf{A}\mathbf{y}\rangle_2 = \langle \mathbf{y}, \mathbf{A}\mathbf{x}\rangle_2 \qquad \text{für alle } \mathbf{x}, \mathbf{y} \in \mathbb{R}^{\mathcal{I}}. \tag{4.20}$$

Da sich in (4.18) die Rollen von x und y vertauschen lassen, ohne dass sich die Einträge ändern, dürfen wir festhalten, dass die Matrix $\mathbf{L}$ symmetrisch ist.

Definition 4.16 (Positiv definit) *Sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine Matrix. Falls*

$$\langle \mathbf{x}, \mathbf{A}\mathbf{x}\rangle_2 > 0 \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$$

gilt, nennen wir $\mathbf{A}$ positiv definit.

Die Matrix $\mathbf{L}$ aus (4.18) ist auch positiv definit:

Lemma 4.17 (Potentialmatrix) *Die durch (4.18) definierte Matrix $\mathbf{L} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ ist positiv definit.*

Beweis. Wir bezeichnen mit

$$\begin{aligned}N_i := \{j \in \bar{\Omega}_h\ :\ & |i_1 - j_1| = h,\ i_2 = j_2,\ i_3 = j_3 \text{ oder} \\ & i_1 = j_1,\ |i_2 - j_2| = h,\ i_3 = j_3 \text{ oder} \\ & i_1 = j_1,\ i_2 = j_2,\ |i_3 - j_3| = h\} \qquad \text{für alle } i \in \bar{\Omega}_h\end{aligned}$$

die Menge der Nachbarpunkte der Gitterpunkte. Diese Mengen erfüllen die Symmetriebedingung $j \in N_i \iff i \in N_j$ für alle $i, j \in \bar{\Omega}_h$.

Sei $\mathbf{x} \in \mathbb{R}^{\mathcal{I}}$. Entsprechend unserer Randbedingung (4.17) setzen wir den Vektor mit Nullrandbedingungen zu $\widehat{\mathbf{x}} \in \mathbb{R}^{\bar{\Omega}_h}$ mit $\widehat{\mathbf{x}}|_{\Omega_h} = \mathbf{x}$ und $\widehat{\mathbf{x}}|_{\partial\Omega_h} = \mathbf{0}$ fort.

Wir erhalten

$$\begin{aligned}
\langle \mathbf{x}, \mathbf{L}\mathbf{x}\rangle_2 &= \sum_{i\in\mathcal{I}} x_i \sum_{j\in\mathcal{I}} \ell_{ij} x_j = \frac{1}{2}\sum_{i\in\mathcal{I}} x_i \sum_{j\in\mathcal{I}} \ell_{ij} x_j + \frac{1}{2}\sum_{j\in\mathcal{I}} x_j \sum_{i\in\mathcal{I}} \ell_{ji} x_i \\
&= \frac{\epsilon}{2h^2}\sum_{i\in\mathcal{I}} x_i \left(6x_i - \sum_{j\in N_i} \hat{x}_j\right) + \frac{\epsilon}{2h^2}\sum_{j\in\mathcal{I}} x_j \left(6x_j - \sum_{i\in N_j} \hat{x}_i\right) \\
&= \frac{\epsilon}{2h^2}\sum_{i\in\mathcal{I}} x_i \sum_{j\in N_i} (x_i - \hat{x}_j) + \frac{\epsilon}{2h^2}\sum_{j\in\mathcal{I}} x_j \sum_{i\in N_j} (x_j - \hat{x}_i) \\
&= \frac{\epsilon}{h^2}\sum_{i\in\bar{\Omega}_h}\sum_{j\in N_i} \hat{x}_i(\hat{x}_i - \hat{x}_j) + \frac{\epsilon}{h^2}\sum_{j\in\bar{\Omega}_h}\sum_{i\in N_j} \hat{x}_j(\hat{x}_j - \hat{x}_i) \\
&= \frac{\epsilon}{h^2}\sum_{i\in\bar{\Omega}_h}\sum_{j\in N_i} \hat{x}_i(\hat{x}_i - \hat{x}_j) - \frac{\epsilon}{h^2}\sum_{j\in\bar{\Omega}_h}\sum_{i\in N_j} \hat{x}_j(\hat{x}_i - \hat{x}_j) \\
&= \frac{\epsilon}{h^2}\sum_{i\in\bar{\Omega}_h}\sum_{j\in N_i} \hat{x}_i(\hat{x}_i - \hat{x}_j) - \frac{\epsilon}{h^2}\sum_{i\in\bar{\Omega}_h}\sum_{j\in N_i} \hat{x}_j(\hat{x}_i - \hat{x}_j) \\
&= \frac{\epsilon}{2h^2}\sum_{i\in\bar{\Omega}_h}\sum_{j\in N_i} (\hat{x}_i - \hat{x}_j)^2 \geq 0.
\end{aligned}$$

Falls $\langle \mathbf{x}, \mathbf{L}\mathbf{x}\rangle_2 = 0$ gelten sollte, müssen benachbarte Punkte des Vektors $\widehat{\mathbf{x}}$ identisch sein, also müssen auch alle Komponenten des Vektors $\mathbf{x}$ identisch sein. Unsere Randbedingung führt in diesem Fall schon zu $\widehat{\mathbf{x}} = \mathbf{0}$, also ist $\mathbf{L}$ positiv definit. ∎

Die Idee des neuen Lösungsverfahrens beruht darauf, die Funktion

$$f : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}, \qquad \mathbf{x} \mapsto \frac{1}{2}\langle \mathbf{x}, \mathbf{L}\mathbf{x}\rangle_2 - \langle \mathbf{x}, \mathbf{b}\rangle_2$$

zu untersuchen, denn sie besitzt die folgende wichtige Eigenschaft:

Lemma 4.18 (Minimierungsaufgabe) *Sei $\mathbf{L} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine selbstadjungierte und positiv definite Matrix.*

Für alle $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ gilt

$$f(\mathbf{x}) \leq f(\mathbf{x} + \theta\mathbf{y}) \qquad \text{für alle } \theta \in \mathbb{R} \tag{4.21}$$

genau dann, wenn

$$\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2 = 0 \tag{4.22}$$

gilt. Daraus folgt, dass f sein globales Minimum für den Vektor $\mathbf{x} \in \mathbb{R}^{\mathcal{I}}$ annimmt, der $\mathbf{L}\mathbf{x} = \mathbf{b}$ erfüllt, also unser Gleichungssystem löst.

Beweis. Seien $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ gegeben. Wir verwenden zunächst (4.20), um die Gleichung

$$f(\mathbf{x} + \theta\mathbf{y}) = \frac{1}{2}\langle \mathbf{x} + \theta\mathbf{y}, \mathbf{L}(\mathbf{x} + \theta\mathbf{y})\rangle_2 - \langle \mathbf{x} + \theta\mathbf{y}, \mathbf{b}\rangle_2$$

$$= \frac{1}{2}\langle \mathbf{x}, \mathbf{L}\mathbf{x}\rangle_2 + \frac{1}{2}\langle \mathbf{x}, \mathbf{L}\theta\mathbf{y}\rangle_2 + \frac{1}{2}\langle \theta\mathbf{y}, \mathbf{L}\mathbf{x}\rangle_2 + \frac{1}{2}\langle \theta\mathbf{y}, \mathbf{L}\theta\mathbf{y}\rangle_2 - \langle \mathbf{x}, \mathbf{b}\rangle_2 - \langle \theta\mathbf{y}, \mathbf{b}\rangle_2$$
$$= f(\mathbf{x}) + \frac{\theta}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{x}\rangle_2 + \frac{\theta}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{x}\rangle_2 + \frac{\theta^2}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2 - \theta\langle \mathbf{y}, \mathbf{b}\rangle_2$$
$$= f(\mathbf{x}) + \theta\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2 + \frac{\theta^2}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2 \tag{4.23}$$

nachzuweisen. Gelte nun (4.21). Wir wollen (4.22) nachweisen. Falls $\mathbf{y} = \mathbf{0}$ gilt ist die Gleichung trivial erfüllt. Ansonsten suchen wir ein $\theta \in \mathbb{R}$, für das (4.23) minimal wird. Das Minimum ist die Nullstelle der Ableitung nach θ, also gerade

$$\theta = -\frac{\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2}{\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2}. \tag{4.24}$$

Da $\mathbf{L}$ positiv definit und $\mathbf{y}$ nicht der Nullvektor ist, ist der Nenner dieses Terms echt positiv, also θ wohldefiniert. Durch Einsetzen in (4.23) erhalten wir

$$f(\mathbf{x} + \theta\mathbf{y}) = f(\mathbf{x}) - \frac{\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2^2}{2\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2}.$$

Nach (4.21) muss aber

$$f(\mathbf{x}) \leq f(\mathbf{x} + \theta\mathbf{y}) = f(\mathbf{x}) - \frac{\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2^2}{2\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2}$$

gelten, also folgt

$$0 \leq -\frac{\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2^2}{2\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2} \leq 0,$$

da $\mathbf{L}$ positiv definit ist. Daraus ergibt sich unmittelbar (4.22).

Sei nun (4.22) vorausgesetzt. Aus (4.23) folgt

$$f(\mathbf{x} + \theta\mathbf{y}) = f(\mathbf{x}) + \theta\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2 + \frac{\theta^2}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2 = f(\mathbf{x}) + \frac{\theta^2}{2}\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2 \geq f(\mathbf{x}),$$

also gerade (4.21), da $\mathbf{L}$ positiv definit ist.

Falls $\mathbf{x}$ das globale Minimum der Funktion f ist, muss (4.21) für *alle* $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ gelten, also insbesondere auch für $\mathbf{y} := \mathbf{L}\mathbf{x} - \mathbf{b}$, so dass wir mit (4.22) die Gleichung

$$0 = \langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2 = \langle \mathbf{L}\mathbf{x} - \mathbf{b}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2 = \|\mathbf{L}\mathbf{x} - \mathbf{b}\|_2^2$$

erhalten. Sie ist äquivalent zu $\mathbf{L}\mathbf{x} = \mathbf{b}$. ∎

Wir können also auch das Minimum der Funktion f suchen, statt das Gleichungssystem zu lösen. Auf den ersten Blick haben wir damit nicht viel gewonnen, auf den zweiten stellen wir allerdings fest, dass wir uns dem Minimum schrittweise *annähern* können: Wir wählen eine beliebige Richtung $\mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$ und versuchen, $\theta \in \mathbb{R}$ so zu wählen, dass $f(\mathbf{x} + \theta\mathbf{y})$ so klein wie möglich wird, also gemäß (4.24). Dann ist

$$\mathbf{x}' := \mathbf{x} + \theta\mathbf{y} = \mathbf{x} - \frac{\langle \mathbf{y}, \mathbf{L}\mathbf{x} - \mathbf{b}\rangle_2}{\langle \mathbf{y}, \mathbf{L}\mathbf{y}\rangle_2}\mathbf{y}$$

eine verbesserte Näherungslösung, mit der wir die Prozedur wiederholen können.

Natürlich stellt sich die Frage, wie die Richtung $\mathbf{y}$ zu wählen ist. Wenn θ relativ klein ist, können wir den quadratischen Term in (4.23) vernachlässigen und stellen fest, dass es gut wäre, wenn der lineare Term $\langle \mathbf{Lx} - \mathbf{b}, \mathbf{y} \rangle_2$ möglichst groß wäre.

Mit der Cauchy-Schwarz-Ungleichung (3.4) gelangen wir zu der Erkenntnis, dass

$$\mathbf{y} := \mathbf{Lx} - \mathbf{b}$$

eine optimale Lösung der vereinfachten Aufgabe ist.

Würden wir diesen Vektor einfach einsetzen, wäre die Durchführung eines Schritts unseres Verfahrens relativ aufwendig: Die Berechung von $\mathbf{y}$ würde eine Matrix-Vektor-Multiplikation erfordern, die Berechnung von $\mathbf{Ly}$ im Nenner des optimalen θ eine weitere. Glücklicherweise lässt sich eine der beiden Multiplikationen einsparen, indem wir einen Hilfsvektor verwenden.

Dazu bezeichnen wir mit $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ die Folge der Näherungslösungen und mit

$$\mathbf{y}^{(m)} := \mathbf{Lx}^{(m)} - \mathbf{b}, \qquad \text{für alle } m \in \mathbb{N}_0 \tag{4.25}$$

die Hilfsvektoren. Der Schritt von $\mathbf{x}^{(m)}$ zu $\mathbf{x}^{(m+1)}$ nimmt dann die Form

$$\begin{aligned} \theta^{(m)} &:= -\frac{\langle \mathbf{y}^{(m)}, \mathbf{y}^{(m)} \rangle_2}{\langle \mathbf{y}^{(m)}, \mathbf{Ly}^{(m)} \rangle_2}, \\ \mathbf{x}^{(m+1)} &:= \mathbf{x}^{(m)} + \theta^{(m)} \mathbf{y}^{(m)} \end{aligned}$$

an. Wir können den nächsten Vektor $\mathbf{y}^{(m+1)}$ durch

$$\begin{aligned} \mathbf{y}^{(m+1)} &= \mathbf{Lx}^{(m+1)} - \mathbf{b} = \mathbf{L}(\mathbf{x}^{(m)} + \theta^{(m)} \mathbf{y}^{(m)}) - \mathbf{b} \\ &= \mathbf{Lx}^{(m)} - \mathbf{b} + \theta^{(m)} \mathbf{Ly}^{(m)} = \mathbf{y}^{(m)} + \theta^{(m)} \mathbf{Ly}^{(m)} \end{aligned}$$

ausdrücken und stellen fest, dass wir für einen Schritt des Verfahrens lediglich *eine* Matrix-Vektor-Multiplikation für die Berechnung des Vektors $\mathbf{a}^{(m)} := \mathbf{Ly}^{(m)}$ benötigen.

Definition 4.19 (Gradientenverfahren) *Sei $\mathbf{L} \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ selbstadjungiert und positiv definit, und seien $\mathbf{x}^{(0)} \in \mathbb{R}^{\mathcal{I}}$ sowie $\mathbf{y}^{(0)} := \mathbf{Lx}^{(0)} - \mathbf{b}$.*

Das Gradientenverfahren *definiert induktiv durch*

$$\begin{aligned} \mathbf{a}^{(m)} &:= \mathbf{Ly}^{(m)}, \\ \theta^{(m)} &:= -\langle \mathbf{y}^{(m)}, \mathbf{y}^{(m)} \rangle_2 / \langle \mathbf{y}^{(m)}, \mathbf{a}^{(m)} \rangle_2 \\ \mathbf{x}^{(m+1)} &:= \mathbf{x}^{(m)} + \theta^{(m)} \mathbf{y}^{(m)}, \\ \mathbf{y}^{(m+1)} &:= \mathbf{y}^{(m)} + \theta^{(m)} \mathbf{a}^{(m)} \qquad \textit{für alle } m \in \mathbb{N}_0 \end{aligned}$$

eine Folge von Näherungslösungen $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ des Gleichungssystems $\mathbf{Lx} = \mathbf{b}$.

Wie nicht anders zu erwarten besteht ein Zusammenhang zwischen dem Gradientenverfahren und dem in Definition 4.10 eingeführten Gradienten:

Übungsaufgabe 4.20 (Gradient) *Beweisen Sie, dass*

$$\nabla f(\mathbf{x}) = \mathbf{L}\mathbf{x} - \mathbf{b} \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}}$$

gilt, dass wir unsere Suchrichtungen $\mathbf{y}$ *also gerade als* Gradienten *der Funktion* f *wählen.*

Dieser Eigenschaft verdankt das Gradientenverfahren seinen Namen.

Anschaulich beschreibt der Gradient in einem Punkt $\mathbf{x}$ *gerade die Richtung, in der* f *am steilsten anwächst. In der entgegengesetzten Richtung fällt die Funktion am steilsten ab, und diese Richtung wählt unser Verfahren.*

Im Vergleich zu der Gauß-Elimination bietet das Gradientenverfahren eine Reihe wichtiger Vorteile:

- Für die Durchführung des Verfahrens benötigen wir nur die Möglichkeit, die Systemmatrix mit einem Vektor zu multiplizieren. In unserem Fall ist $\mathbf{L}$ eine *schwachbesetzte Matrix*, es sind also pro Zeile und Spalte jeweils nur wenige Einträge ungleich null, so dass sich die Matrix-Vektor-Multiplikation in $\mathcal{O}(\#\mathcal{I})$ Rechenoperationen durchführen lässt.
- Die Matrix-Vektor-Multiplikation lässt sich gut parallelisieren.
- Da unser Gleichungssystem ohnehin nur eine Näherung der Potentialgleichung darstellt, ist es angemessen, auch nur eine Näherung seiner Lösung zu berechnen.
- Dank (4.25) können wir in jedem Schritt einfach abschätzen, wie genau die aktuelle Näherung ist.

Leider hat das Gradientenverfahren auch einen erheblichen Nachteil: Es benötigt sehr viele Schritte, insbesondere für die Matrix $\mathbf{L}$, die bei der Behandlung der Potentialgleichung auftritt.

Die Ursache dafür liegt darin, dass die Näherungslösung $\mathbf{x}^{(1)}$ nach unserer Konstruktion optimal bezüglich der Richtung $\mathbf{y}^{(0)}$ ist, dass aber schon die nächste Näherung $\mathbf{x}^{(2)}$ diese Eigenschaft wieder verlieren kann.

Diesen sehr unwillkommenen Effekt können wir vermeiden, indem wir die Suchrichtungen $\mathbf{y}^{(m)}$ geschickter wählen: Angenommen, eine Näherung $\mathbf{x}^{(m)}$ ist optimal bezüglich einer Richtung $\mathbf{y}^{(\ell)}$. Nach Lemma 4.18 gilt dann

$$0 = \langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{x}^{(m)} - \mathbf{b} \rangle_2.$$

Unser Verfahren berechnet die nächste Näherungslösung durch

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \theta \mathbf{y}^{(m)}.$$

Wieder nach Lemma 4.18 bleibt sie optimal bezüglich $\mathbf{y}^{(\ell)}$, falls

$$\begin{aligned} 0 = \langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{x}^{(m+1)} - \mathbf{b} \rangle_2 &= \langle \mathbf{y}^{(\ell)}, \mathbf{L}(\mathbf{x}^{(m)} + \theta \mathbf{y}^{(m)}) - \mathbf{b} \rangle_2 \\ &= \langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{x}^{(m)} - \mathbf{b} \rangle_2 + \theta \langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 \end{aligned}$$

gilt. Da $\mathbf{x}^{(m)}$ optimal bezüglich $\mathbf{y}^{(\ell)}$ ist, fällt der erste Term weg, so dass nur

$$0 = \theta \langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2$$

übrig bleibt. Diese Bedingung können wir trivial erfüllen, indem wir $\theta = 0$ setzen, aber dann wäre die neue Näherung gleich der alten.

Wesentlich interessanter ist es, sie zu erfüllen, indem wir sicherstellen, dass die Richtungen $\mathbf{y}^{(m)}$ und $\mathbf{y}^{(\ell)}$ *konjugiert* sind, dass also

$$\langle \mathbf{y}^{(\ell)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 = 0$$

gilt. In diesem Fall können wir θ wie zuvor optimal wählen, ohne die Optimalität bezüglich $\mathbf{y}^{(\ell)}$ zu gefährden.

Mit dieser Beobachtung können wir unseren Algorithmus erheblich verbessern: Der erste Schritt wird wie bisher ausgeführt und ergibt eine Näherung $\mathbf{x}^{(1)}$, die optimal bezüglich der Richtung $\mathbf{y}^{(0)}$ ist. Im zweiten Schritt modifizieren wir die Richtung $\mathbf{y}^{(1)}$ so, dass $\mathbf{y}^{(0)}$ und $\mathbf{y}^{(1)}$ konjugiert sind, bevor wir $\mathbf{x}^{(2)}$ mit dieser verbesserten Richtung berechnen. Entsprechend sorgen wir im m-ten Schritt dafür, dass die Richtung $\mathbf{y}^{(m)}$ und alle $\mathbf{y}^{(\ell)}$ mit $\ell < m$ konjugiert sind, damit die Näherung $\mathbf{x}^{(m+1)}$ optimal bezüglich *aller* bereits verwendeten Richtungen ist.

Der Preis für diese Verbesserung ist allerdings, dass wir nun den aktuellen Gradienten

$$\mathbf{g}^{(m)} := \mathbf{L}\mathbf{x}^{(m)} - \mathbf{b}$$

in einem separaten Vektor mitführen müssen, da wir ihn als Ausgangspunkt für die Berechnung der Suchrichtung $\mathbf{y}^{(m)}$ benötigen und nicht in jedem Schritt mit einer zusätzlichen Matrix-Vektor-Multiplikation neu konstruieren wollen. Stattdessen verwenden wir die bereits bekannte Gleichung

$$\mathbf{g}^{(m+1)} = \mathbf{g}^{(m)} + \theta \mathbf{L}\mathbf{y}^{(m)},$$

um den neuen Gradienten aus dem alten zu gewinnen.

Für die Berechnung der konjugierten Suchrichtung verwenden wir den *Gram-Schmidt-Orthogonalisierungsalgorithmus*, der auf der einfachen Idee beruht, von dem neuen Vektor ein geeignetes Vielfaches des alten Vektors abzuziehen. Wir verwenden also den Ansatz

$$\mathbf{y}^{(m+1)} = \mathbf{g}^{(m+1)} - \mu^{(m)} \mathbf{y}^{(m)}$$

und müssen $\mu^{(m)}$ so bestimmen, dass

$$\begin{aligned} 0 = \langle \mathbf{y}^{(m+1)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 &= \langle \mathbf{g}^{(m+1)} - \mu^{(m)} \mathbf{y}^{(m)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 \\ &= \langle \mathbf{g}^{(m+1)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 - \mu^{(m)} \langle \mathbf{y}^{(m)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2 \end{aligned}$$

gilt. Die Lösung ist offenbar

$$\mu^{(m)} = \frac{\langle \mathbf{g}^{(m+1)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2}{\langle \mathbf{y}^{(m)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2}.$$

Mit etwas Aufwand lässt sich beweisen, dass dieser Ansatz bereits sicher stellt, dass nicht nur $\mathbf{y}^{(m+1)}$ und $\mathbf{y}^{(m)}$, sondern *alle* $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(m+1)}$ konjugiert sind. Damit haben wir das gewünschte verbesserte Verfahren erhalten:

Definition 4.21 (Verfahren der konjugierten Gradienten) *Sei* $\mathbf{L} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *symmetrisch und positiv definit, und sei* $\mathbf{x}^{(0)} \in \mathbb{R}^{\mathcal{I}}$.

Wir setzen $\mathbf{g}^{(0)} := \mathbf{L}\mathbf{x}^{(0)} - \mathbf{b}$ *und* $\mathbf{y}^{(0)} := \mathbf{g}^{(0)}$.

Das Verfahren der konjugierten Gradienten *(auch bekannt als* cg-Verfahren*) definiert induktiv durch*

$$\begin{aligned}
\mathbf{a}^{(m)} &:= \mathbf{L}\mathbf{y}^{(m)},\\
\alpha^{(m)} &:= \langle \mathbf{y}^{(m)}, \mathbf{a}^{(m)} \rangle_2,\\
\theta^{(m)} &:= -\langle \mathbf{y}^{(m)}, \mathbf{g}^{(m)} \rangle_2 / \alpha^{(m)},\\
\mathbf{x}^{(m+1)} &:= \mathbf{x}^{(m)} + \theta^{(m)}\mathbf{y}^{(m)},\\
\mathbf{g}^{(m+1)} &:= \mathbf{g}^{(m)} + \theta^{(m)}\mathbf{a}^{(m)},\\
\mu^{(m)} &:= \langle \mathbf{g}^{(m+1)}, \mathbf{a}^{(m)} \rangle_2 / \alpha^{(m)},\\
\mathbf{y}^{(m+1)} &:= \mathbf{g}^{(m+1)} - \mu^{(m)}\mathbf{y}^{(m)} \qquad \text{für alle } m \in \mathbb{N}_0
\end{aligned}$$

eine Folge von Näherungslösungen $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ *des Gleichungssystems* $\mathbf{L}\mathbf{x} = \mathbf{b}$.

Wir können das Verfahren nicht weiter ausführen, falls der Nenner $\langle \mathbf{y}^{(m)}, \mathbf{L}\mathbf{y}^{(m)} \rangle_2$ verschwindet. Da $\mathbf{L}$ positiv definit ist, kann das aber nur geschehen, falls $\mathbf{y}^{(m)} = \mathbf{0}$ gilt. Das kann wiederum nur eintreten falls $\mathbf{g}^{(m)}$ im Aufspann der bisherigen Richtungen $\mathbf{y}^{(0)}, \dots, \mathbf{y}^{(m-1)}$ liegt. Da $\mathbf{x}^{(m)}$ optimal bezüglich dieser Richtungen ist, gilt aber auch

$$0 = \langle \mathbf{L}\mathbf{x}^{(m)} - \mathbf{b}, \mathbf{y}^{(\ell)} \rangle_2 = \langle \mathbf{g}^{(m)}, \mathbf{y}^{(\ell)} \rangle_2 \qquad \text{für alle } \ell \in \{0, \dots, m-1\}.$$

Damit steht $\mathbf{g}^{(m)}$ senkrecht auf dem Aufspann der bisherigen Suchrichtungen, ist aber gleichzeitig in diesem Aufspann enthalten. Der einzige Vektor, der diese beiden Eigenschaften vereint, ist der Nullvektor, es folgt also $\mathbf{g}^{(m)} = \mathbf{0}$. Das Verfahren ist also nur dann nicht weiter ausführbar, wenn die Lösung bereits gefunden ist.

5 Erhaltungsgleichungen und Sattelpunktprobleme

In vielen physikalischen Aufgabenstellungen treten Größen auf, die sich nicht verändern dürfen. Ein Beispiel ist die *Massenerhaltung*: In einem geschlossenen (nicht-relativistischen) System kann Materie weder entstehen noch verschwinden, also muss die Gesamtmasse des Systems immer gleich bleiben. Wenn wir das System approximieren, beispielsweise mittels eines Finite-Differenzen-Verfahrens, wäre es sinnvoll, dafür zu sorgen, dass das approximative System dieselben Erhaltungseigenschaften wie das ursprüngliche System aufweist.

5.1 Grundwasserströmung

Die Bewegung von Wasser in einem porösen Medium, beispielsweise in der Erde, kann mit Hilfe des *Darcy'schen Gesetzes* beschrieben werden. Dieses Gesetz stellt eine Beziehung zwischen dem *Druck* her, unter dem das Wasser steht, und dem *Fluss* des Wassers.

Der Druck in einem Gebiet $\Omega \subseteq \mathbb{R}^3$ ist eine skalarwertige Funktion $p : \Omega \to \mathbb{R}$, die jedem Punkt des Gebiets die Kraft zuordnet, die pro Fläche in ihm wirkt.

Der Fluss ist ein Vektorfeld $f : \Omega \to \mathbb{R}^3$, das angibt, wieviel Wasser pro Fläche austritt: Wenn $n \in \mathbb{R}^3$ der Normalenvektor einer Einheitsfläche ist, beschreibt $\langle f(x), n\rangle_2$, wieviel Wasser diese Fläche in Richtung des Vektors überschreitet.

Darcy'sches Gesetz. Das *Darcy'sche Gesetz* besagt, dass der Fluss direkt proportional zu dem Gradienten des Drucks ist, dass also

$$f(x) + k(x)\nabla p(x) = 0 \qquad \text{für alle } x \in \Omega \tag{5.1}$$

gilt. Die Größe $k(x)$ beschreibt die Durchlässigkeit des Materials in einem Punkt $x \in \Omega$: Wenn $k(x)$ groß ist, fließt das Wasser besonders schnell. Ein Blick auf die Taylor-Entwicklung zeigt, dass der Gradient gerade diejenige Richtung beschreibt, in der der Druck am schnellsten zunimmt, und die Gleichung besagt, dass das Wasser in die entgegengesetzte Richtung fließt. Diese Eigenschaft wird in der Strömungsdynamik als *Impulserhaltung* bezeichnet.

Massenerhaltung. Man sieht leicht, dass durch die Gleichung (5.1) Fluss und Druck nicht eindeutig bestimmt sind: Für eine beliebige Funktion φ können wir zu f das Produkt $k\nabla\varphi$ hinzuaddieren und φ von p subtrahieren, ohne die Gültigkeit der Gleichung zu

verlieren. Wir müssen noch eine weitere Gleichung hinzunehmen, die ein weiteres physikalisches Gesetz beschreibt: Wasser kann nicht einfach entstehen oder verschwinden, und es kann auch (fast) nicht komprimiert werden. Dieses Gesetz der *Massenerhaltung* lässt sich knapp durch die Gleichung

$$\int_{\partial\omega} \langle f(x), n(x)\rangle_2 \, dx = 0 \qquad \text{für alle Gebiete } \omega \subseteq \Omega \tag{5.2}$$

ausdrücken: Das Skalarprodukt $\langle f(x), n(x)\rangle_2$ des Flusses $f(x)$ in einem Randpunkt $x \in \partial\omega$ mit dem *äußeren Normalenvektor* $n(x)$ gibt an, wieviel an diesem Punkt aus dem Gebiet ω heraus fließt. Es ist entsprechend negativ, falls etwas in das Gebiet hinein fließt. Die Gleichung (5.2) besagt gerade, dass sich Ein- und Ausflüsse die Waage halten.

Das Darcy'sche Gesetz (5.1) gemeinsam mit der Massenerhaltungsgleichung (5.2) und geeigneten Randbedingungen führt zu einem System partieller Differentialgleichungen, dass sich lösen lässt. Wir sind selbstverständlich wieder daran interessiert, ein Verfahren zu entwickeln, mit dem sich diese Lösung praktisch approximieren lässt. Allerdings soll dabei die physikalisch wichtige Massenerhaltung nicht nur approximativ erfüllt bleiben, sondern exakt.

Diskretisierung der Massenerhaltung. Wir zuvor beschränken wir uns auf den Fall des Einheitswürfels $\Omega = (0,1)^3$, den wir wieder mit einem Gitter ausstatten. Für ein $n \in \mathbb{N}$ setzen wir

$$h := \frac{1}{n}, \qquad \bar{\omega} := \{ih \ : \ i \in \{0, \dots, n\}\}, \qquad \widehat{\omega} := \{ih \ : \ i \in \{1, \dots, n\}\}.$$

Der Würfel lässt sich in kleine Quader

$$Q_x := (x_1 - h, x_1) \times (x_2 - h, x_2) \times (x_3 - h, x_3) \qquad \text{für alle } x \in \mathcal{J} := \widehat{\omega}^3$$

zerlegen, und wir wollen auf jedem dieser Quader die Massenerhaltung gemäß (5.2) sicherstellen.

Dazu zerlegen wir den Rand des Quaders in seine sechs Seitenflächen, die wir in der Form

$$\begin{aligned}
F_{1,x} &:= \{x_1\} \times [x_2 - h, x_2] \times [x_3 - h, x_3] && \text{für alle } x \in \bar{\mathcal{I}}_1 := \bar{\omega} \times \widehat{\omega} \times \widehat{\omega}, \\
F_{2,x} &:= [x_1 - h, x_1] \times \{x_2\} \times [x_3 - h, x_3] && \text{für alle } x \in \bar{\mathcal{I}}_2 := \widehat{\omega} \times \bar{\omega} \times \widehat{\omega}, \\
F_{3,x} &:= [x_1 - h, x_1] \times [x_2 - h, x_2] \times \{x_3\} && \text{für alle } x \in \bar{\mathcal{I}}_3 := \widehat{\omega} \times \widehat{\omega} \times \bar{\omega}
\end{aligned}$$

darstellen. Als Normalenvektoren legen wir mit

$$n_1 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \qquad n_2 := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad n_3 := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

jeweils die Einheitsvektoren in positiver Koordinatenrichtung fest. Der Quader Q_x besitzt dann die linke und rechte Seitenfläche F_{1,x_1-h,x_2,x_3} und $F_{1,x}$ mit den Normalenvektoren

$-n_1$ und n_1, die obere und untere Seitenfläche F_{2,x_1,x_2-h,x_3} und $F_{2,x}$ mit den Normalenvektoren $-n_2$ und n_2 sowie die vordere und hintere Seitenfläche F_{3,x_1,x_2,x_3-h} und $F_{3,x}$ mit den Normalenvektoren $-n_3$ und n_3. Die Gleichung (5.2) nimmt die Form

$$\begin{aligned}
0 &= \int_{\partial Q_x} \langle f(y), n(y)\rangle_2 \, dy \\
&= -\int_{F_{1,x_1-h,x_2,x_3}} \langle f(y), n_1(y)\rangle_2 \, dy + \int_{F_{1,x}} \langle f(y), n_1(y)\rangle_2 \, dy \\
&\quad - \int_{F_{2,x_1,x_2-h,x_3}} \langle f(y), n_2(y)\rangle_2 \, dy + \int_{F_{2,x}} \langle f(y), n_2(y)\rangle_2 \, dy \\
&\quad - \int_{F_{3,x_1,x_2,x_3-h}} \langle f(y), n_3(y)\rangle_2 \, dy + \int_{F_{3,x}} \langle f(y), n_3(y)\rangle_2 \, dy
\end{aligned}$$

an. Diese Gleichung soll *exakt* gelten, wir können aber nicht das Verhalten der Funktion f auf jeder Seitenfläche durch wenige Koeffizienten vollständig beschreiben.

Stattdessen verwenden wir unmittelbar die Integrale selbst als Unbekannte unseres Gleichungssystems: Wir definieren

$$\begin{aligned}
f_{1,x} &:= \int_{F_{1,x}} \langle f(y), n_1(y)\rangle_2 \, dy && \text{für alle } x \in \bar{\mathcal{I}}_1, \\
f_{2,x} &:= \int_{F_{2,x}} \langle f(y), n_2(y)\rangle_2 \, dy && \text{für alle } x \in \bar{\mathcal{I}}_2, \\
f_{3,x} &:= \int_{F_{3,x}} \langle f(y), n_3(y)\rangle_2 \, dy && \text{für alle } x \in \bar{\mathcal{I}}_3
\end{aligned}$$

und schreiben die Massenerhaltungsgleichung als

$$0 = f_{1,x} - f_{1,x_1-h,x_2,x_3} + f_{2,x} - f_{2,x_1,x_2-h,x_3} + f_{3,x} - f_{3,x_1,x_2,x_3-h} \qquad \text{für alle } x \in \mathcal{J}.$$

Damit haben wir sie in ein System linearer Gleichungen überführt.

Diskretisierung der Darcy-Gleichung. Die Darcy-Gleichung wollen wir ebenfalls durch ein System linearer Gleichungen ersetzen. Da uns wegen unserer Wahl der Unbekannten $f(x)$ nicht unmittelbar zur Verfügung steht, integrieren wir (5.1) über eine Seitenfläche und erhalten

$$\begin{aligned}
0 &= f_{1,x} + \int_{F_{1,x}} k(y) \langle \nabla p(y), n_1(y)\rangle_2 \, dy \\
&= f_{1,x} + \int_{F_{1,x}} k(y) \partial_1 p(y) \, dy \qquad \text{für alle } x \in \mathcal{I}_1.
\end{aligned}$$

Wir approximieren die partielle Ableitung durch den zentralen Differenzenquotienten (2.10c) und gelangen zu

$$0 = f_{1,x} + \int_{F_{1,x}} k(y) \frac{p(y_1 + h/2, y_2, y_3) - p(y_1 - h/2, y_2, y_3)}{h} \, dy \qquad \text{für alle } x \in \mathcal{I}_1.$$

Damit hier keine Werte des Drucks außerhalb des Gebiets benötigt werden müssen wir uns auf die Indexmengen

$$\mathcal{I}_1 := \omega \times \widehat{\omega} \times \widehat{\omega}, \qquad \mathcal{I}_2 := \widehat{\omega} \times \omega \times \widehat{\omega}, \qquad \mathcal{I}_3 := \widehat{\omega} \times \widehat{\omega} \times \omega$$

mit

$$\omega := \{1, \ldots, n-1\}$$

beschränken, die zu den Flächen gehören, die nicht auf dem Rand des Gebiets liegen.

Am einfachsten wäre es, wenn wir das Integral auf der rechten Seite durch eine geeignet skalierte Auswertung des Drucks in einem einzelnen Punkt ersetzen könnten. Das ist tatsächlich möglich und führt auch lediglich zu einem Fehler, der proportional zu h^2 ist:

Lemma 5.1 (Mittelpunktregel) *Seien $h \in \mathbb{R}_{>0}$ und eine Funktion $g \in C^2[-h/2, h/2]$ gegeben. Dann existiert ein $\eta \in [-h/2, h/2]$ mit*

$$\int_{-h/2}^{h/2} g(s)\,ds - hg(0) = \frac{h^3}{24} g''(\eta).$$

Mit dieser Approximation folgt

$$\begin{aligned}
&\int_{F_{1,x}} k(y) \frac{p(y_1 + h/2, y_2, y_3) - p(y_1 - h/2, y_2, y_3)}{h}\,dy \\
&= \int_{-h}^{0}\int_{-h}^{0} k(x_1, x_2 + s, x_3 + t) \\
&\qquad \frac{p(x_1 + h/2, x_2 + s, x_3 + t) - p(x_1 - h/2, x_2 + s, x_3 + t)}{h}\,ds\,dt \\
&\approx h^2 k(x_1, x_2 - h/2, x_3 - h/2) \\
&\qquad \frac{p(x_1 + h/2, x_2 - h/2, x_3 - h/2) - p(x_1 - h/2, x_2 - h/2, x_3 - h/2)}{h},
\end{aligned}$$

also bietet es sich an, den Druck p durch seine Werte in den Mittelpunkten der Würfel Q_x darzustellen. Dazu führen wir

$$p_x := p(x_1 - h/2, x_2 - h/2, x_3 - h/2) \qquad \text{für alle } x \in \mathcal{J}$$

ein und schreiben die Approximation der Darcy-Gleichung in der kompakten Form

$$0 = f_{1,x} + hk(x_1, x_2 - h/2, x_3 - h/2)(p_{x_1+h,x_2,x_3} - p_x) \qquad \text{für alle } x \in \mathcal{I}_1.$$

Entsprechend erhalten wir auch

$$\begin{aligned}
0 &= f_{2,x} + hk(x_1 - h/2, x_2, x_3 - h/2)(p_{x_1,x_2+h,x_3} - p_x) && \text{für alle } x \in \mathcal{I}_2, \\
0 &= f_{3,x} + hk(x_1 - h/2, x_2 - h/2, x_3)(p_{x_1,x_2,x_3+h} - p_x) && \text{für alle } x \in \mathcal{I}_3.
\end{aligned}$$

Insgesamt haben wir die Darcy-Gleichung (5.1) und die Massenerhaltungsgleichung (5.2) durch das lineare Gleichungssystem

$$f_{1,x} + hk(x_1, x_2 - h/2, x_3 - h/2)(p_{x_1+h,x_2,x_3} - p_x) = 0 \qquad \text{für alle } x \in \mathcal{I}_1,$$

$$
\begin{aligned}
f_{2,x} + hk(x_1 - h/2, x_2, x_3 - h/2)(p_{x_1,x_2+h,x_3} - p_x) = 0 \quad &\text{für alle } x \in \mathcal{I}_2,\\
f_{3,x} + hk(x_1 - h/2, x_2 - h/2, x_3)(p_{x_1,x_2,x_3+h} - p_x) = 0 \quad &\text{für alle } x \in \mathcal{I}_3,\\
f_{1,x} - f_{1,x_1-h,x_2,x_3} + f_{2,x} - f_{2,x_1,x_2-h,x_3} + f_{3,x} - f_{3,x_1,x_2,x_3-h} = 0 \quad &\text{für alle } x \in \mathcal{J}
\end{aligned}
$$

ersetzt. Um die Symmetrie des Systems etwas deutlicher zu betonen dividieren wir die ersten drei Zeilen durch $hk(\ldots)$ und multiplizieren die letzte mit -1, um zu

$$
\begin{aligned}
\frac{f_{1,x}}{hk(x_1, x_2 - h/2, x_3 - h/2)} + (p_{x_1+h,x_2,x_3} - p_x) = 0 \quad &\text{für alle } x \in \mathcal{I}_1,\\
\frac{f_{2,x}}{hk(x_1 - h/2, x_2, x_3 - h/2)} + (p_{x_1,x_2+h,x_3} - p_x) = 0 \quad &\text{für alle } x \in \mathcal{I}_2,\\
\frac{f_{3,x}}{hk(x_1 - h/2, x_2 - h/2, x_3)} + (p_{x_1,x_2,x_3+h} - p_x) = 0 \quad &\text{für alle } x \in \mathcal{I}_3,\\
f_{1,x_1-h,x_2,x_3} - f_{1,x} + f_{2,x_1,x_2-h,x_3} - f_{2,x} + f_{3,x_1,x_2,x_3-h} - f_{3,x} = 0 \quad &\text{für alle } x \in \mathcal{J}
\end{aligned}
$$

zu gelangen. Indem wir die Flüsse zu Vektoren $\mathbf{f}_1 \in \mathbb{R}^{\mathcal{I}_1}$, $\mathbf{f}_2 \in \mathbb{R}^{\mathcal{I}_2}$ und $\mathbf{f}_3 \in \mathbb{R}^{\mathcal{I}_3}$ und die Drücke zu einem Vektor $\mathbf{p} \in \mathbb{R}^{\mathcal{J}}$ zusammenfassen, können wir das System kurz in der Form

$$
\begin{pmatrix}
\mathbf{A}_1 & & & \mathbf{B}_1\\
& \mathbf{A}_2 & & \mathbf{B}_2\\
& & \mathbf{A}_3 & \mathbf{B}_3\\
\mathbf{B}_1^* & \mathbf{B}_2^* & \mathbf{B}_3^* &
\end{pmatrix}
\begin{pmatrix}
\mathbf{f}_1\\ \mathbf{f}_2\\ \mathbf{f}_3\\ \mathbf{p}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{d}
\end{pmatrix}
\tag{5.3}
$$

darstellen, wobei $\mathbf{A}_1 \in \mathbb{R}^{\mathcal{I}_1 \times \mathcal{I}_1}$, $\mathbf{A}_2 \in \mathbb{R}^{\mathcal{I}_2 \times \mathcal{I}_2}$ und $\mathbf{A}_3 \in \mathbb{R}^{\mathcal{I}_3 \times \mathcal{I}_3}$ Diagonalmatrizen und $\mathbf{B}_1 \in \mathbb{R}^{\mathcal{I}_1 \times \mathcal{J}}$, $\mathbf{B}_2 \in \mathbb{R}^{\mathcal{I}_2 \times \mathcal{J}}$ und $\mathbf{B}_3 \in \mathbb{R}^{\mathcal{I}_3 \times \mathcal{J}}$ durch

$$
b_{1,ij} = \begin{cases} -1 & \text{falls } j = i,\\ 1 & \text{falls } j_1 = i_1 + h,\ j_2 = i_2,\ j_3 = i_3,\\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i \in \mathcal{I}_1,\ j \in \mathcal{J},
$$

$$
b_{2,ij} = \begin{cases} -1 & \text{falls } j = i,\\ 1 & \text{falls } j_1 = i_1,\ j_2 = i_2 + h,\ j_3 = i_3,\\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i \in \mathcal{I}_2,\ j \in \mathcal{J},
$$

$$
b_{3,ij} = \begin{cases} -1 & \text{falls } j = i,\\ 1 & \text{falls } j_1 = i_1,\ j_2 = i_2,\ j_3 = i_3 + h,\\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i \in \mathcal{I}_3,\ j \in \mathcal{J}
$$

gegeben sind und der Vektor $\mathbf{d} \in \mathbb{R}^{\mathcal{J}}$ für jedes $i \in \mathcal{J}$ die Flüsse über die Seitenflächen des Quaders Q_i aufnimmt, die auf dem Rand des Gebiets Ω liegen.

Die Matrix dieses Systems ist zwar symmetrisch, allerdings kann man sich leicht überlegen, dass sie wegen des rechten unteren Null-Blocks nicht positiv definit sein kann. Also lässt sich das Verfahren der konjugierten Gradienten nicht unmittelbar anwenden, wir müssen uns auf die Suche nach einem alternativen Lösungsverfahren begeben.

5.2 Uzawa-Verfahren

Allgemein betrachten wir das durch

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \tag{5.4}$$

für $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und $\mathbf{B} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ gegebene Gleichungssystem. In unserem Fall ist $\mathbf{A}$ positiv definit und symmetrisch, dann spricht man von einem *Sattelpunktproblem*.

Block-Elimination. Da die Matrix $\mathbf{A}$ positiv definit ist, ist sie auch invertierbar, also können wir die erste Zeile der Gleichung (5.4) mit $\mathbf{B}^*\mathbf{A}^{-1}$ multiplizieren und von der zweiten Zeile subtrahieren, um

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ & -\mathbf{B}^*\mathbf{A}^{-1}\mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 - \mathbf{B}^*\mathbf{A}^{-1}\mathbf{b}_1 \end{pmatrix}$$

zu erhalten. Anschaulich entspricht dieser Schritt einer Gauß-Elimination mit Matrizen anstelle von Koeffizienten. Indem wir die zweite Zeile mit -1 multiplizieren, folgt

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ & \mathbf{B}^*\mathbf{A}^{-1}\mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{B}^*\mathbf{A}^{-1}\mathbf{b}_1 - \mathbf{b}_2 \end{pmatrix}. \tag{5.5}$$

Die erste Diagonalmatrix ist nach Definition positiv definit. Die zweite Diagonalmatrix $\mathbf{S} := \mathbf{B}^*\mathbf{A}^{-1}\mathbf{B}$ bezeichnet man als das *Schur-Komplement*. Wir können festhalten, dass $\mathbf{S}$ symmetrisch ist.

Schur-Komplement-System. Damit wir das System (5.5) durch einfaches Block-Rückwärtseinsetzen lösen können, müssen wir die Lösbarkeit des Systems

$$\mathbf{S}\mathbf{x}_2 = \mathbf{B}^*\mathbf{A}^{-1}\mathbf{b}_1 - \mathbf{b}_2 \tag{5.6}$$

untersuchen. Weil $\mathbf{A}$ positiv definit ist, gilt mit Lemma 4.14

$$\langle \mathbf{A}^{-1}\mathbf{x}, \mathbf{x}\rangle_2 = \langle \mathbf{A}^{-1}\mathbf{A}\mathbf{A}^{-1}\mathbf{x}, \mathbf{x}\rangle_2 = \langle \mathbf{A}(\mathbf{A}^{-1}\mathbf{x}), \mathbf{A}^{-1}\mathbf{x}\rangle_2 > 0 \qquad \text{für alle } x \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\},$$

also muss auch $\mathbf{A}^{-1}$ positiv definit sein. Daraus folgt

$$\langle \mathbf{B}^*\mathbf{A}^{-1}\mathbf{B}\mathbf{x}, \mathbf{x}\rangle_2 = \langle \mathbf{A}^{-1}\mathbf{B}\mathbf{x}, \mathbf{B}\mathbf{x}\rangle_2 \geq 0 \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{J}},$$

also ist das Schur-Komplement $\mathbf{S} = \mathbf{B}^*\mathbf{A}^{-1}\mathbf{B}$ positiv semidefinit und der Kern dieser Matrix ist gerade der Kern der Matrix $\mathbf{B}$.

Für die Lösbarkeit des Gleichungssystems (5.6) ist eher das Bild der Matrix $\mathbf{S}$ von Bedeutung, das wir als nächstes untersuchen.

Erinnerung 5.2 (Dimensionssatz) *Sei* $\mathbf{A} \in \mathbb{R}^{n\times m}$. Bild *und* Kern *der Matrix sind durch*

$$\begin{aligned}\operatorname{Bild}\mathbf{A} &:= \{\mathbf{y}\in\mathbb{R}^n \ : \ \textit{es existiert ein } \mathbf{x}\in\mathbb{R}^m \textit{ mit } \mathbf{y}=\mathbf{A}\mathbf{x}\},\\ \operatorname{Kern}\mathbf{A} &:= \{\mathbf{x}\in\mathbb{R}^m \ : \ \mathbf{A}\mathbf{x}=\mathbf{0}\}\end{aligned}$$

definierte Teilräume der Räume $\mathbb{R}^n$ *und* $\mathbb{R}^m$. *Ihre Dimensionen erfüllen die Gleichung*

$$\dim\operatorname{Bild}\mathbf{A} + \dim\operatorname{Kern}\mathbf{A} = n.$$

Nach Lemma 4.14 gilt

$$\langle \mathbf{B}^*\mathbf{x},\mathbf{y}\rangle_2 = \langle \mathbf{x},\mathbf{B}\mathbf{y}\rangle_2 = \langle \mathbf{x},\mathbf{0}\rangle_2 = 0 \qquad \text{für alle } \mathbf{x}\in\mathbb{R}^{\mathcal{I}},\ \mathbf{y}\in\operatorname{Kern}B, \tag{5.7}$$

also steht das Bild der Matrix $\mathbf{B}^*$ senkrecht auf dem Kern der Matrix $\mathbf{B}$, so dass wir

$$\dim\operatorname{Bild}\mathbf{B}^* + \dim\operatorname{Kern}\mathbf{B} \le n := \#\mathcal{J} \tag{5.8}$$

erhalten. Nach Dimensionssatz gilt

$$\dim\operatorname{Bild}\mathbf{S} + \dim\operatorname{Kern}\mathbf{S} = n,$$

und da wir bereits die Gleichungen

$$\operatorname{Bild}\mathbf{S}\subseteq\operatorname{Bild}\mathbf{B}^*, \qquad \operatorname{Kern}\mathbf{S} = \operatorname{Kern}\mathbf{B}$$

kennen, folgt

$$\dim\operatorname{Bild}\mathbf{B}^* + \dim\operatorname{Kern}\mathbf{B} \ge \dim\operatorname{Bild}\mathbf{S} + \dim\operatorname{Kern}\mathbf{S} = n.$$

Mit (5.8) erhalten wir so

$$\dim\operatorname{Bild}\mathbf{B}^* + \dim\operatorname{Kern}\mathbf{B} = n \tag{5.9}$$

und damit auch

$$\dim\operatorname{Bild}\mathbf{B}^* = n - \dim\operatorname{Kern}\mathbf{B} = n - \dim\operatorname{Kern}\mathbf{S} = \dim\operatorname{Bild}\mathbf{S}.$$

Aus $\operatorname{Bild}\mathbf{S}\subseteq\operatorname{Bild}\mathbf{B}^*$ folgt damit

$$\operatorname{Bild}\mathbf{S} = \operatorname{Bild}\mathbf{B}^*,$$

das Bild des Schur-Komplements ist also gerade das Bild der Matrix $\mathbf{B}^*$. Damit (5.6) lösbar ist, muss also die rechte Seite im Bild der Matrix $\mathbf{B}^*$ liegen. Das ist offenbar genau dann der Fall, wenn $\mathbf{b}_2$ in diesem Bildraum liegt, und das ist wegen (5.7) und (5.9) wiederum genau dann der Fall, wenn $\mathbf{b}_2$ senkrecht auf dem Kern der Matrix $\mathbf{B}$ steht.

Für das die Grundwasserströmung beschreibende System (5.3) entspricht $\mathbf{B}$ dem Gradienten, der nur für konstante Funktionen verschwindet, also muss $\mathbf{b}_2$ senkrecht auf dem konstanten Vektor stehen: Die Summe über alle Komponenten muss gleich null sein, damit eine Lösung existiert. Das ist physikalisch sinnvoll, denn sonst wären die Mengen des über die Seitenflächen des Würfels Ω ein- und ausfließenden Wassers nicht im Gleichgewicht.

cg-Verfahren. Das cg-Verfahren (vgl. Definition 4.21) lässt sich auch auf selbstadjungierte positiv *semi*definite Matrizen anwenden, wir können also das System (5.6) mit seiner Hilfe angehen.

Die Berechnung des Gradienten im m-ten Schritt nimmt dann die Form

$$\mathbf{g}_2^{(m)} = \mathbf{S}\mathbf{x}_2^{(m)} - (\mathbf{B}^*\mathbf{A}^{-1}\mathbf{b}_1 - \mathbf{b}_2) = \mathbf{B}^*\mathbf{A}^{-1}(\mathbf{B}\mathbf{x}_2^{(m)} - \mathbf{b}_1) + \mathbf{b}_2$$

an, der Vektor $\mathbf{a}_2^{(m)}$ berechnet sich durch

$$\mathbf{a}_2^{(m)} = \mathbf{S}\mathbf{y}^{(m)} = \mathbf{B}^*\mathbf{A}^{-1}\mathbf{B}\mathbf{y}_2^{(m)}.$$

Die Koeffizienten $\alpha^{(m)}$ und $\theta^{(m)}$ lassen sich wie zuvor durch

$$\alpha^{(m)} = \langle \mathbf{a}_2^{(m)}, \mathbf{y}_2^{(m)} \rangle_2, \qquad \theta^{(m)} = -\langle \mathbf{g}_2^{(m)}, \mathbf{y}_2^{(m)} \rangle_2 / \alpha^{(m)}$$

berechnen, die neuen Iterierten ergeben sich per

$$\mathbf{x}_2^{(m+1)} = \mathbf{x}_2^{(m)} + \theta^{(m)}\mathbf{y}_2^{(m)}, \qquad \mathbf{g}_2^{(m+1)} = \mathbf{g}_2^{(m)} + \theta^{(m)}\mathbf{a}_2^{(m)}, \tag{5.10}$$

und die nächste Suchrichtung können wir durch

$$\mu^{(m)} = \langle \mathbf{g}_2^{(m+1)}, \mathbf{a}_2^{(m)} \rangle_2 / \alpha^{(m)}, \qquad \mathbf{y}^{(m+1)} = \mathbf{g}^{(m+1)} - \mu^{(m)}\mathbf{y}^{(m)}$$

bestimmen.

Uzawa-Verfahren. Wir sind nicht nur an $\mathbf{x}_2$, sondern auch an $\mathbf{x}_1$ interessiert. Ein naiver Zugang bestünde darin, $\mathbf{x}_2$ hinreichend genau zu berechnen und dann in (5.5) den Vektor $\mathbf{x}_1$ durch Rückwärtseinsetzen zu bestimmen, also

$$\mathbf{A}\mathbf{x}_1 = \mathbf{b}_1 - \mathbf{B}\mathbf{x}_2$$

zu lösen. Die Idee des *Uzawa-Verfahrens* besteht darin, die Berechnung des Vektors $\mathbf{x}_1$ simultan zu der Berechnung des Vektors $\mathbf{x}_2$ durchzuführen: Wir definieren

$$\mathbf{x}_1^{(m)} = \mathbf{A}^{-1}(\mathbf{b}_1 - \mathbf{B}\mathbf{x}_2^{(m)}) \qquad \text{für alle } m \in \mathbb{N}_0$$

und stellen fest, dass dann

$$\mathbf{g}_2^{(m)} = \mathbf{B}^*\mathbf{A}^{-1}(\mathbf{B}\mathbf{x}_2^{(m)} - \mathbf{b}_1) + \mathbf{b}_2 = \mathbf{b}_2 - \mathbf{B}^*\mathbf{x}_1^{(m)} \qquad \text{für alle } m \in \mathbb{N}_0$$

gilt, zumindest im ersten Schritt wird $\mathbf{x}_1^{(0)}$ also „nebenbei“ ausgerechnet.

Durch Einsetzen von (5.10) in die Gleichung für $\mathbf{x}_1^{(m+1)}$ erhalten wir

$$\begin{aligned}\mathbf{x}_1^{(m+1)} &= \mathbf{A}^{-1}(\mathbf{b}_1 - \mathbf{B}\mathbf{x}_2^{(m+1)}) = \mathbf{A}^{-1}(\mathbf{b}_1 - \mathbf{B}(\mathbf{x}_2^{(m)} + \theta^{(m)}\mathbf{y}_2^{(m)})) \\ &= \mathbf{x}_1^{(m)} - \theta^{(m)}\mathbf{A}^{-1}\mathbf{B}\mathbf{y}_2^{(m)} \qquad \text{für alle } m \in \mathbb{N}_0,\end{aligned}$$

und der Vektor $\mathbf{y}_1^{(m)} := \mathbf{A}^{-1}\mathbf{B}\mathbf{y}_2^{(m)}$ muss bei der Berechnung des Vektors $\mathbf{a}_2^{(m)}$ ohnehin bestimmt werden. Wir können also mit einer einzigen Linearkombination dafür sorgen, dass in jedem Schritt das zu $\mathbf{x}_2^{(m)}$ passende $\mathbf{x}_1^{(m)}$ zu unserer Verfügung steht. In jedem Schritt erfordert die Uzawa-Iteration also je eine Matrix-Vektor-Multiplikation mit $\mathbf{B}$ und $\mathbf{B}^*$ und das Lösen eines linearen Gleichungssystems mit der Matrix $\mathbf{A}$. In unserem Fall ist $\mathbf{A}$ eine Diagonalmatrix, so dass sich $\mathbf{A}^{-1}$ sehr einfach auswerten lässt.

Definition 5.3 (Uzawa-Verfahren) *Sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ symmetrisch und positiv definit, sei $\mathbf{B} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$, und sei eine Anfangsnäherung $\mathbf{x}_2^{(0)} \in \mathbb{R}^{\mathcal{J}}$ gegeben.*

Wir setzen $\mathbf{x}_1^{(0)} := \mathbf{A}^{-1}(\mathbf{b}_1 - \mathbf{B}\mathbf{x}_2^{(0)})$, $\mathbf{g}_2^{(0)} := \mathbf{b}_2 - \mathbf{B}^*\mathbf{x}_1^{(0)}$ und $\mathbf{y}_2^{(0)} := \mathbf{g}_2^{(0)}$.

Das Uzawa-Verfahren *definiert induktiv durch*

$$\begin{aligned}
\mathbf{y}_1^{(m)} &:= \mathbf{A}^{-1}\mathbf{B}\mathbf{y}_2^{(m)},\\
\mathbf{a}_2^{(m)} &:= \mathbf{B}^*\mathbf{y}_1^{(m)},\\
\alpha^{(m)} &:= \langle \mathbf{a}_2^{(m)}, \mathbf{y}_2^{(m)}\rangle_2,\\
\theta^{(m)} &:= -\langle \mathbf{g}_2^{(m)}, \mathbf{y}_2^{(m)}\rangle_2/\alpha^{(m)},\\
\mathbf{x}_2^{(m+1)} &:= \mathbf{x}_2^{(m)} + \theta^{(m)}\mathbf{y}_2^{(m)},\\
\mathbf{x}_1^{(m+1)} &:= \mathbf{x}_1^{(m)} - \theta^{(m)}\mathbf{y}_1^{(m)},\\
\mathbf{g}_2^{(m+1)} &:= \mathbf{g}_2^{(m)} + \theta^{(m)}\mathbf{a}_2^{(m)},\\
\mu^{(m)} &:= \langle \mathbf{g}_2^{(m+1)}, \mathbf{a}_2^{(m)}\rangle_2/\alpha^{(m)},\\
\mathbf{y}_2^{(m+1)} &:= \mathbf{g}_2^{(m+1)} - \mu^{(m)}\mathbf{y}_2^{(m)} \qquad \text{für alle } m \in \mathbb{N}_0
\end{aligned}$$

eine Folge von Näherungslösungen $(\mathbf{x}_1^{(m)}, \mathbf{x}_2^{(m)})$ des Gleichungssystems

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}.$$

Die Folge der Vektoren $\mathbf{x}_2^{(m)}$ entspricht dabei der Folge, die das Verfahren der konjugierten Gradienten (vgl. Definition 4.21) für das Schur-Komplement-System (5.6) berechnet.

6 Grundlagen des Finite-Elemente-Verfahrens

Bisher haben wir die Diskretisierung partieller Differentialgleichungen lediglich für den Fall behandelt, dass das zugrundeliegende Gebiet der Einheitswürfel ist. Die bisher verwendeten Verfahren lassen sich innerhalb gewisser Grenzen auch für allgemeinere Gebiete einsetzen, allerdings müssen dafür häufig Abstriche bei der Konvergenz oder bei den Eigenschaften der entstehenden Matrix, beispielsweise ihrer Symmetrie, hingenommen werden.

In diesem Kapitel widmen wir uns dem *Finite-Elemente-Verfahren*, das es uns ermöglicht, sowohl relativ allgemeine Gebiete als auch eine breite Auswahl an Differentialgleichungen auf diesen Gebieten elegant zu behandeln.

6.1 Darstellung eines Gebiets

Ein erster Schritt auf dem Weg zu einem allgemeineren Diskretisierungsverfahren besteht darin, zu klären, wie wir allgemeine Gebiete im Rechner effizient darstellen können.

Der Einfachheit halber beschränken wir uns dabei auf Polygon- und Polyedergebiete, also auf Gebiete, die sich aus Dreiecken oder Tetraedern zusammensetzen lassen. Tetraeder, Dreiecke, Kanten und Punkte lassen sich dabei einheitlich als *Simplizes* schreiben.

Definition 6.1 (Simplex) *Sei $d \in \mathbb{N}$ und sei $r \in \{0, \dots, d\}$. Wir bezeichnen mit*

$$\mathcal{S}_r := \{t \subseteq \mathbb{R}^d \ : \ \#t = r + 1\}$$

die Menge aller Mengen von $r + 1$ Punkten im d-dimensionalen Raum.

Die konvexe Hülle einer Menge $t \in \mathcal{S}_r$, also

$$\omega_t := \left\{ \sum_{x \in t} \alpha_x x \ : \ \alpha_x \in [0,1] \text{ für alle } x \in t, \ \sum_{x \in t} \alpha_x = 1 \right\},$$

*nennen wir einen r-*dimensionalen Simplex *(Plural* Simplizes*) mit* Eckpunkten *t.*

Den r-dimensionalen Simplex ω_t zu Eckpunkten $t \in \mathcal{S}_r$ nennen wir regulär*, falls ein $x_0 \in t$ so existiert, dass die Menge*

$$\{x - x_0 \ : \ x \in t, \ x \neq x_0\}$$

linear unabhängig ist. In diesem Fall ist es egal, welches $x_0 \in t$ wir gewählt haben.

Einen nicht regulären Simplex nennen wir ausgeartet*.*

Einen nulldimensionalen Simplex nennen wir einen Knoten*, einen eindimensionalen eine* Kante*, einen zweidimensionalen ein* Dreieck *und einen dreidimensionalen einen* Tetraeder*.*

Beispielsweise definiert

$$t := \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\}$$

einen zweidimensionalen regulären Simplex im dreidimensionalen Raum, nämlich ein Dreieck, während

$$t := \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix} \right\}$$

zu einem ausgearteten Simplex führt.

Als Beispiele für dreidimensionale Simplizes betrachten wir

$$t := \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\},$$

die Eckpunktmenge eines regulären Tetraeders, und

$$t := \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\},$$

die Eckpunkte eines ausgearteten Tetraeders.

Unser Ziel ist es, aus Simplizes kompliziertere Gebiete zusammenzusetzen. Dabei ist uns wichtig, dass wir auf den so zusammengesetzten Gebieten effizient rechnen können, dass also beispielsweise die Simplizes sich (abgesehen von Randpunkten) nicht überschneiden. Aus gewissen Gründen ist es auch sinnvoll, sogenannte *hängende Knoten* zu verbieten, also Knoten eines Simplex', die auf einer Kante oder einem Dreieck eines anderen Simplex' liegen. Entsprechend sind auch *hängende Kanten* nicht willkommen, also Kanten eines Simplex', die auf einem Dreieck eines anderen liegen.

Diese Bedingung können wir für beliebige Simplizes ω_t, ω_s knapp durch

$$\omega_t \cap \omega_s = \omega_{t \cap s}$$

ausdrücken: Der Schnitt der beiden Simplizes muss wieder ein Simplex sein, der von gemeinsamen Eckpunkten erzeugt wird.

Definition 6.2 (Triangulation) *Sei $\Omega \subseteq \mathbb{R}^d$ ein Gebiet. Eine Menge $\mathcal{T} \subseteq \mathcal{S}_d$ nennen wir eine* Triangulation *des Gebiets Ω, falls die Bedingungen*

$$\bar{\Omega} = \bigcup_{t \in \mathcal{T}} \omega_t, \tag{6.1a}$$

$$\omega_t \cap \omega_s = \omega_{t\cap s} \qquad \text{für alle } t, s \in \mathcal{T} \tag{6.1b}$$

$$\omega_t \text{ ist regulär} \qquad \text{für alle } t \in \mathcal{T} \tag{6.1c}$$

gelten.

Bemerkung 6.3 (Implementierung) *Bei der Beschreibung einer Triangulation im Rechner ist es häufig sinnvoll, Simplizes höherer Dimension aus solchen niedrigerer Dimension zusammenzusetzen. Knoten werden dann einfach durch ihre Koordinaten dargestellt, Kanten als Paare zweier Knoten, Dreiecke als Tripel dreier Kanten und Tetraeder als Quadrupel vierer Dreiecke. Dieser Zugang bietet den Vorteil, dass alle für die Definition relevanten mathematischen Objekte auch in der Implementierung eine Entsprechung finden.*

Er hat allerdings den Nachteil, dass beispielsweise der Zugriff auf die Eckpunkte eines Tetraeders nicht unmittelbar möglich ist, sondern wir sie auf einem Umweg über Dreiecke und Kanten konstruieren müssen. Beispielsweise können wir den Eckpunkt gegenüber einem Dreieck ermitteln, indem wir den gemeinsamen Punkt zweier Kanten finden, die nicht zu dem Dreieck gehören.

6.2 Variationsformulierung

Wenn sich Triangulationen eignen, um Gebiete darzustellen, stellt sich die Frage, ob wir mit ihrer Hilfe auch Differentialgleichungen diskretisieren können. Wir könnten beispielsweise versuchen, aus den Knoten einer Triangulation Differenzenquotienten zu konstruieren, mit denen sich Ableitungen approximieren lassen. Dieser Ansatz ist allerdings relativ aufwendig, insbesondere für dreidimensionale Gebiete. Wesentlich attraktiver ist der Zugang über eine *Variationsformulierung*, die wir nun herleiten werden.

Als Beispiel wählen wir die Potentialgleichung auf einem Gebiet $\Omega \subseteq \mathbb{R}^d$ mit Nullrandbedingung: Gegeben ist eine Funktion $f \in C(\Omega)$, gesucht ist eine Funktion $u \in C^2(\bar{\Omega})$, die

$$-\nabla \cdot \nabla u(x) = f(x) \qquad \text{für alle } x \in \Omega, \tag{6.2a}$$

$$u(x) = 0 \qquad \text{für alle } x \in \partial\Omega \tag{6.2b}$$

erfüllt. Gerade bei Gebieten mit Ecken oder Kanten lässt sich zeigen, dass eine Lösung, sofern sie denn überhaupt existiert, nicht in allen Punkten des Gebiets zweimal stetig differenzierbar sein kann.

Um dieses Problem zu vermeiden gehen wir zu einem gewichteten Mittel über: Wir setzen voraus, dass das Gebiet Ω beschränkt ist. Dann können wir die Gleichung (6.2a) mit Funktionen $v \in C(\Omega)$ multiplizieren und beide Seiten integrieren, um

$$-\int_\Omega v(x)\nabla \cdot \nabla u(x)\,dx = \int_\Omega v(x)f(x)\,dx \qquad \text{für alle } v \in C(\Omega) \tag{6.3}$$

zu erhalten. Da wir nicht erwarten können, dass u immer zweimal differenzierbar ist, sind wir daran interessiert, eine Formulierung zu finden, die ohne diese Eigenschaft auskommt.

Eine Lösung ist die *partielle Integration*, die im eindimensionalen Fall die Gestalt

$$\int_a^b v(x)u'(x)\,dx = -\int_a^b v'(x)u(x)\,dx + v(b)u(b) - v(a)u(a) \tag{6.4}$$

für Funktionen $u, v \in C^1[a,b]$ annimmt und es uns offenbar ermöglicht, Ableitungen von u zu v wechseln zu lassen. Im mehrdimensionalen Fall gilt eine ähnliche Gleichung.

Erinnerung 6.4 (Gauß-Integralsatz) *Sei eine Funktion $u \in C^1(\bar{\Omega}, \mathbb{R}^d)$ gegeben. Mit $n : \partial\Omega \to \mathbb{R}^d$ bezeichnen wir den* äußeren Einheitsnormalenvektor*, der jedem Punkt des Rands $\partial\Omega$ einen Einheitsvektor zuordnet, der senkrecht auf dem Rand steht und aus dem Gebiet heraus weist. Dann gilt*

$$\int_\Omega \nabla \cdot u(x)\,dx = \int_{\partial\Omega} \langle n(x), u(x)\rangle_2\,dx. \tag{6.5}$$

In einer sehr vereinfachten Form ist er uns schon bei der Herleitung der Divergenz (vgl. Definition 4.6) begegnet.

Erinnerung 6.5 (Partielle Integration) *Seien $u \in C^1(\bar{\Omega}, \mathbb{R}^d)$ und $v \in C^1(\bar{\Omega}, \mathbb{R})$ gegeben. Aus der Produktregel folgt*

$$\nabla \cdot (vu)(x) = \langle \nabla v(x), u(x)\rangle_2 + v(x)\nabla \cdot u(x) \qquad \text{für alle } x \in \Omega,$$

so dass wir durch Einsetzen in den Gauß-Integralsatz

$$\int_\Omega \langle \nabla v(x), u(x)\rangle_2\,dx + \int_\Omega v(x)\nabla \cdot u(x)\,dx = \int_\Omega \nabla \cdot (vu)(x)\,dx = \int_{\partial\Omega} \langle n(x), (vu)(x)\rangle_2\,dx$$

erhalten. Indem wir den linken Term auf die rechte Seite bringen und die Zahlen $v(x)$ aus dem rechten Skalarprodukt herausziehen erhalten wir

$$\int_\Omega v(x)\nabla \cdot u(x)\,dx = -\int_\Omega \langle \nabla v(x), u(x)\rangle_2\,dx + \int_{\partial\Omega} v(x)\langle n(x), u(x)\rangle_2\,dx \tag{6.6}$$

als mehrdimensionales Gegenstück der Formel (6.4).

Wir wollen die Gleichung (6.6) auf (6.3) anwenden, um die Divergenz von u auf v zu verlagern. Das ist nur zulässig, falls v stetig differenzierbar ist. Um uns störende Randterme zu ersparen setzen wir zusätzlich voraus, dass v auf dem Rand des Gebiets verschwinden soll, wir betrachten also nur Funktionen aus dem Raum

$$\begin{aligned} C_0^1(\Omega) &:= C_0^1(\Omega, \mathbb{R}) \text{ mit} \\ C_0^1(\Omega, \mathbb{R}^d) &:= \{v \in C(\bar{\Omega}, \mathbb{R}^d) \ : \ v|_\Omega \in C^1(\Omega, \mathbb{R}^d),\ v|_{\partial\Omega} = 0\}. \end{aligned}$$

Dann folgt aus (6.3) mit partieller Integration

$$\int_\Omega \langle \nabla v(x), \nabla u(x)\rangle_2\,dx = \int_\Omega v(x)f(x)\,dx \qquad \text{für alle } v \in C_0^1(\Omega).$$

Da in dieser Formulierung nur noch die ersten Ableitungen der Funktion u auftreten, genügt es, $u \in C_0^1(\Omega)$ vorauszusetzen. Wir haben also eine neue Aufgabenstellung gefunden:

Gegeben ist $f \in C(\Omega)$, gesucht ist $u \in C_0^1(\Omega)$ mit

$$\int_\Omega \langle \nabla v(x), \nabla u(x) \rangle_2 \, dx = \int_\Omega v(x) f(x) \, dx \qquad \text{für alle } v \in C_0^1(\Omega). \tag{6.7}$$

Derartige Aufgaben nennt man *Variationsaufgaben*, weil nach einer Lösung $u \in C_0^1(\Omega)$ gesucht ist, die für *alle* $v \in C_0^1(\Omega)$ gilt, also für *variierende* „Testfunktionen".

Die Variationsformulierung ist in unserem Fall deutlich schwächer als die ursprüngliche Gleichung: Statt zweifacher Differenzierbarkeit genügt die einfache, und statt der punktweisen Gültigkeit der Differentialgleichung genügt die Gültigkeit im Integral-Mittel.

Die Bedingung ist allerdings leider immer noch nicht schwach genug: In vielen Anwendungsfällen wird sich nicht mal eine einmal stetig differenzierbare Funktion u finden lassen, die die Gleichung (6.7) erfüllt.

6.3 Hilbert-Räume

Unser Ziel besteht darin, Bedingungen zu finden, unter denen die Variationsaufgabe (6.7) eine, eventuell geeignet verallgemeinerte, Lösung besitzt.

Die Behandlung von Variationsaufgaben ist besonders elegant möglich, wenn die beteiligten Vektorräume mit einer geeignet gewählten Verallgemeinerung des *Skalarprodukts* ausgestattet werden.

Wir fixieren allgemein einen $\mathbb{R}$-Banach-Raum V und bezeichnen seine Norm mit $\|\cdot\|_V$.

Definition 6.6 (Bilinearform) *Eine Abbildung $a : V \times V \to \mathbb{R}$ nennen wir eine* Bilinearform*, falls*

$$a(v + \alpha w, u) = a(v, u) + \alpha a(w, u), \tag{6.8a}$$
$$a(v, u + \alpha w) = a(v, u) + \alpha a(v, w) \qquad \textit{für alle } u, v, w \in V,\ \alpha \in \mathbb{R} \tag{6.8b}$$

gelten, falls a also linear in beiden Argumenten ist.

Wir können uns Bilinearformen als Verallgemeinerung von Matrizen vorstellen, indem wir das euklidische Skalarprodukt verwenden: Für eine Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ ist

$$a(\mathbf{v}, \mathbf{u}) := \langle \mathbf{v}, \mathbf{A}\mathbf{u} \rangle_2 \qquad \text{für alle } \mathbf{v}, \mathbf{u} \in \mathbb{R}^{\mathcal{I}} \tag{6.9}$$

eine Bilinearform. Uns bereits bekannte Eigenschaften von Matrizen übertragen sich auf Bilinearformen.

Definition 6.7 (Symmetrisch) *Eine Bilinearform a nennen wir* symmetrisch*, falls*

$$a(u, v) = a(v, u) \qquad \textit{für alle } u, v \in V. \tag{6.10}$$

Definition 6.8 (Positiv definit) *Eine Bilinearform a nennen wir* positiv definit*, falls*

$$a(u, u) > 0 \qquad \textit{für alle } u \in V \setminus \{0\}. \tag{6.11}$$

Übungsaufgabe 6.9 (Bilinearformen und Matrizen) *Sei $a : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ eine Bilinearform. Beweisen Sie, dass eine Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ existiert, die (6.9) erfüllt.*

Zeigen Sie, dass die Matrix $\mathbf{A}$ genau dann symmetrisch ist, falls die Bilinearform a diese Eigenschaft besitzt.

Das euklidische Skalarprodukt ist selbst eine symmetrische und wegen $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x}\rangle_2$ auch positiv definite Bilinearform. An diesen beiden Eigenschaften können wir uns orientieren, um auch allgemeinere Skalarprodukte einzuführen.

Definition 6.10 (Skalarprodukt) *Eine symmetrische und positiv definite Bilinearform $a : V \times V \to \mathbb{R}$ nennen wir ein* Skalarprodukt *auf V.*

Definition 6.11 (Hilbert-Raum) *Sei $a : V \times V \to \mathbb{R}$ ein Skalarprodukt auf V.*

Falls die Norm durch das Skalarprodukt gemäß

$$\|u\|_V = \sqrt{a(u,u)} \qquad \text{für alle } u \in V \tag{6.12}$$

induziert wird, nennen wir V einen Hilbert-Raum *und schreiben das Skalarprodukt als*

$$\langle v, u\rangle_V := a(v,u) \qquad \text{für alle } u, v \in V.$$

Hilbert-Räume können wir uns als Verallgemeinerungen des Raums $\mathbb{R}^n$ vorstellen, bei denen viele wichtige Eigenschaften erhalten bleiben und beispielsweise bei der Lösung von Gleichungssystemen verwendet werden können.

Es stellt sich die Frage, ob zu einem Hilbert-Raum zwei verschiedene Skalarprodukte gehören können, die beide die Gleichung (6.12) erfüllen. Diese Frage können wir beantworten, indem wir aus

$$\begin{aligned} \|v+u\|_V^2 &= \langle v+u, v+u\rangle_V \\ &= \langle v, v\rangle + \langle v, u\rangle_V + \langle u, v\rangle_V + \langle v, v\rangle_V \\ &= \|v\|_V^2 + 2\langle v, u\rangle_V + \|u\|_V^2 \qquad \text{für alle } u, v \in V \end{aligned}$$

die *Polarisationsgleichung*

$$\|v+u\|_V^2 - \|v-u\|_V^2 = 4\langle v, u\rangle_V \qquad \text{für alle } u, v \in V$$

herleiten und daraus folgern, dass das Skalarprodukt bereits durch die Norm eindeutig festgelegt ist.

Eine weitere wichtige Konsequenz der Gleichung (6.12) ist die folgende Aussage, mit deren Hilfe sich Skalarprodukte beschränken lassen.

Lemma 6.12 (Cauchy-Schwarz) *Sei V ein Hilbert-Raum. Dann gilt*

$$|\langle v, u\rangle_V| \leq \|v\|_V \|u\|_V \qquad \text{für alle } u, v \in V. \tag{6.13}$$

Gleichheit gilt genau dann, wenn u und v linear abhängig sind.

Beweis. Seien $u, v \in V$ gegeben. Falls $v = 0$ gelten sollte, folgt die Behauptung aus

$$\langle v, u\rangle_V = \langle 0, u\rangle_V = 0 = \|0\|_V \|u\|_V = \|v\|_V \|u\|_V.$$

Wir können uns also auf den Fall $v \neq 0$ beschränken. In Hinblick auf den zweiten Teil der Behauptung erscheint es sinnvoll, nach einem Vielfachen αv des Vektors v mit $\alpha \in \mathbb{R}$ zu suchen, das u möglichst gut approximiert, also

$$\|u - \alpha v\|_V^2 = \|u\|_V^2 - 2\alpha\langle v, u\rangle_V + \alpha^2 \|v\|_V^2$$

minimiert. Das Minimum wird gerade für

$$\alpha := \frac{\langle v, u\rangle_V}{\|v\|_V^2}$$

angenommen, und für diesen Skalierungsfaktor erhalten wir

$$0 \leq \|u - \alpha v\|_V^2 = \|u\|_V^2 - 2\frac{\langle v, u\rangle_V^2}{\|v\|_V^2} + \frac{\langle v, u\rangle_V^2}{\|v\|_V^4}\|v\|_V^2 = \|u\|_V^2 - \frac{\langle v, u\rangle_V^2}{\|v\|_V^2}.$$

Indem wir beide Seiten mit $\|v\|_V^2$ multiplizieren ergibt sich

$$0 \leq \|v\|_V^2 \|u - \alpha v\|_V^2 = \|v\|_V^2 \|u\|_V^2 - \langle v, u\rangle_V^2,$$

und das ist bereits der erste Teil unserer Behauptung.

Falls $|\langle v, u\rangle_V| = \|v\|_V \|u\|_V$ gilt, verschwindet die rechte Seite der obigen Ungleichung, so dass unmittelbar $\|u - \alpha v\|_V^2 = 0$ folgt, also $u = \alpha v$. ∎

Der Raum $\mathbb{R}^n$ ist mit dem euklidischen Skalarprodukt

$$\langle \mathbf{x}, \mathbf{y}\rangle_2 = \sum_{i \in \mathcal{I}} x_i y_i \qquad \text{für alle } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

ein Hilbert-Raum, und die in (3.4) gegebene Cauchy-Schwarz-Ungleichung ergibt sich als Spezialfall aus Lemma 6.12.

Die Variationsaufgabe (6.7) lässt sich mit $V = C_0^1(\Omega)$ in der Form

$$a(v, u) = \beta(v) \qquad \text{für alle } v \in V \tag{6.14}$$

schreiben, wenn wir die Abkürzungen

$$a(v, u) = \int_\Omega \langle \nabla v(x), \nabla u(x)\rangle_2 \, dx \qquad \text{für alle } v, u \in V, \tag{6.15a}$$

$$\beta(v) = \int_\Omega v(x) f(x) \, dx \qquad \text{für alle } v \in V \tag{6.15b}$$

einführen. Es lässt sich leicht nachprüfen, dass das so definierte $a : V \times V \to \mathbb{R}$ eine Bilinearform ist. Wenn wir uns unter einer Bilinearform eine Verallgemeinerung einer Matrix vorstellen, entspricht (6.14) einem linearen Gleichungssystem. Für die Lösbarkeit eines linearen Gleichungssystems ist entscheidend, ob die rechte Seite im Bild der Matrix liegt. Um zu klären, ob das der Fall ist, greifen wir auf den Begriff der Stetigkeit zurück.

Erinnerung 6.13 (Stetige lineare Funktionen) *Eine lineare Funktion $\lambda : V \to \mathbb{R}$ ist genau dann stetig, wenn sie* beschränkt *ist, wenn also eine Zahl $C \in \mathbb{R}_{\geq 0}$ existiert, die*

$$|\lambda(v)| \leq C\|v\|_V \qquad \text{für alle } v \in V \text{ erfüllt.}$$

Definition 6.14 (Funktional) *Eine beschränkte lineare Abbildung $\lambda : V \to \mathbb{R}$ von dem Raum V in den zugehörigen Körper $\mathbb{R}$ nennen wir ein* Funktional.

Den Raum aller Funktionale auf V nennen wir den Dualraum *von V und schreiben ihn als*

$$V' := \{\lambda : V \to \mathbb{R} \ : \ \lambda \text{ linear und beschränkt}\}.$$

Mit der Dualnorm

$$\|\lambda\|_{V'} := \sup\left\{\frac{|\lambda(v)|}{\|v\|_V} \ : \ v \in V \setminus \{0\}\right\} \qquad \text{für alle } \lambda \in V'$$

wird der Dualraum zu einem $\mathbb{R}$-Banach-Raum.

Wenn wir einen geeigneten Raum mit einer geeigneten Norm wählen, wird die rechte Seite β aus (6.14) ein Funktional sein. Wenn die rechte Seite dieser Gleichung stetig in v ist, muss die linke Seite es auch sein, so dass es sich anbietet, auch die Stetigkeit der Bilinearform a zu fordern. Analog zu Erinnerung 6.13 lässt auch sie sich durch eine Form der Beschränktheit charakterisieren.

Definition 6.15 (Stetigkeit) *Eine Bilinearform $a : V \times V \to \mathbb{R}$ nennen wir* stetig, *falls eine Zahl $C_S \in \mathbb{R}_{\geq 0}$ existiert, die*

$$|a(v,u)| \leq C_S\|v\|_V\|u\|_V \qquad \text{für alle } u, v \in V \text{ erfüllt.} \tag{6.16}$$

Ein lineares Gleichungssystem ist eindeutig lösbar, wenn der Kern der Matrix nur den Nullvektor enthält. Das ist beispielsweise sichergestellt, wenn die Matrix positiv definit ist. Im unendlich-dimensionalen Fall genügt es nicht, dass die Bilinearform nur positiv definit ist, sie muss sich von der Null weg beschränken lassen.

Definition 6.16 (Koerzivität) *Eine Bilinearform $a : V \times V \to \mathbb{R}$ nennen wir* koerziv, *falls sie stetig ist und eine Zahl $C_E \in \mathbb{R}_{\geq 0}$ existiert, die*

$$|a(v,v)| \geq C_E\|v\|_V^2 \qquad \text{für alle } v \in V \text{ erfüllt.} \tag{6.17}$$

Stetigkeit und Koerzivität genügen, um die Variationsaufgabe zu lösen:

Satz 6.17 (Lax-Milgram) *Sei V ein $\mathbb{R}$-Hilbertraum. Sei $a : V \times V \to \mathbb{R}$ eine koerzive Bilinearform und sei $\beta \in V'$ ein Funktional. Dann existiert genau ein $u \in V$ mit*

$$a(v,u) = \beta(v) \qquad \text{für alle } v \in V.$$

Dieses u erfüllt die Abschätzung

$$\|u\|_V \leq \frac{1}{C_E}\|\beta\|_{V'}.$$

6.4 Schwache Ableitungen

Leider lässt sich der Satz 6.17 von Lax-Milgram nicht unmittelbar auf unsere Variationsaufgabe (6.14) anwenden, denn der Raum $C_0^1(\Omega)$ ist kein Hilbert-Raum.

Deshalb verallgemeinern wir die Variationsaufgabe ein letztes Mal, um zu einer Formulierung zu gelangen, in der $C_0^1(\Omega)$ durch einen Hilbert-Raum ersetzt wird.

Unser Ausgangspunkt ist ein uns bereits bekannter Hilbert-Raum, nämlich $\mathbb{R}^n$ mit dem euklidischen Skalarprodukt

$$\langle \mathbf{x}, \mathbf{y} \rangle_2 = \sum_{i=1}^{n} x_i y_i \qquad \text{für alle } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Wir ersetzen die Vektoren durch Funktionen und die Summe durch ein geeignetes Integral und erhalten einen Hilbert-Raum, der aus Funktionen besteht.

Erinnerung 6.18 (Quadratintegrable Funktionen) *Eine Abbildung $u : \Omega \to \mathbb{R}^d$, für die das Integral*

$$\int_\Omega \|u(x)\|_2^2 \, dx$$

im Sinn des Lebesgue-Integrals existiert, nennen wir quadratintegrabel.

Quadratintegrable Funktionen bilden den Raum

$$L^2(\Omega, \mathbb{R}^d) := \{f : \Omega \to \mathbb{R}^d \ : \ f \text{ ist quadratintegrabel}\},$$

der mit der Norm

$$\|u\|_{L^2} := \left(\int_\Omega \|u(x)\|_2^2 \, dx \right)^{1/2} \qquad \text{für alle } u \in L^2(\Omega, \mathbb{R}^d) \tag{6.18}$$

ein Banach-Raum ist. Mit dem Skalarprodukt

$$\langle v, u \rangle_{L^2} := \int_\Omega \langle v(x), u(x) \rangle_2 \, dx \qquad \text{für alle } u, v \in L^2(\Omega, \mathbb{R}^d) \tag{6.19}$$

ist er auch ein Hilbert-Raum.

Für $d = 1$ verwenden wir die Abkürzung $L^2(\Omega) = L^2(\Omega, \mathbb{R})$.

Ein Blick auf (6.19) zeigt, dass unsere Gleichung (6.7) sinnvoll bleibt, sofern $\nabla u, \nabla v \in L^2(\Omega, \mathbb{R}^d)$ und $v, f \in L^2(\Omega)$ gelten. Die letztere Bedingung ist bereits erklärt worden, die erste allerdings noch nicht: Was ist der Gradient einer Funktion, die nicht in $C_0^1(\Omega)$ liegt?

Bei der Herleitung der Variationsformulierung (6.7) haben wir bereits auf die partielle Integration zurückgegriffen. Mit ihrer Hilfe können wir auch eine *schwache Ableitung* definieren: Für Funktionen $u \in C^1(\Omega)$ gilt nach (6.6) die Gleichung

$$\int_\Omega \langle \nabla u(x), v(x) \rangle_2 \, dx = -\int_\Omega u(x) \nabla \cdot v(x) \, dx \qquad \text{für alle } v \in C_0^1(\Omega, \mathbb{R}^d).$$

Falls u nicht mehr differenzierbar ist, können wir entsprechend nach einer Funktion $w \in L^2(\Omega, \mathbb{R}^d)$ suchen, die

$$\int_\Omega \langle w(x), v(x) \rangle_2 \, dx = -\int_\Omega u(x) \nabla \cdot v(x) \, dx \qquad \text{für alle } v \in C_0^1(\Omega, \mathbb{R}^d)$$

erfüllt. Für differenzierbares u gilt $w = \nabla u$, anderenfalls ist w eine verallgemeinerte Form des Gradienten.

Definition 6.19 (Schwache Ableitung) *Sei $u \in L^2(\Omega)$. Falls ein $w \in L^2(\Omega, \mathbb{R}^d)$ mit*

$$\int_\Omega \langle w(x), v(x) \rangle_2 \, dx = -\int_\Omega u(x) \nabla \cdot v(x) \, dx \qquad \textit{für alle } v \in C_0^1(\Omega, \mathbb{R}^d) \tag{6.20}$$

existiert, nennen wir die Funktion u schwach differenzierbar *und schreiben $\nabla u := w$ für ihren* schwachen Gradienten.

Den Raum aller schwach differenzierbaren Funktionen auf Ω bezeichnen wir mit

$$H^1(\Omega) := \{u \in L^2(\Omega) \ : \ u \textit{ ist schwach differenzierbar}\}.$$

Diesen Raum nennen wir auch Sobolew-Raum. *Mit der Norm*

$$\|u\|_{H^1} := \sqrt{\|u\|_{L^2}^2 + \|\nabla u\|_{L^2}^2} \qquad \textit{für alle } u \in H^1(\Omega) \tag{6.21}$$

ist er ein Banach-Raum und mit dem Skalarprodukt

$$\langle v, u \rangle_{H^1} := \langle v, u \rangle_{L^2} + \langle \nabla v, \nabla u \rangle_{L^2} \qquad \textit{für alle } u, v \in H^1(\Omega) \tag{6.22}$$

auch ein Hilbert-Raum.

Damit die Variationsformulierung weiterhin sinnvoll ist, brauchen wir auch eine Verallgemeinerung des Raums $C_0^1(\Omega)$, der neben der Differenzierbarkeit auch die Randbedingungen enthält. Hier tritt eine technische Schwierigkeit auf: Für stetige Funktionen $u \in C(\bar{\Omega})$ ist die Einschränkung $u|_{\partial\Omega}$ auf den Rand des Gebiets definiert, bei Funktionen $u \in L^2(\Omega)$ ist das nicht mehr ohne weiteres der Fall. Allerdings kann man immerhin für Funktionen $u \in H^1(\Omega)$ die Einschränkung auf den Rand verallgemeinern. Auf die Details gehen wir nicht weiter ein, wir halten lediglich fest, dass für jede Funktion $u \in H^1(\Omega)$ die Einschränkung $u|_{\partial\Omega}$ sinnvoll definiert ist, und verwenden den Hilbert-Raum

$$H_0^1(\Omega) := \{u \in H^1(\Omega) \ : \ u|_{\partial\Omega} = 0\}$$

als Verallgemeinerung des Raums $C_0^1(\Omega)$.

Wir setzen $V := H_0^1(\Omega)$ und verallgemeinern (6.15), indem wir den Gradienten durch den schwachen Gradienten ersetzen sowie lediglich $f \in L^2(\Omega)$ fordern und so zu

$$a(v, u) = \int_\Omega \langle \nabla v(x), \nabla u(x) \rangle_2 \, dx \qquad \text{für alle } u, v \in V, \tag{6.23}$$

$$\beta(v) = \int_\Omega v(x) f(x) \, dx \qquad \text{für alle } v \in V \tag{6.24}$$

gelangen. Damit können wir die Variationsgleichung (6.7) in die folgende Form bringen:

Gegeben ist $\beta \in V'$, gesucht ist $u \in V$ mit

$$a(v,u) = \beta(v) \qquad \text{für alle } v \in V. \tag{6.25}$$

Das ist genau die Form, auf die sich der Satz 6.17 von Lax-Milgram anwenden ließe, falls seine Voraussetzungen erfüllt sind. Also prüfen wir diese Voraussetzungen nach.

Erstens muss $\beta \in V'$ gelten. Offenbar ist β linear, also bleibt lediglich die Beschränkheit nachzuweisen. Da $f \in L^2(\Omega)$ gilt und dieser Raum ein Hilbert-Raum ist, können wir die Cauchy-Schwarz-Ungleichung (6.13) anwenden, um

$$|\beta(v)| = \left| \int_\Omega v(x) f(x)\, dx \right| = |\langle v, f \rangle_{L^2}| \leq \|v\|_{L^2} \|f\|_{L^2} \qquad \text{für alle } v \in L^2(\Omega)$$

zu erhalten. Mit

$$\|v\|_{L^2} \leq \sqrt{\|v\|_{L^2}^2 + \|\nabla v\|_{L^2}^2} = \|v\|_{H^1} \qquad \text{für alle } v \in H^1(\Omega)$$

folgt daraus

$$|\beta(v)| \leq \|f\|_{L^2} \|v\|_{H^1} \qquad \text{für alle } v \in V,$$

also dürfen wir $\beta \in V'$ mit $\|\beta\|_{V'} \leq \|f\|_{L^2}$ festhalten.

Zweitens muss die Bilinearform a stetig sein. Mit der Cauchy-Schwarz-Ungleichung erhalten wir zunächst

$$\begin{aligned} |a(v,u)| &= \left| \int_\Omega \langle \nabla v(x), \nabla u(x) \rangle_2 \, dx \right| \\ &= |\langle \nabla v, \nabla u \rangle_{L^2}| \leq \|\nabla v\|_{L^2} \|\nabla u\|_{L^2} \qquad \text{für alle } v, u \in V. \end{aligned}$$

Aufgrund der Abschätzung

$$\|\nabla v\|_{L^2} \leq \sqrt{\|v\|_{L^2}^2 + \|\nabla v\|_{L^2}^2} = \|v\|_{H^1} \qquad \text{für alle } v \in V$$

folgt unmittelbar

$$|a(v,u)| \leq \|v\|_V \|u\|_V \qquad \text{für alle } u, v \in V,$$

also ist a eine stetige Bilinearform.

Drittens muss die Bilinearform auch koerziv sein. An diesem Punkt spielen die Randbedingungen eine wichtige Rolle, mit deren Hilfe wir die folgende Hilfsaussage gewinnen können:

Erinnerung 6.20 (Friedrichs-Ungleichung) *Es existiert ein $C_\Omega \in \mathbb{R}_{>0}$ mit*

$$\|v\|_{L^2}^2 \leq C_\Omega \|\nabla v\|_{L^2}^2 \qquad \textit{für alle } v \in H_0^1(\Omega).$$

Mit der Friedrichs-Ungleichung erhalten wir

$$\begin{aligned}\|v\|_{H^1}^2 &= \|v\|_{L^2}^2 + \|\nabla v\|_{L^2}^2 = \int_\Omega v(x)^2\,dx + \int_\Omega \|\nabla v(x)\|_2^2\,dx \\ &\le (C_\Omega + 1)\int_\Omega \|\nabla v(x)\|_2^2\,dx = (C_\Omega+1)\int_\Omega \langle \nabla v(x), \nabla v(x)\rangle_2\,dx \\ &= (C_\Omega+1)a(v,v) \qquad \text{für alle } v \in H_0^1(\Omega), \end{aligned} \tag{6.26}$$

also ist die Bilinearform a koerziv mit $C_E = 1/(C_\Omega + 1)$. Damit lässt sich der Satz 6.17 von Lax-Milgram anwenden und wir haben gezeigt, dass die Variationsaufgabe (6.25) eine eindeutig bestimmte Lösung besitzt, die

$$\|u\|_{H^1} \le (C_\Omega + 1)\|f\|_{L^2}$$

erfüllt. Falls die ursprüngliche Differentialgleichung (6.2) eine Lösung u besitzt, löst diese Funktion u auch die Variationsaufgabe (6.25). Da die Variationsaufgabe *eindeutig* lösbar ist, wird ihre Lösung mit der der ursprünglichen Gleichung übereinstimmen.

6.5 Galerkin-Diskretisierung

Ausgehend von einer Variationsaufgabe können wir auch praktisch durchführbare Näherungsverfahren entwickeln. Die grundlegende Idee der *Galerkin-Diskretisierung* besteht darin, den unendlich-dimensionalen Hilbert-Raum V durch einen endlich-dimensionalen Teilraum $V_h \subseteq V$ zu ersetzen. An die Stelle der Variationsaufgabe (6.25) tritt damit die folgende Aufgabe:

Gegeben ist $\beta \in V'$, gesucht ist $u_h \in V_h$ mit

$$a(v_h, u_h) = \beta(v_h) \qquad \text{für alle } v_h \in V_h. \tag{6.27}$$

Die für den Satz von Lax-Milgram wichtigen Eigenschaften der Bilinearform a und des Funktionals β bleiben auch auf einem Teilraum gültig, so dass auch die diskretisierte Variationsaufgabe (6.27) eindeutig lösbar ist.

Bemerkung 6.21 (Minimierung) *Sei die Bilinearform a symmetrisch und positiv definit. Analog zu Lemma 4.18 können wir die Lösungen der Variationsaufgaben dann auch als Lösungen von Minimierungsaufgaben charakterisieren:*
$u \in V$ löst die Variationsaufgabe (6.25) genau dann, wenn

$$\frac{1}{2}a(u,u) - \beta(u) \le \frac{1}{2}a(v,v) - \beta(v) \qquad \textit{für alle } v \in V \textit{ gilt,}$$

und $u_h \in V_h$ löst die Aufgabe (6.27) genau dann, wenn

$$\frac{1}{2}a(u_h,u_h) - \beta(u_h) \le \frac{1}{2}a(v_h,v_h) - \beta(v_h) \qquad \textit{für alle } v_h \in V_h \textit{ gilt.}$$

Für eine symmetrische Bilinearform bedeutet also die Galerkin-Diskretisierung lediglich, dass das Minimum einer Funktion in dem Teilraum V_h statt in V gesucht wird.

Da die diskretisierte Variationsaufgabe in dem endlich-dimensionalen Raum V_h formuliert ist, können wir sie in eine für den Rechner zugängliche Form überführen: Wir wählen eine Basis $(\varphi_i)_{i\in\mathcal{I}}$ des Raums V_h und stellen u_h und v_h durch Koeffizientenvektoren bezüglich dieser Basis dar, also als

$$u_h = \sum_{j\in\mathcal{I}} x_j\varphi_j, \qquad v_h = \sum_{i\in\mathcal{I}} y_i\varphi_i \tag{6.28}$$

mit $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{\mathcal{I}}$. Durch Einsetzen in (6.27) erhalten wir

$$\sum_{i\in\mathcal{I}} y_i\beta(\varphi_i) = \beta(v_h) = a(v_h, u_h) = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} y_i a(\varphi_i, \varphi_j)x_j.$$

Zur Abkürzung definieren wir den Vektor $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ und die Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ durch

$$b_i = \beta(\varphi_i), \qquad \text{für alle } i \in \mathcal{I}, \tag{6.29a}$$
$$a_{ij} = a(\varphi_i, \varphi_j) \qquad \text{für alle } i, j \in \mathcal{I} \tag{6.29b}$$

und können die Variationsaufgabe (6.27) mit (6.28) kompakt als

$$\langle \mathbf{y}, \mathbf{A}\mathbf{x}\rangle_2 = \langle \mathbf{y}, \mathbf{b}\rangle_2 \qquad \text{für alle } \mathbf{y} \in \mathbb{R}^{\mathcal{I}}$$

schreiben. Damit haben wir ein lineares Gleichungssystem erhalten.

Satz 6.22 (Lineares Gleichungssystem) *Ein Vektor $u_h \in V_h$ löst die diskretisierte Variationsaufgabe (6.27) genau dann, wenn der gemäß (6.28) zugeordnete Vektor $\mathbf{x} \in \mathbb{R}^{\mathcal{I}}$ das lineare Gleichungssystem*

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{6.30}$$

mit der Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und dem Vektor $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ löst, die durch (6.29) gegeben sind.

Beweis. Sei $u_h \in V_h$ eine Lösung der Variationsaufgabe (6.27), sei $i \in \mathcal{I}$. Indem wir in der Variationsaufgabe $v_h = \varphi_i$ einsetzen erhalten wir

$$b_i = \beta(\varphi_i) = a(\varphi_i, u_h) = \sum_{j\in\mathcal{I}} a(\varphi_i, \varphi_j)x_j = \sum_{j\in\mathcal{I}} a_{ij}x_j = (\mathbf{A}\mathbf{x})_i.$$

Also ist $\mathbf{x}$ eine Lösung des Gleichungssystems (6.30).

Sei nun $\mathbf{x}$ eine Lösung dieses Gleichungssystems. Sei $v_h \in V_h$ gegeben, und sei $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ der gemäß (6.28) zugeordnete Vektor. Dann gilt

$$\begin{aligned}\beta(v_h) &= \sum_{i\in\mathcal{I}} y_i\beta(\varphi_i) = \sum_{i\in\mathcal{I}} y_i b_i = \sum_{i\in\mathcal{I}} y_i(\mathbf{A}\mathbf{x})_i \\ &= \sum_{i\in\mathcal{I}} y_i \sum_{j\in\mathcal{I}} a_{ij}x_j = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}} y_i a(\varphi_i, \varphi_j)x_j = a(v_h, u_h).\end{aligned}$$

Damit ist u_h eine Lösung der Variationsaufgabe (6.27). ∎

Statt die Variationsaufgabe zu behandeln können wir also auch einfach ein lineares Gleichungssystem lösen. Dieser Ansatz ist besonders elegant, weil sich nützliche Eigenschaften der Bilinearform a unmittelbar auf die Matrix $\mathbf{A}$ übertragen: Falls a symmetrisch ist, ist auch $\mathbf{A}$ symmetrisch, und falls a positiv definit ist, gilt dasselbe auch für die Matrix $\mathbf{A}$.

In unserem Anwendungsfall ist die in (6.23) definierte Bilinearform sowohl symmetrisch als auch positiv definit, so dass wir beispielsweise das Verfahren der konjugierten Gradienten einsetzen können, um das Gleichungssystem (6.30) zu behandeln.

Natürlich stellt sich die Frage, wie groß der Fehler ist, mit dem wir rechnen müssen, wenn wir die ursprüngliche Variationsaufgabe (6.25) durch die Galerkin-Näherung (6.27) ersetzen. Bei der Beantwortung dieser Frage hilft uns eine besondere Eigenschaft der Galerkin-Diskretisierung, die *Galerkin-Orthogonalität*:

Lemma 6.23 (Galerkin-Orthogonalität) *Seien $u \in V$ und $u_h \in V_h$ die Lösungen der Variationsaufgaben (6.25) und (6.27). Dann gilt*

$$a(v_h, u - u_h) = 0 \qquad \text{für alle } v_h \in V_h. \tag{6.31}$$

Falls a ein Skalarprodukt ist, bedeutet diese Gleichung gerade, dass der Diskretisierungsfehler $u - u_h$ bezüglich dieses Skalarprodukts senkrecht auf dem gesamten Raum V_h steht.

Beweis. Sei $v_h \in V_h$. Da $V_h \subseteq V$ gilt, können wir v_h auch in (6.25) einsetzen, um

$$a(v_h, u) = \beta(v_h)$$

zu erhalten. Indem wir von dieser Gleichung (6.27) subtrahieren ergibt sich

$$a(v_h, u - u_h) = a(v_h, u) - a(v_h, u_h) = \beta(v_h) - \beta(v_h) = 0.$$

∎

Aus der Galerkin-Orthogonalität lässt sich eine Aussage über den Diskretisierungsfehler gewinnen.

Satz 6.24 (Céa) *Die Bilinearform $a : V \times V \to \mathbb{R}$ sei koerziv. Seien $u \in V$ und $u_h \in V_h$ die Lösungen der Variationsaufgaben (6.25) und (6.27). Dann gilt*

$$\|u - u_h\|_V \leq \frac{C_S}{C_E} \|u - w_h\|_V \qquad \text{für alle } w_h \in V_h,$$

die Galerkin-Approximation u_h ist also „fast so gut“ wie die bestmögliche Approximation von u im Teilraum V_h.

Beweis. Sei $w_h \in V_h$. Mit der Koerzivität (6.17) der Bilinearform und der Galerkin-Orthogonalität (6.31), angewendet auf $v_h := u_h - w_h$, erhalten wir

$$C_E \|u - u_h\|_V^2 \leq a(u - u_h, u - u_h) = a(u - u_h, u - u_h) + a(u - u_h, u_h - w_h)$$

$$= a(u - u_h, u - u_h + u_h - w_h) = a(u - u_h, u - w_h).$$

Wir setzen die Stetigkeit (6.16) ein und gelangen zu

$$C_E \|u - u_h\|_V^2 \leq a(u - u_h, u - w_h) \leq C_S \|u - u_h\|_V \|u - w_h\|_V,$$

und aus dieser Abschätzung folgt unmittelbar

$$C_E \|u - u_h\|_V \leq C_S \|u - w_h\|_V.$$

Per Division durch C_E ergibt sich die Behauptung. ∎

Bemerkung 6.25 (Energienorm) *Falls die Bilinearform a nicht nur koerziv, sondern auch symmetrisch ist, ist a ein Skalarprodukt auf V und wir können die durch*

$$\|v\|_a := \sqrt{a(v,v)} \qquad \text{für alle } v \in V$$

definierte Energienorm *als alternative Norm auf V verwenden.*

Infolge von Koerzivität und Stetigkeit gilt

$$C_E \|v\|_V^2 \leq a(v,v) = \|v\|_a^2 = a(v,v) \leq C_S \|v\|_V^2 \qquad \text{für alle } v \in V,$$

also sind die Energienorm und die Hilbert-Raum-Norm $\|\cdot\|_V$ äquivalent, so dass V auch mit der Energienorm und a als Skalarprodukt ein Hilbert-Raum ist.

Mit Hilfe der Galerkin-Orthogonalität (6.31) und der Cauchy-Schwarz-Ungleichung (6.13), angewendet auf das Skalaprodukt a, erhalten wir für alle $w_h \in V_h$ die Ungleichung

$$\begin{aligned} \|u - u_h\|_a^2 &= a(u - u_h, u - u_h) = a(u - u_h, u - u_h) + a(u - u_h, u_h - w_h) \\ &= a(u - u_h, u - w_h) \leq \|u - u_h\|_a \|u - w_h\|_a, \end{aligned}$$

aus der unmittelbar

$$\|u - u_h\|_a \leq \|u - w_h\|_a \qquad \text{für alle } w_h \in V_h$$

folgt. Bezüglich der Energienorm ist also u_h tatsächlich die bestmögliche Approximation der Lösung u im Teilraum V_h.

6.6 Finite-Elemente-Basis

Der Galerkin-Ansatz lässt sich mit fast beliebigen Teilräumen $V_h \subseteq V$ anwenden, also stellt sich die Frage, welche Teilräume besonders gut geeignet sind. Drei Aspekte sind dabei von besonderer Bedeutung:

- Der Raum V_h muss ein Teilraum des Raums V sein. In unserem Fall bedeutet das, dass die Elemente des Raums schwach differenzierbar sein müssen.

- Der Raum V_h muss so gewählt sein, dass sich u durch ein $w_h \in V_h$ gut approximieren lässt. Nach dem Satz 6.24 von Céa ist dann auch u_h eine gute Näherung der Lösung u.
- Wir müssen eine Basis des Raums finden können, mit der die Matrix $\mathbf{A}$ und der Vektor $\mathbf{b}$ aus (6.29) sich effizient verarbeiten lassen.

Um einen Raum schwach differenzierbarer Funktionen auf dem Gebiet Ω zu konstruieren, bietet es sich an, von besonders einfachen Funktionen auszugehen, nämlich von *Polynomen*. Polynome sind sogar unendlich oft differenzierbar, sie lassen sich einfach auswerten und auch ansonsten gut verarbeiten. Eine Beziehung zu einer Triangulation $\mathcal{T}$ des Gebiets Ω lässt sich besonders einfach herstellen, indem wir fordern, dass eine Funktion $v_h \in V_h$ auf jedem Simplex ω_t mit $t \in \mathcal{T}$ ein Polynom sein muss. Um Schwierigkeiten am Rand der Simplizes zu vermeiden definieren wir das *Innere* eines Simplex' t durch

$$\mathring{\omega}_t := \left\{ \sum_{x \in t} \alpha_x x \ : \ \alpha_x \in (0,1) \text{ für alle } x \in t, \ \sum_{x \in t} \alpha_x < 1 \right\}.$$

Die Ränder des Simplex' haben keinen Einfluss auf den Wert des Lebesgue-Integrals, so dass wir ein Integral über Ω als die Summe der Integrale über $\mathring{\omega}_t$ mit $t \in \mathcal{T}$ darstellen können.

Definition 6.26 (Stückweise Polynome) *Sei $m \in \mathbb{N}_0$, sei $\mathcal{T}$ eine Triangulation eines Gebiets Ω. Dann nennen wir*

$$\Pi_{\mathcal{T},m} := \{u : \Omega \to \mathbb{R} \ : \ u|_{\mathring{\omega}_t} \text{ ist ein Polynom höchstens } m\text{-ten Grades für alle } t \in \mathcal{T}\}$$

den Raum der stückweisen Polynome m-ten Grades.

Elemente dieses Raums sind zumindest auf allen Simplizes der Triangulation sogar „stark“ differenzierbar, wir müssen nur noch prüfen, ob sie insgesamt noch schwach differenzierbar sind.

Satz 6.27 (Schwache Differenzierbarkeit) *Eine Funktion $u \in \Pi_{\mathcal{T},m}$ ist genau dann schwach differenzierbar, wenn sie stetig ist.*

Beweis. Wir werden lediglich beweisen, dass aus der Stetigkeit der Funktion u bereits die schwache Differenzierbarkeit folgt. Für die umgekehrte Implikation sind theoretische Hilfsmittel erforderlich, auf die wir an dieser Stelle nicht näher eingehen können.

Sei also $u \in \Pi_{\mathcal{T},m}$ stetig. Wir müssen nachprüfen, dass die Bedingung (6.20) gilt. Dazu werden wir das Integral über Ω in Integrale über die Simplizes ω_t mit $t \in \mathcal{T}$ zerlegen. Nach Definition finden wir Polynome u_t m-ter Ordnung so, dass

$$u|_{\mathring{\omega}_t} = u_t|_{\mathring{\omega}_t} \qquad \text{für alle } t \in \mathcal{T} \tag{6.32}$$

gilt. Insbesondere können wir auf den Teilgebieten u_t statt $u|_{\mathring{\omega}_t}$ integrieren.

Damit können wir auf jedem einzelnen Simplex ω_t die partielle Integration (6.6) verwenden, um eine Ableitung zu berechnen. Um die dabei auftretenden Randintegrale behandeln zu können führen wir die Mengen

$$\mathcal{F}_t := \{s \in \mathcal{S}_{d-1} \ : \ s \subseteq t\} \qquad \text{für alle } t \in \mathcal{T}$$

der *Seiten* eines Simplex' $t \in \mathcal{T}$ ein. Alle Seiten einer Triangulation bezeichnen wir mit

$$\mathcal{F}_\mathcal{T} := \bigcup_{t \in \mathcal{T}} \mathcal{F}_t.$$

Für jede Seite $s \in \mathcal{F}_\mathcal{T}$ fixieren wir einen Einheitsnormalenvektor $n_s \in \mathbb{R}^d$. Da in der Gleichung (6.6) *äußere* Normalenvektoren benötigt werden, führen wir

$$\sigma(t,s) := \begin{cases} 1 & \text{falls } n_s \text{ äußerer Normalenvektor für } t \text{ ist,} \\ -1 & \text{ansonsten} \end{cases} \qquad \text{für alle } t \in \mathcal{T},\ s \in \mathcal{F}_t$$

ein und halten fest, das $\sigma(t,s)n_s$ dann immer ein äußerer Normalenvektor auf der Seite s des Simplex' t ist.

Für jede Seite $s \in \mathcal{F}_\mathcal{T}$ ist die Menge

$$\mathcal{T}_s := \{t \in \mathcal{T} \ : \ s \in \mathcal{F}_t\}$$

der an s angrenzenden Simplizes nach Definition nicht leer. Falls für $t_1, t_2 \in \mathcal{T}_s$ die Gleichung $\sigma(t_1,s) = \sigma(t_2,s)$ gilt, folgt aus der Definition der Triangulation bereits $t_1 = t_2$. Da σ nur zwei unterschiedliche Werte annehmen kann, kann $\mathcal{T}_s$ höchstens zwei Simplizes $t_1, t_2 \in \mathcal{T}$ enthalten, und für diese Simplizes muss $\sigma(t_1,s) = -\sigma(t_2,s)$ gelten.

Nach diesen Vorarbeiten können wir uns dem Beweis des Satzes zuwenden. Seien $u \in \Pi_{\mathcal{T},m}$ und $v \in C_0^1(\Omega, \mathbb{R}^d)$ gegeben. Mit (6.32) und partieller Integration gemäß (6.6) erhalten wir

$$\begin{aligned} -\int_\Omega u(x)\nabla \cdot v(x)\,dx &= \sum_{t\in\mathcal{T}} -\int_{\omega_t} u(x)\nabla\cdot v(x)\,dx = \sum_{t\in\mathcal{T}} -\int_{\omega_t} u_t(x)\nabla\cdot v(x)\,dx \\ &= \sum_{t\in\mathcal{T}} \int_{\omega_t} \langle \nabla u_t(x), v(x)\rangle_2\,dx - \int_{\partial\omega_t} u_t(x)\langle n(x), v(x)\rangle_2\,dx \\ &= \sum_{t\in\mathcal{T}} \int_{\omega_t} \langle \nabla u_t(x), v(x)\rangle_2\,dx - \sum_{s\in\mathcal{F}_t} \int_{\omega_s} u_t(x)\langle \sigma(t,s)n_s(x), v(x)\rangle_2\,dx. \end{aligned}$$

Da erste Integral stellt für uns kein Problem dar, denn wir können den schwachen Gradienten der Funktion u einfach aus den Gradienten ∇u_t zusammensetzen. Für das zweite Integral dagegen ist in (6.20) kein Platz, wir müssen also nachprüfen, dass es verschwindet. Es gilt

$$\sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{F}_t} \int_{\omega_s} u_t(x)\langle \sigma(t,s)n_s(x), v(x)\rangle_2\,dx$$

$$= \sum_{s\in\mathcal{F}_\mathcal{T}} \sum_{t\in\mathcal{T}_s} \int_{\omega_s} u_t(x)\langle\sigma(t,s)n_s(x), v(x)\rangle_2 \, dx.$$

Wir untersuchen die Summe

$$\sum_{t\in\mathcal{T}_s} \int_{\omega_s} u_t(x)\langle\sigma(t,s)n_s(x), v(x)\rangle_2 \, dx$$

für eine Seite $s \in \mathcal{F}_\mathcal{T}$. Wir haben bereits gesehen, dass $\mathcal{T}_s$ nur entweder einen Simplex oder zwei Simplizes enthalten kann.

Im ersten Fall liegt s auf dem Rand des Gebiets Ω, so dass die Funktion v auf ω_s gleich null ist, also auch das Integral.

Im zweiten Fall finden wir $t_1, t_2 \in \mathcal{T}$ mit $\mathcal{T}_s = \{t_1, t_2\}$, $\sigma(t_1, s) = 1$ und $\sigma(t_2, s) = -1$, so dass wir

$$\begin{aligned}
&\sum_{t\in\mathcal{T}_s} \int_{\omega_s} u(x)\langle\sigma(t,s)n_s(x), v(x)\rangle_2 \, dx \\
&\quad= \int_{\omega_s} u_{t_1}(x)\langle\sigma(t_1,s)n_s(x), v(x)\rangle_2 \, dx + \int_{\omega_s} u_{t_2}(x)\langle\sigma(t_2,s)n_s(x), v(x)\rangle_2 \, dx \\
&\quad= \int_{\omega_s} u_{t_1}(x)\langle n_s(x), v(x)\rangle_2 \, dx - \int_{\omega_s} u_{t_2}(x)\langle n_s(x), v(x)\rangle_2 \, dx \\
&\quad= \int_{\omega_s} (u_{t_1}(x) - u_{t_2}(x))\langle n_s(x), v(x)\rangle_2 \, dx
\end{aligned}$$

erhalten. Da u stetig ist, gilt $u_{t_1}|_{\omega_s} = u_{t_2}|_{\omega_s}$, also ist jedes dieser Integrale gleich null.

Damit verschwinden alle Randintegrale, wir haben

$$-\int_\Omega u(x)\nabla\cdot v(x)\,dx = \sum_{t\in\mathcal{T}} \int_{\omega_t} \langle\nabla u_t(x), v(x)\rangle_2 \, dx$$

bewiesen. Indem wir

$$w(x) := \begin{cases} \nabla u_t(x) & \text{falls } x \in \mathring{\omega}_t \text{ für ein } t \in \mathcal{T}, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } x \in \Omega$$

setzen, erhalten wir

$$-\int_\Omega u(x)\nabla\cdot v(x)\,dx = \sum_{t\in\mathcal{T}} \int_{\omega_t} \langle w(x), v(x)\rangle_2 \, dx,$$

also gilt (6.20) und w ist der schwache Gradient der Funktion u. ∎

Wenn wir also den Raum V_h aus stückweisen Polynomen zusammensetzen wollen, brauchen wir lediglich sicher zu stellen, dass die Polynome an den Seiten der Simplizes stetig ineinander übergehen.

Definition 6.28 (Stetige stückweise Polynome) *Sei $m \in \mathbb{N}_0$, sei $\mathcal{T}$ eine Triangulation eines Gebiets Ω. Dann nennen wir*

$$\mathcal{P}_{\mathcal{T},m} := \{u \in C(\Omega) \ : \ u|_{\mathring{\omega}_t} \text{ ist ein Polynom höchstens } m\text{-ten Grades für alle } t \in \mathcal{T}\}$$

den Raum der stetigen stückweisen Polynome m-ten Grades*. Falls sich die Triangulation aus dem Kontext ergibt, verwendet man häufig $\mathcal{P}_m$ als Abkürzung für $\mathcal{P}_{\mathcal{T},m}$.*

*Nach Satz 6.27 gilt $\mathcal{P}_{\mathcal{T},m} \subseteq H^1(\Omega)$, man bezeichnet die Räume als H^1-*konform*.*

Man kann sich relativ leicht überlegen, dass durch die Stetigkeitsbedingung der Raum $\mathcal{P}_{\mathcal{T},0}$ lediglich auf dem gesamten Gebiet konstante Funktionen enthält, also für die Approximation allgemeiner Lösungen eher unbrauchbar ist.

Der Raum $\mathcal{P}_{\mathcal{T},1}$ hingegen ist für unsere Zwecke reichhaltig genug und soll deswegen im Folgenden näher untersucht werden. Insbesondere sind wir daran interessiert, eine Basis dieses Raums zu finden, die dafür sorgt, dass die Matrix $\mathbf{A}$ eine Struktur aufweist, die für die Implementierung nützlich ist.

Besonders günstig wäre eine Basis, die dazu führt, dass möglichst viele Einträge der Matrix $\mathbf{A}$ gleich null sind und deshalb nicht berechnet und auch nicht abgespeichert werden müssen. Dazu bietet es sich an, die *Träger* der Basisfunktionen zu untersuchen, die durch

$$\operatorname{supp} \varphi_i := \overline{\{x \in \Omega \ : \ \varphi_i(x) \neq 0\}} \qquad \text{für alle } i \in \mathcal{I}$$

gegeben sind. Außerhalb ihres Trägers verschwindet die Basisfunktion φ_i, so dass sich (6.29) in der Form

$$a_{ij} = \int_{\operatorname{supp}\varphi_i \cap \operatorname{supp}\varphi_j} \langle \nabla\varphi_i(x), \nabla\varphi_j(x)\rangle_2 \, dx \qquad \text{für alle } i,j \in \mathcal{I}$$

schreiben lässt, denn zu dem Integral tragen nur diejenigen Punkte des Gebiets etwas bei, in denen der Integrand ungleich null ist. Falls die Träger zweier Basisfunktionen φ_i und φ_j disjunkt sind oder ihr Schnitt kein Volumen (für $d = 3$) oder keine Fläche (für $d = 2$) aufweist, folgt somit $a_{ij} = 0$, so dass wir diesen Koeffizienten weder ausrechnen noch abspeichern müssen.

Der Raum der linearen Polynome wird auf d-dimensionalen Gebieten durch

$$\{1, x_1, x_2, \ldots, x_d\}$$

aufgespannt, also gerade durch $d+1$ linear unabhängige Funktionen. Es lässt sich zeigen, dass daraus bereits folgt, dass eine Funktion $u_h \in \mathcal{P}_{\mathcal{T},1}$ auf jedem ω_t mit $t \in \mathcal{T}$ bereits durch ihre Werte in den $d+1$ Eckpunkten t eindeutig festgelegt ist.

Definition 6.29 (Knotenmenge) *Die Menge der Eckpunkte* aller *Simplizes einer Triangulation nennen wir deren* Knotenmenge *und bezeichnen sie mit*

$$\mathcal{N}_{\mathcal{T}} := \bigcup_{t \in \mathcal{T}} t.$$

Die Punkte $i \in \mathcal{N}_{\mathcal{T}}$ nennen wir Knotenpunkte *der Triangulation.*

Damit ist eine Funktion $u_h \in \mathcal{P}_{\mathcal{T},1}$ durch ihre Werte in den Knoten $\mathcal{N}_\mathcal{T}$ definiert. Für die Konstruktion einer Basis bietet es sich an, diese Werte so zu wählen, dass möglichst kleine Träger entstehen. Aus unserer Betrachtung folgt, dass die Funktion u_h auf ω_t gerade dann gleich null ist, wenn sie in den Punkten t gleich null ist. Am einfachsten ist es, eine Basisfunktion so zu wählen, dass sie in genau einem Knotenpunkt ungleich null ist, beispielsweise gleich eins, und in allen anderen gleich null. Dann ist ihr Träger so klein wie möglich.

Um praktisch mit diesen *Knotenbasisfunktionen* arbeiten zu können, empfiehlt es sich, sie explizit für jeden Simplex auszudrücken. Diese Aufgabe lässt sich besonders elegant lösen, indem wir auf die aus Erinnerung 4.1 bekannte Determinante zurückgreifen.

Lemma 6.30 (Lokale Basisfunktion) *Sei $t \in \mathcal{T}$, sei $i \in t$.*

Falls $d = 2$ gilt, finden wir $j, k \in t$ mit $t = \{i, j, k\}$ und definieren

$$\varphi_{i,t}(x) := \frac{\det(x - j, k - j)}{\det(i - j, k - j)} \qquad \text{für alle } x \in \mathbb{R}^2.$$

Dann folgen

$$\varphi_{i,t}(i) = 1, \qquad \varphi_{i,t}(j) = 0, \qquad \varphi_{i,t}(k) = 0. \tag{6.33}$$

Falls $d = 3$ gilt, finden wir $j, k, \ell \in t$ mit $t = \{i, j, k, \ell\}$ und definieren

$$\varphi_{i,t}(x) := \frac{\det(x - j, k - j, \ell - j)}{\det(i - j, k - j, \ell - j)} \qquad \text{für alle } x \in \mathbb{R}^3.$$

Dann folgen

$$\varphi_{i,t}(i) = 1, \qquad \varphi_{i,t}(j) = 0, \qquad \varphi_{i,t}(k) = 0, \qquad \varphi_{i,t}(\ell) = 0. \tag{6.34}$$

Da $\varphi_{i,t}$ jeweils ein lineares Polynom ist, ist es durch die Werte in den Eckpunkten eindeutig festgelegt, also insbesondere unabhängig von der Reihenfolge, in der j, k, ℓ gewählt werden.

Beweis. Wir nutzen aus, dass die Determinante verschwindet, falls ihre Argumente linear abhängig sind. Für $d = 2$ erhalten wir damit

$$\varphi_{i,t}(j) = \frac{\det(j - j, k - j)}{\det(i - j, k - j)} = \frac{\det(0, k - j)}{\det(i - j, k - j)} = 0,$$
$$\varphi_{i,t}(k) = \frac{\det(k - j, k - j)}{\det(i - j, k - j)} = 0.$$

Da t regulär ist, sind $i - j$ und $k - j$ linear unabängig, so dass wir $\det(i - j, k - j) \neq 0$ und damit $\varphi_{i,t}(i) = 1$ erhalten.

Für $d = 3$ können wir entsprechend verfahren. ∎

Bemerkung 6.31 (Basis) *Aus Lemma 6.30 folgt insbesondere, dass eine stückweise lineare Funktion durch ihre Werte in den Knotenpunkten eindeutig festgelegt ist: Beispielsweise für $d = 2$ haben wir auf einem regulären Dreieck ω_t mit $t = \{i, j, k\}$ drei Funktionen $\varphi_{i,t}$, $\varphi_{j,t}$ und $\varphi_{k,t}$. Diese Funktionen sind linear unabhängig, denn falls uns $\alpha_i, \alpha_j, \alpha_k \in \mathbb{R}$ mit*

$$\alpha_i \varphi_{i,t} + \alpha_j \varphi_{j,t} + \alpha_k \varphi_{k,t} = 0$$

gegeben sind, folgt durch Einsetzen von i, j und k in (6.33) jeweils

$$\begin{aligned}
\alpha_i &= \alpha_i \varphi_{i,t}(i) + \alpha_j \varphi_{j,t}(i) + \alpha_k \varphi_{k,t}(i) = 0,\\
\alpha_j &= \alpha_i \varphi_{i,t}(j) + \alpha_j \varphi_{j,t}(j) + \alpha_k \varphi_{k,t}(j) = 0,\\
\alpha_k &= \alpha_i \varphi_{i,t}(k) + \alpha_j \varphi_{j,t}(k) + \alpha_k \varphi_{k,t}(k) = 0.
\end{aligned}$$

Also ist $\{\varphi_{i,t}, \varphi_{j,t}, \varphi_{k,t}\}$ eine Basis des Raums der linearen Polynome, und die Koeffizienten eines beliebigen linearen Polynoms sind durch dessen Werte in den Eckpunkten i, j und k gegeben.

Entsprechendes gilt auch für $d = 3$ und reguläre Tetraeder.

Aus den lokalen Knotenbasisfunktionen können wir globale Knotenbasisfunktionen zusammensetzen, indem wir die Funktion auf jedem Simplex ihres Trägers definieren.

Definition 6.32 (Knotenbasis) *Für jeden Knoten $i \in \mathcal{N}_\mathcal{T}$ definieren wir die* Knotenbasisfunktion *durch*

$$\varphi_i(x) := \begin{cases} \varphi_{i,t}(x) & \text{falls } x \in \omega_t,\ i \in t, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } x \in \bar{\Omega}.$$

Die Funktion ist wohldefiniert, denn falls $x \in \omega_t$ und $x \in \omega_s$ für zwei $t, s \in \mathcal{T}$ gilt, folgt aus der Linearität von $\varphi_{i,t}$ und $\varphi_{i,s}$, dass beide Funktionen auf dem Simplex $\omega_t \cap \omega_s = \omega_{t \cap s}$ (nach Definition 6.2) identisch sind, dass also $\varphi_{i,t}(x) = \varphi_{i,s}(x)$ gilt.

Die Menge $(\varphi_i)_{i \in \mathcal{N}_\mathcal{T}}$ nennen wir die Knotenbasis *des Raums $\mathcal{P}_{\mathcal{T},1}$.*

Der Raum $\mathcal{P}_{\mathcal{T},1}$ ist zwar ein Teilraum des Sobolew-Raums $H^1(\Omega)$, erfüllt jedoch nicht die Randbedingung, die wir für den Raum $H^1_0(\Omega)$ zusätzlich aufgenommen haben. Diese Bedingung lässt sich allerdings einfach umsetzen: Wir schließen alle Randpunkte aus der Knotenmenge $\mathcal{N}_\mathcal{T}$ aus und stellen so sicher, dass von den verbleibenden Knotenbasisfunktionen aufgespannte Funktionen auf dem Rand verschwinden.

Definition 6.33 (Ansatzraum) *Wir definieren die Menge der* inneren Knoten *der Triangulation durch*

$$\mathcal{I} := \{i \in \mathcal{N}_\mathcal{T} \ : \ i \in \Omega\}$$

und setzen

$$V_h := \left\{ \sum_{i \in \mathcal{I}} x_i \varphi_i \ : \ \mathbf{x} \in \mathbb{R}^\mathcal{I} \right\}.$$

Dann gilt $V_h \subseteq H^1_0(\Omega)$.

6.7 Aufstellen des Gleichungssystems

Da wir nun über einen passenden Ansatzraum und eine geeignete Basis desselben verfügen, besteht unsere nächste Aufgabe darin, die Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und den Vektor $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ gemäß (6.29) zu konstruieren.

Nach Definition 6.2 setzt sich das Gebiet Ω aus den Simplizes ω_t der Triangulation zusammen, so dass wir die in (6.29) auftretenden Integrale in Integrale über die Simplizes zerlegen können. Da die Knotenbasisfunktion φ_i auf allen Simplizes $t \in \mathcal{T}$ mit $i \notin t$ verschwindet, ergibt sich

$$\begin{aligned} a_{ij} = a(\varphi_i, \varphi_j) &= \int_\Omega \langle \nabla\varphi_i(x), \nabla\varphi_j(x)\rangle_2 \, dx \\ &= \sum_{t\in\mathcal{T}} \int_{\omega_t} \langle \nabla\varphi_i(x), \nabla\varphi_j(x)\rangle_2 \, dx = \sum_{\substack{t\in\mathcal{T}\\ i\in t, j\in t}} \int_{\omega_t} \langle \nabla\varphi_i(x), \nabla\varphi_j(x)\rangle_2 \, dx \\ &= \sum_{\substack{t\in\mathcal{T}\\ i\in t, j\in t}} \int_{\omega_t} \langle \nabla\varphi_{i,t}(x), \nabla\varphi_{j,t}(x)\rangle_2 \, dx \qquad \text{für alle } i,j \in \mathcal{I}. \end{aligned}$$

Entsprechend erhalten wir

$$\begin{aligned} b_i = \beta(\varphi_i) &= \int_\Omega \varphi_i(x) f(x) \, dx \\ &= \sum_{t\in\mathcal{T}} \int_{\omega_t} \varphi_i(x) f(x) \, dx = \sum_{\substack{t\in\mathcal{T}\\ i\in t}} \int_{\omega_t} \varphi_i(x) f(x) \, dx \\ &= \sum_{\substack{t\in\mathcal{T}\\ i\in t}} \int_{\omega_t} \varphi_{i,t}(x) f(x) \, dx \qquad \text{für alle } i \in \mathcal{I}. \end{aligned}$$

Wir können also das Gleichungssystem aufstellen, indem wir lediglich auf den einzelnen Elementen der Triangulation rechnen.

Übungsaufgabe 6.34 (Gradienten) *Beweisen Sie unter den Voraussetzungen des Lemmas 6.30, dass der Gradient der lokalen Basisfunktion für $d = 2$ durch*

$$\nabla\varphi_{i,t}(x) = \frac{1}{\det(i-j, k-j)} \begin{pmatrix} k_2 - j_2 \\ j_1 - k_1 \end{pmatrix} \qquad \textit{für alle } x \in \omega_t$$

und für $d = 3$ durch

$$\begin{aligned} \nabla\varphi_{i,t}(x) &= \frac{1}{\det(i-j, k-j, \ell-j)} \begin{pmatrix} (k_2 - j_2)(\ell_3 - j_3) - (k_3 - j_3)(\ell_2 - j_2) \\ (k_3 - j_3)(\ell_1 - j_1) - (k_1 - j_1)(\ell_3 - j_3) \\ (k_1 - j_1)(\ell_2 - j_2) - (k_2 - j_2)(\ell_1 - j_1) \end{pmatrix} \\ &= \frac{(k-j) \times (\ell-j)}{\det(i-j, k-j, \ell-j)} \qquad \textit{für alle } x \in \omega_t \end{aligned}$$

gegeben ist.

Die Tatsache, dass der Gradient auf jedem Simplex konstant ist, erleichtert uns das Aufstellen der Matrix $\mathbf{A}$ erheblich: Für den Beitrag des Simplex ω_t zu a_{ij} brauchen wir nur das Skalarprodukt der Gradienten von $\varphi_{i,t}$ und $\varphi_{j,t}$ zu berechnen und mit der Fläche beziehungsweise dem Volumen des Simplex zu multiplizieren.

Diese Fläche beziehungsweise dieses Volumen lässt sich einfach aus der Determinante gewinnen: Für $d = 2$ beträgt die Fläche eines Dreiecks ω_t mit Eckpunkten $t = \{i, j, k\}$ gerade

$$\frac{1}{2}|\det(i - j, k - j)|,$$

für $d = 3$ beträgt das Volumen eines Tetraeders ω_t mit Eckpunkten $t = \{i, j, k, \ell\}$ gerade

$$\frac{1}{6}|\det(i - j, k - j, \ell - j)|.$$

Dieselben Determinanten treten auch bei der Bestimmung der Gradienten auf, so dass wir sie nur einmal zu berechnen brauchen.

Übungsaufgabe 6.35 (Determinanten) *In der Praxis benötigen wir häufig lokale Basisfunktionen für alle Eckpunkte eines Simplex'. In dieser Situation können wir die mehrfache Berechnung der Determinante vermeiden, indem wir für $d = 2$ die Gleichung*

$$\det(j - k, i - k) = \det(i - j, k - j)$$

und für $d = 3$ die Gleichung

$$\det(j - k, \ell - k, i - k) = -\det(i - j, k - j, \ell - j)$$

verwenden. Wenn wir die Eckpunkte zyklisch durchlaufen, können wir diese Gleichungen wiederholt anwenden und brauchen die aufwendige Berechnung der Determinante nur einmal pro Simplex durchzuführen.

Bedauerlicherweise ist die rechte Seite f in den meisten Anwendungsfällen nicht konstant, so dass wir die Integrale

$$\int_{\omega_t} \varphi_{i,t}(x) f(x)\, dx$$

berechnen oder wenigstens approximieren müssen. Integrale werden in der Praxis häufig durch Quadraturformeln (vgl. Lemma 5.1) approximiert, die lediglich die Auswertung der zu integrierenden Funktion in einigen Punkten benötigen. Entsprechend können wir auch bei Simplizes vorgehen.

Definition 6.36 (Kantenmittelpunktregel) *Sei ω_t ein regulärer Simplex.*

Falls $d = 2$ gilt, können wir Integrale über ω_t mit $t = \{i, j, k\}$ durch

$$\int_{\omega_t} g(x)\, dx \approx \frac{|\det(i - j, k - j)|}{6} \left(g((i + j)/2) + g((j + k)/2) + g((k + i)/2)\right)$$

approximieren. Falls $d = 3$ gilt, erhalten wir für $t = \{i, j, k, \ell\}$ die Näherung

$$\int_{\omega_t} g(x)\,dx \approx \frac{|\det(i-j, k-j, \ell-j)|}{36} \left(g((i+j)/2) + g((j+k)/2) + g((k+i)/2) + g((i+\ell)/2) + g((j+\ell)/2) + g((k+\ell)/2)\right).$$

In beiden Fällen verwenden wir also die Werte der Funktion g in den Kantenmittelpunkten und wählen den Skalierungsfaktor so, dass wir für $g = 1$ die Fläche beziehungsweise das Volumen des Simplex' erhalten.

Da unsere Knotenbasisfunktionen auf den Kanten nur die Werte $1/2$ oder null annehmen können, vereinfacht sich die Berechnung der Integrale noch etwas weiter: Für $d = 2$ beispielsweise erhalten wir mit $t = \{i, j, k\}$ die Näherung

$$\int_{\omega_t} \varphi_{i,t}(x) f(x)\,dx \approx \frac{|\det(i-j, k-j)|}{12} \left(f((i+j)/2) + f((i+k)/2)\right),$$

da $\varphi_{i,t}((j+k)/2) = 0$ gilt. Die Kantenmittelpunktregel berechnet eine relativ gute Näherung des Integrals: Während die in (5.1) eingeführte Mittelpunktregel lediglich lineare Polynome exakt integrieren konnte, gelingt das der Kantenmittelpunktregel auch noch für quadratische.

Übungsaufgabe 6.37 (Kantenmittelpunktregel) *Sei $d = 2$, seien*

$$i = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad j = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad k = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Dann nimmt die Kantenmittelpunktregel die Gestalt

$$\int_{\omega_t} g(x)\,dx \approx \frac{1}{6}\left(g\begin{pmatrix} 1/2 \\ 0 \end{pmatrix} + g\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + g\begin{pmatrix} 0 \\ 1/2 \end{pmatrix}\right)$$

an. Beweisen Sie, dass für

$$g \in \{\varphi_{i,t}, \varphi_{j,t}, \varphi_{k,t}, \varphi_{i,t}\varphi_{j,t}, \varphi_{j,t}\varphi_{k,t}, \varphi_{i,t}\varphi_{k,t}\}$$

beide Seiten identisch sind, die Quadraturformel also dem exakten Integral entspricht.

Folgern Sie daraus, dass die Kantenmittelpunktregel für alle quadratischen Polynome das exakte Ergebnis berechnet.

Bei den Rechenoperationen, die wir für das Aufstellen der Matrix und des Vektors benötigen, treten viele Größen auf, die von dem Simplex abhängen, auf dem wir gerade arbeiten, beispielsweise Determinanten, Gradienten oder Werte der Funktion f in den Kantenmittelpunkten.

Wenn wir beispielsweise die Berechnung der Matrix $\mathbf{A}$ so gestalten würden, dass wir in einer äußeren Schleife über alle $i, j \in \mathcal{I}$ laufen und dann in einer inneren Schleife

über alle $t \in \mathcal{T}$ mit $i, j \in t$, würden die auf dem Simplex beruhenden Größen mehrfach berechnet werden.

Wir können den Rechenaufwand erheblich reduzieren, indem wir stattdessen in einer äußeren Schleife über alle *Simplizes* $t \in \mathcal{T}$ laufen und dann in einer inneren Schleife über alle $i, j \in t$. Bei dieser Vorgehensweise brauchen wir die Determinante und die Gradienten nur für jeden Simplex einmal zu berechnen.

Definition 6.38 (Elementmatrix und Elementvektor) *Sei $t \in \mathcal{T}$. Die Matrix $\mathbf{A}_t \in \mathbb{R}^{t \times t}$ mit*

$$a_{t,ij} := \int_{\omega_t} \langle \nabla \varphi_{i,t}(x), \nabla \varphi_{j,t}(x) \rangle_2 \, dx \qquad \text{für alle } i, j \in t$$

nennen wir die Elementmatrix *für den Simplex t. Den Vektor $\mathbf{b}_t \in \mathbb{R}^t$ mit*

$$b_{t,i} := \int_{\omega_t} \varphi_{i,t}(x) f(x) \, dx \qquad \text{für alle } i \in t$$

nennen wir den Elementvektor *für diesen Simplex.*

Mit Hilfe der Elementmatrizen und -vektoren erhalten wir

$$a_{ij} = \sum_{\substack{t \in \mathcal{T} \\ i,j \in t}} a_{t,ij} \qquad \text{für alle } i, j \in \mathcal{I}, \tag{6.35a}$$

$$b_i = \sum_{\substack{t \in \mathcal{T} \\ i \in t}} b_{t,i} \qquad \text{für alle } i \in \mathcal{I}, \tag{6.35b}$$

und genau so sollten wir die Matrix $\mathbf{A}$ und den Vektor $\mathbf{b}$ auch in einer konkreten Implementierung aufstellen: Wir durchlaufen in einer äußeren Schleife alle Simplizes $t \in \mathcal{T}$, berechnen die Elementmatrix $\mathbf{A}_t$ und den Elementvektor $\mathbf{b}_t$, und addieren ihre Einträge zu den entsprechenden Einträgen der Matrix $\mathbf{A}$ und des Vektors $\mathbf{b}$.

7 Implementierung und Anwendungen des Finite-Elemente-Verfahrens

Die konkrete Umsetzung einer Finite-Elemente-Methode (kurz FEM) besteht in der Regel aus mehreren Phasen:

- Zunächst muss eine Triangulation (häufig auch einfach *Gitter* genannt) des zu behandelnden Gebiets konstruiert werden.
- Anschließend sind die *Steifigkeitsmatrix* $\mathbf{A}$ und der *Lastvektor* $\mathbf{b}$ aufzustellen.
- Das resultierende lineare Gleichungssystem muss gelöst werden, um den Koeffizientenvektor $\mathbf{x}$ der Galerkin-Lösung zu erhalten.

In diesem Kapitel widmen wir uns diesen Aufgaben und gehen darauf ein, wie sich Finite-Elemente-Verfahren für weitere Anwendungsbeispiele entwickeln lassen.

7.1 Gittererzeugung

Um ein Finite-Elemente-Verfahren anwenden zu können, benötigen wir zunächst eine Triangulation des Gebiets Ω, auf dem die zu lösende Gleichung gegeben ist. In praktischen Anwendungen kann so eine Triangulation beispielsweise dem CAD-Programm entnommen werden, mit dem ein Ingenieur ein Bauteil oder ein Gebäude konstruiert hat. Diese Aufgabe erfordert häufig relativ komplexe Algorithmen, auf die wir an dieser Stelle nicht weiter eingehen können.

Selbst wenn eine Triangulation vorliegt sind wir noch nicht fertig: Nach dem Satz 6.24 von Céa ist die durch das Finite-Elemente-Verfahren berechnet Näherungslösung zwar nahe an der besten Approximation, die in dem verwendeten Ansatzraum möglich ist, diese Approximation kann aber immer noch zu schlecht sein. Es lässt sich nachweisen, dass der Fehler der Approximation von der *Gitterweite*

$$h_{\mathcal{T}} := \max\{\|i-j\| \ : \ i,j \in t,\ i \neq j,\ t \in \mathcal{T}\}$$

abhängt, also der Länge der längsten in der Triangulation auftretenden Kante. Unter geeigneten Annahmen erhalten wir

$$\|u-u_h\|_{L^2} \leq C_0 h_{\mathcal{T}}^2, \qquad \|u-u_h\|_{H^1} \leq C_1 h_{\mathcal{T}}$$

mit geeigneten (von u abhängenden) Konstanten $C_0, C_1 \in \mathbb{R}_{>0}$. Um eine hohe Genauigkeit zu erreichen, müssen wir also eine Triangulation konstruieren, deren Kanten eine gegebene maximale Länge nicht überschreiten.

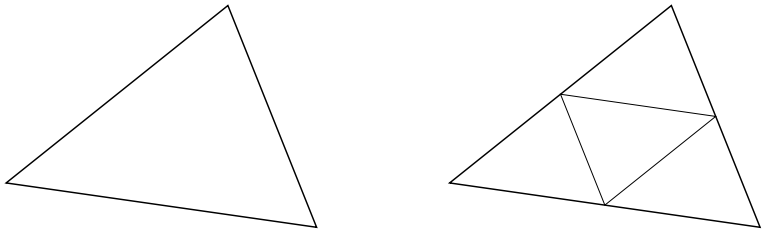

Abbildung 7.1: Rote Verfeinerung eines Dreiecks

Besonders einfach lässt sich diese Aufgabe durch eine *Gitterverfeinerung* lösen: Wir beginnen mit einer Triangulation $\mathcal{T}_0$ und zerlegen manche oder alle der in ihr enthaltenen Simplizes in kleinere Simplizes, um eine neue Triangulation $\mathcal{T}_1$ zu erhalten. Falls die in ihr enthaltenen Simplizes klein genug sind, können wir aufhören, anderenfalls wiederholen wir den Algorithmus, um zunehmend feinere Triangulationen $\mathcal{T}_2, \mathcal{T}_3, \ldots$ zu gewinnen.

Definition 7.1 (Verfeinerung) *Seien $\mathcal{T}$ und $\mathcal{T}'$ Triangulationen. Wir nennen $\mathcal{T}'$ eine* Verfeinerung *der Triangulation $\mathcal{T}$, falls für jedes $t \in \mathcal{T}$ ein $k \in \mathbb{N}$ und $t'_1, \ldots, t'_k \in \mathcal{T}'$ mit*

$$\omega_t = \bigcup_{\nu=1}^{k} \omega_{t'_\nu}$$

existieren und wenn für jedes $t' \in \mathcal{T}'$ ein $t \in \mathcal{T}$ mit

$$\omega_{t'} \subseteq \omega_t$$

existiert. Jeder Simplex der „groben“ Triangulation $\mathcal{T}$ muss sich also als Vereinigung von Simplizes der „feinen“ Triangulation $\mathcal{T}'$ darstellen lassen, und letztere darf darüber hinaus keine weiteren Simplizes enthalten.

Eine Verfeinerung einer Triangulation $\mathcal{T}$ wird in der Regel konstruiert, indem man zunächst entscheidet, welche Simplizes $t \in \mathcal{T}$ zerlegt werden sollen und welche unverändert übernommen werden können. Anschließend werden die ausgewählten Simplizes in kleinere Simplizes zerlegt.

Rote Verfeinerung. Besonders einfach ist die sogenannte „globale rote Verfeinerung“, bei der jedes Dreieck durch Verbinden der drei Kantenmittelpunkte in vier kongruente (d.h. durch Verschiebung, Rotation und Skalierung ineinander überführbare) Dreiecke zerlegt wird (siehe Abbildung 7.1).

Diese relativ einfach umsetzbare Technik bietet den großen Vorteil, dass die Form der Dreiecke der Triangulierung erhalten bleibt und sich die Kongruenz zwischen dem „Vaterdreieck“ und seinen Söhnen ausnutzen lässt, um beispielsweise die Berechnung der Elementmatrizen erheblich zu beschleunigen.

Im Rahmen unserer Datenstruktur müssen wir darauf achten, dass Eckpunkte und Kanten korrekt von benachbarten Dreiecken geteilt werden: Falls bei der Unterteilung

eines Dreiecks ein neuer Eckpunkt im Mittelpunkt einer Kante angelegt wurde, muss sichergestellt sein, dass ein Dreieck auf der anderen Seite der Kante denselben Eckpunkt verwendet. Würde es nämlich ebenfalls einen neuen Eckpunkt anlegen, wäre der Kantenmittelpunkt in der neuen Triangulation doppelt vertreten und die Stetigkeit der Basisfunktionen nicht mehr gesichert.

Diese Aufgabe lässt sich relativ einfach lösen, indem wir den Kantenmittelpunkt der Kante zuordnen, statt dem Dreieck, das ihn für seine Unterteilung benötigt. Entsprechend können wir mit den Kanten verfahren: Bei der Zerlegung einer Kante entstehen zwei neue Kanten, die wir dieser Kante zuordnen. Hinzu kommen für jedes Dreieck drei Kanten, die die Kantenmittelpunkte verbinden. Der Algorithmus kann also in vier Phasen ablaufen:

1. Konstruiere für jede Kante einen Kantenmittelpunkt.
2. Konstruiere für jede Kante zwei Kanten, indem ihre beiden Eckpunkte mit dem Kantenmittelpunkt verbunden werden.
3. Konstruiere für jedes Dreieck drei Kanten, die ihre Kantenmittelpunkte verbinden.
4. Konstruiere für jedes Dreieck vier Teildreiecke.

Falls wir eine Datenstruktur verwenden, in der Eckpunkte, Kanten und Dreiecke jeweils als eigenständige Objekte auftreten, lässt sich dieser Algorithmus besonders einfach umsetzen.

Lokale Verfeinerung. Falls die Lösung u des Variationsproblems sich in Teilen des Gebiets sehr schnell verändert, während sie in anderen Teilen „glatt“ ist, empfiehlt es sich, die Triangulierung *lokal* zu verfeinern, also nur die Dreiecke zu zerlegen, die in dem interessanten Bereich liegen.

Die rote Verfeinerung eignet sich zu diesem Zweck nur sehr bedingt: Wenn wir ein einzelnes Dreieck zerlegen, entstehen neue Ecken in seinen Kantenmittelpunkten, denen keine Ecken in den benachbarten Dreiecken entsprechen. Diese Ecken bezeichnet man als *hängende Knoten*, sie verletzen die Bedingung (6.1b) der Definition 6.2 der Triangulierung und würden zu unstetigen Basisfunktionen führen. Um hängende Knoten zu vermeiden, bliebe uns bei der roten Verfeinerung nur die Möglichkeit, auch die Nachbardreiecke zu zerlegen, und dieser Effekt würde sich fortsetzen, bis die gesamte Triangulierung verfeinert ist.

Grün-blauer Abschluss. Einen einfachen Ausweg bietet der sogenannte *grün-blaue Abschluss*:

Falls in einem Dreieck, das nicht verfeinert werden soll, genau ein hängender Knoten auftritt, wird er mit dem ihm gegenüberliegenden Eckpunkt verbunden, um sicherzustellen, dass eine wohldefinierte Triangulierung entsteht. Man spricht in diesem Fall von einer *grünen Verfeinerung*.

Falls in einem Dreieck, das nicht verfeinert werden soll, genau zwei hängende Knoten auftreten, wird der erste mit dem ihm gegenüber liegenden Eckpunkt verbunden, der

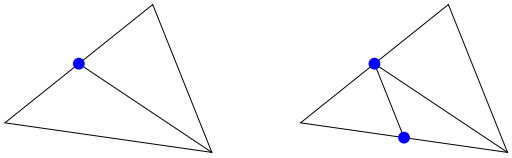

Abbildung 7.2: Grün/blauer Abschluss eines Dreiecks, falls nur eine oder nur zwei Kanten zerlegt werden sollen

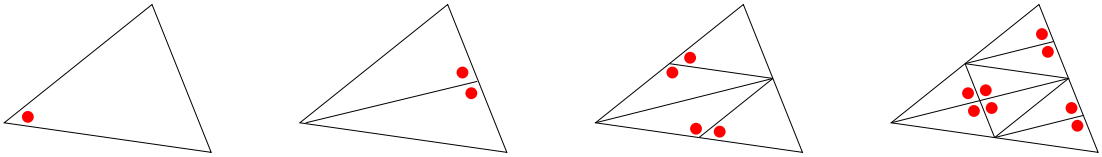

Abbildung 7.3: *Newest vertex bisection* angewendet auf ein Dreieck und die dabei entstehenden Teildreiecke. Der jeweils neueste Eckpunkt in jedem Dreieck ist markiert.

zweite mit dem ihm in dem so entstandenen kleineren Dreieck gegenüber liegenden. Diese Vorgehensweise nennt man die *blaue Verfeinerung*.

Falls drei hängende Knoten auftreten, wird das Dreieck rot verfeinert.

Da die grüne und blaue Verfeinerung lediglich die Konsistenz der Triangulierung herstellt, nachdem die interessanten Dreiecke rot verfeinert wurden, spricht man bei dieser Technik von dem „grün-blauen Abschluss“ der Triangulierung.

Durch den Abschluss (siehe Abbildung 7.2) erhalten wir zwar eine lokal verfeinerte Triangulierung, bei wiederholter Anwendung können allerdings Dreiecke mit sehr spitzen Winkeln in der Triangulation auftreten, die der Approximationsgüte schaden.

Bisektion. Das Problem der schrumpfenden Innenwinkel lässt sich mit Algorithmen verhindern, bei deren Entwicklung die lokale Gitterverfeinerung von Anfang an in Betracht gezogen wurde. Besonders einfach umzusetzen sind dabei *Bisektionsverfahren*, die jeweils eine Dreieckskante unterteilen und dabei einer Strategie folgen, die sicherstellt, dass die Innenwinkel nicht zu klein werden.

Als Beispiel untersuchen wir die Strategie „*newest vertex bisection*“, bei der wir uns in jedem Dreieck merken, welcher ihrer Eckpunkte als letzter entstanden ist. Unterteilt wird dann jeweils die Kante, die ihm gegenüber liegt. Zwar entstehen bei diesem Ansatz zunächst Dreiecke, die nicht zueinander kongruent sind und deren Innenwinkel kleiner geworden sein können, allerdings lässt sich beweisen, dass sich die Lage bei wiederholten Verfeinerungsschritten nicht weiter (siehe Abbildung 7.3) verschlimmert, weil entstehende Dreiecke kongruent zu ihren Vorgängern sind.

Elimination hängender Knoten. Wenn wir diese Strategie anwenden, um eine lokale Verfeinerung zu erreichen, werden in der Regel hängende Knoten auftreten, die wir geeignet behandeln müssen. Ein einfacher Zugang besteht darin, in mehreren Generationen vorzugehen: In der ersten Generation werden alle Dreiecke unterteilt, in denen wir die Auflösung steigern wollen. Dabei können hängende Knoten in den Kantenmittelpunkten der Dreiecke entstehen. In der zweiten Generation werden alle Dreiecke der ersten Generation unterteilt, bei denen einer der Kantenmittelpunkte einen hängenden Knoten enthält. Es ist nicht garantiert, dass dadurch die hängenden Knoten eliminiert werden, es können sogar weitere hängede Knoten entstehen. In der dritten Generation werden alle Dreiecke der zweiten Generation unterteilt, bei denen einer der Kantenmittelpunkte einen hängenden Knoten enthält. Auch dabei können weitere hängende Knoten entstehen, also wiederholen wir die Prozedur.

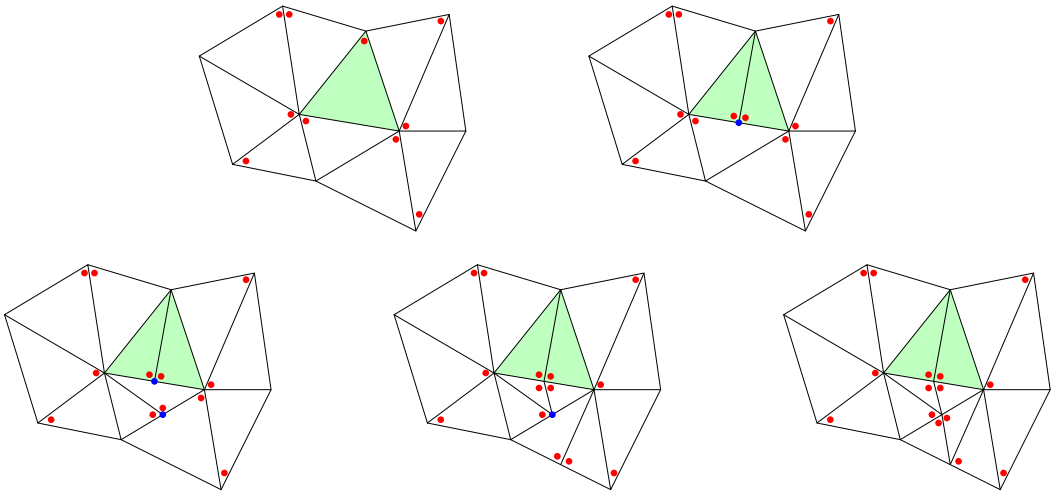

Abbildung 7.4: Folgen der lokalen Verfeinerung eines Dreiecks (grün): Um hängende Knoten (blau) zu vermeiden, müssen zwei weitere Dreiecke ebenfalls verfeinert werden.

Man kann sich überlegen, dass bei dieser Konstruktion hängende Knoten nur in den Kantenmittelpunkten der Ausgangstriangulation auftreten können: Dem jüngsten Eckpunkt gegenüber liegt immer die älteste Kante. Wenn ein Dreieck einmal unterteilt worden ist, sind die ältesten Kanten der beiden Teildreiecke jeweils die zwei nicht unterteilten Kanten des ursprünglichen Dreiecks. Bei einer weiteren Unterteilung entstehen also neue Eckpunkte nur in den Kantenmittelpunkten des ursprünglichen Dreiecks. Ein hängender Knoten in einem Kantenmittelpunkt dieses Dreiecks ist demnach nach höchstens zwei Unterteilungen beseitigt, und bei der ersten Unterteilung kann offenbar höchstens ein hängender Knoten auf der ältesten Kante des ursprünglichen Dreiecks entstanden sein, aber keine weiteren.

Daraus folgt, dass nach einer endlichen Anzahl von Schritten alle hängenden Knoten eliminiert sein werden. Allerdings kann bei einer ungünstigen Verteilung der jüngsten

Eckpunkte in dem Ausgangsgitter die Situation eintreten, dass alle Dreiecke verfeinert werden müssen, um eine wohldefinierte Triangulierung zu erhalten, so dass die Verfeinerung nicht mehr als „lokal“ zu bezeichnen wäre. Ein Beispiel für eine Triangulierung, bei der die Verfeinerung eines Dreiecks erst nach vier Generationen abgeschlossen ist, findet sich in Abbildung 7.4.

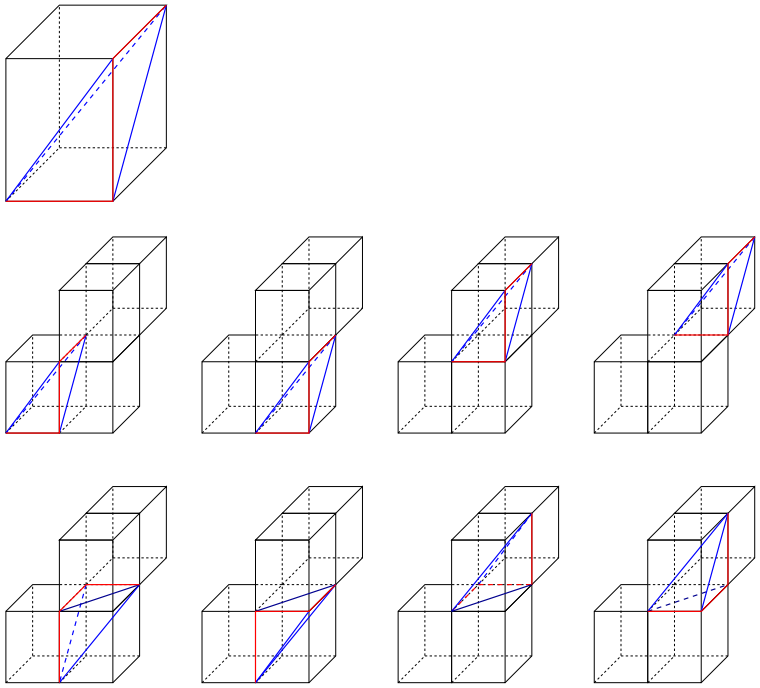

Abbildung 7.5: Verfeinerung eines Tetraeders mit dem Bey-Algorithmus. Der Kantenzug $ijk\ell$ ist in jedem Tetraeder rot markiert.

Dreidimensionaler Fall. Die Verfeinerung einer dreidimensionalen Triangulation ist schwieriger als die einer zweidimensionalen, da es nicht möglich ist, einen Tetraeder in zu ihm ähnliche Teiltetraeder zu zerlegen, so dass ein Gegenstück der roten Verfeinerung nicht existiert.

Es gibt allerdings Algorithmen, die ähnliche Eigenschaften aufweisen. Ein Beispiel ist der Algorithmus von Bey [1], der auf der Idee beruht, den zu verfeinernden Tetraeder zu einem Parallelepiped (Spat) zu ergänzen. Ein Parallelepiped lässt sich einfach in acht ähnliche Parallelepipede zerlegen, und jedes davon wieder in sechs Tetraeder. Unter diesen insgesamt 48 Tetraedern können wir acht auswählen, die eine Zerlegung des ursprünglichen Tetraeders bilden, und vier von ihnen sind ihm sogar ähnlich.

Bei einer weiteren Zerlegung der Parallelepipede bleiben sie einander ähnlich, und wenn wir den Wechsel zwischen Tetraedern und Parallelepipeden immer in derselben Weise vollziehen, bleiben die Tetraeder späterer „Generationen“ ähnlich zu denen der

ersten, so dass keine Entartung der Winkel auftreten kann.

Die Konstruktion ist in Abbildung 7.5 dargestellt. Algorithmisch lässt sich sich der Algorithmus sehr knapp beschreiben, indem man die Anordnung der einen Tetraeder beschreibenden Eckpunkte festhält und so dafür sorgt, dass dem Tetraeder immer das Parallelepiped zugeordnet wird, aus dem er ursprünglich entstanden ist. Deshalb ändern wir in diesem Abschnitt die Notation und schreiben t als Tupel $t = (i, j, k, \ell)$ statt als Menge $\{i, j, k, \ell\}$. Der durch t beschriebene Tetraeder wird durch den Bey-Algorithmus zerlegt in

$$\begin{aligned} t_1 &:= (i, m_{ij}, m_{ik}, m_{i\ell}), & t_2 &:= (m_{ij}, j, m_{jk}, m_{j\ell}), \\ t_3 &:= (m_{ik}, m_{jk}, k, m_{k\ell}), & t_4 &:= (m_{i\ell}, m_{j\ell}, m_{k\ell}, \ell), \\ t_5 &:= (m_{ij}, m_{ik}, m_{i\ell}, m_{j\ell}), & t_6 &:= (m_{ij}, m_{ik}, m_{jk}, m_{j\ell}), \\ t_7 &:= (m_{ik}, m_{i\ell}, m_{j\ell}, m_{k\ell}) & t_8 &:= (m_{ik}, m_{jk}, m_{j\ell}, m_{k\ell}), \end{aligned}$$

wobei die Kantenmittelpunkte durch

$$\begin{aligned} m_{ij} &:= (i+j)/2, & m_{ik} &:= (i+k)/2, & m_{i\ell} &:= (i+\ell)/2, \\ m_{jk} &:= (j+k)/2, & m_{j\ell} &:= (j+\ell)/2, & m_{k\ell} &:= (k+\ell)/2 \end{aligned}$$

gegeben sind. In typischen Implementierungen treten die Eckpunkte ohnehin immer in einer bestimmten Reihenfolge auf, so dass der Bey-Algorithmus ohne zusätzlichen Speicherbedarf umgesetzt werden kann.

7.2 Aufstellen des Gleichungssystems

Das Gleichungssystem (6.30) lässt sich meistens am einfachsten aufstellen, indem man sich der Gleichungen (6.35) bedient: Wir durchlaufen mit einer Schleife sämtliche Elemente $t \in \mathcal{T}$ der Triangulation und berechnen für jedes davon die Elementmatrix

$$a_{t,ij} = \int_{\omega_t} \langle \nabla\varphi_i(x), \nabla\varphi_j(x) \rangle \, dx \qquad \text{für alle } i, j \in t$$

und den Elementvektor

$$b_{t,i} = \int_{\omega_t} \varphi_i(x) f(x) \, dx \qquad \text{für alle } i \in t.$$

Anschließend addieren wir die Koeffizienten der beiden zu den Einträgen der Gesamtmatrix $\mathbf{A}$ und des Gesamtvektors $\mathbf{b}$, verwenden also

$$\begin{aligned} a_{ij} &\leftarrow a_{ij} + a_{t,ij} && \text{für alle } i, j \in t, \\ b_i &\leftarrow b_i + b_{t,i} && \text{für alle } i \in t. \end{aligned}$$

Sofern wir zuvor $\mathbf{A}$ und $\mathbf{b}$ mit Nullen gefüllt haben, enthalten beide nach Verarbeitung sämtlicher Elemente die korrekten Koeffizienten gemäß (6.35).

Da bei diesem Prozess die Matrix $\mathbf{A}$ und der Vektor $\mathbf{b}$ für das Gesamtgebiet aus den Beiträgen der Teilgebiete zusammengesetzt werden, ist der Begriff der *Assemblierung* (aus dem Englischen *to assemble*, zusammensetzen) für diesen Algorithmus üblich.

Die Tatsache, dass es nie erforderlich ist, die gesamte Geometrie auf einmal zu betrachten, ermöglicht es uns, die Finite-Elemente-Methode sehr flexibel einzusetzen, beispielsweise indem wir neben Dreiecken und Tetraedern weitere Elemente wie Vierecke, Quader, Prismen oder Pyramiden in die Triangulation aufnehmen, um die zu verarbeitenden Gebiete einfacher beschreiben zu können.

Es ist auch möglich, unterschiedliche Arten von Basisfunktionen zu mischen, beispielsweise um auszunutzen, dass die Lösung der Differentialgleichung in manchen Teilen des Gebiets besonders glatt ist.

Referenzelement. Für beide Anwendungen ist es sinnvoll, die Konstruktion der Elementmatrix und des Elementvektors etwas allgemeiner zu untersuchen. Ein wichtiges Hilfsmittel ist dabei die Verwendung eines *Referenzelements*.

Im zweidimensionalen Fall können wir beispielsweise das durch

$$\hat{t} := \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

gegebene Referenzdreieck verwenden. Ein allgemeines Dreieck $t = \{i, j, k\}$ lässt sich mit Hilfe der affinen Abbildung

$$\Phi_t : \omega_{\hat{t}} \to \omega_t, \qquad \hat{x} \mapsto i + (j - i)\hat{x}_1 + (k - i)\hat{x}_2,$$

auf das Referenzdreieck zurückführen: Sofern t regulär ist, ist Φ_t eine bijektive Abbildung mit

$$\Phi_t \begin{pmatrix} 0 \\ 0 \end{pmatrix} = i, \qquad \Phi_t \begin{pmatrix} 1 \\ 0 \end{pmatrix} = j, \qquad \Phi_t \begin{pmatrix} 0 \\ 1 \end{pmatrix} = k. \tag{7.1}$$

Auf dem Referenzdreieck lassen sich die lokalen Basisfunktionen relativ einfach angeben, nämlich als

$$\hat{\varphi}_{0,0}(\hat{x}) = 1 - \hat{x}_1 - \hat{x}_2, \qquad \hat{\varphi}_{1,0}(\hat{x}) = \hat{x}_1, \qquad \hat{\varphi}_{0,1}(\hat{x}) = \hat{x}_2,$$

und die Berechnung ihrer Gradienten ist sehr einfach, da gerade

$$\nabla\hat{\varphi}_{0,0} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \qquad \nabla\hat{\varphi}_{1,0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \nabla\hat{\varphi}_{0,1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

gelten. Da sich ω_t als Bild von $\omega_{\hat{t}}$ unter Φ_t darstellen lässt, bietet es sich an, auch die Basisfunktionen auf ω_t durch Basisfunktionen auf $\omega_{\hat{t}}$ darzustellen. Da Φ_t affin ist, sind die Abbildungen

$$\varphi_{t,i} := \hat{\varphi}_{0,0} \circ \Phi_t^{-1}, \qquad \varphi_{t,j} := \hat{\varphi}_{1,0} \circ \Phi_t^{-1}, \qquad \varphi_{t,k} := \hat{\varphi}_{0,1} \circ \Phi_t^{-1} \tag{7.2}$$

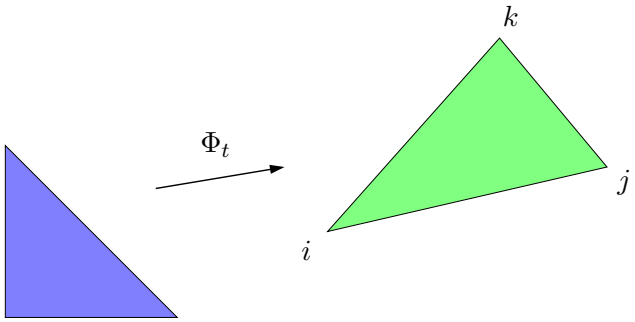

Abbildung 7.6: Die Abbildung Φ_t überführt das Referenzdreieck $\omega_{\hat{t}}$ auf ein allgemeines Dreieck ω_t mit den Eckpunkten $t = \{i, j, k\}$.

lineare Polynome, die außerdem wegen (7.1) die Gleichungen

$$\begin{aligned}
\varphi_{t,i}(i) &= 1, & \varphi_{t,i}(j) &= 0, & \varphi_{t,i}(k) &= 0,\\
\varphi_{t,j}(i) &= 0, & \varphi_{t,j}(j) &= 1, & \varphi_{t,j}(k) &= 0,\\
\varphi_{t,k}(i) &= 0, & \varphi_{t,k}(j) &= 0, & \varphi_{t,k}(k) &= 1
\end{aligned}$$

erfüllen, und da stückweise lineare Funktionen durch ihre Werte in den Eckpunkten eines Dreiecks bereits eindeutig festgelegt sind, folgen

$$\varphi_{t,i} = \varphi_i|_t, \qquad \varphi_{t,j} = \varphi_j|_t, \qquad \varphi_{t,k} = \varphi_k|_t.$$

Demnach stellt uns (7.2) eine alternative Darstellung der lokalen Basisfunktionen zur Verfügung, die es uns ermöglicht, einerseits einige Berechnungen etwas geschickter auszuführen und andererseits das Finite-Elemente-Verfahren erheblich zu verallgemeinern.

Transformierte Gradienten. Aus (7.2) ergibt sich eine alternative Möglichkeit für die Berechnung der Gradienten der Basisfunktionen: Wir bezeichnen mit

$$D\Phi_t := \begin{pmatrix} j_1 - i_1 & k_1 - i_1 \\ j_2 - i_2 & k_2 - i_2 \end{pmatrix}$$

die Jacobi-Matrix der Abbildung Φ_t, also die Matrix der ersten partiellen Ableitungen der einzelnen Komponenten der Abbildung. Mit Hilfe der Kettenregel und des Umkehrsatzes erhalten wir

$$\nabla\varphi_{t,i} = \nabla(\hat{\varphi}_{0,0} \circ \Phi_t^{-1}) = (D\Phi_t^{-1})^*(\nabla\hat{\varphi}_{0,0}) \circ \Phi_t^{-1}, \tag{7.3a}$$

$$\nabla\varphi_{t,j} = \nabla(\hat{\varphi}_{1,0} \circ \Phi_t^{-1}) = (D\Phi_t^{-1})^*(\nabla\hat{\varphi}_{1,0}) \circ \Phi_t^{-1}, \tag{7.3b}$$

$$\nabla\varphi_{t,k} = \nabla(\hat{\varphi}_{0,1} \circ \Phi_t^{-1}) = (D\Phi_t^{-1})^*(\nabla\hat{\varphi}_{0,1}) \circ \Phi_t^{-1}. \tag{7.3c}$$

Die Gradienten der Referenz-Basisfunktionen $\hat{\varphi}_{0,0}$, $\hat{\varphi}_{1,0}$ und $\hat{\varphi}_{0,1}$ sind uns bereits bekannt, so dass wir lediglich die Inverse $D\Phi_t^{-1}$ der Jacobi-Matrix zu berechnen brauchen, um die Gradienten der lokalen Basisfunktionen zu erhalten.

Transformierte Integrale. Auch die Berechnung der Integrale über ω_t lässt sich mit Hilfe des Referenzelements bewerkstelligen, indem wir sie geeignet transformieren.

Erinnerung 7.2 (Variablentransformation) *Seien $U, V \subseteq \mathbb{R}^d$ offene Gebiete, und sei $\Phi : U \to V$ eine bijektive stetig differenzierbare Abbildung derart, dass $D\Phi(x)$ für alle $x \in U$ invertierbar ist.*

Falls $f : V \to \mathbb{R}$ integrierbar ist, ist auch $f \circ \Phi |\det D\Phi|$ integrierbar und es gilt

$$\int_V f(x)\,dx = \int_U f(\Phi(\hat{x}))|\det D\Phi(\hat{x})|\,d\hat{x}. \tag{7.4}$$

Für reguläre Simplizes ist Φ_t bijektiv und die Jacobi-Matrix $D\Phi_t$ ist konstant und invertierbar, so dass sich die Variablentransformationsformel (7.4) anwenden lässt, um

$$\int_{\omega_t} f(x)\,dx = \int_{\omega_{\hat{t}}} f(\Phi_t(\hat{x}))|\det D\Phi_t(\hat{x})|\,d\hat{x}$$

zu erhalten. Beispielsweise ergibt sich für die Einträge der Elementmatrix mit (7.3) die Gleichung

$$\begin{aligned} a_{t,ij} &= \int_{\omega_t} \langle \nabla\varphi_i(x), \nabla\varphi_j(x)\rangle\,dx \\ &= \int_{\omega_{\hat{t}}} |\det D\Phi_t(\hat{x})|\langle \nabla\varphi_i(\Phi_t(\hat{x})), \nabla\varphi_j(\Phi_t(\hat{x}))\rangle\,d\hat{x} \\ &= \int_{\omega_{\hat{t}}} |\det D\Phi_t(\hat{x})|\langle (D\Phi_t^{-1}(\hat{x}))^*\nabla\hat{\varphi}_{0,0}(\hat{x}), (D\Phi_t^{-1}(\hat{x}))^*\nabla\hat{\varphi}_{1,0}(\hat{x})\rangle\,d\hat{x}. \end{aligned} \tag{7.5}$$

Wenn wir die Einträge der Elementmatrix mit Hilfe dieser Gleichung berechnen, können wir auf Φ_t^{-1} vollständig verzichten und benötigen lediglich $D\Phi_t$ sowie die Transponierte von $D\Phi_t^{-1}$. Das eröffnet uns die Möglichkeit, auch allgemeinere Elemente zu verwenden.

In unserem einfachen Fall ist Φ_t affin, also $D\Phi_t$ konstant, also sind auch $D\Phi_t^{-1}$ und $(D\Phi_t^{-1})^*$ konstant, so dass sich die Formel in die einfachere Form

$$a_{t,ij} = |\det D\Phi_t| \int_{\omega_{\hat{t}}} \langle (D\Phi_t^{-1})^*\nabla\hat{\varphi}_{0,0}(\hat{x}), (D\Phi_t^{-1})^*\nabla\hat{\varphi}_{1,0}(\hat{x})\rangle\,d\hat{x}$$

bringen lässt. Für unsere linearen Basisfunktionen sind $\nabla\hat{\varphi}_{0,0}$ und $\nabla\hat{\varphi}_{1,0}$ konstant, so dass sich wegen

$$\int_{\omega_{\hat{t}}} 1\,d\hat{x} = 1/2$$

die noch etwas einfachere Gleichung

$$a_{t,ij} = \frac{|\det D\Phi_t|}{2}\langle (D\Phi_t^{-1})^*\nabla\hat{\varphi}_{0,0}, (D\Phi_t^{-1})^*\nabla\hat{\varphi}_{1,0}\rangle$$

ergibt, in der ausschließlich praktisch berechenbare Ausdrücke auftreten.

Gekrümmte Elemente. Die Gleichung (7.5) erlaubt es uns allerdings, auch mit sehr viel allgemeineren Elementen als Dreiecken und Tetraedern zu arbeiten: Wenn wir eine *beliebige* Abbildung $\Phi_t : \omega_{\hat{t}} \to \Phi_t(\omega_{\hat{t}})$ einsetzen, die die Voraussetzungen der Erinnerung 7.2 erfüllt, bleibt (7.5) gültig. Demzufolge könnten wir ein Gebiet $\Omega \subseteq \mathbb{R}^d$ auch durch

$$\bar{\Omega} = \bigcup_{t \in \mathcal{T}} \Phi_t(\omega_{\hat{t}})$$

beschreiben. An die Stelle der Verträglichkeitsbedingung (6.1b) könnte eine Bedingung der Form

$$\exists t_0, s_0 \subseteq \hat{t} : \Phi_t(\omega_{\hat{t}}) \cap \Phi_s(\omega_{\hat{t}}) = \Phi_t(\omega_{t_0}) = \Phi_s(\omega_{s_0}) \qquad \text{für alle } t, s \in \mathcal{T}$$

treten. Wenn wir eine geeignete Konstruktion für die Abbildungen Φ_t wählen, lässt sich die Gültigkeit dieser Bedingung leicht nachprüfen. Wenn wir sicherstellen wollen, dass sich die lokal definierte Basisfunktionen zu *stetigen* globalen Basisfunktionen zusammensetzen lassen, müssen wir zusätzliche Forderungen an die Form der Abbildungen Φ_t und Φ_s auf ω_{t_0} und ω_{s_0} stellen.

Allgemeine Basisfunktionen. Eine weitere Verallgemeinerung besteht darin, andere als die bisher verwendeten stückweise linearen Basisfunktionen einzusetzen: Wenn wir eine beliebige Basis $(\varphi_i)_{i \in \mathcal{I}}$ verwenden wollen, bei der $\mathcal{I}$ nicht mehr unbedingt die Menge der inneren Eckpunkte der Triangulation ist, können wir immer noch mit Hilfe der Gleichung (6.35) arbeiten, indem wir für jedes $t \in \mathcal{T}$ die Menge

$$\mathcal{I}_t := \{i \in \mathcal{I} \ : \ \varphi_i|_{\omega_t} \neq 0\}$$

der auf ω_t nicht verschwindenden Basisfunktionen einführen und Elementmatrizen und -vektoren für diese Menge durch

$$a_{t,ij} := \int_{\omega_t} \langle \nabla \varphi_i(x), \nabla \varphi_j(x) \rangle \, dx \qquad \text{für alle } i, j \in \mathcal{I}_t,$$

$$b_{t,i} := \int_{\omega_t} \varphi_i(x) f(x) \, dx \qquad \text{für alle } i \in \mathcal{I}_t$$

definieren. Je nach Wahl der Basisfunktionen kann es möglich sein, die Integrale analytisch zu bestimmen, es kann aber auch erforderlich werden, eine Quadraturformel wie die Kantenmittelpunktregel (vgl. Definition 6.36) einzusetzen. Besonders attraktiv sind dabei natürlich Quadraturformeln, die Polynome des auftretenden Grades exakt behandeln. Die Kantenmittelpunktregel beispielsweise führt für lineare und quadratische Polynome zu exakten Ergebnissen.

Auch im Fall allgemeiner Basisfunktionen ist es häufig sehr sinnvoll, mit einem Referenzelement zu arbeiten. Beispielsweise könnten wir auf dem Referenzdreieck $\hat{t}$ lokale Basisfunktionen $(\hat{\varphi}_i)_{i \in \widehat{\mathcal{I}}}$ definieren und für jedes Dreieck $t \in \mathcal{T}$ der Triangulation eine bijektive Abbildung $\iota_t : \widehat{\mathcal{I}} \to \mathcal{I}_t$ fixieren, die den Referenz-Basisfunktionen die Basisfunktionen auf ω_t so zuordnet, dass wieder

$$\varphi_{\iota_t(i)}|_{\omega_t} = \hat{\varphi}_i \circ \Phi_t^{-1} \qquad \text{für alle } i \in \widehat{\mathcal{I}} \tag{7.6}$$

gilt. Dann folgt wie zuvor

$$a_{t,\iota_t(i)\iota_t(j)} = \int_{\omega_{\hat{\iota}}} |\det D\Phi_t(\hat{x})| \langle (D\Phi_t^{-1}(\hat{x}))^* \nabla\hat{\varphi}_i(\hat{x}), (D\Phi_t^{-1}(\hat{x}))^* \nabla\hat{\varphi}_j(\hat{x})\rangle \, d\hat{x},$$
$$b_{t,\iota_t(i)} = \int_{\omega_{\hat{\iota}}} |\det D\Phi_t(\hat{x})| \hat{\varphi}_i(\hat{x}) f(\Phi_t(\hat{x})) \, d\hat{x} \qquad \text{für alle } i,j \in \widehat{\mathcal{I}}.$$

Auch im Fall allgemeiner Basisfunktionen können wir also die gesamte Berechnung auf die Betrachtung des Referenzelements zurückführen.

Allerdings müssen wir darauf achten, dass die Abbildung ι_t, die den lokalen Indizes aus der Menge $\widehat{\mathcal{I}}$ die globalen Indizes aus der Menge $\mathcal{I}_t \subseteq \mathcal{I}$ zuordnet, so gewählt ist, dass die gemäß (7.6) zusammengesetzten globalen Basisfunktionen φ_i stetig sind.

Quadratische Basisfunktionen. Als Beispiel untersuchen wir quadratische Basisfunktionen. Sie lassen sich besonders einfach konstruieren, indem wir wieder dafür sorgen, dass sie in bestimmten Punkten gleich eins und in alle anderen gleich null sind. Da der Raum der quadratischen Polynome für $d = 2$ sechsdimensional und für $d = 3$ zehndimensional ist, sind quadratische Polynome nicht mehr durch die Werte in den Eckpunkten eines Dreiecks beziehungsweise Tetraeders eindeutig festgelegt. Wir können dieses Problem lösen, indem wir die Kantenmittelpunkte hinzunehmen: Für $d = 2$ haben wir dann drei Eckpunkte und drei Kantenmittelpunkte, für $d = 3$ haben wir vier Eckpunkte und sechs Kantenmittelpunkte.

Wir beschränken uns hier auf den zweidimensionalen Fall und wollen lokale Basisfunktionen auf dem Referenzdreieck definieren. Dazu können wir ausnutzen, dass das Produkt zweier linearer Funktionen eine quadratische Funktion ist. Insbesondere erhalten wir durch die Multiplikation zweier der bisher verwendeten linearen Basisfunktionen eine quadratische Funktion, die in drei Eckpunkten und zwei Kantenmittelpunkten verschwindet und im verbliebenen Kantenmittelpunkt den Wert $1/4$ annimmt, so dass wir mit

$$\begin{aligned}
\hat{\varphi}_3(\hat{x}) &:= 4\varphi_{1,0}(\hat{x})\varphi_{0,1}(\hat{x}) = 4\hat{x}_1\hat{x}_2, \\
\hat{\varphi}_4(\hat{x}) &:= 4\varphi_{0,1}(\hat{x})\varphi_{0,0}(\hat{x}) = 4\hat{x}_2(1-\hat{x}_1-\hat{x}_2), \\
\hat{\varphi}_5(\hat{x}) &:= 4\varphi_{0,0}(\hat{x})\varphi_{1,0}(\hat{x}) = 4(1-\hat{x}_1-\hat{x}_2)\hat{x}_1
\end{aligned}$$

geeignete Basisfunktionen für die drei Kantenmittelpunkte gefunden haben. Da diese Funktionen in den Kantenmittelpunkten gerade gleich eins sind, können wir sie verwenden, um die bisherigen linearen Basisfunktionen so zu modifizieren, dass sie in den Kantenmittelpunkten verschwinden. Beispielsweise nimmt $\varphi_{0,0}$ in $(1/2, 0)$ und $(0, 1/2)$ den Wert $1/2$ an und verschwindet in $(1/2, 1/2)$, so dass

$$\begin{aligned}
\hat{\varphi}_0(\hat{x}) &:= \hat{\varphi}_{0,0}(\hat{x}) - (\hat{\varphi}_4(\hat{x}) + \hat{\varphi}_5(\hat{x}))/2 = \hat{\varphi}_{0,0}(\hat{x})(1 - 2\hat{\varphi}_{0,1}(\hat{x}) - 2\hat{\varphi}_{1,0}(\hat{x})) \\
&= \hat{\varphi}_{0,0}(\hat{x})(2\hat{\varphi}_{0,0}(\hat{x}) - 1)
\end{aligned}$$

im Punkt $(0, 0)$ den Wert 1 annimmt, in $(1, 0)$, $(0, 1)$ und $(1/2, 1/2)$ wegen des ersten Faktors und in $(1/2, 0)$ und $(0, 1/2)$ wegen des zweiten Faktors verschwindet. Entsprechend

erhalten wir

$$\hat{\varphi}_1(\hat{x}) := \hat{\varphi}_{1,0}(\hat{x})(2\hat{\varphi}_{1,0}(\hat{x}) - 1),$$
$$\hat{\varphi}_2(\hat{x}) := \hat{\varphi}_{0,1}(\hat{x})(2\hat{\varphi}_{0,1}(\hat{x}) - 1).$$

Da jede der Funktionen $(\hat{\varphi}_i)_{i=0}^5$ in einem der sechs Punkte gleich eins ist und in allen anderen verschwindet, müssen die Funktionen linear unabhängig sein. Da sie alle in dem sechsdimensionalen Raum der quadratischen Funktionen liegen, sind sie also eine Basis dieses Raums.

Für ein durch $t = \{i, j, k\}$ gegebenes Dreieck können wir die Menge der Eckpunkte und Kantenmittelpunkte durch

$$\mathcal{I}_t := \{i, j, k, (j+k)/2, (k+i)/2, (i+j)/2\}$$

definieren und durch

$$\iota_t(0) := i, \qquad \iota_t(1) := j, \qquad \iota_t(2) := k,$$
$$\iota_t(3) := (j+k)/2, \qquad \iota_t(4) := (k+i)/2, \qquad \iota_t(5) := (i+j)/2$$

eine Zuordnung der lokalen Indizes aus $\widehat{\mathcal{I}} := \{0, \ldots, 5\}$ zu den globalen definieren. Die Numerierung folgt dabei der Systematik, dass $\iota_t(i+3)$ jeweils der Kantenmittelpunkt ist, der $\iota_t(i)$ gegenüber liegt, die globale Indexmenge aller Eck- und Kantenmittelpunkte (mit Ausnahme der Randpunkte) ist durch

$$\mathcal{I} := \bigcup_{t \in \mathcal{T}} \mathcal{I}_t \setminus \partial\Omega$$

gegeben.

Auf jeder Kante eines Dreiecks liegen bei unserer Konstruktion jeweils drei Punkte (einschließlich der Eckpunkte), und eine quadratische Funktion ist auf der (eindimensionalen) Kante durch ihre Werte in drei Punkten bereits eindeutig festgelegt, so dass der von uns konstruierte Ansatzraum aus stetigen Funktionen besteht, also nach Satz 6.27 ein Teilraum des Sobolew-Raums $H^1(\Omega)$ ist.

Damit können wir ihn wie den Raum der stückweise linearen Funktionen in unserer Galerkin-Diskretisierung verwenden. Infolge der zusätzlichen Unbekannten in den Kantenmittelpunkten wird dabei ein wesentlich größeres lineares Gleichungssystem entstehen, allerdings kann es sein, dass sich bei einer hinreichend glatten Lösung u auch eine wesentlich höhere Genauigkeit ergibt, so dass eventuell eine sehr viel gröber aufgelöste Triangulation bereits ausreicht.

Viereckselemente. Die in diesem Abschnitt eingeführten Techniken lassen sich auch verwenden, um statt Dreiecken (oder Tetraedern) allgemeinere Elemente einzusetzen. Als Beispiel beschäftigen wir uns mit Viereckselementen, denen in vielen praktischen Anwendungen der Vorzug gegenüber Dreieckselementen gegeben wird.

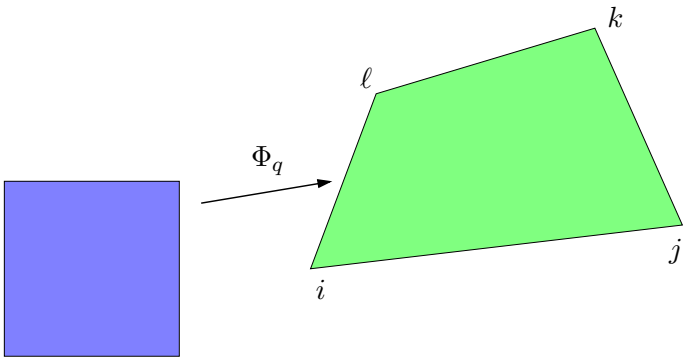

Abbildung 7.7: Die Abbildung Φ_q überführt das Referenzquadrat auf ein allgemeines Viereck mit den Eckpunkten i, j, k, ℓ.

Als Referenzelement verwenden wir das Einheitsquadrat

$$\hat{\omega} := [0,1] \times [0,1],$$

dem wir mit Hilfe der Abbildung

$$\Phi_q : \hat{\omega} \to \Phi_q(\hat{\omega}), \qquad \hat{x} \mapsto i + (j-i)\hat{x}_1 + (\ell - i)\hat{x}_2 + (k - \ell + i - j)\hat{x}_1\hat{x}_2,$$

das Viereck $\omega_q := \Phi_q(\hat{\omega})$ mit den Eckpunkten $i, j, k, \ell \in \mathbb{R}^2$ zuordnen. Falls $\ell - k = i - j$ (gleichwertig mit $\ell - i = k - j$) gilt, ist das Viereck ein Parallelogramm und der letzte Term verschwindet, so dass Φ_q wieder eine affine Abbildung ist. Im allgemeinen Fall müssen wir damit leben, dass $D\Phi_q$ nicht konstant ist, so dass für die Berechnung der Elementmatrizen eine Quadraturformel erforderlich wird und wir $D\Phi_q(\hat{x})$, $\det D\Phi_q(\hat{x})$ und $(D\Phi_q(\hat{x})^{-1})^*$ in jedem Quadraturpunkt berechnen müssen.

Wir sind daran interessiert, analog zu dem bei den Dreiecken verwendeten Ansatz Basisfunktionen zu definieren, die jeweils in einem Eckpunkt gleich eins sind und in den anderen verschwinden. Auf dem Referenzelement bieten sich *bilineare* Funktionen an, also solche, die in $\hat{x}_1$ und $\hat{x}_2$ linear sind. Die bilinearen Funktionen

$$\hat{\varphi}_0(\hat{x}) := (1-\hat{x}_1)(1-\hat{x}_2), \qquad \hat{\varphi}_1(\hat{x}) := \hat{x}_1(1-\hat{x}_2),$$
$$\hat{\varphi}_2(\hat{x}) := \hat{x}_1\hat{x}_2, \qquad \hat{\varphi}_3(\hat{x}) := (1-\hat{x}_1)\hat{x}_2$$

sind in jeweils einem der Eckpunkte $(0,0)$, $(1,0)$, $(1,1)$ und $(0,1)$ gleich eins und verschwinden in den anderen. Also sind sie linear unabhängig und bilden eine Basis des vierdimensionalen Raums der bilinearen Funktionen. Wir können sie mittels

$$\iota_q(0) := i, \qquad \iota_q(1) := j, \qquad \iota_q(2) := k, \qquad \iota_q(3) := \ell$$

den Eckpunkten des Vierecks ω_q zuordnen Nun können wir wieder mittels

$$\varphi_{\iota_q(i)} = \hat{\varphi}_i \circ \Phi_q^{-1} \qquad \text{für alle } i \in \mathcal{I}_{\hat{t}} := \{0,1,2,3\}$$

Basisfunktionen auf ω_q definieren. Es lässt sich relativ einfach nachrechnen, dass diese Basisfunktionen auf jeder Kante durch ihre Werte in den zugehörigen Eckpunkten eindeutig festgelegt sind, so dass wir wieder einen Ansatzraum aus stetigen Funktionen erhalten.

Bei der Berechnung der Matrix $\mathbf{A}$ und des Vektor $\mathbf{b}$ können wir analog zu (7.5) vorgehen, um die Elementmatrizen und -vektoren zu berechnen und mit ihrer Hilfe die Gesamtmatrix und den Gesamtvektor zu assemblieren. Wie bereits erwähnt müssen wir dabei auf eine Quadraturformel zurückgreifen, falls ω_q kein Parallelogramm und $D\Phi_q$ damit nicht konstant ist. Der Rechenaufwand wächst in diesem Fall erheblich, weil für jeden Quadraturpunkt $D\Phi_q$ berechnet und invertiert werden muss.

Da unsere Basisfunktionen auf Viereckselementen ebenso wie die linearen Basisfunktionen auf Dreieckselementen durch die Werte in den Eckpunkten eindeutig festgelegt sind, können wir auch beide Sorten von Elementen ohne größeren Aufwand in einer Triangulation mischen, sofern wir darauf achten, dass für jedes Element jeweils die richtigen Elementmatrizen und -vektoren aufgestellt werden.

7.3 Schnelle Lösungsverfahren

Die Galerkin-Diskretisierung führt zu einem linearen Gleichungssystem der Form

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

bei dem die Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und der Vektor $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ gegeben sind und wir den Vektor $\mathbf{x} \in \mathbb{R}^{\mathcal{I}}$ suchen. Wir haben bereits gesehen, dass die Matrix $\mathbf{A}$ symmetrisch und positiv definit ist, falls die Bilinearform a diese Eigenschaften aufweist. In unserem Modellproblem ist das der Fall, so dass sich das Verfahren der konjugierten Gradienten einsetzen lässt, um das Gleichungssystem zu lösen.

Allerdings tritt dabei eine Schwierigkeit auf: Die Anzahl der Schritte, die das Verfahren für die Berechnung einer ausreichend genauen Näherungslösung benötigt, nimmt zu, wenn wir das Gitter verfeinern. Für praxistaugliche Genauigkeiten entsteht dadurch ein sehr hoher Rechenaufwand, obwohl der Aufwand *pro Schritt* des Verfahrens nicht allzu hoch ist.

Wir können dieses Problem vermeiden, indem wir die besonderen Eigenschaften unserer Aufgabenstellung ausnutzen: Das lineare Gleichungssystem ist nicht beliebig, sondern resultiert aus der Galerkin-Diskretisierung eines koerziven Variationsproblems auf einem bestimmten Gitter. Indem wir geschickt mehrere Gitter mit unterschiedlich feiner Auflösung kombinieren, können wir ein Verfahren konstruieren, das wesentlich schneller als die bisher behandelten arbeitet.

Eindimensionale Korrektur. Als Vorbereitung untersuchen wir ein einfaches Iterationsverfahren: Falls wir eine Näherungslösung $\tilde{u}_h \in V_h$ der Variationsaufgabe

$$a(v_h, u_h) = \beta(v_h) \qquad \text{für alle } v_h \in V_h$$

kennen, die nur in einem Punkt von der echten Lösung u_h abweicht, falls also $\tilde{u}_h = u_h - \alpha\varphi_i$ mit $\alpha \in \mathbb{R}$ und $i \in \mathcal{I}$ gilt, können wir die exakte Lösung berechnen, indem wir

$$a(v_h, \tilde{u}_h) = a(v_h, u_h - \alpha\varphi_i) = a(v_h, u_h) - a(v_h, \alpha\varphi_i) = \beta(v_h) - a(v_h, \alpha\varphi_i)$$

ausnutzen und die Testfunktion $v_h = \varphi_i$ einsetzen: Es folgt

$$\alpha a(\varphi_i, \varphi_i) = a(\varphi_i, \alpha\varphi_i) = \beta(\varphi_i) - a(\varphi_i, \tilde{u}_h), \tag{7.7}$$

so dass wir α direkt berechnen können, ohne die exakte Lösung u_h kennen zu müssen.

Falls $\tilde{u}_h \in V_h$ eine beliebige Funktion ist, können wir dieselbe Vorgehensweise einsetzen, um immerhin noch eine Verbesserung zu erreichen: Wir berechnen α mit (7.7) und verwenden $\tilde{u}'_h := \tilde{u}_h + \alpha\varphi_i$ als eine neue Näherung der Lösung u_h.

Um zu beurteilen, wie viel besser die neue Näherung ist, können wir die durch

$$\|v\|_a := \sqrt{a(v,v)} \qquad \text{für alle } v \in V$$

gegebene *Energienorm* verwenden (vgl. Bemerkung 6.25). Da die Bilinearform koerziv ist, ist die Energienorm äquivalent zu der Norm $\|\cdot\|_V$ des Hilbert-Raums V. Sie eignet sich allerdings besonders gut, um Aussagen über die Konvergenz von Iterationsverfahren zu formulieren.

Lemma 7.3 (Fehlerreduktion) *Sei $\tilde{u}_h \in V_h$, sei $i \in \mathcal{I}$. Mit dem durch (7.7) gegebenen α und $\tilde{u}'_h := \tilde{u}_h + \alpha\varphi_i$ gilt*

$$\|u_h - \tilde{u}'_h\|_a^2 \leq \|u_h - \tilde{u}_h\|_a^2 - \frac{a(\varphi_i, u_h - \tilde{u}_h)^2}{a(\varphi_i, \varphi_i)}.$$

Beweis. Nach Konstruktion (7.7) gilt

$$\alpha a(\varphi_i, \varphi_i) = \beta(\varphi_i) - a(\varphi_i, \tilde{u}_h) = a(\varphi_i, u_h) - a(\varphi_i, \tilde{u}_h) = a(\varphi_i, u_h - \tilde{u}_h).$$

Für den Fehler erhalten wir

$$\begin{aligned}
a(u_h - \tilde{u}'_h, u_h - \tilde{u}'_h) &= a(u_h - \tilde{u}_h - \alpha\varphi_i, u_h - \tilde{u}_h - \alpha\varphi_i) \\
&= a(u_h - \tilde{u}_h, u_h - \tilde{u}_h) - 2\alpha a(\varphi_i, u_h - \tilde{u}_h) + \alpha^2 a(\varphi_i, \varphi_i) \\
&= a(u_h - \tilde{u}_h, u_h - \tilde{u}_h) - 2\alpha a(\varphi_i, u_h - \tilde{u}_h) + \alpha a(\varphi_i, u_h - \tilde{u}_h) \\
&= a(u_h - \tilde{u}_h, u_h - \tilde{u}_h) - \alpha a(\varphi_i, u_h - \tilde{u}_h) \\
&= a(u_h - \tilde{u}_h, u_h - \tilde{u}_h) - \frac{a(\varphi_i, u_h - \tilde{u}_h)}{a(\varphi_i, \varphi_i)} a(\varphi_i, u_h - \tilde{u}_h),
\end{aligned}$$

und das ist die gewünschte Abschätzung. ∎

Die neue Näherungslösung $\tilde{u}'_h$ wird also näher an der exakten Lösung u_h liegen, sofern $a(\varphi_i, u_h - \tilde{u}_h) \neq 0$ gilt, sofern also der Fehler $u_h - \tilde{u}_h$ nicht gerade senkrecht auf φ_i steht. In diesem Fall könnte $\tilde{u}_h$ auch nicht mehr verbessert werden, indem man ein Vielfaches von φ_i hinzuaddiert.

Gauß-Seidel-Iteration. Die Idee der *Gauß-Seidel-Iteration* besteht darin, der Reihe nach für alle Indizes $i \in \mathcal{I}$ die beschriebene Korrektur durchzuführen.

Der Einfachheit halber gehen wir davon aus, dass die Indizes fortlaufend numeriert sind, dass also $\mathcal{I} = \{1, \ldots, n\}$ gilt. Wir setzen voraus, dass eine Näherungslösung $u_h^{(m)} \in V_h$ der Variationsaufgabe vorliegt und definieren $u^{(m,0)} := u_h^{(m)}$.

Nun durchlaufen wir $i = 1, 2, \ldots, n$ der Reihe nach und berechnen jeweils $\alpha_i \in \mathbb{R}$ mit

$$\alpha_i a(\varphi_i, \varphi_i) = \beta(\varphi_i) - a(\varphi_i, u^{(m,i-1)}), \tag{7.8}$$

so dass wir die Korrektur

$$u^{(m,i)} := u^{(m,i-1)} + \alpha_i \varphi_i$$

durchführen können. Nach n dieser einzelnen Korrekturen erhalten wir eine verbesserte Näherungslösung $u_h^{(m+1)} := u^{(m,n)}$.

Nach Lemma 7.3 kann jeder der Einzelschritte die Energienorm des Fehlers nur reduzieren, so dass der ungünstige Fall

$$\|u_h - u_h^{(m+1)}\|_a = \|u_h - u_h^{(m)}\|_a$$

ausschließlich auftreten kann, wenn *keiner* der Teilschritte eine Verbesserung gebracht hat. Das ist gerade der Fall, wenn

$$a(\varphi_i, u_h - u_h^{(m)}) = 0 \qquad \text{für alle } i \in \mathcal{I}$$

gilt, und da $(\varphi_i)_{i\in\mathcal{I}}$ eine Basis des Raums V_h ist, der wiederum $u_h - u_h^{(m)}$ enthält, kann das nur passieren, wenn $u_h^{(m)} = u_h$ gilt, wenn uns also schon die exakte Lösung vorliegt.

Matrixdarstellung. Die Gauß-Seidel-Iteration hat den großen Vorteil, dass sie sich einfach implementieren lässt: Wenn wir mit $\mathbf{x}^{(m,i)} \in \mathbb{R}^{\mathcal{I}}$ den Koeffizientenvektor eines Zwischenergebnisses

$$u^{(m,i)} = \sum_{j\in\mathcal{I}} x_j^{(m,i)} \varphi_j$$

bezeichnen, nimmt (7.8) die Form

$$\begin{aligned} \alpha_i a_{ii} = \alpha_i a(\varphi_i, \varphi_i) &= \beta(\varphi_i) - a(\varphi_i, u^{(m,i-1)}) \\ &= b_i - \sum_{j\in\mathcal{I}} a_{ij} x_j^{(m,i-1)} = (\mathbf{b} - \mathbf{A}\mathbf{x}^{(m,i-1)})_i \end{aligned}$$

an, so dass wir

$$\alpha_i = \frac{(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m,i-1)})_i}{a_{ii}}$$

erhalten und der Vektor des nächsten Teilschritts durch

$$x_j^{(m,i)} := \begin{cases} x_j^{(m,i-1)} + \alpha_i & \text{falls } i = j, \\ x_j^{(m,i-1)} & \text{ansonsten} \end{cases} \qquad \text{für alle } j \in \mathcal{I}$$

gegeben ist. Das resultierende Verfahren lässt sich wie folgt zusammenfassen:

Definition 7.4 (Gauß-Seidel-Verfahren) *Sei $\mathcal{I} = \{1, \dots, n\}$, sei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ positiv definit, sei $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$.*

Ausgehend von einem Ausgangsvektor $\mathbf{x}^{(0)} \in \mathbb{R}^{\mathcal{I}}$ berechnet das Gauß-Seidel-Verfahren *mit der Vorschrift*

$$\begin{aligned}
\mathbf{x}^{(m,0)} &:= \mathbf{x}^{(m)}, \\
x_j^{(m,i)} &:= \begin{cases} x_i^{(m,i-1)} + a_{ii}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m,i-1)})_i & \text{falls } i = j, \\ x_j^{(m,i-1)} & \text{ansonsten,} \end{cases} \\
\mathbf{x}^{(m+1)} &:= \mathbf{x}^{(m,n)} \qquad \text{für alle } m \in \mathbb{N}_0,\ i, j \in \mathcal{I}
\end{aligned}$$

eine Folge $(\mathbf{x}^{(m)})_{m=0}^{\infty}$ von Näherungslösungen des Gleichungssystems $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Ein großer Vorteil des Gauß-Seidel-Verfahrens besteht darin, dass es sich sehr einfach implementieren lässt und keinen zusätzlichen Speicherplatz benötigt, wenn wir es in der folgenden Form schreiben:

```
procedure gauss_seidel(A, b, var x);
for i ∈ I do begin
  r ← b_i;
  for j ∈ I do r ← r − a_ij x_j;
  x_i ← x_i + r/a_ii
end
```

Hier enthält $\mathbf{x}$ die aktuelle Näherungslösung, die mit der neuen Näherung überschrieben wird. Die Reihenfolge, in der die äußere Schleife durchlaufen wird, legt dabei implizit die Numerierung der Indexmenge fest.

In unserem Fall ist die Matrix $\mathbf{A}$ schwachbesetzt, in jeder Zeile sind also nur wenige Einträge ungleich null, so dass sich ein Schritt der Gauß-Seidel-Iteration in $\mathcal{O}(n)$ Operationen ausführen lässt.

Bemerkung 7.5 (Jacobi-Iteration) *Da für die Berechnung des Teilergebnisses $\mathbf{x}^{(m,i)}$ immer $\mathbf{x}^{(m,i-1)}$ benötigt wird, müssen die einzelnen Schritte der Gauß-Seidel-Iteration sequentiell ausgeführt werden, so dass eine effiziente Umsetzung auf einem Parallel- oder Vektorrechner Schwierigkeiten bereiten kann.*

Dieses Problem kann man lösen, indem man die Jacobi-Iteration *verwendet, die alle Korrekturterme ausgehend von $\mathbf{x}^{(m,0)} = \mathbf{x}^{(m)}$ berechnet:*

$$x_j^{(m+1)} := x_j^{(m)} + \theta a_{jj}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m)})_j \qquad \text{für alle } j \in \mathcal{I},$$

wobei $\theta \in \mathbb{R}_{>0}$ ein Dämpfungsfaktor ist, der benötigt wird, um die Konvergenz des Verfahrens sicher zu stellen. Die Implementierung dieses Verfahrens erfordert in der Regel zusätzlichen Speicherplatz, um beispielsweise den alten Vektor $\mathbf{x}^{(m)}$ aufzubewahren, der für die Berechnung aller Komponenten des neuen Vektors benötigt wird und deshalb nicht direkt überschrieben werden kann.

Bemerkung 7.6 (Unterraum-Korrektur) *Das Gauß-Seidel-Verfahren lässt sich allgemeiner interpretieren: Mit der Berechnung von α lösen wir das Variationsproblem*

$$a(\gamma\varphi_i, \alpha\varphi_i) = \beta(\gamma\varphi_i) - a(\gamma\varphi_i, \tilde{u}_h) \qquad \text{für alle } \gamma \in \mathbb{R},$$

auf dem von φ_i aufgespannten eindimensionalen Teilraum.

Wenn ein allgemeiner Teilraum $W_h \subseteq V_h$ gegeben ist, können wir entsprechend ein $w_h \in W_h$ mit

$$a(v_h, w_h) = \beta(v_h) - a(v_h, \tilde{u}_h) \qquad \text{für alle } v_h \in W_h$$

suchen und eine verbesserte Näherung $\tilde{u}'_h := \tilde{u}_h + w_h$ konstruieren. Man spricht dann von einer Unterraum-Korrektur.

Grobgitterkorrektur. Wenn wir die Gauß-Seidel- oder die Jacobi-Iteration auf das lineare Gleichungssystem anwenden, das aus der Variationsformulierung entstanden ist, werden wir feststellen, dass beide wesentlich langsamer als das bereits bekannte Verfahren der konjugierten Gradienten konvergieren. Die Ursache liegt darin, dass beide nur lokale Korrekturen durchführen: Im Schritt von $\mathbf{x}^{(m,i-1)}$ zu $\mathbf{x}^{(m,i)}$ werden nur die Elemente $x_j^{(m,i-1)}$ berücksichtigt, für die $a_{ij} \neq 0$ gilt. Bei linearen Basisfunktionen sind das gerade diejenigen Eckpunkte, die durch eine Kante unserer Triangulation mit i verbunden sind, also die unmittelbare Nachbarschaft dieses Punkts.

Man kann sich überlegen, dass aufgrund dieser Eigenschaft zwar hochfrequente Anteile des Fehlers $u_h - u_h^{(m)}$ schnell reduziert werden, dass niedrigfrequente Anteile allerdings nur sehr langsam gegen null konvergieren. Dadurch entsteht der Effekt, dass der Fehler nach einer Anzahl von Gauß-Seidel- oder Jacobi-Schritten zwar "glatt" ist, aber potentiell immer noch relativ groß. Wir bezeichnen Verfahren mit dieser Eigenschaft als *Glättungsverfahren.*

Wenn wir ein schnelles Verfahren erhalten wollen, sollten wir also nach einer Möglichkeit suchen, um glatte Fehler zu eliminieren.

Die Idee der *Grobgitterkorrektur* besteht darin, diese Fehler auf einer gröberen Triangulation desselben Gebiets anzunähern. Da sie glatt sind, sollte sich eine gute Näherung gewinnen lassen, die wir anschließend als Korrektur verwenden können.

Sei also $\mathcal{T}_H$ eine Triangulation des Gebiets Ω, und sei $\mathcal{T}_h$ eine Verfeinerung dieser Triangulation. Dann ist der Ansatzraum V_H zu $\mathcal{T}_H$ ein Teilraum des Ansatzraums V_h zu $\mathcal{T}_h$, denn jede Funktion, die auf der groben Triangulation stückweise linear ist, ist es offenbar auch auf der feinen.

Unser Ziel ist es, den Fehler $u_h - u_h^{(m)}$ auf der gröberen Triangulation zu approximieren, wir suchen also ein $u_H \in V_H$ mit

$$u_H \approx u_h - u_h^{(m)}.$$

Da uns u_h nicht zur Verfügung steht, müssen wir die Näherung u_H indirekt berechnen, indem wir ausnutzen, dass u_h die Lösung des Variationsproblems ist, dass also

$$a(v_h, u_h) = \beta(v_h) \qquad \text{für alle } v_h \in V_h$$

gilt. Es bietet sich an, u_H als Lösung eines Variationsproblems auf dem Raum V_H zu charakterisieren, nämlich durch

$$a(v_H, u_H) = a(v_H, u_h - u_h^{(m)}) \qquad \text{für alle } v_H \in V_H.$$

Wegen $V_H \subseteq V_h$ ist das äquivalent zu

$$a(v_H, u_H) = \beta(v_H) - a(v_H, u_h^{(m)}) \qquad \text{für alle } v_H \in V_H, \tag{7.9a}$$

und bei dieser Formulierung lässt sich die rechte Seite praktisch für jedes $v_H \in V_H$ auswerten, und das Variationsproblem lässt sich wie bisher als lineares Gleichungssystem schreiben und auflösen.

Mit der so gewonnenen Lösung $u_H \in V_H$ können wir nun eine Korrektur unserer Näherungslösung gewinnen, indem wir

$$u_h^{(m+1)} := u_h^{(m)} + u_H \tag{7.9b}$$

setzen. Es handelt sich insgesamt um eine der in Bemerkung 7.6 eingeführten Unterraumkorrekturen auf dem Raum V_H, der zu der gröberen Triangulation gehört.

Bei einer roten Verfeinerung einer zweidimensionalen Triangulation erhöht sich die Anzahl der Eckpunkte der Triangulation ungefähr um den Faktor 4, bei einer Bey-Verfeinerung einer dreidimensionalen Triangulation ist es der Faktor 8. Demzufolge dürfen wir davon ausgehen, dass das lineare Gleichungssystem auf der groben Triangulation wesentlich weniger Unbekannte aufweist als das auf der feinen und sich deshalb wesentlich schneller lösen lässt.

Prolongation. Wie schon im Fall des Gauß-Seidel-Verfahrens wollen wir auch die Grobgitterkorrektur durch Matrizen und Vektoren darstellen. Dabei tritt die Schwierigkeit auf, dass in (7.9) Funktionen aus V_H gemischt mit Funktionen aus V_h auftreten. Als Lösung führen wir eine Matrix ein, die zwischen beiden Räumen vermittelt: Seien $(\varphi_{H,i})_{i\in\mathcal{I}_H}$ und $(\varphi_{h,i})_{i\in\mathcal{I}_h}$ die Basen der beiden Räume, und seien mit

$$P_H : \mathbb{R}^{\mathcal{I}_H} \to V_H, \qquad \mathbf{y}_H \mapsto \sum_{i\in\mathcal{I}_H} y_{H,i}\varphi_{H,i},$$

$$P_h : \mathbb{R}^{\mathcal{I}_h} \to V_h, \qquad \mathbf{y}_h \mapsto \sum_{i\in\mathcal{I}_h} y_{h,i}\varphi_{h,i},$$

die zugehörigen Basis-Isomorphismen bezeichnet. Da $V_H \subseteq V_h$ gilt, können wir die Matrix

$$\mathbf{p} := P_h^{-1} P_H \in \mathbb{R}^{\mathcal{I}_h \times \mathcal{I}_H}$$

definieren, die die Eigenschaft

$$P_H = P_h \mathbf{p} \tag{7.10}$$

besitzt, die also gerade die Koeffizienten einer Funktion $v_H \in V_H$ bezüglich der Basis $(\varphi_{H,i})_{i\in\mathcal{I}_H}$ in die Koeffizienten in der Basis $(\varphi_{h,i})_{i\in\mathcal{I}_h}$ umrechnet. Diese Matrix nennt man

im allgemeinen Fall *Prolongation*, in unserem Fall auch *Interpolation*, weil sie gerade der linearen Interpolation der Ansatzfunktionen auf dem groben Gitter entspricht.

Wir bezeichnen mit $\mathbf{A}_H \in \mathbb{R}^{\mathcal{I}_H \times \mathcal{I}_H}$ und $\mathbf{A}_h \in \mathbb{R}^{\mathcal{I}_h \times \mathcal{I}_h}$ wie üblich die Matrizen mit

$$\begin{aligned} a_{H,ij} &= a(\varphi_{H,i}, \varphi_{H,j}) && \text{für alle } i,j \in \mathcal{I}_H, \\ a_{h,ij} &= a(\varphi_{h,i}, \varphi_{h,j}) && \text{für alle } i,j \in \mathcal{I}_h \end{aligned}$$

und mit $\mathbf{b}_h \in \mathbb{R}^{\mathcal{I}_h}$ den Vektor

$$b_{h,i} = \beta(\varphi_{h,i}) \qquad \text{für alle } i \in \mathcal{I}_h.$$

Nun können wir (7.9a) in Matrixnotation überführen: Wir setzen

$$v_H = P_H \mathbf{y}_H, \qquad u_H = P_H \mathbf{x}_H, \qquad u_h^{(m)} = P_h \mathbf{x}^{(m)}$$

und erhalten

$$\begin{aligned} \langle \mathbf{y}_H, \mathbf{A}_H \mathbf{x}_H \rangle_2 &= \sum_{i \in \mathcal{I}_H} \sum_{j \in \mathcal{I}_H} y_{H,i} a_{H,ij} x_{H,j} = \sum_{i \in \mathcal{I}_H} \sum_{j \in \mathcal{I}_H} a(y_{H,i}\varphi_{H,i}, x_{H,j}\varphi_{H,j}) \\ &= a(P_H \mathbf{y}_H, P_H \mathbf{x}_H) = a(v_H, u_H) = \beta(v_H) - a(v_H, u_h^{(m)}) \\ &= \beta(P_H \mathbf{y}_H) - a(P_H \mathbf{y}_H, P_h x_h^{(m)}) = \beta(P_h \mathbf{p}\mathbf{y}_H) - a(P_h \mathbf{p}\mathbf{y}_H, P_h x_h^{(m)}) \\ &= \sum_{i \in \mathcal{I}_h} (\mathbf{p}\mathbf{y}_H)_i \beta(\varphi_{h,i}) - \sum_{i \in \mathcal{I}_h} \sum_{j \in \mathcal{I}_h} (\mathbf{p}\mathbf{y}_H)_i a_{h,ij} x_j^{(m)} \\ &= \sum_{i \in \mathcal{I}_h} (\mathbf{p}\mathbf{y}_H)_i \beta(\varphi_{h,i}) - \sum_{i \in \mathcal{I}_h} (\mathbf{p}\mathbf{y}_H)_i (\mathbf{A}_h \mathbf{x}^{(m)})_i \\ &= \langle \mathbf{p}\mathbf{y}_H, \mathbf{b}_h \rangle - \langle \mathbf{p}\mathbf{y}_H, \mathbf{A}_h \mathbf{x}^{(m)} \rangle_2 = \langle \mathbf{p}\mathbf{y}_H, \mathbf{b}_h - \mathbf{A}_h \mathbf{x}^{(m)} \rangle_2 \\ &= \langle \mathbf{y}_H, \mathbf{p}^*(\mathbf{b}_h - \mathbf{A}_h \mathbf{x}^{(m)}) \rangle_2. \end{aligned}$$

Da diese Gleichung für alle $\mathbf{y}_H \in \mathbb{R}^{\mathcal{I}_H}$ gelten muss, folgt

$$\mathbf{A}_H \mathbf{x}_H = \mathbf{p}^*(\mathbf{b}_h - \mathbf{A}_h \mathbf{x}^{(m)}).$$

Die rechte Seite dieses linearen Gleichungssystem lässt sich mit Hilfe der Matrix $\mathbf{p}$ explizit berechnen, durch das Auflösen des Systems erhalten wir $\mathbf{x}_H$.

Die Korrektur (7.9b) nimmt die Gestalt

$$\begin{aligned} P_h \mathbf{x}^{(m+1)} &= u_h^{(m+1)} = u_h^{(m)} + u_H = P_h \mathbf{x}^{(m)} + P_H \mathbf{x}_H \\ &= P_h \mathbf{x}^{(m)} + P_h \mathbf{p}\mathbf{x}_H = P_h(\mathbf{x}^{(m)} + \mathbf{p}\mathbf{x}_H) \end{aligned}$$

an, so dass sich

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \mathbf{p}\mathbf{x}_H$$

ergibt. Wir können das Ergebnis unserer Bemühungen wie folgt zusammenfassen:

Definition 7.7 (Grobgitterkorrektur) *Die Abbildung*

$$(\mathbf{x}_h^{(m)}, \mathbf{b}_h) \mapsto \mathbf{x}_h^{(m)} + \mathbf{p}\mathbf{A}_H^{-1}\mathbf{p}^*(\mathbf{b}_h - \mathbf{A}_h \mathbf{x}^{(m)})$$

bezeichnen wir als die Grobgitterkorrektur *zu der Näherungslösung* $\mathbf{x}^{(m)}$ *und der rechten Seite* $\mathbf{b}_h$.

Restriktion. Die Matrix $\mathbf{p}$ übersetzt Koeffizientenvektoren einer Funktion bezüglich der Basis $(\varphi_{H,i})_{i\in\mathcal{I}_H}$ in Koeffizientenvektoren derselben Funktion bezüglich der Basis $(\varphi_{h,i})_{i\in\mathcal{I}_h}$.

Die Matrix $\mathbf{r} := \mathbf{p}^*$, die im Kontext der Grobgitterkorrektur auftritt, spielt eine andere Rolle: Der Vektor $\mathbf{b}_h$ beispielsweise ist kein Koeffizientenvektor einer Funktion, vielmehr ist er gemäß

$$b_{h,i} = \beta(\varphi_{h,i}) \qquad \text{für alle } i \in \mathcal{I}_h$$

durch das Einsetzen der Basisfunktionen $\varphi_{h,i}$ in das Funktional $\beta \in V'$ gegeben. Der Vektor

$$\mathbf{b}_H := \mathbf{r}\mathbf{b}_h = \mathbf{p}^*\mathbf{b}_h \in \mathbb{R}^{\mathcal{I}_H}$$

entspricht dem Einsetzen der Basisfunktionen $\varphi_{H,i}$ in dasselbe Funktional.

Die *Restriktionsmatrix* $\mathbf{r} \in \mathbb{R}^{\mathcal{I}_H\times\mathcal{I}_h}$ hat also die Aufgabe, aus den Werten eines Funktionals für die Basis $(\varphi_{h,i})_{i\in\mathcal{I}_h}$ die Werte desselben Funktionals für die Basis $(\varphi_{H,i})_{i\in\mathcal{I}_H}$ zu konstruieren.

Kurz gesagt überführt die Prolongationsmatrix Funktionen aus V_H in Funktionen aus V_h, während die Restriktionsmatrix Funktionale aus V_h' in Funktionale aus V_H' überführt.

Da die Restriktionsmatrix im Kontext der Galerkin-Diskretisierung immer die Transponierte der Prolongationsmatrix ist, brauchen wir sie in der Regel nicht gesondert abzuspeichern oder zu berechnen.

Übungsaufgabe 7.8 (Galerkin-Eigenschaft) *Beweisen Sie, dass unsere Matrizen die* Galerkin-Eigenschaft *erfüllen, dass nämlich die Gleichung*

$$\mathbf{A}_H = \mathbf{r}\mathbf{A}_h\mathbf{p}$$

gilt. Diese Gleichung wird in manchen Implementierungen benutzt, um die Matrix $\mathbf{A}_H$ aus $\mathbf{A}_h$ zu konstruieren, sie ist aber auch für die Suche nach Programmfehlern nützlich.

Übungsaufgabe 7.9 (Projektion) *Wir bezeichnen mit*

$$\langle \mathbf{y}_h, \mathbf{x}_h\rangle_A := \langle \mathbf{y}_h, \mathbf{A}_h\mathbf{x}_h\rangle_2 \qquad \textit{für alle } \mathbf{x}_h, \mathbf{y}_h \in \mathbb{R}^{\mathcal{I}_h}$$

das Energie-Skalarprodukt *und mit $\|\mathbf{x}_h\|_A := \sqrt{\langle \mathbf{x}_h, \mathbf{x}_h\rangle_A}$ die zugehörige* Energienorm. *Beweisen Sie, dass die Grobgitterkorrektur*

$$\mathbf{G} := \mathbf{I} - \mathbf{p}\mathbf{A}_H^{-1}\mathbf{r}\mathbf{A}_h$$

die Gleichung

$$\langle \mathbf{p}\mathbf{y}_H, \mathbf{G}\mathbf{d}_h\rangle_A = 0 \qquad \textit{für alle } \mathbf{d}_h \in \mathbb{R}^{\mathcal{I}_h},\ \mathbf{y}_H \in \mathbb{R}^{\mathcal{I}_H}$$

erfüllt. Folgern Sie daraus, dass $\|\mathbf{G}\mathbf{d}_h\|_A \le \|\mathbf{d}_h - \mathbf{p}\mathbf{y}_H\|_A$ für alle $\mathbf{d}_h \in \mathbb{R}^{\mathcal{I}_h}$ und $\mathbf{y}_H \in \mathbb{R}^{\mathcal{I}_H}$ gilt, dass also die Grobgitterkorrektur das Residuum $\mathbf{d}_h$ so weit reduziert, wie es durch Subtraktion eines Elements des Raums V_H möglich ist.

Zweigitterverfahren. Mit der Gauß-Seidel- und der Jacobi-Iteration verfügen wir über Glättungsverfahren, die binnen weniger Iterationsschritte dafür sorgen, dass der verbliebene Fehler glatt ist. Wir dürfen erwarten, dass sich solche Fehler auf einer gröberen Triangulation gut approximieren lassen und die Grobgitterkorrektur sie deshalb effizient reduzieren wird. Also liegt es nahe, beide Iterationen zu kombinieren, um schnelle Konvergenz zu erreichen.

Das Ergebnis wird als *Zweigitterverfahren* bezeichnet und lässt sich wie folgt zusammenfassen:

procedure zgv($\mathbf{A}_h$, $\mathbf{b}_h$, $\mathbf{A}_H$, $\mathbf{p}$, **var** $\mathbf{x}_h$);
for $k = 1$ **to** ν **do** gauss_seidel($\mathbf{A}_h$, $\mathbf{b}_h$, $\mathbf{x}_h$);
$\mathbf{d}_h \leftarrow \mathbf{b}_h - \mathbf{A}_h\mathbf{x}_h$;
$\mathbf{b}_H \leftarrow \mathbf{r}\mathbf{d}_h$;
Löse $\mathbf{A}_H\mathbf{x}_H = \mathbf{b}_H$;
$\mathbf{x}_h \leftarrow \mathbf{x}_h + \mathbf{p}\mathbf{x}_H$

Hier gibt $\nu \in \mathbb{N}$ an, wieviele Schritte des Glättungsverfahrens durchgeführt werden sollen, bevor wir den Fehler für glatt genug erachten, um ihn auf dem gröberen Gitter darstellen zu können. Es lässt sich beweisen, dass wir durch eine geeignete Wahl dieses Parameters ein Verfahren erhalten, das *beliebig schnell* konvergiert. Allerdings bedeutet ein großes ν natürlich auch einen hohen Rechenaufwand, weil viele Schritte des Glättungsverfahrens anfallen.

Aufstellen der Prolongationsmatrix. Die Prolongationsmatrix $\mathbf{p} \in \mathbb{R}^{\mathcal{I}_h \times \mathcal{I}_H}$ haben wir zwar schon definiert, aber unsere Definition ist wenig praxistauglich, solange wir keinen effizienten Weg finden, um P_h^{-1} auszuwerten. Diese Aufgabe lässt sich in unserem Modellproblem sehr einfach lösen: Die von uns verwendete Basis $(\varphi_{h,i})_{i\in\mathcal{I}}$ ist einen *Knotenbasis*, es gilt nämlich nach Definition

$$\varphi_{h,i}(j) = \begin{cases} 1 & \text{falls } j = i, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i, j \in \mathcal{I}_h,$$

wobei wir ausnutzen, dass $\mathcal{I}$ gerade die Menge der inneren Eckpunkte der Triangulation ist, also insbesondere aus geometrischen Punkten in dem Gebiet Ω besteht.

Aus dieser Eigenschaft folgt unmittelbar

$$v_h = \sum_{i\in\mathcal{I}} v_h(i)\varphi_{h,i} \qquad \text{für alle } v_h \in V_h, \tag{7.11}$$

denn die Gültigkeit dieser Gleichung lässt sich in den Eckpunkten einfach durch Einsetzen überprüfen und überträgt sich dann auf die gesamte Funktion, da stückweise lineare Funktionen durch ihre Werte in den Eckpunkten bereits eindeutig festgelegt sind.

Wenn wir einen Vektor $\mathbf{y}_h \in \mathbb{R}^{\mathcal{I}_h}$ durch

$$y_{h,i} := v_h(i) \qquad \text{für alle } i \in \mathcal{I}_h$$

definieren, nimmt die Gleichung (7.11) die Gestalt

$$v_h = \sum_{i \in \mathcal{I}} y_{h,i} \varphi_{h,i} = P_h \mathbf{y}_h$$

an, also gilt gerade $\mathbf{y}_h = P_h^{-1} v_h$. Die Umkehrfunktion des Basis-Isomorphismus lässt sich demnach gewinnen, indem wir die gegebene Funktion in den Gitterpunkten auswerten.

Für die Berechnung der Prolongationsmatrix ergibt sich damit

$$p_{ij} = (P_h^{-1} \varphi_{H,j})_i = \varphi_{H,j}(i) \qquad \text{für alle } i \in \mathcal{I}_h,\ j \in \mathcal{I}_H.$$

Wir brauchen also lediglich für jeden Eckpunkt $i \in \mathcal{I}_h$ der feinen Triangulation ein Element $t \in \mathcal{T}_H$ der groben Triangulation zu finden, das ihn enthält, und können dann die zu diesem Element gehörenden drei oder vier Basisfunktionen $\varphi_{H,j}$, $j \in t$, in diesem Punkt auswerten, um die von null verschiedenen Einträge der i-ten Zeile der Prolongationsmatrix zu berechnen.

Falls die feine Triangulation $\mathcal{T}_h$ durch eine der von uns in Abschnitt 7.1 diskutierten Verfeinerungsstrategien aus $\mathcal{T}_H$ entstanden ist, wird die Berechnung sogar noch wesentlich einfacher: In diesem Fall kann $i \in \mathcal{I}_h$ nur ein Eckpunkt der Triangulation $\mathcal{T}_H$ oder ein Kantenmittelpunkt sein, so dass sich

$$p_{ij} = \begin{cases} 1 & \text{falls } i = j, \\ 1/2 & \text{falls } i \text{ auf einer Kante mit Endpunkt } j, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i \in \mathcal{I}_h,\ j \in \mathcal{I}_H$$

ergibt und sich die Prolongationsmatrix sehr einfach und effizient aus Informationen über die Verfeinerung konstruieren lässt. Eine Zeile der Matrix $\mathbf{p}$ enthält dann nur einen oder zwei von null verschiedene Einträge.

Mehrgitterverfahren. Um das Zweigitterverfahren durchführen zu können, müssen wir dazu in der Lage sein, das Gleichungssystem $\mathbf{A}_H \mathbf{x}_H = \mathbf{b}_H$ auf der groben Triangulation zu lösen. Die grobe Triangulation darf dabei nicht zu grob sein, denn wir sind darauf angewiesen, dass sich der nach wenigen Schritten des Glättungsverfahrens verbliebene Fehler auf dieser Triangulation noch hinreichend gut darstellen lässt.

Dadurch entsteht das Problem, dass bei einer sehr feinen Triangulation $\mathcal{T}_h$ auch die grobe Triangulation $\mathcal{T}_H$ noch sehr viele Punkte aufweist, so dass auch das zu dieser Triangulation gehörende Gleichungssystem noch sehr groß ist. Deshalb wäre es wünschenswert, das direkte Lösen dieses Gleichungssystems ebenfalls zu vermeiden.

Die Idee des *Mehrgitterverfahrens* besteht darin, einfach auch auf der groben Triangulation zu glätten und eine Korrektur auf einer noch gröberen Triangulation durchzuführen. In dieser Weise können wir rekursiv fortschreiten, bis wir eine Triangulation erreicht haben, die nur noch sehr wenige Punkte enthält, so dass wir sie direkt behandeln können.

Definition 7.10 (Gitterhierarchie) *Sei $(\mathcal{T}_\ell)_{\ell=0}^L$ eine Familie von Triangulationen derart, dass für alle $\ell \in \{1, \dots, L\}$ jeweils $\mathcal{T}_\ell$ eine Verfeinerung der Triangulation $\mathcal{T}_{\ell-1}$ ist. Dann bezeichnen wir $(\mathcal{T}_\ell)_{\ell=0}^L$ als* Gitterhierarchie.

Im Folgenden gehen wir davon aus, dass uns eine Gitterhierarchie $(\mathcal{T}_\ell)_{\ell=0}^L$ zur Verfügung steht. Wir bezeichnen mit $V_\ell \subseteq V$ den zu der Triangulation $\mathbf{T}_\ell$ gehörenden Ansatzraum und mit $(\varphi_{\ell,i})_{i\in\mathcal{I}_\ell}$ die entsprechende Basis.

Die dieser Basis zugeordnete Matrix nennen wir $\mathbf{A}_\ell \in \mathbb{R}^{\mathcal{I}_\ell\times\mathcal{I}_\ell}$.

Den Austausch von Informationen zwischen dem Raum V_ℓ und dem Raum $V_{\ell-1}$ bewerkstelligen wir mit Prolongationsmatrizen

$$\mathbf{p}_\ell := P_\ell^{-1} P_{\ell-1} \in \mathbb{R}^{\mathcal{I}_\ell\times\mathcal{I}_{\ell-1}} \qquad \text{für alle } \ell \in \{1,\dots,L\},$$

die Funktionen aus $V_{\ell-1}$ in Funktionen in V_ℓ überführen, und Restriktionsmatrizen

$$\mathbf{r}_\ell := \mathbf{p}_\ell^* \qquad \text{für alle } \ell \in \{1,\dots,L\},$$

die Funktionale aus V_ℓ' in Funktionale in $V_{\ell-1}'$ verwandeln.

Das Mehrgitterverfahren können wir dann wie folgt als rekursiven Algorithmus formulieren:

procedure mgv(ℓ);
if $\ell = 0$ **then**
  Löse $\mathbf{A}_\ell\mathbf{x}_\ell = \mathbf{b}_\ell$
else begin
  for $k = 1$ **to** ν_1 **do** gauss_seidel($\mathbf{A}_\ell$, $\mathbf{b}_\ell$, $\mathbf{x}_\ell$);
  $\mathbf{d}_\ell \leftarrow \mathbf{b}_\ell - \mathbf{A}_\ell\mathbf{x}_\ell$;
  $\mathbf{b}_{\ell-1} \leftarrow \mathbf{r}_\ell\mathbf{d}_\ell$;
  $\mathbf{x}_{\ell-1} \leftarrow \mathbf{0}$;
  for $k = 1$ **to** γ **do** mgv($\ell - 1$);
  $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell + \mathbf{p}_\ell\mathbf{x}_{\ell-1}$;
  for $k = 1$ **to** ν_2 **do** gauss_seidel($\mathbf{A}_\ell$, $\mathbf{b}_\ell$, $\mathbf{x}_\ell$);
end

Falls wir auf der gröbsten Triangulation $\mathcal{T}_0$ angekommen sind, wird direkt gelöst. Anderenfalls werden ν_1 Schritte des Glättungsverfahrens durchgeführt, man spricht von einer *Vorglättung*. Es folgen der Transfer des Residuums $\mathbf{d}_\ell$ auf die gröbere Triangulation $\mathcal{T}_{\ell-1}$ sowie γ rekursive Aufrufe des Mehrgitterverfahrens, mit denen eine Näherung $\mathbf{x}_{\ell-1}$ des Fehlers konstruiert wird. Da wir hoffen, dass der Fehler klein sein wird, ist $\mathbf{x}_{\ell-1} = \mathbf{0}$ ein akzeptabler Anfangswert für diesen Teil des Verfahrens. Anschließend wird die genäherte Grobgitterkorrektur zu der aktuellen Näherung $\mathbf{x}_\ell$ addiert. Da in diesem Schritt neue hochfrequente Störungen hinzugefügt worden sein könnten, führen wir ν_2 Schritte des Glättungsverfahrens durch, insgesamt als *Nachglättung* bezeichnet.

Rechenaufwand V-Zyklus. Besonders effizient ist das Mehrgitterverfahren natürlich, wenn nur ein einziger rekursiver Aufruf auf jeder Stufe der Gitterhierarchie erforderlich ist, wenn also $\gamma = 1$ in dem obigen Algorithmus verwendet wird. In Abbildung 7.8 sind die für eine Gitterhierarchie mit $L = 4$ anfallenden Rechenschritte dargestellt. Ihrer charakteristischen Abfolge auf den verschiedenen Stufen der Hierarchie verdankt dieser Sonderfall den Namen *V-Zyklus-Mehrgitterverfahren*.

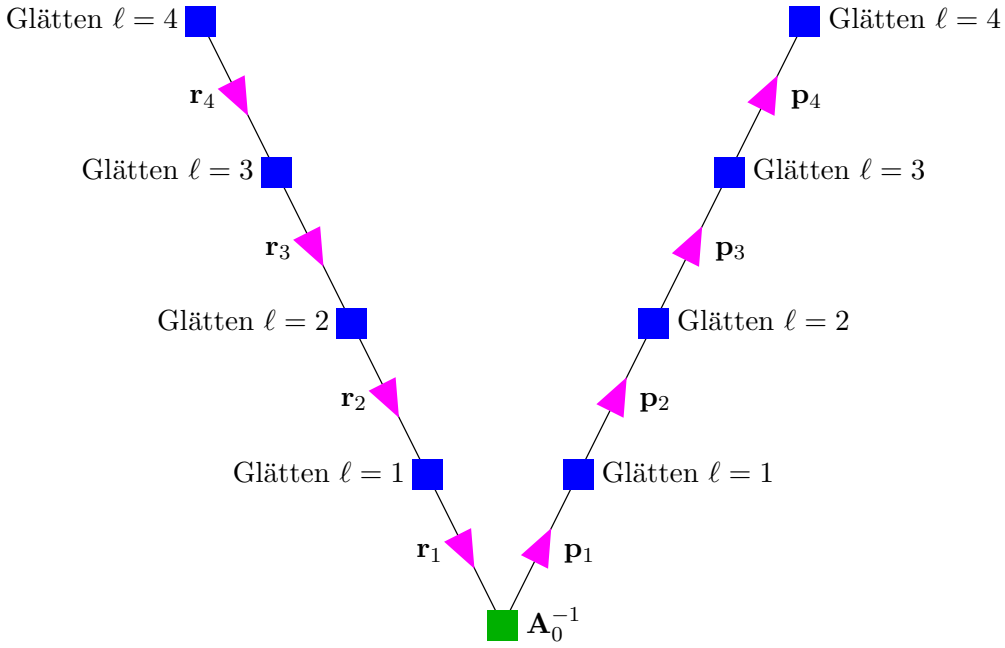

Abbildung 7.8: V-Zyklus-Mehrgitterverfahren

Unser Ziel ist es, den Rechenaufwand zu analysieren. Dazu bezeichnen wir mit $n_\ell := \#\mathcal{I}_\ell$ die Anzahl der Indizes auf der ℓ-ten Stufe der Hierarchie. Da das Gauß-Seidel-Verfahren bei einer schwachbesetzten Matrix nur wenige Rechenoperationen pro Index ausführt, dürfen wir davon ausgehen, dass eine Konstante C_{gl} so existiert, dass ein Schritt des Glättungsverfahrens auf Stufe ℓ nicht mehr als $C_{\mathrm{gl}} n_\ell$ Operationen erfordert. Der Einfachheit halber können wir annehmen, dass diese Konstante groß genug ist, um auch den Aufwand für das Lösen auf der gröbsten Stufe in $C_{\mathrm{gl}} n_0$ Operationen bewerkstelligen zu können.

In unserem Fall enthält jede Zeile der Prolongationsmatrizen $\mathbf{p}_\ell$ höchstens zwei von null verschiedene Einträge, so dass wir davon ausgehen dürfen, dass Prolongation und Restriktion zwischen den Stufen ℓ und $\ell - 1$ jeweils nicht mehr als $C_{\mathrm{pr}} n_\ell$ Operationen erfordern. Damit benötigt der gesamte Mehrgitterschritt nicht mehr als

$$\begin{aligned}\sum_{\ell=1}^{L} C_{\mathrm{gl}}(\nu_1+\nu_2)n_\ell + C_{\mathrm{gl}} n_0 + \sum_{\ell=1}^{L} 2C_{\mathrm{pr}} n_\ell &\leq \sum_{\ell=0}^{L} (C_{\mathrm{gl}}(\nu_1+\nu_2) + 2C_{\mathrm{pr}})n_\ell \\ &= (C_{\mathrm{gl}}(\nu_1+\nu_2) + 2C_{\mathrm{pr}}) \sum_{\ell=0}^{L} n_\ell\end{aligned}$$

Operationen. Wir sind daran interessiert, nachzuweisen, dass sich der Gesamtaufwand proportional zu n_L verhält. Unsere Verfeinerungsstrategien legen es nahe, anzunehmen, dass eine Konstante C_{hi} so existiert, dass

$$C_{\mathrm{hi}} 2^{d(\ell-1)} \leq n_\ell \leq C_{\mathrm{hi}} 2^{d\ell} \qquad \text{für alle } \ell \in \{0, \dots, L\}$$

existiert, denn jede Verfeinerung führt im zweidimensionalen Fall zu einer Vervierfachung der Dreiecke und im dreidimensionalen zu einer Verachtfachung der Tetraeder. Mit dieser Annahme gelangen wir mit der geometrischen Summenformel zu der Schranke

$$\begin{aligned}
(C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}) \sum_{\ell=0}^{L} n_\ell &\leq (C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}) \sum_{\ell=0}^{L} C_{\mathrm{hi}} 2^{d\ell} \\
&= (C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}) C_{\mathrm{hi}} \frac{2^{d(L+1)}-1}{2^d-1} \\
&\leq \frac{2^{2d}}{2^d-1} (C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}) C_{\mathrm{hi}} 2^{d(L-1)} \\
&\leq \frac{2^{2d}}{2^d-1} (C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}) n_L \\
&\leq C_{\mathrm{mg}} n_L
\end{aligned}$$

mit der Konstanten

$$C_{\mathrm{mg}} := \frac{4^d}{2^d-1} (C_{\mathrm{gl}}(\nu_1+\nu_2)+2C_{\mathrm{pr}}).$$

Damit haben wir gezeigt, dass der Aufwand sich durch eine Funktion beschränken lässt, die proportional zu der Anzahl der Unbekannten wächst. Der Rechenaufwand des Mehrgitterverfahrens ist von *optimaler* Ordnung.

Geschachtelte Iteration. Falls die Lösung $u \in V$ der ursprünglichen Variationsaufgabe (6.25) hinreichend gutartig ist, lässt sich beweisen, dass der Diskretisierungsfehler sich wie

$$\|u - u_\ell\|_V \leq C_{\mathrm{di}} n_\ell^{-1/d} \qquad \text{für alle } \ell \in \{0, \dots, L\} \tag{7.12}$$

verhält, wobei $u_\ell \in V_\ell$ die Näherung der Galerkin-Diskretisierung mit dem Raum V_ℓ bezeichnet. Mit einer Verfeinerung der Triangulation geht also auch eine höhere Genauigkeit der Lösung einher.

Wir approximieren u_ℓ, indem wir eine Anfangsnäherung $u_\ell^{(0)} \in V_\ell$ wählen und m Schritte ausführen, um eine Näherung $u_\ell^{(m)} \in V_\ell$ zu finden. Unser Ziel ist es, sicher zu stellen, dass $u_\ell^{(m)}$ ähnlich genau wie u_ℓ ist, dass also beispielsweise

$$\|u - u_\ell^{(m)}\|_V \leq 2C_{\mathrm{di}} n_\ell^{-1/d}$$

gilt. Jeder Schritt unseres Mehrgitterverfahrens reduziert den Fehler mindestens um einen Faktor ϱ, so dass

$$\|u_\ell - u_\ell^{(m)}\|_V \leq \varrho^m \|u_\ell - u_\ell^{(0)}\|_V$$

gilt. Um unser Ziel zu erreichen sollten wir

$$\begin{aligned}\|u - u_\ell^{(m)}\|_V &= \|u - u_\ell + u_\ell - u_\ell^{(m)}\|_V \le \|u - u_\ell\|_V + \|u_\ell - u_\ell^{(m)}\|_V \\ &\le C_{\mathrm{di}} n_\ell^{-1/d} + \varrho^m \|u_\ell - u_\ell^{(0)}\|_V \overset{!}{\le} 2C_{\mathrm{di}} n_\ell^{-1/d}\end{aligned}$$

sicherstellen, also gerade

$$\varrho^m \|u - u_\ell^{(0)}\|_V \le C_{\mathrm{di}} n_\ell^{-1/d}.$$

Wenn wir $\varrho = e^{-\alpha}$ mit $\alpha > 0$ setzen und den natürlichen Logarithmus der Ungleichung betrachten folgt

$$\begin{aligned}e^{-\alpha m}\|u - u_\ell^{(0)}\|_V &\le C_{\mathrm{di}} n_\ell^{-1/d}, \\ -\alpha m + \ln(\|u - u_\ell^{(0)}\|_V) &\le \ln(C_{\mathrm{di}}) - \frac{1}{d}\ln(n_\ell), \\ m &\ge \frac{1}{\alpha}(-\ln(C_{\mathrm{di}}) + \ln(n_\ell)/d + \ln(\|u - u_\ell^{(0)}\|_V))\end{aligned}$$

Würden wir beispielsweise mit $u_\ell^{(0)} = 0$ anfangen, müsste m proportional zu $\ln(n_\ell)$ wachsen, damit $u_\ell^{(m)}$ die geforderte Genauigkeit erreicht. Der Rechenaufwand pro Stufe würde als doch wieder von der Anzahl der auf dieser Stufe vorhandenen Unbekannten abhängen, wenn auch nur logarithmisch.

Dieses Problem lässt sich lösen, indem wir den Anfangsvektor geschickter wählen: Wir berechnen zunächst $u_{\ell-1}^{(m)} \in V_{\ell-1}$ auf der nächstgröberen Stufe $\ell-1$ so genau, dass unsere Bedingung

$$\|u - u_{\ell-1}^{(m)}\|_V \le 2C_{\mathrm{di}} n_{\ell-1}^{-1/d}$$

erfüllt ist. Wenn wir der Einfachheit halber $2^d n_{\ell-1} \ge n_\ell$ annehmen, folgt $n_{\ell-1}^{-1/d} \le 2n_\ell^{-1/d}$, so dass wir

$$\|u - u_{\ell-1}^{(m)}\|_V \le 4C_{\mathrm{di}} n_\ell^{-1/d}$$

erhalten. Wenn wir nun $u_{\ell-1}^{(m)}$ als Anfangsnäherung auf der Gitterstufe ℓ verwenden, also $u_\ell^{(0)} := u_{\ell-1}^{(m)}$ setzen, ergibt sich

$$m \ge \frac{1}{\alpha}(-\ln(C_{\mathrm{di}}) + \ln(n_\ell)/d + \ln(4) + \ln(C_{\mathrm{di}}) - \ln(n_\ell)/d) = \frac{1}{\alpha}\ln(4).$$

Also genügt bei dieser Wahl der Anfangsnäherung eine konstante Anzahl von Schritten.

Der resultierende Algorithmus ist als *geschachtelte Iteration* (im Englischen auch als *full multigrid*) bekannt:

procedure nested_iteration;
for $\ell = L$ **downto** 1 **do** $\mathbf{b}_{\ell-1} \leftarrow \mathbf{r}_\ell \mathbf{b}_\ell$;
Löse $\mathbf{A}_0\mathbf{x}_0 = \mathbf{b}_0$;
for $\ell = 1$ **to** L **do begin**
  $\mathbf{x}_\ell \leftarrow \mathbf{p}_\ell \mathbf{x}_{\ell-1}$;
  for $k = 1$ **to** κ **do** mgv(ℓ);
end

In der ersten Schleife werden rechte Seiten für alle Stufen der Hierarchie mit Hilfe der Restriktionsmatrix berechnet. Da jeweils $V_{\ell-1} \subseteq V_\ell$ gilt, entstehen dabei dieselben Vektoren, die auch bei einer direkten Diskretisierung entstanden wären, man vermeidet aber die potentiell aufwendige Quadratur auf allen Stufen außer der feinsten.

Anschließend wird auf der gröbsten Stufe der Hierarchie gelöst. Es folgt der Kern des Algorithmus: Das Ergebnis der jeweils vorangehenden Stufe wird mit Hilfe der Prolongationsmatrix auf die aktuelle Stufe übertragen und als Anfangsvektor verwendet, der dann mit κ Mehrgitterschritten verbessert wird, bis er die gewünschte Genauigkeit erreicht.

7.4 Konvergenz symmetrischer Iterationsverfahren

Um die Qualität eines iterativen Lösungsverfahrens für ein lineares Gleichungssystem beurteilen zu können, ist es von großer Bedeutung, Aussagen über dessen Konvergenzrate zu treffen, also über den Faktor, um den ein Schritt des Verfahrens den Fehler reduziert. Einige für die Analyse der Konvergenzrate wichtige Techniken sollen in diesem Abschnitt vorgestellt werden. Die Darstellung folgt dabei im Wesentlichen dem Buch [4].

Wir untersuchen Iterationsverfahren, die das lineare Gleichungssystem

$$\mathbf{A}\mathbf{x}^* = \mathbf{b} \tag{7.13}$$

lösen sollen, wobei $\mathbf{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine positiv definite und selbstadjungierte Matrix und $\mathbf{b} \in \mathbb{R}^{\mathcal{I}}$ ein beliebiger Vektor sein soll. Die Lösung bezeichnen wir mit $\mathbf{x}^*$, da wir sie als Grenzwert einer Folge $\mathbf{x}^0, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$ von Näherungslösungen berechnen werden.

Wir beschränken uns dabei auf *symmetrische lineare Iterationsverfahren*. Ein typisches Beispiel eines solchen Verfahrens ist die in Bemerkung 7.5 vorgestellte Jacobi-Iteration, bei der die neue Iterierte $\mathbf{x}^{(m+1)}$ aus der alten Iterierten $\mathbf{x}^{(m)}$ durch

$$x_j^{(m+1)} = x_j^{(m)} + \frac{\theta}{a_{jj}}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m)})_j \qquad \text{für alle } j \in \mathcal{I}$$

hervorgeht. Wenn wir mit $\mathbf{W} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ die mit $1/\theta$ skalierte Diagonale der Matrix $\mathbf{A}$ bezeichnen, die durch

$$w_{ij} := \begin{cases} a_{ii}/\theta & \text{falls } i = j, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i, j \in \mathcal{I}$$

gegeben ist, lässt sich die Jacobi-Iteration kompakt in der Form

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \mathbf{W}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m)}) \qquad \text{für alle } m \in \mathbb{N}_0 \tag{7.14}$$

schreiben. Man kann beweisen, dass jedes lineare Iterationsverfahren, das die exakte Lösung des linearen Gleichungssystem als Fixpunkt besitzt, mit einer geeigneten Matrix $\mathbf{W}$ in dieser Form geschrieben werden kann.

Definition 7.11 (Lineares Iterationsverfahren) *Sei $\Phi : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}}$ eine Abbildung. Wir nennen Φ ein* lineares Iterationsverfahrens*, falls eine invertierbare Matrix $\mathbf{W} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ mit*

$$\Phi(\mathbf{x}) = \mathbf{x} + \mathbf{W}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}) \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}} \tag{7.15}$$

existiert. Jeder Anfangsvektor $\mathbf{x}^{(0)} \in \mathbb{R}^{\mathcal{I}}$ definiert eine Folge von Iterierten $(\mathbf{x}^{(m)})_{m=0}^{\infty}$ *durch*

$$\mathbf{x}^{(m+1)} := \Phi(\mathbf{x}^{(m)}) \qquad \text{für alle } m \in \mathbb{N}_0.$$

Lemma 7.12 (Eindeutigkeit) *Falls Φ ein lineares Iterationsverfahren ist, gibt es genau eine Matrix $\mathbf{W}$, die (7.15) erfüllt.*

Beweis. Seien $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ invertierbare Matrizen mit

$$\begin{aligned} \Phi(\mathbf{x}) &= \mathbf{x} + \mathbf{W}_1^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}), \\ \Phi(\mathbf{x}) &= \mathbf{x} + \mathbf{W}_2^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}) \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}}. \end{aligned}$$

Subtraktion der Gleichungen ergibt

$$\mathbf{0} = (\mathbf{W}_1^{-1} - \mathbf{W}_2^{-1})(\mathbf{b} - \mathbf{A}\mathbf{x}) \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}}.$$

Sei nun $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ beliebig. Wir setzen $\mathbf{x} := \mathbf{A}^{-1}(\mathbf{b} - \mathbf{y})$ und erhalten

$$\mathbf{0} = (\mathbf{W}_1^{-1} - \mathbf{W}_2^{-1})(\mathbf{b} - (\mathbf{b} - \mathbf{y})) = (\mathbf{W}_1^{-1} - \mathbf{W}_2^{-1})\mathbf{y},$$

also folgt mit $\mathbf{W}_1^{-1} = \mathbf{W}_2^{-1}$ auch $\mathbf{W}_1 = \mathbf{W}_2$. ∎

Wenn wir die Konvergenzrate einer derartigen Iteration als Reduktionsfaktor des Fehlers $\mathbf{x}^* - \mathbf{x}^{(m)}$ in einer geeigneten Norm definieren, sollten wir diesen Fehler näher untersuchen.

Lemma 7.13 (Fehlerfortpflanzung) *Es gilt*

$$\mathbf{x}^* - \mathbf{x}^{(m+1)} = (\mathbf{I} - \mathbf{W}^{-1}\mathbf{A})(\mathbf{x}^* - \mathbf{x}^{(m)}) \qquad \text{für alle } m \in \mathbb{N}_0.$$

Beweis. Sei $m \in \mathbb{N}_0$. Mit (7.15) und (7.13) erhalten wir

$$\begin{aligned} \mathbf{x}^* - \mathbf{x}^{(m+1)} &= \mathbf{x}^* - \mathbf{x}^{(m)} - \mathbf{W}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(m)}) \\ &= \mathbf{x}^* - \mathbf{x}^{(m)} - \mathbf{W}^{-1}(\mathbf{A}\mathbf{x}^* - \mathbf{A}\mathbf{x}^{(m)}) \\ &= (\mathbf{I} - \mathbf{W}^{-1}\mathbf{A})(\mathbf{x}^* - \mathbf{x}^{(m)}). \end{aligned}$$

∎

Definition 7.14 (Iterationsmatrix) *Sei Φ ein lineares Iterationsverfahren. Wir bezeichnen*

$$\mathbf{M} := \mathbf{I} - \mathbf{W}^{-1}\mathbf{A}$$

als die zugehörige Iterationsmatrix.

Mit der Iterationsmatrix können wir Lemma 7.13 kurz als

$$\mathbf{x} - \mathbf{x}^{(m+1)} = \mathbf{M}(\mathbf{x} - \mathbf{x}^{(m)}) \qquad \text{für alle } m \in \mathbb{N}_0 \tag{7.16}$$

schreiben. Die Iterationsmatrix beschreibt also gerade die Entwicklung des Fehlers.

Für eine beliebige Norm $\|\cdot\|$ und die von ihr induzierte Matrixnorm

$$\|\mathbf{X}\| := \sup\left\{\frac{\|\mathbf{X}\mathbf{y}\|}{\|\mathbf{y}\|} \;:\; \mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\} \qquad \text{für alle } \mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$$

erhalten wir

$$\|\mathbf{x} - \mathbf{x}^{(m+1)}\| \leq \|\mathbf{M}\|\,\|\mathbf{x} - \mathbf{x}^{(m)}\| \qquad \text{für alle } m \in \mathbb{N}_0.$$

Falls also $\|\mathbf{M}\| < 1$ gilt, wird das Verfahren konvergieren, und es wird um so schneller konvergieren, je kleiner die Matrixnorm ist.

Die Frage ist nun, welche Norm für unsere Zwecke gut geeignet ist. Die Matrizen $\mathbf{W}^{-1}$ und $\mathbf{A}$ sind selbstadjungiert, und diese Eigenschaft lässt sich ausnutzen, um beispielsweise die von der euklidischen Norm induzierte *Spektralnorm*

$$\|\mathbf{X}\|_2 := \sup\left\{\frac{\|\mathbf{X}\mathbf{y}\|_2}{\|\mathbf{y}\|_2} \;:\; \mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\} \qquad \text{für alle } \mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$$

abzuschätzen. Den Namen „Spektralnorm“ verdankt sie ihrer engen Beziehung zu dem *Spektrum* einer Matrix, also der Menge ihrer Eigenwerte.

Definition 7.15 (Spektrum) *Sei $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine Matrix. Ein $\lambda \in \mathbb{R}$ nennen wir einen* Eigenwert *der Matrix $\mathbf{X}$, falls ein Vektor $\mathbf{e} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$ mit*

$$\mathbf{X}\mathbf{e} = \lambda\mathbf{e}$$

existiert. In diesem Fall nennen wir $\mathbf{e}$ einen Eigenvektor *der Matrix zu dem Eigenwert λ. Die Menge*

$$\sigma(\mathbf{X}) := \{\lambda \in \mathbb{R} \;:\; \lambda \text{ ist Eigenwert von } \mathbf{X}\}$$

nennen wir das Spektrum *der Matrix.*

Für selbstadjungierte Matrizen lassen sich die Eigenwerte als lokale Extrema des *Rayleigh-Quotienten* charakterisieren.

Lemma 7.16 (Rayleigh-Quotient) *Sei $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine selbstadjungierte Matrix. Wir definieren die Abbildung*

$$\Lambda_X : \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\} \to \mathbb{R}, \qquad \mathbf{y} \mapsto \frac{\langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle}{\langle \mathbf{y}, \mathbf{y}\rangle}.$$

Diese Abbildung besitzt ein Minimum und ein Maximum. Beide Extrema sind Eigenwerte der Matrix $\mathbf{X}$.

Beweis. Wir stellen fest, dass für jedes $\alpha \in \mathbb{R} \setminus \{0\}$ und jeden Vektor $\mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$ die Gleichung

$$\Lambda_X(\alpha\mathbf{y}) = \frac{\langle \alpha\mathbf{y}, \mathbf{X}\alpha\mathbf{y}\rangle}{\langle \alpha\mathbf{y}, \alpha\mathbf{y}\rangle} = \frac{\alpha^2\langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle}{\alpha^2\langle \mathbf{y}, \mathbf{y}\rangle} = \Lambda_X(\mathbf{y})$$

gilt. Demnach genügt es, Minimum und Maximum auf der Einheitssphäre

$$\mathcal{S} := \{y \in \mathbb{R}^{\mathcal{I}} \ : \ \|\mathbf{y}\|_2 = 1\}$$

zu suchen. Diese Menge ist nach dem Satz von Heine-Borel kompakt, also muss die stetige Abbildung Λ_X auf ihr ein Minimum und ein Maximum annehmen.

Sei $\lambda \in \mathbb{R}$ das Maximum, und sei $\mathbf{e} \in \mathcal{S}$ ein Urbild, also ein Vektor mit $\Lambda_X(\mathbf{e}) = \lambda$. Wir wollen zeigen, dass $\mathbf{e}$ ein Eigenvektor zu dem Eigenwert λ ist.

Sei $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ und sei $\delta \in \mathbb{R}_{>0}$ mit $\delta\|\mathbf{y}\|_2 \leq 1$ fixiert. Wir definieren

$$\begin{aligned} f : (-\delta, \delta) \to \mathbb{R}, &\qquad t \mapsto \langle \mathbf{e} + t\mathbf{y}, \mathbf{X}(\mathbf{e} + t\mathbf{y})\rangle = \langle \mathbf{e}, \mathbf{X}\mathbf{e}\rangle + 2t\langle \mathbf{y}, \mathbf{X}\mathbf{e}\rangle + t^2\langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle, \\ g : (-\delta, \delta) \to \mathbb{R}, &\qquad t \mapsto \langle \mathbf{e} + t\mathbf{y}, \mathbf{e} + t\mathbf{y}\rangle = \langle \mathbf{e}, \mathbf{e}\rangle + 2t\langle \mathbf{y}, \mathbf{e}\rangle + t^2\langle \mathbf{y}, \mathbf{y}\rangle, \end{aligned}$$

und halten fest, dass wegen der Cauchy-Schwarz-Ungleichung (6.13)

$$\begin{aligned} g(t) &= \|\mathbf{e}\|_2^2 + 2t\langle \mathbf{y}, \mathbf{e}\rangle + t^2\|\mathbf{y}\|_2^2 \geq \|\mathbf{e}\|_2^2 - 2|t|\,\|\mathbf{y}\|_2\|\mathbf{e}\|_2 + t^2\|\mathbf{y}\|_2^2 \\ &= (\|\mathbf{e}\|_2 - |t|\,\|\mathbf{y}\|)^2 = (1 - |t|\,\|\mathbf{y}\|)^2 > 0 \qquad \text{für alle } t \in (-\delta, \delta) \end{aligned}$$

gilt. Damit ist

$$h : (-\delta, \delta) \to \mathbb{R}, \qquad t \mapsto \Lambda_X(\mathbf{e} + t\mathbf{y}) = \frac{f(t)}{g(t)},$$

eine stetig differenzierbare Abbildung, die ihr Maximum in $t = 0$ annimmt, also muss $h'(0) = 0$ gelten. Mit dieser Gleichung erhalten wir

$$\begin{aligned} 0 = h'(0) &= \frac{f'(0)g(0) - f(0)g'(0)}{g(0)^2} \\ &= \frac{2\langle \mathbf{y}, \mathbf{X}\mathbf{e}\rangle\langle \mathbf{e}, \mathbf{e}\rangle - 2\langle \mathbf{e}, \mathbf{X}\mathbf{e}\rangle\langle \mathbf{y}, \mathbf{e}\rangle}{\langle \mathbf{e}, \mathbf{e}\rangle^2} = 2\frac{\langle \mathbf{y}, \mathbf{X}\mathbf{e} - \Lambda_X(\mathbf{e})\mathbf{e}\rangle}{\langle \mathbf{e}, \mathbf{e}\rangle}. \end{aligned}$$

Da wir diese Gleichung für jedes $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ gilt, folgt $\mathbf{X}\mathbf{e} = \Lambda_X(\mathbf{e})\mathbf{e}$, also ist $\mathbf{e}$ ein Eigenvektor zu dem Eigenwert $\lambda = \Lambda_X(\mathbf{e})$. Den Nachweis, dass auch das Minimum der Abbildung Λ_X ein Eigenwert ist, können wir analog führen. ∎

Erinnerung 7.17 (Orthogonale Matrizen) *Eine Matrix* $\mathbf{Q} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *heißt* orthogonal*, falls* $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$ *gilt. Diese Eigenschaft ist äquivalent dazu, dass*

$$\|\mathbf{Q}\mathbf{z}\|_2 = \|\mathbf{z}\|_2 \qquad \text{für alle } \mathbf{z} \in \mathbb{R}^{\mathcal{I}}$$

gilt. Jede orthogonale Matrix ist invertierbar mit $\mathbf{Q}^{-1} = \mathbf{Q}^*$.

Satz 7.18 (Hauptachsentransformation) *Sei* $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *eine selbstadjungierte Matrix. Dann existieren eine orthogonale Matrix* $\mathbf{Q} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *und eine Diagonalmatrix* $\mathbf{D} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *mit* $\mathbf{Q}^*\mathbf{X}\mathbf{Q} = \mathbf{D}$.

Die Diagonalelemente der Matrix $\mathbf{D}$ *sind die Eigenwerte der Matrix* $\mathbf{X}$*, die Spalten der Matrix* $\mathbf{Q}$ *sind entsprechende Eigenvektoren.*

Beweis. Ohne Beschränkung der Allgemeinheit untersuchen wir nur Indexmengen der Form $\mathcal{I} = \{1, \ldots, n\}$ mit $n \in \mathbb{N}$. Wir führen den Beweis per Induktion über n.

Induktionsanfang: Für $n = 1$ ist jede Matrix auch eine Diagonalmatrix, also können wir $\mathbf{Q} = 1$ und $\mathbf{D} = \mathbf{X}$ setzen.

Induktionsvoraussetzung: Sei $n \in \mathbb{N}$ so gegeben, dass die Aussage für alle selbstadjungierten Matrizen $\mathbf{X} \in \mathbb{R}^{n\times n}$ gilt.

Induktionsschritt: Sei $\mathbf{X} \in \mathbb{R}^{(n+1)\times(n+1)}$. Nach Lemma 7.16 finden wir einen Eigenwert $\lambda \in \mathbb{R}$ und einen passenden Eigenvektor $\mathbf{e} \in \mathbb{R}^{n+1}$ mit $\|\mathbf{e}\|_2 = 1$. Sei $\delta_1 \in \mathbb{R}^{n+1}$ der erste kanonische Einheitsvektor $\delta_1 = (1, 0, \ldots, 0)$, und sei $\mathbf{Q}_1 \in \mathbb{R}^{(n+1)\times(n+1)}$ die Householder-Transformation, die $\mathbf{Q}_1^*\mathbf{e} = \alpha\delta_1$ mit $|\alpha| = 1$ erfüllt. Es folgt

$$\mathbf{Q}_1^*\mathbf{X}\mathbf{Q}_1\delta_1 = \frac{1}{\alpha}\mathbf{Q}_1^*\mathbf{X}\mathbf{e} = \lambda\frac{1}{\alpha}\mathbf{Q}_1^*\mathbf{e} = \lambda\delta_1,$$

und da $\mathbf{Q}_1^*\mathbf{X}\mathbf{Q}_1$ selbstadjungiert ist, erhalten wir

$$\mathbf{Q}_1^*\mathbf{X}\mathbf{Q}_1 = \begin{pmatrix} \lambda & \\ & \widehat{\mathbf{X}} \end{pmatrix}$$

mit einer selbstadjungierten Matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{n\times n}$. Nach Induktionsvoraussetzung existieren eine orthogonale Matrix $\widehat{\mathbf{Q}} \in \mathbb{R}^{n\times n}$ und eine Diagonalmatrix $\widehat{\mathbf{D}} \in \mathbb{R}^{n\times n}$ mit $\widehat{\mathbf{Q}}^*\widehat{\mathbf{X}}\widehat{\mathbf{Q}} = \widehat{\mathbf{D}}$, so dass wir insgesamt

$$\begin{pmatrix} 1 & \\ & \widehat{\mathbf{Q}} \end{pmatrix}^* \mathbf{Q}_1^*\mathbf{X}\mathbf{Q}_1 \begin{pmatrix} 1 & \\ & \widehat{\mathbf{Q}} \end{pmatrix} = \begin{pmatrix} \lambda & \\ & \widehat{\mathbf{Q}}^*\widehat{\mathbf{X}}\widehat{\mathbf{Q}} \end{pmatrix} = \begin{pmatrix} \lambda & \\ & \widehat{\mathbf{D}} \end{pmatrix}$$

erhalten. Mit

$$\mathbf{Q} := \mathbf{Q}_1 \begin{pmatrix} 1 & \\ & \widehat{\mathbf{Q}} \end{pmatrix}, \qquad \mathbf{D} := \begin{pmatrix} \lambda & \\ & \widehat{\mathbf{D}} \end{pmatrix}$$

folgt daraus die Behauptung. ∎

Folgerung 7.19 (Spektralnorm) *Sei $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine selbstadjungierte Matrix. Dann gilt*

$$\|\mathbf{X}\|_2 = \max\{|\lambda| \ : \ \lambda \in \sigma(\mathbf{X})\}.$$

Beweis. Nach Satz 7.18 existieren eine orthogonale Matrix $\mathbf{Q} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und eine Diagonalmatrix $\mathbf{D} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ mit $\mathbf{X} = \mathbf{QDQ}^*$.

Da orthogonale Transformationen die euklidische Norm — und damit auch die Spektralnorm — unverändert lassen, folgt $\|\mathbf{X}\|_2 = \|\mathbf{QDQ}^*\|_2 = \|\mathbf{D}\|_2$.

Sei $j \in \mathcal{I}$ ein Index mit $|d_{jj}| \geq |d_{ii}|$ für alle $i \in \mathcal{I}$. Sei $\mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$. Dann gilt

$$\frac{\|\mathbf{Dy}\|_2^2}{\|\mathbf{y}\|_2^2} = \frac{\sum_{i\in\mathcal{I}} d_{ii}^2 y_i^2}{\sum_{i\in\mathcal{I}} y_i^2} \leq \frac{\sum_{i\in\mathcal{I}} d_{jj}^2 y_i^2}{\sum_{i\in\mathcal{I}} y_i^2} = d_{jj}^2,$$

also $\|\mathbf{X}\|_2 = \|\mathbf{D}\|_2 \leq |d_{jj}|$. Wenn wir den durch

$$\delta_{j,i} := \begin{cases} 1 & \text{falls } i = j, \\ 0 & \text{ansonsten} \end{cases} \qquad \text{für alle } i \in \mathcal{I}$$

gegebenen kanonischen Einheitsvektor $\delta_j \in \mathbb{R}^{\mathcal{I}}$ in $\mathbf{D}$ einsetzen, erhalten wir

$$\frac{\|\mathbf{D}\delta_j\|_2}{\|\delta_j\|_2} = \frac{\|d_{jj}\delta_j\|_2}{\|\delta_j\|_2} = |d_{jj}|,$$

und damit $\|\mathbf{X}\|_2 = \|\mathbf{D}\|_2 = |d_{jj}|$. Da d_{jj} der betragsgrößte Eigenwert der Matrix $\mathbf{X}$ ist, ist der Beweis vollständig. ∎

Leider ist gerade die für die Konvergenz relevante Iterationsmatrix $\mathbf{M}$ in der Regel nicht selbstadjungiert. Wir können allerdings eine Norm wählen, mit der wir die Untersuchung der Iterationsmatrix auf die einer symmetrischen Matrix zurückführen können, nämlich die durch

$$\|\mathbf{x}\|_A := \sqrt{\langle \mathbf{x}, \mathbf{Ax}\rangle} \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}}$$

definierte *Energienorm*. Da $\mathbf{A}$ als positiv definit und selbstadjungiert vorausgesetzt ist, ist die Energienorm wohldefiniert.

Insbesondere besitzt $\mathbf{A}$ aber auch eine *Cholesky-Zerlegung* $\mathbf{A} = \mathbf{R}^*\mathbf{R}$ mit einer invertierbaren Matrix $\mathbf{R} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$, mit deren Hilfe sich die Energienorm auf die euklidische Norm zurückführen lässt. Es gilt nämlich

$$\|\mathbf{x}\|_A^2 = \langle \mathbf{x}, \mathbf{Ax}\rangle = \langle \mathbf{x}, \mathbf{R}^*\mathbf{Rx}\rangle = \langle \mathbf{Rx}, \mathbf{Rx}\rangle = \|\mathbf{Rx}\|_2^2 \qquad \text{für alle } \mathbf{x} \in \mathbb{R}^{\mathcal{I}},$$

also insbesondere $\|\mathbf{x}\|_A = \|\mathbf{Rx}\|_2$. Die Cholesky-Zerlegung ist nicht eindeutig, wir können die Vorzeichen jeder Zeile der Matrix $\mathbf{R}$ beliebig wählen. Das ist für unsere Zwecke zwar ohne Belang, da es die Norm nicht beeinflusst, aber damit im Folgenden beispielsweise die Matrix $\widehat{\mathbf{A}}$ wohldefiniert ist, legen wir fest, dass wir diejenige Cholesky-Faktorisierung verwenden bei der alle Diagonalelemente der Matrix $\mathbf{R}$ positiv sind.

Von der Energienorm wird die Norm

$$\|\mathbf{X}\|_A = \sup\left\{\frac{\|\mathbf{Xy}\|_A}{\|\mathbf{y}\|_A} \;:\; \mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\} \qquad \text{für alle } \mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$$

induziert, die wir mit Hilfe der Cholesky-Zerlegung auf die Spektralnorm zurückführen können.

Lemma 7.20 (Energienorm) *Für alle* $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *gilt*

$$\|\mathbf{X}\|_A = \|\mathbf{RXR}^{-1}\|_2.$$

Beweis. Sei $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$. Mit der Substitution $\mathbf{Ry} = \mathbf{z}$ erhalten wir

$$\begin{aligned}\|\mathbf{X}\|_A &= \sup\left\{\frac{\|\mathbf{Xy}\|_A}{\|\mathbf{y}\|_A} \;:\; \mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\}\\ &= \sup\left\{\frac{\|\mathbf{RXy}\|_2}{\|\mathbf{Ry}\|_2} \;:\; \mathbf{y} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\}\\ &= \sup\left\{\frac{\|\mathbf{RXR}^{-1}\mathbf{z}\|_2}{\|\mathbf{z}\|_2} \;:\; \mathbf{z} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}\right\} = \|\mathbf{RXR}^{-1}\|_2.\end{aligned}$$

■

Für die Iterationsmatrix folgt

$$\|\mathbf{M}\|_A = \|\mathbf{R}(\mathbf{I} - \mathbf{W}^{-1}\mathbf{A})\mathbf{R}^{-1}\|_2 = \|\mathbf{I} - \mathbf{RW}^{-1}\mathbf{R}^*\mathbf{RR}^{-1}\|_2 = \|\mathbf{I} - \mathbf{RW}^{-1}\mathbf{R}^*\|_2.$$

Falls also $\mathbf{W}$ selbstadjungiert ist, wird die *transformierte Iterationsmatrix*

$$\widehat{\mathbf{M}} := \mathbf{I} - \mathbf{RW}^{-1}\mathbf{R}^* \tag{7.17}$$

ebenfalls selbstadjungiert sein. Falls wir die Spektralnorm dieser Matrix abschätzen könnten, würde sich mit $\|\mathbf{M}\|_A = \|\widehat{\mathbf{M}}\|_2$ eine Konvergenzaussage in der Energienorm ergeben.

Definition 7.21 (Symmetrisches Iterationsverfahren) *Ein lineares Iterationsverfahren* $\Phi : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}}$ *nennen wir* symmetrisch*, falls die gemäß (7.15) zugehörige Matrix* $\mathbf{W}$ *selbstadjungiert ist.*

Falls das Verfahren Φ symmetrisch ist, ist die gemäß (7.17) definierte transformierte Iterationsmatrix $\widehat{\mathbf{M}}$ selbstadjungiert, also nach Satz 7.18 diagonalisierbar mit ausschließlich reellen Eigenwerten. Da $\mathbf{M} = \mathbf{R}^{-1}\widehat{\mathbf{M}}\mathbf{R}$ gilt, ist dann auch die Iterationsmatrix $\mathbf{M}$ diagonalisierbar mit reellen Eigenwerten.

Sei Φ im Folgenden als symmetrisch vorausgesetzt.

Um eine Konvergenzaussage zu erhalten, bietet es sich wegen (7.16 an, nach Schranken für $\|\mathbf{M}\|_A = \|\widehat{\mathbf{M}}\|_2$ zu suchen. Konvergenz ist sicher gestellt, falls

$$\|\mathbf{M}\|_A = \|\widehat{\mathbf{M}}\|_2 < 1$$

gilt, und nach Folgerung 7.19 ist das genau dann der Fall, wenn die Beträge aller Eigenwerte echt kleiner als eins sind.

Aussagen über die Größe der Eigenwerte lassen sich elegant formulieren, indem man eine partielle Ordnung auf der Menge der selbstadjungierten Matrizen definiert.

Definition 7.22 (Partielle Ordnung) *Seien* $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *selbstadjungiert. Falls*

$$\langle \mathbf{z}, \mathbf{X}\mathbf{z}\rangle < \langle \mathbf{z}, \mathbf{Y}\mathbf{z}\rangle \qquad \text{für alle } \mathbf{z} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}$$

gilt, schreiben wir $\mathbf{X} < \mathbf{Y}$*. Falls*

$$\langle \mathbf{z}, \mathbf{X}\mathbf{z}\rangle \leq \langle \mathbf{z}, \mathbf{Y}\mathbf{z}\rangle \qquad \text{für alle } \mathbf{z} \in \mathbb{R}^{\mathcal{I}}$$

gilt, schreiben wir $\mathbf{X} \leq \mathbf{Y}$*.*

Lemma 7.23 (Elementare Eigenschaften) *Seien* $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *selbstadjungierte Matrizen. Dann gelten*

$$\mathbf{X} \leq \mathbf{Y} \wedge \mathbf{Y} \leq \mathbf{Z} \Longrightarrow \mathbf{X} \leq \mathbf{Z}, \tag{7.18a}$$
$$\mathbf{X} < \mathbf{Y} \wedge \mathbf{Y} \leq \mathbf{Z} \Longrightarrow \mathbf{X} < \mathbf{Z}, \tag{7.18b}$$
$$\mathbf{X} \leq \mathbf{Y} \Longrightarrow \mathbf{X} + \mathbf{Z} \leq \mathbf{Y} + \mathbf{Z}, \tag{7.18c}$$
$$\mathbf{X} \leq \mathbf{Y} \Longrightarrow -\mathbf{Y} \leq -\mathbf{X}, \tag{7.18d}$$
$$\mathbf{X} \leq \mathbf{Y} \Longrightarrow \alpha\mathbf{X} \leq \alpha\mathbf{Y} \qquad \text{für alle } \alpha \in \mathbb{R}_{\geq 0}, \tag{7.18e}$$
$$\mathbf{X} < \mathbf{Y} \Longrightarrow \alpha\mathbf{X} < \alpha\mathbf{Y} \qquad \text{für alle } \alpha \in \mathbb{R}_{>0}. \tag{7.18f}$$

Beweis. Die Aussagen folgen unmittelbar aus der Definition 7.22. ∎

Insbesondere bedeutet $\mathbf{0} < \mathbf{X}$, dass $\mathbf{X}$ selbstadjungiert und positiv definit ist. Diese partielle Ordnung hat den Vorteil, dass sie unter *Kongruenztransformationen* invariant ist, die wir verwenden können, um die benötigten Aussagen herzuleiten.

Lemma 7.24 (Kongruenztransformation) *Seien* $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *selbstadjungiert. Sei* $\mathbf{Z} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$*. Es gilt*

$$\mathbf{X} \leq \mathbf{Y} \Longrightarrow \mathbf{Z}^*\mathbf{X}\mathbf{Z} \leq \mathbf{Z}^*\mathbf{Y}\mathbf{Z}.$$

Falls $\mathbf{Z}$ *injektiv ist, gilt*

$$\mathbf{X} < \mathbf{Y} \Longrightarrow \mathbf{Z}^*\mathbf{X}\mathbf{Z} < \mathbf{Z}^*\mathbf{Y}\mathbf{Z}.$$

Falls $\mathbf{Z}$ *invertierbar ist, gelten*

$$\mathbf{X} < \mathbf{Y} \iff \mathbf{Z}^*\mathbf{X}\mathbf{Z} < \mathbf{Z}^*\mathbf{Y}\mathbf{Z},$$
$$\mathbf{X} \leq \mathbf{Y} \iff \mathbf{Z}^*\mathbf{X}\mathbf{Z} \leq \mathbf{Z}^*\mathbf{Y}\mathbf{Z}.$$

Beweis. Wir setzen $\widehat{\mathbf{X}} := \mathbf{Z}^*\mathbf{X}\mathbf{Z}$ und $\widehat{\mathbf{Y}} := \mathbf{Z}^*\mathbf{Y}\mathbf{Z}$.

Gelte $\mathbf{X} \leq \mathbf{Y}$, sei $\mathbf{z} \in \mathbb{R}^{\mathcal{J}}$. Dann folgt

$$\langle \mathbf{z}, (\widehat{\mathbf{Y}} - \widehat{\mathbf{X}})\mathbf{z}\rangle = \langle \mathbf{z}, \mathbf{Z}^*(\mathbf{Y} - \mathbf{X})\mathbf{Z}\mathbf{z}\rangle = \langle \mathbf{Z}\mathbf{z}, (\mathbf{Y} - \mathbf{X})(\mathbf{Z}\mathbf{z})\rangle \geq 0,$$

also $\widehat{\mathbf{X}} \leq \widehat{\mathbf{Y}}$.

Sei nun $\mathbf{Z}$ injektiv. Gelte $\mathbf{X} < \mathbf{Y}$, sei $\mathbf{z} \in \mathbb{R}^{\mathcal{J}} \setminus \{\mathbf{0}\}$. Dann folgt $\mathbf{Z}\mathbf{z} \neq \mathbf{0}$, also per Definition

$$\langle \mathbf{z}, (\widehat{\mathbf{Y}} - \widehat{\mathbf{X}})\mathbf{z}\rangle = \langle \mathbf{z}, \mathbf{Z}^*(\mathbf{Y} - \mathbf{X})\mathbf{Z}\mathbf{z}\rangle = \langle \mathbf{Z}\mathbf{z}, (\mathbf{Y} - \mathbf{X})(\mathbf{Z}\mathbf{z})\rangle > 0,$$

und damit $\widehat{\mathbf{X}} < \widehat{\mathbf{Y}}$.

Falls $\mathbf{Z}$ invertierbar ist, folgen die restlichen Aussagen, indem wir in den bereits bewiesenen die Rollen der Matrizen $\widehat{\mathbf{X}}$ und $\mathbf{X}$ sowie $\widehat{\mathbf{Y}}$ und $\mathbf{Y}$ tauschen und $\mathbf{Z}$ durch $\mathbf{Z}^{-1}$ ersetzen. ∎

Lemma 7.25 (Spektrum und Norm) *Sei $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ selbstadjungiert, und seien $\alpha, \beta \in \mathbb{R}$ gegeben mit $\alpha \leq \beta$.*

Es gilt $\sigma(\mathbf{X}) \subseteq (\alpha, \beta)$ genau dann, wenn $\alpha\mathbf{I} < \mathbf{X} < \beta\mathbf{I}$ gilt.

Es gilt $\sigma(\mathbf{X}) \subseteq [\alpha, \beta]$ genau dann, wenn $\alpha\mathbf{I} \leq \mathbf{X} \leq \beta\mathbf{I}$ gilt.

Es gilt $\|\mathbf{X}\|_2 \leq \beta$ genau dann, wenn $-\beta\mathbf{I} \leq \mathbf{X} \leq \beta\mathbf{I}$ gilt.

Beweis. Nach Satz 7.18 existieren eine orthogonale Matrix $\mathbf{Q} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ und eine Diagonalmatrix $\mathbf{D} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ mit $\mathbf{Q}^*\mathbf{X}\mathbf{Q} = \mathbf{D}$.

Da orthogonale Matrizen invertierbar sind folgt mit Lemma 7.24

$$\alpha\mathbf{I} < \mathbf{X} < \beta\mathbf{I} \quad \Longleftrightarrow \quad \alpha\mathbf{Q}^*\mathbf{Q} < \mathbf{Q}^*\mathbf{X}\mathbf{Q} < \beta\mathbf{Q}^*\mathbf{Q},$$

also wegen $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$ insbesondere

$$\alpha\mathbf{I} < \mathbf{X} < \beta\mathbf{I} \quad \Longleftrightarrow \quad \alpha\mathbf{I} < \mathbf{D} < \beta\mathbf{I}.$$

Da $\mathbf{D}$ eine Diagonalmatrix ist, ist $\alpha\mathbf{I} < \mathbf{D} < \beta\mathbf{I}$ äquivalent zu

$$\alpha \sum_{i\in\mathcal{I}} |z_i|^2 = \alpha\langle \mathbf{z}, \mathbf{I}\mathbf{z}\rangle < \langle \mathbf{z}, \mathbf{D}\mathbf{z}\rangle = \sum_{i\in\mathcal{I}} d_{ii}|z_i|^2 \tag{7.19a}$$

$$= \langle \mathbf{z}, \mathbf{D}\mathbf{z}\rangle < \beta\langle \mathbf{z}, \mathbf{I}\mathbf{z}\rangle = \beta \sum_{i\in\mathcal{I}} |z_i|^2 \qquad \text{für alle } \mathbf{z} \in \mathbb{R}^{\mathcal{I}} \setminus \{\mathbf{0}\}. \tag{7.19b}$$

Falls nun $\sigma(\mathbf{X}) = \sigma(\mathbf{D}) \subseteq (\alpha, \beta)$ gilt, folgt $d_{ii} \in (\alpha, \beta)$ für alle $i \in \mathcal{I}$ und damit (7.19), also $\alpha\mathbf{I} < \mathbf{X} < \beta\mathbf{I}$.

Falls umgekehrt $\alpha\mathbf{I} < \mathbf{X} < \beta\mathbf{I}$ gilt, gilt (7.19), und indem wir kanonische Einheitvektoren einsetzen, erhalten wir $d_{ii} \in (\alpha, \beta)$ für alle $i \in \mathcal{I}$, also auch $\sigma(\mathbf{X}) = \sigma(\mathbf{D}) \subseteq (\alpha, \beta)$.

Die zweite Aussage folgt analog, die dritte ergibt sich dann aus der Kombination mit Folgerung 7.19. ∎

Bemerkung 7.26 (Alternativer Beweis) *Die für uns besonders wichtige dritte Aussage des obigen Lemmas lässt sich auch beweisen, ohne auf die Hauptachsentransformation zurückzugreifen. Insbesondere behält dieser alternative Beweis seine Gültigkeit, falls wir statt in $\mathbb{R}^{\mathcal{I}}$ in einem unendlich-dimensionalen Hilbertraum arbeiten.*

Wir illustrieren die Vorgehensweise am Fall $\alpha = 1$, sei also $\mathbf{X} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ eine selbstadjungierte Matrix, die

$$-\mathbf{I} \leq \mathbf{X} \leq \mathbf{I}$$

erfüllt, es gelte demnach

$$-\langle \mathbf{z}, \mathbf{z}\rangle \leq \langle \mathbf{z}, \mathbf{X}\mathbf{z}\rangle \leq \langle \mathbf{z}, \mathbf{z}\rangle \qquad \text{für alle } \mathbf{z} \in \mathbb{R}^{\mathcal{I}}. \tag{7.20}$$

Sei $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$. Indem wir die linke Ungleichung in (7.20) auf $\mathbf{z} = \mathbf{y} - \mathbf{X}\mathbf{y}$ anwenden, folgt

$$\begin{aligned}\langle \mathbf{y} - \mathbf{X}\mathbf{y}, \mathbf{y} - \mathbf{X}\mathbf{y}\rangle &\geq -\langle \mathbf{y} - \mathbf{X}\mathbf{y}, \mathbf{X}(\mathbf{y} - \mathbf{X}\mathbf{y})\rangle \\ &= -\langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle + \langle \mathbf{X}\mathbf{y}, \mathbf{X}\mathbf{y}\rangle - \langle \mathbf{X}\mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle \\ &= -\langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle + 2\langle \mathbf{X}\mathbf{y}, \mathbf{X}\mathbf{y}\rangle - \langle \mathbf{X}\mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle.\end{aligned}$$

Wenn wir die rechte Ungleichung in (7.20) auf $\mathbf{z} = \mathbf{y} + \mathbf{X}\mathbf{y}$ anwenden, ergibt sich entsprechend

$$\begin{aligned}\langle \mathbf{y} + \mathbf{X}\mathbf{y}, \mathbf{y} + \mathbf{X}\mathbf{y}\rangle &\geq \langle \mathbf{y} + \mathbf{X}\mathbf{y}, \mathbf{X}(\mathbf{y} + \mathbf{X}\mathbf{y})\rangle \\ &= \langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle + \langle \mathbf{X}\mathbf{y}, \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{X}\mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle \\ &= \langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle + 2\langle \mathbf{X}\mathbf{y}, \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{X}\mathbf{y}, \mathbf{X}^2\mathbf{y}\rangle.\end{aligned}$$

Durch Addition beider Ungleichungen gelangen wir zu

$$\begin{aligned}4\|\mathbf{X}\mathbf{y}\|_2^2 = 4\langle \mathbf{X}\mathbf{y}, \mathbf{X}\mathbf{y}\rangle &\leq \langle \mathbf{y} - \mathbf{X}\mathbf{y}, \mathbf{y} - \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{y} + \mathbf{X}\mathbf{y}, \mathbf{y} + \mathbf{X}\mathbf{y}\rangle \\ &= \|\mathbf{y}\|_2^2 - \langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle - \langle \mathbf{X}\mathbf{y}, \mathbf{y}\rangle + \|\mathbf{X}\mathbf{y}\|_2^2 + \|\mathbf{y}\|_2^2 + \langle \mathbf{y}, \mathbf{X}\mathbf{y}\rangle + \langle \mathbf{X}\mathbf{y}, \mathbf{y}\rangle + \|\mathbf{X}\mathbf{y}\|_2^2 \\ &= 2\|\mathbf{y}\|_2^2 + 2\|\mathbf{X}\mathbf{y}\|_2^2.\end{aligned}$$

Wir subtrahieren $2\|\mathbf{X}\mathbf{y}\|_2^2$ auf beiden Seiten dieser Ungleichung und dividieren durch zwei, um das gewünschte Ergebnis $\|\mathbf{X}\mathbf{y}\|_2^2 \leq \|\mathbf{y}\|^2$ zu erhalten. Da wir diese Ungleichung für beliebige $\mathbf{y} \in \mathbb{R}^{\mathcal{I}}$ gezeigt haben, folgt $\|\mathbf{X}\|_2 \leq 1$.

Lemma 7.27 (Inverse) *Seien $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ selbstadjungierte Matrizen, die $\mathbf{0} < \mathbf{X} \leq \mathbf{Y}$ erfüllen. Dann gilt $\mathbf{0} < \mathbf{Y}^{-1} \leq \mathbf{X}^{-1}$.*

Beweis. Wir verwenden eine Cholesky-Zerlegung $\mathbf{R}^*\mathbf{R} = \mathbf{X}$ der Matrix $\mathbf{X}$ und folgern mit Lemma 7.24, dass

$$\mathbf{I} \leq (\mathbf{R}^*)^{-1}\mathbf{Y}\mathbf{R}^{-1} =: \widehat{\mathbf{Y}}$$

gilt. Nach Lemma 7.25 sind alle Eigenwerte der Matrix $\widehat{\mathbf{Y}}$ größer oder gleich eins. Die Eigenwerte ihrer Inversen $\widehat{\mathbf{Y}}^{-1}$ sind deren Kehrwerte, müssen also positiv und kleiner oder gleich eins sein. Wieder mit Lemma 7.25 folgt

$$\mathbf{0} < \mathbf{R}\mathbf{Y}^{-1}\mathbf{R}^* = \widehat{\mathbf{Y}}^{-1} \leq \mathbf{I},$$

und Lemma 7.24 führt zu

$$\mathbf{0} < \mathbf{Y}^{-1} \leq \mathbf{R}^{-1}(\mathbf{R}^*)^{-1} = \mathbf{X}^{-1}.$$

■

Satz 7.28 (Konvergenz) *Sei Φ ein symmetrisches lineares Iterationsverfahren, sei $\varrho \in [0,1)$. Es gilt $\|M\|_A \leq \varrho$ genau dann, wenn*

$$\frac{1}{1+\varrho}\mathbf{A} \leq \mathbf{W} \leq \frac{1}{1-\varrho}\mathbf{A}$$

gilt. Falls $\mathbf{0} < \mathbf{A} < 2\mathbf{W}$ gilt, konvergiert das Verfahren.

Beweis. Nach Lemma 7.20 gilt $\|\mathbf{M}\|_A = \|\widehat{\mathbf{M}}\|_2$, und nach Lemma 7.25 gilt $\|\widehat{\mathbf{M}}\|_2 \leq \varrho$ genau dann, wenn

$$-\varrho\mathbf{I} \leq \widehat{\mathbf{M}} \leq \varrho\mathbf{I}$$

gilt. Das ist nach Lemma 7.24 äquivalent zu

$$\begin{aligned}
-\varrho\mathbf{I} \leq \mathbf{I} - \mathbf{R}\mathbf{W}^{-1}\mathbf{R}^* &\leq \varrho\mathbf{I},\\
-(1+\varrho)\mathbf{I} \leq -\mathbf{R}\mathbf{W}^{-1}\mathbf{R}^* &\leq (\varrho-1)\mathbf{I},\\
(1+\varrho)\mathbf{I} \geq \mathbf{R}\mathbf{W}^{-1}\mathbf{R}^* &\geq (1-\varrho)\mathbf{I},\\
(1+\varrho)\mathbf{R}^{-1}(\mathbf{R}^*)^{-1} \geq \mathbf{W}^{-1} &\geq (1-\varrho)\mathbf{R}^{-1}(\mathbf{R}^*)^{-1},\\
(1+\varrho)\mathbf{A}^{-1} \geq \mathbf{W}^{-1} &\geq (1-\varrho)\mathbf{A}^{-1}.
\end{aligned}$$

Nun können wir Lemma 7.27 anwenden, um

$$\frac{1}{1+\varrho}\mathbf{A} \leq \mathbf{W} \leq \frac{1}{1-\varrho}\mathbf{A}$$

zu erhalten.

Gelte nun $\mathbf{A} < 2\mathbf{W}$. Sei $\mathcal{S} := \{\mathbf{y} \in \mathbb{R}^{\mathcal{I}} \ : \ \|\mathbf{y}\|_2 = 1\}$ wieder die Einheitssphäre. Da

$$\Lambda_{A,W} : \mathcal{S} \to \mathbb{R}, \qquad \mathbf{y} \mapsto \frac{\langle \mathbf{y}, \mathbf{W}\mathbf{y}\rangle}{\langle \mathbf{y}, \mathbf{A}\mathbf{y}\rangle},$$

wegen $\mathbf{0} < \mathbf{A}$ eine wohldefinierte stetige Abbildung von der nach dem Satz von Heine-Borel kompakten Menge $\mathcal{S}$ nach $\mathbb{R}$ ist, nimmt sie ein Minimum $\alpha \in \mathbb{R}$ und ein Maximum $\beta \in \mathbb{R}$ an. Wegen $\mathbf{A} < 2\mathbf{W}$ muss dabei $\alpha > 1/2$ gelten. Mit diesen Extremwerten gilt

$$\alpha\langle \mathbf{y}, \mathbf{A}\mathbf{y}\rangle \leq \langle \mathbf{y}, \mathbf{W}\mathbf{y}\rangle \leq \beta\langle \mathbf{y}, \mathbf{A}\mathbf{y}\rangle \qquad \text{für alle } \mathbf{y} \in \mathbb{R}^{\mathcal{I}},$$

also auch $\alpha\mathbf{A} \leq \mathbf{W} \leq \beta\mathbf{A}$. Um den ersten Teil unseres Beweises anwenden zu können, brauchen wir lediglich $\varrho \in (0,1]$ mit

$$\frac{1}{1+\varrho} \leq \alpha, \qquad \beta \leq \frac{1}{1-\varrho}$$

zu finden. Die erste Ungleichung ist äquivalent zu $\varrho \geq 1/\alpha - 1$, die zweite zu $\varrho \geq 1 - 1/\beta$, also setzen wir $\varrho := \max\{1/\alpha - 1, 1 - 1/\beta\}$, um beide zu erfüllen.

Wegen $\alpha > 1/2$ gilt $1/\alpha - 1 < 2 - 1 = 1$, und wegen $\beta \geq \alpha > 1/2$ auch $1 - 1/\beta < 1$, so dass wir insgesamt $\varrho < 1$ erhalten.

Sollte $1/\alpha - 1 \leq 0$ gelten, so folgt $1 \leq \alpha \leq \beta$, also $1 - 1/\beta \geq 0$, so dass auch $\varrho \geq 0$ sicher gestellt ist. ∎

Mit Hilfe dieses Kriteriums können wir unmittelbar die folgende Konvergenzaussage für das Jacobi-Verfahren gewinnen.

Folgerung 7.29 (Konvergenz der Jacobi-Iteration) *Es existiert ein $\theta_{\max} \in \mathbb{R}_{>0}$ derart, dass die Jacobi-Iteration für alle $\theta \in (0, \theta_{\max})$ konvergiert.*

Beweis. Sei $D \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ der Diagonalteil der Matrix $\mathbf{A}$. Da $\mathbf{A}$ positiv definit ist, sind alle Diagonalelemente echt größer als null. Sei μ das kleinste Diagonalelement, also der kleinste Eigenwert der Matrix $\mathbf{D}$, und sei λ der größte Eigenwert der Matrix $\mathbf{A}$.

Wir setzen $\theta_{\max} := \frac{2\mu}{\lambda}$. Sei $\theta \in (0, \theta_{\max})$. Dann gilt nach Lemma 7.25

$$\mathbf{A} \leq \lambda\mathbf{I} = \frac{2}{\theta_{\max}}\mu\mathbf{I} < \frac{2}{\theta}\mu\mathbf{I} \leq \frac{2}{\theta}\mathbf{D}.$$

Wegen $\mathbf{W} = \frac{1}{\theta}\mathbf{D}$ folgt die Behauptung mit Satz 7.28. ∎

Während sich die Konvergenz des Jacobi-Verfahrens für jede positiv definite Matrix sicherstellen lässt, sind für einen brauchbaren Konvergenzbeweis des Mehrgitterverfahrens zusätzliche Voraussetzungen erforderlich.

Wie bisher bezeichnen wir die Systemmatrizen auf den verschiedenen Gitterstufen mit $(\mathbf{A}_\ell)_{\ell=0}^L$ und die Prolongationsmatrizen mit $(\mathbf{p}_\ell)_{\ell=1}^L$.

Wir gehen davon aus, dass die Glättungsverfahren symmetrische lineare Iterationen mit den Matrizen $(\mathbf{W}_\ell)_{\ell=0}^L$ sind und $\mathbf{W}_0 = \mathbf{A}_0$ gilt, da auf dem gröbsten Gitter exakt gelöst werden soll.

Das *symmetrische V-Zyklus-Mehrgitterverfahren* nimmt dann die folgende Form an:

procedure smgv(ℓ);
if $\ell = 0$ **then**
  Löse $\mathbf{A}_\ell\mathbf{x}_\ell = \mathbf{b}_\ell$
else begin
  for $k = 1$ **to** ν **do** $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell + \mathbf{W}_\ell^{-1}(\mathbf{b}_\ell - \mathbf{A}_\ell\mathbf{x}_\ell)$;
  $\mathbf{d}_\ell \leftarrow \mathbf{b}_\ell - \mathbf{A}_\ell\mathbf{x}_\ell$;
  $\mathbf{x}_{\ell-1} \leftarrow \mathbf{0}$; $\mathbf{b}_{\ell-1} \leftarrow \mathbf{p}_\ell^*\mathbf{d}_\ell$;
  smgv($\ell - 1$);
  $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell + \mathbf{p}_\ell\mathbf{x}_{\ell-1}$;
  for $k = 1$ **to** ν **do** $\mathbf{x}_\ell \leftarrow \mathbf{x}_\ell + \mathbf{W}_\ell^{-1}(\mathbf{b}_\ell - \mathbf{A}_\ell\mathbf{x}_\ell)$
end

Es unterscheidet sich von dem bisher behandelten Mehrgitterverfahren dadurch, dass ein symmetrisches Glättungsverfahren zum Einsatz kommt und gleich viele Vor- und Nachglättungsschritte ausgeführt werden.

Für den Konvergenzbeweis benötigen wir die folgenden Voraussetzungen: Erstens müssen die System- und Prolongationsmatrizen die *Galerkin-Eigenschaft* besitzen, es muss also

$$\mathbf{A}_{\ell-1} = \mathbf{p}_\ell^* \mathbf{A}_\ell \mathbf{p}_\ell \qquad \text{für alle } \ell \in \{1, \dots, L\} \tag{7.21}$$

gelten. Diese Eigenschaft ergibt sich in unserem Fall unmittelbar aus der Tatsache, dass wir eine Galerkin-Diskretisierung mit geschachtelten Ansatzräumen verwenden.

Zweitens müssen die Glättungsverfahren die *Glättungseigenschaft* besitzen, es muss also

$$\mathbf{A}_\ell \leq \mathbf{W}_\ell \qquad \text{für alle } \ell \in \{1, \dots, L\} \tag{7.22}$$

gelten. Diese Eigenschaft lässt sich in der Regel durch eine geeignete Wahl eines Dämpfungsparameters sicherstellen, in unserem Modellproblem erweist sich beispielsweise $\theta = 1/2$ als ausreichend.

Drittens muss die *Approximationseigenschaft* gelten, es muss also eine Konstante $C_A \in \mathbb{R}_{>0}$ mit

$$\mathbf{A}_\ell^{-1} - \mathbf{p}_\ell \mathbf{A}_{\ell-1}^{-1} \mathbf{p}_\ell^* \leq C_A \mathbf{W}_\ell^{-1} \qquad \text{für alle } \ell \in \{1, \dots, L\} \tag{7.23}$$

existieren. Die linke Seite dieser Gleichung beschreibt, wie gut die Lösung der Grobgittergleichung die Lösung der Feingittergleichung approximiert. In unserem Fall wird sich der kleinste Eigenwert der Matrix $\mathbf{W}_\ell^{-1}$ ungefähr proportional zu der Gitterschrittweise h_ℓ^2 verhalten, so dass wir voraussetzen müssen, dass der Approximationsfehler entsprechend sinkt. Für unser Modellproblem lässt sich mit einem gewissen Aufwand nachrechnen, dass die Ungleichung mit $C_A = 4$ für alle Gitterstufen erfüllt ist.

Wir haben bereits gesehen, dass die Fehlerfortpflanzung eines linearen Iterationsverfahrens durch die Iterationsmatrix beschrieben wird, also werden wir nun diese Matrix für das V-Zyklus-Mehrgitterverfahren ermitteln. Die Iterationsmatrix auf Stufe $\ell \in \{0, \dots, L\}$ bezeichnen wir mit $\mathbf{M}_\ell$. Da wir voraussetzen, dass auf dem gröbsten Gitter exakt gelöst wird, gilt $\mathbf{M}_0 = \mathbf{0}$.

Auf feineren Gittern setzt sich die Iterationsmatrix aus der Matrix

$$\mathbf{M}_{\mathrm{gl},\ell} := \mathbf{I} - \mathbf{W}_\ell^{-1} \mathbf{A}_\ell \qquad \text{für alle } \ell \in \{0, \dots, L\}$$

des Glättungsverfahrens und der Matrix der approximativen Grobgitterkorrektur zusammen. Letztere müssen wir nun näher untersuchen.

Da die Grobgitterkorrektur nur *approximativ* erfolgt, lösen wir nicht die exakte Grobgittergleichung

$$\mathbf{A}_{\ell-1} \mathbf{x}_{\ell-1}^* = \mathbf{b}_{\ell-1},$$

sondern führen lediglich einen Schritt des Mehrgitterverfahrens auf Stufe $\ell - 1$ aus, ausgehend von dem Anfangswert $\mathbf{0}$. Nach diesem Schritt wird der Fehler durch

$$\mathbf{x}_{\ell-1}^* - \mathbf{x}_{\ell-1} = \mathbf{M}_{\ell-1}(\mathbf{x}_{\ell-1}^* - \mathbf{0})$$

gegeben sein, es wird also

$$\mathbf{x}_{\ell-1} = (\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{x}_{\ell-1}^* = (\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{A}_{\ell-1}^{-1}\mathbf{b}_{\ell-1}$$

gelten. Nach Ausführung der approximativen Grobgitterkorrektur verbleibt damit der Fehler

$$\begin{aligned}\mathbf{x}_\ell^* - (\mathbf{x}_\ell + \mathbf{p}_\ell\mathbf{x}_{\ell-1}) &= (\mathbf{x}_\ell^* - \mathbf{x}_\ell) - \mathbf{p}_\ell(\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{A}_{\ell-1}^{-1}\mathbf{b}_{\ell-1}\\ &= (\mathbf{x}_\ell^* - \mathbf{x}_\ell) - \mathbf{p}_\ell(\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*(\mathbf{b}_\ell - \mathbf{A}_\ell\mathbf{x}_\ell)\\ &= (\mathbf{x}_\ell^* - \mathbf{x}_\ell) - \mathbf{p}_\ell(\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell(\mathbf{x}_\ell^* - \mathbf{x}_\ell),\end{aligned}$$

also ist die Iterationsmatrix durch

$$\begin{aligned}\mathbf{M}_{\mathrm{gg},\ell} &:= \mathbf{I} - \mathbf{p}_\ell(\mathbf{I} - \mathbf{M}_{\ell-1})\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell\\ &:= \mathbf{I} - \mathbf{p}_\ell\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell + \mathbf{p}_\ell\mathbf{M}_{\ell-1}\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell\end{aligned}$$

gegeben. Für den vollständigen Mehrgitterschritt ergibt sich die Iterationsmatrix

$$\mathbf{M}_\ell := \begin{cases}\mathbf{0} & \text{falls } \ell = 0,\\ \mathbf{M}_{\mathrm{gl},\ell}^\nu\mathbf{M}_{\mathrm{gg},\ell}\mathbf{M}_{\mathrm{gl},\ell}^\nu & \text{ansonsten}\end{cases} \qquad \text{für alle } \ell \in \{0,\dots,L\}. \tag{7.24}$$

Um Konvergenz in der Energienorm zu zeigen, müssen wir nach Lemma 7.20 die Spektralnorm der Matrizen

$$\widehat{\mathbf{M}}_\ell := \mathbf{R}_\ell\mathbf{M}_\ell\mathbf{R}_\ell^{-1} \qquad \text{für alle } \ell \in \{0,\dots,L\}$$

beschränken, wobei $\mathbf{R}_\ell^*\mathbf{R}_\ell = \mathbf{A}_\ell$ für jedes $\ell \in \{0,\dots,L\}$ wieder die Cholesky-Zerlegung bezeichnet.

Dazu transformieren wir die Matrizen, aus denen sich $\mathbf{M}_\ell$ zusammensetzt:

$$\begin{aligned}\widehat{\mathbf{M}}_{\mathrm{gl},\ell} &:= \mathbf{R}_\ell\mathbf{M}_{\mathrm{gl},\ell}\mathbf{R}_\ell^{-1} = \mathbf{R}_\ell(\mathbf{I} - \mathbf{W}_\ell^{-1}\mathbf{A}_\ell)\mathbf{R}_\ell^{-1} = \mathbf{I} - \mathbf{R}_\ell\mathbf{W}_\ell^{-1}\mathbf{R}_\ell^* = \mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1},\\ \widehat{\mathbf{M}}_{\mathrm{gg},\ell} &:= \mathbf{R}_\ell\mathbf{M}_{\mathrm{gg},\ell}\mathbf{R}_\ell^{-1} = \mathbf{R}_\ell(\mathbf{I} - \mathbf{p}_\ell\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell + \mathbf{p}_\ell\mathbf{M}_{\ell-1}\mathbf{A}_{\ell-1}^{-1}\mathbf{p}_\ell^*\mathbf{A}_\ell)\mathbf{R}_\ell^{-1}\\ &= \mathbf{I} - \mathbf{R}_\ell\mathbf{p}_\ell\mathbf{R}_{\ell-1}^{-1}(\mathbf{R}_{\ell-1}^*)^{-1}\mathbf{p}_\ell^*\mathbf{R}_\ell + \mathbf{R}_\ell\mathbf{p}_\ell\mathbf{R}_{\ell-1}^{-1}\mathbf{R}_{\ell-1}\mathbf{M}_{\ell-1}\mathbf{R}_{\ell-1}^{-1}(\mathbf{R}_{\ell-1}^*)^{-1}\mathbf{p}_{\ell-1}^*\mathbf{R}_\ell^*\\ &= \mathbf{I} - \widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^* + \widehat{\mathbf{p}}_\ell\widehat{\mathbf{M}}_{\ell-1}\widehat{\mathbf{p}}_\ell^*,\end{aligned}$$

wobei wir die transformierte Prolongationsmatrix durch

$$\widehat{\mathbf{p}}_\ell := \mathbf{R}_\ell\mathbf{p}_\ell\mathbf{R}_{\ell-1}^{-1} \qquad \text{für alle } \ell \in \{1,\dots,L\}$$

und die transformierte Matrix des Glättungsverfahrens durch

$$\widehat{\mathbf{W}}_\ell := (\mathbf{R}_\ell^*)^{-1}\mathbf{W}_\ell\mathbf{R}_\ell^{-1} \qquad \text{für alle } \ell \in \{0,\dots,L\}$$

definieren. Aus (7.24 folgt dann

$$\widehat{\mathbf{M}}_\ell := \begin{cases}\mathbf{0} & \text{falls } \ell = 0,\\ \widehat{\mathbf{M}}_{\mathrm{gl},\ell}^\nu\widehat{\mathbf{M}}_{\mathrm{gg},\ell}\widehat{\mathbf{M}}_{\mathrm{gl},\ell}^\nu & \text{ansonsten}\end{cases} \qquad \text{für alle } \ell \in \{0,\dots,L\}. \tag{7.25}$$

Unsere Voraussetzungen (7.21), (7.22) und (7.23) lassen sich auf die transformierten Matrizen übertragen:

Lemma 7.30 (Galerkin-Eigenschaft) *Aus (7.21) folgt*

$$\mathbf{0} \leq \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* \leq \mathbf{I}, \qquad \mathbf{0} \leq \mathbf{I} - \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* \leq \mathbf{I} \qquad \textit{für alle } \ell \in \{1, \dots, L\}, \tag{7.26}$$

Beweis. Sei $\ell \in \{1, \dots, L\}$. Es gilt

$$\begin{aligned} \widehat{\mathbf{p}}_\ell^* \widehat{\mathbf{p}}_\ell &= (\mathbf{R}_{\ell-1}^*)^{-1} \mathbf{p}_\ell^* \mathbf{R}_\ell^* \mathbf{R}_\ell \mathbf{p}_\ell \mathbf{R}_{\ell-1}^{-1} = (\mathbf{R}_{\ell-1}^*)^{-1} \mathbf{p}_\ell^* \mathbf{A}_\ell \mathbf{p}_\ell \mathbf{R}_{\ell-1}^{-1} \\ &= (\mathbf{R}_{\ell-1}^*)^{-1} \mathbf{A}_{\ell-1} \mathbf{R}_{\ell-1}^{-1} = \mathbf{I} \qquad \text{für alle } \ell \in \{1, \dots, L\}. \end{aligned}$$

Aus dieser Eigenschaft folgt unmittelbar

$$(\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*)^2 = \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* = \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* \qquad \text{für alle } \ell \in \{1, \dots, L\},$$

die Matrix $\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*$ ist also eine *Projektion* (es ist sogar eine *orthogonale Projektion*). Sei $\ell \in \{1, \dots, L\}$, und sei $\mathbf{e} \in \mathbb{R}^{\mathcal{I}_\ell}$ ein Eigenvektor der Matrix $\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*$ zu einem Eigenwert $\lambda \in \mathbb{R}$. Dann gilt

$$\lambda \mathbf{e} = (\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*)\mathbf{e} = (\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*)(\widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^*)\mathbf{e} = \lambda^2 \mathbf{e},$$

also insbesondere $\lambda = \lambda^2$ und damit $\lambda \in \{0, 1\}$. Mit Lemma 7.25 folgt (7.26). ∎

Lemma 7.31 (Glättungseigenschaft) *Aus (7.22) folgt*

$$\mathbf{0} < \widehat{\mathbf{W}}_\ell^{-1} \leq \mathbf{I} \qquad \textit{für alle } \ell \in \{1, \dots, L\}. \tag{7.27}$$

Beweis. Sei $\ell \in \{1, \dots, L\}$. Mit Lemma 7.24 erhalten wir

$$\begin{aligned} \mathbf{0} &< \mathbf{A}_\ell \leq \mathbf{W}_\ell, \\ \mathbf{0} &< \mathbf{R}_\ell^* \mathbf{R}_\ell \leq \mathbf{R}_\ell^* \mathbf{W}_\ell \mathbf{R}_\ell, \\ \mathbf{0} &< \mathbf{I} \leq \widehat{\mathbf{W}}_\ell. \end{aligned}$$

Daraus folgt mit Lemma 7.27 die Behauptung. ∎

Lemma 7.32 (Approximationseigenschaft) *Aus (7.23) folgt*

$$\mathbf{I} - \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* \leq C_A \widehat{\mathbf{W}}_\ell^{-1} \qquad \textit{für alle } \ell \in \{1, \dots, L\}. \tag{7.28}$$

Beweis. Sei $\ell \in \{1, \dots, L\}$. Mit Lemma 7.24 erhalten wir

$$\begin{aligned} \mathbf{A}_\ell^{-1} - \mathbf{p}_\ell \mathbf{A}_{\ell-1}^{-1} \mathbf{p}_\ell^* &\leq C_A \mathbf{W}_\ell^{-1}, \\ \mathbf{R}_\ell \mathbf{A}_\ell^{-1} \mathbf{R}_\ell^* - \mathbf{R}_\ell \mathbf{p}_\ell \mathbf{A}_{\ell-1}^{-1} \mathbf{p}_\ell^* \mathbf{R}_\ell^* &\leq C_A \mathbf{R}_\ell \mathbf{W}_\ell^{-1} \mathbf{R}_\ell^*, \\ \mathbf{R}_\ell (\mathbf{R}_\ell^* \mathbf{R}_\ell)^{-1} \mathbf{R}_\ell^* - \mathbf{R}_\ell \mathbf{p}_\ell (\mathbf{R}_{\ell-1}^* \mathbf{R}_{\ell-1})^{-1} \mathbf{p}_\ell^* \mathbf{R}_\ell^* &\leq C_A \widehat{\mathbf{W}}_\ell^{-1}, \\ \mathbf{I} - \widehat{\mathbf{p}}_\ell \widehat{\mathbf{p}}_\ell^* &\leq C_A \widehat{\mathbf{W}}_\ell^{-1}, \end{aligned}$$

und das ist die gewünschte Abschätzung. ∎

Satz 7.33 (Konvergenz V-Zyklus) *Unter den Voraussetzungen (7.21), (7.22) und (7.23) gilt*

$$\|\mathbf{M}_\ell\|_A \leq \frac{C_A}{C_A + 2\nu} \qquad \text{für alle } \ell \in \{0, \dots, L\},\ \nu \in \mathbb{N}.$$

Beweis. Zur Abkürzung definieren wir $\varrho := C_A/(C_A + 2\nu)$. Dank Lemma 7.20 wissen wir, dass es genügt,

$$\|\mathbf{M}_\ell\|_A = \|\widehat{\mathbf{M}}_\ell\|_2 \leq \varrho \qquad \text{für alle } \ell \in \{0, \dots, L\}$$

zu beweisen. Wir zeigen im Folgenden

$$\mathbf{0} \leq \widehat{\mathbf{M}}_\ell \leq \varrho\mathbf{I} \qquad \text{für alle } \ell \in \{0, \dots, L\},$$

denn daraus folgt mit Lemma 7.25 unsere Aussage.

Wir führen den Beweis per Induktion.

Induktionsanfang: Für $\ell = 0$ gilt wegen $\mathbf{W}_0 = \mathbf{A}_0$ die Gleichung

$$\mathbf{M}_\ell = \mathbf{I} - \mathbf{W}_0^{-1}\mathbf{A}_0 = \mathbf{I} - \mathbf{A}_0^{-1}\mathbf{A}_0 = \mathbf{0},$$

also auch $\widehat{\mathbf{M}}_\ell = \mathbf{0}$.

Induktionsvoraussetzung: Sei $\ell \in \mathbb{N}$ so gewählt, dass

$$\mathbf{0} \leq \widehat{\mathbf{M}}_{\ell-1} \leq \varrho\mathbf{I}$$

gilt.

Induktionsschritt: Mit (7.25) gilt

$$\begin{aligned}\widehat{\mathbf{M}}_\ell &= \widehat{\mathbf{M}}_{\mathrm{gl},\ell}^{\nu}\widehat{\mathbf{M}}_{\mathrm{gg},\ell}\widehat{\mathbf{M}}_{\mathrm{gl},\ell}^{\nu} \\ &= (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu(\mathbf{I} - \widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^* + \widehat{\mathbf{p}}_\ell\widehat{\mathbf{M}}_{\ell-1}\widehat{\mathbf{p}}_\ell^*)(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu.\end{aligned}$$

Mit der Induktionsvoraussetzung und Lemma 7.24 haben wir

$$\mathbf{0} \leq \widehat{\mathbf{p}}_\ell\widehat{\mathbf{M}}_{\ell-1}\widehat{\mathbf{p}}_\ell^* \leq \varrho\widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^*,$$

so dass wir mit Lemma 7.24 und (7.18c) unmittelbar

$$\mathbf{0} \leq (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu(\mathbf{I} - \widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^*)(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu \leq \widehat{\mathbf{M}}_\ell$$

sowie auch

$$\begin{aligned}\widehat{\mathbf{M}}_\ell &= (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu(\mathbf{I} - \widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^* + \widehat{\mathbf{p}}_\ell\widehat{\mathbf{M}}_{\ell-1}\widehat{\mathbf{p}}_\ell^*)(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu \\ &\leq (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu(\mathbf{I} - (1-\varrho)\widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^*)(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu \\ &\leq (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu(\varrho\mathbf{I} + (1-\varrho)(\mathbf{I} - \widehat{\mathbf{p}}_\ell\widehat{\mathbf{p}}_\ell^*)(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu\end{aligned}$$

erhalten. Mit der Approximationseigenschaft (7.28) sowie Lemma 7.24 und (7.18c) folgt

$$\widehat{\mathbf{M}}_\ell \leq (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu (\varrho\mathbf{I} + (1-\varrho)C_A \widehat{\mathbf{W}}_\ell^{-1})(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu.$$

Wir stellen fest, dass es sich bei dem rechten Ausdruck um ein Polynom der Matrix $\mathbf{X} := \widehat{\mathbf{W}}_\ell^{-1}$ handelt, denn mit

$$p(x) := (1-x)^\nu (\varrho + (1-\varrho)C_A x)(1-x)^\nu$$

gilt gerade

$$p(\mathbf{X}) = (\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu (\varrho\mathbf{I} + (1-\varrho)C_A \widehat{\mathbf{W}}_\ell^{-1})(\mathbf{I} - \widehat{\mathbf{W}}_\ell^{-1})^\nu.$$

Damit sind die Eigenwerte der Matrix $p(\mathbf{X})$ gegeben durch

$$\sigma(p(\mathbf{X})) = \{p(\lambda) \ : \ \lambda \in \sigma(\mathbf{X})\}.$$

Aufgrund der Glättungseigenschaft (7.27) gilt $\mathbf{0} < \mathbf{X} \leq \mathbf{I}$, also mit Lemma 7.25 auch $\sigma(\mathbf{X}) \subseteq [0,1]$, so dass wir lediglich das Maximum des Polynoms p auf dem Intervall $[0,1]$ zu bestimmen brauchen.

Diese Aufgabe lösen wir, indem wir nach Nullstellen der Ableitung p' suchen, die nach Produktregel durch

$$\begin{aligned}
p'(x) &= -2\nu(1-x)^{2\nu-1}(\varrho + (1-\varrho)C_A x) + (1-x)^{2\nu}(1-\varrho)C_A \\
&= (1-x)^{2\nu-1}((1-x)(1-\varrho)C_A - 2\nu(\varrho + (1-\varrho)C_A x)) \\
&= (1-x)^{2\nu-1}(C_A - \varrho C_A - 2\nu\varrho - ((1-\varrho)C_A + 2\nu(1-\varrho)C_A)x) \\
&= (1-x)^{2\nu-1}(C_A - \varrho(C_A + 2\nu) - ((1-\varrho)C_A + 2\nu(1-\varrho)C_A)x) \\
&= (1-x)^{2\nu-1}(0 - ((1-\varrho)C_A + 2\nu(1-\varrho)C_A)x) \\
&= -(1-x)^{2\nu-1}((1-\varrho)C_A + 2\nu(1-\varrho)C_A)x
\end{aligned}$$

gegeben ist. Offenbar besitzt sie nur die Nullstellen 0 und 1, also kann p in $[0,1]$ nur in den beiden Randpunkten sein Maximum annehmen. Es folgt

$$\max\{p(\lambda) \ : \ \lambda \in [0,1]\} = \max\{p(0), p(1)\} = \max\{\varrho, 0\} = \varrho.$$

Mit Lemma 7.25 ergibt sich daraus schließlich

$$\widehat{\mathbf{M}}_\ell \leq p(\mathbf{X}) \leq \varrho\mathbf{I}.$$

■

7.5 Strukturmechanik*

Ein wichtiges Anwendungsgebiet der Finite-Elemente-Methode ist die Simulation strukturmechanischer Phänomene, beispielsweise der Statik eines Gebäudes. Bei derartigen

Anwendungen ist es von großer Bedeutung, kompliziert geformte Gebiete mit unterschiedlichen Materialeigenschaften zuverlässig im Computer nachbilden zu können, deshalb ist die Methode der finiten Elemente ideal geeignet. Ich orientiere mich in diesem Kapitel an dem Buch [2].

Wir interessieren uns dafür, wie sich ein elastischer Körper unter Einwirkung einer Kraft verformt. Die Grundlage eines mathematischen Modells für diese elastische Deformation ist eine verallgemeinerte Form des Federgesetzes von Hooke, das wir in seiner einfachsten Form bereits in der Gleichung (2.18) kennengelernt haben.

Der Körper im Ruhezustand, also ohne einwirkende Kräfte, wird durch ein Gebiet $\Omega \subseteq \mathbb{R}^3$ dargestellt, das einfach alle Punkte aufnimmt, die im Körper liegen. Eine Verformung des Körpers beschreiben wir durch die *Verschiebung*

$$u : \Omega \to \mathbb{R}^3,$$

die angibt, um wieviel jeder Punkt des Körpers gegenüber dem Ruhezustand verschoben ist. Dem Punkt $x \in \Omega$ im Ruhezustand wird also der Punkt $x + u(x)$ im verformten Körper zugeordnet.

Im Fall des Federgesetzes haben wir gesehen, dass die Kraft, mit der die Feder sich einer Auslenkung aus dem Ruhezustand entgegenwirkt, proportional zu der *relativen* Auslenkung ist. Im Grenzwert würden wir zu der Ableitung der Auslenkung gelangen, und diese Ableitung der Auslenkung ist auch für die Verallgemeinerung auf elastische Körper sinnvoll: Man kann für jeden Punkt des Körpers eine *Verzerrungsmatrix* definieren, die beschreibt, wie der Körper in der Nähe dieses Punkts verformt wurde. Allerdings hängt diese Matrix nichtlinear von der Verschiebung ab, so dass sie für die von uns bisher behandelten Verfahren weniger gut geeignet ist. Glücklicherweise ist man bei strukturmechanischen Anwendungen in der Regel nur an kleinen Verschiebungen interessiert, und für solche Verschiebungen ist die durch

$$\epsilon_{ij}(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad \text{für alle } i,j \in \{1,2,3\}$$

gegebene *symmetrische Ableitung* $\epsilon(u) \in \mathbb{R}^{3\times 3}$ eine geeignete Approximation der Verzerrung.

Ein elastisches Material reagiert auf eine Verzerrung, indem es eine Kraft ausübt, die der Verzerrung entgegenwirkt. Diese Kraft wird über die *Spannungsmatrix* $\sigma(u)$ mit Hilfe der Gleichung

$$-\nabla \cdot \sigma(u(x)) = f(x) \qquad \text{für alle } x \in \Omega \tag{7.29}$$

ausgedrückt. Die Rolle der Federkonstante, die die Beziehung zwischen der Verzerrung und der Spannung beschreibt, übernimmt der *Steifigkeitstensor* $C \in \mathbb{R}^{3\times 3\times 3\times 3}$ gemäß

$$\sigma_{ij}(u(x)) = \sum_{k,\ell=1}^{3} c_{ijk\ell}\epsilon_{k\ell}(u(x)) \qquad \text{für alle } x \in \Omega.$$

Ein besonders einfacher Sonderfall ist die verallgemeinerte Form des Federgesetzes von Hooke, das durch die Gleichung

$$\sigma(u(x)) = \frac{E}{1+\nu}\left(\epsilon(u(x)) + \frac{\nu}{1-2\nu}\mathbf{I}\,\mathrm{spur}(\epsilon(u(x)))\right) \qquad \text{für alle } x \in \Omega,$$

beschrieben ist. Hier heißt $E \in \mathbb{R}_{>0}$ der *Elastizitätsmodul* (nicht „das“) und $\nu \in (0, 1/2)$ die *Querkontraktion*, während

$$\mathrm{spur}(\mathbf{X}) = x_{11} + x_{22} + x_{33} \qquad \text{für alle } \mathbf{X} \in \mathbb{R}^{3\times 3}$$

die übliche Spur einer Matrix bezeichnet. Zur Abkürzung führen wir

$$\mu := \frac{E}{1+\nu}, \qquad \lambda := \frac{E}{1+\nu}\frac{\nu}{1-2\nu}$$

ein und erhalten

$$\sigma(u(x)) = \mu\epsilon(u(x)) + \lambda\mathbf{I}\,\mathrm{spur}(\epsilon(u(x)) \qquad \text{für alle } x \in \Omega.$$

Um zu einer Differentialgleichung in der bisherigen Form zu gelangen berechnen wir die einzelnen Komponenten dieser Gleichung: Für die Spur erhalten wir

$$\begin{aligned}\mathrm{spur}(\epsilon(u(x))) &= \epsilon_{11}(u(x)) + \epsilon_{22}(u(x)) + \epsilon_{33}(u(x))\\ &= \frac{1}{2}\left(2\frac{\partial u_1}{\partial x_1}(x) + 2\frac{\partial u_2}{\partial x_2}(x) + 2\frac{\partial u_3}{\partial x_3}(x)\right)\\ &= \nabla\cdot u(x) \qquad \text{für alle } x \in \Omega,\end{aligned}$$

so dass sich für die Divergenz des rechten Terms

$$\nabla\cdot(\mathbf{I}\,\mathrm{spur}(\epsilon(u(x))) = \begin{pmatrix}\frac{\partial}{\partial x_1}\nabla\cdot u(x)\\ \frac{\partial}{\partial x_2}\nabla\cdot u(x)\\ \frac{\partial}{\partial x_3}\nabla\cdot u(x)\end{pmatrix} = \nabla(\nabla\cdot u)(x)$$

ergibt. Für die Divergenz der symmetrischen Ableitung erhalten wir

$$\begin{aligned}\nabla\cdot\epsilon(x) &= \frac{1}{2}\begin{pmatrix}\frac{\partial}{\partial x_1}2\frac{\partial u_1}{\partial x_1}(x) + \frac{\partial}{\partial x_2}\left(\frac{\partial u_1}{\partial x_2}(x) + \frac{\partial u_2}{\partial x_1}(x)\right) + \frac{\partial}{\partial x_3}\left(\frac{\partial u_1}{\partial x_3}(x) + \frac{\partial u_3}{\partial x_1}(x)\right)\\ \frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_2}(x) + \frac{\partial u_2}{\partial x_1}(x)\right) + \frac{\partial}{\partial x_2}2\frac{\partial u_2}{\partial x_2}(x) + \frac{\partial}{\partial x_3}\left(\frac{\partial u_2}{\partial x_3}(x) + \frac{\partial u_3}{\partial x_2}(x)\right)\\ \frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_3}(x) + \frac{\partial u_3}{\partial x_1}(x)\right) + \frac{\partial}{\partial x_2}\left(\frac{\partial u_2}{\partial x_3}(x) + \frac{\partial u_3}{\partial x_2}(x)\right) + \frac{\partial}{\partial x_3}2\frac{\partial u_3}{\partial x_3}(x)\end{pmatrix}\\ &= \frac{1}{2}\begin{pmatrix}\frac{\partial^2 u_1}{\partial x_1^2}(x) + \frac{\partial^2 u_1}{\partial x_2^2}(x) + \frac{\partial^2 u_1}{\partial x_3^2}(x) + \frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}(x) + \frac{\partial u_2}{\partial x_2}(x) + \frac{\partial u_3}{\partial x_3}(x)\right)\\ \frac{\partial^2 u_2}{\partial x_1^2}(x) + \frac{\partial^2 u_2}{\partial x_2^2}(x) + \frac{\partial^2 u_2}{\partial x_3^2}(x) + \frac{\partial}{\partial x_2}\left(\frac{\partial u_1}{\partial x_1}(x) + \frac{\partial u_2}{\partial x_2}(x) + \frac{\partial u_3}{\partial x_3}(x)\right)\\ \frac{\partial^2 u_3}{\partial x_1^2}(x) + \frac{\partial^2 u_3}{\partial x_2^2}(x) + \frac{\partial^2 u_3}{\partial x_3^2}(x) + \frac{\partial}{\partial x_3}\left(\frac{\partial u_1}{\partial x_1}(x) + \frac{\partial u_2}{\partial x_2}(x) + \frac{\partial u_3}{\partial x_3}(x)\right)\end{pmatrix}\\ &= \frac{1}{2}\begin{pmatrix}\nabla\cdot\nabla u_1(x) + \frac{\partial}{\partial x_1}\nabla\cdot u(x)\\ \nabla\cdot\nabla u_2(x) + \frac{\partial}{\partial x_2}\nabla\cdot u(x)\\ \nabla\cdot\nabla u_3(x) + \frac{\partial}{\partial x_3}\nabla\cdot u(x)\end{pmatrix} = \frac{1}{2}\begin{pmatrix}\nabla\cdot\nabla u_1(x)\\ \nabla\cdot\nabla u_2(x)\\ \nabla\cdot\nabla u_3(x)\end{pmatrix} + \frac{1}{2}\nabla(\nabla\cdot u)(x)\end{aligned}$$

so dass die Gleichung (7.29) die Form

$$-\frac{\mu}{2}\begin{pmatrix}\nabla\cdot\nabla u_1(x)\\ \nabla\cdot\nabla u_2(x)\\ \nabla\cdot\nabla u_3(x)\end{pmatrix}-\frac{\mu+2\lambda}{2}\nabla(\nabla\cdot u)(x)=f(x)\qquad\text{für alle } x\in\Omega$$

annimmt. Diese Gleichung ist in der Literatur unter dem Namen *Lamé-Gleichung* oder *Navier-Cauchy-Gleichung* bekannt.

Wie bei der Potentialgleichung (4.15) müssen wir Randbedingungen einführen, um die Eindeutigkeit einer Lösung sicher zu stellen. Der Einfachheit halber gehen wir davon aus, dass alle Randpunkte sich nicht verschieben lassen, dass also

$$u(x)=0\qquad\text{für alle } x\in\partial\Omega$$

gilt. Damit steht uns eine partielle Differentialgleichung zur Verfügung, mit deren Hilfe wir zu jeder gegebenen Kraft die im Körper auftretenden Verschiebungen ermitteln können. Dabei ist allerdings zu beachten, dass die Gleichung nur für *kleine* Verschiebungen gilt, da beispielsweise schon die symmetrische Ableitung nur eine Näherung der Verzerrungsmatrix ist.

Variationsformulierung. Finite-Elemente-Verfahren beruhen auf einer Variationsformulierung der Differentialgleichung. Bei deren Herleitung gehen wir wie bei der Potentialgleichung vor: Wir multiplizieren die Gleichung mit einer Testfunktion $v\in C_0^1(\Omega,\mathbb{R}^3)$, diesmal im Skalarprodukt, und integrieren über Ω, um

$$\begin{aligned}&-\frac{\mu}{2}\int_\Omega\left\langle\begin{pmatrix}v_1(x)\\ v_2(x)\\ v_3(x)\end{pmatrix},\begin{pmatrix}\nabla\cdot\nabla u_1(x)\\ \nabla\cdot\nabla u_2(x)\\ \nabla\cdot\nabla u_3(x)\end{pmatrix}\right\rangle_2 dx\\ &\quad-\frac{\mu+2\lambda}{2}\int_\Omega\langle v(x),\nabla(\nabla\cdot u)(x)\rangle_2\,dx=\int_\Omega\langle v(x),f(x)\rangle_2\,dx\qquad\text{für alle } v\in C_0^1(\Omega,\mathbb{R}^3)\end{aligned}$$

zu erhalten. Für den ersten Term erhalten wir mit partieller Integration gemäß der Gleichung (6.6) den Ausdruck

$$\begin{aligned}&-\frac{\mu}{2}\int_\Omega\left\langle\begin{pmatrix}v_1(x)\\ v_2(x)\\ v_3(x)\end{pmatrix},\begin{pmatrix}\nabla\cdot\nabla u_1(x)\\ \nabla\cdot\nabla u_2(x)\\ \nabla\cdot\nabla u_3(x)\end{pmatrix}\right\rangle_2 dx\\ &\qquad=-\frac{\mu}{2}\int_\Omega v_1(x)\nabla\cdot\nabla u_1(x)+v_2(x)\nabla\cdot\nabla u_2(x)+v_3(x)\nabla\cdot\nabla u_3(x)\,dx\\ &\qquad=\frac{\mu}{2}\int_\Omega\langle\nabla v_1(x),\nabla u_1(x)\rangle_2+\langle\nabla v_2(x),\nabla u_2(x)\rangle_2+\langle\nabla v_3(x),\nabla u_3(x)\rangle_2\,dx,\end{aligned}$$

während sich für den zweiten Term die Gleichung

$$-\frac{\mu+2\lambda}{2}\int_\Omega\langle v(x),\nabla(\nabla\cdot u)(x)\rangle_2\,dx=\frac{\mu+2\lambda}{2}\int_\Omega(\nabla\cdot v)(x)(\nabla\cdot u)(x)\,dx$$

ergibt. In beiden Fällen haben wir ausgenutzt, dass $v|_{\partial\Omega} = 0$ gilt, so dass alle Randterme wegfallen. Indem wir die Bilinearform

$$a(v,u) := \frac{\mu}{2}\int_\Omega \sum_{k=1}^{3} \langle \nabla v_k(x), \nabla u_k(x)\rangle_2 \, dx + \frac{\mu+2\lambda}{2}\int_\Omega (\nabla\cdot v)(x)(\nabla\cdot u)(x)\,dx \qquad (7.30)$$
$$\text{für alle } u,v \in H_0^1(\Omega,\mathbb{R}^3)$$

und das Funktional

$$\beta(v) := \int_\Omega \langle v(x), f(x)\rangle_2 \, dx \qquad \text{für alle } v \in H_0^1(\Omega,\mathbb{R}^3)$$

definieren erhalten wir das folgende Variationsproblem:

Finde $u \in H_0^1(\Omega,\mathbb{R}^3)$ mit

$$a(v,u) = \beta(v) \qquad \text{für alle } v \in H_0^1(\Omega,\mathbb{R}^3).$$

Elliptizität. Mit Hilfe der Cauchy-Schwarz-Ungleichung lässt sich sehr einfach nachrechnen, dass die Bilinearform a stetig auf $H_0^1(\Omega,\mathbb{R}^3)$ ist. Um den Lösbarkeitssatz 6.17 von Lax-Milgram anwenden zu können, benötigen wir allerdings auch die Koerzivität. In unserem Fall können wir sie relativ einfach erhalten: Nach Definition gilt wegen $\mu > 0$ schon

$$a(v,v) \geq \frac{\mu}{2}\sum_{k=1}^{3} \|\nabla v_k\|_{L^2}^2 \qquad \text{für alle } v \in H_0^1(\Omega,\mathbb{R}^3),$$

so dass wir mit der Friedrichs-Ungleichung (vgl. Erinnerung 6.20)

$$a(v,v) \geq \frac{\mu}{2(C_\Omega+1)}\sum_{k=1}^{3} \|v_k\|_{H^1}^2 = \frac{\mu}{2(C_\Omega+1)}\|v\|_{H^1}^2 \qquad \text{für alle } v \in H_0^1(\Omega,\mathbb{R}^3)$$

erhalten, also die gewünschte Koerzivität.

Wir können allerdings auch direkt mit der symmetrischen Ableitung argumentieren: Zur Abkürzung führen wir das *Frobenius-Skalarprodukt*

$$\langle \mathbf{X}, \mathbf{Y}\rangle_F := \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij}y_{ij} \qquad \text{für alle } \mathbf{X},\mathbf{Y} \in \mathbb{R}^{3\times 3}$$

ein und untersuchen die Größe

$$\begin{aligned}\langle \epsilon(v(x)), \epsilon(u(x))\rangle_F &= \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j}(x) + \frac{\partial v_j}{\partial x_i}(x)\right)\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}(x) + \frac{\partial u_j}{\partial x_i}(x)\right)\\ &= \frac{1}{4}\sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial v_i}{\partial x_j}(x)\frac{\partial u_i}{\partial x_j}(x) + \frac{\partial v_j}{\partial x_i}(x)\frac{\partial u_j}{\partial x_i}(x)\end{aligned}$$

$$
\begin{aligned}
&\quad + \frac{\partial v_i}{\partial x_j}(x)\frac{\partial u_j}{\partial x_i}(x) + \frac{\partial v_j}{\partial x_i}(x)\frac{\partial u_i}{\partial x_j}(x) \\
&= \frac{1}{2}\sum_{i=1}^{3}\langle \nabla v_i(x), \nabla u_i(x)\rangle_2 + \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\partial v_i}{\partial x_j}(x)\frac{\partial u_j}{\partial x_i}(x).
\end{aligned}
$$

Für zweimal stetig differenzierbare Funktionen u und v können wir mit partieller Integration die Gleichung

$$
\int_\Omega \frac{\partial v_i}{\partial x_j}(x)\frac{\partial u_j}{\partial x_i}(x)\,dx = -\int_\Omega v_i(x)\frac{\partial^2 u_j}{\partial x_i \partial x_j}(x)\,dx = \int_\Omega \frac{\partial v_i}{\partial x_i}(x)\frac{\partial u_j}{\partial x_j}(x)\,dx
$$

erhalten, die sich auch auf $u, v \in H_0^1(\Omega, \mathbb{R}^3)$ überträgt, so dass sich insgesamt

$$
\begin{aligned}
&\int_\Omega \langle \epsilon(v(x)), \epsilon(u(x))\rangle_F\,dx \\
&\qquad = \frac{1}{2}\int_\Omega \sum_{i=1}^{3}\langle \nabla v_i(x), \nabla u_i(x)\rangle_2\,dx + \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega \frac{\partial v_i}{\partial x_j}(x)\frac{\partial u_j}{\partial x_i}(x)\,dx \\
&\qquad = \frac{1}{2}\int_\Omega \sum_{i=1}^{3}\langle \nabla v_i(x), \nabla u_i(x)\rangle_2\,dx + \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega \frac{\partial v_i}{\partial x_i}(x)\frac{\partial u_j}{\partial x_j}(x)\,dx \\
&\qquad = \frac{1}{2}\int_\Omega \sum_{i=1}^{3}\langle \nabla v_i(x), \nabla u_i(x)\rangle_2\,dx + \frac{1}{2}\int_\Omega (\nabla\cdot v)(x)(\nabla\cdot u)(x)\,dx
\end{aligned}
$$

ergibt. Ein Vergleich mit der Definition (7.30) der Bilinearform führt zu der alternativen Darstellung

$$
\begin{aligned}
a(v,u) = \mu\int_\Omega \langle \epsilon(v(x)), \epsilon(u(x))\rangle_F\,dx + \lambda\int_\Omega (\nabla\cdot v)(x)(\nabla\cdot u)(x)\,dx \\
\text{für alle } u, v \in H_0^1(\Omega, \mathbb{R}^3).
\end{aligned}
$$

Für die symmetrische Ableitung $\epsilon(u)$ gibt es ein Gegenstück der Friedrichs-Ungleichung:

Erinnerung 7.34 (Korn-Ungleichung) *Es existiert ein $C'_\Omega \in \mathbb{R}_{>0}$ mit*

$$
\|v\|_{H^1}^2 \le C'_\Omega \|\epsilon(v)\|_{L^2}^2 \qquad \text{für alle } v \in H_0^1(\Omega, \mathbb{R}^3),
$$

wobei die L^2-Norm der matrixwertigen Funktion $\epsilon(v)$ per Frobenius-Skalarprodukt definiert ist.

Mit unserer alternativen Darstellung der Bilinearform folgt aus der Korn-Ungleichung unmittelbar

$$
a(v,v) \ge \mu\|\epsilon(v)\|_{L^2}^2 \ge \frac{\mu}{C'_\Omega}\|v\|_{H^1}^2 \qquad \text{für alle } v \in H_0^1(\Omega, \mathbb{R}^3),
$$

also ist die Koerzivität auch für die alternative Darstellung der Bilinearform gegeben.

Ansatzraum. Die Variationsformulierung für die Lamé-Gleichung unterscheidet sich von der für die Potentialgleichung dadurch, dass u und v vektorwertige Funktionen sind. Deshalb benötigen wir für die Finite-Elemente-Diskretisierung auch einen Ansatzraum, der sich aus solchen Funktionen zusammensetzt.

Die einfachste Idee besteht darin, die Knotenbasisfunktionen, die wir schon für die Potentialgleichung verwendet haben, einfach zu vektorwertigen Funktionen zu erweitern, indem wir die Funktionswerte mit einigen Nulleinträgen in Vektoren verwandeln: Wir definieren

$$\varphi_{1,i}(x) = \begin{pmatrix} \varphi_i(x) \\ 0 \\ 0 \end{pmatrix}, \quad \varphi_{2,i}(x) = \begin{pmatrix} 0 \\ \varphi_i(x) \\ 0 \end{pmatrix}, \quad \varphi_{3,i}(x) = \begin{pmatrix} 0 \\ 0 \\ \varphi_i(x) \end{pmatrix} \qquad \text{für alle } i \in \mathcal{I}.$$

Die Basis unseres Ansatzraums ist dann durch $(\varphi_{k,i})_{(k,i)\in\mathcal{I}_3}$ mit der erweiterten Indexmenge $\mathcal{I}_3 := \{1,2,3\} \times \mathcal{I}$ gegeben.

Die Berechnung der Einträge der zugehörigen Matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I}_3 \times \mathcal{I}_3}$ gestaltet sich in diesem Fall besonders einfach: Für $(k,i),(\ell,j) \in \mathcal{I}_3$ gilt im Fall $k = \ell$ gerade

$$a_{ik,j\ell} = \frac{\mu}{2} \int_\Omega \langle \nabla\varphi_i(x), \nabla\varphi_j(x) \rangle_2 \, dx + \frac{\mu + 2\lambda}{2} \int_\Omega \frac{\partial\varphi_i}{\partial x_k}(x) \frac{\partial\varphi_j}{\partial x_k}(x) \, dx,$$

während für $k \neq \ell$ der erste Term entfällt, da die k-te Komponente der Funktion $\varphi_{\ell,j}$ verschwindet, also auch ihr Gradient. Damit bleibt nur

$$a_{ik,j\ell} = \frac{\mu + 2\lambda}{2} \int_\Omega \frac{\partial\varphi_i}{\partial x_k}(x) \frac{\partial\varphi_j}{\partial x_\ell}(x) \, dx$$

übrig. Wir wissen bereits (vgl. Übungsaufgabe 6.34), wie sich die Gradienten $\nabla\varphi_i$ und $\nabla\varphi_j$ berechnen lassen. Dank $\frac{\partial\varphi_i}{\partial x_k} = (\nabla\varphi_i)_k$ stehen uns damit alle Größen zur Verfügung, die wir für die Berechnung der Matrixeinträge benötigen.

Außerdem dürfen wir den Gleichungen entnehmen, dass $a_{ik,j\ell}$ nur dann ungleich null sein kann, falls sich die Träger der Basisfunktionen φ_i und φ_j überschneiden. Da es sich um Knotenbasisfunktionen handelt, ist das genau dann der Fall, wenn ein Element $t \in \mathcal{T}$ der Triangulation mit $i,j \in t$ existiert, wenn also i und j durch eine Kante der Triangulation verbunden sind. Damit lässt sich ähnlich einfach wie bei der Potentialgleichung vorhersagen, an welchen Stellen der Matrix $\mathbf{A}$ von null verschiedene Einträge auftreten können.

Wie bei der Potentialgleichung können wir die Matrix aufstellen, indem wir alle Elemente $t \in \mathcal{T}$ durchlaufen, die Gradienten der Basisfunktionen auf ω_t berechnen, die Integrale auswerten und zu den korrespondierenden Koeffizienten der Matrix $\mathbf{A}$ addieren.

Für den Vektor $\mathbf{b} \in \mathbb{R}^{\mathcal{I}_3}$ erhalten wir

$$b_{ik} = \int_\Omega \varphi_i(x) f_k(x) \, dx,$$

so dass sich die einzelnen Einträge wie bei der Potentialgleichung berechnen lassen, indem wir die einzelnen Koordinaten $k \in \{1,2,3\}$ separat behandeln.

7.6 Grundwasserströmung *

In Abschnitt 5.1 haben wir uns bereits mit der Frage beschäftigt, wie sich Grundwasserströmungen simulieren lassen. Das in diesem Abschnitt eingeführte Finite-Differenzen-Verfahren hat allerdings den Nachteil, sich nicht gut auf allgemeine Geometrien übertragen zu lassen. Außerdem berücksichtigt es nicht, dass bei vielen Aufgabenstellungen aus dem Bereich der Geophysik das Gebiet nicht aus einem einheitlichen Material besteht, sondern beispielsweise aus Schichten unterschiedlicher Materialien, beispielsweise Sand und Fels, die unterschiedliche Eigenschaften aufweisen.

Modell. Der Finite-Elemente-Ansatz bietet uns die Möglichkeit, diese Aufgabenstellung wesentlich eleganter und allgemeiner zu behandeln. Die Grundlage unserer Betrachtung bildet wieder das aus (5.1) bekannte Darcy'sche Gesetz

$$u(x) + k(x)\nabla p(x) = 0 \qquad \text{für alle } x \in \Omega,$$

bei dem wir den Fluss nun mit $u : \Omega \to \mathbb{R}^3$ und den Druck mit $p : \Omega \to \mathbb{R}$ bezeichnen. $k : \Omega \to \mathbb{R}$ ist weiterhin ein Materialparameter, der die Durchlässigkeit des Bodens in jedem Punkt des Gebiets beschreibt.

Hinzu kommt das bereits aus (5.2) bekannte Prinzip der Massenerhaltung

$$\int_{\partial\omega} \langle u(x), n(x)\rangle_2 \, dx = 0 \qquad \text{für alle Gebiete } \omega \subseteq \Omega.$$

Hier ist $n : \partial\omega \to \mathbb{R}^3$ wieder der äußere Einheitsnormalenvektor des Gebiets ω.

Variationsformulierung. Beide Gesetze werden wir nun äquivalent umformulieren, um zu einer für unsere Zwecke geeigneten Variationsformulierung zu gelangen.

Im Fall des Darcy'schen Gesetzes genügt es, f auf die rechte Seite zu bringen, mit einer Testfunktion $v \in C_0^1(\Omega, \mathbb{R}^3)$ zu multiplizieren und partiell zu integrieren, um

$$\begin{aligned} u(x) + k(x)\nabla p(x) &= 0 \\ \nabla p(x) &= -\frac{u(x)}{k(x)} \\ \int_\Omega \langle v(x), \nabla p(x)\rangle_2 \, dx &= -\int_\Omega \frac{\langle v(x), u(x)\rangle_2}{k(x)} \, dx \\ -\int_\Omega \nabla \cdot v(x) p(x) \, dx &= -\int_\Omega \frac{\langle v(x), u(x)\rangle_2}{k(x)} \, dx \end{aligned} \tag{7.31}$$

zu erhalten. Wir stellen fest, dass in dieser Formulierung der Druck p nicht differenzierbar zu sein braucht.

Aus der Gleichung für die Massenerhaltung folgt mit Hilfe des Gauß'schen Integralsatzes (vgl. Erinnerung 6.4 und die Motivation der Definition 4.6), dass

$$\nabla \cdot u(x) = 0 \qquad \text{für alle } x \in \Omega$$

gelten muss. Um zu einer Variationsformulierung zu gelangen, multiplizieren wir auch diese Gleichung mit einer Testfunktion $q \in C_0^1(\Omega)$ und erhalten

$$\int_\Omega q(x) \nabla \cdot u(x)\, dx = 0.$$

Ein Vergleich mit (7.31) zeigt, dass auf der linken Seite beider Gleichungen ein Integral über das Produkt aus einer skalarwertigen Funktion und der Divergenz einer zweiten Funktion auftritt, also bietet es sich an, eine passende Bilinearform zu definieren.

Bilinearformen. Vorläufig setzen wir deshalb

$$b(v,p) := -\int_\Omega \nabla \cdot v(x) p(x)\, dx \qquad \text{für alle } v \in C^1(\Omega, \mathbb{R}^3),\ p \in C(\Omega)$$

und schreiben die Massenerhaltung in der Form

$$b(u,q) = 0 \qquad \text{für alle } q \in C(\Omega).$$

Für die Gleichung (7.31) benötigen wir noch eine Bilinearform für die rechte Seite, die wir vorläufig als

$$a(v,u) := \int_\Omega \frac{\langle v(x), u(x) \rangle_2}{k(x)}\, dx \qquad \text{für alle } v, u \in C^1(\Omega, \mathbb{R}^3)$$

definieren und so zu

$$a(v,u) + b(v,p) = 0$$

gelangen. Damit erhalten wir insgesamt das System

$$\begin{aligned} a(v,u) + b(v,p) &= 0 && \text{für alle } v \in C_0^1(\Omega, \mathbb{R}^3), \\ b(u,q) &= 0 && \text{für alle } q \in C(\Omega). \end{aligned}$$

Um eine nicht-triviale Lösung zu erhalten müssen wir noch die Randbedingungen einfließen lassen. Dazu bietet es sich an, eine Funktion $u_D \in C^1(\Omega, \mathbb{R}^3)$ zu wählen, die die richtigen Randwerte aufweist, und dann die Lösung in der Form $u + u_D$ mit $u \in C_0^1(\Omega, \mathbb{R}^3)$ zu schreiben. Indem wir den bekannten Wert auf die rechte Seite bringen ergibt sich

$$\begin{aligned} a(v,u) + b(v,p) &= -a(v,u_D) && \text{für alle } v \in C_0^1(\Omega, \mathbb{R}^3), \\ b(u,q) &= -b(u_D,q) && \text{für alle } q \in C(\Omega). \end{aligned}$$

Die Struktur dieser Aufgabe ähnelt der der Gleichung (5.3), die wir bei dem Finite-Differenzen-Ansatz erhalten haben.

Sobolew-Räume. Wie schon bei der Behandlung der Potentialgleichung stellt sich heraus, dass Hilbert-Räume für die Untersuchung dieses Variationsproblems wesentlich besser geeignet sind als die Räume der stetig differenzierbaren Funktionen, die wir bei unserer Herleitung verwendet haben. Ein Blick auf die Bilinearformen a und b zeigt, dass nichts dagegen sprechen würde, die Funktionen p und q aus dem Raum $L^2(\Omega)$ der quadratintegrablen Funktionen zu wählen. Interessanter sind die Funktionen u und v, deren Divergenz berechnet werden muss. Da nur die Divergenz benötigt wird, aber keine weiteren Ableitungen, empfiehlt es sich, einen passenden Sobolew-Raum zu konstruieren.

Für differenzierbare $u \in C^1(\Omega, \mathbb{R}^3)$ und $v \in C_0^1(\Omega)$ können wir gemäß (6.6) partiell integrieren und erhalten

$$\int_\Omega v(x) \nabla \cdot u(x)\, dx = -\int_\Omega \langle \nabla v(x), u(x) \rangle_2\, dx,$$

also bietet es sich an, diese Gleichung auch für die Definition einer verallgemeinerten Divergenz heranzuziehen:

Definition 7.35 (Schwache Divergenz) *Sei $u \in L^2(\Omega, \mathbb{R}^3)$. Falls ein $w \in L^2(\Omega)$ mit*

$$\int_\Omega v(x) w(x)\, dx = -\int_\Omega \langle \nabla v(x), u(x) \rangle_2\, dx \qquad \text{für alle } v \in C_0^1(\Omega) \tag{7.32}$$

existiert, sagen wir, dass u eine schwache Divergenz *besitzt, die wir mit $\nabla \cdot u := w$ bezeichnen.*

Den Raum aller Funktionen mit schwacher Divergenz auf Ω bezeichnen wir mit

$$H(\mathrm{div}, \Omega) := \{u \in L^2(\Omega, \mathbb{R}^3) \ : \ u \text{ besitzt eine schwache Divergenz}\}.$$

Mit der durch

$$\|u\|_{H(\mathrm{div})} := \sqrt{\|u\|_{L^2}^2 + \|\nabla \cdot u\|_{L^2}^2} \qquad \text{für alle } u \in H(\mathrm{div}, \Omega)$$

definierten Norm ist er ein Banach-Raum und mit dem durch

$$\langle v, u \rangle_{H(\mathrm{div})} := \langle v, u \rangle_{L^2} + \langle \nabla \cdot v, \nabla \cdot u \rangle_{L^2} \qquad \text{für alle } u, v \in H(\mathrm{div}, \Omega)$$

definierten Skalarprodukt auch ein Hilbert-Raum.

Falls u im üblichen Sinn differenzierbar ist, stimmt die übliche Divergenz mit der schwachen Divergenz überein. Es gibt allerdings Funktionen, die eine schwache Divergenz besitzen, aber keine klassische.

Spuroperator. Die Anforderungen an eine Funktion $u \in H(\mathrm{div}, \Omega)$ sind deutlich schwächer als an eine Funktion aus $H^1(\Omega, \mathbb{R}^3)$, beispielsweise ist die Einschränkung einer solchen Funktion auf den Rand des Gebiets nicht mehr wohldefiniert. Allerdings

lassen sich wenigstens bestimmte Anteile der Funktion noch auf den Rand einschränken: Für ein beliebiges $x \in \partial\Omega$ ist $n(x)$ ein Einheitsvektor, so dass

$$u(x) = \langle n(x), u(x)\rangle_2 n(x) + (u(x) - \langle n(x), u(x)\rangle_2 n(x))$$

gilt. Der erste Summand auf der rechten Seite verläuft in Richtung des Normalenvektors, wir bezeichnen ihn als dessen *Normalenkomponente*. Der zweite Summand steht senkrecht auf dem Normalenvektor, wir nennen ihn die *Tangentialkomponente*.

Es lässt sich nun nachweisen, dass die Normalenkomponente einer Funktion aus $H(\operatorname{div}, \Omega)$ sich noch sinnvoll definieren lässt, dass also die Funktion

$$\langle n, u\rangle_2 : \partial\Omega \to \mathbb{R}, \qquad x \mapsto \langle n(x), u(x)\rangle_2,$$

sich in einem geeigneten Sinn von $u \in C(\Omega, \mathbb{R}^3)$ auf $u \in H(\operatorname{div}, \Omega)$ übertragen lässt.

Deshalb dürfen wir den Raum

$$H_0(\operatorname{div}, \Omega) := \{u \in H(\operatorname{div}, \Omega) \ : \ \langle n, u\rangle_2 = 0\}$$

einführen, der alle Funktionen aus $H(\operatorname{div}, \Omega)$ aufnimmt, deren Normalenkomponente auf dem Rand des Gebiets verschwindet. Diesen Raum verwenden wir als Verallgemeinerung des Raums $C_0^1(\Omega, \mathbb{R}^3)$.

Dieser Ansatz ist physikalisch durchaus sinnvoll: Wir können steuern, wieviel in ein Volumen hinein oder aus ihm heraus fließt, aber nicht, wie sich das Wasser parallel zum Rand bewegt.

Variationsformulierung im Hilbert-Raum. Mit Hilfe dieser Hilbert-Räume können wir die Variationsaufgabe in ihre endgültige Form bringen: Wir definieren $V := H_0(\operatorname{div}, \Omega)$ und $Q := L^2(\Omega)$ sowie die Bilinearformen

$$a : V \times V \to \mathbb{R}, \qquad (v, u) \mapsto \int_\Omega \frac{\langle v(x), u(x)\rangle_2}{k(x)}\, dx,$$
$$b : V \times Q \to \mathbb{R}, \qquad (v, p) \mapsto \int_\Omega \nabla \cdot v(x) p(x)\, dx$$

und suchen nach einem Paar $(u, p) \in V \times Q$ mit

$$\begin{aligned} a(v, u) + b(v, p) &= -a(v, u_D) && \text{für alle } v \in V, \\ b(u, q) &= -b(u_D, q) && \text{für alle } q \in Q. \end{aligned}$$

Indem wir „beide Zeilen addieren“ können wir auch zu einer Variationsaufgabe in der gewohnten Form kommen:

$$a(v, u) + b(v, p) + b(u, q) = -a(v, u_D) - b(u_D, q) \qquad \text{für alle } (v, q) \in V \times Q.$$

Die linke Seite ist eine Bilinearform auf dem Raum $V \times Q$, die rechte Seite ein Funktional auf diesem Raum. Wenn wir in dieser Gleichung $q = 0$ einsetzen, erhalten wir die

erste Zeile des ursprünglichen Problems, mit $v = 0$ dagegen die zweite, so dass beide Formulierungen tatsächlich gleichwertig sind.

Unter Verwendung des Satzes 6.17 von Lax-Milgram lässt sich beweisen, dass diese Variationsaufgabe eine Lösung $(u, p) \in V \times Q$ besitzt. Diese Lösung ist nicht eindeutig, allerdings lassen sich zwei Lösungen ineinander überführen, indem man zu p eine Konstante addiert. Falls es uns wichtig ist, eine eindeutig lösbare Aufgabe zu behandeln, könnten wir beispielsweise den Raum Q auf diejenigen Funktionen einschränken, deren Integral verschwindet, also auf

$$Q_\perp := \{q \in Q \ : \ \langle 1, q\rangle_{L^2} = 0\}.$$

In dem Teilraum $V \times Q_\perp$ besitzt die Variationsaufgabe dann nur noch genau eine Lösung.

Ansatzraum. Wir wollen die Variationsaufgabe wieder mit einem Galerkin-Ansatz behandeln, also V und Q durch endlich-dimensionale Räume $V_h \subseteq V$ und $Q_h \subseteq Q$ ersetzen. Es bietet sich an, wieder einen Raum aus stückweisen Polynomen zu verwenden. Damit der Raum in $H(\mathrm{div}, \Omega)$ enthalten ist, müssen wir nachprüfen, unter welchen Bedingungen stückweise polynomiale Funktionen eine schwache Divergenz besitzen.

Sei also $\mathcal{T}$ eine Triangulation des Gebiets Ω, und sei $u \in \Pi^d_{\mathcal{T},m}$ ein stückweises Polynom m-ten Grades. Um die Bedingung der Definition 7.35 nachprüfen zu können wählen wir eine Testfunktion $v \in C^1_0(\Omega)$ und untersuchen das Integral

$$-\int_\Omega \langle \nabla v(x), u(x)\rangle_2 \, dx = \sum_{t\in\mathcal{T}} -\int_{\omega_t} \langle \nabla v(x), u(x)\rangle_2 \, dx.$$

Da $u|_{\omega_t}$ ein Polynom ist, können wir mit der Formel (6.6) partiell integrieren und erhalten

$$-\int_{\omega_t} \langle \nabla v(x), u(x)\rangle_2 \, dx = \int_{\omega_t} v(x) \nabla \cdot u(x) \, dx - \int_{\partial\omega_t} v(x) \langle n(x), u(x)\rangle_2 \, dx$$

für alle $t \in \mathcal{T}$. Der erste Term auf der rechten Seite ist uns willkommen, denn er erlaubt es uns, eine Funktion $w \in L^2(\Omega)$ durch

$$w|_{\omega_t} := \nabla \cdot u|_{\omega_t} \qquad \text{für alle } t \in \mathcal{T}$$

zu definieren und so die gewünschte schwache Divergenz zu erhalten. Dem zweiten Term entnehmen wir die Bedingung, die der Raum erfüllen muss, um in $H(\mathrm{div}, \Omega)$ enthalten zu sein: Wenn wir, wie schon im Beweis des Satzes 6.27, die Seiten eines Simplex $t \in \mathcal{T}$ mit

$$\mathcal{F}_t := \{s \in \mathcal{S}_{d-1} \ : \ s \subseteq t\}$$

bezeichnen, können wir das Randintegral in der Form

$$\int_{\partial\omega_t} v(x) \langle n(x), u(x)\rangle_2 \, dx = \sum_{s\in\mathcal{F}_t} \int_{\omega_s} v(x) \langle n(x), u(x)\rangle_2 \, dx$$

schreiben. Falls eine Seite $s \in \mathcal{F}_t$ zu dem äußeren Rand $\partial\Omega$ des Gebiets gehört, gilt $v|_{\omega_s} = 0$, so dass das entsprechende Teilintegral verschwindet.

Anderenfalls gibt es genau einen weiteren Simplex $t' \in \mathcal{T}$, der ebenfalls diese Seite besitzt, und das Integral über diese Seite tritt auch in unserer Summe auf. Wir brauchen lediglich dafür zu sorgen, dass sich die beiden Integrale über diese Seite gegenseitig auslöschen. Da der äußere Normalenvektor von ω_t auf ω_s gerade in die entgegengesetzte Richtung des äußeren Normalenvektors von $\omega_{t'}$ auf ω_s weist, genügt dafür die Stetigkeit der Normalenkomponente der Funktion u, es muss also

$$\lim_{\substack{y \to x \\ y \in \omega_t}} \langle n(x), u(y) \rangle_2 = \lim_{\substack{z \to x \\ z \in \omega_{t'}}} \langle n(x), u(z) \rangle_2 \qquad \text{für alle } x \in \omega_s$$

gelten. Im Vergleich mit Satz 6.27 sehen wir, dass der Raum $H(\operatorname{div}, \Omega)$ wesentlich schwächere Ansprüche als $H^1(\Omega, \mathbb{R}^3)$ stellt: Stückweise Polynome brauchen nicht stetig über die Elementgrenzen hinweg zu sein, es genügt, wenn ihre Normalenkomponenten stetig sind.

Raviart-Thomas-Elemente. Die schwächeren Voraussetzungen, die an stückweise Polynome in $H(\operatorname{div}, \Omega)$ gestellt werden, lassen sich ausnutzen, um besonders einfache Ansatzräume zu konstruieren. Ein Beispiel sind die *Raviart-Thomas-Elemente*, die nicht nur stückweise Polynome sind, sondern bei denen im einfachsten Fall sogar für jedes $t \in \mathcal{T}$ ein $a \in \mathbb{R}^d$ und ein $b \in \mathbb{R}$ so existieren, dass

$$u(x) = a + bx \qquad \text{für alle } x \in \omega_t$$

gilt. Während bei linearen Polynomen für jede Komponente der vektorwertigen Funktion $d+1$ Freiheitsgrade zur Verfügung stehen, also insgesamt $d(d+1)$, genügen bei Raviart-Thomas-Elementen insgesamt $d+1$ Freiheitsgrade.

Die Stetigkeit der Normalenkomponente lässt sich bei diesen Funktionen besonders einfach sicherstellen: Sei $s \in \mathcal{S}_{d-1}$ eine Seite eines Elements $t \in \mathcal{T}$ der Triangulation, und sei n ein Normalenvektor auf dieser Seite. Wenn wir zwei Punkte $x, y \in \omega_s$ betrachten, muss ihre Differenz senkrecht auf dem Normalenvektor stehen, wir haben also

$$\langle n, x - y \rangle_2 = 0.$$

Daraus folgt bereits

$$\langle n, u(y) \rangle_2 = \langle n, a + by \rangle_2 = \langle n, a + by \rangle_2 + b\langle n, x - y \rangle_2 = \langle n, a + bx \rangle_2 = \langle n, u(x) \rangle_2$$

die Normalenkomponente der Funktion u ist demnach auf der gesamten Seitenfläche konstant. Um die Stetigkeit der Normalenkomponente bei Raviart-Thomas-Funktionen sicherzustellen genügt es deshalb, dafür zu sorgen, dass in einem einzigen Punkt jeder Seite, beispielsweise im Mittelpunkt, die Normalenkomponenten der beiden angrenzenden Elemente übereinstimmen.

Wir können Basisfunktionen konstruieren, deren Normalenkomponenten jeweils auf einer Seite eines Elements ungleich null sind und auf allen anderen verschwinden. Als Beispiel untersuchen wir den dreidimensionalen Fall: Sei $t = \{i, j, k, \ell\} \in \mathcal{S}_3$. Wir suchen eine Basisfunktion

$$\varphi(x) = a + bx \qquad \text{für alle } x \in \omega_t,$$

deren Normalenkomponente auf der Seite $s = \{j, k, \ell\}$ ungleich null ist und auf den Seiten $\{i, j, k\}$, $\{i, k, \ell\}$ und $\{i, j, \ell\}$ verschwindet. Da die Normalenkomponenten auf jeder Seite konstant sind, genügt es, sie in einem einzigen Punkt verschwinden zu lassen. Wir entscheiden uns für den Punkt i, der zu allen drei Seiten gehört. Mit dem Kreuzprodukt (vgl. (4.1)) können wir Normalenvektoren der drei Seiten konstruieren und erhalten die Gleichungen

$$\begin{aligned}\langle a + bi, (j - i) \times (k - i)\rangle_2 &= 0,\\ \langle a + bi, (k - i) \times (\ell - i)\rangle_2 &= 0,\\ \langle a + bi, (j - i) \times (\ell - i)\rangle_2 &= 0.\end{aligned}$$

Alle drei Gleichungen lassen sich sehr einfach erfüllen, indem wir $a := -bi$ setzen. Wir müssen allerdings noch überprüfen, ob es bei dieser Wahl möglich ist, auf der Seite $s = \{j, k, \ell\}$ eine von null verschiedene Normalenkomponente zu erreichen. Es genügt wieder, die Prüfung nur in einem einzigen Punkt durchzuführen, wir entscheiden uns für den Punkt j. Dann gilt

$$\begin{aligned}\langle a + bj, (k - j) \times (\ell - j)\rangle_2 &= \langle -bi + bj, (k - j) \times (\ell - j)\rangle_2\\ &= -b\langle (i - j), (k - j) \times (\ell - j)\rangle_2\\ &= -b \det(i - j, k - j, \ell - j).\end{aligned}$$

Wenn wir voraussetzen, dass der Tetraeder t regulär ist, sind die Vektoren $i - j$, $k - j$ und $\ell - j$ linear unabhängig und die Determinante damit ungleich null. Also können wir die Bedingung erfüllen, indem wir ein beliebiges $b \neq 0$ einsetzen. Für die Wahl $b = 1$ ergibt sich beispielsweise

$$\varphi(x) = x - i \qquad \text{für alle } x \in \omega_t.$$

Entsprechend können wir Basisfunktionen für die anderen Seiten konstruieren, und diese Basisfunktionen lassen sich zu globalen Basisfunktionen in $H(\mathrm{div}, \Omega)$ zusammensetzen.

Bemerkung 7.36 ($H(\mathrm{curl}, \Omega)$) *Analog zu der schwachen Divergenz lässt sich auch eine schwache Rotation $\nabla \times u$ definieren. Der Raum $H(\mathrm{curl}, \Omega)$ aller L^2-Funktionen mit einer schwachen Rotation kann dann entsprechend mit einer Norm und einem Skalarprodukt zu einem Hilbert-Raum gemacht werden.*

Dieser Raum eignet sich gut für die Behandlung der Maxwell-Gleichungen, falls wir eine Formulierung wählen, in der wir die Gauß-Gleichungen vernachlässigen dürfen.

Stückweise polynomiale Ansatzräume müssen eine stetige Tangentialkomponente aufweisen, um in $H(\mathrm{curl}, \Omega)$ enthalten zu sein. Diese Bedingung lässt sich beispielsweise mit Nédélec-Elementen *erfüllen, bei denen wir für jedes $t \in \mathcal{T}$ die Existenz zweier Vektoren $a, b \in \mathbb{R}^3$ mit*

$$u(x) = a + b \times x \qquad \textit{für alle } x \in \omega_t$$

fordern. Analog zu den Raviart-Thomas-Elementen ist die Tangentialkomponente dieser Funktionen auf jeder Seitenfläche konstant, so dass sich die Stetigkeitsbedingung einfach umsetzen lässt.

8 Parallelisierung

Viele der von uns behandelten Verfahren führen zu einem sehr hohen Rechenaufwand, beispielsweise wenn sehr viele Partikel aufeinander Kräfte ausüben oder wenn eine Triangulation mit sehr vielen Punkten zum Einsatz kommen muss, um eine komplizierte Geometrie darzustellen oder eine hohe Genauigkeit zu erreichen.

In diesen Fällen empfiehlt es sich, nach Möglichkeiten zu suchen, mit denen sich die Rechenzeit reduzieren lässt. Ein sehr erfolgreich Ansatz besteht darin, die anfallenden Rechenoperationen auf mehr als nur ein Rechenwerk, mehr als nur einen Prozessor oder sogar mehr als nur einen Rechner zu verteilen, so dass mehrere Aufgaben gleichzeitig ausgeführt werden können.

8.1 Vektorisierung

Um die Rechenleistung bei gleichbleibender Taktfrequenz zu steigern, muss ein Computer in die Lage versetzt werden, mehrere Aufgaben gleichzeitig auszuführen. Es müssen also mindestens mehrere Rechenwerke (abgekürzt ALU für *arithmetic logic unit*) vorhanden sein.

Vektorrechner beruhen auf der Idee, möglichst nur die Anzahl der Rechenwerke zu erhöhen, aber nicht die sonstigen Bestandteile eines Prozessors wie den Befehlszähler oder die Datenregister. Das hat zur Folge, dass ein Vektorrechner im Prinzip immer nur einen Befehl ausführt, allerdings für mehrere Datensätze gleichzeitig. Man spricht von einem SIMD-Modell, Abkürzung für *single instruction, multiple data.*

Beispiel: Addition. Als Beispiel können wir eine Vektoraddition untersuchen. Wenn wir einen Vektoren $\mathbf{y} \in \mathbb{R}^n$ zu einem Vektor $\mathbf{x} \in \mathbb{R}^n$ addieren wollen, führt ein konventioneller Prozessor n Additionen aus, nämlich

$$x_1 \leftarrow x_1 + y_1, \dots \qquad\qquad x_n \leftarrow x_n + y_n.$$

Ein Vektorrechner mit m-fachem Rechenwerk würde für ein durch m teilbares n dagegen nur n/m Vektoradditionen ausführen, nämlich

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \leftarrow \begin{pmatrix} x_1 + y_1 \\ \vdots \\ x_m + y_m \end{pmatrix}, \qquad \begin{pmatrix} x_{n-m+1} \\ \vdots \\ x_n \end{pmatrix} \leftarrow \begin{pmatrix} x_{n-m+1} + y_{n-m+1} \\ \vdots \\ x_n + y_n \end{pmatrix}.$$

Die Rechenzeit sinkt dabei um den Faktor m, während der Schaltungsaufwand nur geringfügig wächst.

Als Beispiel können wir die von Intel eingeführte *Streaming SIMD Extension* (SSE) verwenden (vgl. Intel Intrinsics Guide), bei der Variablen des Typs `__m128` zum Einsatz kommen, die jeweils 128 Bit aufnehmen, die als vier 32-Bit-Gleitkommazahlen interpretiert werden. Für die Sprache C werden in der Datei `xmmintrin.h` die nötigen Typen und Funktionen definiert, unter anderem die Funktion `_mm_load_ps`, mit der der Inhalt eines vierelementigen Arrays in eine Vektorvariable kopiert wird, die Funktion `_mm_store_ps`, die den entgegengesetzten Kopiervorgang durchführt, und die Funktion `_mm_add_ps`, die zwei Vektorvariablen komponentenweise addiert. Unsere Vektoraddition nimmt dann die folgende Form an:

```
void
vector_add(int n, float *x, const float *y)
{
  __m128 vx, vy;
  int i;

  for(i=0; i+3<n; i+=4) {
    vx = _mm_load_ps(x+i);
    vy = _mm_load_ps(y+i);
    vx = _mm_add_ps(vx, vy);
    _mm_store_ps(x+i, vx);
  }

  for(; i<n; i++)
    x[i] += y[i];
}
```

In der ersten Schleife werden jeweils vier Gleitkommazahlen addiert, bis nur noch ein Rest von höchstens drei Zahlen übrig bleibt, der mit der zweiten Schleife versorgt wird.

Beispiel: Exponentialfunktion. Obwohl ihr Name es nahelegt, sind Vektorrechner keineswegs auf Operationen der linearen Algebra beschränkt. Beispielsweise können wir auch die Exponentialfunktion für m Werte gleichzeitig berechnen: Dazu nähern wir sie durch eine abgebrochene Exponentialreihe

$$\exp(x) \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}$$

an, die wir effizienter als

$$\exp(x) \approx 1 + x\left(1 + \frac{x}{2}\left(1 + \frac{x}{3}\left(1 + \frac{x}{4}\right)\right)\right)$$

schreiben können. Die Auswertung der Klammern erfolgt dann in der Reihenfolge

$$y \leftarrow 1 + x/4,$$
$$y \leftarrow 1 + xy/3$$

$$y \leftarrow 1 + xy/2$$
$$y \leftarrow 1 + xy,$$

die sich auf einem Vektorrechner als

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \leftarrow \begin{pmatrix} 1 + x_1/4 \\ \vdots \\ 1 + x_m/4 \end{pmatrix}, \qquad \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \leftarrow \begin{pmatrix} 1 + x_1 y_1/3 \\ \vdots \\ 1 + x_m y_m/3 \end{pmatrix},$$
$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \leftarrow \begin{pmatrix} 1 + x_1 y_1/2 \\ \vdots \\ 1 + x_m y_m/2 \end{pmatrix}, \qquad \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \leftarrow \begin{pmatrix} 1 + x_1 y_1 \\ \vdots \\ 1 + x_m y_m \end{pmatrix}$$

für m Werte parallel ausführen lässt.

Fallunterscheidungen. Ein großer Nachteil eines konventionellen Vektorrechners ist die diesem Ansatz innewohnende Schwierigkeit bei der Behandlung von Fallunterscheidungen: Wenn wir beispielsweise die Nullstellen des quadratischen Polynoms

$$x^2 - 2px + q = 0$$

berechnen wollen, können wir das mit den wohlbekannten Formeln

$$x_1 = p + \sqrt{p^2 - q}, \qquad x_2 = p - \sqrt{p^2 - q}$$

erledigen. Mathematisch wären wir damit fertig, aber bei der Umsetzung auf einem Computer, der mit *Gleitkommazahlen* arbeitet, können wir in große Schwierigkeiten geraten, falls $|q|$ klein im Verhältnis zu $|p|^2$ ist, so dass für die Wurzel $\sqrt{p^2 - q} \approx |p|$ gilt.

Falls p negativ ist, wird dann bei der Berechnung von x_1 die Differenz zwischen dem negativen p und der positiven Wurzel $\sqrt{p^2 - q} \approx |p|$ berechnet, so dass sich viele Ziffern gegenseitig auslöschen und nur eine geringe Genauigkeit erreicht wird. x_2 dagegen lässt sich in diesem Fall problemlos berechnen, da zwei negative Zahlen addiert werden und keine Auslöschung auftritt.

Falls p dagegen positiv ist, lässt sich x_1 genau berechnen, während nun x_2 ungenau ermittelt wird.

Immerhin können wir in beiden Fällen wenigstens eine Nullstelle genau bestimmen, und da

$$x^2 - 2px + q = (x - x_1)(x - x_2) = x^2 - (x_1 + x_2)x + x_1 x_2$$

gilt, können wir mit Hilfe der Gleichung $q = x_1 x_2$ mit einer einzigen Division die zweite Nullstelle rekonstruieren.

Der Einfachheit halber ändern wir die Numerierung so, dass $|x_1| \geq |x_2|$ gilt und die stabil zu berechnende betragsgrößte Nullstelle jeweils als erstes bestimmt wird. Dann ergibt sich der folgende Algorithmus:

$y \leftarrow \sqrt{p^2 - q}$;
if $p \geq 0$ **then** $x_1 \leftarrow p + y$;
else $x_1 \leftarrow p - y$;
$x_2 \leftarrow q / x_1$

Wenn wir diesen Algorithmus auf einem Vektorrechner umsetzen wollen, stehen wir vor dem Problem, dass *abhängig vom Wert der Variablen* p unterschiedliche Befehle ausgeführt werden müssen. Wenn wir einen Vektor $\mathbf{p} \in \mathbb{R}^m$ verarbeiten wollen, müssten also unterschiedliche Befehle auf unterschiedliche Komponenten angewendet werden, und das ist bei einem Vektorrechner gerade nicht vorgesehen.

Eine einfache Lösung dieses Problems besteht darin, bitweise logische Verknüpfungen einzusetzen: In Anlehnung an die in der Programmiersprache C übliche Notation bezeichnen wir mit & die bitweise Und-Verknüpfung, mit | die bitweise Oder-Verknüpfung und mit $\sim$ das bitweise Komplement. Wenn wir nun einer Variablen z entweder den Wert x oder den Wert y zuweisen wollen, können wir das ohne Fallunterscheidung mit der Formel

$$z \leftarrow (x \;\&\; m) \mid (y \;\&\; \sim m)$$

bewerkstelligen: Falls $m = 0$ gilt, erhalten wir $z = y$, falls $m = \sim 0$ gilt, erhalten wir stattdessen $z = x$.

Um diesen Weg in unserem Beispiel beschreiten zu können, benötigen wir also eine Funktion, die den Wert ~ 0 zurückgibt, falls $p \geq 0$ gilt, und anderenfalls 0. Die Befehlssätze vieler Vektorrechner enthalten eine entsprechende Funktion, die wir hier als

$$\mathrm{geq}(x, y) := \begin{cases} \sim 0 & \text{falls } x \geq y, \\ 0 & \text{ansonsten} \end{cases}$$

schreiben. Mit ihrer Hilfe können wir unseren Algorithmus nun ohne Fallunterscheidungen formulieren, beispielsweise mit SSE-Befehlen:

```
y = _mm_sqrt_ps(_mm_sub_ps(_mm_mul_ps(p, p), q));
m = _mm_cmpge_ps(p, zero);
x1 = _mm_or_ps(_mm_and_ps(m, _mm_add_ps(p, y)),
               _mm_andnot_ps(m, _mm_sub_ps(p, y)));
x2 = _mm_div_ps(q, x1);
```

In der zweiten Zeile wird die Bitmaske m konstruiert, in der dritten und vierten werden die beiden möglichen Werte `_mm_sub_ps(p, y)` und `_mm_add_ps(p, y)` für x_1 ermittelt, mit m oder $\sim m$ bitweise und-verknüpft und dann mit `_mm_or_ps` bitweise oder-verknüpft, um das erste Ergebnis zu erhalten.

Dieser Lösungsansatz weist den für Vektorrechner typischen Nachteil auf, dass immer *beide* Zweige der Fallunterscheidung durchlaufen werden müssen, da manche Komponenten eine Folge von Rechenschritten erfordern könnten und manche die andere. Deshalb ist es sehr empfehlenswert, bei der Programmierung von Vektorrechnern sehr sorgfältig darauf zu achten, möglichst wenige Fallunterscheidungen zu verwenden und die unterschiedlichen Fälle möglichst schnell wieder zusammenzuführen.

SIMT. Da bei der Berechnung aufwendiger dreidimensionaler Grafiken sehr viele sehr ähnliche Rechenoperationen ausgeführt werden müssen, sind Grafikprozessoren in der Regel als Vektorrechner aufgebaut. Um ihre Programmierung zu erleichtern kommt häufig eine Variante des SIMD-Modells zum Einsatz, die als SIMT bezeichnet wird, Abkürzung für *single instruction, multiple threads*.

In diesem Modell werden die Rechenwerke durch *processing elements*, kurz PEs, ersetzt, die über einen kleinen Satz von Variablen verfügen, zu denen auch ein Befehlszähler gehört. Alle PEs führen dasselbe Programm aus.

In jedem Taktzyklus wählt eine übergeordnete Instanz, der *Scheduler*, einen Befehl des Programms aus, der von allen PEs ausgeführt wird, deren Befehlszähler auf ihn zeigt. Alle anderen PEs tun in diesem Taktzyklus nichts.

Dieser Zugang bietet den Vorteil, dass wir einen Vektorrechner wie einen konventionellen Rechner programmieren können und dass Fallunterscheidungen über die Befehlszähler automatisch abgearbeitet werden und der Programmfluss anschließend wieder zusammengeführt wird. Den prinzipiellen Nachteil, dass immer nur ein Befehl zur Zeit ausgeführt werden kann, behebt dieser Ansatz allerdings nicht.

8.2 Symmetrische Multiprozessorsysteme

Symmetrische Multiprozessorsysteme vermeiden den Nachteil des Vektorrechners, indem nicht nur mehrere Rechenwerke gleichzeitig arbeiten, sondern mehrere vollwertige Prozessoren. Dadurch können alle Prozessoren jederzeit beliebige Befehle ausführen, so dass sich sehr viel allgemeinere Algorithmen problemlos implementieren lassen. Der Nachteil eines Multiprozessorsystems besteht darin, dass ein Prozessor wesentlich aufwendiger als ein Rechenwerk ist, so dass deutlich weniger Prozessoren als Rechenwerke auf einen Chip passen.

Mehrkernprozessoren. Multiprozessorsysteme treten häufig in Form von *Mehrkernprozessoren* auf, also von Chips, die sich der Außenwelt gegenüber wie ein Prozessor verhalten, beispielsweise mit nur einer Speicherschnittstelle, die aber intern aus mehreren Prozessorkernen zusammengesetzt sind, die sich den Zugriff auf die externen Schnittstellen teilen.

Für die Programmierung spielt der Unterschied zwischen mehreren „echten“ Prozessoren und einem Prozessor mit mehreren Kernen keine große Rolle, für die Rechenleistung kann er allerdings sehr wichtig sein: Da sich bei einem Mehrkernprozessor alle Kerne dieselbe Speicherschnittstelle teilen, ist der maximale Durchsatz begrenzt. Das kann bei speicherintensiven Anwendungen dazu führen, dass die meisten Kerne die meiste Zeit damit verbringen, auf Daten aus dem Speicher zu warten.

Geteilter Speicher. In einem symmetrischen Multiprozessorsystem teilen sich alle Prozessoren denselben Hauptspeicher, oder es wird zumindest durch geeignete Schaltungen und Kommunikationsprotokolle für die auf den Prozessoren laufende Programme der Anschein erweckt, als würden sie auf denselben Speicher zugreifen.

Dadurch ist der Austausch von Informationen zwischen den Programmen relativ einfach, andererseits entsteht aber auch die Notwendigkeit, Zugriffe auf den Speicher zu synchronisieren. Wenn beispielsweise ein auf Prozessor A laufendes Programm auf Daten zugreifen muss, die ein auf Prozessor B laufendes Programm berechnet, muss sichergestellt sein, dass die Zugriffe des Programms A erst erfolgen, wenn Programm B seine Ergebnisse an der verabredeten Stelle im Speicher deponiert hat.

Insbesondere bei komplexen Algorithmen kann die Synchronisation der nebeneinander ablaufenden Programmen eine sehr anspruchsvolle Aufgabe sein. Die Fehlersuche in derartigen Programmen gestaltet sich schwierig, weil die Reihenfolge, in der die Programme ausgeführt werden, auf modernen Betriebssystemen kaum vorhersagbar ist und so Fehler vorkommen, die nur selten und nicht reproduzierbar auftreten.

Multithreading. Die Programmierung eines SMP-Systems erfolgt häufig auf Grundlage des Multithreading-Konzepts: Innerhalb eines Prozesses können mehrere *Threads* existieren. Ein Thread entspricht einem „Handlungsstrang“ innerhalb des Prozesses, also einer Folge von Befehlen, die der Reihe nach ausgeführt werden. Dementsprechend verfügt er über einen eigenen Befehlszähler, der angibt, welcher Befehl im nächsten Schritt ausgeführt werden soll, über Register, die die Daten für diese Befehle aufnehmen können, und über einen Kellerspeicher, mit dem sich lokale Variablen und Rücksprungsadressen von Funktionsaufrufen verwalten lassen. Das Betriebssystem übernimmt dann die Aufgabe, den einzelnen Threads Prozessoren oder Prozessorkerne zuzuordnen, die sie ausführen.

Da die Threads innerhalb desselben Prozesses laufen, teilen sie sich unter anderem dieselbe Sicht auf den Hauptspeicher und Dateien, so dass ein Programmierer die Vorzüge des geteilten Speichers uneingeschränkt nutzen kann.

OpenMP. Ein erfolgreicher Standard für die Thread-Programmierung ist OpenMP. Anders als frühere Standards wie POSIX Threads ist OpenMP eng mit dem Compiler verbunden, der ein C- oder Fortran-Programm in Maschinensprache übersetzt. Beispielsweise „versteht“ OpenMP Schleifen und Variablen, so dass in einfachen Fällen wenige Befehle genügen, um eine Berechnung auf mehrere Threads zu verteilen.

OpenMP fügt dem konventionellen Programmiermodell das Konzept des *parallelen Abschnitts* hinzu, in den ein Programm eintreten kann, wenn es Arbeit auf mehrere Threads verteilen möchte. Der parallele Abschnitt wird von mehreren Threads ausgeführt, die untereinander Daten austauschen und sich bei Bedarf synchronisieren können. Die Threads entstehen bei Eintritt in den parallelen Abschnitt, und der Abschnitt wird erst verlassen, wenn alle Threads sein Ende erreicht haben. Im OpenMP-Jargon nennt man die Threads, die gemeinsam bei Eintritt in denselben parallelen Abschnitt entstanden sind, ein *Team*. Jeder Thread hat innerhalb des Teams eine Nummer, durch die er eindeutig identifiziert werden kann und mit deren Hilfe wir ihm beispielsweise Aufgaben zuordnen können.

In der C-Ausprägung des OpenMP-Standards wird ein paralleler Abschnitt durch einen strukturierten Block dargestellt, also durch eine Folge von Befehlen, die in ge-

schweifte Klammern `{`, `}` eingeschlossen ist, dem die Zeile `#pragma omp parallel` vorangestellt ist. `#pragma`-Anweisungen sind eine Möglichkeit, C-Programme um zusätzliche Funktionen zu ergänzen. Ein Compiler kann auf eine `#pragma`-Zeile reagieren, er kann sie aber auch ignorieren. Durch diese Konvention ist sichergestellt, dass auch ein Compiler, der OpenMP nicht kennt, zumindest einfache OpenMP-Programme trotzdem korrekt übersetzen kann. Ein besonders einfaches Beispiel ist das folgende Programm:

```
#include <stdio.h>
#include <omp.h>
int
main()
{
  #pragma omp parallel
  {
    int me = omp_get_thread_num();
    int teamsize = omp_get_num_threads();
    printf("Thread %d of %d reporting\n", me, teamsize);
  }
  return 0;
}
```

Wichtig ist natürlich der parallele Abschnitt, in dem die Variablen `me` und `teamsize` gesetzt und ausgegeben werden. Die beiden Funktionen `omp_get_thread_num` und `omp_get_num_threads` sind in der zu dem OpenMP-Standard gehörenden Datei `omp.h` definiert. Die Funktion `omp_get_thread_num` ermittelt die Nummer des Threads innerhalb des aktuellen Teams, die Funktion `omp_get_num_threads` ermittelt die Anzahl der Threads in diesem Team.

Arbeitsteilung. Diese beiden Zahlen genügen, um Arbeit auf mehrere Threads zu verteilen. Wenn wir beispielsweise eine Funktion für alle Elemente eines Arrays auswerten möchten, können wir das folgende Programmfragment verwenden:

```
#pragma omp parallel
{
  int me = omp_get_thread_num();
  int teamsize = omp_get_num_threads();
  int i;
  for(i=n*me/teamsize; i<n*(me+1)/teamsize; i++)
    y[i] = exp(0.3 * x[i] - 0.5) * sin(M_PI * 0.5 * x[i]);
}
```

Die Schleife ist so formuliert, dass jeder Thread einen zusammenhängenden Teil des Arrays bearbeitet und alle Threads für ungefähr gleich viele Einträge verantwortlich sind. Da diese Form der Arbeitsteilung relativ häufig auftritt, sieht OpenMP eine `#pragma`-Zeile vor, um `for`-Schleifen zu verteilen:

```
#pragma omp parallel
{
  int i;
#pragma omp for
  for(i=0; i<n; i++)
    y[i] = exp(0.3 * x[i] - 0.5) * sin(M_PI * 0.5 * x[i]);
}
```

In diesem Fall übernimmt OpenMP die Verteilung der in der Schleife anfallenden Arbeit auf die Threads des aktuellen Teams, so dass uns die Indexberechnung erspart bleibt. Ein weiterer Vorteil dieser Variante besteht darin, dass ein Compiler ohne OpenMP-Unterstützung die `#pragma`-Zeilen einfach ignoriert und trotzdem ein korrektes Programm entsteht.

Nichtdeterministisches Verhalten. Da das Betriebssystem darüber entscheidet, wann welcher Thread auf welchem Prozessor zur Ausführung gelangt, können wir in der Regel nicht vorhersagen, wann welche Befehle ausgeführt werden. Ein korrektes OpenMP-Programm muss deshalb so geschrieben werden, dass es unabhängig von der Reihenfolge der Threads korrekt arbeitet.

Hinzu kommen Seiteneffekte, die durch den gemeinsamen Zugriff auf den Speicher entstehen. Als Beispiel eignet sich das folgende Programmfragment:

```
int i = 0;
#pragma omp parallel
{
  i++;
}
printf("%d threads\n", i);
```

Auf den ersten Blick könnte man erwarten, dass in der letzten Zeile die Anzahl der Threads im Team ausgegeben wird. Tatsächlich sind alle Ergebnisse zwischen eins und der Teamgröße möglich, weil der Befehl `i++` bei vielen Prozessoren in mehreren Schritten ausgeführt wird: Zunächst wird der aktuelle Wert der Variablen aus dem Speicher in ein Prozessorregister gelesen, dann wird um eins hochgezählt, und anschließend wird das Ergebnis in den Speicher zurückgeschrieben. Falls beispielsweise alle Prozessoren den Wert aus dem geteilten Speicher lesen, bevor einer der anderen den aktualisierten Wert zurückgeschrieben hat, würden *alle* eine Null erhalten, hochzählen, und eine Eins in den Speicher schreiben.

Absicherung von geteilten Ressourcen. Derartige Probleme treten vor allem bei dem Zugriff auf den gemeinsamen Speicher auf, aber auch bei Verwendung von Dateien oder bestimmter Peripheriegeräte. OpenMP sieht einen Mechanismus vor, um sicher zu stellen, dass immer nur ein Thread zur Zeit Zugriff auf ein Objekt hat: Variablen des Typs `omp_lock_t` simulieren ein „Schloss“, das jeweils geöffnet oder geschlossen sein kann.

Die Funktion `omp_set_lock` wartet, bis ein Schloss offen ist, und schließt es daraufhin. Die Funktion `omp_unset_lock` öffnet das Schloss wieder. Mit diesen beiden Funktionen lassen sich Zugriffe auf gemeinsam genutzte Programmobjekte absichern: Für jedes Objekt wird ein Schloss angelegt, und bevor ein Thread es verändern kann, muss er das Schloss „hinter sich abschließen“, um sich den alleinigen Zugriff zu sichern. Sobald er fertig ist, öffnet er das Schloss wieder, so dass andere Threads es belegen können.

In dem folgenden Beispiel verwenden wir diesen Mechanismus, um Konflikte zwischen mehreren Threads zu vermeiden, die etwas in dieselbe Datei schreiben sollen:

```
omp_lock_t file_access;
omp_init_lock(&file_access);
#pragma omp parallel
{
  omp_set_lock(&file_access);
  fprintf(out, "Hello world\n");
  omp_unset_lock(&file_access);
}
omp_destroy_lock(&file_access);
```

Die Funktionen `omp_init_lock` und `omp_destroy_lock` dienen dabei der Initialisierung und der Freigabe des Schlosses.

8.3 Verteiltes Rechnen

Bei einem Vektorrechner liegen lediglich die Rechenwerke mehrfach vor, bei einem SMP-Rechner die Prozessoren. Einen Schritt weiter geht man bei einem *verteilten Rechner*, bei dem mehrere vollständige Rechner zum Einsatz kommen, die über ein geeignetes Netzwerk miteinander verbunden sind.

Dieser Zugang bietet einige Vorteile: Es lassen sich sehr große Rechnersysteme mit Hunderten oder Tausenden von Rechnern zusammenstellen, defekte Rechner lassen sich mit geringem Aufwand austauschen, jeder Rechner hat ungeteilten Zugriff auf seinen Speicher und seine sonstige Hardware. Der Nachteil besteht darin, dass wir bei einem verteilten System den Austausch von Daten zwischen den einzelnen Rechnern selber explizit implementieren müssen.

MPI. Als Standard für die Programmierung verteilter Rechnersysteme hat sich MPI etabliert, das *Message Passing Interface*. Der Name ist Programm: MPI beschreibt die Interaktion zwischen den einzelnen Knoten des Rechnernetzes durch Funktionen, die Daten von einem Knoten zu einem anderen übertragen oder auch einen Datenaustausch in größeren Gruppen von Knoten bewerkstelligen.

Da ein MPI-Programm auf mehreren Rechnern laufen soll, wird es in der Regel nicht einfach aufgerufen, sondern mit einem separaten Befehl gestartet, der es auf allen beteiligten Rechnern aufruft und den dabei entstehenden Prozessen zusätzliche Informationen gibt, mit deren Hilfe sie mit den anderen beteiligten Prozessen Daten austauschen

können. Viele Implementierungen des MPI-Standard sehen dafür ein Programm namens `mpirun` vor.

Da MPI-Funktionen in der Regel in einer separaten Bibliothek enthalten sind, muss bei der Übersetzung eines MPI-Programms darauf geachtet werden, dass diese Bibliothek eingebunden wird. Gerade bei großen Systemen kann der Fall eintreten, dass der Rechner, auf dem das MPI-Programm übersetzt wird, eine andere Architektur als die Rechner hat, auf denen es ausgeführt werden soll. In diesem Fall kann nicht der übliche Compiler des Entwicklungssystems zum Einsatz kommen, stattdessen muss ein Cross-Compiler verwendet werden, der ein Programm erzeugt, das auf dem verteilten System ausgeführt werden kann. Um diese Fragen kümmert sich bei vielen MPI-Implementierungen ein Programm namens `mpicc`, das im Wesentlichen wie ein gewöhnlicher C-Compiler arbeitet, aber ein Programm erzeugt, das für die Ausführung auf dem verteilten System vorgesehen ist und deshalb nur mit `mpirun` (oder seinem Äquivalent) gestartet werden sollte.

Communicator. Damit der Austausch von Nachrichten zwischen den verschiedenen Knoten des Rechnernetzes geordnet vonstatten gehen kann, führt der MPI-Standard den Begriff des *Communicators* ein. Ein Communicator beschreibt einen Kontext, in dem Kommunikation stattfindet, beispielsweise enthält er Angaben darüber, welche Knoten an der Kommunikation teilnehmen und in welcher Reihenfolge einzelne Nachrichten gesendet wurden, damit sichergestellt ist, dass sie auch in dieser Reihenfolge wieder empfangen werden.

Ein Communicator wird durch den Typ `MPI_Comm` repräsentiert. Wenn die Ausführung eines MPI-Programms beginnt, ist der Communicator `MPI_COMM_WORLD` definiert, zu dem alle Knoten gehören, auf denen das Programm gestartet wurde. Ein MPI-Programm kann mit Hilfe geeigneter Funktionen weitere Communicator-Objekte anlegen, um beispielsweise Teilmengen der Knoten zu definieren, die Daten austauschen sollen.

Für die Verteilung der Rechenarbeit auf die einzelnen Knoten ist das natürlich von Interesse, wieviele Knoten zusammenarbeiten und der wievielte davon den aktuellen Prozess ausführt. Mit den Funktionen `MPI_Comm_size` und `MPI_Comm_rank` können wir diese beiden Informationen in Erfahrung bringen. Ein einfaches MPI-Programm, in dem sich alle Prozesse melden, könnte wie folgt aussehen:

```
#include <mpi.h>
#include <stdio.h>
int
main(int argc, char **argv)
{
  int size, rank;
  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  printf("Node %d of %d reporting\n", rank, size);
  MPI_Finalize();
```

```
  return 0;
}
```

Die Funktion `MPI_Init` initialisiert die MPI-Bibliothek (beispielsweise indem Befehlszeilenparameter ausgewertet und Kommunikationskanäle geöffnet werden), während die Funktion `MPI_Finalize` für einen geordneten Abschluss des MPI-Programms sorgt.

Nachrichten. Der MPI-Standard unterscheidet zwischen *Punkt-zu-Punkt-Kommunikation*, bei der ein Knoten des Netzes einem anderen eine Nachricht schickt, und *kollektiver Kommunikation*, an der alle Knoten beteiligt sind.

Für die Punkt-zu-Punkt-Kommunikation genügen im Wesentlichen zwei Funktionen: `MPI_Send` schickt eine Nachricht von dem aktuellen Prozess zu einem anderen, `MPI_Recv` empfängt eine von einem anderen Prozess eintreffende Nachricht. Eine Nachricht besteht dabei aus mehreren Teilen: Sie enthält natürlich die Daten, die übertragen werden sollen. Sie enthält aber auch die Nummer des Empfängers und eine zusätzliche Markierung (engl. *message tag*), mit der unterschiedliche Sorten von Nachrichten unterschieden werden können. Außerdem gibt sie den Communicator an, in dessen Kontext die Nachricht übertragen werden soll.

Im folgenden Beispiel sendet der Knoten mit der Nummer null eine Zufallszahl an den Knoten mit der Nummer eins:

```
MPI_Status status;
int x;
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if(rank == 0) {
  x = rand();
  printf("Sending %d\n", x);
  MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
}
else if(rank == 1) {
  MPI_Recv(&x, 1, MPI_INT, 0, MPI_ANY_TAG, MPI_COMM_WORLD, &status);
  printf("Received %d\n", x);
}
```

Die ersten drei Parameter der Funktionen `MPI_Send` und `MPI_Recv` beschreiben die Daten, die übertragen werden sollen durch einen Zeiger auf ihren Ort, ihre Anzahl und ihren Typ. Der vierte Parameter gibt den Kommunikationspartner an, der fünfte die Markierung, der sechste den Communicator, in dessen Kontext die Operation ausgeführt werden soll. Die Funktion `MPI_Recv` verfügt über einen siebten Parameter, mit dem sich weitere Informationen über die empfangene Nachricht gewinnen lassen.

Nicht-blockierende Kommunikation. Falls Knoten Daten senden und empfangen sollen, kann es schwierig sein, die Sende- und Empfangsfunktionen geeignet abzustimmen: Die Empfangsfunktion blockiert die Ausführung des Prozesses, der sie aufruft, bis sie

Daten empfangen hat. Die Sendefunktion kann ebenfalls blockieren. Wenn wir nun beispielsweise ein Programm schreiben, in dem alle Prozesse zuerst senden und dann empfangen wollen, kann es sein, dass alle in der Sendefunktion blockiert werden, weil keiner der Prozesse jemals dazu kommt, die Empfangsfunktion aufzurufen.

Eine elegante Lösung dieses Problems sind *nicht-blockierende* Funktionen wie beispielsweise `MPI_Isend` und `MPI_Irecv`, die nicht abwarten, bis Daten vollständig verschickt oder empfangen wurden:

```
MPI_Request send_req;
int next, prev;
int x, y;
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
next = (rank+1) % size; prev = (rank+size-1) % size;
MPI_Isend(&x, 1, MPI_INT, next, 0, MPI_COMM_WORLD, &send_req);
MPI_Recv(&y, 1, MPI_INT, prev, MPI_ANY_TAG, MPI_COMM_WORLD, 0);
MPI_Wait(&send_req, 0);
```

Die Funktion `MPI_Isend` stellt eine Nachricht für den Knoten `next` bereit und verwendet die Variable `send_req`, um den Zustand dieser Nachricht zu überwachen. Unabhängig davon, ob die Nachricht empfangen wurde oder nicht, kehrt die Funktion dann in das aufrufende Programm zurück, sie blockiert also dessen Ausführung nicht weiter. Also kann nun `MPI_Recv` eine Nachricht von dem Knoten `prev` empfangen. Anschließend wartet die Funktion `MPI_Wait`, der wir die Variable `send_req` übergeben, darauf, dass die korrespondierende Nachricht empfangen wird. Diese Funktion kann wieder die Programmausführung blockieren, bis das geschehen ist. Bevor `MPI_Wait` nicht erfolgreich abgeschlossen wurde, dürfen wir die Variable `x` nicht verändern, denn wir wissen nicht, ob sie schon an den Empfänger übertragen wurde oder nicht.

Kollektive Kommunikation. Häufig müssen Daten an mehrere Knoten des Rechnernetzes übertragen werden. Wenn wir beispielsweise die Variable `x` des Knotens 0 an alle anderen Knoten schicken wollten, könnten wir das mit dem folgenden Programmfragment tun:

```
MPI_Comm_size(MPI_COMM_WORLD, &size);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if(rank == 0) {
  for(j=1; j<size; j++)
    MPI_Send(&x, 1, MPI_INT, j, 0, MPI_COMM_WORLD);
}
else
  MPI_Recv(&x, 1, MPI_INT, 0, MPI_ANY_TAG, MPI_COMM_WORLD, 0);
```

Diese Vorgehensweist ist zwar einfach, aber auch ineffizient: Nach dem ersten Sendevorgang verfügt nicht mehr nur Knoten 0 über den korrekten Wert von `x`, sondern auch

Knoten 1. Also könnte in einem zweiten Schritt nicht nur 0 Daten an 2 schicken, sondern gleichzeitig auch 1 Daten an 3. In dieser Weise würde sich in jedem Schritt die Anzahl der Knoten, die das korrekte `x` kennen, verdoppeln, so dass insgesamt nur $\lceil \log_2(\texttt{size}) \rceil$ Schritte erforderlich wären. Möglich wird diese deutlich Reduktion der Schrittzahl, weil alle Knoten sich an der Kommunikation beteiligen können.

In der Praxis hängt der Geschwindigkeitsgewinn natürlich von den Eigenschaften des verwendeten Netzwerks und eventuellen weiteren Umständen ab, die herauszufinden aufwendig wäre. Deshalb sieht der MPI-Standard spezielle Funktionen vor, die Daten an alle Knoten eines Communicators senden oder von ihnen empfangen, so dass die konkrete Implementierung die bestmögliche Vorgehensweise wählen kann, ohne dass die Programmierer davon etwas erfahren müssen. In unserem Beispiel ist die Funktion `MPI_Bcast` geeignet, die eine Nachricht an alle Knoten eines Communicators sendet:

```
int x;
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if(rank == 0) x = rand();
MPI_Bcast(&x, 1, &MPI_INT, 0, MPI_COMM_WORLD);
```

In diesem Fragment berechnet Knoten 0 eine Zufallszahl, die dann mit `MPI_Bcast` übertragen wird. Dabei ist wichtig, dass diese Funktion von *allen* Knoten aufgerufen wird, denn sie wird die Ausführung des Programms blockieren, bis das geschehen ist.

Wenn wir beispielsweise den Wert der Variablen `x` nur an einige der in dem Communicator `MPI_COMM_WORLD` enthaltenen Knoten schicken wollen, müssen wir einen neuen Communicator definieren, der nur diese Knoten enthält.

Index

Literaturverzeichnis

[1] Jürgen Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995.

[2] D. Braess. *Finite Elemente*. Springer, 3rd edition, 2003.

[3] L. Euler. *Institutionum calculi integralis*. Lipsiae Et Berolini, 1768.

[4] W. Hackbusch. *Iterative Solution of Large Sparse Systems*. Springer-Verlag New York, 1994.

[5] K. Heun. Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Zeitschrift für Mathematik und Physik*, 45:23–38, 1900.

[6] R. Hooke. *Lectures de potentia restitutiva, or, of Spring: explaining the power of spring bodies: to which are added some collections*. John Martyn, Printer to the Royal Society, 1678.

[7] I. Newton. *Philosophiæ Naturalis Principia Mathematica*. 1687.