Skript zur Vorlesung

Mathematische Grundlagen IV (CES) – Numerik

Benjamin Berkels

SS 2022

RWTH Aachen

letzte Änderung: 12. Juli 2022

Dieses Skript wurde mit Hilfe von KOMA-Script und LaTeX gesetzt.

Vorwort

Dieses Skript ist auf Basis verschiedener Varianten des Numerikteils der Vorlesung „Mathematische Grundlagen IV (CES)“ entstanden und basiert insbesondere auf Skripten von Prof. Frank, Prof. Sander und Prof. Torrilhon, sowie dem Buch „Numerik für Ingenieure und Naturwissenschaftler“ von W. Dahmen und A. Reusken. Es erhebt weder Anspruch auf Vollständigkeit noch auf Korrektheit. Sollten Ihnen Fehler auffallen, würden wir uns über eine Mail an `berkels@aices.rwth-aachen.de` freuen, damit folgende Generationen in den Genuss einer korrigierten Fassung kommen.

Im Frühling 2022, Benjamin Berkels

Inhaltsverzeichnis

1 Diskrete Fourier-Transformation

Die Fourier-Transformation ist ein wichtiges Werkzeug sowohl in der Analysis als auch in der Numerik. Grob gesprochen zerlegt die Fourier-Transformation ein Signal (z.B. einen Ton) in seine „Frequenzen“. Beispiele für Anwendungen sind:

- Lösung von partiellen Differentialgleichungen durch Reihenansatz bzw. durch die Fourier-Transformation
- Approximation von Funktionen durch Orthonormalsysteme (siehe Mathe 1 und 2)
- Schnelle Berechnung von Produkten großer Zahlen und Faltungen
- Schnelle Interpolation
- Datenkompression (z.B. nutzen MP3 und JPEG Varianten der Fourier-Transformation)

Der praktische Erfolg beruht auf einer speziellen Einsicht, die den Aufwand der Berechnung der Fourier-Transformierten von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n \log n)$ (wobei n die Anzahl der Datenpunkte ist) reduziert. Die sogenannte Fast Fourier Transform (FFT) gilt als einer der wichtigsten Algorithmen überhaupt. Zugeschrieben wird die Idee Cooley und Tukey (1965); allerdings gab es die Idee auch schon 1805 von Gauss, um die Bahnen von Asteroiden bestimmen zu können. Im Folgenden leiten wir zunächst die diskrete Fourier Transformation als Lösung einer Interpolationsaufgabe her und dann den Algorithmus für die FFT in seinen Grundzügen.

1.1 Trigonometrische Interpolation

Wir beginnen direkt in komplexer Notation und beschränken uns auf periodische Funktionen, d.h. Funktionen $f\colon \mathbb{R} \to \mathbb{C}$, so dass es ein $P > 0$ existiert mit $f(x) = f(x+P)$ für alle $x \in \mathbb{R}$. P nennt man Periode. Ohne Beschränkung der Allgemeinheit sei $P = 2\pi$. In diesem Kapitel seien daher (wenn nicht anders angegeben) alle Funktionen der Art $f\colon [0, 2\pi] \to \mathbb{C}$. Wir benutzen die Notation $\overline{f(x)}$ für die zu $f(x)$ komplex konjugierte Zahl.

Wir erinnern uns, dass

$$\langle f, g\rangle \coloneqq \frac{1}{2\pi} \int_0^{2\pi} f(x)\overline{g(x)}\,\mathrm{d}x$$

ein inneres Produkt zweier Funktionen f und g in $L^2([0, 2\pi]; \mathbb{C})$ definiert.

Definition 1.1. Für $j \in \mathbb{Z}$ sei

$$e_j : [0, 2\pi] \to \mathbb{C}, x \mapsto e^{ijx} ,$$

wobei $i \in \mathbb{C}$ die imaginäre Einheit bezeichnet. Die Funktionen e_j, $j \in \mathbb{Z}$, heißen *Grundschwingungen.*

Lemma 1.2. *Für $j, k \in \mathbb{Z}$ gilt*

$$\langle e_j, e_k\rangle = \frac{1}{2\pi} \int_0^{2\pi} e^{ijx} e^{-ikx}\,\mathrm{d}x = \delta_{j,k} .$$

Beweis. Mit Hilfe der Identität $e^{ix} = \cos x + i \sin x$ lässt sich die Aussage leicht nachrechnen (Übung). □

Bemerkung 1.3. Lemma 1.2 zeigt, dass die Grundschwingungen $(e_j)_{j\in\mathbb{Z}}$ ein Orthonormalsystem bzgl. $\langle\cdot,\cdot\rangle$ sind. Wir können nun auf den von einer Menge von Grundschwingungen aufgespannten Raum projizieren. Für $n \in \mathbb{N}$ seien

$$S_n(f;x) \coloneqq \sum_{|k|\le n} \langle f, e_k\rangle e^{ikx} \text{ und } U_{2n+1} \coloneqq \operatorname{span}\{e_k : \ |k| \le n\}.$$

Damit ist $S_n(f;\cdot) = \sum_{|k|\le n}\langle f, e_k\rangle e_k \in U_{2n+1}$ die Orthogonalprojektion von $f \in L^2([0,2\pi];\mathbb{C})$ auf den Unterraum U_{2n+1} und stellt somit eine Approximation von f. Es gilt

$$\langle f, e_k\rangle = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\,\mathrm{d}x \ .$$

In diesem Sinne geben die Koeffizienten an, wie viel von der k-ten „Frequenz“ e_k im Signal f enthalten ist. Die Koeffizienten $\langle f, e_k\rangle$ heißen *Fourier-Koeffizienten*. Die Projektion $S_n(f;\cdot)$ nennt man *Fourier-Teilsumme*.

Bemerkung 1.4. Man kann zeigen, dass das Orthonormalsystem der Grundschwingungen $(e_j)_{j\in\mathbb{Z}}$ auch vollständig ist, d.h. eine Art Basis von $L^2([0,2\pi];\mathbb{C})$ bildet. Genauer gilt:

$$S_n(f,\cdot) \underset{n\to\infty}{\to} f \quad \text{für alle} \quad f \in L^2([0,2\pi];\mathbb{C}) \ .$$

Man beachte, dass diese Konvergenz nicht punktweise gilt, sondern in L^2 (mehr dazu im Theorieteil). Die dabei entstehende Reihe

$$\sum_{k\in\mathbb{Z}} \langle f, e_k\rangle e^{ikx}$$

nennt man *Fourier-Reihe*.

Bemerkung 1.5. Um die Fourier-Teilsummen numerisch zu berechnen, könnte man prinzipiell die Integrale in den Fourier-Koeffizienten durch Quadraturformeln approximieren. Dies ist aber sehr aufwendig. Wir betrachten im Folgenden den Weg über Interpolation, der aber auch im Nachgang als Quadratur interpretiert werden kann. Wir betrachten äquidistante Stützstellen

$$x_j = \frac{2\pi j}{n},\ j = 0,\dots,n-1$$

im Intervall $[0,2\pi]$. Die komplexe Zahl

$$\varepsilon_n \coloneqq e^{-2\pi i/n}$$

heißt *n-te Einheitswurzel*, denn es gilt

$$(\varepsilon_n^j)^n = \varepsilon_n^{jn} = e^{-2\pi ij} = 1,\ j = 0,\dots,n-1,$$

d.h. die Zahlen ε_n^j, $j = 0,1,\dots,n-1$, teilen den Einheitskreis in gleich große Stücke.

Zur Interpolation verwenden wir die folgenden Funktionenräume.

Definition 1.6. Für $m \in \mathbb{N}$ sei

$$\mathcal{T}_m \coloneqq \operatorname{span}\{e_j : \ 0 \le j < m\}$$

Die Elemente von $\mathcal{T}_m$ heißen *(komplexe) trigonometrischen Polynome vom Grad (kleiner-gleich)* $m-1$.

Bemerkung 1.7. Da die Grundschwingungen $(e_j)_{j\in\mathbb{Z}}$ ein Orthonormalsystem bilden, sind sie insbesondere linear unabhängig und es folgt $\dim \mathcal{T}_m = m$. Ferner gilt

$$\mathcal{T}_m = \left\{ [0,2\pi] \mapsto \mathbb{C}, x \mapsto \sum_{j=0}^{m-1} c_j e^{ijx} : \ c_0, \ldots, c_{m-1} \in \mathbb{C} \right\}.$$

Setzt man $z = e^{ix}$, so hat jedes Element von $\mathcal{T}_m$ die Form

$$\sum_{j=0}^{m-1} c_j z^j,$$

d.h. die Elemente von $\mathcal{T}_m$ sind komplexe Polynome, die wegen $|e^{ix}| = 1$ nur auf dem Einheitskreis in der komplexen Ebene ausgewertet werden. Zusammen mit der Identität $e^{ix} = \cos x + i \sin x$ ist dies der Grund, dass man hier von komplexen trigonometrischen Polynomen spricht.

Satz 1.8.

(i) Seien $p_1 \in \mathcal{T}_m$ und $p_2 \in \mathcal{T}_n$. Dann gilt $p_1 p_2 \in \mathcal{T}_{m+n-1}$.

(ii) Sei $0 \neq p \in \mathcal{T}_m$. Dann hat p höchstens $m-1$ verschiedene Nullstellen in $[0, 2\pi)$.

Beweis.

(i) Wegen $p_1 \in \mathcal{T}_m$ und $p_2 \in \mathcal{T}_n$ gibt es Koeffizienten c_j und d_k mit

$$p_1(x) = \sum_{j=0}^{m-1} c_j e^{ijx} \text{ und } p_2(x) = \sum_{k=0}^{n-1} d_k e^{ikx}.$$

Daraus folgt

$$p_1(x)p_2(x) = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} c_j d_k e^{i(j+k)x} \in \mathcal{T}_{m+n-1}.$$

(ii) Angenommen $p = \sum_{j=0}^{m-1} c_j e^{ijx} \not\equiv 0$ hat m verschiedenen Nullstellen $x_1, \ldots, x_m \in [0, 2\pi)$. Wir betrachten das komplexe Polynom

$$q : \mathbb{C} \to \mathbb{C}, z \mapsto \sum_{j=0}^{m-1} c_j z^j.$$

Dann gilt

$$q(e^{ix_k}) = p(x_k) = 0 \text{ für alle } k \in \{1, ..., m\}.$$

Ferner sind wegen $x_k \in [0, 2\pi)$ die e^{ix_k} verschieden, damit hat das komplexe Polynom q m verschiedene Nullstellen, ist aber vom Grad $m-1$. Somit muss $q = 0$ gelten. Damit folgt aber auch dass die Koeffizienten c_j alle Null seien müssen und somit folgt auch $p = 0$. Ein Widerspruch zu unserer Annahme $p \not\equiv 0$. □

Wir betrachten nun die folgende Interpolationsaufgabe bzgl. der trigonometrischen Polynome.

Problem 1.9. Gegeben seien Werte $y_k \in \mathbb{C}$ $(k = 0, \ldots, n-1)$, sowie Punkte $x_k = \frac{2\pi k}{n} \in [0, 2\pi)$ $(k = 0, \ldots, n-1)$. Finde $T_n \in \mathcal{T}_n$ so dass

$$T_n(x_k) = y_k, \ k = 0, \ldots, n-1.$$

Bemerkung 1.10. Sei $f : [0, 2\pi] \to \mathbb{C}$ eine Funktion. Durch die Wahl $y_k = f(x_k)$ bedeutet die Interpolationsaufgabe in diesem Fall ein $T_n \in \mathcal{T}_n$ zu finden, das an den Stellen x_k mit f übereinstimmt. Die Punkte x_k nennt man auch Stützstellen.

Bemerkung 1.11. Da es sich letztlich nach Substitution um Polynome handelt, kann man die Lösbarkeit dieser Aufgabe (also Existenz und Eindeutigkeit des trigonometrischen Interpolationspolynoms) genauso beweisen wie im Fall des Standard-Interpolationspolynoms (also mit Hilfe des Lagrange-Polynoms, sowie des Fundamentalsatzes der Algebra; siehe Mathe 1).

Es gibt jedoch eine einfachere Möglichkeit die Existenz zu zeigen. Als Vorbereitung darauf zeigen wir zunächst eine Hilfsaussage.

Lemma 1.12. *Für $m \in \{-n+1, \ldots, n-1\}$, gilt*

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{-i2\pi mj/n} = \delta_{m,0}.$$

Beweis. Für $m = 0$ ergibt sich sofort

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{-i2\pi 0j/n} = \frac{1}{n} \sum_{j=0}^{n-1} 1 = 1.$$

Für den Fall $m \neq 0$ erinnern wir uns an die Summenformel der geometrischen Reihe: Es gilt

$$\sum_{j=0}^{n-1} q^j = \frac{1-q^n}{1-q} \text{ für } q \neq 1.$$

Da $|m| \leq n-1$, gilt $e^{-i2\pi m/n} \neq 1$ und es folgt, dass

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{-i2\pi mj/n} = \frac{1}{n} \sum_{j=0}^{n-1} (e^{-i2\pi m/n})^j = \frac{1}{n} \frac{1 - e^{-i2\pi m}}{1 - e^{-i2\pi m/n}} = \frac{1}{n} \frac{1-1}{1 - e^{-i2\pi m/n}} = 0. \qquad \square$$

Satz 1.13. *Seien $y = (y_k)_{k=0}^{n-1} \equiv (y_0, \ldots, y_{n-1})^T \in \mathbb{C}^n$ und $x_k = \frac{2\pi k}{n}$ für $k = 0, \ldots, n-1$. Ferner sei*

$$d_j(y) := \frac{1}{n} \sum_{l=0}^{n-1} y_l e^{-ijx_l} = \frac{1}{n} \sum_{l=0}^{n-1} y_l \varepsilon_n^{jl}, \quad j = 0, \ldots, n-1 .$$

Dann erfüllt das trigonometrische Polynom

$$T_n(y; x) := \sum_{j=0}^{n-1} d_j(y) e^{ijx}$$

die Interpolationsbedingungen $T_n(y; x_k) = y_k$ für alle $k = 0, \ldots, n-1$. Ferner ist $T_n(y; \cdot)$ das einzige Element aus $\mathcal{T}_n$, das diese Interpolationsbedingungen erfüllt.

Bemerkung 1.14. Für den Fall $y_k = f(x_k)$ beachte man die formale Ähnlichkeit zur Fourier-Teilsumme. Die Fourier-Koeffizienten werden hier durch eine Rechteckregel approximiert.

Beweis von Satz 1.13. Mit Lemma 1.12 ergibt sich

$$\begin{aligned} T_n(y;x_k) &= \sum_{j=0}^{n-1} d_j(y) e^{i2\pi jk/n} = \frac{1}{n}\sum_{j=0}^{n-1}\left(\sum_{l=0}^{n-1} y_l e^{-ijx_l}\right) e^{i2\pi jk/n} \\ &= \sum_{l=0}^{n-1} y_l \frac{1}{n}\sum_{j=0}^{n-1} e^{-i2\pi j(l-k)/n} \overset{\text{Lemma 1.12}}{=} \sum_{l=0}^{n-1} y_l \delta_{l-k,0} = y_k. \end{aligned}$$

Angenommen $T \in \mathcal{T}_n$ erfüllt ebenfalls die Interpolationsbedingungen. Dann hat $T - T_n(y;\cdot) \in \mathcal{T}_n$ n verschiedenen Nullstellen, denn für $k \in \{0,\dots,n-1\}$ gilt

$$T(x_k) - T_n(y;x_k) = y_k - y_k = 0\ .$$

Aus Satz 1.8 (ii) folgt, dass $T - T_n(y;\cdot)$ das Nullpolynom ist, d.h. es gilt $T = T_n(y;\cdot)$. □

1.2 Diskrete Fourier-Transformation

Definition 1.15. Wir bezeichnen den Vektor

$$\hat{y} \coloneqq d(y) \coloneqq (d_0(y),\dots,d_{n-1}(y))^T \in \mathbb{C}^n$$

als die *diskrete Fourier-Transformierte* von y. Die Abbildung $\mathcal{F}: \mathbb{C}^n \to C^n, y \mapsto \hat{y}$ heißt *diskrete Fourier-Transformation (DFT)*. Die Komponenten von $\hat{y}$ bezeichnen wir auch mit $\hat{y}_k$.

Bemerkung 1.16. Da die DFT linear ist, lässt sie sich als Matrix darstellen. Sei dazu $B = \left(b^0|\cdots|b^{n-1}\right) \in \mathbb{C}^{n\times n}$ die Matrix mit den Spalten

$$b^j = \left(e^{-2\pi ijk/n}\right)_{k=0}^{n-1} \equiv \left(1, e^{-2\pi ij/n},\dots,e^{-2\pi ij(n-1)/n}\right)^T \in \mathbb{C}^n\ .$$

Dann gilt $\hat{y} = \frac{1}{n}By$, also ist $\frac{1}{n}B$ die Matrixdarstellung der DFT. Die Matrix B ist voll besetzt und wird damit in der Praxis nicht explizit aufgestellt. Sie erlaubt es aber sehr einfach die inverse DFT herzuleiten.

Satz 1.17. *Die Menge der Vektoren $\{b^0,\dots,b^{n-1}\}$ bildet eine Orthogonalbasis des $\mathbb{C}^n$ bzgl. des unitären Standardskalarproduktes $\langle\cdot,\cdot\rangle\colon \mathbb{C}^n\times\mathbb{C}^n \to \mathbb{C}$. Genauer gilt*

$$\langle b^k, b^l\rangle = n\delta_{k,l}\ , \quad k,l \in \{0,\dots,n-1\}\ .$$

Insbesondere gilt $(\frac{1}{n}B)^{-1} = \overline{B}^T$, d.h. die Inverse der DFT ist

$$y_j = \sum_{k=0}^{n-1} \hat{y}_k e^{2\pi ijk/n},\ j = 0,\dots,n-1.$$

Beweis. Die Orthogonalität folgt mit Hilfe von Lemma 1.12

$$\langle b^k, b^l\rangle = \sum_{j=0}^{n-1} b_j^k \overline{b_j^l} = \sum_{j=0}^{n-1} e^{-2\pi ikj/n} e^{2\pi ilj/n} = \sum_{j=0}^{n-1} e^{-2\pi i(k-l)j/n} \overset{\text{Lemma 1.12}}{=} n\delta_{k-l,0} = n\delta_{k,l}.$$

Aus $\langle b^k, b^l\rangle = n\delta_{k,l}$ folgt $(\overline{B}^T B) = nI_n$ und daraus $(\frac{1}{n}B)^{-1} = \overline{B}^T$. Für die adjungierte Matrix $\overline{B}^T$ gilt

$$(\overline{B}^T)_{k,l} = (\overline{B})_{l,k} = \overline{b_l^k} = e^{2\pi ikl/n} .$$

Daraus folgt direkt die Darstellung der Inversen. □

Bemerkung 1.18. Mit anderen Worten ist die DFT ein Basiswechsel von der kanonischen Basis in die Basis $\overline{b}^0, \ldots, \overline{b}^{n-1}$, denn es gilt

$$y = \sum_{k=0}^{n-1} \hat{y}_k \overline{b^k} .$$

Wir bezeichnen die *inverse (diskrete) Fourier-Transformierte* von y mit $\check{y}$, bzw. die *inverse (diskrete) Fourier-Transformation* mit $\mathcal{F}^{-1}\colon \mathbb{C}^n \to \mathbb{C}^n$. Es gilt also $\check{y} = \mathcal{F}^{-1}(y)$.

Bemerkung 1.19. Seien $f : [0, 2\pi] \to \mathbb{C}$ und $y_k \coloneqq f(x_k)$. Neben der Interpretation der Formel für $d_j(y)$ als Quadraturformel für die Fourier-Koeffizienten gibt es weitere Analogien zur Fourier-Teilsumme. Für Funktionen $f, g : [0, 2\pi] \to \mathbb{C}$ definieren wir das (diskrete) innere Produkt $\langle\cdot,\cdot\rangle_n$ als:

$$\langle f, g\rangle_n \coloneqq \frac{1}{n}\sum_{l=0}^{n-1} f(x_l)\overline{g(x_l)}.$$

Mit den Grundschwingungen e_k gilt dann

$$d_j(y) = \frac{1}{n}\sum_{l=0}^{n-1} y_l e^{-ijx_l} = \frac{1}{n}\sum_{l=0}^{n-1} f(x_l)\overline{e_j(x_l)} = \langle f, e_j\rangle_n = \langle T_n(y;\cdot), e_j\rangle_n.$$

Mehr noch: Es gilt

$$\langle e_j, e_k\rangle_n = \frac{1}{n}\sum_{l=0}^{n-1} e_j(x_l)\overline{e_k(x_l)} = \frac{1}{n}\sum_{l=0}^{n-1} e^{ijx_l}\overline{e^{ikx_l}} = \frac{1}{n}\sum_{l=0}^{n-1} \overline{b_l^j}\,\overline{b_l^k} = \frac{1}{n}\overline{\langle b^j, b^k\rangle} = \delta_{j,k}.$$

Also sind die e_j nicht nur bezüglich des kontinuierlichen inneren Produkts $\langle\cdot,\cdot\rangle$, sondern auch bezüglich des diskreten Analogons $\langle\cdot,\cdot\rangle_n$ ein Orthonormalsystem. Ferner ist das trigonometrische Polynom $T_n(y;\cdot)$ eine Orthogonalprojektion von f auf $\mathcal{T}_m$ bezüglich des diskreten inneren Produkts $\langle\cdot,\cdot\rangle_n$.

Beispiel 1.20. Wir betrachten den Fall $n = 4$ und die folgenden Daten

x_k	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$
y_k	2	0	2	0

Die zugehörige Einheitswurzel ist $\varepsilon_4 = e^{-i\pi/2} = -i$. Weiterhin gilt:

$$d_0(y) = \frac{1}{4}\sum_{l=0}^{3} y_l 1 = \frac{1}{4}(2+0+2+0) = 1$$

$$d_1(y) = \frac{1}{4}\sum_{l=0}^{3} y_l(-i)^l = \frac{1}{4}(2\cdot 1 + 0\cdot(-i) + 2\cdot(-1) + 0\cdot(+i)) = 0$$

$$d_2(y) = \frac{1}{4}\sum_{l=0}^{3} y_l(-i)^{2l} = \frac{1}{4}(2\cdot 1 + 0\cdot(-i)^2 + 2\cdot(-i)^4 + 0\cdot(-i)^6) = 1$$

$$d_3(y) = \frac{1}{4}\sum_{l=0}^{3} y_l(-i)^{3l} = \frac{1}{4}(2\cdot 1 + 0\cdot(-i)^3 + 2\cdot(-i)^6 + 0\cdot(-i)^9) = 0$$

Damit folgt

$$T_4(y;x) = \sum_{j=0}^{3} d_j(y)e^{ijx} = 1\cdot e^0 + 1\cdot e^{2ix} = 1 + \cos 2x + i\sin 2x\ .$$

Der Real- und Imaginärteil dieses trigonometrischen Interpolationspolynoms ist in Abbildung 1.1 zusammen mit den Daten y skizziert.

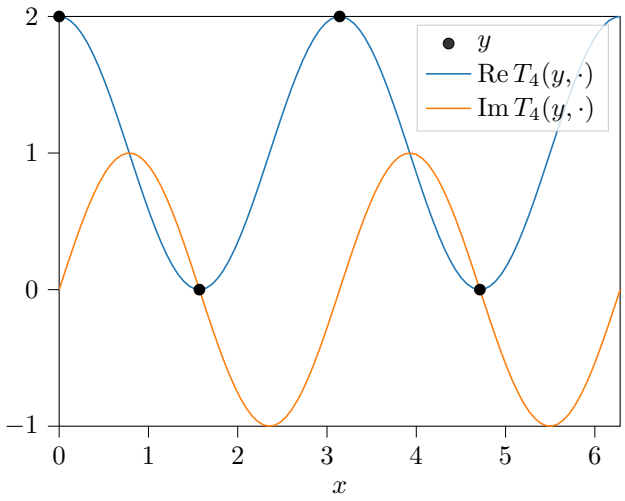


Abbildung 1.1: Skizze des Real- und Imaginärteils des trigonometrischen Polynoms $T_4(y;\cdot)$ und der Daten y.

Bemerkung 1.21. Es gibt diverse Varianten der diskreten Fourier-Transformation, u.a. die reelle trigonometrische Interpolation, die wir kurz darstellen wollen.

Zunächst stellt sich die Frage nach einem geeigneten Raum um reelle Werte y_k mit reellen Funktionen zu interpolieren. Als Ausgangspunkt betrachten wir hierzu Real- und Imaginärteil der Grundschwingungen e_j, also $\phi_j := \mathrm{Re}(e_j) = \cos(j\cdot)$ und $\psi_j := \mathrm{Im}(e_j) = \sin(j\cdot)$. Man rechnet leicht nach, das diese orthogonal bzgl. des Skalarproduktes $\langle\cdot,\cdot\rangle$ sind. Zusammen mit $\phi_0 \equiv 1$

und $\psi_0 \equiv 0$ führt das auf den reellen Raum

$$\hat{\mathcal{T}}_{2p+1} = \left\{ \alpha_0 + \sum_{j=1}^{p} (\alpha_j \cos(jx) + \beta_j \sin(jx)) : \ \alpha_j, \beta_j \in \mathbb{R} \right\} = \operatorname{span}\{\phi_0, \dots \phi_p, \psi_1, \dots \psi_p\}.$$

Aus der Orthogonalität der aufspannenden Funktionen folgt $\dim \hat{\mathcal{T}}_{2p+1} = 2p+1$.

Zu gegeben Werten $y_k \in \mathbb{R}$, $k = 0, \dots, n-1$, sowie den Stützstellen $x_k = \frac{2\pi k}{n}$ suchen wir nun ein $\hat{T}_n \in \hat{\mathcal{T}}_n$ so dass gilt

$$\hat{T}_n(x_k) = y_k \ , \quad k = 0, \dots, n-1 \ .$$

Wir nehmen dabei an, dass n ungerade ist, d.h. $n = 2p+1$ mit $p \in \mathbb{N}_0$.

Dieses Problem kann sehr ähnlich wie im komplexen Fall gelöst werden. Setzt man

$$A_j(y) \coloneqq \frac{2}{n} \sum_{l=0}^{n-1} y_l \cos(jx_l) \quad \text{und} \quad B_j(y) \coloneqq \frac{2}{n} \sum_{l=0}^{n-1} y_l \sin(jx_l) \ ,$$

so erfüllt

$$\hat{T}_n(y; x) = \frac{1}{2} A_0(y) + \sum_{j=1}^{\frac{n-1}{2}} \big(A_j(y) \cos(jx) + B_j(y) \sin(jx) \big)$$

die Interpolationsbedingungen $\hat{T}_n(y; x_k) = y_k$, $k = 0, \dots, n-1$. In der Tat ist auch hier Lemma 1.12 der Schlüssel um diese Aussage zu zeigen. Um Lemma 1.12 anwenden zu können, nutzt man die Identitäten

$$\cos(x) = \tfrac{1}{2} \left(e^{ix} + e^{-ix} \right) \ \text{und} \ \sin(x) = \tfrac{1}{2i} \left(e^{ix} - e^{-ix} \right) .$$

Die Koeffizienten $A_j(y)$ und $B_j(y)$ lassen sich effizient mit der diskreten Fourier-Transformation berechnen. Hierzu wird aus dem reellen Vektor y ein komplexer Vektor halber Länge konstruiert, auf den man die Fourier-Transformation anwendet. Hieraus kann man dann die gesuchten Koeffizienten mit sehr geringem Aufwand berechnen.

1.3 Schnelle Fourier-Transformation

Nun zur schon oft erwähnten schnellen Fourier-Transformation (Fast Fourier Transform, FFT).

Bemerkung 1.22. Wir erinnern uns, dass die Koeffizienten gegeben sind durch

$$d_j(y) = \frac{1}{n} \sum_{l=0}^{n-1} y_l e^{-i2\pi jl/n} = \frac{1}{n} \sum_{l=0}^{n-1} y_l \varepsilon_n^{jl} \ , \quad j = 0, \dots, n-1 \ ,$$

mit der n-ten Einheitswurzel ε_n. Naiv ausgeführt erfordert die diskrete Fourier-Transformation also die Bestimmung von n Koeffizienten, deren Berechnung jeweils $\mathcal{O}(n)$ Operationen erfordert, mithin insgesamt einen Aufwand von $\mathcal{O}(n^2)$. Man sieht sehr schnell, dass die Behandlung von Datensätzen mit $n \approx 10^6$ auf normalen Rechnern unmöglich wird (1 GFLOP = 10^9 Operationen). Eine solche Datenmenge ist jedoch alles andere als ungewöhnlich. Bei einer Audio-Samplingrate von 48 kHz, d.h. 48 000 Datenpunkten pro Sekunde entsprechen 5 Minuten schon $1.44 \cdot 10^7$ Datenpunkten. Durch geschicktes Rechnen mit komplexen Zahlen lässt sich der Aufwand auf $\mathcal{O}(n \log n)$, also praktisch lineares Verhalten, senken. Erst dies macht Audio- und Videokompression möglich. Die FFT gilt daher zurecht als einer der wichtigsten Algorithmen unserer Zeit.

Bemerkung 1.23. Wir betrachten zunächst den Fall, dass sich n faktorisieren lässt, so dass $n = n_1 n_2$ mit $n_1, n_2 \in \mathbb{N}$. Dies nutzen wir um die Summation über einen Index der Koeffizienten d_j in eine Summierung über zwei Indizes aufzuspalten. Seien dazu

$$l(a,b) = an_1 + b \text{ mit } a = 0, \dots, n_2 - 1 \text{ und } b = 0, \dots, n_1 - 1,$$
$$j(c,d) = cn_2 + d \text{ mit } c = 0, \dots, n_1 - 1 \text{ und } d = 0, \dots, n_2 - 1.$$

Für die Koeffizienten ergibt sich dann

$$d_{j(c,d)}(y) = \frac{1}{n} \sum_{l=0}^{n-1} y_l \varepsilon_n^{j(c,d)l} = \frac{1}{n} \sum_{a=0}^{n_2-1} \sum_{b=0}^{n_1-1} y_{l(a,b)} \varepsilon_n^{j(c,d)l(a,b)}.$$

Wegen

$$\varepsilon_n^{j(c,d)l(a,b)} = \varepsilon_n^{(cn_2+d)(an_1+b)} = \varepsilon_n^{(cn_2+d)an_1+(cn_2+d)b} = \varepsilon_n^{acn+b(cn_2+d)+adn_1} = \underbrace{\varepsilon_n^{acn}}_{=1} \varepsilon_n^{bj(c,d)} \varepsilon_{n_2}^{ad}$$

folgt somit

$$\begin{aligned} d_{j(c,d)}(y) &= \frac{1}{n} \sum_{a=0}^{n_2-1} \sum_{b=0}^{n_1-1} y_{l(a,b)} \varepsilon_n^{bj(c,d)} \varepsilon_{n_2}^{ad} \\ &= \frac{1}{n} \sum_{b=0}^{n_1-1} \varepsilon_n^{bj(c,d)} \underbrace{\sum_{a=0}^{n_2-1} y_{l(a,b)} \varepsilon_{n_2}^{ad}}_{=:\tilde{y}(b,d)} = \frac{1}{n} \sum_{b=0}^{n_1-1} \varepsilon_n^{bj(c,d)} \tilde{y}(b,d) \ . \end{aligned}$$

Diese Darstellung reduziert den Aufwand zu der Berechnung der d_j deutlich, denn:

- Der Aufwand zu Berechnung eines $\tilde{y}(b,d)$ ist $\mathcal{O}(n_2)$, insgesamt müssen wir n_1 ($b = 0, \dots, n_1 - 1$) mal n_2 ($d = 0, \dots, n_2 - 1$) dieser Werte berechnen, zusammen also ein Aufwand von $\mathcal{O}(n_2 n)$.
- Nach Berechnung der $\tilde{y}(b,d)$, verbleibt die Berechnung von n mal $d_{j(c,d)}$, jeweils mit einem Aufwand von $\mathcal{O}(n_1)$, zusammen also ein Aufwand von $\mathcal{O}(n_1 n)$.

Insgesamt ist der Aufwand also $\mathcal{O}((n_1 + n_2)n)$, was (bei großem n) wesentlich weniger ist als der Aufwand der naiven Umsetzung von $\mathcal{O}(n^2) = \mathcal{O}((n_1 n_2)n)$.

Bemerkung 1.24. Die obige Zerlegungsidee (das sogenannte Teile-und-herrsche-Verfahren, engl.: divide-and-conquer) kann rekursiv angewendet werden solange n_1 oder n_2 noch faktorisierbar sind. Der einfachste Fall ist hier, wenn n eine Zweierpotenz ist, d.h. $n = 2^L$ mit $L \in \mathbb{N}$. In diesem Fall läuft dies auf einen Aufwand von $\mathcal{O}(L2^L) = \mathcal{O}(n \log n)$ hinaus. Dies werden wir im Folgenden ausführen.

Satz 1.25. *Seien $y = (y_j)_{j=0}^{n-1} \in \mathbb{C}^n$ und $n = 2m$ mit $m \in \mathbb{N}$. Dann gilt für $k = 0, \dots, \frac{n-2}{2}$*

$$d_{2k}(y) = \frac{1}{m} \sum_{j=0}^{m-1} \frac{1}{2}(y_j + y_{j+m}) \varepsilon_m^{kj} \ ,$$
$$d_{2k+1}(y) = \frac{1}{m} \sum_{j=0}^{m-1} \frac{1}{2}(y_j - y_{j+m}) \varepsilon_{2m}^{j} \varepsilon_m^{kj} \ .$$

Beweis. Zunächst gilt

$$\varepsilon_{2m}^{2k(m+j)} = \varepsilon_{2m}^{2km} \cdot \varepsilon_{2m}^{2kj} = (e^{-i2\pi/(2m)})^{2km} \cdot (e^{-i2\pi/(2m)})^{2kj} = 1 \cdot (e^{-i2\pi/m})^{kj} = \varepsilon_m^{kj}.$$

Für die geraden Koeffizienten ergibt sich daraus

$$\begin{aligned} d_{2k}(y) &= \frac{1}{2m} \sum_{j=0}^{2m-1} y_j \varepsilon_{2m}^{j2k} = \frac{1}{2m} \left(\sum_{j=0}^{m-1} y_j \varepsilon_{2m}^{j2k} + \sum_{j=0}^{m-1} y_{j+m} \varepsilon_{2m}^{(j+m)2k} \right) \\ &= \frac{1}{2m} \sum_{j=0}^{m-1} \left(y_j \varepsilon_m^{jk} + y_{j+m} \varepsilon_{2m}^{2k(m+j)} \right) = \frac{1}{m} \sum_{j=0}^{m-1} \frac{1}{2} (y_j + y_{j+m}) \varepsilon_m^{kj}. \end{aligned}$$

Analog folgt aus

$$\varepsilon_{2m}^{(2k+1)(m+j)} = \varepsilon_{2m}^{2k(m+j)} \varepsilon_{2m}^{m+j} = \varepsilon_m^{kj} \varepsilon_{2m}^{m+j} = \varepsilon_m^{kj} (-1) \varepsilon_{2m}^{j}$$

für die ungeraden Koeffizienten

$$\begin{aligned} d_{2k+1}(y) &= \frac{1}{2m} \sum_{j=0}^{2m-1} y_j \varepsilon_{2m}^{j(2k+1)} = \frac{1}{2m} \left(\sum_{j=0}^{m-1} y_j \varepsilon_{2m}^{j(2k+1)} + \sum_{j=0}^{m-1} y_{j+m} \varepsilon_{2m}^{(j+m)(2k+1)} \right) \\ &= \frac{1}{2m} \sum_{j=0}^{m-1} \left(y_j \varepsilon_{2m}^{j2k} \varepsilon_{2m}^{j} + y_{j+m} \varepsilon_m^{kj} (-1) \varepsilon_{2m}^{j} \right) = \frac{1}{m} \sum_{j=0}^{m-1} \frac{1}{2} (y_j - y_{j+m}) \varepsilon_{2m}^{j} \varepsilon_m^{kj}. \end{aligned}$$

□

Bemerkung 1.26. Was hilft uns das? Die Identitäten für d_{2k} bzw. d_{2k+1} besagen, dass sich die Bestimmung der diskreten Fourier-Transformation der Länge $n = 2m$ auf die Durchführung von zwei diskreten Fourier-Transformationen der halben Länge $n/2 = m$ zurückführen lässt. Falls wir die Potenzen der Einheitswurzeln (genauer die Faktoren $\varepsilon_{2m}^j/2$) vorberechnen und speichern, so erfordert die Berechnung der Argumente für die Fourier-Transformationen der halben Länge m, also von

$$\frac{1}{2}(y_j + y_{j+m}) \quad \text{sowie} \quad \frac{1}{2}(y_j - y_{j+m})\varepsilon_{2m}^j \,, \quad j = 0, \ldots, m-1 \,,$$

jeweils $2m$, also insgesamt $4m = 2n$ Operationen (zusätzlich zu den beiden Fourier-Transformationen mit halber Länge). Ist nun $n = 2^L$ eine Zweierpotenz, so lässt sich diese Prozedur L mal wiederholen. Bezeichnet $A(n)$ den Aufwand zur Berechnung der diskreten Fourier-Transformation der Länge $n = 2^L$, so erhalten wir die folgende rekursive Beziehung

$$A(2^L) = 2 \cdot 2^L + 2A(2^{L-1}).$$

Per Induktion erhalten wir folgende Aussage.

Satz 1.27. *Die diskrete Fourier-Transformation lässt sich für $n = 2^L$ mit $L \in \mathbb{N}$ mit einem Aufwand von*

$$A(2^L) = L2^{L+1} = 2n \log_2 n$$

Operationen durchführen.

Beweis. Wie oben angedeutet beweisen wir die Aussage per Induktion nach L.

- Induktionsanfang $L = 1$: Folgt direkt aus $A(2) = 4$.
- Induktionsschluss $L - 1 \to L$: Es gilt

$$A(2^L) \overset{\text{s.o.}}{=} 2 \cdot 2^L + 2A(2^{L-1}) \overset{\text{I.V.}}{=} 2^{L+1} + 2(L-1)2^L = 2^{L+1} + (L-1)2^{L+1} = L2^{L+1}. \quad \square$$

Bemerkung 1.28.

(i) Eine direkte Umsetzung obiger Rekursion führt auf folgenden Algorithmus:

function FFT($y = (y_k)_{k=0}^{n-1} \in \mathbb{C}^n$)
if $n = 1$ **then**
  return y_0
else
  $a = \frac{1}{2}\text{FFT}\left((y_k + y_{k+n/2})_{k=0}^{n/2-1}\right)$
  $b = \frac{1}{2}\text{FFT}\left(\left(e^{\frac{-2\pi i k}{n}}(y_k - y_{k+n/2})\right)_{k=0}^{n/2-1}\right)$
  return $\left[a_0, b_0, a_1, b_1, \ldots, a_{n/2-1}, b_{n/2-1}\right]$
end if
end function

(ii) Das Aufwandsverhalten lässt sich auch im Wesentlichen realisieren, wenn n keine Zweierpotenz ist. Allerdings wird der Algorithmus dann technischer.

(iii) Es gibt viele Implementierungsdetails, die die FFT weiter beschleunigen. Implementierungen sind extrem hardwareorientiert und arbeiten oftmals direkt auf der Bit-Darstellung der Zahlen. Hiermit sind auch iterative statt rekursive Ansätze möglich.

Nun beispielhaft eine innermathematische Anwendung:

Definition 1.29. Die *(diskrete) periodische Faltung* zweier Vektoren $x, y \in \mathbb{C}^n$ ist definiert als

$$(x * y)_k = \frac{1}{n} \sum_{j=0}^{n-1} x_{[(k-j) \bmod n]} y_j \ , \quad k = 0, \ldots, n-1 \ ,$$

wenn $x = (x_0, \ldots, x_{n-1})^T$ und $y = (y_0, \ldots, y_{n-1})^T$.

Satz 1.30 (Diskretes Faltungstheorem)**.** *Seien $x, y \in \mathbb{C}^n$. Dann gilt*

$$\mathcal{F}(x * y) = \mathcal{F}(x) \odot \mathcal{F}(y),$$

wobei $\odot$ die elementweise Multiplikation zweier Vektoren bezeichnet. Mit anderen Worten wandelt die DFT die Faltung in eine punktweise Multiplikation um.

Beweis. Zu zeigen ist, dass

$$d_j(x * y) = d_j(x) d_j(y) \text{ für } j = 0, \ldots, n-1.$$

Dies folgt mit Hilfe von $\varepsilon_n^{jk} = \varepsilon_n^{j(k+n)}$:

$$\begin{aligned} d_j(x*y) &= \frac{1}{n}\sum_{l=0}^{n-1}(x*y)_l\varepsilon_n^{jl} = \frac{1}{n^2}\sum_{l=0}^{n-1}\sum_{k=0}^{n-1} x_{[(l-k) \bmod n]}y_k\varepsilon_n^{jl} \\ &= \frac{1}{n^2}\sum_{k=0}^{n-1}\underbrace{\sum_{l=0}^{n-1} x_{[(l-k) \bmod n]}\varepsilon_n^{j(l-k)}}_{=\sum_{l=0}^{n-1} x_l\varepsilon_n^{jl}} y_k\varepsilon_n^{jk} \\ &= \left(\frac{1}{n}\sum_{l=0}^{n-1} x_l\varepsilon_n^{jl}\right)\left(\frac{1}{n}\sum_{k=0}^{n-1} y_k\varepsilon_n^{jk}\right) = d_j(x)d_j(y)\ . \end{aligned}$$

□

Bemerkung 1.31. Das Faltungstheorem ermöglicht eine schnelle Faltung, d.h. die Faltung mit einem Aufwand von $\mathcal{O}(n\log n)$ zu berechnen. Hierzu werden die beiden Argumente der Faltung jeweils Fourier-transformiert, dann elementweise multipliziert und das Ergebnis mit der inversen Fourier-Transformation schließlich zurück transformiert, d.h.

$$(x*y) = \mathcal{F}^{-1}(\mathcal{F}(x)\odot\mathcal{F}(y))\ .$$

Beispiel 1.32 (Multiplikation großer Zahlen). Das Faltungstheorem kann man zur Multiplikation zweier großer (ganzer) Zahlen benutzen. Im Dezimalsystem seien

$$a = \sum_{i=0}^{m} a_i 10^i, \quad b = \sum_{j=0}^{n} b_j 10^j\ ,$$

wobei $a_0,\ldots,a_m,b_0,\ldots,b_n \in \{0,\ldots,9\}$. Dann gilt mit $a_i = 0$ für $i > m$ und $b_j = 0$ für $j > n$ und $j < 0$, dass

$$\begin{aligned} ab &= \left(\sum_{i=0}^{m} a_i 10^i\right)\left(\sum_{j=0}^{n} b_j 10^j\right) = \sum_{i=0}^{m}\sum_{j=0}^{n} a_i b_j 10^{i+j} = \sum_{k=0}^{n+m}\sum_{i+j=k} a_i b_j 10^{i+j} \\ &= \sum_{k=0}^{n+m} 10^k \sum_{l=0}^{k} a_l b_{k-l} = \sum_{k=0}^{n+m} 10^k \sum_{l=0}^{n+m} a_l b_{k-l}. \end{aligned}$$

Die Ziffern des Produkts sind also gegebenen durch die (lineare) Faltung von a und b, die sich mittels FFT extrem schnell berechnen lässt.

2 Finite Differenzenverfahren

2.1 Idee & Fragestellungen: das Poisson-Problem

Ein essentieller Baustein zur numerischen Lösung von Differentialgleichungen ist die Möglichkeit Ableitung numerisch zu approximieren. Eine sehr verbreitete Methode dazu sind sogenannte finite Differenzen.

Definition 2.1. Für $n \in \mathbb{N}$ seien $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T, \boldsymbol{h} = (h_1, \dots, h_n)^T \in \mathbb{R}^n$. Ferner sei $f \in C^k(\mathbb{R})$ mit $k \in \mathbb{N}$. Dann nennt man

$$\mathfrak{D}_{\boldsymbol{\alpha}}^{\boldsymbol{h}}[f](x) \coloneqq \sum_{j=1}^{n} \alpha_j f(x + h_j)$$

finite Differenz p-ter Ordnung für die k-te Ableitung von f an der Stelle $x \in \mathbb{R}$, falls

$$\mathfrak{D}_{\boldsymbol{\alpha}}^{\boldsymbol{h}}[f](x) = f^{(k)}(x) + \mathcal{O}(h^p)$$

gilt, wobei $h = \max\limits_{j=1,\dots,n} |h_j|$.

Beispiel 2.2.

(i) Für $h \in (0, \infty)$ heißen die Differenzen

$$\frac{f(x+h) - f(x)}{h} \quad \text{und} \quad \frac{f(x) - f(x-h)}{h}$$

Vorwärtsdifferenzenquotient bzw. *Rückwärtsdifferenzenquotient*. Falls $f \in C^2(\mathbb{R})$ folgt aus der Taylorentwicklung, dass

$$f(x+h) = f(x) + hf'(x) + \mathcal{O}(h^2) \;\Rightarrow\; \frac{f(x+h) - f(x)}{h} = f'(x) + \mathcal{O}(h)\,.$$

Analog zeigt man

$$\frac{f(x) - f(x-h)}{h} = f'(x) + \mathcal{O}(h)\,.$$

Beide Differenzenquotienten sind somit jeweils eine finite Differenz erster Ordnung für die erste Ableitung.

(ii) Die Differenz

$$\frac{f(x+h) - f(x-h)}{2h}$$

wird auch als *der zentrale Differenzenquotient* für f' bezeichnet. Falls $f \in C^3(\mathbb{R})$ gilt nach Taylor

$$f(x+h) = f(x) + hf'(x) + \tfrac{1}{2}h^2 f''(x) + \mathcal{O}(h^3)\,,$$
$$f(x-h) = f(x) - hf'(x) + \tfrac{1}{2}h^2 f''(x) + \mathcal{O}(h^3)\,.$$

Durch Subtrahieren der beiden Identitäten und Umformen erhält man

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \mathcal{O}(h^2)\,,$$

was zeigt, dass es sich um eine finite Differenz zweiter Ordnung für die erste Ableitung handelt.

(iii) Falls $f \in C^4(\mathbb{R})$ gilt nach Taylor

$$f(x+h) = f(x) + hf'(x) + \tfrac{1}{2}h^2 f''(x) + \tfrac{1}{6}h^3 f'''(x) + \mathcal{O}(h^4) \,,$$

$$f(x-h) = f(x) - hf'(x) + \tfrac{1}{2}h^2 f''(x) - \tfrac{1}{6}h^3 f'''(x) + \mathcal{O}(h^4) \,.$$

Durch Addieren der beiden Identitäten und Umformen erhält man

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = f''(x) + \mathcal{O}(h^2) \,.$$

Dies ist eine finite Differenz zweiter Ordnung für die zweite Ableitung und wird *der zentrale Differenzenquotient zweiter Ordnung* genannt.

Bemerkung 2.3.

(i) Die Taylorentwicklung ermöglicht auch einen allgemeineren Ansatz zur Konstruktion von finite Differenzen. Für $f \in C^{m+1}(\mathbb{R})$ gilt (siehe Mathe 1)

$$f(x+h_j) = \sum_{l=0}^{m} \frac{f^{(l)}(x)}{l!}(h_j)^l + \mathcal{O}\big((h_j)^{m+1}\big) \,.$$

Damit können wir also $f(x+h_j)$ in der finiten Differenz durch das Taylorpolynom ersetzen, was zu folgendem Ansatz führt:

$$\mathfrak{D}^{\boldsymbol{h}}_{\boldsymbol{\alpha}}[f](x) = \sum_{j=1}^{n} \alpha_j \sum_{l=0}^{m} \frac{f^{(l)}(x)}{l!}(h_j)^l = \sum_{l=0}^{m} f^{(l)}(x) \sum_{j=1}^{n} \frac{1}{l!}\alpha_j (h_j)^l \,.$$

Falls $\alpha_j, h_j \in \mathbb{R}$ für $j = 1, \ldots, n$ die Bedingungen

$$\sum_{j=1}^{n} \frac{1}{l!}\alpha_j (h_j)^l = \delta_{l,k} \quad \text{für } l = 0, \ldots, m$$

erfüllen und $m \geq k$, so ist $\mathfrak{D}^{\boldsymbol{h}}_{\boldsymbol{\alpha}}[f](x)$ eine finite Differenz für die k-te Ableitung von f (Übung: welcher Ordnung?). Für feste h_j ist für das Bestimmen der α_j ein lineares Gleichungssystem zu lösen. Falls $m+1 < n$, ist das Gleichungssystem unterbestimmt. Man sollte also $m+1 \geq n$ wählen.

Beispiel: $m = 1$, $n = 2$, $h_1 = 0$, $h_2 = h$, $k = 1$, führt auf

$$\begin{aligned} l = 0: &\quad \alpha_1 + \alpha_2 = 0 \,, \\ l = 1: &\quad \alpha_2 h = 1 \,. \end{aligned}$$

Die Lösung ist $\alpha_1 = -\frac{1}{h}$, $\alpha_2 = \frac{1}{h}$, also genau der Vorwärtsdifferenzenquotient.

(ii) Üblicherweise wählt man die h_j äquidistant mit $m \in \mathbb{N}_0$ Schritten vor x und $n \in \mathbb{N}_0$ Schritten nach x, d.h.

$$\boldsymbol{h} = \big(-mh, -(m-1)h, \ldots, -h, 0, h, \ldots, (n-1)h, nh\big)^T \in \mathbb{R}^{n+m+1}$$

mit $h \in (0, \infty)$. Damit ergibt sich

$$\mathfrak{D}_{\boldsymbol{\alpha}}^{\boldsymbol{h}}[f](x) = \sum_{j=-m}^{n} \alpha_j f(x + jh) \, .$$

Den Fall $m = n$ nennt man *zentrale Differenz*, die Fälle $m = 0$ bzw. $n = 0$ nennt man jeweils *einseitige Differenzen*, wobei $m = 0$ speziell *rechtsseitige Differenz / Vorwärtsdifferenz* und $n = 0$ *linksseitige Differenz / Rückwärtsdifferenz* heißen.

Beispiel 2.4. Hier noch zwei weitere Beispiele für finite Differenzen (Übung):

$$\frac{1}{2h}\left(f(x-2h) - 4f(x-h) + 3f(x)\right) = f'(x) + \mathcal{O}(h^2) \, ,$$

$$\frac{1}{h^2}\left(\tfrac{7}{54} f(x-2h) + \tfrac{81}{110} f(x - \tfrac{2}{3}h) - \tfrac{640}{297} f(x + \tfrac{1}{4}h) + \tfrac{58}{45} f(x+h)\right) = f''(x) + \mathcal{O}(h) \, .$$

Problem 2.5 (Poisson-Problem)**.** Sei $\Omega \subset \mathbb{R}^2$ offen und zusammenhängend. Zu einem gegebenen *Quellterm* $f \colon \Omega \to \mathbb{R}$ und gegebenen *Randwerten* $g \colon \partial\Omega \to \mathbb{R}$ finde $u \in C^2(\Omega) \cap C(\overline{\Omega})$, so dass

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{auf } \partial\Omega. \end{cases}$$

Hierbei ist Δ der Laplace-Operator, d.h. $\Delta u \equiv \operatorname{div}(\operatorname{grad} u) = \partial_{xx} u + \partial_{yy} u$.

Bemerkung 2.6. In der Literatur wird das Poisson-Problem oftmals nur mit homogenen (d.h. Null) Randwerten angegeben. Um zu sehen, dass dies keine größere Einschränkung darstellt, sei $G \colon \overline{\Omega} \to \mathbb{R}$ hinreichend glatt mit $G|_{\partial\Omega} = g$. Dann löst u das Poisson-Problem mit Quellterm f und Randwert g löst, genau dann wenn $w \coloneqq u - G$ das homogene Poisson-Problem mit Quellterm $f + \Delta G$ löst.

Die Idee ist es nun das Poisson-Problem mittels finiter Differenzen zu lösen. Einen solchen Ansatz nennt man auch *finite Differenzenverfahren* (kurz: *FD-Verfahren*).

Bemerkung 2.7. Wir betrachten Problem 2.5 auf dem Einheitsquadrat $\Omega = (0,1)^2$ (vgl. Abbildung 2.2a). Dazu verwenden wir ein sog. *kartesisches Gitter*, genauer gesagt zwei kartesische Gitter, nämlich

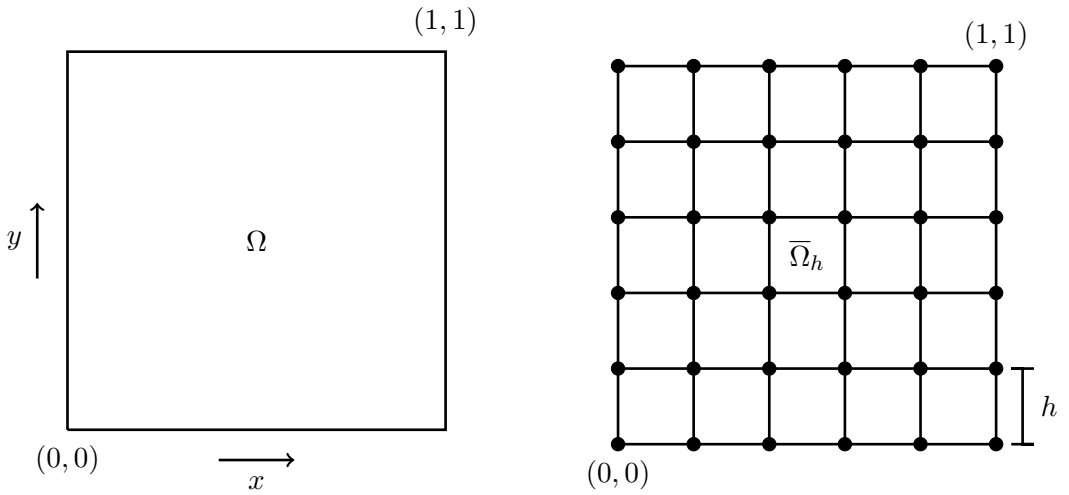
$$\Omega_h \coloneqq \{(ih, jh) \, : \, 1 \le i, j \le n-1\} \quad \text{und} \quad \overline{\Omega}_h \coloneqq \{(ih, jh) \, : \, 0 \le i, j \le n\}$$

mit *Gitterweite* $h = \frac{1}{n}$, $n \in \mathbb{N}$. In Abbildung 2.2b ist das Gitter für $n = 5$ skizziert. Hierbei bezeichnen wir das Gitter des offenen Gebiets mit Ω_h und das Gitter des abgeschlossenen Gebiets, d.h. mit Gitterpunkten auf dem Rand $\partial\Omega$, mit $\overline{\Omega}_h$. Achtung: Die Notation $\overline{\Omega}_h$ bedeutet also, dass $\overline{\Omega}$ mit h diskretisiert wird, es ist nicht der Abschluss von Ω_h gemeint!

Bemerkung 2.8. Wir werden im Folgenden hauptsächlich auf dem Einheitsquadrat mit den äquidistanten Gitterpunkten wie in Bemerkung 2.7 arbeiten. Andere rechteckige Gebiete und eine nicht äquidistante rechteckige Verteilung der Gitterpunkte sind aber ebenfalls analog möglich.

Statt die Lösung des Poisson-Problems u als Funktion auf Ω zu bestimmen, wollen wir nur die Werte von u auf den Gitterpunkten berechnen, d.h. gesucht sind die Punktwerte (die „Unbekannten“)

$$\left\{u(x_i, y_i) \, : \, (x_i, y_i) \in \overline{\Omega}_h\right\} \, ,$$



(a) Einheitsquadrat $\Omega = (0,1)^2$ (b) Gitter

Abbildung 2.2: Einheitsquadrat und ein Gitter mit Gitterweite $h = \frac{1}{5}$.

für gegebene Gitter Ω_h und $\overline{\Omega}_h$. Zur Approximation von Δu nutzen wir formal den zentralen Differenzenquotienten 2. Ordnung, jeweils getrennt für die 2. Ableitung nach x und die 2. Ableitung nach y. Ist u glatt genug, so können wir für ein festes y auf die Abbildung $x \mapsto u(x,y)$ den zentralen Differenzenquotienten anwenden und erhalten

$$\frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} = \partial_{xx} u(x,y) + \mathcal{O}(h^2) \ ,$$

für $(x,y) \in \Omega$ und $h > 0$ hinreichend klein. Analog approximiert man $\partial_{yy} u$ und erhält insgesamt die *Differenzenformel*

$$(\Delta_h u)(x,y) \coloneqq \tfrac{1}{h^2}\big[u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) - 4u(x,y)\big] \ , \quad (2.1)$$

die den Laplace-Operator in den Gitterpunkten $(x,y) = (x_i, y_i) \equiv \xi \in \Omega_h$ wie folgt approximiert:

$$(\Delta_h u)(x,y) = (\Delta u)(x,y) + O(h^2) \ .$$

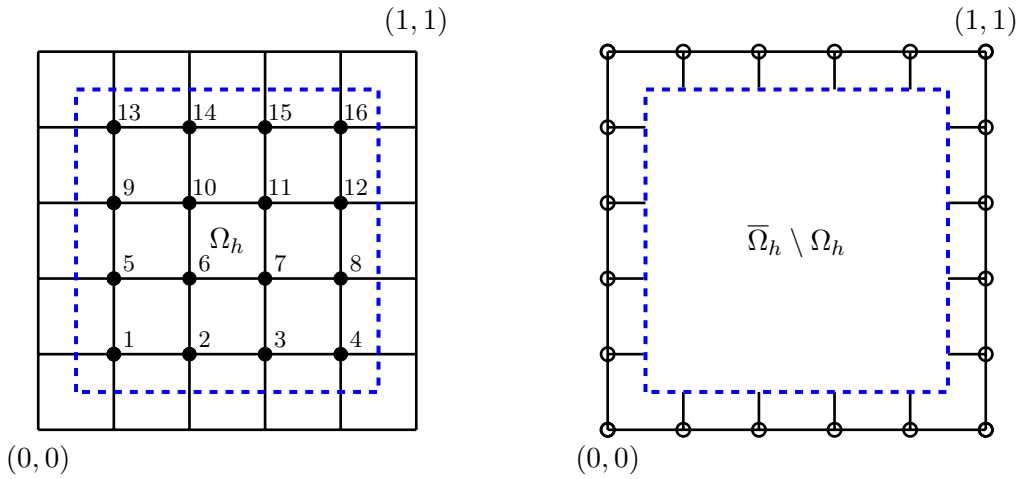
Eine andere, häufig verwendete, Schreibweise ist die des *Differenzensterns* (engl.: *stencil*)

$$[-\Delta_h]_\xi = \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} \ , \quad \xi \in \Omega_h \ . \quad (2.2)$$

Definition 2.9. Der Raum

$$l^2(\Omega_h) \coloneqq \left\{ u : \Omega_h \to \mathbb{R} \ : \ \sum_{\xi \in \Omega_h} u(\xi)^2 < \infty \right\}$$

ist die Menge aller *Gitterfunktionen* auf dem Gitter Ω_h. Analog wird $l^2(\overline{\Omega}_h)$ definiert.



(a) Gitter mit Standardanordnung der Gitterpunkte.

(b) Gitterpunkte auf dem Rand $\overline{\Omega}_h \setminus \Omega_h$.

Abbildung 2.3: Gitterpunkte mit Standardanordnung und Gitterpunkte auf dem Rand.

Bemerkung 2.10.

- Durch

$$\langle u, v\rangle_h := \sum_{\xi\in\Omega_h} u(\xi)v(\xi)\ , \quad \|u\|_2 = \sqrt{\langle u, u\rangle_h} \quad \text{und} \quad \|u\|_\infty = \max_{\xi\in\Omega_h} |u(\xi)|$$

  erhält man ein Skalarprodukt, sowie zwei verschieden Normen auf dem Raum der Gitterfunktionen $l^2(\Omega_h)$. Analoges gilt für $l^2(\overline{\Omega}_h)$.

- Da Ω_h nur endlich viele Punkte enthält, ist die Bedingung $\sum_{\xi\in\Omega_h} u(\xi)^2 < \infty$ für jedes $u : \Omega_h \to \mathbb{R}$ erfüllt. Relevant ist diese Bedingung also nur für Definitionsbereiche mit unendlich vielen Punkten.

Nun können wir das **diskretisierte Poisson-Problem** formulieren:

Problem 2.11 (Diskretisiertes Poisson-Problem)**.** Zu gegebenem Quellterm f und Randwerten g finde $u_h \in l^2(\overline{\Omega}_h)$, so dass

$$\begin{cases} -(\Delta_h u_h)(\xi) = f(\xi) & \text{für } \xi \in \Omega_h \\ u_h(\xi) = g(\xi) & \text{für } \xi \in \overline{\Omega}_h \setminus \Omega_h. \end{cases} \tag{2.3}$$

Bemerkung 2.12. Da $u_h \mapsto \Delta_h u_h$ linear in u_h ist, beschreibt (2.3) ein lineares Gleichungssystem und lässt sich als Matrix-Vektor-Produkt

$$Ax = b$$

schreiben. Die tatsächliche Form hängt von der Nummerierung der Unbekannten bzw. Gitterpunkte ab. Wir wählen die Standardanordnung, d.h. die Gitterpunkte werden Zeilenweise durchnummeriert (s. Abbildung 2.3a). Damit folgt für die Darstellung:

$$A_1 x = b\ , \quad A_1 \in \mathbb{R}^{m\times m},\ m := (n-1)^2\ , \tag{2.4}$$

mit

$$A_1 = h^{-2}\begin{pmatrix} T & -I & & & 0 \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ 0 & & & -I & T \end{pmatrix} \quad \text{und} \quad x = \begin{pmatrix} u(h,h) \\ u(2h,h) \\ \vdots \\ u((n-1)h,h) \\ u(h,2h) \\ \vdots \\ u((n-1)h,2h) \\ \vdots \\ u((n-1)h,(n-1)h) \end{pmatrix} \in \mathbb{R}^m \ .$$

Hier ist I die Einheitsmatrix $I \in \mathbb{R}^{(n-1)\times(n-1)}$ und T die Matrix

$$T = \begin{pmatrix} 4 & -1 & & & 0 \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ 0 & & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{(n-1)\times(n-1)} \ ,$$

Die Einträge ergeben sich aus den Koeffizienten der Differenzenformel (2.1) bzw. des Differenzensterns (2.2). Die rechte Seite $b \in \mathbb{R}^m$ setzt sich aus der rechten Seite f des Poisson-Problems und den Randwerten g zusammen (vgl. (2.3)). Der Vektor b lässt sich insgesamt schreiben als

$$b = (b_1^T,\ b_2^T,\ \ldots,\ b_{n-2}^T,\ b_{n-1}^T)^T \in \mathbb{R}^m \ ,$$

wobei die $n-1$ Vektoren der Länge $n-1$ gegebenen sind durch

$$b_1 = \begin{pmatrix} f(h,h) + h^{-2}\left(g(h,0) + g(0,h)\right) \\ f(2h,h) + h^{-2} g(2h,0) \\ \vdots \\ f(1-2h,h) + h^{-2} g(1-2h,0) \\ f(1-h,h) + h^{-2}\left(g(1-h,0) + g(1,h)\right) \end{pmatrix} \ ,$$

$$b_j = \begin{pmatrix} f(h,jh) + h^{-2} g(0,jh) \\ f(2h,jh) \\ \vdots \\ f(1-2h,jh) \\ f(1-h,jh) + h^{-2} g(1,jh) \end{pmatrix} \ , \quad 2 \le j \le n-2 \ ,$$

$$b_{n-1} = \begin{pmatrix} f(h,1-h) + h^{-2}\left(g(h,1) + g(0,1-h)\right) \\ f(2h,1-h) + h^{-2} g(2h,1) \\ \vdots \\ f(1-2h,1-h) + h^{-2} g(1-2h,1) \\ f(1-h,1-h) + h^{-2}\left(g(1-h,1) + g(1,1-h)\right) \end{pmatrix} \ .$$

Abbildung 2.4 zeigt eine (numerische) Lösung von Problem 2.11 mit $f \equiv 1$ und $g \equiv 0$ für $n = 40$.

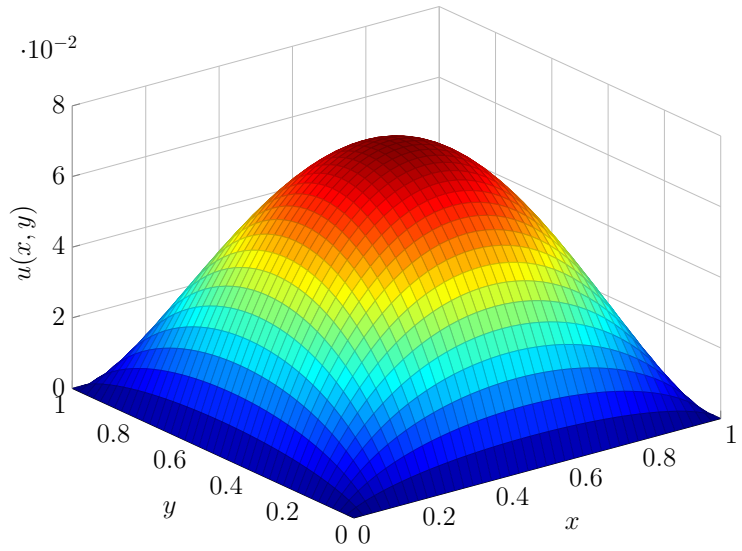


Abbildung 2.4: Lösung der diskretisierten Poisson-Gleichung für $f \equiv 1$ und $g \equiv 0$ auf einem Gitter mit $n = 40$.

Es stellen sich zwei wichtige Fragen:

(i) Wie hängt die Genauigkeit der Approximation u_h von der Diskretisierung, insbesondere der Gitterweite h ab? Das führt zu der Frage nach dem sog. *Diskretisierungsfehler*.

(ii) Was sind gute Verfahren zum Lösen von $A_1 x = b$? Kann man ausnutzen, dass die Matrix A_1 *dünnbesetzt* ist?

2.2 Fourier-Analyse für das Poisson-Problem

Im Folgenden wollen wir uns auf das diskretisierte Poisson-Problem 2.11 auf dem Einheitsquadrat mit kartesischen Gittern beschränken. Zur Fehleranalyse wollen wir daher die Matrix der Matrix-Vektor-Darstellung (2.4) der Diskretisierung verwenden. Dazu wird eine präzisere Beschreibung der Beziehung zwischen der Matrix-Vektor-Darstellung und der Diskretisierung (2.3) benötigt.

Bemerkung 2.13. Eine bzgl. des Skalarproduktes $\langle \cdot, \cdot \rangle_h$ orthogonale Basis des Raums $l^2(\Omega_h)$, der alle Gitterfunktionen auf Ω_h beinhaltet, ist gegeben durch die Basis-Gitterfunktionen

$$\phi_i(\xi_j) = \delta_{i,j} \ , \quad 1 \leq i, j \leq m \ ,$$

mit den Gitterpunkten $\xi_j \in \Omega_h$. Eine Gitterfunktion $v_h \in l^2(\Omega_h)$ lässt sich dann schreiben als

$$v_h = \sum_{i=1}^{m} x_i \phi_i \text{ mit } x_i = v_h(\xi_i) \ .$$

Der Differenzenoperator $\Delta_h \colon l^2(\overline{\Omega}_h) \to l^2(\Omega_h)$ gegeben durch

$$u \mapsto (\Omega_h \to \mathbb{R}, \xi \mapsto (\Delta_h u)(\xi)) \ ,$$

ist offenbar nicht bijektiv, denn $\dim(l^2(\overline{\Omega}_h)) = (n+1)^2 > (n-1)^2 = \dim(l^2(\Omega_h))$. Betrachtet man allerdings den Teilraum

$$l_0^2(\overline{\Omega}_h) \coloneqq \left\{ v_h \in l^2(\overline{\Omega}_h) \ : \ v_h(\xi) = 0 \text{ für alle } \xi \in \overline{\Omega}_h \setminus \Omega_h \right\} \subset l^2(\overline{\Omega}_h) \ ,$$

so ist der Operator eingeschränkt auf diesen Teilraum, d.h.

$$\Delta_h \colon l_0^2(\overline{\Omega}_h) \to l^2(\Omega_h)$$

bijektiv (Übung). Der Raum $l_0^2(\overline{\Omega}_h)$ ist der Raum der Gitterfunktionen, die Null auf dem Rand sind. Da die Funktionen $(\phi_i)_{1\leq i\leq m}$ eine Basis von $l^2(\Omega_h)$ bilden, lässt sich der Differenzenoperator $\Delta_h \colon l_0^2(\overline{\Omega}_h) \to l^2(\Omega_h)$ in dieser Basis schreiben. Dies ist aber gerade die Matrix A_1 in (2.4), d.h. es gilt

$$\Delta_h v_h = w_h \equiv \sum_{i=1}^{m} y_i \phi_i \;\Leftrightarrow\; A_1 x = y \; .$$

Daraus lässt sich ebenfalls die Äquivalenz der Normen

$$\|v_h\|_* = \|x\|_* \; , \quad \|\Delta_h\|_* = \|A_1\|_* \; , \quad \|\Delta_h^{-1}\|_* = \|A_1^{-1}\|_* \; , \; * \in \{2, \infty\} \; .$$

folgern (Übung).

Durch die Verwendung eines äquidistanten Gitters erhalten wir einen Differenzenstern der in allem Gitterpunkten gleich ist (2.2). Das erlaubt die Analyse des Problems mittels der sog. *Fourieranalyse*.

Bemerkung 2.14. Da die Matrix $A_1 \in \mathbb{R}^{m\times m}$ symmetrisch ist, folgt dass alle Eigenwerte reell sind, d.h. $\lambda_i(A_1) \in \mathbb{R}$. Außerdem existiert eine orthogonale Basis aus Eigenvektoren von A_1. Mit Hilfe der Fourieranalyse lassen sich diese Eigenwerte und -vektoren explizit angeben. Die trigonometrische Funktionen

$$e^{\nu,\mu}(x, y) \coloneqq \sin(\nu\pi x)\sin(\mu\pi y) \; , \quad \nu, \mu \in \mathbb{N} \; ,$$

sind Null auf dem Rand des Einheitsquadrats $\Omega = (0, 1)^2$. Außerdem handelt es sich hierbei um Eigenfunktionen des Laplace-Operators, denn es gilt

$$-\Delta e^{\nu,\mu} = \big((\pi\nu)^2 + (\pi\mu)^2\big) e^{\nu,\mu} \; .$$

Für unsere Analyse beschränken wir die Eigenfunktionen auf unser Gitter

$$e_h^{\nu,\mu}(\xi) \coloneqq e^{\nu,\mu}(\xi) \quad \text{für} \quad \xi \in \overline{\Omega}_h \quad \text{und} \quad 1 \leq \nu, \mu \leq n-1 \; .$$

Wegen

$$\begin{aligned}\langle e^{\nu,\mu}, e^{\nu',\mu'}\rangle_h = \langle e_h^{\nu,\mu}, e_h^{\nu',\mu'}\rangle_h &= \sum_{\xi\in\Omega_h} e_h^{\nu,\mu}(\xi) e_h^{\nu',\mu'}(\xi) = \sum_{j=1}^{n-1}\sum_{k=1}^{n-1} e_h^{\nu,\mu}(jh, kh) e_h^{\nu',\mu'}(jh, kh) \\ &= \sum_{j=1}^{n-1} \sin(\nu\pi jh)\sin(\nu'\pi jh) \sum_{k=1}^{n-1} \sin(\mu\pi kh)\sin(\mu'\pi kh)\end{aligned}$$

lässt sich mit $\sin(x) = \frac{1}{2i}\left(e^{ix} - e^{-ix}\right)$ und Lemma 1.12 zeigen (Übung), dass

$$\sum_{j=1}^{n-1} \sin(\nu\pi jh)\sin(\nu'\pi jh) = \frac{n}{2}\delta_{\nu,\nu'} \; .$$

Hieraus folgt, dass für die Eigenfunktionen

$$\langle e_h^{\nu,\mu}, e_h^{\nu',\mu'}\rangle_h = 0 \quad \text{für } (\nu,\mu) \neq (\nu',\mu') \,,$$
$$\text{und} \quad \left\|e_h^{\nu,\mu}(\xi)\right\|_2^2 = \frac{1}{4}h^{-2}$$

gilt. Also sind die Eigenfunktionen $(e_h^{\nu,\mu})_{1\leq\nu,\mu\leq n-1}$ orthogonal. Zusammen mit $\dim(l^2(\Omega_h)) = (n-1)^2$ folgt, dass sie eine orthogonale Basis von $l^2(\Omega_h)$ bilden.

Satz 2.15 (Eigenwerte des diskreten Laplace-Operators)**.** *Es gilt*

$$-(\Delta_h e_h^{\nu,\mu})(\xi) = \lambda_{\nu,\mu} e_h^{\nu,\mu}(\xi) \quad \textit{für} \quad \xi \in \Omega_h \,, 1 \leq \nu,\mu \leq n-1$$

mit den Eigenwerten

$$\lambda_{\nu,\mu} = \tfrac{4}{h^2}(\sin^2(\tfrac{1}{2}\pi\nu h) + \sin^2(\tfrac{1}{2}\pi\mu h)) \,.$$

Beweis. Diese Behauptung sieht man folgendermaßen. Zunächst gilt

$$\begin{aligned}
-(\Delta_h e_h^{\nu,\mu})(x,y) &= \tfrac{-1}{h^2}[e_h^{\nu,\mu}(x-h,y) + e_h^{\nu,\mu}(x+h,y) - 2e_h^{\nu,\mu}(x,y) \\
&\qquad + e_h^{\nu,\mu}(x,y-h) + e_h^{\nu,\mu}(x,y+h) - 2e_h^{\nu,\mu}(x,y)] \\
&= \tfrac{1}{h^2}[(-\sin(\nu\pi(x-h)) + 2\sin(\nu\pi x) - \sin(\nu\pi(x+h)))\sin(\mu\pi y) \\
&\qquad + \sin(\nu\pi x)\,(-\sin(\mu\pi(y-h)) + 2\sin(\mu\pi y) - \sin(\mu\pi(y+h)))]
\end{aligned}$$

Aus den trigonometrischen Identitäten

$$\sin(x+y) = \sin x\cos y + \sin y\cos x \quad \text{und} \quad \sin^2 x = \tfrac{1}{2}(1-\cos 2x)$$

sowie daraus, dass Kosinus gerade und Sinus ungerade ist, folgt

$$\begin{aligned}
&-\sin(\nu\pi(x-h)) + 2\sin(\nu\pi x) - \sin(\nu\pi(x+h)) \\
=& -\sin(\nu\pi x)\cos(-\nu\pi h) - \sin(-\nu\pi h)\cos(\nu\pi x) + 2\sin(\nu\pi x) \\
&-\sin(\nu\pi x)\cos(\nu\pi h) - \sin(\nu\pi h)\cos(\nu\pi x) \\
=& -2\sin(\nu\pi x)\cos(-\nu\pi h) + 2\sin(\nu\pi x) \\
=& \,2\sin(\nu\pi x)(1-\cos(\nu\pi h)) = 4\sin^2(\tfrac{1}{2}\nu\pi h)\sin(\nu\pi x).
\end{aligned}$$

Insgesamt erhalten wir daher

$$\begin{aligned}
-(\Delta_h e_h^{\nu,\mu})(x,y) &= \tfrac{4}{h^2}[\sin^2(\tfrac{1}{2}\nu\pi h)\sin(\nu\pi x)\sin(\mu\pi y) + \sin(\nu\pi x)\sin^2(\tfrac{1}{2}\mu\pi h)\sin(\mu\pi y)] \\
&= \lambda_{\nu,\mu} e_h^{\nu,\mu}(x,y).
\end{aligned}$$

□

Die Eigenfunktionen $e^{\nu,\mu}$ des Laplace-Operators sind (eingeschränkt auf $\overline{\Omega}_h$) also auch Eigenfunktionen des diskreten Laplace-Operators. In Vektorschreibweise erhält man dann mit $e_h^{\nu,\mu} \equiv \left(e_h^{\nu,\mu}(\xi_1),\ldots,e_h^{\nu,\mu}(\xi_m)\right)^T \in \mathbb{R}^m$, $m=(n-1)^2$, dem Vektor der Werte der Eigenfunktion auf Ω_h, dass

$$A_1 e_h^{\nu,\mu} = \lambda_{\nu,\mu} e_h^{\nu,\mu}$$

gilt. Also sind die $e_h^{\nu,\mu}$ orthogonale Eigenvektoren der Matrix A_1 zu den zugehörigen Eigenwerten $\lambda_{\nu,\mu}$ aus Satz 2.15. Da die Eigenwerte positiv sind, folgt daraus sofort:

Lemma 2.16. *Die Matrix A_1 der diskretisierten Poisson-Gleichung ist symmetrisch positiv definit.*

Außerdem lässt sich mit den Eigenwerten die Konditionszahl der Matrix A_1 leicht abschätzen.

Lemma 2.17. *Für die Konditionszahl der Matrix A_1 gilt*

$$\kappa_2(A_1) = \frac{\cos^2(\frac{1}{2}\pi h)}{\sin^2(\frac{1}{2}\pi h)} = \left(\frac{2}{\pi h}\right)^2 \left(1 + \mathcal{O}(h^2)\right) .$$

Beweis. Da die Matrix symmetrisch positiv definit ist und $\sin^2$ auf $[0, \frac{\pi}{2}]$ wachsend ist, gilt für die Konditionszahl

$$\kappa_2(A_1) = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{\lambda_{n-1,n-1}}{\lambda_{1,1}} = \frac{\frac{4}{h^2}(2\sin^2(\frac{1}{2}\pi(n-1)h))}{\frac{4}{h^2}(2\sin^2(\frac{1}{2}\pi 1h))} = \frac{\sin^2(\frac{1}{2}\pi - \frac{1}{2}\pi h)}{\sin^2(\frac{1}{2}\pi h)} = \frac{\cos^2(\frac{1}{2}\pi h)}{\sin^2(\frac{1}{2}\pi h)} .$$

Hier haben wir $\sin(x+\frac{\pi}{2}) = \cos(x)$ benutzt. Zur Abschätzung der Konditionszahl in Abhängigkeit von der Gitterweite h nutzen wir die Taylorentwicklungen (Übung)

$$\cos(x) = 1 - \tfrac{1}{2}x^2 + \mathcal{O}(x^4) \text{ und } \frac{x}{\sin x} = 1 + \tfrac{1}{6}x^2 + \mathcal{O}(x^4) .$$

Damit folgt

$$\frac{\cos^2(\frac{1}{2}\pi h)}{\sin^2(\frac{1}{2}\pi h)} = \frac{1}{(\frac{1}{2}\pi h)^2}\cos^2(\tfrac{1}{2}\pi h)\left(\frac{\frac{1}{2}\pi h}{\sin(\frac{1}{2}\pi h)}\right)^2 = \left(\frac{2}{\pi h}\right)^2\left(1 + \mathcal{O}(h^2)\right) . \qquad \square$$

Die zwei Eigenschaften der Matrix A_1, die wir gerade gezeigt haben, sind wichtig für die numerische Lösung von Problem 2.11.

Bemerkung 2.18 (Eigenschaften der Matrix A_1).

(i) Die Tatsache, dass die Matrix A_1 symmetrisch positiv definit ist, ist eine gute Eigenschaft, die von Lösungsverfahren für lineare Gleichungssysteme ausgenutzt werden kann (später mehr).

(ii) Die Konditionszahl wächst quadratisch mit abnehmender Gitterweite $\kappa_2(A_1) \sim \frac{1}{h^2}$. Wie wir im Kapitel über iterative Lösungsverfahren sehen werden, ist dies schlecht für die numerische Lösung des Problems.

2.3 Konvergenztheorie

Es stellt sich nun noch die Frage, ob das vorgestellte Verfahren überhaupt die richtige Lösung approximiert und, wenn ja, mit welcher Genauigkeit.

Definition 2.19. Sei $u_h \in l^2(\overline{\Omega}_h)$ die Lösung von Problem 2.11 und $u \in C^2(\Omega) \cap C(\overline{\Omega})$ die Lösung von Problem 2.5. Dann nennt man

$$e_h := u|_{\overline{\Omega}_h} - u_h$$

den *Diskretisierungsfehler* des FD-Verfahrens zur Lösung der Poisson-Problems.

Bemerkung 2.20. u_h ist eine approximierte Lösung des Poisson-Problems und u die exakte Lösung. Damit misst e_h gerade die Diskrepanz zwischen der approximierten und der exakten Lösung, wenn man die exakte Lösung an den entsprechenden Gitterpunkten auswertet.

Da die approximierte Lösung die Randdaten erfüllt, gilt $u_h(\xi) = u(\xi)$ für alle $\xi \in \overline{\Omega}_h \setminus \Omega_h$. Daraus folgt, dass der Diskretisierungsfehler auf dem Rand verschwindet, d.h. $e_h(\xi) = 0$ für alle $\xi \in \overline{\Omega}_h \setminus \Omega_h$ und somit gilt $e_h \in l_0^2(\overline{\Omega}_h)$.

Der Diskretisierungsfehler kann wie folgt abgeschätzt werden:

$$\|e_h\| = \|\Delta_h^{-1}\Delta_h e_h\| = \|\Delta_h^{-1}(-\Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h})\| \leq \|\Delta_h^{-1}\|\| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\| \,. \tag{2.5}$$

Hieraus folgt sofort folgender Satz:

Satz 2.21. *Seien u, u_h, e_h wie in Definition 2.19 und $\|\cdot\|$ eine Norm in $l^2(\Omega_h)$. Falls*

(i) $\|(-\Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h})\| \to 0$ *für* $h = \frac{1}{n} \to 0$, *(Konsistenz)*

(ii) $\|\Delta_h^{-1}\| \leq C$ *für alle* $h = \frac{1}{n} > 0$ *(Stabilität)*

erfüllt sind, dann folgt $\|e_h\| \to 0$ für $h \to 0$. In diesem Sinne konvergiert u_h gegen u.

Achtung: $m = \dim(l^2(\Omega_h))$ und somit auch $\|\cdot\|$ hängen von h ab.

Bemerkung 2.22. Obiger Satz bedeutet im Wesentlichen:

Konsistenz + Stabilität ⇒ Konvergenz

Die beiden Terme in der Abschätzung des Diskretisierungsfehler bedeuten folgendes:

- Der Term $\| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|$ quantifiziert die Größe des Defekts, der resultiert, wenn man die exakte Lösung in das Diskretisierungsverfahren einsetzt. Der Term wird auch *Konsistenzfehler* genannt.

- Der Term $\|\Delta_h^{-1}\|$ misst die *Stabilität* des Verfahrens. Er bewertet die Fortpflanzung des Konsistenzfehlers und Verstärkung von Daten- und Rundungsfehlern.

Konsistenz erhält man durch eine entsprechende Taylorentwicklung, Stabilität durch Untersuchen der Matrix. Zur Analyse des Verfahrens bleibt noch die Wahl einer geeigneten Norm. Die stärkste Forderung ist die Wahl der ∞-Norm. In dem Fall ist der Fehler in jedem Punkt gleichmäßig beschränkt.

Bei ausreichender Glattheit der Lösung u lässt sich für die Konsistenz folgende Abschätzung zeigen:

Lemma 2.23 (Konsistenz)**.** *Seien $u \in C^4(\overline{\Omega})$ und $C := \max\{\|\frac{\partial^4 u}{\partial x^4}\|_{\infty,\overline{\Omega}}, \|\frac{\partial^4 u}{\partial y^4}\|_{\infty,\overline{\Omega}}\}$. Dann gelten:*

(i) $\| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|_\infty \leq \frac{1}{6}Ch^2$,

(ii) $\| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|_2 \leq \frac{1}{6}Ch$.

Beweis. Zu einem $g \in C^4(\mathbb{R})$ und $x \in \mathbb{R}$, $h > 0$, existieren nach Taylor $\eta_1 \in (x, x+h)$ und $\eta_2 \in (x-h, x)$ mit

$$g(x+h) = g(x) + hg'(x) + \tfrac{1}{2}h^2 g''(x) + \tfrac{1}{6}h^3 g'''(x) + \tfrac{1}{24}h^4 g^{(4)}(\eta_1) ,$$
$$g(x-h) = g(x) - hg'(x) + \tfrac{1}{2}h^2 g''(x) - \tfrac{1}{6}h^3 g'''(x) + \tfrac{1}{24}h^4 g^{(4)}(\eta_2) .$$

Durch Addieren der beiden Identitäten, Umformen und dem Zwischenwertsatz (es existiert $\eta \in (x-h, x+h)$ mit $g(\eta) = \frac{1}{2}(g(\eta_1) + g(\eta_2))$) erhält man

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + \tfrac{1}{12}h^2 g^{(4)}(\eta) .$$

Sei nun $\xi = (x, y) \in \Omega_h$. Dann existieren $\eta \in (x-h, x+h)$ und $\tilde{\eta} \in (y-h, y+h)$, so dass

$$\begin{aligned}(-\Delta_h u)(\xi) &= -\frac{\partial^2 u}{\partial x^2}(\xi) - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\eta, y) - \frac{\partial^2 u}{\partial y^2}(\xi) - \frac{h^2}{12}\frac{\partial^4 u}{\partial y^4}(x, \tilde{\eta}) \\ &= \underbrace{-\Delta u(\xi)}_{=f(\xi)} - \frac{h^2}{12}\left(\frac{\partial^4 u}{\partial x^4}(\eta, y) + \frac{\partial^4 u}{\partial y^4}(x, \tilde{\eta})\right) .\end{aligned}$$

Daraus folgt

$$\max_{\xi \in \Omega_h} |(-\Delta_h u)(\xi) - f(\xi)| \le \frac{h^2}{12} \underbrace{\left(\left\|\frac{\partial^4 u}{\partial x^4}\right\|_{\infty,\overline{\Omega}} + \left\|\frac{\partial^4 u}{\partial y^4}\right\|_{\infty,\overline{\Omega}}\right)}_{\le 2C} \le C\frac{h^2}{6}.$$

Die zweite zu beweisende Ungleichung folgt aus der folgenden Abschätzung der Normen für $v_h \in l^2(\Omega_h)$

$$\|v_h\|_2 \le \sqrt{m} \max_{\xi \in \Omega_h} |v_h(\xi)| = (n-1)\|v_h\|_\infty \le \tfrac{1}{h}\|v_h\|_\infty. \qquad \square$$

Es bleibt noch die Stabilität des Verfahrens zu zeigen. Das ist in der ∞-Norm etwas aufwendiger als der Nachweis der Konsistenz.

Bemerkung 2.24. Für eine Funktion $g \in L^2(\Omega)$ ist

$$\|g\|_{L^2(\Omega)} = \sqrt{\int_\Omega g(x)^2 \, \mathrm{d}x}$$

die Standard L^2-Norm. Falls wir das Integral per Rechteckregel auf dem Gitter Ω_h approximieren, ergibt sich

$$\|g\|^2_{L^2(\Omega)} \approx \sum_{\xi \in \Omega_h} \int_{Q_h(\xi)} g(x)^2 \, \mathrm{d}x \approx \sum_{\xi \in \Omega_h} h^2 g(\xi)^2 = h^2 \, \|g|_{\Omega_h}\|_2^2 ,$$

wobei $Q_h(\xi) = [\xi_1 - h/2, \xi_1 + h/2] \times [\xi_2 - h/2, \xi_2 + h/2]$ das Quadrat mit Mittelpunkt ξ und Seitenlänge h ist. Diese Diskretisierung der L^2-Norm motiviert die Definition folgender gewichteter Norm

$$\|v_h\|_{2,h} \coloneqq h\|v_h\|_2 , \quad v_h \in l^2(\Omega_h) .$$

Aufgrund der Normäquivalenz ($\|v_h\|_2 = \|y\|_2$ falls $y \in \mathbb{R}^m$ der Koeffizientenvektor von $v_h \in l^2(\Omega_h)$ ist) gilt somit auch

$$\|v_h\|_{2,h} = h\|y\|_2 =: \|y\|_{2,h} \ ,$$

weshalb wir die gewichtete Norm manchmal auch als gewichtete Euklid-Norm bezeichnen. Mit Lemma 2.23 erhalten wir sofort

$$\| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|_{2,h} \leq \frac{1}{6} C h^2 \ ,$$

also Konsistenz in der gewichteten Norm.

In der gewichteten Norm gilt auch die Stabilität des Verfahrens (zur Erinnerung: wir betrachten die kartesische Standard-Diskretisierung des Einheitsquadrats):

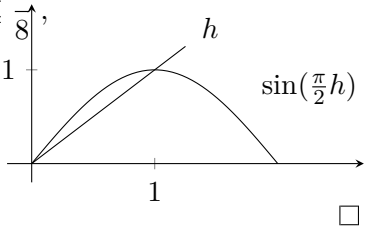
Lemma 2.25 (Stabilität)**.** *Es gilt die (Operatornorm) Abschätzung*

$$\|\Delta_h^{-1}\|_{2,h} = \|\Delta_h^{-1}\|_2 \leq \frac{1}{8} \ .$$

Beweis. Wegen der Normäquivalenz $\|A_1^{-1}\|_2 = \|\Delta_h^{-1}\|_2$ genügt es die Aussagen für die Matrix A_1 zu zeigen. Da A_1 symmetrisch positiv definit ist, gilt

$$\|A_1^{-1}\|_2 = \left(\min_{\lambda \in \sigma(A_1)} \lambda \right)^{-1} \overset{\text{Satz 2.15}}{=} \frac{1}{\lambda_{1,1}} \overset{\text{Satz 2.15}}{=} \frac{h^2}{8 \sin^2(\frac{1}{2}\pi h)} \leq \frac{1}{8},$$

denn $\sin(\frac{1}{2}\pi h) \geq h$ für $h \in [0,1]$.



Ferner gilt für eine beliebige Matrix $A \in \mathbb{R}^{m \times m}$

$$\|A\|_{2,h} = \sup_{x \neq 0} \frac{\|Ax\|_{2,h}}{\|x\|_{2,h}} = \sup_{x \neq 0} \frac{h\,\|Ax\|_2}{h\,\|x\|_2} = \|A\|_2 \ .$$ □

Aus Lemma 2.23 (Konsistenz) und Lemma 2.25 (Stabilität) folgt mit (2.5), d.h.

$$\|e_h\| \leq \|\Delta_h^{-1}\| \| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|$$

sofort die Konvergenz:

Satz 2.26 (Konvergenz)**.** *Sei $e_h := u|_{\overline{\Omega}_h} - u_h$ der Fehler bei der Diskretisierung der Poisson-Gleichung. Falls $u \in C^4(\overline{\Omega})$ und $C := \max\{\|\frac{\partial^4 u}{\partial x^4}\|_{\infty,\overline{\Omega}}, \|\frac{\partial^4 u}{\partial y^4}\|_{\infty,\overline{\Omega}}\}$, konvergiert das Verfahren und es gelten*

(i) $\|e_h\|_{2,h} \leq \frac{1}{48} C h^2$,

(ii) $\|e_h\|_2 \leq \frac{1}{48} C h$.

Bemerkung 2.27.

- Auf endlich-dimensionalen Räumen sind alle Normen äquivalent (Mathe 1). Aber für $h \to 0$ gilt dann $\dim(l^2(\Omega_h)) \to \infty$, so dass der Unterschied zwischen Normen wächst. Somit ist die Konvergenzaussage in der gewichteten Norm schwach im Vergleich zur ∞-Norm.

- Die Voraussetzung $u \in C^4(\overline{\Omega})$ ist eine sehr starke Forderung, die in der Realität oft nicht erfüllt ist. Dazu mehr in Mathe 5 („schwache Lösungen“).

Wir fragen uns nun, ob die Forderung nach Stabilität auch in der ∞-Norm erfüllt ist, also ob $\|\Delta_h^{-1}\|_\infty = \mathcal{O}(1)$? Die Antwort ist „ja“, aber wir benötigen noch etwas Vorarbeit. In der Tat besitzen Matrizen, die bei der Diskretisierung partieller Differentialgleichungen entstehen, oftmals sehr ähnliche Eigenschaften.

Definition 2.28. Eine Matrix $A \in \mathbb{R}^{m\times m}$ mit Komponenten $A = (a_{i,j})_{1\le i,j\le m}$ heißt:

(i) *strikt diagonal dominant*, wenn gilt

$$\sum_{j=1, j\neq i}^{m} |a_{i,j}| < |a_{i,i}| \,, \quad 1 \le i \le m \,,$$

und *schwach diagonal dominant*, wenn das Gleichheitszeichen zugelassen ist,

(ii) *irreduzibel*, wenn keine Permutationsmatrix P existiert, so dass

$$PAP^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} \,, \quad B_{11} \in \mathbb{R}^{k\times k} \,,\ 1 \le k < m \,,$$

(iii) *irreduzibel diagonal dominant*, falls A schwach diagonal dominant ist, in mindestens einer Zeile aber die strikte Ungleichung erfüllt und A irreduzibel ist.

Bemerkung 2.29. Irreduzibilität ist äquivalent zur sog. Ketteneigenschaft: Eine Matrix $A \in \mathbb{R}^{m\times m}$ besitzt die *Ketteneigenschaft*, wenn zu jedem Paar Indizes (i, j), $i, j \in \{1, \dots, m\}$ ein $k \in \mathbb{N}$ sowie $i_1, \dots, i_{k-1} \in \{1, \dots, m\}$ existieren mit $a_{i_{l-1},i_l} \neq 0$ für alle $l \in \{1, \dots, k\}$, wobei $i_0 = i$, $i_k = j$.

Beispiel 2.30. Gegeben seien die Matrizen

$$M_1 = \begin{pmatrix} 1 & 0 & 3 \\ 2 & 1 & 4 \\ 3 & 0 & 2 \end{pmatrix} \,, \quad M_2 = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 4 & 3 \\ 0 & 1 & 2 \end{pmatrix} \,, \quad \text{und} \quad M_3 = \begin{pmatrix} 2 & 1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \,.$$

Dann ist (Übung) die Matrix M_1 reduzibel (also nicht irreduzibel), die Matrix M_2 irreduzibel, aber nicht irreduzibel diagonal dominant, und die Matrix M_3 irreduzibel diagonal dominant.

Definition 2.31. Eine Matrix $A \in \mathbb{R}^{m\times m}$ mit Komponenten $A = (a_{i,j})_{1\le i,j\le m}$ heißt:

(i) *L_0-Matrix*, wenn $a_{i,j} \le 0$ für alle $i \neq j$,

(ii) *L-Matrix*, falls A eine L_0-Matrix ist mit $a_{i,i} > 0$ für alle $1 \le i \le m$,

(iii) *M-Matrix*, falls A eine invertierbare L_0-Matrix ist mit $(A^{-1})_{i,j} \ge 0$ für alle $1 \le i, j \le m$.

Der nächste Satz setzt nun die letzten beiden Matrixdefinitionen in Relation.

Satz 2.32. *Sei $A \in \mathbb{R}^{m\times m}$ eine L-Matrix. Ist A außerdem strikt diagonal dominant oder irreduzibel diagonal dominant, dann ist A eine M-Matrix.*

Der Beweis findet sich z.B. in dem Buch „Iterative Solution of Large Sparse Systems of Equations“ von Wolfgang Hackbusch (2. Auflage, Theorem C.48) oder in dem Buch „Iterative solution of nonlinear equations in several variables“ von Ortega und Rheinboldt.

Bemerkung 2.33. Aus dem obigen Satz folgt, dass A_1 (also die Systemmatrix des diskretisierten Poisson-Problems) eine M-Matrix ist. Die Irreduzibilität von A_1 zeigt man mit Hilfe der Blockstruktur von A_1 und der Bandstruktur der Blöcke (Übung). Die anderen Bedingung sind offensichtlich erfüllt.

Mit diesem Wissen lässt sich nun die Stabilität des Verfahrens zeigen.

Lemma 2.34 (Stabilität)**.** *Es gilt die Abschätzung*

$$\|\Delta_h^{-1}\|_\infty = \|A_1^{-1}\|_\infty \leq \frac{1}{8}.$$

Beweis. Da A_1 eine M-Matrix ist, gilt insbesondere $(A_1^{-1})_{i,j} \geq 0$. Daraus folgt

$$\|A_1^{-1}\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{m} |(A_1^{-1})_{i,j}| = \max_{1\leq i\leq m} \sum_{j=1}^{m} (A_1^{-1})_{i,j} = \max_{1\leq i\leq m} (A_1^{-1}\xi)_i = \|A_1^{-1}\xi\|_\infty \qquad (*)$$

für $\xi = (1, 1, \ldots, 1)^T \in \mathbb{R}^m$. Es bleibt also $\|A_1^{-1}\xi\|_\infty$ abschätzen. Betrachte dazu den kontinuierlichen Fall. Dann ist

$$u\colon [0,1]^2 \to \mathbb{R}, (x,y) \mapsto u(x,y) \coloneqq x(1-x) + y(1-y)$$

Lösung des Poisson-Problems in $\Omega = (0,1)^2$, wenn

$$g \coloneqq u|_{\partial\Omega} \quad \text{und} \quad f \coloneqq 4 \text{ in } \Omega$$

als Randdaten bzw. rechte Seite vorgegeben sind. Für die entsprechende diskrete Lösung u_h gilt dann

$$A_1 u_h = b \geq 4\xi \quad \Rightarrow \quad u_h = A_1^{-1} b \geq 4A_1^{-1}\xi \,,$$

da $g \geq 0$ und $(A_1^{-1})_{i,j} \geq 0$. Damit folgt

$$\frac{1}{4}\|u_h\|_\infty \geq \|A_1^{-1}\xi\|_\infty \overset{(*)}{=} \|A_1^{-1}\|_\infty \;\Rightarrow\; \|A_1^{-1}\|_\infty \leq \frac{1}{4}\|u_h\|_\infty \,.$$

Da die konkrete Lösung u quadratisch ist, also ein Polynom zweiten Grades, folgt

$$\frac{\partial^4 u}{\partial x^4} = \frac{\partial^4 u}{\partial y^4} = 0 \overset{\text{Lemma 2.23}}{\Rightarrow} \| - \Delta_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|_\infty = 0 \,,$$

d.h. $u|_{\overline{\Omega}_h}$ ist eine Lösung des diskreten Problems. Da A_1 als M-Matrix insbesondere invertierbar ist, ist die Lösung des diskreten Problems eindeutig, also folgt $u|_{\overline{\Omega}_h} = u_h$, und somit

$$(u_h)_i = u(\xi_i) \;\Rightarrow\; \|u_h\|_\infty \leq \max_{(x,y)\in\overline{\Omega}} |u(x,y)| = |u(\tfrac{1}{2},\tfrac{1}{2})| = \frac{1}{2}.$$

Insgesamt folgt also

$$\|A_1^{-1}\|_\infty \leq \frac{1}{4}\|u_h\|_\infty \leq \frac{1}{8} \,.$$

□

Insgesamt folgt somit auch die Konvergenz des Verfahrens in der ∞-Norm:

Satz 2.35 (Konvergenz)**.** *Sei $e_h := u|_{\overline{\Omega}_h} - u_h$ der Fehler bei der Diskretisierung der Poisson-Gleichung. Falls $u \in C^4(\overline{\Omega})$ und $C := \max\{\|\frac{\partial^4 u}{\partial x^4}\|_{\infty,\overline{\Omega}}, \|\frac{\partial^4 u}{\partial y^4}\|_{\infty,\overline{\Omega}}\}$, konvergiert das Verfahren und es gilt*

$$\|e_h\|_\infty \leq \tfrac{1}{48} C h^2.$$

Beweis. Folgt mit (2.5) aus Stabilität (Lemma 2.34) und Konsistenz (Lemma 2.23). □

Bemerkung 2.36. Der Beweis für die Stabilität des Verfahrens in der ∞-Norm verwendet eine explizite Lösung des Poisson-Problems. Das schränkt die Gültigkeit jedoch nicht ein, d.h. die Aussage gilt unabhängig von f und g, benutzt allerdings explizit $\Omega = (0,1)^2$.

2.4 Konvektions-Diffusions-Gleichung

Wir erweitern nun die Klasse der partiellen Differentialgleichungen, für die wir numerische Verfahren mittels finiter Differenzen betrachten.

Problem 2.37 (Konvektions-Diffusions-Problem)**.** Sei $\Omega := (0,1)^d \subset \mathbb{R}^d$. Zu einem gegebenem Quellterm $f\colon \Omega \to \mathbb{R}$ und Randwerten $g\colon \partial\Omega \to \mathbb{R}$, sowie zu gegebenem $\varepsilon > 0$ und $v \in \mathbb{R}^d$, finde $u \in C^2(\Omega) \cap C(\overline{\Omega})$, so dass

$$\begin{aligned} -\varepsilon \Delta u + v \cdot \nabla u &= f \text{ in } \Omega\,, \\ u &= g \text{ auf } \partial\Omega\,. \end{aligned}$$

Bemerkung 2.38. Das Konvektions-Diffusions-Problem (Problem 2.37) modelliert zwei kombinierte Arten, wie sich eine Substanz ausbreiten kann:

- **Diffusion**: $-\varepsilon\Delta u$,
- **Konvektion** in Richtung $v \in \mathbb{R}^d$: $v \cdot \nabla u = \sum_{j=1}^d v_j \partial_{x_j} u$ (vgl. dazu die Transportgleichung $\partial_t u + v \cdot \nabla u = 0$ aus dem Theorie Teil).

Die Schwierigkeit an dieser Gleichung ist, dass sich die zwei Teile unterschiedlich verhalten, und unterschiedliche numerische Techniken benötigen. Eigentlich ist die Gleichung elliptisch und zweiter Ordnung. Aber häufig ist ε sehr klein, so dass man fast eine Gleichung erster Ordnung hat (*Konvektionsdominanz*).

Bemerkung 2.39. Wir betrachten zunächst den Spezialfall $d = 1$, also $\Omega = (0,1)$, $f \equiv 1$ und $g \equiv 0$: Finde u, so dass

$$\begin{aligned} -\varepsilon u'' + v u' &= 1 \text{ in } \Omega\,, \\ u &= 0 \text{ auf } \partial\Omega\,. \end{aligned}$$

Zuerst untersuchen wir zwei Grenzfälle separat:

(i) Grenzfall $v = 0$:

$$\begin{aligned} -\varepsilon u'' &= 1 \text{ in } \Omega, \\ u &= 0 \text{ auf } \partial\Omega. \end{aligned}$$

Aus der Differentialgleichung ergibt sich sofort

$$u(x) = -\tfrac{1}{2\varepsilon}x^2 + c_1 x + c_2$$

mit $c_1, c_2 \in \mathbb{R}$. Weiter folgt aus den Randwerten: $0 = u(0) = c_2$ und $0 = u(1) = -\frac{1}{2\varepsilon} + c_1$, d.h. $c_1 = \frac{1}{2\varepsilon}$. Die Lösung ist also eine „Delle“, deren Höhe von ε abhängt.

(ii) Grenzfall $\varepsilon = 0$:

$$\begin{aligned} v u' &= 1 \text{ in } \Omega \,, \\ u &= 0 \text{ auf } \partial\Omega \,. \end{aligned}$$

Hier folgt aus der Differentialgleichung sofort

$$u(x) = \tfrac{1}{v}x + c$$

mit $c \in \mathbb{R}$. Aus den Randwerten folgt wiederum $0 = u(0) = c$ und $0 = u(1) = \frac{1}{v} + c$. Es können offensichtlich nicht beide Randwerte erfüllt sein, d.h. es existiert keine Lösung.

Ersetzt man die Randbedingung durch die „Anfangsbedingung“ $u(0) = 0$, so ist das Problem wohlgestellt. Die Information fließt von links nach rechts (d.h. entlang der Charakteristiken). Statt den linken Randwert als Anfangsbedingung zu nutzen, könnte man auch den rechten Randwert benutzen. Was ist richtig? Hierzu müssen wir das Verhalten der Lösung des allgemeinen Falls im Grenzübergang $\varepsilon \to 0$ untersuchen.

Nun betrachten wir den allgemeinen Fall $v \neq 0$ und $\varepsilon \neq 0$. Die Gleichung

$$-\varepsilon u'' + v u' = 1$$

ist eine lineare ODE 2. Ordnung. Wir nutzen folgenden Ansatz um eine Lösung zu bestimmen:

$$u(x) = u_{\text{hom}}(x) + u_{\text{inh}}(x) \,.$$

Für $v \neq 0$ ist $u_{\text{inh}}(x) = \frac{1}{v}x$ eine Lösung der inhomogenen Gleichung. Für die homogene Gleichung machen wir den Ansatz

$$u_{\text{hom}}(x) = c_1 + c_2 e^{\lambda x} \,,$$

wobei $c_1, c_2, \lambda \in \mathbb{R}$. Aus der ODE folgt die Bedingung $-\varepsilon c_2 \lambda^2 e^{\lambda x} + v c_2 \lambda e^{\lambda x} = 0$ für alle $x \in \Omega$, also

$$(-\varepsilon\lambda + v)\lambda c_2 = 0 \quad \Rightarrow \quad c_2 = 0 \ \vee \ \lambda = 0 \ \vee \ \lambda = \tfrac{v}{\varepsilon} \,.$$

Damit ist

$$u_{\text{hom}}(x) = c_1 + c_2 e^{\frac{v}{\varepsilon}x}$$

für alle $c_1, c_2 \in \mathbb{R}$ eine Lösung der homogenen Gleichung und wir erhalten insgesamt

$$u(x) = c_1 + c_2 e^{\frac{v}{\varepsilon}x} + \frac{1}{v}x \,.$$

Durch dir Randwerte erhalten wir

$$u(0) = 0 \quad \Rightarrow \quad c_1 + c_2 = 0 \quad \text{und}$$
$$u(1) = 0 \quad \Rightarrow \quad c_1 + c_2 e^{\frac{v}{\varepsilon}} + \frac{1}{v} = 0 \, .$$

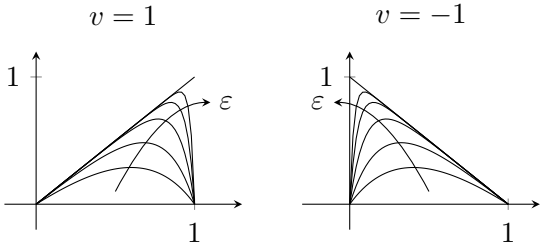
Somit gilt $c_2 = -c_1$ und es folgt weiter

$$c_1(1 - e^{\frac{v}{\varepsilon}}) = -\frac{1}{v} \quad \Rightarrow \quad c_1 = -\frac{1}{v(1 - e^{\frac{v}{\varepsilon}})} \quad \Rightarrow \quad c_2 = \frac{1}{v(1 - e^{\frac{v}{\varepsilon}})} \, .$$

Insgesamt erhalten wir also

$$u(x) = \frac{1}{v}\left(x - \frac{1 - e^{\frac{v}{\varepsilon}x}}{1 - e^{\frac{v}{\varepsilon}}}\right)$$

als Lösung. Die Lösung ist eine „Delle“, die zum Rand $x = 1$ ($v > 0$), bzw. $x = 0$ ($v < 0$) hin aufgeschoben ist. Je kleiner ε, desto höher schiebt sich die Lösung zum Rand hin auf.



Für $v > 0$ und $x \in (0, 1)$ folgt mit der Regel von de L'Hospital

$$\lim_{\varepsilon \to 0} \frac{1 - e^{\frac{vx}{\varepsilon}}}{1 - e^{\frac{v}{\varepsilon}}} = \lim_{\varepsilon \to 0} \frac{-e^{\frac{vx}{\varepsilon}}(-\frac{vx}{\varepsilon^2})}{-e^{\frac{v}{\varepsilon}}(-\frac{v}{\varepsilon^2})} = \lim_{\varepsilon \to 0} \frac{e^{\frac{vx}{\varepsilon}}x}{e^{\frac{v}{\varepsilon}}} = \lim_{\varepsilon \to 0} x e^{\frac{v(x-1)}{\varepsilon}} = 0 \, ,$$

da $v(x - 1) < 0$. Für $v < 0$ und $x \in (0, 1)$ folgt direkt

$$\lim_{\varepsilon \to 0} \frac{1 - e^{\frac{vx}{\varepsilon}}}{1 - e^{\frac{v}{\varepsilon}}} = 1 \, ,$$

da $v < 0$ und $vx < 0$. Für die Lösung u ergibt sich also im Grenzfall

$$u(x) = \begin{cases} \frac{1}{v}x & v > 0 \\ \frac{1}{v}(x - 1) & v < 0 \, . \end{cases}$$

Im Fall $v > 0$ ist $u(0) = 0$ erfüllt, im Fall $v < 0$ ist $u(1) = 0$ erfüllt. Das Vorzeichen von v spielt also ein wichtige Rolle!

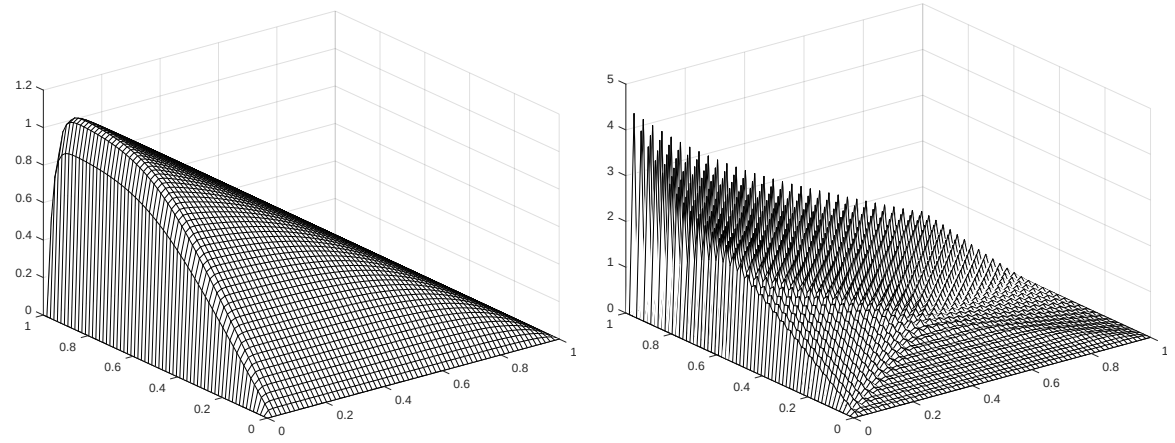
Bemerkung 2.40. Im Fall $d = 2$ und somit $\Omega = (0, 1)^2$ ist die Transportrichtung durch den Vektor $v \in \mathbb{R}^2$ gegeben. Die Information wird entlang von Charakteristiken transportiert. Dabei ist der *Einströmrand* des Gebietes die Menge $\{x \in \partial\Omega : v \cdot n(x) < 0\}$, wobei $n(x) \in \mathbb{R}^d$ die (nach außen zeigende) Normale (oder: Normalenvektor) auf dem Gebietsrand $x \in \partial\Omega$ bezeichnet.

Für die Diskretisierung verwenden wir, genau wie bei der Poisson-Gleichung, die kartesischen Gitter Ω_h und $\overline{\Omega_h}$ aus Bemerkung 2.7. Dabei wird der diffusive Teil der Differentialgleichung, also der Laplace Operator Δ, wie zuvor durch den Differenzenstern (2.2) Δ_h diskretisiert.

Aus Abschnitt 2.1 kennen wir nun drei naheliegende Möglichkeiten die partiellen Ableitungen in dem konvektiven Teil $v \cdot \nabla u$ zu diskretisieren. Für die partielle Ableitung $\frac{\partial u}{\partial x}$ haben wir zum Beispiel:

(i) Zentrale Differenzen:

$$\frac{\partial u}{\partial x}(x, y) = \frac{1}{2h}[-u(x - h, y) + u(x + h, y)] + \mathcal{O}(h^2) \, ,$$



$\varepsilon = 10^{-2}$ $\varepsilon = 10^{-4}$

Abbildung 2.5: Lösungen des Konvektions-Diffusions-Problems ($f \equiv 1$, $g \equiv 0$ und $v = (\cos\beta, \sin\beta)^T$ für $\beta = 5\pi/6$) mit zentralen Differenzen ($n = 2^6$) für verschiedenen Werte von ε.

(ii) Linksseitiger Differenzenquotient:

$$\frac{\partial u}{\partial x}(x,y) = \frac{1}{h}[-u(x-h,y) + u(x,y)] + \mathcal{O}(h) \,,$$

(iii) Rechtsseitiger Differenzenquotient:

$$\frac{\partial u}{\partial x}(x,y) = \frac{1}{h}[-u(x,y) + u(x+h,y)] + \mathcal{O}(h) \,.$$

Die entsprechenden Differenzensterne sind:

$$\frac{1}{2h}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}_\xi \,, \quad \frac{1}{h}\begin{bmatrix}-1 & 1 & 0\end{bmatrix}_\xi \,, \quad \text{bzw.} \quad \frac{1}{h}\begin{bmatrix}0 & -1 & 1\end{bmatrix}_\xi$$

jeweils für $\xi \in \Omega_h$.

Zentrale Differenzen machen den kleinsten Fehler, aber sie führen zu Instabilitäten. Abbildung 2.5 zeigt die numerische Lösung einer Konvektions-Diffusions-Gleichung, die mit zentralen Differenzen für den konvektiven Teil gelöst wurde. Für kleines ε sind deutlich die Instabilitäten zu erkennen.

Als Ausweg für die Instabilitäten verwendet man einseitige Differenzen für die ersten Ableitungen im konvektiven Teil des Operators. Die Richtung des Differenzenquotienten richtet sich dabei nach der Richtung der Informationsausbreitung bzgl. des Einströmrandes und Konvektionsrichtung v. Stromaufwärts heißt Upwind-Richtung, weshalb diese einseitige Diskretisierung oft als *Upwind–Verfahren* bezeichnet wird.

In dem betrachteten Fall geht die Strömung „von links nach rechts“ (in der x-Komponente) wenn $v_1 > 0$, so dass die Diskretisierung in die entgegengesetzte Richtung erfolgen sollte. Wir wählen also diesem Fall den Differenzenstern

$$\begin{cases} \frac{1}{h}\begin{bmatrix}-1 & 1 & 0\end{bmatrix}_\xi & \text{falls } v_1 > 0 \,, \\ \frac{1}{h}\begin{bmatrix}0 & -1 & 1\end{bmatrix}_\xi & \text{falls } v_1 \leq 0 \,, \end{cases}$$

welcher das Vorzeichen von v_1 berücksichtigt. Dies lässt sich deutlich knapper schreiben. Dazu definieren wir

$$v_1^+ := \max(0, v_1) \quad \text{und} \quad v_1^- := \min(0, v_1) \; .$$

Dann ist die Upwind-Differenz von $v_1 \frac{\partial u}{\partial x}$ der Differenzenstern

$$[D_x^{\text{up}}]_\xi := \frac{1}{h} \begin{bmatrix} -v_1^+ & |v_1| & v_1^- \end{bmatrix}_\xi \; , \quad \xi \in \Omega_h \; .$$

Analog ist die Upwind-Differenz von $v_2 \frac{\partial u}{\partial y}$ der Differenzenstern

$$[D_y^{\text{up}}]_\xi := \frac{1}{h} \begin{bmatrix} v_2{}^- \\ |v_2| \\ -v_2^+ \end{bmatrix}_\xi \; , \quad \xi \in \Omega_h \; .$$

Achtung: Die Upwind-Differenzensterne hängen von v ab.

Mit Hilfe der Notation des Upwind-Differenzensterns ergibt sich:

Problem 2.41 (Diskretisiertes Konvektions-Diffusions-Problem)**.** Seien Ω_h und $\overline{\Omega_h}$ die kartesischen Gitter des Einheitsquadrats aus Bemerkung 2.7. Finde $u_h \in l^2(\overline{\Omega}_h)$, so dass

$$\begin{aligned} [(-\varepsilon \Delta_h + D_x^{\text{up}} + D_y^{\text{up}}) u_h](\xi) &= f(\xi) \text{ für } \xi \in \Omega_h \; , \\ u_h(\xi) &= g(\xi) \text{ für } \xi \in \overline{\Omega}_h \setminus \Omega_h \; . \end{aligned}$$

Bemerkung 2.42. Der vollständige Differenzenstern von Problem 2.41 ist somit

$$\begin{aligned} [-\varepsilon \Delta_h + D_x^{\text{up}} + D_y^{\text{up}}]_\xi &= \frac{1}{h^2} \begin{bmatrix} & -\varepsilon + h v_2^- & \\ -\varepsilon - h v_1^+ & 4\varepsilon + h(|v_1| + |v_2|) & -\varepsilon + h v_1^- \\ & -\varepsilon - h v_2^+ & \end{bmatrix}_\xi \\ &=: \begin{bmatrix} & -\tilde{n} & \\ -\tilde{w} & +\tilde{z} & -\tilde{o} \\ & -\tilde{s} & \end{bmatrix}_\xi \end{aligned}$$

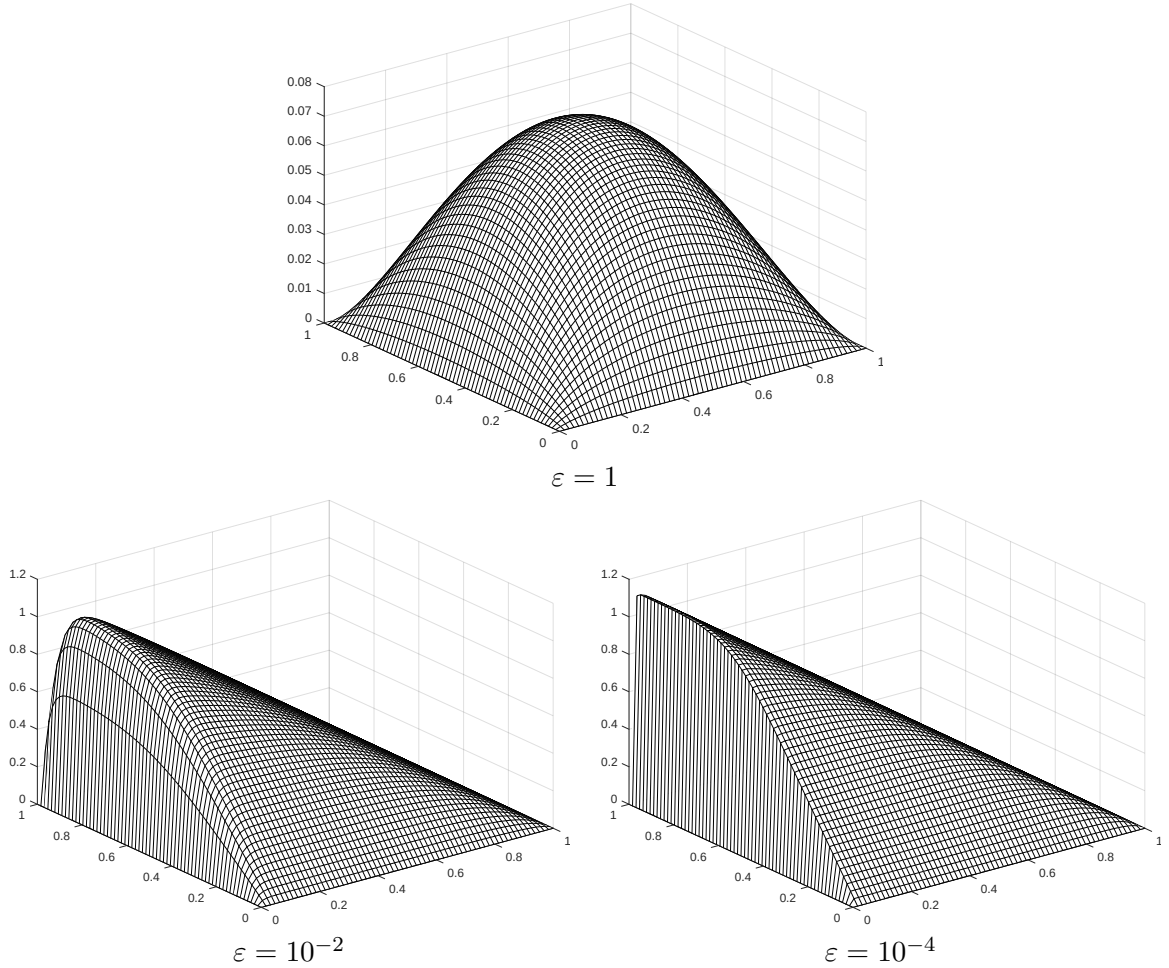
Wie bei der Poisson–Gleichung erhält man durch die Diskretisierung ein lineares Gleichungssystem

$$A_2 x = b \; , \quad A_2 \in \mathbb{R}^{(n-1)^2 \times (n-1)^2} \; .$$

In der Standardanordnung hat A_2 die folgende Bandstruktur:

$$A_2 = h^{-2} \begin{pmatrix} T & -\tilde{n}I & & & 0 \\ -\tilde{s}I & T & -\tilde{n}I & & \\ & \ddots & \ddots & \ddots & \\ & & -\tilde{s}I & T & -\tilde{n}I \\ 0 & & & -\tilde{s}I & T \end{pmatrix} \; , \quad T = \begin{pmatrix} \tilde{z} & -\tilde{o} & & & 0 \\ -\tilde{w} & \tilde{z} & -\tilde{o} & & \\ & \ddots & \ddots & \ddots & \\ & & -\tilde{w} & \tilde{z} & -\tilde{o} \\ 0 & & & -\tilde{w} & \tilde{z} \end{pmatrix} \; .$$

Abbildung 2.6 zeigt die Lösung des diskreten Konvektions-Diffusions-Problems mit Upwind-Differenzen für verschiedene Werte von ε.



$\varepsilon = 1$

$\varepsilon = 10^{-2}$

$\varepsilon = 10^{-4}$

Abbildung 2.6: Lösung des Konvektions-Diffusions-Problems ($f \equiv 1$, $g \equiv 0$ und $v = (\cos\beta, \sin\beta)^T$ für $\beta = 5\pi/6$) mit Upwind-Differenzen ($n = 2^6$) für verschiedenen Werte von ε.

Bei der Fehleranalyse für das Konvektions-Diffusions-Problem gehen wir analog zu der Fehleranalyse des Poisson-Problems vor. Sei $u \in C^2(\Omega) \cap C(\overline{\Omega})$ die Lösung der Konvektions-Diffusions-Gleichung und $u_h \in l^2(\overline{\Omega}_h)$ die Lösung des diskreten Problems. Wir wollen den Diskretisierungsfehler $e_h \coloneqq u|_{\overline{\Omega}_h} - u_h$ abschätzen.

Wir führen die kürzere Notation

$$\Xi_h \coloneqq -\varepsilon\Delta_h + D_x^{\mathrm{up}} + D_y^{\mathrm{up}} \colon l^2(\overline{\Omega}_h) \to l^2(\Omega_h)$$

für den diskretisierten Differentialoperator ein. Ferner erinnern wir uns, dass gilt (wie Δ_h ist auch Ξ_h invertierbar auf $l_0^2(\overline{\Omega}_h)$; Übung):

$$\|e_h\| = \|\Xi_h^{-1}\Xi_h e_h\| = \|\Xi_h^{-1}(\Xi_h u|_{\overline{\Omega}_h} - f|_{\Omega_h})\| \leq \underbrace{\|\Xi_h^{-1}\|}_{\text{Stabilität}} \underbrace{\|\Xi_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|}_{\text{Konsistenzfehler}} \tag{2.6}$$

für entsprechende Normen. Hier betrachten wir direkt die Konvergenz in der ∞-Norm. Wir schätzen zunächst den Konsistenzfehler ab.

Lemma 2.43. *Sei $u \in C^4(\overline{\Omega})$ die Lösung des Diffusions-Konvektions-Problems 2.37. Ferner seien $C \coloneqq \max\{\|\frac{\partial^4 u}{\partial x^4}\|_{\infty,\overline{\Omega}}, \|\frac{\partial^4 u}{\partial y^4}\|_{\infty,\overline{\Omega}}\}$ und $D \coloneqq \max\{\|\frac{\partial^2 u}{\partial x^2}\|_{\infty,\overline{\Omega}}, \|\frac{\partial^2 u}{\partial y^2}\|_{\infty,\overline{\Omega}}\}$. Dann gilt*

$$\|\Xi_h u|_{\overline{\Omega}_h} - f|_{\Omega_h}\|_\infty \leq \frac{1}{6}\varepsilon C h^2 + \frac{\sqrt{2}}{2} D \, \|v\|_2 \, h \, .$$

Beweis. Sei $\omega \in C^2(\mathbb{R})$ sowie $x \in \mathbb{R}$ und $h > 0$. Nach Taylor existiert ein $\eta \in (x, x+h)$, so dass

$$\omega(x+h) = \omega(x) + h\omega'(x) + \tfrac{1}{2}h^2\omega''(\eta) \quad \Rightarrow \quad \frac{\omega(x+h) - \omega(x)}{h} = \omega'(x) + \tfrac{h}{2}\omega''(\eta) \, . \tag{$*$}$$

Analog existiert ein $\tilde{\eta} \in (x-h, x)$ mit

$$\omega(x-h) = \omega(x) - h\omega'(x) + \tfrac{1}{2}h^2\omega''(\tilde{\eta}) \quad \Rightarrow \quad \frac{\omega(x) - \omega(x-h)}{h} = \omega'(x) - \tfrac{h}{2}\omega''(\tilde{\eta}) \, . \tag{$\diamond$}$$

Wegen $-v_1^- + v_1^+ = |v_1|$ und $v_1^- + v_1^+ = v_1$ folgt aus „$v_1^-(*) + v_1^+(\diamond)$“, dass

$$\tfrac{1}{h}\big(v_1^-\omega(x+h) + |v_1|\,\omega(x) - v_1^+\omega(x-h)\big) = v_1\omega'(x) + \tfrac{h}{2}\big(v_1^-\omega''(\eta) - v_1^+\omega''(\tilde{\eta})\big) \, .$$

Da entweder $v_1^- = 0$ oder $v_1^+ = 0$, folgt aus obigem, dass es für jedes $\xi = (x, y) \in \Omega_h$ ein $\eta \in (x-h, x+h)$ gibt mit

$$(D_x^{\mathrm{up}}u)(\xi) = v_1 \frac{\partial u}{\partial x}(\xi) - |v_1| \, \frac{h}{2}\frac{\partial^2 u}{\partial x^2}(\eta, y) \, .$$

Dies gilt analog für $D_y^{\mathrm{up}}u$. Im Beweis von Lemma 2.23 haben wir bereits gezeigt, dass $\eta \in (x-h, x+h)$ und $\tilde{\eta} \in (y-h, y+h)$ existieren mit

$$(-\Delta_h u)(\xi) = -\Delta u(\xi) - \frac{h^2}{12}\left(\frac{\partial^4 u}{\partial x^4}(\eta, y) + \frac{\partial^4 u}{\partial y^4}(x, \tilde{\eta})\right) \, .$$

Daraus wieder folgt nun, dass für jedes $\xi \in \Omega_h$ gilt:

$$\begin{aligned}\big|(\Xi_h u|_{\overline{\Omega}_h})(\xi) - f(\xi)\big| &= \big|-\varepsilon\Delta_h u|_{\overline{\Omega}_h}(\xi) + (D_x^{\mathrm{up}} + D_y^{\mathrm{up}})u|_{\overline{\Omega}_h}(\xi) - f(\xi)\big| \\ &\leq \frac{1}{6}\varepsilon C h^2 + |v_1|\frac{h}{2}D + |v_2|\frac{h}{2}D \leq \frac{1}{6}\varepsilon C h^2 + \frac{\sqrt{2}}{2} D \, \|v\|_2 \, h \, .\end{aligned}$$

Hier haben wir verwendet, dass $\|y\|_1 \leq \sqrt{d}\,\|y\|_2$ für $y \in \mathbb{R}^d$. □

Bemerkung 2.44. Analog zur Upwind-Differenz, kann man auch die Downwind-Differenz konstruieren. Mit dieser erhält man die gleiche Konsistenzaussage wie oben. Nutzt man den zentralen Differenzenquotienten zur Diskretisierung von $v \cdot \nabla u$, so erhält man analog zu obigem Konsistenz mit $\mathcal{O}(h^2)$, also eine Ordnung besser (Übung).

Als nächstes zeigen wir die Stabilität.

Lemma 2.45 (Stabilität)**.** *Unabhängig von h und ε gilt die Abschätzung:*

$$\|\Xi_h^{-1}\|_\infty = \|A_2^{-1}\|_\infty \leq \frac{\sqrt{2}}{\|v\|_2} .$$

Beweis. A_2 ist eine irreduzibel diagonaldominante L-Matrix ist (Übung), also auch eine M-Matrix. Es gilt also insbesondere $(A_2^{-1})_{i,j} \geq 0$. Wie im Beweis von Lemma 2.34 folgt auch hier, dass

$$\|A_2^{-1}\|_\infty = \|A_2^{-1}\xi\|_\infty \quad \text{mit} \quad \xi = (1, 1, \ldots, 1)^T \in \mathbb{R}^m \ , \ m = (n-1)^2 \ .$$

Ebenso wie im Beweis von Lemma 2.34 hängt die Matrix auch hier nicht von f und g ab. Wir können daher wieder eine Lösung für eine spezielle Wahl von f und g benutzen. Allerdings müssen wir eine Fallunterscheidung über alle Vorzeichenkombinationen von v_1 und v_2 machen. Wir behandeln hier exemplarisch den Fall $v_1, v_2 \geq 0$. In diesem Fall betrachten wir

$$f \equiv \|v\|_2^2 \quad \text{und} \quad g(x, y) \coloneqq v_1 x + v_2 y \ ,$$

denn dann ist $u(x, y) = v_1 x + v_2 y$ eine Lösung des Konvektions-Diffusions-Problems in $\Omega = (0, 1)^2$. Für die diskrete Lösung u_h gilt dann, da $g \geq 0$ und $(A_2^{-1})_{i,j} \geq 0$, dass

$$A_2 u_h = b \geq \|v\|_2^2 \, \xi \ \Rightarrow \ u_h \geq \|v\|_2^2 \, A_2^{-1}\xi \geq 0 \ \Rightarrow \ \|A_2^{-1}\xi\|_\infty \leq \frac{1}{\|v\|_2^2}\|u_h\|_\infty \ .$$

Die Lösung u ist linear und somit verschwinden insbesondere die zweiten und vierten partiellen Ableitungen von u. Damit ist wegen Lemma 2.43 der Konsistenzfehler gleich 0, d.h. $u|_{\overline{\Omega}_h}$ ist eine Lösung des diskreten Problems. Da A_2 als M-Matrix insbesondere invertierbar ist, ist die Lösung des diskreten Problems eindeutig, also folgt $u_h(\xi) = u(\xi)$ für alle $\xi \in \Omega_h$. Damit folgt

$$\|A_2^{-1}\|_\infty = \|A_2^{-1}\xi\|_\infty \leq \frac{1}{\|v\|_2^2}\|u_h\|_\infty \leq \frac{1}{\|v\|_2^2} \sup_{(x,y)\in\Omega} |u(x, y)| \leq \frac{1}{\|v\|_2^2}(v_1 + v_2) \leq \frac{\sqrt{2}}{\|v\|_2} \ .$$

Für andere Vorzeichen von v_1, v_2 gilt Herleitung analog. Es muss nur g (und damit u) so angepasst werden, dass $g \geq 0$ gilt. □

Wir fassen die Ergebnisse zur Stabilität und zur Konsistenz in einem Satz zusammen.

Satz 2.46 (Konvergenz)**.** *Seien $u \in C^4(\overline{\Omega})$ und C, D wie in Lemma 2.43. Dann gilt*

$$\|e_h\|_\infty \leq \frac{\sqrt{2}\varepsilon}{6\,\|v\|_2} C h^2 + D h.$$

Beweis. Folgt mit (2.6) sofort aus Lemma 2.43 und Lemma 2.45. □

Bemerkung 2.47. Für den Beweis von Lemma 2.45 ist essentiell, dass A_2 eine M-Matrix ist, was sich daraus ergibt, dass A_2 eine irreduzibel diagonaldominante L-Matrix ist. Letzteres wird durch die Verwendung der Upwind-Differenzen unabhängig von h und ε sicher gestellt.

Bei Verwendung der zentralen Differenzen ist der Differenzenstern

$$\frac{1}{h^2}\begin{bmatrix} & -\varepsilon + \frac{h}{2}v_2 & \\ -\varepsilon - \frac{h}{2}v_1 & 4\varepsilon & -\varepsilon + \frac{h}{2}v_1 \\ & -\varepsilon - \frac{h}{2}v_2 & \end{bmatrix}.$$

Hieraus ergeben sich zwei Grenzfälle für die Stabilität des resultierenden Gleichungssystems $\tilde{A}_2 u_h = \tilde{b}$:

(i) $h \leq \frac{2\varepsilon}{\|v\|_\infty}$: $\tilde{A}_2$ ist eine irreduzibel diagonaldominante L-Matrix mit $\|\tilde{A}_2^{-1}\|_\infty \leq C$.

(ii) $h > \frac{2\varepsilon}{\|v\|_\infty}$: $\|\tilde{A}_2^{-1}\|_\infty \leq C(\varepsilon)$ mit $\lim_{\varepsilon \to 0} C(\varepsilon) = \infty$. Dieses Verhalten bezeichnet man als *instabil*.

2.5 Höhere Ordnung, Randwerte und selbstadjungierte Probleme

In diesem Abschnitt diskutieren wir einige weiterführende Themen der FD-Diskretisierung, die Meisten werden wir beispielhaft am eindimensionalen Randwertproblem

$$-u'' = f \quad \text{für} \quad x \in (0,1)\,, \quad u(0) = u(1) = 0\,.$$

skizzieren.

Bemerkung 2.48 (Höhere Ordnung). Moderne FD-Verfahren besitzen eine Konvergenzordnung $\mathcal{O}(h^p)$ mit $p = 2, 3, 4, 5, \ldots$. Es gibt verschiedene Arten solche Verfahren zu konstruieren.

(i) *Finite Differenzen mit höherer Ordnung:* Verwendet man z.B. eine zentrale Differenz mit fünf Punkten in der Konstruktion aus Beispiel 2.3 (i), so erhält man

$$\frac{-u(x-2h) + 16u(x-h) - 30u(x) + 16u(x+h) - u(x+2h)}{12h^2} = u''(x) + \mathcal{O}(h^4)\,.$$

Es ist klar, dass sich diese Prozedur beliebig fortsetzen lässt. Es treten dabei zwei Probleme auf: Zum einen stellt sich die Frage nach der Stabilität. Im obigen Beispiel z.B. erhalten wir keine L-Matrix mehr. Weiterhin wird die Systemmatrix immer voller besetzt, und man hat zudem Probleme, Randwerte zu definieren (im obigen Beispiel benötigt man an jedem Randpunkt zwei Randwerte, von denen aber nur einer gegeben ist).

(ii) *Richardson-Extrapolation:* Die Idee der Extrapolation haben wir schon bei der Romberg-Integration (siehe Mathe 2) kennen gelernt. Hierbei kombiniert man geschickt die Approximationen zu den Schrittweiten h und $h/2$ um den dominanten Fehlerterm zu eliminieren. Die Methode basiert auf der Entwicklung

$$u_h(\xi) - u(\xi) = C_2 h^2 + C_4 h^4 + \ldots\,,$$

wobei u_h die numerische Lösung zur Schrittweite h und u die exakte Lösung bezeichnen. Dann gilt

$$u_{h/2}(\xi) - u(\xi) = \frac{1}{4} C_2 h^2 + \frac{1}{16} C_4 h^4 + \ldots\,.$$

Somit gilt für den extrapolierten Wert in dem Gitterpunkt $\xi \in \Omega_h$:

$$u_{\text{extrap}}(\xi) \coloneqq \frac{1}{3}\big(4u_{h/2}(\xi) - u_h(\xi)\big) = u(\xi) + \mathcal{O}(h^4) \ .$$

Nachteil dieser Methode ist, dass man das Problem zweimal lösen muss, zudem noch auf einem feineren Gitter, d.h. 2^d mal mehr Unbekannte. Dies macht die Methode sehr teuer.

(iii) Die am Meisten verfolgte Idee ist die der sogenannten *kompakten Verfahren* bzw. der sogenannten „Deferred Corrections“, die ähnlich zur Idee der Runge-Kutta-Verfahren (siehe Mathe 3) ist.

Analog zum Vorgehen im Beweis von Lemma 2.23, zeigt man für $u \in C^6(\mathbb{R})$ mit Taylor, dass

$$-\frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = -u''(x) - \frac{1}{12}h^2 u^{(4)}(x) + \mathcal{O}(h^4) \ .$$

Falls u die Differentialgleichung $-u'' = f$ erfüllt und $f \in C^2(\mathbb{R})$, so folgt $-u^{(4)} = f''$ und

$$-\frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = f(x) + \frac{1}{12}h^2 f''(x) + \mathcal{O}(h^4) \ ,$$

Nun erhält man eine Approximation 4. Ordnung indem man die rechte Seite der diskretisierten Differentialgleichung durch die Werte

$$f(\xi) + \frac{1}{12}h^2 f''(\xi)$$

an allen Gitterpunkten $\xi \in \Omega_h$ ersetzt. Falls die zweite Ableitung von f nicht bekannt ist, so genügt auch ein Differenzenquotient, denn

$$f(x) + \frac{1}{12}h^2 f''(x) = f(x) + \frac{1}{12}\big(f(x-h) - 2f(x) + f(x+h)\big) + \mathcal{O}(h^4) \ .$$

Dies nennt man „Deferred Correction“, weil man die rechte Seite modifiziert, um einen numerischen Fehler aufzuheben. Das Verfahren besitzt einen 3-Punkt-Stern und wird daher als kompakt bezeichnet.

Bemerkung 2.49 (Randbedingungen). Falls das Gebiet der Poisson-Gleichung keine einfache Form mehr hat, z.B. krummlinig berandet ist, liegen die Gitterpunkte nicht notwendigerweise auf dem Rand (siehe Abbildung 2.7). Um damit umzugehen, gibt es verschiedene Möglichkeiten.

(i) *nicht-äquidistante Gitter*: Man startet mit einem äquidistanten Gitter und projiziert die Gitterpunkte, die direkt außerhalb des Gebietes liegen, in Richtung aller Koordinatenachsen auf den Rand. Dabei ist zu beachten, dass ein Differenzenquotient der Form

$$u''(x) \approx \frac{1}{(h_r + h_l)/2}\left(\underbrace{\frac{u(x+h_r) - u(x)}{h_r}}_{\approx u'(x+\frac{h_r}{2})} - \underbrace{\frac{u(x) - u(x-h_l)}{h_l}}_{\approx u'(x-\frac{h_l}{2})}\right)$$

nur von erster Ordnung in $h = \max(h_l, h_r)$ ist, falls $h_r \neq h_l$. Dies ist aber nur für die Punkte in Randnähe von Belang. Für $h_r = h_l$ ist obiges genau der zentrale Differenzenquotient 2. Ordnung.

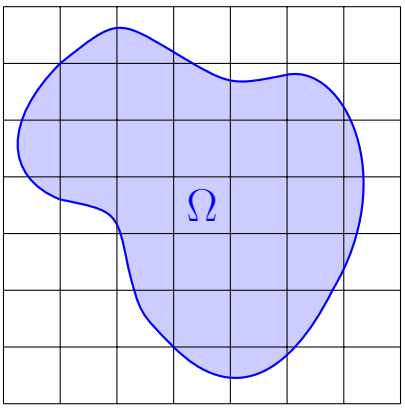
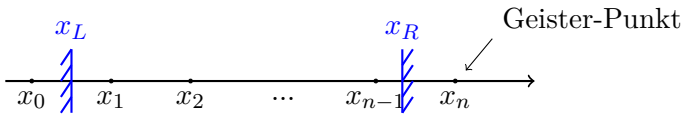


Abbildung 2.7: Gebiet mit krummlinigem Rand auf einem regulären Gitter.

(ii) *„Geister-Punkte“*: Eine andere Möglichkeit bieten Extrapolationstechniken. Man behält die Position eines Gitterpunktes außerhalb des Gebiets und weist diesem mittels Extrapolation einen Wert zu, den man im linearen Gleichungssystem benutzt. In einer Dimension betrachten wir Gitterpunkte *zusammen* mit den Randpunkten x_L und x_R, also $\Omega = (x_L, x_R)$, so dass

$$x_0 < x_L < x_1 < x_2 < \cdots < x_{n-1} < x_R < x_n \; .$$

Dabei seien die Gitterpunkte x_i äquidistant mit Schrittweite h. Ferner seien $\alpha_L, \alpha_R \in (0,1)$,



so dass

$$x_L - x_0 = \alpha_L h \quad \text{und} \quad x_n - x_R = \alpha_R h \; .$$

In jedem inneren Gitterpunkt x_i benutzen wir den üblichen zentralen Differenzenquotienten

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = \frac{1}{h}\left(\frac{u_{i+1} - u_i}{h} - \frac{u_i - u_{i-1}}{h}\right) \; , \quad 1 \le i \le n-1 \; ,$$

wobei $u_j = u(x_j)$. Die Approximation nahe der Ränder, nämlich in den beiden Randpunkten x_L und x_R, ist dabei problematisch, da sie Werte an „Geister-Punkten“ (d.h. Gitterpunkte außerhalb von Ω) benötigen. Dies umgehen wir durch Extrapolation. Am linken Rand bestimmen wir dazu das Interpolationspolynom ersten Grades durch die Tupel (x_L, u_L) und (x_1, u_1). Dieses Polynom ist genau die Gerade:

$$p(x) = u_1 + \frac{x - x_1}{x_1 - x_L}(u_1 - u_L) \; .$$

Als Wert für u_0, also u an der Stelle $x_0 \notin \Omega$, verwenden wir nun

$$\begin{aligned} p(x_0) &= u_1 + \frac{x_0 - x_1}{x_1 - x_L}(u_1 - u_L) = u_1 + \frac{-h}{h + x_0 - x_L}(u_1 - u_L) \\ &= u_1 - \frac{h}{(1-\alpha_L)h}(u_1 - u_L) \; . \end{aligned}$$

Mit $u_0 \approx p(x_0)$ ergibt sich dann aus $u_1 - p(x_0) = \frac{1}{(1-\alpha_L)}(u_1 - u_L)$, dass

$$u''(x_1) \approx \frac{1}{h}\left(\frac{u_2 - u_1}{h} - \frac{u_1 - p(x_0)}{h}\right) = \frac{1}{h}\left(\frac{u_2 - u_1}{h} - \frac{u_1 - u_L}{(1-\alpha_L)h}\right)$$
$$= \frac{1}{h^2}\left(u_2 - u_1 - \frac{u_1 - u_L}{(1-\alpha_L)}\right) .$$

Am rechten Rand geht man analog vor.

Beide Ansätze liefern eine äquivalente Diskretisierung. Es stellt sich zudem heraus, dass diese Diskretisierung von 2. Ordnung ist, obwohl die Approximation am Rand nur von 1. Ordnung ist. Die Anpassung für die Randwerte modifiziert nur die erste und letzte Zeile der Systemmatrix. Durch geeignete Skalierung (Multiplikation des Systems mit einer Diagonalmatrix von links), kann die Matrix leicht symmetrisiert werden.

Selbstadjungierte Probleme

Das Poisson-Problem 2.5 ist ein Spezialfall der folgenden Klasse von PDEs mit ortsabhängigen Koeffizienten.

Problem 2.50 (Poisson-Problem mit ortsabhängigen Koeffizienten)**.** Sei $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$. Zu einem Quellterm $f : \Omega \to \mathbb{R}$, Randwerten $g : \partial\Omega \to \mathbb{R}$ und Koeffizienten $a : \Omega \to \mathbb{R}$ mit $a(x) > 0$, finde $u \in C^2(\Omega) \cap C(\overline{\Omega})$, so dass

$$\begin{aligned} -\operatorname{div}\big(a(x)\nabla u(x)\big) &= f(x) \quad \text{in } \Omega, \\ u(x) &= g(x) \quad \text{auf } \partial\Omega . \end{aligned}$$

Bemerkung 2.51. Mit $a \equiv 1$ erhalten wir genau das ursprüngliche Poisson-Problem (Problem 2.5). Man kann sich die Funktion a als Materialkoeffizient vorstellen. Es ist wichtig, dass der ortsabhängige Materialkoeffizient innerhalb der äußeren Ableitung steht. Dies ergibt sich aus der physikalischen Modellierung. Ist $q\colon \mathbb{R}^3 \to \mathbb{R}^3$ z.B. der Wärmefluss, so gilt mit einer Wärmequelle $f\colon \mathbb{R}^3 \to \mathbb{R}$

$$\operatorname{div} q(x) = f(x) .$$

Der Wärmefluss selbst ist nach dem Fourier'schen Gesetz durch

$$q(x) = -\lambda \nabla u(x)$$

gegeben, wenn u die Temperatur und λ der Wärmeleitkoeffizient ist. Falls λ auch ortsabhängig ist, so ergibt sich mit $a = \lambda$ genau die Gleichung aus Problem 2.50.

Satz 2.52. *Sei $a \in C^\infty(\Omega)$. Dann ist der Operator*

$$\Delta^{(a)}\colon C_0^\infty(\Omega) \to C_0^\infty(\Omega) , \quad u \mapsto -\operatorname{div}(a\nabla u)$$

selbstadjungiert bezüglich des Skalarproduktes

$$\langle u, v\rangle \coloneqq \int_\Omega u(x)v(x)\,\mathrm{d}x \ .$$

Beweis. Seien $u, v \in C_0^\infty(\Omega)$ beliebig. Zu zeigen ist $\langle \Delta^{(a)} u, v\rangle = \langle u, \Delta^{(a)} v\rangle$. Mit zweimaliger partieller Integration folgt

$$\begin{aligned}\langle \Delta^{(a)} u, v\rangle &= \langle -\operatorname{div}(a\nabla u), v\rangle = -\int_\Omega \operatorname{div}(a\nabla u) v \,\mathrm{d}x \\ &= \int_\Omega a\nabla u \cdot \nabla v \,\mathrm{d}x - \int_{\partial\Omega} av\nabla u \cdot n \,\mathrm{d}s \\ &= -\int_\Omega u\operatorname{div}(a\nabla v)\,\mathrm{d}x + \int_{\partial\Omega} au\nabla v \cdot n \,\mathrm{d}s = \langle u, \Delta^{(a)} v\rangle \,. \end{aligned}$$

□

Bemerkung 2.53. Wegen obiger Eigenschaft, d.h. $\langle \Delta^{(a)} u, v\rangle = \langle u, \Delta^{(a)} v\rangle$ für alle $u, v \in C_0^\infty(\Omega)$, nennt man Problem 2.50 selbstadjungiert (vgl. Mathe 1) und man erwartet, dass eine Diskretisierung auf eine symmetrische Systemmatrix führt. Dies ist allerdings nicht bei jeder Diskretisierung der Fall. Wir betrachten hierzu nur den Fall $d = 1$, d.h. die Gleichung vereinfacht sich zu

$$-(a(x)u'(x))' = f(x) \text{ für } x \in (0,1) \,.$$

Zur Diskretisierung betrachten wir verschiedene Möglichkeiten.

(i) Differenziert man das Problem erst aus und approximiert dann, so verliert man die Symmetrie. In der Tat, es gilt

$$-(a(x)u'(x))' = -a'(x)u'(x) - a(x)u''(x) \,.$$

Dies ist eine Konvektions-Diffusions-Gleichung mit mit ortsabhängigen Koeffizienten. Wie wir in Abschnitt 2.4 gesehen haben, müssen wir je nach Vorzeichen von $a'(x)$ eine Upwind-Richtung wählen und einen einseitigen Differenzenquotienten benutzen. Dies führt zu einer nicht-symmetrischen Matrix.

(ii) Wenn wir beide Ableitungen nacheinander naiv diskretisieren, so erhalten wir

$$\begin{aligned}-(a(x)u'(x))' &\approx -\frac{1}{2h}\big(a(x+h)u'(x+h) - a(x-h)u'(x-h)\big) \\ &\approx -\frac{1}{2h}\left(a(x+h)\frac{u(x+2h)-u(x)}{2h} - a(x-h)\frac{u(x)-u(x-2h)}{2h}\right) \,.\end{aligned}$$

Damit ergibt sich eine symmetrische Matrix, allerdings mit einem 5-Punkt-Stern, der sogar Punkte auslässt. Für einen 5-Punkt-Stern gibt es aber, wie bei den Finiten Differenzen höherer Ordnung, wieder Probleme Randwerte zu definieren.

(iii) Besser diskretisiert man auf einem versetzten Gitter:

$$\begin{aligned}-(a(x)u'(x))' &\approx -\frac{1}{h}\big(a(x+\tfrac{h}{2})u'(x+\tfrac{h}{2}) - a(x-\tfrac{h}{2})u'(x-\tfrac{h}{2})\big) \\ &\approx -\frac{1}{h}\left(a(x+\tfrac{h}{2})\frac{u(x+h)-u(x)}{h} - a(x-\tfrac{h}{2})\frac{u(x)-u(x-h)}{h}\right) \,.\end{aligned}$$

Dies ist ein 3-Punkt-Stern, man muss dazu nur den ortsabhängigen Koeffizienten anstatt auf den Gitterpunkten $x \pm h$ auf den versetzten Gitterpunkten $x \pm \frac{h}{2}$ auswerten. Es ergibt sich zudem eine symmetrische Systemmatrix.

2.6 Zeitabhängige Probleme

Wir haben bisher stationäre (elliptische) Probleme der Form

$$\mathcal{L}u = f \quad \text{in } \Omega \subset \mathbb{R}^d$$

mit einem elliptischen Differentialoperator $\mathcal{L}$ betrachtet, z.B. $\mathcal{L} = -\Delta$ oder $\mathcal{L} = -\varepsilon\Delta + v \cdot \nabla$. Ein entsprechendes zeitabhängiges (parabolisches) Problem ist

$$\partial_t u + \mathcal{L}u = f \quad \text{in } \Omega \times (0, T)$$

für die gesuchte Funktion

$$u : \Omega \times (0, T) \to \mathbb{R}, (x, t) \mapsto u(x, t)$$

zu einem gegebenen Endzeitpunkt T. Hierbei bezieht sich $\mathcal{L}$ nur auf Ortsableitungen, d.h. nach x. Für solche zeitabhängigen Probleme sind sowohl Rand- als auch Anfangswerte nötig um u eindeutig bestimmen zu können.

Beispiel 2.54. Ergänzt man die Poisson-Gleichung um $\partial_t u$, so erhält man die Diffusionsgleichung

$$\begin{aligned} \partial_t u(x,t) - \kappa\Delta_x u(x,t) &= f(x,t) \quad \text{für alle } (x,t) \in \Omega \times (0,T) , \\ u(x,t) &= g(x,t) \quad \text{für alle } (x,t) \in \partial\Omega \times (0,T) , && \text{(Randwerte)} \\ u(x,0) &= u_0(x) \quad \text{für alle } x \in \Omega . && \text{(Anfangswerte)} \end{aligned}$$

Hierbei bezeichnet $\Delta_x \equiv \Delta$, wie zuvor, den Laplace Operator, wobei der Index x manchmal verwendet wird um explizit darauf hinzuweisen, dass es sich dabei um einen Operator bezüglich Ortsableitungen handelt. Des Weiteren ist $\kappa > 0$ der konstante Diffusionskoeffizient.

Bemerkung 2.55. Ein Standardansatz zur Diskretisierung von Anfangsrandwertproblemen ist zuerst in der einen Variable und dann in der anderen Variable zu diskretisieren. Dies nennt man *Linienmethode*.

(i) Diskretisiert man zuerst im Ort, so erhält man ein System von ODEs, welches in einem zweiten Schritt in der Zeit diskretisiert wird. Dies nennt man *vertikale Linienmethode*.

(ii) Diskretisiert man zuerst in der Zeit, so erhält man ein System elliptischer PDEs, welches dann im Ort diskretisiert wird. Dies nennt man *horizontale Linienmethode*.

Abbildung 2.8 illustriert die beiden Ansätze schematisch.

Beispiel 2.56. Wir betrachten die vertikale Linienmethode für die Diffusionsgleichung in $\Omega = (0, 1)^2$. Für die Diskretisierung im Ort verwenden wir den gleichen Ansatz wie beim Poisson-Problem 2.11 (kartesisches Gitter), d.h. für $t \in [0, T]$ suchen wir die diskreten Punktwerte

$$u_h(\xi, t) \approx u(\xi, t) \quad \text{für alle } \xi \in \overline{\Omega}_h .$$

Die Werte im Inneren, also in Ω_h, ordnen wir genau wie in Problem 2.11 in einem langen (zeitabhängigen) Vektor $U(t) = (U_1(t), \ldots, U_m(t)) \in \mathbb{R}^m$ an, mit $m = (n-1)^2$ und $U_i(t) \coloneqq u_h(\xi_i, t)$ für alle $\xi_i \in \Omega_h$. Zur Approximation von $-\Delta_x$ nutzen wir zentrale Differenzen 2. Ordnung, d.h. $-\Delta_h$, bzw. die Matrix $A_1 \in \mathbb{R}^{m \times m}$. Dies führt auf das folgende (sehr große) gekoppelte ODE System

$$\partial_t U(t) = -\kappa A_1 U(t) + b(t) ,$$

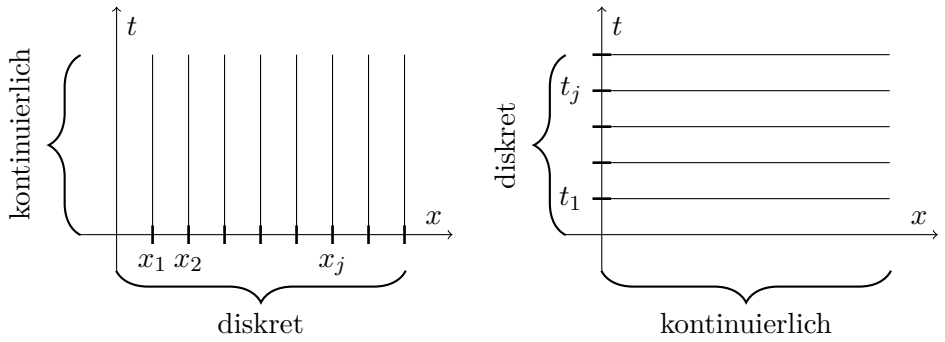


Abbildung 2.8: Vertikale (links) und horizontale (rechts) Linienmethode

wobei $b(t) \in \mathbb{R}^m$ analog zum Poisson-Problem aus $g(\cdot, t)$ und $f(\cdot, t)$ konstruiert wird.

Um dies numerisch zu lösen brauchen wir ein Zeitintegrationsverfahren. Hierzu gibt es verschiedene Möglichkeiten. Hier benutzen wir beispielhaft jedes Mal eine äquidistante Zeitdiskretisierung mit Schrittweite $\Delta t = \frac{T}{K}$ für $K \in \mathbb{N}$, d.h. wir betrachten die diskreten Zeitpunkte $t_k = k\Delta t$ mit $k \in \{0, \dots, K\}$. Ferner nutzen wir die Notation $U^k := U(t_k) \in \mathbb{R}^m$ und $b^k := b(t_k) \in \mathbb{R}^m$.

(i) *Explizites Euler-Verfahren:* Hiermit erhalten wir für $k \in \{0, \dots, K-1\}$:

$$\frac{U^{k+1} - U^k}{\Delta t} = -\kappa A_1 U^k + b^k \quad \Leftrightarrow \quad U^{k+1} = U^k - \Delta t \kappa A_1 U^k + \Delta t b^k \; .$$

Aus Mathe 3 wissen wir, dass hier die Stabilitätsbedingung $\Delta t \leq \frac{1}{\lambda}$ erfüllt sein muss, wobei λ der größte Eigenwert der symmetrisch positiv definiten Matrix κA_1 ist. Mit Satz 2.15 folgt sofort

$$\lambda = \kappa \lambda_{n-1,n-1} = \kappa \tfrac{4}{h^2}(2 \sin^2(\tfrac{1}{2}\pi(n-1)h)) = \kappa \tfrac{8}{h^2} \cos^2(\tfrac{1}{2}\pi h) \; .$$

Der Faktor $\frac{1}{h^2}$ ist für feine Gitter sehr groß, d.h. das ODE System ist *steif*. Die Stabilitätsbedingung ist hier

$$\Delta t \leq \frac{1}{\lambda} = \frac{h^2}{8\kappa \cos^2(\frac{1}{2}\pi h)} \approx \frac{h^2}{8\kappa} \quad \text{für } h \text{ klein} \; .$$

Es muss also $\Delta t \leq Ch^2$ gelten. Dies nennt man auch „parabolische Courant–Friedrichs–Lewy (CFL) Bedingung“. Damit benötigt eine Verfeinerung im Ort eine noch stärkere Verfeinerung in der Zeit, was in der Praxis zu rechenaufwendig sein kann.

(ii) *Implizites Euler-Verfahren:* Hier erhalten wir für $k \in \{0, \dots, K-1\}$:

$$\frac{U^{k+1} - U^k}{\Delta t} = -\kappa A_1 U^{k+1} + b^{k+1} \quad \Leftrightarrow \quad (I + \kappa \Delta t A_1) U^{k+1} = U^k + \Delta t b^{k+1} \; .$$

In jedem Zeitschritt muss hierbei also ein lineares Gleichungssystem gelöst werden. Dafür ist dieser Ansatz stabil für alle $\Delta t > 0$. Man kann zeigen, dass der Konvergenzfehler $\mathcal{O}(\Delta t + h^2)$ ist. Der Fehler im Ort ist also wesentlich kleiner als in der Zeit.

(iii) *Crank–Nicolson Verfahren*: Die Ungleichmäßigkeit des Fehlers in Ort und Zeit bei obigem Ansatz kann man mit der Trapez-Regel vermeiden:

$$\begin{aligned} & \frac{U^{k+1}-U^k}{\Delta t} = -\kappa A_1(\tfrac{1}{2}U^{k+1}+\tfrac{1}{2}U^k)+\tfrac{1}{2}(b^{k+1}+b^k) \\ \Leftrightarrow \quad & \left(I+\kappa\frac{\Delta t}{2}A_1\right)U^{k+1} = \left(I-\kappa\frac{\Delta t}{2}A_1\right)U^k+\frac{\Delta t}{2}(b^{k+1}+b^k)\,, \end{aligned}$$

für $k \in \{0,\dots,K-1\}$. Auch dieser Ansatz ist stabil für alle $\Delta t > 0$. Man kann weiter zeigen, dass der Konvergenzfehler hier nur $\mathcal{O}((\Delta t)^2+h^2)$ ist.

Beispiel 2.57. Nun betrachten wir die horizontale Linienmethode für die Diffusionsgleichung. Für die Diskretisierung in der Zeit nutzen wir die äquidistanten Zeitpunkte wie oben, also $t_k = k\Delta t$ mit $k \in \{0,\dots,K\}$. Somit suchen wir für $x \in \Omega$ die diskreten Punktwerte

$$u^k(x) = u(x,t_k) \quad \text{für } k \in \{0,\dots,K\}\,.$$

Mit dem impliziten Euler-Verfahren ergibt sich für $k \in \{0,\dots,K-1\}$:

$$\frac{u^{k+1}(x)-u^k(x)}{\Delta t} = \kappa\Delta_x u^{k+1}(x) + f(x,t_{k+1})\,.$$

Dies ist ein System elliptischer PDEs

$$\kappa\Delta_x \begin{bmatrix} u^1(x) \\ u^2(x) \\ \vdots \\ u^K(x) \end{bmatrix} = \frac{1}{\Delta t}\begin{bmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}\begin{bmatrix} u^1(x) \\ u^2(x) \\ \vdots \\ u^K(x) \end{bmatrix} + \frac{1}{\Delta t}\begin{bmatrix} -u_0(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} f(x,t_1) \\ \vdots \\ f(x,t_K) \end{bmatrix},$$

wobei $u_0(x) \coloneqq u(x,0)$ als Anfangswert gegebenen ist. Nutzen wir nun $\overline{\Omega}_h$ und Δ_h aus Problem 2.11 zur Approximation von Ω und Δ_x, so erhalten wir genau die gleiche Diskretisierung wie in Beispiel 2.56 (ii). Hier könnten wir aber auch zentrale Differenzen oder Differenzen höherer Ordnung für ∂_t verwenden.

3 Iterative Lösungsverfahren für große dünnbesetzte Gleichungssysteme

3.1 Einführung

Die Finite-Differenzen-Diskretisierung sowohl für das Poisson-Problem als auch für das Konvektions-Diffusions-Problem haben auf lineare Gleichungssysteme $Ax = b$ geführt. Es stellt sich die Frage, wie man diese Gleichungssysteme löst.

Bemerkung 3.1 (Eigenschaften der Gleichungssysteme). Viele Diskretisierungen für PDEs liefern lineare Gleichungssysteme der Form $Ax = b$ mit $A \in \mathbb{R}^{N\times N}$ und $b \in \mathbb{R}^N$ für ein entsprechendes $N \in \mathbb{N}$. Des Weiteren weisen diese Diskretisierungen oftmals ähnliche Eigenschaften auf, welche wir hier aus dem Blickwinkel der Finite-Differenzen-Diskretisierung betrachten.

(i) *Systemgröße N*:

- Für jeden Gitterpunkt gibt es eine Gleichung.
- Typischerweise ist für $\Omega \subset \mathbb{R}^d$ die Anzahl der Gitterpunkte/Knoten/Unbekannten $N \sim h^{-d}$.
- Die Tatsache, dass d im Exponent auftaucht, nennt man auch *Fluch der Dimension* (engl.: *curse of dimensionality*).
- Je feiner das Gitter, desto präziser die Lösung, desto größer aber auch die Matrix.
- „Aktuelle“ Gleichungssysteme haben eine Größe von $N \approx 10^{11}$, oder mehr.

Direkte Verfahren wie Gauss-Elimination und Cholesky-Zerlegung haben einen Rechenaufwand von $\mathcal{O}(N^3)$ und sind damit im Allgemeinen zu teuer zur Lösung der auftretenden Systeme.

(ii) *Dünnbesetztheit:*

- Wir betrachten das Poisson–Problem in 2D. Sei $u_i \in \mathbb{R}$ die diskrete Lösung am Knoten/Gitterpunkt $\xi_i \in \Omega_h$. Mittels zentralem Differenzen-Quotienten und Standardanordnung ist die Gleichung für u_i:

$$4u_i - u_{i-1} - u_{i+1} - u_{i-(n-1)} - u_{i+n-1} = f_i \ .$$

  Die i-te Zeile von A enthält also nur fünf Einträge, die ungleich Null sind.
- **Allgemein:** Die Anzahl der Nicht-Null-Einträge pro Zeile ist durch eine kleine Konstante beschränkt (Differentiation ist eine lokale Operation). Die Konstante ist unabhängig von h, wächst aber mit d.

Die Matrix enthält also nur $\mathcal{O}(N)$ Nicht-Null-Einträge. Solche Matrizen nennt man *dünnbesetzt* (engl.: *sparse*). Um dies in der Praxis auszunutzen, muss man besondere Datenstrukturen verwenden.

Wendet man das Gauss-Verfahren auf solch eine Matrix an, so entstehen bei den Zwischenschritten in der Matrix eine beträchtliche Anzahl von zusätzlichen Einträgen, die von Null verschieden sind („fill-in“). Das Gauss-Verfahren ist deshalb hier nicht nur zu langsam, es braucht auch zu viel Speicher.[1]

[1] Eine besondere Forschungsrichtung sind direkte Verfahren für dünnbesetzte Matrizen.

(iii) *Schlechte Kondition:* Bei der Fourier-Analyse des diskreten Poisson-Problems (vgl. Lemma 2.17) hatten wir gezeigt, dass

$$\kappa_2(A_1) = \mathcal{O}(h^{-2}) .$$

Es gibt ein ähnliches Verhalten für andere Diskretisierungen wie zum Beispiel bei der Finite-Elemente-Methode (Mathe 5). Die schlechte Kondition beeinflusst die Fehlerfortpflanzung in direkten Lösern.

(iv) *Symmetrie und positive Definitheit:* Die entstehenden Matrizen sind häufig symmetrisch und positiv definit, was wir z.B. für das Poisson-Problem gesehen hatten (Lemma 2.16), aber eben nicht für das Konvektions-Diffusions-Problem. Bei der Finite-Elemente-Methode sind die Matrizen für elliptische Probleme immer symmetrisch und positiv definit.

Ab jetzt betrachten wir das folgende, abstrakte Problem:

Problem 3.2. Für $n \in \mathbb{N}$ sei $A \in \mathbb{R}^{n\times n}$ regulär, „groß“ und dünnbesetzt, sowie $b \in \mathbb{R}^n$. Bestimme die Lösung des Gleichungssystem

$$Ax = b ,$$

welche wir von nun an mit $x^* \in \mathbb{R}^n$ bezeichnen wollen.

Bemerkung 3.3. Wir wollen Problem 3.2 iterativ lösen, d.h. die Lösung mit einer Folge (hoffentlich) immer besser approximieren. Hierzu formulieren wir $Ax = b$ als Fixpunktgleichung. Dazu wählen wir $C \in \mathbb{R}^{n\times n}$ regulär und definieren die Abbildung

$$\Phi\colon \mathbb{R}^n \to \mathbb{R}^n, x \mapsto \Phi(x) \coloneqq x + C(b - Ax) .$$

Dann ist die Lösung x^* von $Ax = b$ offensichtlich ein Fixpunkt von Φ, d.h. $\Phi(x^*) = x^*$. Andererseits ist auch jeder Fixpunkt von Φ eine Lösung des linearen Gleichungssystem, da C regulär ist.

Zur Bestimmung eines Fixpunktes betrachten wir die Fixpunktiteration

$$x^{k+1} \coloneqq \Phi(x^k) = x^k + C(b - Ax^k) = (I - CA)x^k + Cb , \quad k \in \mathbb{N}_0.$$

Diese Fixpunktiteration nennt man *lineares Verfahren* zur Lösung von $Ax = b$. Die Matrix $I - CA$ nennt man *Iterationsmatrix* des Verfahrens.

Die entscheidende Frage ist unter welchen Umständen dieses Verfahren konvergiert? Dazu betrachten wir den *Fehler im k-ten Schritt* $e^k \coloneqq x^k - x^* \in \mathbb{R}^n$. Es gilt

$$e^{k+1} = x^{k+1} - x^* = \Phi(x^k) - \Phi(x^*) = (I - CA)e^k .$$

Somit ergibt sich für die Entwicklung des Fehlers:

$$e^k = (I - CA)^k e^0 , \quad k \in \mathbb{N}_0 . \tag{3.1}$$

Die Fehlerfortpflanzung (3.1) ist somit linear, was der Grund ist weshalb man bei dieser Klasse von Verfahren von linearen Verfahren spricht.

Die Konvergenz linearer Verfahren ist mit Hilfe des folgenden wichtigen Satzes gesichert, für welchen wir uns vorher noch an den Spektralradius erinnern (Mathe 2).

Bemerkung 3.4. Der *Spektralradius* $\rho(A)$ einer Matrix $A \in \mathbb{R}^{n\times n}$ ist der betragsmäßig größte Eigenwert von A: $\rho(A) \coloneqq \max\{|\lambda| : \lambda \in \sigma(A)\}$. Des Weiteren gilt (Übung) für alle $A \in \mathbb{R}^{n\times n}$:

(i) $\rho(cA) = |c|\,\rho(A)$ für alle $c \in \mathbb{R}$,

(ii) $\rho(A^k) = \rho(A)^k$ für alle $k \in \mathbb{N}_0$,

(iii) $\rho(A) = \rho(A^T)$.

Satz 3.5. *Seien $A, C \in \mathbb{R}^{n\times n}$ regulär und $b \in \mathbb{R}^n$, $n \in \mathbb{N}$. Ferner x^* die Lösung von $Ax = b$. Dann gilt*

$$\lim_{k\to\infty} \|x^k - x^*\|_2 = 0 \text{ für alle } x^0 \in \mathbb{R}^n \;\Leftrightarrow\; \rho(I - CA) < 1\,.$$

Beweis.

„⇐“: Der Beweis für den allgemeinen Fall findet sich z.B. in dem Buch „Matrix Computations“ von Golub und Van Loan (siehe Theorem 10.1.1 in der dritten Auflage von 1996). Wir zeigen hier nur den Fall, dass die Matrix $I - CA$ diagonalisierbar ist.

Sei also $\rho(I - CA) < 1$. Da $I - CA$ diagonalisierbar ist, existiert eine invertierbare Matrix $T \in \mathbb{R}^{n\times n}$, so dass $T^{-1}(I - CA)T = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) =: D$, wobei $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von $I - CA$ sind. Es gilt weiter:

$$\|D^k\|_2 = \sqrt{\lambda_{\max}\left((D^k)^T D^k\right)} = \sqrt{\lambda_{\max}(D^{2k})} = \sqrt{\rho(D^{2k})} = \rho(I - CA)^k\,.$$

Aus (3.1) folgt dann wegen $I - CA = TDT^{-1}$, dass

$$e^k = (TDT^{-1})^k e^0 = TD^kT^{-1}e^0$$

und somit

$$\|e^k\|_2 \le \|T\|_2 \|T^{-1}e^0\|_2 \rho(I - CA)^k \to 0 \text{ für } k \to \infty\,,$$

da $\rho(I - CA) < 1$.

„⇒“: Beweis durch Widerspruch: Angenommen es sei $\rho(I-CA) \ge 1$. Weiter sei λ der Eigenwert von $I - CA$, der den Spektralradius liefert, d.h. $|\lambda| = \rho(I - CA)$, und v ein zugehöriger Eigenvektor. Für $x^0 \coloneqq v + x^*$, gilt $e^0 = v$ und es folgt

$$\|e^k\|_2 \overset{(3.1)}{=} \|(I - CA)^k v\|_2 = |\lambda|^k \|v\|_2 \overset{|\lambda|\ge 1}{\ge} \|v\|_2 = \|e^0\|_2 > 0 \text{ für alle } k \in \mathbb{N}\,.$$

Es gibt also einen Startwert für den das Verfahren nicht gegen x^* konvergiert, was den Widerspruch liefert. □

Da in endlich-dimensionalen Räumen alle Normen äquivalent sind, ändert sich das Resultat nicht, wenn man eine andere Norm betrachtet. Der Spektralradius einer Matrix ist jedoch oft nur mit Mühe auszurechnen. Allerdings gilt bekanntlich $\rho(B) \le \|B\|$ für jede Matrix und jede Norm (Mathe 2). Aus (3.1) folgt damit nun sofort das folgende Korollar.

Korollar 3.6. *Seien $A, C \in \mathbb{R}^{n\times n}$ regulär und $b \in \mathbb{R}^n$. Für jede Vektornorm $\|\cdot\|$ mit dazugehöriger Matrixnorm gilt*

$$\|x^k - x^*\| \leq \|I - CA\|^k \|x^0 - x^*\| \quad \text{für alle } k \in \mathbb{N}_0 \, .$$

Das lineare Verfahren konvergiert also, wenn $\|I - CA\| < 1$ gilt.

Es bleibt zu klaren, wie schnell die Folge x^k gegen x^* konvergiert. Insbesondere wollen wir verschiedene Verfahren bzgl. ihrer Geschwindigkeit vergleichen können.

Definition 3.7. Für ein lineares Verfahren nennt man

(i) $\sigma_k := \sqrt[k]{\frac{\|e^k\|}{\|e^0\|}}$ *mittlere Fehlerreduktionsrate* der ersten k Schritte und

(ii) $-\ln \sigma_k$ *Konvergenzgeschwindigkeit.*

Bemerkung 3.8.

(i) Die Rate um welche sich der Fehler im k-ten Schritt verändert ist der Quotient $\frac{\|e^k\|}{\|e^{k-1}\|}$. Somit ist σ_k genau das *geometrische Mittel* der Fehlerreduktionsraten der ersten k Schritte, denn es gilt:

$$\sigma_k^k = \prod_{j=1}^{k} \frac{\|e^j\|}{\|e^{j-1}\|} \, .$$

(ii) Für eine Reduktion des Fehlers um den Faktor $R > 1$ gilt

$$\sigma_k^k = \frac{\|e^k\|}{\|e^0\|} \overset{!}{\leq} \frac{1}{R} \;\Leftrightarrow\; k \ln \sigma_k \leq -\ln R \overset{\sigma_k<1}{\Leftrightarrow} k \geq \frac{\ln R}{-\ln \sigma_k} \, .$$

In diesem Sinne ist $-\ln \sigma_k$ also die Konvergenzgeschwindigkeit.

Wir zeigen nun, dass auch die Größe σ_k mit $\rho(I - CA)$ zusammenhängt.

Satz 3.9. *Seien $A, C \in \mathbb{R}^{n\times n}$ regulär und $b \in \mathbb{R}^n$. Ferner sei $I - CA$ diagonalisierbar. Falls e^0 nicht aus der Summe der Eigenräume von $I - CA$ zu Eigenwerten betragsmäßig strikt kleiner $\rho(I - CA)$ ist, so gilt*

$$\lim_{k\to\infty} \sigma_k = \rho(I - CA) \, .$$

Beweis. Da $I - CA$ diagonalisierbar ist, gibt es eine normierte Eigenvektorbasis $v_1, \ldots, v_n$ mit betragsmäßig absteigenden Eigenwerten $|\lambda_1| \geq \ldots \geq |\lambda_n|$. Der Anfangsfehler lässt sich in der Eigenvektorbasis darstellen, d.h. es gibt $c_1, \ldots, c_n \in \mathbb{R}$ mit

$$e^0 = \sum_{i=1}^{n} c_i v_i \, .$$

Nach Voraussetzung gilt o.B.d.A. $c_1 \neq 0$ (sonst Umnummerierung der Eigenwerte die betragsmäßig mit $|\lambda_1|$ übereinstimmen). Falls $C = A^{-1}$ folgt $I - CA = 0$ und $x^1 = A^{-1}b = x^*$, d.h.

$e^1 = 0$, und die Aussage ist klar. Sei nun $C \neq A^{-1}$. Es folgt $I - CA \neq 0$ und somit auch $\lambda_1 \neq 0$. Für $k \in \mathbb{N}$ folgt, dass

$$e^k = (I - CA)^k \sum_{i=1}^{n} c_i v_i = \sum_{i=1}^{n} c_i \lambda_i^k v_i = \lambda_1^k \Bigg(c_1 v_1 + \underbrace{\sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i}_{=:r^k} \Bigg)$$

$$= \lambda_1^k (c_1 v_1 + r^k) \; .$$

Da alle Normen auf dem $\mathbb{R}^n$ äquivalent sind, können wir insbesondere jede Norm nach oben und unten mit der Supremumsnorm zur Basis $v_1, \ldots, v_n$ abschätzen. Da außerdem $c_1 \neq 0$ und $|\frac{\lambda_i}{\lambda_1}| \leq 1$, existieren Konstanten $C_{\min}$ und $C_{\max}$ (unabhängig von k) mit

$$0 < C_{\min} \leq \|c_1 v_1 + r^k\| \leq C_{\max} \; .$$

Insgesamt folgt also

$$\sigma_k = \left(\frac{\|e^k\|}{\|e^0\|} \right)^{\frac{1}{k}} = \frac{|\lambda_1| \|c_1 v_1 + r^k\|^{\frac{1}{k}}}{\|e^0\|^{\frac{1}{k}}} \overset{k \to \infty}{\longrightarrow} |\lambda_1| = \rho(I - CA) \; . \qquad \square$$

Bemerkung 3.10. Wegen Satz 3.9 bezeichnet man $-\ln(\rho(I - CA))$ als *asymptotische Konvergenzgeschwindigkeit*. Für große k ist $\rho(I - CA)$ in etwa die mittlere Fehlerreduktionsrate σ_k.

Bemerkung 3.11. Iterative Verfahren liefern nach endlich vielen Schritten keine exakte Lösung (anders als z.B. die Gauss-Elimination). Allerdings reduzieren sie im Fall der Konvergenz den Fehler erheblich. Der Gesamtaufwand setzt sich aus zwei Aspekten zusammen:

(i) die Anzahl der Schritte, die nötig ist um den Fehler um den Faktor R zu reduzieren,

(ii) und dem Aufwand pro Iterationsschritt.

Eine untere Grenze für den Aufwand für ein System mit n Gleichungen ist $\mathcal{O}(n)$.

3.2 Jacobi- und Gauss-Seidel-Verfahren

Es bleibt die Frage: Wie sollte man C wählen? Je kleiner $\rho(I - CA)$, desto höher ist die Konvergenzgeschwindigkeit. Ideal wäre $C = A^{-1}$, denn dann ist $\rho(I - CA) = 0$ und das lineare Verfahren konvergiert in einem Schritt:

$$x^1 = x^0 + A^{-1}(b - Ax^0) = A^{-1}b = x^* \; .$$

Die Durchführung dieses Schrittes ist aber sehr teuer, denn für die Auswertung von $A^{-1}(b - Ax^0)$ muss das lineare Gleichungssystem $Ac = b - Ax^0$ gelöst werden. Man hat also nichts gewonnen.

Es gilt einen Kompromiss für die folgenden zwei gegensätzlichen Bedingungen zu finden:

(i) C soll A^{-1} möglichst gut approximieren.

(ii) Die Operation $y \mapsto Cy$ soll möglichst „billig“ sein.

Hierbei kann es hilfreich sein, dass die Matrix C nicht explizit ausgerechnet werden muss, denn es reicht Cy auswerten zu können.

Im Folgenden werden nun zwei Verfahren vorgestellt, die zu einer Matrix C führen.

Bemerkung 3.12 (Jacobi-Verfahren). Wir betrachten $A = (a_{ij})_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$ und nehmen an alle Diagonaleinträge von A seien nicht Null (d.h. $a_{ii} \neq 0$). Die i-te Zeile des LGS $Ax = b$ ist

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \ .$$

Idee: Wir lösen für alle $i \in \{1,\dots,n\}$ die $i-$te Zeile nach x_i auf

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

und machen daraus jeweils eine Fixpunktiteration:

$$x_i^{k+1} = a_{ii}^{-1}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^k\right) \quad \text{für alle } i \in \{1,\dots,n\} \ . \tag{3.2}$$

Diese Iteration nennt man *Jacobi-Iteration*, das resultierende Verfahren heißt *Jacobi-Verfahren*. Hierbei ist folgendes zu beachten:

(i) Die Rechnungen für die verschiedenen i sind voreinander unabhängig. Deswegen ist das Verfahren leicht zu parallelisieren.

(ii) In der Praxis geht die Summe natürlich nur über die Einträge der i-ten Zeile von A, die nicht Null sind.

Das Jacobi-Verfahren lässt sich kompakter schreiben. Sei dazu $D \coloneqq \operatorname{diag}(a_{11},\dots,a_{nn}) \equiv \operatorname{diag}(A) \in \mathbb{R}^{n\times n}$. Dann ist die Jacobi-Iteration (3.2) äquivalent zu

$$x^{k+1} = D^{-1}\big(b - (A-D)x^k\big) = D^{-1}b - D^{-1}Ax^k + x^k = (I - D^{-1}A)x^k + D^{-1}b \ .$$

Also ist das Jacobi-Verfahren das lineare Verfahren mit $C = D^{-1}$. Es gibt eine weitere häufig verwendete Schreibweise. Sei L eine untere und U obere Dreiecksmatrix, so dass

$$A = -L + D + -U \ .$$

Die Jacobi-Iteration ist dann äquivalent zu

$$Dx^{k+1} = (L+U)x^k + b \ .$$

Wir erhalten dann sofort das folgende Resultat.

Satz 3.13. *Das Jacobi-Verfahren konvergiert genau dann, wenn* $\rho(I - D^{-1}A) < 1$.

Beweis. Folgt sofort aus Satz 3.5. □

Beispiel 3.14. Leider gilt nicht immer $\rho(I - D^{-1}A) < 1$. Zum Beispiel ist

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \ \Rightarrow \ I - D^{-1}A \overset{D=I}{=} \begin{pmatrix} 0 & -2 \\ -2 & 0 \end{pmatrix}.$$

Somit hat die Iterationsmatrix die Eigenwerte -2 und 2 zu den Eigenvektoren $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}$. Also ist $\rho(I - CA) = 2 > 1$ und das Jacobi-Verfahren konvergiert nicht für dieses A.

Es ist möglich die folgenden hinreichenden Bedingungen herzuleiten:

Satz 3.15. *Es sei eine der beiden folgenden Bedingungen erfüllt:*

(i) Sowohl die Matrix A als auch die Matrix $2D - A$ sind positiv definit.

(ii) Die Matrix A ist irreduzibel diagonaldominant.

Dann konvergiert das Jacobi-Verfahren.

Beweis. Den Beweis findet man beispielsweise in dem Buch „Iterative Solution of Large Sparse Systems of Equations“ von Wolfgang Hackbusch (Theorem 4.4.11 liefert (i), Theorem 6.4.10 liefert (ii)). □

Bemerkung 3.16 (Aufwand des Jacobi-Verfahrens beim Poisson-Problem). Sei $A_1 \in \mathbb{R}^{n\times n}$ die Matrix des diskretisierten Poisson-Problems; zur Erinnerung $n = \mathcal{O}(h^{-2})$. Dann ist $D = 4h^{-2}I$ und die Fourier-Analyse in Satz 2.15 liefert:

$$\begin{aligned}
\rho(I - D^{-1}A_1) &= \max_{\|x\|=1} \left|x^T(I - D^{-1}A_1)x\right| = \max_{\|x\|=1} \left|1 - x^T D^{-1}A_1 x\right| \\
&= \max_{\|x\|=1} \left|1 - \tfrac{1}{4}h^2 x^T A_1 x\right| \\
&= \max\left\{|1 - \tfrac{1}{4}h^2\lambda| \colon \lambda \in \sigma(A_1)\right\} \\
&\overset{\text{Satz 2.15}}{=} \max\left\{|1 - (\sin^2(\tfrac{1}{2}\pi\nu h) + \sin^2(\tfrac{1}{2}\pi\mu h))| \colon 1 \le \nu, \mu \le n-1\right\} \\
&= \left|1 - 2\sin^2(\tfrac{1}{2}\pi h)\right| \\
&= \cos(\pi h) = 1 - \tfrac{1}{2}\pi^2 h^2 + \mathcal{O}(h^4)\,,
\end{aligned}$$

für $0 \le h < \frac{1}{2}$, wobei die letzte Gleichheit aus einer Taylorentwicklung folgt, die wir schon im Beweis von Lemma 2.17 benutzt haben. Also ist die mittlere Fehlerreduktionsrate $\sigma_k \approx 1 - \frac{1}{2}\pi^2 h^2$ für $k \gg 1$ (vgl. Bemerkung 3.10). Dieser Term geht für feine Gitter sehr schnell gegen 1. Um dem Startfehler um den Faktor $R > 1$ zu reduzieren braucht man etwa

$$\frac{\ln R}{-\ln \rho(I - D^{-1}A)} \approx \frac{\ln R}{-\ln(1 - \frac{1}{2}\pi^2 h^2)} \approx \frac{2}{\pi^2 h^2} \ln R = \mathcal{O}(n)$$

Schritte. Hierbei haben wir die Taylorentwicklung $\ln(1 + x) = x + \mathcal{O}(x^2)$ verwendet. Jeder Schritt erfordert $\mathcal{O}(n)$ Operationen, da A_1 dünnbesetzt ist. Also lässt sich der Gesamtaufwand des Jacobi-Verfahrens durch $\mathcal{O}(n^2)$ abschätzen.

Bemerkung 3.17 (Gauss–Seidel-Verfahren). Wir betrachten noch einmal die Jacobi-Iteration, genauer gesagt die Fixpunktiteration für den i-ten Eintrag:

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - a_{i1}x_1^k - \ldots - a_{i(i-1)}x_{i-1}^k - a_{i(i+1)}x_{i+1}^k - \ldots - a_{in}x_n^k\right)\,, \quad i \in \{1, \ldots n\}\,.$$

Wenn wir die x_i^{k+1} für steigendes i nacheinander ausrechnen, dann gibt es bei der Berechnung von x_i^{k+1} für $x_1, \ldots, x_{i-1}$ schon bessere Werte als $x_1^k, \ldots, x_{i-1}^k$, nämlich die Werte der nächsten Iteration $x_1^{k+1}, \ldots, x_{i-1}^{k+1}$. Unter dieser Betrachtung ergibt sich das *Gauss–Seidel-Verfahren* mit der Iterationsvorschrift

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

Hier hängt also x_i^{k+1} von x_{i-1}^{k+1} ab, weshalb sich dieses Verfahren nicht einfach parallelisieren lässt.

Seien $D, -L$ und $-U$ wieder die Diagonale, unterer und oberer Dreiecksanteil von A, so dass $A = D - L - U$. Dann ist die Iterationsvorschrift des Gauss–Seidel-Verfahrens äquivalent zu

$$\begin{aligned} & x^{k+1} = D^{-1}(b + Lx^{k+1} + Ux^k) \quad \Leftrightarrow \quad (D-L)x^{k+1} = Ux^k + b \\ \Leftrightarrow \quad & x^{k+1} = (D-L)^{-1}Ux^k + (D-L)^{-1}b \\ & \quad = x^k + \big((D-L)^{-1}U - I\big)x^k + (D-L)^{-1}b \\ & \quad = x^k + (D-L)^{-1}\underbrace{\big(U - (D-L)\big)}_{=-A}x^k + (D-L)^{-1}b \\ & \quad = \big(I - (D-L)^{-1}A\big)x^k + (D-L)^{-1}b\,. \end{aligned}$$

Also ist das Gauss–Seidel-Verfahren das lineare Verfahren mit $C = (D-L)^{-1}$.

Wir erwarten bessere Konvergenzeigenschaften als beim Jacobi-Verfahren, und in der Tat gilt der folgende Satz.

Satz 3.18. *Es sei eine der beiden folgenden Bedingungen erfüllt:*

(i) Die Matrix A ist positiv definit.

(ii) Die Matrix A ist irreduzibel diagonaldominant.

Dann konvergiert das Gauss-Seidel-Verfahren.

Beweis. Den Beweis findet man beispielsweise in dem Buch „Iterative Solution of Large Sparse Systems of Equations“ von Wolfgang Hackbusch (Theorem 4.4.18 liefert (i), Theorem 6.6.8 liefert (ii)). □

Bemerkung 3.19 (Aufwand des Gauss–Seidel-Verfahrens beim Poisson-Problem). Man kann zeigen, dass $\rho(I - (D-L)^{-1}A_1) = \rho(I - D^{-1}A_1)^2$ gilt (Remark 5.6.8 im Buch von Hackbusch). Dies gilt nicht für alle Matrizen A, aber A_1 erfüllt die nötigen Voraussetzungen, die wir hier allerdings nicht weiter diskutieren wollen. Mit $\rho(I - D^{-1}A_1) = \cos(\pi h)$ (Bemerkung 3.16) folgt

$$\rho(I - (D-L)^{-1}A_1) = \cos^2(\pi h) \approx (1 - \tfrac{1}{2}\pi^2 h^2 + \mathcal{O}(h^4))^2 = 1 - \pi^2 h^2 + \mathcal{O}(h^4)\,.$$

Also ist $\sigma_k \approx 1 - \pi^2 h^2$ für $k \gg 1$ (vgl. Bemerkung 3.10). Um den Startfehler um den Faktor $R > 1$ zu reduzieren braucht man etwa

$$\frac{\ln R}{-\ln \rho(I - (D-L)^{-1}A_1)} \approx \frac{\ln R}{-\ln(1 - \pi^2 h^2)} \approx \frac{1}{\pi^2 h^2} \ln R$$

Iterationen. Hierbei haben wir die Taylorentwicklung $\ln(1+x) = x + \mathcal{O}(x^2)$ verwendet. Die Faustregel ist: Das Gauss-Seidel-Verfahren braucht nur etwa halb so viele Iterationen wie das Jacobi-Verfahren. Der Gesamtaufwand ist jedoch $\mathcal{O}(n^2)$, also ebenfalls quadratisch.

Bemerkung 3.20 (Abbruchkriterien). Es bleibt noch die Frage: Wie viele Iterationen sollte man durchfuhren? Schätzungen wie $\frac{1}{\pi^2 h^2} \ln R$ oben, sind nur für wenige Spezialfälle bekannt. Der häufigste Ansatz ist das Residuum $r^k := b - Ax^k$ zu betrachten. Es gilt

$$\|e^k\| = \|x^* - x^k\| = \|A^{-1}(b - Ax^k)\| \le \|A^{-1}\|\|r^k\|\,.$$

Man kann mit dem Residuum den Fehler von oben abschätzen, falls $\|A^{-1}\|$ bekannt ist. Dies ist allerdings meist ebenso nicht der Fall. Stattdessen bricht man die Iterationen üblicherweise ab, wenn $\frac{\|r^k\|}{\|r^0\|} < \frac{1}{R}$. Damit benutzt man also die Näherung

$$\frac{\|e^k\|}{\|e^0\|} = \frac{\|A^{-1}r^k\|}{\|A^{-1}r^0\|} \approx \frac{\|r^k\|}{\|r^0\|}.$$

Wie gut diese Näherung ist hängt allerdings stark von A^{-1} ab.

3.3 CG-Verfahren

Ab jetzt sei $A \in \mathbb{R}^{n\times n}$ eine symmetrische und positiv definite Matrix. Wir suchen immer noch eine iterative Methode zur Lösung von $Ax = b$.

Bemerkung 3.21. Anstatt als Fixpunktiteration ist unser Ansatz hier das Gleichungssystem als Optimierungsproblem zu formulieren. Dafür hätten wir gerne eine Funktion $f : \mathbb{R}^n \to \mathbb{R}$, deren Gradient

$$\nabla f(x) = Ax - b$$

ist, denn dann ist ein kritischer Punkte von f, also eine Nullstelle von ∇f, eine Lösung von $Ax = b$. Eine solche Funktion ist

$$f(x) \coloneqq \tfrac{1}{2}x^T Ax - b^T x \ .$$

Da A positiv definit ist, hat f einen einzigen Minimierer x^* und dieser ist die eindeutige Lösung von $Ax = b$.

Einer der einfachsten Optimierungsalgorithmen ist das *Verfahren des steilsten Abstiegs*, auch Gradientenabstieg genannt (engl.: *steepest descent* oder *gradient descent*). Die Idee ist, ausgehend von x_k (die Approximation von x^* zum Iterationsschritt k) einen Schritt in die Richtung des steilsten Abstiegs zu machen. Diese *Abstiegsrichtung* ist $-\nabla f(x_k) = b - Ax_k =: r_k$. Somit ergibt sich die Iteration

$$x_{k+1} = x_k + \alpha_k r_k \ , \quad k \in \mathbb{N}_0 \ ,$$

wobei $\alpha_k \in \mathbb{R}$ die noch zu bestimmende *Schrittlänge* bezeichnet. Wie lang sollte der Schritt sein? Dafür führt man eine sog. *Liniensuche* durch. Hierbei minimiert man f ausgehend von x_k entlang der Suchrichtung r_k, für α_k muss also gelten

$$0 \overset{!}{=} \frac{\partial}{\partial\alpha} f(x_k + \alpha r_k)\Big|_{\alpha=\alpha_k} = \nabla f(x_{k+1})^T r_k \ .$$

Wegen $\nabla f(x_{k+1}) = -r_{k+1}$ folgt daraus

$$0 = r_{k+1}^T r_k = (b - Ax_{k+1})^T r^k = \big(b - A(x_k + \alpha_k r_k)\big)^T r_k = \underbrace{(b - Ax_k)}_{r_k}{}^T r_k - \alpha_k \underbrace{(Ar_k)^T r_k}_{=r_k^T A^T r_k},$$

woraus mit $A = A^T$ folgt

$$\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k} \ .$$

Zur Berechnung von α_k benötigt man also eine Matrix-Vektor Multiplikation.

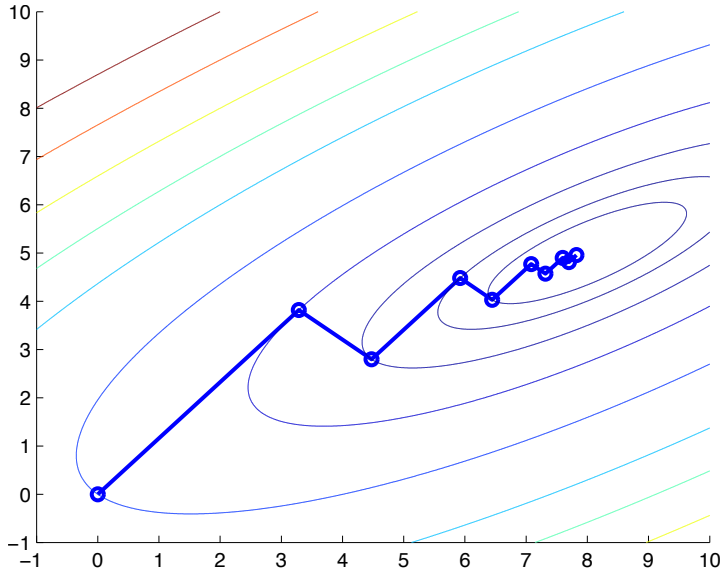


Abbildung 3.9: Schritte des Verfahrens des steilsten Abstiegs und Höhenlinien der Funktion $f(x) = \frac{1}{2}x^T Ax - b^T x$.

Das Verfahren der konjugierten Gradienten (CG–Verfahren)

In Abbildung 3.9 sehen wir, dass das Gradientenverfahren häufig mehrfach in die gleiche Richtung minimiert. Wir sind aber in $\mathbb{R}^n$, deshalb müssen wir den ersten Startwert x_0 eigentlich nur in n Richtungen ändern. Besser als das Gradientenverfahren wäre also doch, wenn wir n (momentan vorgegebene) orthogonale Suchrichtungen $\{d_0, \ldots, d_{n-1}\}$ anstatt der naiven Abstiegsrichtungen hätten und bei jedem Schritt genau die richtige Länge wählten:

$$x_{k+1} = x_0 + \operatorname*{argmin}_{x \in \mathcal{D}_k} \|x^* - x_0 - x\|^2 , \tag{3.3}$$

wobei

$$\mathcal{D}_k \coloneqq \operatorname{span}\{d_0, \ldots, d_k\} .$$

Da $\mathcal{D}_{n-1} = \mathbb{R}^n$, ist somit $x_n = x^*$ und wir sind (spätestens) in n Schritten fertig.

Die Lösung des Optimierungsproblems in (3.3) hat die folgende wichtige Eigenschaft:

Lemma 3.22 (Orthogonalitätsprinzip)**.** *Seien $y \in \mathbb{R}^n$ und*

$$x_{\mathcal{D}} = \operatorname*{argmin}_{x \in \mathcal{D}} \|y - x\|^2 ,$$

wobei $\mathcal{D}$ ein linearer Unterraum des $\mathbb{R}^n$ ist. Dann ist der Fehler $y - x_{\mathcal{D}}$ orthogonal zu $\mathcal{D}$:

$$d^T(y - x_{\mathcal{D}}) = 0 \quad \textit{für alle } d \in \mathcal{D} .$$

Beweis. Folgt direkt aus den Aussagen zur Bestapproximation in Mathe 1. □

Im Fall von (3.3), also $y = x^* - x_0$ und $x_{\mathcal{D}} = x_{k+1} - x_0$, ergibt sich das Orthogonalitätsprinzip

$$d^T(x^* - x_{k+1}) = 0 \quad \text{für alle } d \in \mathcal{D}_k .$$

Damit können wir zeigen, dass die Schritte (3.3) die vereinfachte Form

$$x_{k+1} = x_k + \alpha_k d_k$$

haben. Wegen (3.3) gilt $x_{k+1} - x_0 \in \mathcal{D}_k$, also gibt es Koeffizienten $c_{kj} \in \mathbb{R}$, $j \in \{0, \dots, k\}$, mit

$$x_{k+1} = x_0 + \sum_{j=0}^{k} c_{kj} d_j \ . \tag{3.4}$$

Aus dem Orthogonalitätsprinzip und der Orthogonalität der Suchrichtungen $d_j \in \mathcal{D}_k$ folgt

$$0 = d_i^T (x^* - x_{k+1}) = d_i^T (x^* - x_0 - c_{ki} d_i) \ ,$$

für alle $i \in \{0, \dots, k\}$ und somit

$$c_{ki} = \frac{d_i^T (x^* - x_0)}{d_i^T d_i} = \frac{d_i^T (x^* - x_i)}{d_i^T d_i} =: \alpha_i \ , \quad i \in \{0, \dots, k\} \ ,$$

wobei die vorletzte Gleichung aus der Orthogonalität der Suchrichtungen $d_j \in \mathcal{D}_k$ zusammen mit (3.4) für $k = i - 1$ folgt. Gleichung (3.4) wird also

$$x_{k+1} = x_0 + \sum_{j=0}^{k} \alpha_j d_j = \underbrace{x_0 + \sum_{j=0}^{k-1} \alpha_j d_j}_{=x_k} + \alpha_k d_k = x_k + \alpha_k d_k \ . \tag{3.5}$$

Dabei gibt es aber ein wesentliches Problem: Um die Schrittlänge α_k zu berechnen, benötigen wir x^*, also die Lösung, die wir berechnen wollen. Hier rettet uns die erste Idee des CG-Verfahrens: Wenn wir das normale Skalarprodukt $x^T y$ durch das A-gewichtete Skalarprodukt $x^T A y$ ersetzen, können wir α_k im Wesentlichen genauso berechnen. Das heißt, wenn die Suchrichtungen A-orthogonal sind, wenn also

$$d_i^T A d_j = 0 \quad \text{für } i, j \in \{0, \dots, n-1\} \text{ mit } i \neq j \ , \tag{3.6}$$

gilt, dann gilt alles Obige (auch das Orthogonalitätsprinzip), wenn wir auch die normale Norm $\|\cdot\|$ in (3.3) durch die so genannte *Energienorm* $\|x\|_A^2 := x^T A x$ (und überall das normale Skalarprodukt durch das A-gewichtete Skalarprodukt) ersetzen. Insgesamt erhalten wir also

$$x_{k+1} = x_0 + \operatorname*{argmin}_{x \in \mathcal{D}_k} \|x^* - x_0 - x\|_A^2 \ . \tag{3.7}$$

Dann folgt für die Schrittlänge im k-ten Schritt:

$$\alpha_k = \frac{d_k^T A (x^* - x_k)}{d_k^T A d_k} \overset{Ax^* = b}{=} \frac{d_k^T r_k}{d_k^T A d_k} \ . \tag{3.8}$$

Wegen $Ax^* = b$, können wir die Schrittlänge für diesen Ansatz berechnen ohne x^* zu kennen.

Es bleibt nun nur die Suchrichtungen zu wählen. Hierzu gehen wir wie folgt vor. Zuerst können wir linear unabhängige Vektoren $\{u_0, \dots, u_{n-1}\}$ wählen und davon mittels Gram-Schmidt-Orthogonalisierung die A-orthogonalen Suchrichtungen erzeugen. Die Wahl beim CG-Verfahren ist

$$u_k = r_k \ .$$

Diese Wahl führt zu überraschend vielen Vereinfachungen der benötigten Rechnungen, aber erst müssen wir zeigen, dass die Residuen linear unabhängig sind. Dabei hilft das folgende Lemma.

Lemma 3.23. *Für alle $k \in \{0, \dots, n-1\}$ und $j \in \{0, \dots, k-1\}$ gilt:*

$$0 = d_j^T r_k \ .$$

Beweis. Die Aussage ergibt sich wie folgt:

$$\begin{aligned} d_j^T r_k = d_j^T A(x^* - x^k) &\overset{(3.5)}{=} d_j^T A \left(x^* - x_0 - \sum_{i=0}^{k-1} \alpha_i d_i \right) \overset{(3.6)}{=} d_j^T A(x^* - x_0 - \alpha_j d_j) \\ &= d_j^T A(x^* - x_0) - \alpha_j d_j^T A d_j \overset{(3.8)}{=} d_j^T A(x^* - x_0) - d_j^T r_j = d_j^T A(x^* - x_0 - (x^* - x_j)) \\ &= d_j^T A(x_j - x_0) \overset{(3.5)}{=} d_j^T A \sum_{i=0}^{j-1} \alpha_i d_i \overset{(3.6)}{=} 0 \ . \end{aligned}$$

□

Hiermit können wir nun die lineare Unabhängigkeit der r_i zeigen.

Lemma 3.24. *Sei $n \in \mathbb{N}$ und $j \in \{0, \dots, n-2\}$. Sind $\alpha_0, \dots, \alpha_j \neq 0$, so sind $r_0, \dots, r_j$ linear unabhängig. Ist zusätzlich $r_{j+1} \neq 0$, so sind $r_0, \dots, r_{j+1}$ linear unabhängig.*

Beweis. Aus Lemma 3.23 folgt

$$0 = d_j^T r_k = d_j^T A(x^* - x^k) \text{ für alle } k \in \{0, \dots, n-1\} \text{ und } j \in \{0, \dots, k-1\},$$

d.h. $(x^* - x^k) \perp_A \mathcal{D}_{k-1}$. Hieraus folgt

$$(x^* - x^k) \in \mathcal{D}_{k-1}^{\perp_A} = \operatorname{span}\{d_k, \dots, d_{n-1}\} \text{ für alle } k \in \{0, \dots, n-1\} \ .$$

Durch Anwenden von A auf beiden Seiten folgt

$$r_k \in \operatorname{span}\{Ad_k, \dots, Ad_{n-1}\} \text{ für alle } k \in \{0, \dots, n-1\} \ . \qquad (*)$$

Wegen Gleichung (3.8) folgt aus $\alpha_0, \dots, \alpha_j \neq 0$, dass auch $r_0, \dots, r_j \neq 0$. Wären also $r_0, \dots, r_j$ linear abhängig, so gäbe es ein $k \in \{0, \dots, j-1\}$, mit $r_k \in \operatorname{span}\{r_{k+1}, \dots, r_j\}$. Um die lineare Unabhängigkeit von $r_0, \dots, r_j$ zu zeigen, genügt es also zu zeigen, dass $r_k \notin \operatorname{span}\{r_{k+1}, \dots, r_{n-1}\}$ für alle $k \in \{0, \dots, j-1\}$. Angenommen dies ist falsch. Dann gibt es ein $k \in \{0, \dots, j-1\}$ mit

$$r_k \in \operatorname{span}\{r_{k+1}, \dots, r_{n-1}\} \overset{(*)}{\subset} \operatorname{span}\{Ad_{k+1}, \dots, Ad_{n-1}\} \ \Rightarrow \ d_k^T r_k = 0 \ \overset{(3.8)}{\Rightarrow} \ \alpha_k = 0 \ ,$$

was aber der Voraussetzung an die α_i widerspricht. Hieraus folgt die lineare Unabhängigkeit von $r_0, \dots, r_j$. Mit $r_{j+1} \neq 0$ folgt mit obiger Argumentation auch direkt die lineare Unabhängigkeit von $r_0, \dots, r_{j+1}$. □

Als nächstes wenden wir die Gram-Schmidt-Orthogonalisierung bzgl. dem A-gewichteten Skalarproduktes an (Mathe 1). Dies führt auf

$$d_k = r_k + \sum_{j=0}^{k-1} \beta_{k,j} d_j \ , \qquad (3.9)$$

wobei

$$\beta_{k,j} = -\frac{r_k^T A d_j}{d_j^T A d_j} \ .$$

Hieraus folgt aus Lemma 3.23

$$d_k^T r_k = r_k^T r_k + \sum_{j=0}^{k-1} \beta_{k,j} d_j^T r_k \overset{\text{Lemma 3.23}}{=} r_k^T r_k$$

und damit gilt auch

$$\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k} = \frac{r_k^T r_k}{d_k^T A d_k} \, .$$

Insbesondere bedeutet dies, dass aus $r_k \neq 0$ auch $\alpha_k \neq 0$ folgt. Wegen Lemma 3.24 sind die Residuen also linear unabhängig, solange sie nicht verschwinden. Falls aber ein Residuum verschwindet hätten wir die Lösung bereits gefunden.

Jetzt sehen wir warum die Wahl $u_k = r_k$ besonders günstig war: die meisten Koeffizienten $\beta_{k,j}$ verschwinden. Um dies zu sehen, beobachten wir zuerst, dass

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k d_k) = r_k - \alpha_k A d_k \;\Rightarrow\; A d_k = \frac{1}{\alpha_k}(r_k - r_{k+1}) \, .$$

Somit ist der Zähler von $\beta_{k,j}$ für $j < k$

$$r_k^T A d_j = \frac{1}{\alpha_j} r_k^T (r_j - r_{j+1}) = \begin{cases} -\dfrac{1}{\alpha_{k-1}} r_k^T r_k & \text{falls } j = k-1 \, , \\ 0 & \text{sonst,} \end{cases} \tag{3.10}$$

da $r_k^T r_j = 0$ für $j < k$, denn aus Lemma 3.23 folgt:

$$r_k^T r_j \overset{(3.9)}{=} r_k^T \left(d_j - \sum_{i=0}^{j-1} \beta_{j,i} d_i \right) = 0 \, .$$

Also gilt $\beta_{k,j} = 0$ für $j < k-1$ und die Gram-Schmidt-Orthogonalisierung vereinfacht sich zu

$$d_k = r_k + \beta_{k,k-1} d_{k-1} = r_k - \frac{r_k^T A d_{k-1}}{d_{k-1}^T A d_{k-1}} d_{k-1} \overset{(3.10)}{=} r_k + \frac{r_k^T r_k}{\alpha_{k-1} d_{k-1}^T A d_{k-1}} d_{k-1}$$
$$\overset{(3.8)}{=} r_k + \frac{r_k^T r_k}{r_{k-1}^T d_{k-1}} d_{k-1} \, .$$

Damit haben wir das CG-Verfahren vollständig hergeleitet, welches wir als Algorithmus 1 zusammenfassen.

Bemerkung 3.25. Dieser Algorithmus wurde zuerst von Magnus R. Hestenes und Eduard Stiefel im Jahr 1952 vorgeschlagen. Theoretisch ist der Algorithmus ein direkter Löser für dünnbesetzte lineare Gleichungssystem mit Komplexität $\mathcal{O}(nm)$ (m : Anzahl der Matrixeinträge). Die Funktionalität ist in der Praxis aber schlecht: Rundungsfehler zerstören die A-Orthogonalität der Richtungen, so dass man nach n Iterationen *nicht* die Lösung hat. Zwischenzeitlich geriet das Verfahren in Vergessenheit. Heutzutage gibt es ein Revival als iterative Methode (und deswegen stoppt der Algorithmus oben nicht nach n Schritten, sondern wenn ein Abbruchkriterium erreicht ist).

Algorithm 1 CG-Verfahren

1: Initialisierung: $k = 0$, $d_0 = r_0 = b - Ax_0$
2: **while** Abbruchkriterium nicht erreicht **do**
3: $\alpha_k = \dfrac{r_k^T r_k}{d_k^T A d_k}$
4: $x_{k+1} = x_k + \alpha_k d_k$
5: $r_{k+1} = r_k - \alpha_k A d_k$
6: $d_{k+1} = r_{k+1} + \dfrac{r_{k+1}^T r_{k+1}}{r_k^T r_k} d_k$
7: $k = k + 1$
8: **end while**

Bemerkung 3.26. Man kann die Räume $\mathcal{D}_k$ auch anders charakterisieren. Es gilt nämlich

$$\begin{aligned}\mathcal{D}_k &= \operatorname{span}\{d_0, Ad_0, A^2 d_0, \ldots, A^k d_0\} \\ &= \operatorname{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^k r_0\} \ .\end{aligned}$$

Solche Räume heißen *Krylow-Räume* (nach Alexei Nikolajewitsch Krylow, 1863–1945) und die entsprechenden Verfahren sind *Krylow-Verfahren*. Es gibt neben CG noch weitere Krylow-Verfahren wie zum Beispiel biCGstab, MinRes oder GMRes, welche in Spezialvorlesungen genauer betrachtet werden.

Der teuerste Teil eines CG-Schritts ist das Matrix-Vektor-Produkt Ad_k zu berechnen. Wenn A dünnbesetzt ist, erfordert also ein CG-Schritt $\mathcal{O}(n)$ Operationen. Es gilt der folgende Satz zur Konvergenz des CG-Verfahrens.

Satz 3.27. *Sei κ die Konditionszahl der Matrix A. Dann gilt nach k Schritten des CG-Verfahrens*

$$\|x^* - x_k\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x^* - x_0\|_A$$

Der Beweis findet sich beispielsweise im Artikel „The Conjugate Gradient Method for Linear and Nonlinear Operator Equations“ von James W. Daniel, erschienen im SIAM Journal on Numerical Analysis, 4(1), 10–26.

Bemerkung 3.28 (Aufwand des CG-Verfahrens beim Poisson-Problem). Für die Konditionszahl der Poisson-Matrix $A_1 \in \mathbb{R}^{n \times n}$ mit $n = \mathcal{O}(h^{-2})$, gilt nach Lemma 2.17 $\kappa \approx \frac{4}{\pi^2 h^2}$, also folgt

$$\begin{aligned}\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} &\approx \frac{\frac{2}{\pi h} - 1}{\frac{2}{\pi h} + 1} = \frac{2 - \pi h}{2 + \pi h} = \frac{2 + \pi h - 2\pi h - \pi^2 h^2 + \pi^2 h^2}{2 + \pi h} \\ &= \frac{(1 - \pi h)(2 + \pi h) + \pi^2 h^2}{2 + \pi h} = 1 - \pi h + \frac{\pi^2 h^2}{2 + \pi h} \approx 1 - \pi h \ .\end{aligned}$$

Für eine Fehlerreduktion um den Faktor $R > 1$ in K Schritten muss also gelten

$$2(1 - \pi h)^K \leq \frac{1}{R} \ \Rightarrow \ K \geq -\frac{\ln(2R)}{\ln(1 - \pi h)} \approx \frac{1}{\pi h} \ln(2R) = \mathcal{O}(n^{1/2}) \ .$$

Hierbei haben wir wieder die Taylorentwicklung $\ln(1 + x) = x + \mathcal{O}(x^2)$ verwendet. Insgesamt ist der Aufwand also $\mathcal{O}(n^{3/2})$, was offensichtlich eine Verbesserung zu den Jacobi– und Gauss–Seidel-Verfahren darstellt.

3.4 Vorkonditionierung

Bei linearen iterativen Methoden haben wir die Idee der Vorkonditionierung eigentlich schon gesehen. Um $Ax = b$ zu lösen, haben wir nämlich statt der sog. *Richardson-Iteration*

$$x_{k+1} = x_k + b - Ax_k$$

die Iteration

$$x_{k+1} = x_k + C(b - Ax_k)$$

benutzt, wobei $C \approx A^{-1}$ sollte. Das heißt, eigentlich lösen wir das äquivalente lineare Gleichungssystem $CAx = Cb$. Je nachdem wie man C wählt, bekommt man das Jacobi– oder Gauss-Seidel-Verfahren, oder noch andere Verfahren. Je besser der *Vorkonditionierer* (engl.: *preconditioner*) C die Inverse A^{-1} approximiert, desto schneller konvergiert das Verfahren, da die Konvergenzraten der Methoden normalerweise von der Kondition der Matrix CA abhängen.

Auch beim CG-Verfahren hängt die Konvergenzrate von der Kondition der Matrix ab, vgl. Satz 3.27. Aber für das CG-Verfahren muss A symmetrisch positiv definit sein, wir können also nicht einfach A durch CA ersetzen.

Statt C betrachten wir eine untere Dreiecksmatrix L, die positive Diagonaleinträge hat. Dann gilt

$$Ax = b \quad \Leftrightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\tilde{A}} \underbrace{L^T x}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}} \quad \Leftrightarrow \quad \tilde{A}\tilde{x} = \tilde{b}\,.$$

Die Matrix $\tilde{A}$ ist offensichtlich symmetrisch und man kann auch zeigen, dass sie positiv definit ist (folgt aus dem Trägheitssatz von Sylvester, denn aus diesem folgt für $A \in \mathbb{R}^{N\times N}$ symmetrisch und $S \in \mathbb{R}^{N\times N}$ invertierbar, dass A und S^TAS mit Vielfachheit gezählt die gleiche Anzahl positiver und negativer Eigenwerte haben). Ist L der Cholesky-Faktor von A, so gilt $A = LL^T$ und es folgt $\tilde{A} = I$. Somit bietet es sich an L als Annäherung an den Cholesky-Faktor zu wählen, d.h. so dass $A \approx LL^T$ gilt.

Das CG-Verfahren für $\tilde{A}$ lautet dann: Setze $\tilde{d}_0 := \tilde{r_0} := \tilde{b} - \tilde{A}\tilde{x}_0 = L^{-1}(b - Ax_0)$, und für $k = 0, 1, 2, \ldots$ bestimme jeweils:

$$\begin{cases} \alpha_k = \dfrac{\tilde{r}_k^T\tilde{r}_k}{\tilde{d}_k^T L^{-1}AL^{-T}\tilde{d}_k}\,, \\ \tilde{x}_{k+1} = \tilde{x}_k + \alpha_k\tilde{d}_k\,, \\ \tilde{r}_{k+1} = \tilde{r}_k - \alpha_k L^{-1}AL^{-T}\tilde{d}_k\,, \\ \tilde{d}_{k+1} = \tilde{r}_{k+1} + \dfrac{\tilde{r}_{k+1}^T\tilde{r}_{k+1}}{\tilde{r}_k^T\tilde{r}_k}\tilde{d}_k\,. \end{cases}$$

Das liefert uns $\tilde{x}$, aber wir wollen eigentlich $x = L^{-T}\tilde{x}$. Mit der Notation

$$d_k := L^{-T}\tilde{d}_k\,, \quad x_k := L^{-T}\tilde{x}_k\,, \quad r_k := L\tilde{r}_k \quad \text{und} \quad W^{-1} := L^{-T}L^{-1}$$

ist obige Iterationsvorschrift äquivalent zu

$$\begin{cases} \alpha_k = \dfrac{r_k^T W^{-1} r_k}{d_k^T A d_k}, \\ x_{k+1} = x_k + \alpha_k d_k \ , \\ r_{k+1} = r_k - \alpha_k A d_k \ , \\ d_{k+1} = W^{-1} r_{k+1} + \dfrac{r_{k+1}^T W^{-1} r_{k+1}}{r_k^T W^{-1} r_k} d_k \ . \end{cases}$$

Dieses Verfahren nennt man auch *PCG-Verfahren* (engl.: *preconditioned CG*).

Wenn L eine Annäherung des Cholesky-Faktors von A ist, dann ist W^{-1} eine Annäherung von A^{-1} (genau wie C bei den linearen Verfahren eine Annäherung von A^{-1} war), aber W^{-1} ist auch *positiv definit* (da I positiv definit und symmetrisch ist, folgt dies aus dem Trägheitssatz von Sylvester, hierzu ist nur wichtig, dass L invertierbar ist). Außerdem ist es nicht nötig W^{-1} explizit zu berechnen, es genügt $W^{-1} r_k$ auswerten zu können, also $LL^T x = Wx = r_k$ zu lösen. Bleibt die Frage: Wie wählt man W^{-1}?

Es sind zwar extrem viele verschiedene Möglichkeiten vorgeschlagen worden, leider funktioniert aber nicht ein einziger Vorkonditionierer am besten für jedes lineare Gleichungssystem. Wir schauen uns beispielhaft zwei verbreitete Ansätze an.

Bemerkung 3.29 (Unvollständige Cholesky-Zerlegung als Vorkonditionierer). Selbst für eine dünnbesetzte Matrix A ist der Cholesky-Faktor L vollbesetzt. Deshalb ist das Lösen der Cholesky-Zerlegung zu teuer, und braucht außerdem (zu) viel Speicher. Stattdessen greifen wir auf die sog. *unvollständige Cholesky-Zerlegung* (engl.: *incomplete Cholesky decomposition*) zurück.

Startpunkt ist die LR-Zerlegung. Sei $A = LR$ mit einer unteren Dreiecksmatrix L, welche normiert ist (d.h. $L_{ii} \equiv 1$) und einer oberen Dreiecksmatrix R. Dann gilt für $1 \le i \le k \le n$:

$$A_{ik} = \sum_{j=1}^{n} L_{ij} R_{jk} = \sum_{j=1}^{i} L_{ij} R_{jk} = \underbrace{L_{ii}}_{=1} R_{ik} + \sum_{j=1}^{i-1} L_{ij} R_{jk} \ \Rightarrow \ R_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij} R_{jk} \ .$$

Für $1 \le k \le i - 1 \le n$ gilt:

$$A_{ik} = \sum_{j=1}^{n} L_{ij} R_{jk} = \sum_{j=1}^{k} L_{ij} R_{jk} = L_{ik} R_{kk} + \sum_{j=1}^{k-1} L_{ij} R_{jk} \ \Rightarrow \ L_{ik} = R_{kk}^{-1} \left(A_{ik} - \sum_{j=1}^{k-1} L_{ij} R_{jk} \right) .$$

Mit diesen Formeln kann man prinzipiell eine vollständige LR-Zerlegung berechnen. Um stattdessen eine unvollständige LR-Zerlegung (engl.: *incomplete LU*) zu berechnen, lassen wir alle Einträge aus, für die $A_{ij} = 0$ ist. Es ergibt sich dann Algorithmus 2.

Hiermit ergibt sich zunächst der Vorkonditionierer $W \coloneqq \tilde{L}\tilde{R}$. Falls A symmetrisch ist, kann man zeigen, dass außerdem $\tilde{R} = \tilde{D}\tilde{L}^T$ gilt, wobei $\tilde{D} \coloneqq \operatorname{diag}(\tilde{R})$. Damit erhalten wir eine *unvollständige Cholesky-Zerlegung*:

$$A \approx \tilde{L}\tilde{D}\tilde{L}^T \ ,$$

mit $\tilde{L}_{ij} = 0$ falls $A_{ij} = 0$. Der Vorkonditionierer ist dann $W \coloneqq \tilde{L}\tilde{D}\tilde{L}^T$. Es gilt häufig, dass $\kappa(W^{-1}A) \ll \kappa(A)$. Außerdem ist $\tilde{L}$ dünnbesetzt (da A dünnbesetzt) und damit ist

$$(\tilde{L}\tilde{D}\tilde{L}^T)^{-1} r_k$$

„billig“ zu berechnen.

Algorithm 2 unvollständige LR-Zerlegung

Gegeben: $A \in \mathbb{R}^{n\times n}$.
Setze $\tilde{L} = I$, $\tilde{R} = 0$.
for $i = 1, 2, \ldots, n$ **do**
  for $k = 1, 2, ..., i-1$ **do**
    if $A_{ik} \neq 0$ **then**

$$\tilde{L}_{ik} = \tilde{R}_{kk}^{-1}\left(A_{ik} - \sum_{j=1}^{k-1} \tilde{L}_{ij}\tilde{R}_{jk}\right) .$$

    end if
  end for
  for $k = i, \ldots, n$ **do**
    if $A_{ik} \neq 0$ **then**

$$\tilde{R}_{ik} = A_{ik} - \sum_{j=1}^{i-1} \tilde{L}_{ij}\tilde{R}_{jk} .$$

    end if
  end for
end for

Bemerkung 3.30 (Lineare Verfahren als Vorkonditionierer). Da sowohl W^{-1} als auch C in der Klasse der linearen Verfahren Annäherungen von A^{-1} sind, können wir $W^{-1} = C$ als Vorkonditionierer für das CG-Verfahren wählen, zumindest dann, wenn C symmetrisch positiv definit ist.

Wir können ein lineares Verfahren sogar als Vorkonditionierer einsetzten ohne die Matrix C explizit auszurechnen, da wir im PCG-Verfahren nur $W^{-1}r_k = Cr_k$ auswerten müssen. Zu Erinnerung: Für lineare Verfahren gilt die Iterationsvorschrift

$$x_{k+1} = x_k + C(b - Ax_k) .$$

Wenn wir nun einen Schritt des linearen Verfahrens für $x_0 = 0$ und $b = r_k$ ausführen, so gilt

$$x_1 = x_0 + C(r_k - Ax_0) = 0 + C(r_k - A0) = Cr_k .$$

Ein Schritt des linearen Verfahrens für $x_0 = 0$ und $b = r_k$ liefert also das für die Vorkonditionierung nötige Cr_k. In diesem Sinne können wir lineare Verfahren direkt als Vorkonditionierer verwenden. Zum Beispiel das Jacobi-Verfahren, für welches $C = D^{-1}$, wobei $D = \mathrm{diag}(A)$ gilt. Viele lineare Verfahren werden überhaupt nur betrachtet, um als Vorkonditionierer zu dienen (z.B. Mehrgitter-, Gebietszerlegungsverfahren).

Beim Gauss-Seidel-Verfahren ist $C = (D - L)^{-1}$ allerdings nicht symmetrisch. Stattdessen verwendet man das *symmetrische Gauss–Seidel-Verfahren*. Da A symmetrisch ist, haben wir die Zerlegung $A = -L + D - L^T$. Das symmetrische Gauss-Seidel-Verfahren ist dann gegeben durch:

- *Vorwärts* Iteration: $x_{k+1/2} = x_k + (D - L)^{-1}(b - Ax_k)$,
- *Rückwärts* Iteration: $x_{k+1} = x_{k+1/2} + (D - L^T)^{-1}(b - Ax_{k+1/2})$.

Einsetzen ergibt dann die Iterationsvorschrift:

$$\begin{aligned} x_{k+1} &= x_k + (D-L)^{-1}(b - Ax_k) + (D-L^T)^{-1}\Big(b - A\big(x_k + (D-L)^{-1}(b-Ax_k)\big)\Big) \\ &= x_k + \underbrace{\big((D-L)^{-1} + (D-L^T)^{-1} - (D-L^T)^{-1}A(D-L)^{-1}\big)}_{=:C}(b-Ax_k)\,. \end{aligned}$$

Die obige Matrix C kann noch weiter vereinfacht werden und der Vorkonditionierer $W^{-1} = C$ ist symmetrisch, positiv definit (falls A symmetrisch, positiv definit).

3.5 Mehrgitter-Verfahren

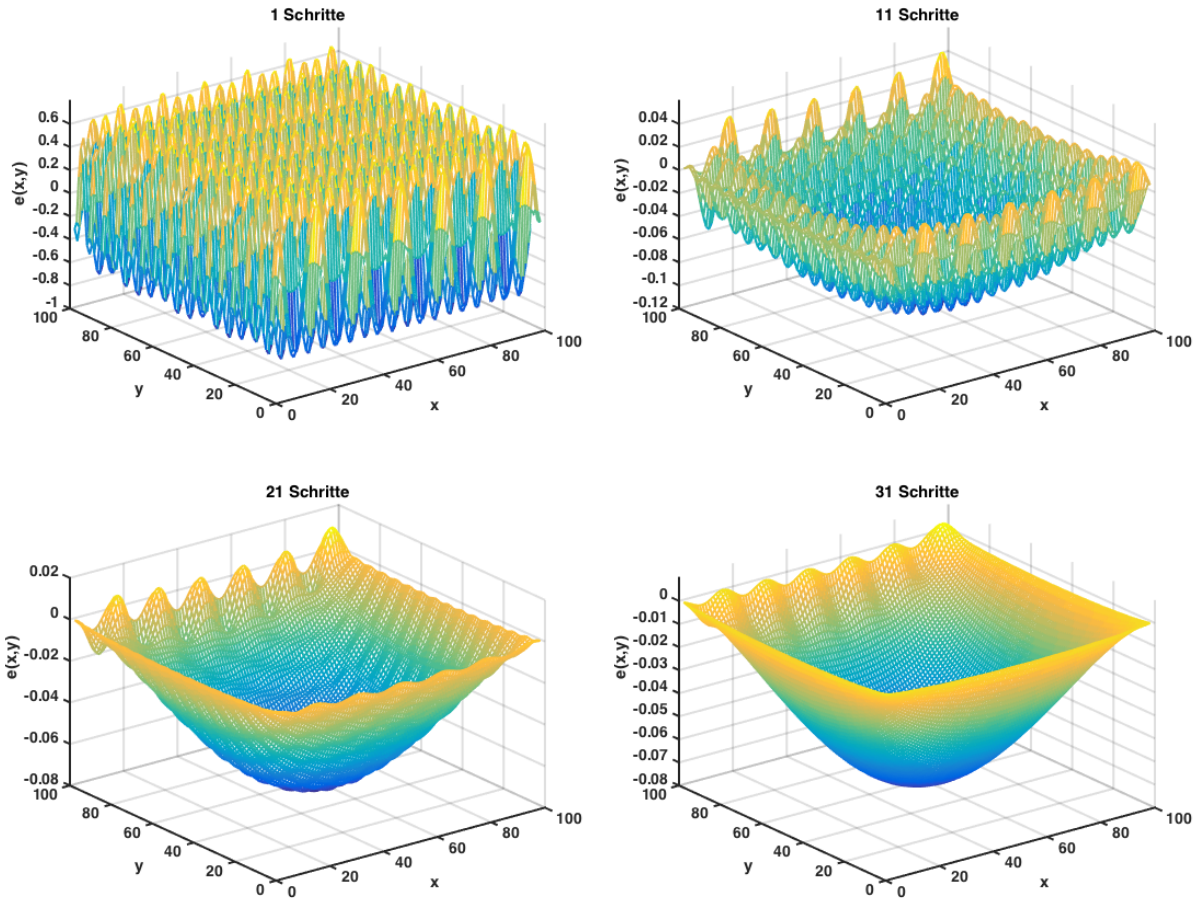


Abbildung 3.10: Fehler nach mehreren Schritten des Jacobi-Verfahrens für das Poisson-Problem.

In numerischen Experimenten (siehe beispielsweise Abbildung 3.10) mit dem Jacobi-Verfahren für das Poisson-Problem erkennt man, dass bereits wenige Iterationen reichen, um stark oszillierende Komponenten im Diskretisierungsfehler der PDE Lösung zu reduzieren. Dies ist zunächst überraschend, hat aber eine einfach einzusehende Erklärung. Wir betrachten dazu die

bekannte Matrix A_1 des diskretisierten Standard Poisson-Problems, also

$$A_1 = \frac{1}{h^2}\begin{pmatrix} T & -I & & & 0 \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ 0 & & & -I & T \end{pmatrix} \quad \text{mit} \quad T = \begin{pmatrix} 4 & -1 & & & 0 \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ 0 & & & -1 & 4 \end{pmatrix} .$$

Die Iterationsmatrix des Jacobi-Verfahrens basiert auf der Zerlegung

$$A_1 = -L + D - U \ ,$$

wobei hier die Diagonale $D = \frac{4}{h^2} I$ ein Vielfaches der Einheitsmatrix ist. Damit folgt für die Iterationsmatrix (vgl. Bemerkung 3.12), dass

$$M \coloneqq I - D^{-1} A_1 = I - \tfrac{h^2}{4} A_1 \ .$$

Die Eigenvektoren dieser Matrix M sind die Gleichen wie die von A_1. Die Eigenwerte von A_1 (vgl. Satz 2.15) sind

$$\lambda_{\nu,\mu} = \tfrac{4}{h^2}\big(\sin^2(\tfrac{1}{2}\pi\nu h) + \sin^2(\tfrac{1}{2}\pi\mu h)\big)$$

und transformieren sich in einfacher Weise, da jeder Vektor ein Eigenvektor von I zum Eigenwert 1 ist. Die Eigenwerte von M sind

$$1 - \tfrac{h^2}{4}\lambda_{\nu,\mu} = 1 - \big(\sin^2(\tfrac{1}{2}\pi\nu h) + \sin^2(\tfrac{1}{2}\pi\mu h)\big)$$

für $1 \le \nu, \mu \le n-1$. Da die Iteration linear ist, gilt für den Fehler $e_k = x_k - x^*$:

$$e_{k+1} = M e_k \ .$$

Zusammen mit der Struktur der Eigenvektoren sieht man, dass die langwelligste Komponente des Fehlers (also $\nu = \mu = 1$) am langsamsten gedämpft wird, während die kurzwelligen Komponenten Dämpfungsfaktoren deutlich kleiner als 1 haben, und somit schon in 10–20 Iterationen effektiv weg gedämpft werden.

Dies führt direkt zur Idee der *Mehrgitter-/Multigrid-Verfahren*: Wir nehmen eine Lösung nach einigen Jacobi-Iterationen, deren kurzwellige Fehler heraus gedämpft sind. Das Residuum wird auf ein grobes Finite Differenzen (FD) Gitter interpoliert, um auf diesem die langwelligen Komponenten zu dämpfen. Dies kann sogar wiederholt werden. Anschließend wird die Lösung schrittweise auf feinere Gitter extrapoliert. Dabei werden wieder einige Jacobi-Iterationen durchgeführt, um Interpolations-/Extrapolationsfehler weg zu dämpfen. Als Algorithmus ergibt sich:

1. Führe eine feste Anzahl von Iterationen der Jacobi-Methode für das Original-Gleichungssystem $Ax = b$ durch. Dies ergibt eine Näherung x_1.
2. Bestimme das Residuum $r_1 = b - Ax_1$.
3. Vergröbere das Residuum auf ein FD Gitter mit doppelter Gitterweite (also $2h$). Dies ergibt $\tilde{r}_1$.

Abbildung 3.11: V-Zyklus (links), W-Zyklus (rechts).

4. Löse das System $\tilde{A}\tilde{e}_1 = \tilde{r}_1$ mittels Jacobi-Iterationen oder exakt, wobei $\tilde{A}$ die Laplace-Matrix auf dem gröberen FD Gitter ist.

5. Der Vektor $\tilde{e}_1$ approximiert den Fehler auf dem groben FD Gitter. Interpoliere den Fehler auf das feinere FD Gitter. Dies ergibt e_1 und damit die neue Approximation $x_2 = x_1 - e_1$.

6. Führe wieder einige Jacobi-Schritte auf dem feinen FD Gitter durch.

Die grundlegende Idee dieses Algorithmus kann rekursiv und in verschiedenen Varianten durchgeführt werden, z.B. im *V-Zyklus* oder *W-Zyklus* (vgl. Abbildung 3.11). Die Wahl hängt vom Problem ab.

Bemerkung 3.31. Eine Verallgemeinerung stellt *algebraic multigrid* (AMG) dar, wo die Operatoren zur Verfeinerung, bzw. Vergröberung direkt aus der Matrix bestimmt werden, unabhängig von der Diskretisierung auf einem Gitter.

Bemerkung 3.32. Der Aufwand für den Multigrid-Algorithmus ist nahezu optimal. Für eine $n \times n$-Matrix beträgt der Aufwand für eine Jacobi-Iteration $\mathcal{O}(n)$. Wir nehmen an, dass $n = 2^J + 1$. Des Weiteren gehen wir wie folgt vor: wir vergröbern bis auf das niedrigste Level, verfeinern dann wieder, und führen jeweils ν Jacobi-Iterationen durch. Dann ist der Aufwand in etwa

$$2\nu \sum_{j=2}^{J} 2^j \approx 4\nu 2^J \approx 4\nu n = \mathcal{O}(n) \; .$$

Dies ist der Aufwand für einen V-Zyklus. Man kann zeigen dass man für das Poisson-Problem $\mathcal{O}(\log n)$ V-Zyklen benötigt werden, um zu einer vorgegebenen Genauigkeit zu kommen. Der Gesamtaufwand ist somit

$$\mathcal{O}(n \log n) \; ,$$

was nahezu optimal ist (man muss ohnehin n Unbekannte bestimmen).

Bemerkung 3.33. Es gibt verschiedene andere Mehrgitter-Methoden, die in Spezialvorlesungen diskutiert werden. Insbesondere sind auch Implementierungsfragen und Parallelisierbarkeit interessant.