Skriptum zur Vorlesung
Numerische Analysis

Helmut Gfrerer
Institut für Numerische Mathematik
Johannes Kepler Universität Linz

Wintersemester 2022/23

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Problemstellungen

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

- lineare und nichtlineare Probleme.

Es werden hier folgende typische Problemstellungen näher untersucht, die Grundbausteine für die Lösung komplizierterer Modelle sind:

- Eigenwertprobleme
- Interpolation
- Approximation und Numerische Integration (als wichtige Hilfsproblemstellungen)
- Numerische Differentiation
- Nichtlineare Gleichungen

Die Besonderheiten des wissenschaftlichen Rechnens und Algorithmen zur Lösung linearer Gleichungssysteme wurden bereits in der KV Algorithmische Methoden II behandelt.

Kapitel 2

Eigenwertprobleme

In diesem Abschnitt behandeln wir das folgende Problem.

Geg.: $A \in \mathbb{C}^{n\times n}$

Ges.: $\lambda \in \mathbb{C}$ und $x \in \mathbb{C}^n$, $x \neq 0$ mit $Ax = \lambda x$.

Eine solche Zahl λ heißt *Eigenwert* von A, der zugehörige Vektor x heißt Eigenvektor.

Solche Probleme treten in verschiedenen Gebieten der Mathematik auf. So ist z.B. die Lösung des DGL-systems $\dot{x} = Bx$, $B \in \mathbb{R}^{n\times n}$ mittels der Eigenwerte und Eigenvektoren von B beschreibar. Die Berechnung der Nullstellen von orthogonalen Polynomen (siehe später) führt auf die Berechnung der Eigenwerte einer symmetrischen und tridiagonalen Matrix. Ein dritter Anwendungsfall ist die schwingende Saite, d.h. die Analyse von Schwingungsvorgängen.

2.1 Theoretische Grundlagen

Einen ersten Ansatz zur Berechnung der Eigenwerte einer Matrix liefert das so genannte charakteristische Polynom:

Definition 2.1. *Sei $A \in \mathbb{C}^{n\times n}$. Das Polynom $p(\lambda)$, gegeben durch*

$$p(\lambda) = \det(A - \lambda I),$$

heißt das charakteristische Polynom von A.

$p(\lambda)$ ist ein Polynom vom Grade n. Es gilt:

Satz 2.1. *$\lambda \in \mathbb{C}$ ist genau dann ein Eigenwert von A, wenn $p(\lambda) = 0$.*

Beweis. Lineare Algebra. □

Die Nullstellen (oder Wurzeln) des charakteristischen Polynoms sind also genau die Eigenwerte von A. Dieser Satz kann daher als Grundlage einer numerischen Berechnung der Eigenwerte über die Nullstellen der Gleichung $p(\lambda) = 0$ dienen, ist aber wegen der numerischen Instabilitt beim Aufstellen des charakteristischen Polynoms nur bedingt geeignet.

Definition 2.2. *Zwei Matrizen A und B heißen ähnlich, wenn es eine reguläre Matrix X gibt, sodass*

$$A = XBX^{-1}.$$

Für ähnliche Matrizen gilt folgende wichtige Aussage:

Satz 2.2. *Ähnliche Matrizen haben die gleichen Eigenwerte.*

Beweis. Lineare Algebra. □

Aufbauend auf Satz 2.2 kann man eine zu A ähnliche Matrix, deren Eigenwerte leicht zu bestimmen sind, konstruieren. So sind z.B. die Diagonalelemente einer Dreiecksmatrix ihre Eigenwerte.

Dass diese Strategie zumindest theoretisch erfolgversprechend ist, zeigt folgende Aussage:

Satz 2.3 (Schur)**.** *Zu jeder Matrix $A \in \mathbb{C}^{n\times n}$ gibt es eine unitäre Matrix U mit $A = UTU^H$ wobei T eine obere Dreiecksmatrix ist. T heißt Schursche-Normalform.*

Beweis. Induktion nach n
$n = 1$: trivial
$n - 1 \to n$: Die Aussage gelte für Matrizen in $\mathbb{C}^{(n-1)\times(n-1)}$ und sei $A \in \mathbb{C}^{n\times n}$, λ_1 Eigenwert von A und x zugehöriger Eigenvektor. Wir wählen eine Householdermatrix $P(= P^H)$ so, dass $Px = \beta e_1$ (siehe Householder-Verfahren). Dann ist $PAP^H e_1 = PA\frac{x}{\beta} = \frac{\lambda_1}{\beta}Px = \lambda_1 e_1 \Rightarrow$

$$PAP^H = \left[\begin{array}{c|c} \lambda_1 & a^T \\ \hline 0 & A_1 \end{array}\right]$$

mit $a \in \mathbb{C}^{n-1}$, $A_1 \in \mathbb{C}^{(n-1)\times(n-1)}$. Nach Induktionsvoraussetzung gibt es U_1 unitär sodass

$$U_1 A_1 U_1^H = \begin{pmatrix} \lambda_2 & \cdots & \cdots \\ & \ddots & \vdots \\ 0 & & \lambda_n \end{pmatrix} \Rightarrow$$

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & U_1 \end{array}\right] PAP^H \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & U_1^H \end{array}\right] = \left[\begin{array}{c|c} \lambda_1 & a^T U_1^H \\ \hline 0 & U_1 A_1 U_1^H \end{array}\right] = \begin{pmatrix} \lambda_1 & \cdots & \cdots \\ & \ddots & \vdots \\ 0 & & \lambda_n \end{pmatrix}$$

und die Behauptung folgt mit $U^H = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & U_1 \end{array}\right) P$, das als Produkt zweier unitärer Matrizen wieder unitär ist. □

Bemerkung: Es gilt

$$T = \begin{pmatrix} \lambda_1 & * & \cdots & * \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix},$$

d.h. die Eigenwerte λ_i stehen auf der Hauptdiagonale.

Der folgende Satz liefert eine Aussage über die ungefähre Lage der Eigenwerte:

Satz 2.4 (Gerschgorin)**.** *Die Vereinigung der Kreisscheiben*

$$K_i = \{\mu \in \mathbb{C} : |\mu - a_{ii}| \leq \sum_{j\neq i} |a_{ij}|\}$$

enthält alle Eigenwerte der Matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$.

Beweis. Sei λ ein Eigenwert von A mit Eigenvektor $x = (x_i)_{i=1}^n$. Dabei sei x so normiert, daß $\| x \|_\infty = 1$. Sei i, $1 \le i \le n$, so gewählt, daß $|x_i| = 1$. Dann haben wir

$$\sum_{j=1}^n a_{ij}x_j = \lambda x_i \quad \Leftrightarrow \quad \sum_{j=1, j\neq i}^n a_{ij}x_j = (\lambda - a_{ii})x_i.$$

Wegen $|x_i| = 1$ und $|x_j| \le 1$ folgt nun

$$|\lambda - a_{ii}| \le \left| \sum_{j=1, j\neq i}^n a_{ij}x_j \right| \le \sum_{j\neq i} |a_{ij}||x_j| \le \sum_{j\neq i} |a_{ij}|,$$

d.h. $\lambda \in K_i$. □

Eine wesentliche Teilklasse von Matrizen sind diagonalisierbare Matrizen.

Definition 2.3. *Sei $A \in \mathbb{C}^{n\times n}$. A heißt diagonalisierbar, falls es eine reguläre Matrix $X = [x_1, \ldots, x_n]$ und eine Diagonalmatrix D gibt mit $A = XDX^{-1}$, wobei*

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}.$$

Bemerkung: Dann sind die λ_i offenbar die Eigenwerte, wobei x_i der zugehörige Eigenvektor ist.

Im Falle diagonalisierbarer Matrizen kann man den folgenden Satz über Datenstörungen zeigen.

Satz 2.5. *Sei $A \in \mathbb{C}^{n\times n}$ eine diagonalisierbare Matrix mit Eigenvektormatrix X, d.h. $A = XDX^{-1}$. Weiters sei $\Delta A \in \mathbb{C}^{n\times n}$. Dann gibt es zu jedem Eigenwert $\bar\lambda$ von $\bar A = A + \Delta A$ einen Eigenwert λ von A mit*

$$|\bar\lambda - \lambda| \le \kappa_2(X) \cdot \|\Delta A\|_2.$$

Beweis. Sei $\bar\lambda$ ein Eigenwert von $\bar A = A + \Delta A$ und nicht Eigenwert von A. Dann ist $A + \Delta A - \bar\lambda I$ singulär. Nun ist auch

$$B = X^{-1}(A + \Delta A - \bar\lambda\, I)X = D - \bar\lambda\, I + X^{-1}\Delta AX$$

singulär. Damit ist auch die Matrix

$$(D - \bar\lambda I)^{-1}B = I + (D - \bar\lambda I)^{-1}X^{-1}\Delta AX := I + C$$

singulär, falls $\bar\lambda$ kein Eigenwert von A ist, d.h. es gibt einen Vektor $v \neq 0$ mit $-Cv = v$. Damit haben wir

$$\frac{(Cv, Cv)}{(v, v)} = 1 \quad \Rightarrow \quad \|C\|_2 \ge 1.$$

Damit gilt

$$\begin{aligned} 1 &\le \|C\|_2 = \|(D - \bar\lambda\, I)^{-1}X^{-1}\Delta AX\|_2 \le \|(D - \bar\lambda\, I)^{-1}\|_2 \|X^{-1}\|_2 \|\Delta A\|_2 \|X\|_2 \\ &= \kappa_2(X)\, \|\Delta A\|_2 \max_{\lambda\in\sigma(A)} \frac{1}{|\bar\lambda - \lambda|}, \end{aligned}$$

also $|\bar\lambda - \lambda| \le \kappa_2(X)\, \|\Delta A\|_2$ für ein $\lambda \in \sigma(A)$. □

Damit ist die Konditionszahl der Eigenvektormatrix der Verstärkungsfaktor für Störungen. Ein besonders günstiger Fall liegt vor, wenn X eine unitäre Matrix ist, denn dann gilt $\kappa_2(X) = 1$ und $|\lambda_{max}(A)| = \rho(A) = \|D\|_2 = \|A\|_2$. Somit folgt

$$\varepsilon_\lambda \leq \varepsilon_A$$

mit $\varepsilon_\lambda = |\bar{\lambda} - \lambda|/|\lambda_{max}(A)|$ und $\varepsilon_A = \|\Delta A\|_2/\|A\|_2$. Das Eigenwertproblem ist in diesem Fall also gut konditioniert.

Jene Matrizen A, für die X unitär gewählt werden kann, die also durch eine Ähnlichkeitstransformation mit einer unitären Matrix diagonalisiert werden können, sind die normalen Matrizen.

Definition 2.4. *Eine Matrix $A \in \mathbb{C}^{n\times n}$ heißt normal, falls $A^H A = AA^H$. Sie heißt hermitesch, falls $A = A^H$.*

Jede hermitesche Matrix ist also normal.

Diese Betrachtungen über die Stabilität gelten jedoch nur für diagonalisierbare Matrizen. Für nicht diagonalisierbare Matrizen gelten lediglich Abschätzungen der Form

$$|\bar{\lambda} - \lambda| = \mathcal{O}(\|\Delta A\|^{\frac{1}{\nu}})$$

wobei ν die höchste Ordnung eines zu λ gehörenden Elemtarteilers ist (ohne Beweis). Damit ist die Berechnung eines Eigenwertes unter Umständen instabil gegenüber Datenfehlern, falls die algebraische Vielfachheit sehr stark von der geometrischen Vielfachheit abweicht.

2.2 Berechnung einzelner Eigenwerte und Eigenvektoren

Einzelne Eigenwerte lassen sich als Nullstellen des charakteristischen Polynoms berechnen, z.B. mit Hilfe des Newton–Verfahrens. Allerdings ist zu beachten, dass die Aufstellung des charakteristischen Polynoms und seiner Ableitung über die Berechnung der Koeffizienten des Polynoms und die anschließende Berechnung der Nullstellen im Allgemeinen kein numerisch stabiler Algorithmus ist. Besser geeignet sind spezielle Rekursionsformeln zur Berechnung von $p(\lambda)$ bzw. $p'(\lambda)$ für einen gegebenen Wert von λ.

So lässt sich das charakteristische Polynom an einer Stelle λ für symmetrische reelle Tridiagonalmatrizen

$$A = \begin{pmatrix} b_1 & a_2 & & \\ a_2 & b_2 & \ddots & \\ & \ddots & \ddots & a_n \\ & & a_n & b_n \end{pmatrix}$$

durch folgende Rekursion berechnen: $p(\lambda) = p_n(\lambda)$ mit

$$\begin{aligned} p_0(\lambda) &= 1 \\ p_1(\lambda) &= b_1 - \lambda \\ p_k(\lambda) &= (b_k - \lambda)\, p_{k-1}(\lambda) - a_k^2\, p_{k-2}(\lambda), \quad k = 2, \ldots, n. \end{aligned}$$

Durch Ableiten der Rekursion für $p(\lambda)$ erhält man eine entsprechende Rekursion für $p'(\lambda)$.

Geeignete Startwerte für das Newton–Verfahren erhält man aus dem Satz von Gerschgorin.

2.2.1 Vektoriteration

Unter der Vektoriteration versteht man das folgende Verfahren. Sei $v^{(0)}$ ein Startvektor mit $\|v^{(0)}\|_2 = 1$. Dann berechnen wir eine Folge von Vektoren $v^{(k)}$ sowie eine Zahlenfolge $\mu^{(k)}$ mittels der Rekursion

$$\begin{aligned} z^{(k+1)} &= Av^{(k)}, \\ \mu^{(k)} &= {v^{(k)}}^H z^{(k+1)}, \\ v^{(k+1)} &= \frac{z^{(k+1)}}{\|z^{(k+1)}\|_2}. \end{aligned}$$

Wir nehmen nun an, daß die Matrix A diagonalisierbar ist, d.h.

$$A = XDX^{-1}$$

mit der Diagonalmatrix $D = \mathrm{diag}(\lambda_i)_{i=1}^n$ und der Eigenvektormatrix $X = [x_1\, x_2\, \ldots x_n]$. O.B.d.A. seien die Eigenwerte λ_i betragsmäßig absteigend angeordnet. Unter diesen Voraussetzungen kann man den folgenden Satz zeigen.

Satz 2.6. *Es sei $A \in \mathbb{C}^{n\times n}$ eine diagonalisierbare Matrix mit den betragsmäßig absteigend sortierten Eigenpaaren (λ_i, x_i). Sei weiters $\lambda_1 = \ldots = \lambda_r$ für $1 \le r \le n$ und zusätzlich $q := \frac{|\lambda_{r+1}|}{|\lambda_r|} < 1$, falls $r < n$ bzw. $\lambda_1 \neq 0$, falls $r = n$ ($q := 0$). Weiters sei $e := \sum_{i=1}^r \alpha_i x_i \neq 0$, wobei*

$$v^{(0)} = \sum_{i=1}^n \alpha_i x_i.$$

Dann konvergiert die Folge $\mu^{(k)}$ gegen λ_1 und für jeden Index ν, für den $e_\nu \neq 0$, ist die Folge $v^{(k)}/v_\nu^{(k)}$ für hinreichend großes k wohldefiniert und konvergiert gegen den zu λ_1 gehörenden Eigenvektor e/e_ν. Es gelten die Fehlerabschätzungen

$$\begin{aligned} |\mu^{(k)} - \lambda_1| &\le C_1 q^k \quad (\textit{bzw. } |\mu^{(k)} - \lambda_1| \le C_2 q^{2k}, \textit{ falls } A \textit{ normal}) \\ \|\frac{v^{(k)}}{v_\nu^{(k)}} - \frac{e}{e_\nu}\| &\le C_3 q^k. \end{aligned}$$

Beweis. Wir betrachten nur den Fall $r < n$. Der Einfachheit halber sei , $\|x_i\|_2 = 1$, $i = 1, \ldots, n$. Wie man sich leicht überlegt, gilt $v^{(k)} = w^{(k)}/\|w^{(k)}\|_2$ mit

$$w^{(k)} := A^k v^{(0)} = \sum_{i=1}^n \lambda_i^k \alpha_i x_i = \lambda_1^k (\sum_{i=1}^r \alpha_i x_i + \sum_{i=r+1}^n \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i) = \lambda_1^k (e + y^{(k)}) \tag{2.1}$$

mit $y^{(k)} := \sum_{i=r+1}^n \alpha_i (\lambda_i/\lambda_1)^k x_i$. Weiters ist $z^{(k+1)} = Av^{(k)} = Aw^{(k)}/\|w^{(k)}\|_2 = w^{(k+1)}/\|w^{(k)}\|_2$. Damit folgt

$${v^{(k)}}^H z^{(k+1)} = \frac{{w^{(k)}}^H w^{(k+1)}}{\|w^{(k)}\|_2^2} = \frac{\bar\lambda_1^k (e + y^{(k)})^H \lambda_1^{k+1} (e + y^{(k+1)})}{|\lambda_1|^{2k} (e + y^{(k)})^H (e + y^{(k)})} = \lambda_1 \frac{(e + y^{(k)})^H (e + y^{(k+1)})}{(e + y^{(k)})^H (e + y^{(k)})}$$

und daher

$$\mu^{(k)} - \lambda_1 = \lambda_1 \left(\frac{(e + y^{(k)})^H (e + y^{(k+1)})}{(e + y^{(k)})^H (e + y^{(k)})} - 1 \right) = \lambda_1 \frac{(e + y^{(k)})^H (y^{(k+1)} - y^{(k)})}{(e + y^{(k)})^H (e + y^{(k)})}$$

Wegen $\|y^{(k)}\|_2 \le q^k \sum_{i=r+1}^n |\alpha_i| \to 0$ folgt mit Hilfe der Cauchy-Schwarzschen Ungleichung

$$\begin{aligned} |\mu^{(k)} - \lambda_1| &\le |\lambda_1| \frac{\|e + y^{(k)}\|_2 \|y^{(k+1)} - y^{(k)}\|_2}{\|e + y^{(k)}\|_2^2} \le |\lambda_1| \frac{\|y^{(k+1)}\|_2 + \|y^{(k)}\|_2}{\|e + y^{(k)}\|_2} \\ &= \frac{|\lambda_1|(1+q)\sum_{i=r+1}^n |\alpha_i|}{\|e + y^{(k)}\|_2} q^k \\ &\le C_1 q^k \end{aligned}$$

Im Fall dass A normal ist, ist e als Linearkombination von $x_1, \ldots, x_r$ orthogonal auf $x_{r+1}, \ldots, x_n$ und daher $e^H(y^{(k+1)} - y^{(k)}) = 0$. Damit ergibt sich

$$\begin{aligned} |\mu^{(k)} - \lambda_1| &\le |\lambda_1| \frac{|{y^{(k)}}^H (y^{(k+1)} - y^{(k)})|}{\|e + y^{(k)}\|_2^2} \le |\lambda_1| \frac{\|y^{(k)}\|_2 \|y^{(k+1)} - y^{(k)}\|_2}{\|e + y^{(k)}\|_2^2} \\ &\le \frac{|\lambda_1|(1+q)(\sum_{i=r+1}^n |\alpha_i|)^2}{\|e + y^{(k)}\|_2^2} q^{2k} \\ &\le C_2 q^{2k} \end{aligned}$$

Für die letzte Fehlerabschätzung beachten wir, dass aus (2.1) die Beziehung $w^{(k)}/\lambda_1^k = e + y^{(k)}$ und damit $w_\nu^{(k)} \neq 0$ für hinreichend großes k folgt. Wegen $v^{(k)} = w^{(k)}/\|w^{(k)}\|$ ist daher auch $v_\nu^{(k)} \neq 0$ und $v^{(k)}/v_\nu^{(k)}$ wohl definiert. $v^{(k)}/v_\nu^{(k)}$ ist aber genau jenes Vielfache des Vektors $v^{(k)}$, bei dem die ν-te Komponente gleich 1 ist, da aber auch $w^{(k)}$ bzw $e + y^{(k)}$ ein Vielfaches von $v^{(k)}$ ist, gilt $v^{(k)}/v_\nu^{(k)} = (e + y^{(k)})/(e + y^{(k)})_\nu$ und daher

$$\left\| \frac{v^{(k)}}{v_\nu^{(k)}} - \frac{e}{e_\nu} \right\| = \left\| \frac{e + y^{(k)}}{(e + y^{(k)})_\nu} - \frac{e}{e_\nu} \right\| = \frac{\|y^{(k)} e_\nu - y_\nu^{(k)} e\|}{|(e + y^{(k)})_\nu e_\nu|} \le C_3 q^k.$$

Das beweist die Behauptung. □

Die wesentliche Voraussetzung in diesem Satz ist nicht die Diagonalisierbarkeit, sondern dass die betragsmäßig größten Eigenwerte alle gleich sind. Z.B. kann eine reelle Matrix A ja konjugiert komplexe Eigenwerte besitzen, wird die Vektoriteration mit einem rellen Vektor $v^{(0)}$ gestartet, sind natürlich alle $\mu^{(k)}$ ebenfalls reell und können daher nicht zu einer komplexen Zahl konvergieren.

Die Normierung über den Index ν dient nur dazu, eine konvergente Folge zu erzielen. Vernachlässigt man die Forderung der Konvergenz, so kann man sich auch mit der Folge $(v^{(k)})$ begnügen: Diese Folge konvergiert zwar nicht unbedingt, allerdings nähern sich die Folgenelemente der Menge aller Eigenvektoren zum Eigenwert λ_1 mit Norm 1 immer genauer an (Fehlerabschätzung: Cq^k)

A priori kennt man keinen Index ν, für den $e_\nu \neq 0$. Deshalb kann man versuchen, eine Folge $(\nu^{(k)})$ zu konstruieren, die gegen so einen Index konvergiert, d.h. ab einem hinreichend großen k konstant ist. Eine naheliegende Wahl wäre durch $|v_{\nu^{(k)}}^{(k)}| = \max_{i=1,\ldots,n} |v_i^{(k)}|$ gegeben, dies muss aber nicht unbedingt funktionieren und man wählt daher

$$\begin{aligned} \nu^{(0)} &\in \{j \,|\, |v_j^{(0)}| = \max_{i=1,\ldots,n} |v_i^{(0)}|\} \\ \nu^{(k)} &\in \begin{cases} \{\nu^{(k-1)}\} & \text{falls } |v_{\nu^{(k-1)}}^{(k)}| \ge 0.99 \max_{i=1,\ldots,n} |v_i^{(k)}| \\ \{j \,|\, |v_j^{(k)}| = \max_{i=1,\ldots,n} |v_i^{(k)}|\} & \text{sonst} \end{cases} \end{aligned}$$

Bemerkung: Die Näherung für den Eigenwert ist genau der sogenannte Rayleigh–Quotient. Es gilt:

$$\mu^{(k)} = \frac{(Av^{(k)}, v^{(k)})_2}{(v^{(k)}, v^{(k)})_2}.$$

2.2.2 Inverse Vektoriteration (Wielandt)

Wie eben gezeigt, "konvergiert" die Vektoriteration gegen einen Eigenvektor zum betragsgrößten Eigenwert λ_1 der Matrix A. Falls man einen Eigenvektor zum betragskleinsten Eigenwert λ_n benötigt, wendet man das eben beschriebene Verfahren auf die Matrix A^{-1} an. Die Matrix A^{-1} besitzt nun die Eigenwerte $\frac{1}{\lambda_i}$, wobei $\frac{1}{\lambda_n}$ der betragsgrößte und $\frac{1}{\lambda_1}$ der betragskleinste Eigenwert ist. Nach Satz 2.6 angewandt auf A^{-1}, konvergiert nun das Verfahren

$$\begin{aligned} \text{Löse} \quad Az^{(k+1)} &= v^{(k)}, \\ \text{Berechne} \quad \mu^{(k)} &= {v^{(k)}}^H z^{(k+1)}, \\ \text{Setze} \quad v^{(k+1)} &= \frac{z^{(k+1)}}{\|z^{(k+1)}\|_2} \end{aligned}$$

gegen den Kehrwert des betragskleinsten Eigenwert λ_n bzw. gegen einen zugehörigen Eigenvektor. Der Konvergenzfaktor beträgt nun

$$q = \left| \frac{\lambda_n}{\lambda_{n-1}} \right|.$$

Wir wenden nun die Vektoriteration auf die Matrix $(A - \bar{\lambda}I)^{-1}$ an, wobei $\bar{\lambda}$ eine Näherung eines Eigenwertes λ_j ist. Ausgehend von einem Startvektor $v^{(0)}$, berechnet man also eine Folge von Vektoren mit

$$\begin{aligned} \text{Löse} \quad (A - \bar{\lambda}I)z^{(k+1)} &= v^{(k)}, \\ \text{Berechne} \quad \mu^{(k)} &= {v^{(k)}}^H z^{(k+1)}, \\ \text{Setze} \quad v^{(k+1)} &= \frac{z^{(k+1)}}{\|z^{(k+1)}\|_2}. \end{aligned}$$

Die Matrix $(A - \bar{\lambda}I)^{-1}$ besitzt die Eigenpaare $\left(\frac{1}{\lambda_i - \bar{\lambda}}, x_i\right)$. Sei $1 \leq j \leq n$, $j \in \mathbb{N}$ so gewählt, daß

$$|\lambda_j - \bar{\lambda}| \leq |\lambda_i - \bar{\lambda}|, \quad \forall i = 1, \ldots, n,$$

d.h. $|\lambda_j - \bar{\lambda}|$ ist der betragskleinste Eigenwert von $A - \bar{\lambda}I$. Basierend auf der obigen Argumentation konvergiert das Verfahren mit dem Konvergenzfaktor

$$q = \max_{\substack{i=1,\ldots,n \\ i \neq j}} \left| \frac{\lambda_j - \bar{\lambda}}{\lambda_i - \bar{\lambda}} \right|,$$

d.h. die Konvergenz ist um so besser, je besser die Näherung $\bar{\lambda}$ für den Eigenwert λ_j ist.

2.3 Der QR-Algorithmus

Das Ziel des QR-Algorithmus ist die Konstruktion einer Folge von Matrizen $A^{(k)}$, die alle zu A ähnlich sind und in gewisser Weise gegen eine Schur-Normalform von A konvergieren.

Die Basisversion des Algorithmus besteht aus folgender Iteration:

QR-Algorithmus:

$A^{(1)} = A$

Für $k = 1, 2, \ldots$ berechne:

1. $Q_k R_k = A^{(k)}$ (QR-Zerlegung)
2. $A^{(k+1)} = R_k Q_k$

Der erste Teilschritt besteht aus einer QR-Zerlegung von $A^{(k)}$, z.B. mit dem Householder-Verfahren Im zweiten Teilschritt werden die Faktoren der QR-Zerlegung in umgekehrter Reihenfolge miteinander multipliziert, um die nächste Iterierte $A^{(k+1)}$ zu erhalten.

Satz 2.7. *Sei $A^{(k)}$ mittels QR-Algorithmus berechnet. Dann sind alle $A^{(k)}$ zueinander ähnlich.*

Beweis. Es gilt

$$A^{(k+1)} \quad = \quad R_k Q_k = Q_k^H Q_k R_k Q_k = Q_k^H A^{(k)} Q_k$$

Da Q_k unitär ist, folgt mit $Q_k^{-1} = Q_k^H$ die Behauptung. □

Es läßt sich nun zeigen, daß die so konstruierte Folge gegen die Schursche Normalform konvergiert.

Satz 2.8. *Sei $A \in \mathbb{C}^{n \times n}$ eine Matrix mit betragsmäßig verschiedenen Eigenwerten, d.h.*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|.$$

Weiterhin sei X die Matrix der Eigenvektoren (betragsmäßig absteigend angeordnet) und $Y = X^{-1}$. Außerdem besitze Y eine Dreieckszerlegung: $Y = L_Y \cdot R_Y$. Dann gilt für die Folge der Matrizen $A^{(k)} = (a_{ij}^{(k)})$, die durch den QR-Algorithmus erzeugt werden:

1. $\lim_{k\to\infty} a_{ij}^{(k)} = 0$ *für alle i, j mit $i > j$ bzw. genauer* $a_{ij}^{(k)} = \mathcal{O}(|\frac{\lambda_i}{\lambda_j}|^k)$.

2. $\lim_{k\to\infty} a_{ii}^{(k)} = \lambda_i$ *für $i = 1, 2, \ldots, n$.*

Beweis. Aus der Konstruktion ergibt sich

$$A^{(k+1)} = Q_k^H A^{(k)} Q_k = Q_k^H Q_{k-1}^H A^{(k-1)} Q_{k-1} Q_k = \ldots = (Q_1 \ldots Q_k)^H A^{(1)} (Q_1 \ldots Q_k) = \hat{Q}_k^H A \hat{Q}_k \tag{2.2}$$

mit $\hat{Q}_k := Q_1 \ldots Q_k$ (unitär) und daher $A\hat{Q}_k = \hat{Q}_k A^{(k+1)}$. Wir zeigen nun mit vollständiger Induktion dass

$$A^k = \hat{Q}_k \hat{R}_k \forall k, \tag{2.3}$$

wobei $\hat{R}_k := R_k \ldots R_1$. Dies ist sicherlich richtig für $k = 1$. Sei nun die Aussage richtig für k und wir wollen zeigen dass sie auch für $k+1$ gilt. Wegen $\hat{Q}_{k+1} = \hat{Q}_k Q_{k+1}$ und $\hat{R}_{k+1} = R_{k+1}\hat{R}_k$ folgt

$$A^{k+1} = AA^k = A\hat{Q}_k\hat{R}_k = \hat{Q}_k A^{(k+1)} \hat{R}_k = \hat{Q}_k Q_{k+1} R_{k+1} \hat{R}_k = \hat{Q}_{k+1}\hat{R}_{k+1}$$

und damit ist die Behauptung gezeigt.

Weiters gilt auch für alle k

$$A^k = XD^kX^{-1} = XD^kL_YR_Y \tag{2.4}$$

mit $D = \mathrm{diag}(\lambda_i)$. Wir setzen nun $L^{(k)} := D^k L_Y D^{-k}$ und behaupten

$$\lim_{k\to\infty} L^{(k)} = I. \tag{2.5}$$

Tatsächlich gilt $L_{ii}^{(k)} = (L_Y)_{ii} = 1$ für $i = 1, \ldots, n$ und

$$\lim_{k\to\infty} L_{ij}^{(k)} = \lim_{k\to\infty} \frac{\lambda_i^k}{\lambda_j^k}(L_Y)_{ij} = \lim_{k\to\infty} \left(\frac{\lambda_i}{\lambda_j}\right)^k (L_Y)_{ij} = 0, i > j.$$

Für die Eigenvektormatrix X berechnen wir nun eine QR-Zerlegung $X = Q_X R_X$ und anschließend QR-Zerlegungen $R_X L^{(k)} = Q_L^{(k)} R_L^{(k)}\ \forall k$. Für jedes k definieren wir uns eine unitäre Diagonalmatrix Σ^k durch

$$\Sigma_{ii}^{(k)} = \begin{cases} \overline{(Q_L^k)_{ii}}/|(Q_L^k)_{ii}| & \text{falls } (Q_L^k)_{ii} \neq 0 \\ 1 & \text{sonst} \end{cases}$$

und setzen $\hat{Q}_L^{(k)} = Q_L^{(k)}\Sigma^{(k)}$, $\hat{R}_L^{(k)} = {\Sigma^{(k)}}^H R_L^{(k)}$. Offensichtlich ist $\hat{Q}_L^{(k)}$ unitär, $\hat{R}_L^{(k)}$ eine rechte obere Dreiecksmatrix und $\hat{Q}_L^{(k)}\hat{R}_L^{(k)} = Q_L^{(k)}\Sigma^{(k)}{\Sigma^{(k)}}^H R_L^{(k)} = Q_L^{(k)} R_L^{(k)}$. Weiters gilt für die Diagonalelemente $(\hat{Q}_L^{(k)})_{ii} = |(Q_L^k)_{ii}| \in \mathbb{R}_+$. Wir zeigen als nächstes $\lim_{k\to\infty} \hat{Q}_L^{(k)} = I$. Sei dazu $\tilde{Q}$ ein beliebiger Häufungspunkt der Folge $\left(\hat{Q}_L^{(k)}\right)$. Dann ist sicherlich $\tilde{Q}$ orthogonal und $\tilde{Q}_{ii} \in \mathbb{R}_+$, $i = 1, \ldots, n$. Aus

$$\hat{Q}_L^{(k)} = R_X L^{(k)} \left(R_L^{(k)}\right)^{-1},$$

(2.5) und

$$\limsup_{k\to\infty} \|\left(R_L^{(k)}\right)^{-1}\| = \limsup_{k\to\infty} \|\hat{Q}_L^{(k)} R_X^{-1} \left(L^{(k)}\right)^{-1}\| \leq \limsup_{k\to\infty} \|\hat{Q}_L^{(k)}\|\|R_X^{-1}\|\|\left(L^{(k)}\right)^{-1}\| = \|R_X^{-1}\|$$

folgt, dass $\tilde{Q}$ eine rechte obere Dreiecksmatrix ist. Damit ist aber $\tilde{Q}$ eine Diagonalmatrix deren Diagonalelemente Betrag 1 haben und da die Diagonalelemente nichtnegative reelle Zahlen sind, folgt $\tilde{Q} = I$. Damit ist die Einheitsmatrix der einzige Häufungspunkt der beschränkten Folge $\left(\hat{Q}_L^{(k)}\right)$, was gleichbedeutend mit unserer Behauptung $\lim_{k\to\infty} \hat{Q}_L^{(k)} = I$ ist.

Aus (2.3) und (2.4) folgt nun

$$A^k = \hat{Q}_k \hat{R}_k = X D^k L_Y R_Y = Q_X R_X L^{(k)} D^k R_Y = Q_X \hat{Q}_L^{(k)} \hat{R}_L^{(k)} D^k R_Y$$

und damit

$$\left(\hat{Q}_L^{(k)}\right)^H Q_X^H \hat{Q}_k = \hat{R}_L^{(k)} D^k R_Y \hat{R}_k^{-1}.$$

Also ist $\left(\hat{Q}_L^{(k)}\right)^H Q_X^H \hat{Q}_k$ gleichzeitig eine unitäre Matrix und eine rechte obere Dreiecksmatrix und damit gleich einer Diagonalmatrix $\hat{\Sigma}^{(k)}$ deren Diagonalelemente Betrag 1 haben. Daraus folgt

$$\lim_{k\to\infty} \left(\hat{Q}_k \hat{\Sigma}^{(k)H}\right) = \lim_{k\to\infty} \left(Q_X \hat{Q}_L^{(k)}\right) = Q_X$$

und damit mit (2.2)

$$\lim_{k\to\infty} \hat{\Sigma}^{(k)} A^{(k+1)} \hat{\Sigma}^{(k)H} = \lim_{k\to\infty} \left(\left(\hat{Q}_k \hat{\Sigma}^{(k)}\right)^H A \left(\hat{Q}_k \hat{\Sigma}^{(k)H}\right)\right) = Q_X^H X D X^{-1} Q_X = R_X D R_X^{-1}.$$

$R_X D R_X^{-1}$ ist aber eine rechte obere Dreiecksmatrix deren Diagonale genau die Elemente von D sind. Weiters sind die Elemente von $\hat{\Sigma}^{(k)} A^{(k+1)} \hat{\Sigma}^{(k)H}$ betragsmäßig gleich denen von $A^{(k+1)}$ und die Diagonalen der beiden Matrizen sind identisch. Daraus ergibt sich die Behauptung. □

Bemerkung:

1. Die Konvergenz des QR-Algorithmus folgt auch ohne die Voraussetzung an Y, allerdings sind dann die Eigenwerte als Grenzwerte der Diagonalelemente von $A^{(k)}$ nicht mehr der Größe nach angeordnet.

2. Die Nebendiagonalelemente $a_{ij}^{(k)}$ mit $i < j$ müssen nicht unbedingt konvergieren.

3. In vielen Fällen sind die Eigenwerte betragsmäßig nicht getrennt, z.B. kann einen reelle nichtsymmetrische Matrix konjugiert komplexe Eigenwerte haben. Sei nun $|\lambda_1| > \ldots > |\lambda_r| = |\lambda_{r+1}| > \ldots > |\lambda_n|$. Ist A diagonalisierbar und sind die restlichen Voraussetzungen von Satz 2.8 erfüllt, so kann gezeigt werden:

   (a) $\lim_{k\to\infty} a_{ij}^{(k)} = 0$, $a_{ij}^{(k)} = \mathcal{O}(|\frac{\lambda_i}{\lambda_j}|^k)$, $\forall i > j$, $(i,j) \neq (r+1, r)$

   (b) $\lim_{k\to\infty} a_{ii}^{(k)} = \lambda_i$, $i \neq r, r+1$

   (c) Die Eigenwerte der Untermatrizen

   $$\begin{pmatrix} a_{rr}^{(k)} & a_{r,r+1}^{(k)} \\ a_{r+1,r}^{(k)} & a_{r+1,r+1}^{(k)} \end{pmatrix}$$

   konvergieren gegen λ_r und λ_{r+1} (obwohl die Untermatrizen im allgemeinen divergieren!).

Die Konvergenz des QR-Verfahrens ist jedoch relativ langsam, falls die Eigenwerte dicht beeinander liegen, d.h. es werden relativ viele Iterationen benötigt. Desweiteren erfordert die Berechnung der QR-Zerlegung pro Iterationsschritt $\mathcal{O}(n^3)$ Operationen. Um den Rechenaufwand zu verringern, sollte die Matrix zunächst erst einmal auf einfachere Gestalt gebracht werden.

2.3.1 Reduktion auf einfache Gestalt

Definition 2.5. *Eine Matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ heißt **(rechte obere) Hessenberg-Matrix**, genau dann wenn $a_{ij} = 0$ für $i > j + 1$, d.h.*

$$A = \begin{pmatrix} * & \cdots & \cdots & \cdots & * \\ * & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & * \\ 0 & \cdots & 0 & * & * \end{pmatrix}$$

Satz 2.9. *Zu jeder Matrix $A \in \mathbb{C}^{n\times n}$ existiert eine unitäre Matrix U mit $U^H AU = H$, wobei H eine Hessenberg-Matrix ist.*

Beweis. Der Beweis erfolgt konstruktiv. Sei

$$A = \begin{bmatrix} a_{11} & b_1^T \\ a_1 & A_1 \end{bmatrix}, \quad a_1, b_1 \in \mathbb{C}^{n-1}.$$

Mittels Householder-Verfahren wähle man eine unitäre Matrix $P_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ mit

$$P_1 a_1 = k e_1,$$

wobei e_1 der erste Einheitsvektor ist. Dann ist die Matrix

$$U_1 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & P_1^H \end{bmatrix}$$

unitär und

$$\begin{aligned} U_1^H A U_1 &= \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & P_1 \end{bmatrix} \begin{bmatrix} a_{11} & b_1^T \\ a_1 & A_1 \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & P_1^H \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & b_1^T P_1^H \\ k e_1 & P_1 A_1 P_1^H \end{bmatrix}. \end{aligned}$$

Wir wenden den Algorithmus auf $\tilde{A}_1 = P_1 A_1 P_1^H$ wieder an und nach insgesamt $n-2$ Schritten erhält man eine Hessenberg-Matrix H mit

$$U_{n-2}^H \cdots U_2^H U_1^H A U_1 U_2 \cdots U_{n-2} = H.$$

Mit $U = U_1 U_2 \cdots U_{n-2}$ folgt dann die Behauptung. □

Bemerkung: Es ist leicht nachzuweisen, daß die durch den QR-Algorithmus erzeugten Matrizen wieder Hessenberg sind. Eine orthogonale Faktorisierung von Hessenbergmatrizen wird zweckmäßigerweise nicht mit dem Householderverfahren, sondern mit sogenannten *Givensrotationen* durchgeführt (Aufwand: $\mathcal{O}(n^2)$).

2.3.2 Der Spezialfall symmetrischer Matrizen

Für den Spezialfall einer symmetrischen, reellen Matrix A kann man die Aussagen noch spezifizieren.

Satz 2.10. *Zu jeder symmetrischen Matrix $A \in \mathbb{R}^{n\times n}$ existiert eine orthogonale Matrix Q mit $Q^T A Q = T$, wobei T eine Tridiagonalmatrix ist.*

Beweis. Satz 2.9. □

Weiterhin kann man für tridiagonale Matrizen stets sichern, daß alle Eigenwerte voneinander verschieden sind. Dies ist für die Konvergenz des QR-Verfahrens von Bedeutung.

Satz 2.11. *Sei*

$$T = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \\ 0 & \beta_2 & \alpha_3 & \beta_3 & \ddots \\ \vdots & & & \ddots & \\ 0 & \dots & 0 & \beta_{n-1} & \alpha_n \end{pmatrix}$$

eine symmetrische Tridiagonalmatrix. Weiterhin sei T irreduzibel, d.h. $\beta_i \neq 0$, $i = 1, \dots, n-1$. Dann sind alle Eigenwerte einfach.

Beweis. Sei λ Eigenwert und x zugehöriger Eigenvektor $\Rightarrow$

$$\begin{aligned}
x_2 &= \frac{1}{\beta_1}(\lambda - \alpha_1)x_1, \\
x_3 &= \frac{1}{\beta_2}(-\beta_1 x_1 + (\lambda - \alpha_2)x_2), \\
&\dots, \\
x_n &= \frac{1}{\beta_{n-1}}(-\beta_{n-2}x_{n-2} + (\lambda - \alpha_{n-1})x_{n-1}).
\end{aligned}$$

$x_1 = 0$ ist daher nicht möglich, da sonst $x = 0$. Wir können daher jeden Eigenvektor mit $x_1 = 1$ normieren, damit ist aber x eindeutig bestimmt und die geometrische Vielfachheit ist 1. Da für symmetrische Matrizen die geometrische Vielfachheit gleich der algebraischen Vielfachheit ist, folgt die Behauptung. □

Ist T reduzibel, d.h. es gibt ein j mit $\beta_j = 0$, dann kann man T in zwei Blöcke T_1 und T_2 zerlegen, d.h.

$$T = \begin{bmatrix} T_1 & \mathbf{0} \\ \mathbf{0} & T_2 \end{bmatrix}$$

und die Eigenwerte separat berechnen. Diese Strategie ist auch beim QR-Algorithmus empfehlenswert, wenn man erreicht hat, daß $|a^{(k)}_{j+1,j}| \ll \sqrt{|a^{(k)}_{jj} a^{(k)}_{j+1,j+1}|}$ ist. Dann zerlegt man $A^{(k)}$ in zwei Blöcke und berechnet die Eigenwerte der Blöcke.

Bemerkung: Der Rechenaufwand pro Iterationsschritt beträgt nun nur noch $\mathcal{O}(n)$ Operationen.

2.3.3 Modifikationen des QR-Algorithmus

Sind 2 betragsmäßig aufeinanderfolgende Eigenwerte sehr stark getrennt, d.h. $|\lambda_1| > \dots |\lambda_r| \gg |\lambda_{r+1}| > \dots > |\lambda_n|$, so wird nach einigen Iterationen die Matrix $A^{(k)}$ folgende Struktur besitzen:

$$A^{(k)} = \left(\begin{array}{ccc|ccc}
a^{(k)}_{11} & \dots & a^{(k)}_{1r} & a^{(k)}_{1,r+1} & \dots & a^{(k)}_{1n} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
a^{(k)}_{r1} & \dots & a^{(k)}_{rr} & a^{(k)}_{r,r+1} & \dots & a^{(k)}_{rn} \\
\hline
0 & \dots & 0 & a^{(k)}_{r+1,r+1} & \dots & a^{(k)}_{r+1,n} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \dots & 0 & a^{(k)}_{n,r+1} & \dots & a^{(k)}_{nn}
\end{array}\right)$$

Dabei bedeutet 0 im linken unteren Block, dass die Elemente "hinreichend klein" sind, um numerisch als Null zu gelten. (Falls wir mit einer Hessenbergmatrix gestartet haben, dann sind sowieso alle Elemente im linken unteren Block bis auf das Element $a^{(k)}_{r+1,r}$ gleich 0). Dann sind die Eigenwerte der Gesamtmatrix natürlich die Eigenwerte der Diagonalblöcke und man wird die Diagonalblöcke gesondert betrachten, was natürlich den Aufwand verringert.

Eine Konvergenzbeschleunigung des QR-Verfahrens erreicht man durch eine Shift-Strategie. Hierbei macht man sich zu Nutze, dass die Eigenwerte der Matrix $A - \mu I$ genau die um μ verschobenen Eigenwerte von A sind, d.h. $\sigma(A - \mu I) = \sigma(A) - \mu$:

$A^{(1)} = A$

Für $k = 1, 2, \ldots$ berechne:

1. $Q_k R_k = A^{(k)} - \mu_k I$ (QR-Zerlegung)
2. $A^{(k+1)} = R_k Q_k + \mu_k I$,

d.h. es wird die QR-Zerlegung von $A^{(k)} - \mu_k I$ berechnet. Als geeigneter Shiftparameter bietet sich eine Schätzung für den kleinsten Eigenwert an. Die Konvergenzgeschwindigkeit für das Element $a_{nj}^{(k)}$ ist dann $\mathcal{O}(|\frac{\lambda_n - \mu_k}{\lambda_j - \mu_k}|^k$, was natürlich für gute Näherungen $\mu_k \approx \lambda_n$ sehr schnell ist.

Es werden die folgenden beiden Strategien verwendet:

1. $\mu_k = a_{nn}^{(k)}$, oder
2. μ_k ist der Eigenwert λ der 2×2 Matrix

$$\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{nn}^{(k)} \end{pmatrix},$$

für den $|a_{nn}^{(k)} - \lambda|$ kleiner ist.

Für reelle Matrizen führt die erste Strategie in der Regel zu guten Ergebnissen, solange keine konjugiert komplexen Eigenwerte auftreten. Wird die zweite Strategie für relle Matrizen angewendet, so kann der Shiftparameter μ_k für nichtsymmetrische reelle Matrizen $A^{(k)}$ natürlich komplex sein, was in einer komplexen Matrix $A^{(k+1)}$ resultiert. Führt man im nächsten Schriit allerdings einen Shift mit konjugiert komplexem Parameter $\mu_{k+1} = \bar{\mu}_k$ durch, so kann gezeigt werden, dass die resultierende Matrix $A^{(k+2)}$ wieder reell ist. Darüber hinaus kann diese Doppel-Shift-Strategie so konstruiert werden, dass nur reelle Arithmetik verwendet wird.

Kapitel 3

Interpolation

Beim Interpolationsproblem sind gegeben $n+1$ reelle oder komplexe *Stützstellen* $x_0, \ldots, x_n$ und $n+1$ Stützwerte $f_0, \ldots, f_n$. Gesucht ist eine *Interpolationsfunktion* f mit $f(x_i) = f_i$, $i = 0, \ldots, n$.

Natürlich schränkt man sich bei der gesuchten Interpolationsfunktion auf gewisse Funktionenklassen ein: Normalerweise wird die gesuchte Interpolationsfunktion durch $n+1$ Parameter $a_0, \ldots, a_n$ festgelegt sein: $f(x) = \Phi(x; a_0, \ldots, a_n)$.

Lineare Interpolation ist dadurch gekennzeichnet, dass die Funktion Φ linear von den Parametern a_i abhängt:

$$\Phi(x; a_0, \ldots, a_n) = a_0\Phi_0(x) + a_1\Phi_1(x) + \ldots + a_n\Phi_n(x)$$

Dies inkludiert den klassischen Fall der *Polynominterpolation* mit $\Phi_i(x) = x^i$:

$$\Phi(x; a_0, \ldots, a_n) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

Andere wichtige Fälle sind *trigonometrische Interpolation* und *Spline Interpolation.*

Bei der *nichtlinearen Interpolation* ist dagegen eine nichtlineare Abhängigkeit von den Koeffizienten gegeben, z.B.bei der *rationalen Interpolation*

$$\Phi(x; \alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_l) = \frac{\alpha_0 + \alpha_1x + \ldots \alpha_kx^k}{\beta_0 + \beta_1x + \ldots + \beta_lx^l}$$

oder bei der *exponentiellen Interpolation*

$$\Phi(x; \alpha_0, \ldots, \alpha_k, \lambda_0, \ldots, \lambda_k) = \alpha_0e^{\lambda_0x} + \ldots + \alpha_ke^{\lambda_kx}.$$

3.1 Polynominterpolation

3.1.1 Interpolationsformel von Lagrange

Im folgenden sei $\mathcal{P}_n$ die Menge aller reellen oder komplexen Polynome vom Grad kleiner oder gleich n.

Satz 3.1. *Für beliebige $n+1$ Wertepaare (x_i, f_i), $i = 0, \ldots, n$ mit paarweise verschiedenen Stützstellen x_i (d.h. $x_i \neq x_j$ für $i \neq j$) gibt es ein eindeutiges Polynom $P \in \mathcal{P}_n$ mit*

$$P(x_i) = f_i, \ i = 0, \ldots, n.$$

Beweis. Wir zeigen die Eindeutigkeit per Widerspruchsbeweis. Seien $P_1 \neq P_2 \in \mathcal{P}_n$ 2 verschiedene Polynome mit der Intepolationseigenschaft

$$P_1(x_i) = P_2(x_i) = f_i,\ i = 0, \ldots, n.$$

Dann ist auch $P_2 - P_1 \in \mathcal{P}_n$ und besitzt $n+1$ Nullstellen $x_0, \ldots, x_n$. Ein nichttriviales Polynom vom Grad kleiner oder gleich n besitzt aber höchstens n Nullstellen und daher ist $P_2 - P_1 \equiv 0$, Widerspruch.

Für die Existenz konstruieren wir uns das Interpolationspolynom explizit mittels der Polynome $L_i \in \mathcal{P}_n$, $i = 0, \ldots, n$,

$$L_i(x) := \frac{(x - x_0) \ldots (x - x_{i-1})(x - x_{i+1}) \ldots (x - x_n)}{(x_i - x_0) \ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots (x_i - x_n)}. \tag{3.1}$$

Offensichtlich ist

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j. \end{cases}$$

Für das Polynom

$$P(x) := \sum_{i=0}^{n} f_i L_i(x) = \sum_{i=0}^{n} f_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{3.2}$$

folgt damit unmittelbar die Interpolationseigenschaft $P(x_i) = f_i$, $i = 0, \ldots, n$ □

Bezeichnung: Die Polynome $L_i(x)$ definiert durch (3.1) heißen *Lagrange Polynome*. Die Formel (3.2) heißt *Interpolationsformel von Lagrange*.

Die Interpolationsformel von Lagrange ist aus Aufwandsgründen für praktische Zwecke nur in Ausnahmefällen geeignet.

3.1.2 Der Neville Algorithmus

Der Neville Algorithmus ist gut geeignet für die Auswertung des Interpolationspolynoms an einer einzelnen Stelle x. Er löst das Interpolationsproblem sukzessive auf einer kleineren Menge von Stützstellen und verbindet diese Teillösungen zu einer Lösung des Gesamtproblems.

Für eine Teilmenge $\{i_0, \ldots, i_k\} \subset \{0, \ldots, n\}$ bezeichnen wir mit $P_{i_0 \ldots i_k} \in \mathcal{P}_k$ das Interpolationspolynom mit

$$P_{i_0 \ldots i_k}(x_{i_j}) = f_{i_j},\ j = 0, \ldots, k$$

Lemma 3.1. *Es gilt die Rekursion*

$$P_i(x) \equiv f_i, \tag{3.3}$$

$$P_{i_0 \ldots i_k}(x) = \frac{(x - x_{i_0}) P_{i_1 \ldots i_k}(x) - (x - x_{i_k}) P_{i_0 \ldots i_{k-1}}(x)}{x_{i_k} - x_{i_0}} \tag{3.4}$$

Beweis. Die Initialisierung (3.3) ist trivial. Für die allgemeine Formel setzen wir $S := P_{i_1 \ldots i_k}$, $T = P_{i_0 \ldots i_{k-1}}$ und bezeichnen die rechte Seite von (3.4) mit R, d.h.

$$R(x) = \frac{(x - x_{i_0}) S(x) - (x - x_{i_k}) T(x)}{x_{i_k} - x_{i_0}}$$

Der Grad von R ist klarerweise kleiner oder gleich k. Wegen $S(x_{i_k}) = f_{i_k}$ und $T(x_{i_0}) = f_{i_0}$ gilt

$$R(x_{i_k}) = \frac{(x_{i_k} - x_{i_0})f_{i_k} - 0}{x_{i_k} - x_{i_0}} = f_{i_k},\ R(x_{i_0}) = \frac{0 - (x_{i_0} - x_{i_k})f_{i_0}}{x_{i_k} - x_{i_0}} = f_{i_0}.$$

Für $0 < j < k$ gilt $S(x_{i_j}) = T(x_{i_j}) = f_{i_j}$ und daher

$$R(x_{i_j}) = \frac{(x_{i_j} - x_{i_0})f_{i_j} - (x_{i_j} - x_{i_k})f_{i_j}}{x_{i_k} - x_{i_0}} = f_{i_j}$$

und damit folgt die Behauptung aus Satz 3.1. □

Das gesuchte Interpolationspolynom ist natürlich das Polynom $P_{01\ldots n}$ und wird im Neville Algorithmus z.B. durch das folgende Tableau realisiert:

	$k=0$	1	2	3
x_0	$f_0 = P_0(x)$			
		$P_{01}(x)$		
x_1	$f_1 = P_1(x)$		$P_{012}(x)$	
		$P_{12}(x)$		$P_{0123}(x)$
x_2	$f_2 = P_2(x)$		$P_{123}(x)$	
		$P_{23}(x)$		
x_3	$f_3 = P_3(x)$			

Die 1.Spalte ist durch die Funktionswerte f_i gegeben, die Einträge in den nachfolgenden Spalten ergeben sich rekursiv von den beiden Nachbarn in der vorhergehenden Spalte.

Mit der Abkürzung

$$T_{i+k,k} := P_{i,i+1,\ldots,i+k}$$

ergibt sich das folgende Tableau

	$k=0$	1	2	3
x_0	$f_0 = T_{00}(x)$			
		$T_{11}(x)$		
x_1	$f_1 = T_{10}(x)$		$T_{22}(x)$	
		$T_{21}(x)$		$T_{33}(x)$
x_2	$f_2 = T_{20}(x)$		$T_{32}(x)$	
		$T_{31}(x)$		
x_3	$f_3 = T_{30}$			

Die Rekursion (3.3), (3.4) kann auch in der folgenden, effizienteren Form angeschrieben werden, wobei man ausnützt dass die T_{ik} auch als Elemente einer linken unteren Dreiecksmatrix aufgefasst werden können, die zeilenweise berechnet wird:

$$\begin{aligned} T_{i0}(x) &= f_i \\ T_{i,k}(x) &= \frac{(x - x_{i-k})T_{i,k-1}(x) - (x - x_i)T_{i-1,k-1}(x)}{x_i - x_{i-k}} \\ &= T_{i,k-1}(x) + \frac{T_{i,k-1}(x) - T_{i-1,k-1}(x)}{x_i - x_{i-k}}(x - x_i), \quad i = 0, \ldots, n; 1 \le k \le i \end{aligned}$$

Dies ergibt den folgenden Algorithmus für die Auswertung an einer Stelle x:

Algorithmus 3.1 (Neville Algorithmus)**.**
Input: $n+1$ Wertepaare (x_i, f_i), $i = 0, \ldots, n$, Argument x
Output: Wert des Interpolationspoynoms $P(x)$, für das $P(x_i) = f_i$.

```
{   for i = 0,...,n do
    {   t_i = f_i;
        for j = i-1,...,0 do
            t_j = t_{j+1} + (t_{j+1}-t_j)*(x-x_i)/(x_i-x_j);
    }
    return t_0;
}
```

Nach Ausführung der inneren Schleife gilt $t_j = T_{i,i-j}(x)$, $0 \le j \le i$. Der gewünschte Wert $T_{nn}(x)$ steht zum Schluss im Element t_0.

3.1.3 Interpolationsformel von Newton: Dividierte Differenzen

Newton's Interpolationsformel ist geeignet, falls man das Interpolationspolynom explizit aufstellen bzw. an mehreren Stellen auswerten will. Dazu verwenden wir für das Interpolationspolynom den Ansatz

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}) \quad (3.5)$$

Mit diesem Ansatz kann das Polynom an der Stelle x rekursiv mit dem sogenannten *Horner Schema* ausgewertet werden ($p = P(x)$):

```
p = a_n;
for i = n-1,...,0 do
    p = p*(x-x_i)+a_i;
```

Im Prinzip könnte man die Koeffizienten a_i sukzessive aus den folgenden Gleichungen berechnen:

$$\begin{aligned} f_0 &= P(x_0) = a_0 \\ f_1 &= P(x_1) = a_0 + a_1(x_1 - x_0) \\ f_2 &= P(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ &\vdots \end{aligned}$$

Dies würde jedoch n Divisionen und $n(n-1)$ Multiplikationen erfordern. Es gibt jedoch einen anderen Algorithmus, der lediglich $n(n+1)/2$ Divisionen erfordert. Dazu beachten wir, dass sich die Polynome $P_{i_0 i_1 \ldots i_k}$ und $P_{i_0 i_1 \ldots i_{k-1}}$ sich durch ein Polynom vom Grad k mit den k Nullstellen $x_{i_0}, \ldots, x_{i_{k-1}}$ unterscheidet, da beide Polynome an diesen Stützstellen die gleichen Werte besitzen. Es gibt also einen eindeutigen Koeffizienten $f_{i_0 i_1 \ldots i_k}$ sodass

$$P_{i_0 i_1 \ldots i_k}(x) = P_{i_0 i_1 \ldots i_{k-1}}(x) + f_{i_0 i_1 \ldots i_k}(x - x_{i_0})(x - x_{i_1})\ldots(x - x_{i_{k-1}}) \quad (3.6)$$

Unter Verwendung von $P_{i_0}(x) = f_{i_0}$ ergibt sich daher

$$P_{i_0 i_1 \ldots i_k}(x) = f_{i_0} + f_{i_0 i_1}(x - x_{i_0}) + \ldots + f_{i_0 i_1 \ldots i_k}(x - x_{i_0})(x - x_{i_1})\ldots(x - x_{i_{k-1}}),$$

wobei die Koeffizienten $f_{i_0 i_1 \ldots i_k}$ *k-te dividierte Differenzen* genannt werden.

Lemma 3.2. *1.*

$$f_{i_0 i_1 \dots i_k} = \frac{f_{i_1 \dots i_k} - f_{i_0 \dots i_{k-1}}}{x_{i_k} - x_{i_0}}. \tag{3.7}$$

2. Falls $(j_0, j_1, \dots, j_k)$ eine Permutation der Indizes $(i_0, i_1, \dots, i_k)$ ist, so gilt

$$f_{j_0 j_1 \dots j_k} = f_{i_0 i_1 \dots i_k}$$

Beweis. Aus Gleichung (3.6) wissen wir dass $f_{i_0 i_1 \dots i_k}$ der Koeffizient von x^k ist. Da $f_{i_1 \dots i_k}$ und $f_{i_0 \dots i_{k-1}}$ die Koeffizienten von x^{k-1} in $P_{i_1 \dots i_k}$ und $P_{i_0 \dots i_{k-1}}$ sind, folgt aus der Rekursion (3.4) die Beziehung (3.7). Da weiters das Interpolationspolynom $P_{i_0 i_1 \dots i_k}$ eindeutig durch die Wertepaare festgelegt ist folgt dei zweite Behauptung aus der Eindeutigkeit des Koeffizienten von x^k. □

Die Berechnung der dividierten Differenzen erfolgt analog zum Neville Algorithmus:

	$k=0$	$k=1$	$k=2$		$k=n$
x_0	f_0				
		f_{01}			
x_1	f_1		f_{012}		
		f_{12}		$\ddots$	
x_2	f_2		$\vdots$		$f_{012\dots n}$
		$\vdots$		$\ddots$	
$\vdots$	$\vdots$		$f_{n-2,n-1,n}$		
		$f_{n-1,n}$			
x_n	f_n				

Uns interessieren für unseren Ansatz (3.5) genau die Koeffizienten $a_k = f_{012\dots k}$, die mit dem folgenden Analogon zu Algorithmus 3.1 berechnet werden können:

Algorithmus 3.2 (Dividierte Differenzen)**.**
Input: $n+1$ Wertepaare (x_i, f_i), $i = 0, \dots, n$
Output: Koeffizienten $a_0, \dots a_n$ des Interpolationspolynoms $P(x)$ gegeben durch (3.5)

```
{   for i = 0,...,n do
    {   t_i = f_i;
        for j = i-1,...,0 do
            t_j = (t_{j+1}-t_j)/(x_i-x_j);
        a_i = t_0;
    }
}
```

3.1.4 Der Fehler bei der Polynominterpolation

Wir nehmen nun an, dass die Stützwerte die Funktionswerte einer Funktion f sind, $f_i = f(x_i)$. Wir wollen nun wissen, wie genau das Interpolationspolynom P die Funktion f für Argumente x, die ungleich den Stützstellen sind, approximiert.

Satz 3.2. *Sei $f : \mathbb{R} \to \mathbb{R}$ eine $n+1$ mal differenzierbare Funktion und sei P das durch $n+1$ Wertepaare $(x_i, f_i = f(x_i))$, $i = 0, 1, \ldots, n$ eindeutig bestimmte Interpolationspolynom. Dann gibt es für jedes $\bar{x}$ eine Zahl ξ in dem kleinsten Intervall $I[x0, \ldots, x_n, \bar{x}]$, das $\bar{x}$ und alle Stützstellen x_i beinhaltet, sodass*

$$f(\bar{x}) - P(\bar{x}) = \frac{\omega(\bar{x}) f^{n+1}(\xi)}{(n+1)!},$$

wobei

$$\omega(x) := (x - x_0)(x - x_1) \ldots (x - x_n)$$

Beweis. Falls $\bar{x} = x_i$ (bzw. äquivalent, $\omega(\bar{x}) = 0$), ist die Aussage trivial. Sei nun also $\omega(\bar{x}) \neq 0$,

$$K := \frac{f(\bar{x}) - P(\bar{x})}{\omega(\bar{x})}, \quad F(x) := f(x) - P(x) - K\omega(x).$$

Dann besitzt F zumindest die $n+2$ Nullstellen $\bar{x}$, $x_0, \ldots, x_n$ in I. Sukzessive Anwendung des Satzes von Rolle ergibt, dass F' zumindest $n+1$ Nullstellen in I, F'' zumindest n Nullstellen in I, usw., $F^{(n+1)}$ zumindest eine Nullstelle ξ in I besitzt. Da $P^{(n+1)}(\xi) = 0$ ist

$$F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)! = 0$$

und aus der Definition von K folgt

$$K = \frac{f(\bar{x}) - P(\bar{x})}{\omega(\bar{x})} = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

und damit die Behauptung. □

Außerhalb des Intervals $I[x_0, \ldots, x_n]$ steigt der Wert von $\omega(x)$ sehr stark an. Die Verwendung des Interpolationspolynoms außerhalb des Intervals $I[x_0, \ldots, x_n]$, genannt *Extrapolation*, sollte daher vermieden werden.

Sei nun $f \in C^\infty[a, b]$ eine reelle Funktion, die auf $[a, b]$ unendlich oft differenzierbar ist. Zu jeder Unterteilung $a = x_0 < x_1 < \ldots < x_n = b$ kann man dann das Interpolationspolynom P_n aufstellen und man könnte erwarten dass P_n gegen f konvergiert, falls die Feinheit der Unterteilung, $\max_i |x_{i+1} - x_i|$ gegen 0 geht. Dies ist aber im allgemeinen nicht der Fall, wie man sich an Hand von Gegenbeispielen überlegen kann: Interpolationspolynome von einem hohen Grad haben üblicherweise einen sehr stark oszillierenden Anteil.

3.1.5 Hermite Interpolation

Bei der *Hermite Interpolation* sind gegeben $m+1$ paarweise verschiedene Stützstellen $\xi_0, \ldots, \xi_m$ und Werte $y_i^{(k)}$, $k = 0, \ldots, n_i - 1$; $i = 0, \ldots, m$. Gesucht ist ein Polynom $P \in \mathcal{P}_n$, wobei

$$n + 1 := \sum_{i=0}^{m} n_i,$$

sodass die folgende Interpolationbedingung erfüllt ist:

$$P^{(k)}(\xi_i) = y_i^{(k)}, \ k = 0, \ldots, n_i - 1; \ i = 0, \ldots, m \tag{3.8}$$

Ähnlich zu Satz 3.1 zeigt man, dass dieses Problem eine eindeutige Lösung besitzt: Sind nämlich P_1, P_2 2 Interpolationspolynome, so ist $P_1 - P_2 \in \mathcal{P}_n$ und es gilt $(P_1 - P_2)^{(k)}(\xi_i)) = 0$, $0 \le i \le m$, $0 \le k < n_i$. Dann ist ξ_i eine n_i-fache Nullstelle von $P_1 - P_2$: $P_1 - P_2$ besitzt also mehr als n Nullstellen und verschwindet daher identisch. Die Existenz ist eine Konsequenz der Eindeutigkeit: Die $n+1$ Werte $P^{(k)}(\xi_i)$ in (3.8) hängen linear von den gesuchten $n+1$ Koeffizienten $a_0, \ldots, a_n$ des Interpolationspolynoms P ab. Die Systemmatrix ist regulär wegen der Eindeutigkeit der Lösung, da eine lineare Abbildung von $\mathbb{R}^{n+1}$ nach $\mathbb{R}^{n+1}$ genau dann bijektiv ist, wenn sie injektiv ist.

Die Berechnung des Interpolationspolynoms geschieht am besten mit der Methode der dividierten Differenzen für die $n+1$ Wertepaare

$$\underbrace{(x_0, f_0), \ldots, (x_{n_0-1}, f_{n_0-1})}_{\equiv(\xi_0, y_0^{(0)})}, \underbrace{(x_{n_0}, f_{n_0}), \ldots, (x_{n_0+n_1-1}, f_{n_0+n_1-1})}_{(\xi_1, y_1^{(0)})}, \ldots, \underbrace{\ldots, (x_n, f_n)}_{\equiv(\xi_m, y_m^{(0)})}$$

Dann kann man $f_{i,\ldots,i+k}$ für $x_i = x_{i+k}$ natürlich nicht mit (3.7) berechnen, wir benötigen dazu folgenden Grenzübergang: Seien $\zeta_i < \zeta_{i+1} < \ldots < \zeta_{i+k}$ paarweise verschiedene Stützstellen mit zugehörigen Stützwerten $P(\zeta_j)$, $i \le j \le i+k$. Betrachten wir nun den Fehler bei der Interpolation von P durch $P_{i,\ldots,i+k-1}$ an der Stelle ζ_{i+k}, so ergibt sich aus Satz 3.2

$$P(\zeta_{i+k}) = P_{i,\ldots,i+k-1}(\zeta_{i+k}) + \frac{P^{(k)}(\bar{\zeta})}{k!}(\zeta_{i+k} - \zeta_i) \ldots (\zeta_{i+k} - \zeta_{i+k-1})$$

mit $\bar{\zeta} \in I(\zeta_i, \ldots, \zeta_{i+k})$. Aus $P_{i,\ldots,i+k}(\zeta_{i+k}) = P(\zeta_{i+k})$ und (3.6) folgt $f_{i,\ldots,i+k} = \frac{P^{(k)}(\bar{\zeta})}{k!}$ und daher für $\lim \zeta_i = \lim \zeta_{i+k} = x_i$ die Beziehung $f_{i,\ldots,i+k} = \frac{P^{(k)}(x_i)}{k!}$. Mit der Definition $r(i) := l$, $\sum_{j=0}^{l-1} n_j \le i < \sum_{j=0}^{l} n_j$, $l = 0, \ldots, m$ ergibt sich daher die folgende Rekursion:

$$\begin{aligned} f_i &= y_{r(i)}^{(0)},\ i = 0, \ldots, n \\ f_{i,\ldots,i+k} &= \begin{cases} y_{r(i)}^{(k)}/k! & \text{falls } x_i = x_{i+k} \\ (f_{i+1,\ldots,i+k} - f_{i,\ldots,i+k-1})/(x_{i+k} - x_i) & \text{sonst} \end{cases} \end{aligned}$$

3.2 Spline Interpolation

Sei $\Delta := \{a = x_0 < x_1 < \ldots < x_n = b\}$ eine Partition des Intervalls $[a, b]$.

Definition 3.1. *Eine reelle Funktion $S_\Delta : [a, b] \to \mathbb{R}$ heißt* kubischer Spline *auf Δ genau dann wenn S_Δ auf $[a, b]$ zweimal stetig differenzierbar ist und S_Δ auf jedem Teilintervall $[x_i, x_{i+1}]$, $i = 0, \ldots, n-1$ mit einem Polynom 3.Grades übereinstimmt.*

Ein kubischer Spline setzt sich also stückweise aus kubischen Polynomen zusammen, sodass die Funktionswerte und ersten beiden Ableitungen an den Stützstellen x_i übereinstimmen.

Sei nun $Y := \{y_0, \ldots, y_n\}$ ein Menge von zugehörigen Stützwerten. Wir bezeichnen mit $S_\Delta(Y; \cdot)$ eine interpolierende Splinefunktion S_Δ mit $S_\Delta(Y; x_i) = y_i$, $i = 0, \ldots, n$.

Wir betrachten nun die Existenz solcher interpolierenden Splinefunktionen.

Satz 3.3. *Für jede Menge $Y := \{y_0, \ldots, y_n\}$ gibt es eine eindeutig bestimmte interpolierende Splinefunktion, die zusätzlich eine der 3 Bedingungen erfüllt:*

1. $S''_\Delta(Y;a) = S''_\Delta(Y;b) = 0$.
2. S_Δ *ist periodisch, d.h.* $S''_\Delta(Y;a) = S''_\Delta(Y;b)$, $S'_\Delta(Y;a) = S'_\Delta(Y;b)$ *(nur sinnvoll wenn* $y_0 = y_n$*).*
3. $S'_\Delta(Y;a) = y'_0$, $S'_\Delta(Y;b) = y'_n$ *für gegebenes* y'_0, y'_n

Beweis. Der Beweis ist konstruktiv. Sei

$$h_{j+1} := x_{j+1} - x_j, \quad M_j = S''_\Delta(Y;x_j).$$

Da $S''_\Delta(Y;\cdot)$ auf jedem Teilintervall $[x_j, x_{j+1}]$ linear ist folgt die Darstellung

$$S''_\Delta(Y;x) = M_j \frac{x_{j+1}-x}{h_{j+1}} + M_{j+1}\frac{x-x_j}{h_{j+1}}, \quad x \in [x_j, x_{j+1}]$$

und nach Integration

$$\begin{aligned} S'_\Delta(Y;x) &= -M_j\frac{(x_{j+1}-x)^2}{2h_{j+1}} + M_{j+1}\frac{(x-x_j)^2}{2h_{j+1}} + A_j, \quad x \in [x_j, x_{j+1}] \\ S_\Delta(Y;x) &= M_j\frac{(x_{j+1}-x)^3}{6h_{j+1}} + M_{j+1}\frac{(x-x_j)^3}{6h_{j+1}} + A_j(x-x_j) + B_j, \quad x \in [x_j, x_{j+1}] \end{aligned}$$

Aus der Interpolationsbedingung ergibt sich

$$\begin{aligned} S_\Delta(Y;x_j) = M_j\frac{h_{j+1}^2}{6} + B_j &= y_j \\ S_\Delta(Y;x_{j+1}) = M_{j+1}\frac{h_{j+1}^2}{6} + A_j h_{j+1} + B_j &= y_{j+1} \end{aligned}$$

und daher

$$B_j = y_j - M_j\frac{h_{j+1}^2}{6}, \quad A_j = \frac{y_{j+1}-y_j}{h_{j+1}} - \frac{h_{j+1}}{6}(M_{j+1}-M_j)$$

Die Stetigkeit von $S'_\Delta(Y;\cdot)$ an den Stützstellen x_j, $j = 1,\ldots,n-1$ ergibt die $n-1$ Gleichungen $S'_\Delta(Y;x_j^-) = S'_\Delta(Y;x_j^+)$ und daher

$$M_j\frac{h_j}{2} + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(M_j - M_{j-1}) = -M_j\frac{h_{j+1}}{2} + \frac{y_{j+1}-y_j}{h_{j+1}} - \frac{h_{j+1}}{6}(M_{j+1}-M_j)$$

bzw.

$$\frac{h_j}{6}M_{j-1} + \frac{h_j+h_{j+1}}{3}M_j + \frac{h_{j+1}}{6}M_{j+1} = \frac{y_{j+1}-y_j}{h_{j+1}} - \frac{y_j-y_{j-1}}{h_j}, \ j = 1,\ldots,n-1$$

Dies sind $n-1$ Gleichungen für die $n+1$ Unbekannten M_j, $j = 0,\ldots,n$, wir benötigen also noch zwei Bedingungen:

1. $S''_\Delta(Y;a) = M_0 = S''_\Delta(Y;b) = M_n = 0$. Die restlichen M_j ergeben sich als eindeutige Lösung des Gleichungssystems

$$\begin{pmatrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} \\ & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_n} - \frac{y_{n-1}-y_{n-2}}{h_{n-1}} \end{pmatrix}$$

2. S_Δ ist periodisch, d.h. $S''_\Delta(Y;a) = S''_\Delta(Y;b)$, $S'_\Delta(Y;a) = S'_\Delta(Y;b)$ (Voraussetzung $y_0 = y_n$): Dies ergibt die Gleichungen $M_0 = M_n$ und

$$\frac{h_n}{6}M_{n-1} + \frac{h_n + h_1}{3}M_n + \frac{h_1}{6}M_1 = \frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n}.$$

Wir müssen also das Gleichungssystem

$$\begin{pmatrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & & \frac{h_1}{6} \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} & \\ \frac{h_1}{6} & & & \frac{h_n}{6} & \frac{h_n+h_1}{3} & \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \frac{y_3-y_2}{h_3} - \frac{y_2-y_1}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_n} - \frac{y_{n-1}-y_{n-2}}{h_{n-1}} \\ \frac{y_1-y_n}{h_1} - \frac{y_n-y_{n-1}}{h_n} \end{pmatrix}$$

lösen.

3. $S'_\Delta(Y;a) = y'_0$, $S'_\Delta(Y;b) = y'_n$ für gegebenes y'_0, y'_n: Dies führt auf die Gleichungen

$$\frac{h_1}{3}M_0 + \frac{h_1}{6}M_1 = \frac{y_1 - y_0}{h_1} - y'_0, \ \frac{h_n}{6}M_{n-1} + \frac{h_n}{3}M_n = y'_n - \frac{y_n - y_{n-1}}{h_n}$$

und die Lösung des Gleichungssystems

$$\begin{pmatrix} \frac{h_1}{3} & \frac{h_1}{6} & & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} & \frac{h_n}{6} \\ & & & & \frac{h_n}{6} & \frac{h_n}{3} \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} \frac{y_1-y_0}{h_1} - y'_0 \\ \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_n} - \frac{y_{n-1}-y_{n-2}}{h_{n-1}} \\ y'_n - \frac{y_n-y_{n-1}}{h_n} \end{pmatrix}$$

In jedem der 3 Fälle ist die Systemmatrix symmetrisch und positive definit, da strikt diagonaldominant. Damit sind die 2.Ableitungen an den Stützstellen eindeutig bestimmt und nach unseren Überlegungen auch die Splinefunktion. □

Für kubische Splines gelten die folgenden Konvergenzeigenschaften:

Satz 3.4. *Sei $f \in C^4([a,b])$, $L := \max_{x\in[a,b]} |f^{(4)}(x)|$, $K := \frac{\max_j x_{j+1}-x_j}{\min_j x_{j+1}-x_j}$ und S_Δ der kubische Spline, der die Werte von f an den Stützstellen $x_0, \ldots, x_n \in \Delta$ interpoliert und $S'_\Delta(a) = f'(a)$, $S'_\Delta(b) = f'(b)$ erfüllt. Dann gilt*

$$\max_{x\in[a,b]} |f^{(k)}(x) - S^{(k)}_\Delta(x)| \le 2LK(\max_j x_{j+1} - x_j)^{4-k}, \quad k = 0,1,2,3$$

Beweis. siehe Literatur □

Daneben haben Splines nicht wie Polynome die unangenehme Eigenschaft, dass sie bei feiner werdender Unterteilung stark oszillieren. Es gilt die folgende Minimaleigenschaft:

Satz 3.5. *Sei $f \in C^2([a,b])$ und S_Δ der kubische Spline, der die Werte von f an den Stützstellen $x_0, \ldots, x_n \in \Delta$ interpoliert und eine der 3 folgenden Bedingungen erfüllt:*

1. $S''_\Delta(a) = S''_\Delta(b) = 0$.
2. S_Δ *ist periodisch, d.h.* $S''_\Delta(a) = S''_\Delta(b)$, $S'_\Delta(a) = S'_\Delta(b)$, *falls auch* $f(a) = f(b)$, $f'(a) = f'(b)$, *f"(a)=f"(b)*
3. $S'_\Delta(a) = f'(a)$, $S'_\Delta(b) = f'(b)$

Dann gilt

$$\int_a^b (f''(x))^2 \,\mathrm{d}x - \int_a^b (S''_\Delta(x))^2 \,\mathrm{d}x = \int_a^b (f''(x) - S''_\Delta(x))^2 \,\mathrm{d}x \geq 0,$$

und damit auch

$$\int_a^b (S''_\Delta(x))^2 \; dx \leq \int_a^b (f''(x))^2 \,\mathrm{d}x$$

Beweis. Mittels partieller Integration folgt für $j = 1, \ldots, n$

$$\begin{aligned}
\int_{x_{j-1}}^{x_j} (f''(x) - S''_\Delta(x))S''_\Delta(x) \,\mathrm{d}x &= (f'(x) - S'_\Delta(x))S''_\Delta(x)|_{x_{j-1}}^{x_j} - \int_{x_{j-1}}^{x_j} (f'(x) - S'_\Delta(x))S'''_\Delta(x) \,\mathrm{d}x \\
&= (f'(x) - S'_\Delta(x))S''_\Delta(x)|_{x_{j-1}}^{x_j} - (f(x) - S_\Delta(x))S'''_\Delta(x)|_{x_{j-1}^+}^{x_j^-} \\
&= (f'(x) - S'_\Delta(x))S''_\Delta(x)|_{x_{j-1}}^{x_j}
\end{aligned}$$

und nach Summation

$$\int_a^b (f''(x) - S''_\Delta(x))S''_\Delta(x) \,\mathrm{d}x = (f'(x) - S'_\Delta(x))S''_\Delta(x)|_a^b$$

Der Term auf der rechten Seite verschwindet aber in jedem der 3 Fälle und daher ist $\int_a^b (f''(x) - S''_\Delta(x))S''_\Delta(x) \,\mathrm{d}x = 0$. Damit folgt aber

$$\begin{aligned}
\int_a^b (f''(x) - S''_\Delta(x))^2 \,\mathrm{d}x &= \int_a^b (f''(x))^2 - (S''_\Delta(x))^2 - 2(f''(x) - S''_\Delta(x))S''_\Delta(x) \,\mathrm{d}x \\
&= \int_a^b (f''(x))^2 \,\mathrm{d}x - \int_a^b (S''_\Delta(x))^2 \,\mathrm{d}x
\end{aligned}$$

und unmittelbar die restlichen Behauptungen.

□

Kapitel 4

Numerische Integration

Gegeben ist nun eine stetige Funktion $f : [a, b] \to \mathbb{R}$, gesucht ist

$$\mathcal{I}(f) := \int_a^b f(x) \ \mathrm{d}x, \tag{4.1}$$

bzw. etwas allgemeiner, für eine Gewichtsfunktion $w : (a, b) \to \mathbb{R}_+ \setminus \{0\}$ ist gesucht

$$\mathcal{I}_w(f) := \int_a^b f(x)w(x) \ \mathrm{d}x, \tag{4.2}$$

4.1 Die Newton-Cotes Formeln

Eine einfache Möglichkeit, eine Näherung für $\mathcal{I}_w(f)$ zu berechnen, ist wie folgt: Wähle $n + 1$ Stützstellen $a \leq x_0 < x_1 < \ldots < x_n \leq b$ und ersetze in (4.2)f durch das Interpolationspolynom P_n:

$$\mathcal{I}_w(f) \approx \mathcal{I}_w^n(f) := \int_a^b P_n(x)w(x) \ \mathrm{d}x$$

Mittels der Lagrange Interpolationsformel (3.2) $P_n(x) = \sum_{i=0}^n f(x_i)L_i(x)$ ergibt sich

$$\mathcal{I}_w^n(f) = \int_a^b P_n(x)w(x) \ \mathrm{d}x = \int_a^b \sum_{i=0}^n f(x_i)L_i(x)w(x) \ \mathrm{d}x = \sum_{i=0}^n f(x_i) \underbrace{\int_a^b L_i(x)w(x) \ \mathrm{d}x}_{:=\alpha_i}$$

wobei die Koeffizienten α_i, $i = 0, \ldots, n$ nicht mehr von f abhängen, sondern nur mehr von n und den gewählten Stützstellen $x_0, \ldots, x_n$.

Satz 4.1. *Für jedes n seien $n + 1$ Stützstellen $a \leq x_0^{(n)} < \ldots x_1^{(n)} < \ldots < x_n^{(n)} \leq b$ und $n + 1$ Gewichte $\alpha_i^{(n)}$, $i = 0, \ldots, n$ gegeben, die die Quadraturformel*

$$\mathcal{I}_w^n : C([a, b]) \to \mathbb{R}, \quad f \to \mathcal{I}_w^n(f) := \sum_{i=0}^n \alpha_i^{(n)} f(x_i^{(n)})$$

definieren. Weiters seien die folgenden beiden Voraussetzungen erfüllt:

1. $\exists M \forall n : \ \sum_{i=0}^n |\alpha_i^{(n)}| \leq M$

2. Für jedes Polynom P gilt $\lim_{n\to\infty} \mathcal{I}_w^n(P) = \mathcal{I}_w(P)$.

Dann gilt für jede stetige Funktion $f : [a, b] \to \mathbb{R}$

$$\lim_{n\to\infty} \mathcal{I}_w^n(f) = \mathcal{I}_w(f).$$

Beweis. Wir verwenden den Satz von Weierstraß, dass die Polynome dicht in $C([a, b])$ liegen, d.h.

$$\forall f \in C([a,b])\forall \epsilon > 0 \exists m \exists P \in \mathcal{P}_m : \ \|f - P\|_\infty = \max_{t\in[a,b]} |f(t) - P(t)| < \epsilon$$

Sei nun $f \in C([a, b])$ und $\epsilon > 0$ beliebig aber fest. Wir wählen m und $P \in \mathcal{P}_m$ sodass

$$\|f - P\|_\infty < \frac{\epsilon}{2(M + W)},$$

wobei $W := \int_a^b w(x)\ \mathrm{d}x$, und anschließend N, sodass

$$\forall n \geq N : \ |\mathcal{I}_w(P) - \mathcal{I}_w^n(P)| \leq \frac{\epsilon}{2}.$$

Dann gilt für $n \geq N$

$$\begin{aligned}
|\mathcal{I}_w(f) - \mathcal{I}_w^n(f)| &= |\mathcal{I}_w(P) - \mathcal{I}_w^n(P) + \mathcal{I}_w(f - P) - \mathcal{I}_w^n(f - P)| \\
&\leq |\mathcal{I}_w(P) - \mathcal{I}_w^n(P)| + |\mathcal{I}_w(f - P)| + |\mathcal{I}_w^n(f - P)| \\
&\leq \frac{\epsilon}{2} + \int_a^b |f(x) - P(x)|w(x)\ \mathrm{d}x + \sum_{i=0}^{n} |\alpha_i^{(n)}| \cdot |f(x_i^{(n)}) - P(x_i^{(n)})| \\
&\leq \frac{\epsilon}{2} + \|f - P\|_\infty (\int_a^b w(x)\ \mathrm{d}x + \sum_{i=0}^{n} |\alpha_i^{(n)}|) < \epsilon,
\end{aligned}$$

woraus die Behauptung folgt. □

Für unseren obigen Zugang integrieren wir Polynome vom Grad n exakt, die 2.Voraussetzung $\lim_{n\to\infty} \mathcal{I}_w^n(P) = \mathcal{I}_w(P)$ ist also erfüllt. Bezüglich der 1.Voraussetzung beachten wir, dass $\sum_{i=0}^n \alpha_i = \mathcal{I}_w(1) = \int_a^b w(x)\ \mathrm{d}x$. Können wir also die Stützstellen $x_0, \ldots, x_n$ so wählen, dass die resultierenden Gewichte α_i, $i = 0, \ldots, n$ alle nicht negativ sind, so erhalten wir für $n \to \infty$ Konvergenz der berechneten Näherungen zum Wert des Integrals.

Für den Fall $w \equiv 1$, $h := (b - a)/n$, $x_i = a + ih$, $i = 0, \ldots, n$ ergeben sich die folgenden Koeffizienten $\alpha_i = \frac{b-a}{s}\sigma_i$ (Newton-Cotes Formeln)

n	σ_i							s	Fehler	Name
1	1	1						2	$h^3 \frac{1}{12} f''(\xi)$	Trapezregel
2	1	4	1					6	$h^5 \frac{1}{90} f^{(4)}(\xi)$	Simpsonregel
3	1	3	3	1				8	$h^5 \frac{3}{80} f^{(4)}(\xi)$	3/8-Regel
4	7	32	12	32	7			90	$h^7 \frac{8}{945} f^{(6)}(\xi)$	Milneregel
5	19	75	50	50	75	19		288	$h^7 \frac{275}{12096} f^{(6)}(\xi)$	
6	41	216	27	272	27	216	41	840	$h^9 \frac{9}{1400} f^{(8)}(\xi)$	Weddleregel

Hierbei bezeichnet ξ eine Zwischenstelle aus dem Intervall (a, b). Interessant ist, dass für die Regeln mit geradem n die Genauigkeit um eine Ordnung höher ist als das entsprechende

Interpolationspolynom. Wir betrachten kurz die Herleitung des Fehlers für die Trapezregel und die Simpsonregel. Sei dazu vorerst $[a,b] = [-1,1]$. Mittels partieller Integration erhält man

$$\begin{aligned}\mathcal{I}(f) &= f(x)x|_{-1}^{1} - \int_{-1}^{1} f'(x)x \ \mathrm{d}x = (f(1)+f(-1)) - f'(x)\frac{x^2}{2}|_{-1}^{1} + \int_{-1}^{1} f''(x)\frac{x^2}{2} \ \mathrm{d}x \\ &= \mathcal{I}^1(f) - \frac{1}{2}(f'(1)-f'(-1)) + \int_{-1}^{1} f''(x)\frac{x^2}{2} \ \mathrm{d}x \end{aligned} \tag{4.3}$$

Weiters ist $(f'(1)-f'(-1)) = \int_{-1}^{1} f''(x) \ \mathrm{d}x$ und daher nach dem Mittelwertsatz der Integralrechnung

$$\mathcal{I}(f) - \mathcal{I}^1(f) = \frac{1}{2}\int_{-1}^{1} f''(x)(x^2-1) \ \mathrm{d}x = \frac{1}{2}f''(\xi)\int_{-1}^{1}(x^2-1) \ \mathrm{d}x = -\frac{2}{3}f''(\xi) = -\frac{2^3}{12}f''(\xi)$$

Für die Simpsonregel beachtet man, dass aus (4.3) nach neuerlicher partieller Integration

$$\mathcal{I}(f) = \mathcal{I}^1(f) - \frac{1}{2}(f'(1)-f'(-1)) + \frac{1}{6}(f''(1)+f''(-1)) + \frac{1}{24}\int_{-1}^{1} f^{(4)}(x)(x^4-1) \ \mathrm{d}x \tag{4.4}$$

folgt. Aus

$$\begin{aligned} f(1) &= f(0) + f'(0) + \frac{1}{2}f''(0) + \frac{1}{6}f'''(0) + \frac{1}{6}\int_{0}^{1} f^{(4)}(x)(1-x)^3 \ \mathrm{d}x \\ f(-1) &= f(0) - f'(0) + \frac{1}{2}f''(0) - \frac{1}{6}f'''(0) + \frac{1}{6}\int_{-1}^{0} f^{(4)}(x)(1+x)^3 \ \mathrm{d}x \end{aligned}$$

folgt

$$(f(1)+f(-1)-2f(0)) - f''(0) - \frac{1}{6}\int_{-1}^{1} f^{(4)}(x)(1-|x|)^3 \ \mathrm{d}x = 0. \tag{4.5}$$

Weiters ist

$$\begin{aligned} f'(1) &= f'(0) + f''(0) + \frac{1}{2}f'''(0) + \frac{1}{2}\int_{0}^{1} f^{(4)}(x)(1-x)^2 \ \mathrm{d}x \\ f'(-1) &= f'(0) - f''(0) + \frac{1}{2}f'''(0) - \frac{1}{2}\int_{-1}^{0} f^{(4)}(x)(1+x)^2 \ \mathrm{d}x \end{aligned}$$

und daher

$$(f'(1)-f'(-1)) - 2f''(0) - \frac{1}{2}\int_{-1}^{1} f^{(4)}(x)(1-|x|)^2 \ \mathrm{d}x = 0 \tag{4.6}$$

Analog erhält man

$$f''(1) + f''(-1) - 2f''(0) - \int_{-1}^{1} f^{(4)}(x)(1-|x|) \ \mathrm{d}x = 0. \tag{4.7}$$

Zählt man nun zur rechten Seite von (4.4) den Ausdruck $-\frac{2}{3}(4.5)+\frac{1}{2}(4.6)-\frac{1}{6}(4.7)$ hinzu, ergibt sich durch

$$\begin{aligned}\mathcal{I}(f) &= \mathcal{I}^2(f)+\int_{-1}^{1} f^{(4)}(x)\left(\frac{x^4-1}{24}+\frac{(1-|x|)^3}{9}-\frac{(1-|x|)^2}{4}+\frac{(1-|x|)}{6}\right)\,\mathrm{d}x \\ &= \mathcal{I}^2(f)+\frac{1}{72}f^{(4)}(\xi)\int_{-1}^{1}\underbrace{(3x^4-8|x|^3+6x^2-1)}_{\leq 0 \text{ auf } [-1,1]}\,\mathrm{d}x = \mathcal{I}^2(f)-\frac{(2/2)^5}{90}f^{(4)}(\xi)\end{aligned}$$

die gewünschte Fehlerabschätzung. Im allgemeinen Fall $f:[a,b]\to\mathbb{R}$ betrachten wir die Funktion $\tilde{f}:[-1,1]\to\mathbb{R}$, $\tilde{f}(x):=f(a+\frac{x+1}{2}(b-a))$. Wegen $\mathcal{I}(f)=\frac{b-a}{2}\mathcal{I}(\tilde{f})$, $\mathcal{I}^n(f)=\frac{b-a}{2}\mathcal{I}^n(\tilde{f})$, $\tilde{f}^{(k)}(\xi)=\left(\frac{b-a}{2}\right)^k f^{(k)}(\tilde{\xi})$ folgt die Behauptung.

Für größere n treten bei den Newton-Cotes Formeln negative Gewichte auf und man erhält nicht unbedingt Konvergenz bei feiner werdender Stützstellenwahl. Deshalb werden die Newton-Cotes Formeln meist nur für $n=1$ oder $n=2$ verwendet, dann aber nicht auf dem gesamten Intervall $[a,b]$ sondern auf einer Anzahl von Teilintervallen.

Wir untersuchen dies für eine gleichmäßige Unterteilung $x_i=a+ih$, $i=0,\ldots,N$, $h:=(b-a)/N$. Für jedes Teilintervall $[x_i,x_{i+1}]$ erhält man mit der Trapezeregel für $\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x$ die Näherung $I_i:=\frac{h}{2}(f(x_i)+f(x_{i+1}))$ und damit insgesamt

$$\mathcal{I}(f)\approx T_h(f):=\sum_{i=0}^{N-1} I_i = h\left(\frac{f(a)}{2}+f(a+h)+f(a+2h)+\ldots+f(b-h)+\frac{f(b)}{2}\right)$$

Für den Fehler gilt für $f\in C^2[a,b]$

$$I_i-\int_{x_i}^{x_{i+1}} f(x) = \frac{h^3}{12}f''(\xi_i),\ \xi_i\in[x_i,x_{i+1}]$$

und daher mit Hilfe des Zwischenwertsatzes

$$T_h(f)-\mathcal{I}(f)=\frac{h^3}{12}\sum_{i=0}^{N-1}f''(\xi_i)=\frac{h^2}{12}(b-a)\frac{1}{N}\sum_{i=0}^{N-1}f''(\xi_i)=\frac{h^2}{12}(b-a)f''(\xi),\ \xi\in[a,b]$$

da $\min_{x\in[a,b]}f''(x)\leq\frac{1}{N}\sum_{i=0}^{N-1}f''(\xi_i)\leq\max_{x\in[a,b]}f''(x)$ und f'' stetig ist.

Für die Simpsonregel (N gerade) erhält man für das Teilintervall $[x_{2i},x_{2i+1},x_{2i+2}]$, $i=0,\ldots\frac{N}{2}-1$ die Näherung $\frac{2h}{6}(f(x_{2i})+4f(x_{2i+1})+f(x_{2i+2}))$ und daher

$$\mathcal{I}(f)\approx S_h(f):=\frac{h}{3}(f(a)+4f(a+h)+2f(a+2h)+4f(a+3h)+\ldots+2f(b-2h)+4f(b-h)+f(b)).$$

Für den Fehler folgt analog wie bei der Trapezregel

$$S_h(f)-\mathcal{I}(f)=\frac{h^4}{180}(b-a)f^{(4)}(\xi),\ \xi\in[a,b]$$

Konvergenz von $T_h(f)$ bzw. $S_h(f)$ gegen $\mathcal{I}(f)$ für $h\to 0$ folgt aus Satz 4.1.

4.1.1 Extrapolation

Für festes $f \in C^{m+2}[a,b]$ gilt die Entwicklung

$$T_h(f) = \tau_0 + \tau_1 h^2 + \tau_2 h^2 + \ldots \tau_m h^{2m} + \alpha_{m+1}(h) h^{2m+2},$$

wobei $\tau_0 = \int_a^b f(x)\ \mathrm{d}x$ und α_{m+1} gleichmäßig beschränkt ist (ohne Beweis!). Die Koeffizienten $\tau_0 \ldots \tau_m$ haängen dabei nur von f ab und nicht von h.

Die Idee bei der Extrapolation ist nun wie folgt: Für eine Reihe von Schrittweiten

$$h_0 := \frac{b-a}{n_0},\ h_1 := \frac{b-a}{n_1}, \ldots, h_m = \frac{b-a}{n_m}$$

bestimmt man entsprechende Näherungen mit der Trapezregel

$$T_{i0} := T_{h_i}(f),\ i = 0, \ldots, m$$

Bezeichnet $\tilde{T}_{mm}(h)$ das Interpolationspolynom

$$\tilde{T}_{mm}(h) = a_0 + a_1 h^2 + \ldots + a_m h^{2m}$$

in h^2, für das $\tilde{T}_{mm}(h_i) = T_{h_i}(f)$, $i = 0, \ldots, m$ gilt, dann wählen wir $\tilde{T}_{mm}(0)$ als Näherung für das gewünschte Integral τ_0. Zur Berechnung von $\tilde{T}_{mm}(0)$ ist natürlich der Neville Algorithmus bestens geeignet: Die Rekursionsformel lautet nun für $x_i = h_i^2$, $x = 0$

$$T_{ik} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{(\frac{h_{i-k}}{h_i})^2 - 1}$$

Normalerweise werden die folgenden Schrittweiten verwendet:

$$h_1 = \frac{h_0}{2}, h_2 = \frac{h_1}{2}, h_3 = \frac{h_2}{2}, \ldots, h_i = \frac{h_{i-1}}{2} \quad \textit{(Rombergextrapolation)}$$
$$h_1 = \frac{h_0}{2}, h_2 = \frac{h_0}{3}, h_3 = \frac{h_1}{2}, \ldots, h_i = \frac{h_{i-2}}{2}, i = 3, 4, \ldots$$

Die zweite Schrittweitenfolge hat den Vorteil, dass der Aufwand zur Berechnung von $T_{h_i}(f)$ nicht so stark anwächst.

Normalerweise begnügt man sich mit relativ kleinen Werten von m ($m \leq 6$): Extrapolation entspricht auch einer linearen Interpolationsformel $\sum_i \alpha_i f(x_i)$ und für größere Werte von m wird die Summe der absoluten Gewichte $\sum_i |\alpha_i|$ groß. Tatsächlich ist es so, dass jeder der berechneten Werte T_{ik} einer Integrationsformel entspricht: Man berechnet daher nicht das gesamte Tableau an Interpolationswerten, sondern nur die ersten m Spalten und bricht die Berechnung ab, falls

$$|T_{ik} - T_{i+1,k}| \leq \epsilon s$$

für ein k zwische 0 und m gilt, wobei s eine grobe Näherung für das gesuchte Integral ist.

Diese Extrapolation ist natürlich nicht auf Integration beschränkt, sie kann natürlich auf jedes Berechnungsproblem angewandt werden, für das die Näherungswerte $B(h)$ in Abhängigkeit von einem Parameter h die Form

$$B(h) = \beta_0 + \beta_1 h^{1\gamma} + \beta_2 h^{2\gamma} + \ldots + \beta_m h^{m\gamma} + \beta_{m+1}(h) h^{(m+1)\gamma}$$

besitzen (z.B. Numerische Differentiation)

4.2 Gauß-Quadratur

Für eine Integrationsformel $\mathcal{I}_w^n(f) = \sum_{i=0}^n \alpha_i f(x_i)$ kann man sowohl die Gewichte α_i als auch die Stützstellen x_i, also insgesamt $2n+2$ Werte, wählen. Man könnte nun versuchen, diese Paramter so zu bestimmen, dass Polynome vom Grad $2n+1$ noch exakt integriert werden. Man sucht also α_i, x_i, $i = 0, \ldots, n$ als Lösung des nichtlinearen Gleichungssystems

$$\begin{aligned}
\sum_{i=0}^n \alpha_i \cdot 1 &= \int_a^b 1 \cdot w(x) \ \mathrm{d}x \\
\sum_{i=0}^n \alpha_i \cdot x_i &= \int_a^b x w(x) \ \mathrm{d}x \\
\sum_{i=0}^n \alpha_i \cdot x_i^2 &= \int_a^b x^2 w(x) \ \mathrm{d}x \\
&\vdots \\
\sum_{i=0}^n \alpha_i \cdot x_i^{2n+1} &= \int_a^b x^{2n+1} w(x) \ \mathrm{d}x
\end{aligned}$$

Im Allgemeinfall läßt sich dieses nichtlineare Gleichungssystem nicht explizit lösen. Daher geht man anders heran. Sei

$$\langle f, g \rangle_w := \int_a^b f(x) g(x) w(x) \ \mathrm{d}x = \mathcal{I}_w(fg)$$

das gewichtete Skalarprodukt. Wir zeigen nun den folgenden Satz:

Satz 4.2. *Seien $x_0 < x_1 < \ldots < x_n$ die Stützstellen der Quadraturformel $\mathcal{I}_w^n(f)$, wobei die Gewichte so gewählt sind, daß $\mathcal{I}(f) = \mathcal{I}_w^n(f)$ für alle $f \in \mathcal{P}_n$. Dann sind die folgenden zwei Aussagen äquivalent:*

- *Es gilt $\mathcal{I}_w(f) = \mathcal{I}_w^n(f)$ für alle $f \in \mathcal{P}_{2n+1}$*
- *Für $h(x) = \prod_{i=0}^n (x - x_i)$ gilt $\langle h, z \rangle_w = 0$ für alle $z \in \mathcal{P}_n$.*

Beweis. Zum Beweis der einen Richtung sei $z \in \mathcal{P}_n$. Wegen $h \in \mathcal{P}_{n+1}$ gilt $hz \in \mathcal{P}_{2n+1}$. Also gilt nach Voraussetzung $\mathcal{I}_w(hz) = \mathcal{I}_w^n(hz)$, d.h.

$$0 = \sum_{i=0}^n \alpha_i h(x_i) z(x_i) = \int_a^b h(x) z(x) w(x) \ \mathrm{d}x = \langle h, z \rangle_w.$$

Für den Beweis der Rückrichtung dividieren wir ein beliebiges $f \in \mathcal{P}_{2n+1}$ durch $h \in \mathcal{P}_{n+1}$ mit Rest und erhalten

$$f(x) = h(x) z(x) + y(x) \quad z, y \in \mathcal{P}_n,$$

woraus mit der Voraussetzung $\langle h, z \rangle_w = 0$, d.h. $\mathcal{I}_w(hz) = 0$, die Gleichung

$$\mathcal{I}_w(f) = \mathcal{I}_w(hz) + \mathcal{I}_w(y) = \mathcal{I}_w(y)$$

folgt. Da $h(x_i) = 0$, $i = 0, \ldots, n$, ist $y(x)$ gerade das Interpolationspolynom von f, d.h.

$$\mathcal{I}_w(f) = \mathcal{I}_w(y) = \sum_{i=0}^{n} \alpha_i y(x_i) = \sum_{i=0}^{n} \alpha_i f(x_i) = \mathcal{I}_w^n(f).$$

□

Bemerkung: Aus dem obigen Satz ergibt sich, daß h gerade das Polynom $(n+1)$-ten Grades ist, das orthogonal auf allen Polynomen n-ten Grades ist. Die Nullstellen des Orthogonalpolynoms $p_{n+1}(x)$ zum Gewicht $w(x)$ in $[a, b]$ sind gerade die Stützstellen der entsprechenden Gaußquadratur.

Nun müssen also die Nullstellen der entsprechenden orthogonalen Polynome berechnet werden.

Im allgemeinen gibt es aber dafür keine expliziten Formeln. Die Nullstellen lassen sich aber als Eigenwerte einer tridiagonalen Matrix numerisch stabil berechnen. Dazu nutzen wir den folgenden Satz:

Satz 4.3. *Es seien $p_j(x) \in \mathcal{P}_{j,1}$, $j = 0, \ldots, n$, eine Familie orthogonaler Polynome bezüglich des Skalarprodukts $\langle \cdot, \cdot \rangle_w$ mit führendem Koeffizienten 1. Dann gehorchen diese einer dreigliedrigen Rekursionsformel*

$$p_0(x) = 1, \qquad p_1(x) = x - \delta_0, \tag{4.8}$$

$$p_{j+1}(x) = (x - \delta_j) p_j(x) - \beta_j^2 p_{j-1}(x) \quad j \geq 1. \tag{4.9}$$

Beweis. Aufgrund der Voraussetzung $p_j(x) \in \mathcal{P}_{j,1}$ sind die Polynome $p_j(x)$, $j = 0, \ldots, n$ und $x p_n(x)$ linear unabhängig. Deshalb gilt

$$p_{n+1}(x) = \gamma_{n+1} x p_n(x) + \sum_{k=0}^{n} \gamma_k p_k(x). \tag{4.10}$$

Wir multiplizieren nun (4.10) mit $w(x) p_j(x)$, $j \leq n$ und integrieren anschließend von a bis b. Dann erhalten wir

$$\begin{aligned}
\int_a^b p_{n+1}(x) p_j(x) w(x) \ \mathrm{d}x &= \gamma_{n+1} \int_a^b p_n(x) x p_j(x) w(x) \ \mathrm{d}x \\
&\quad + \sum_{k=0}^{n} \gamma_k \int_a^b p_k(x) p_j(x) w(x) \ \mathrm{d}x \\
&= \gamma_{n+1} \int_a^b p_n(x) x p_j(x) w(x) \ \mathrm{d}x + \sum_{k=0}^{n} \gamma_k \delta_{jk} c_k \\
&= \gamma_{n+1} \int_a^b p_n(x) x p_j(x) w(x) \ \mathrm{d}x + \gamma_j c_j, \quad c_j > 0,
\end{aligned}$$

wegen der Orthogonalität der Polynome. Aus den Orthogonalitäten

$$\int_a^b p_{n+1}(x) p_j(x) w(x) \ \mathrm{d}x = 0 \quad j \neq n+1$$

und

$$\int_a^b p_n(x) x p_j(x) w(x) \ \mathrm{d}x = 0 \quad j < n-1$$

folgt nun $\gamma_j = 0$ für $j < n-1$, d.h. wir haben

$$p_{n+1}(x) = \gamma_{n+1} x p_n(x) + \gamma_n p_n(x) + \gamma_{n-1} p_{n-1}(x). \tag{4.11}$$

Da der führende Koeffizient 1 ist, ergibt sich $\gamma_{n+1} = 1$ durch Koeffizientenvergleich. Wegen $p_n(x) - x^n \in \mathcal{P}_{n-1}$ gilt

$$\int_a^b p_n(x) x^n w(x) \ \mathrm{d}x = \int_a^b p_n(x) p_n(x) w(x) \ \mathrm{d}x = c_n > 0$$

und analog

$$\int_a^b p_{n-1}(x) x^{n-1} w(x) \ \mathrm{d}x = c_{n-1} > 0.$$

Multiplizieren wir (4.11) mit $x^{n-1} w(x)$ und integrieren über $[a, b]$, ergibt sich

$$\begin{aligned} 0 &= \int_a^b p_{n+1}(x) x^{n-1} w(x) \ \mathrm{d}x \\ &= \int_a^b x p_n(x) x^{n-1} w(x) \ \mathrm{d}x + \gamma_n \int_a^b p_n(x) x^{n-1} w(x) \ \mathrm{d}x + \gamma_{n-1} \int_a^b p_{n-1} x^{n-1} w(x) \ \mathrm{d}x \\ &= c_n + \gamma_{n-1} c_{n-1} \end{aligned}$$

und daher $\gamma_{n-1} = -c_n / c_{n-1} =: -\beta_n^2 < 0$. Mit $\delta_n := -\gamma_n$ ergibt sich die Behauptung. □

Die Koeffizienten werden nun in die symmetrische und tridiagonale Matrix

$$T_n = \begin{bmatrix} \delta_0 & \beta_1 & 0 & \dots & 0 \\ \beta_1 & \delta_1 & \beta_2 & 0 & \dots \\ 0 & \beta_2 & \delta_2 & \beta_3 & \\ \vdots & & & \ddots & \beta_{n-1} \\ 0 & \dots & 0 & \beta_{n-1} & \delta_{n-1} \end{bmatrix} \in \mathbb{R}^{n,n}$$

geschrieben. Wir zeigen nun

Satz 4.4. *Sei p_n definiert über* (4.9)*. Dann gilt* $\det(xI - T_n) = p_n(x)$*, d.h. die Nullstellen von p_n sind die Eigenwerte von T_n.*

Beweis. Der Beweis erfolgt mit Induktion. Für $n = 1$ und $n = 0$ ist die Behauptung offensichtlich:

$$p_1(x) = \det(xI - T_1) = x - \delta_0.$$

Für $n > 1$ entwicklen wir die Determinante von $xI - T_{n+1}$ nach der letzten Zeile und erhalten mit der Induktionsvoraussetzung

$$\begin{aligned} \det(xI - T_{n+1}) &= \det(xI - T_n)(x - \delta_n) - \beta_n^2 \det(xI - T_{n-1}) \\ &= (x - \delta_n) p_n(x) - \beta_n^2 p_{n-1}(x) = p_{n+1}(x). \end{aligned}$$

□

Der folgende Satz zeigt uns dass bei Gaussquadratur nur nichtnegative Gewichte entstehen und daher Konvergenz für $n \to \infty$ vorliegt.

Satz 4.5. *Seien $x_0, \ldots, x_n$ die Stützstellen der Gaußquadratur zur Gewichtsfunktion w auf $[a, b]$. Dann gilt für die Gewichte*

$$\alpha_i = \int_a^b L_i(x)^2 w(x)\, \mathrm{d}x > 0,\ i = 0, \ldots, n,$$

wobei L_i das Lagrangepolynom (3.1) bezeichnet.

Beweis. Das Lagrangepolynom L_i ist eine Polynom aus $\mathcal{P}_n$ und daher gilt $L_i(x)^2 \in \mathcal{P}_{2n}$. Damit ist die Integrationsformel exkat und wir erhalten wegen $L_i(x_j) = \delta_{ij}$ die behauptete Gleichung

$$\int_a^b L_i(x)^2 w(x)\, \mathrm{d}x = \sum_{j=0}^{n} \alpha_j L_i(x_j)^2 = \alpha_i.$$

□

Bemerkung: Die Gewichte α_i lassen sich aus der ersten Komponente des zum Eigenwert x_i gehörigen Eigenvektors $v^{(i)}$ berechnen:

$$\alpha_i = (v_1^{(i)})^2,\ i = 0, \ldots, n,$$

wobei ${v^{(i)}}^T v^{(i)} = \int_a^b w(x)\ \mathrm{d}x$. Die Berechnung von Eigenwerten und Eigenvektoren erfolgt nun numerisch stabil mit dem QR-Algorithmus.

Aus der Theorie orthogonaler Polynome lassen sich für verschiedene Gewichte entsprechende orthogonale Funktionen konstruieren:

a	b	$w(x)$	Name	Formel
-1	1	$(1-x)^\alpha(1+x)^\beta$	Jacobi-P.	$P_n^{(\alpha,\beta)}(x) = \frac{1}{2^n n! w(x)} \frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(w(x)(x^2-1)^n\right)$
0	∞	e^{-x}	Laguerre-P.	$\ell_n(x) = \frac{1}{n! e^{-x}} \frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(e^{-x}x^n\right)$
$-\infty$	∞	e^{-x^2}	Hermite-P.	$H_n(x) = \frac{(-1)^n}{e^{-x^2}} \frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(e^{-x^2}\right)$

Der praktische interessanteste Fall sind dabei die Jacobi-Polynome. Im Falle $\alpha = \beta = 0$ erhält man dann die Legendreschen Polynome $L_n(x) = P_n^{(0,0)}(x)$, für $\alpha = \beta = -\frac{1}{2}$ die Tschebyscheffschen Polynome 1. Art $T_n(x) = P_n^{(-0.5,-0.5)}(x) = \cos n \arccos x$.

Beispiel: Sei $w(x) = 1$ und $[a, b] = [-1, 1]$, d.h.

$$\mathcal{I}(f) = \int_{-1}^{1} f(x)\, \mathrm{d}x \approx \sum_{i=0}^{n} \alpha_i f(x_i) = \mathcal{I}^n(f).$$

Dann sind die orthogonalen Polynome die Legendrepolynome

$$L_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n$$

mit

$$\int_{-1}^{1} L_n(x) L_m(x)\, \mathrm{d}x = \frac{2}{2m+1}\delta_{nm}.$$

Die entsprechende Rekursionsformel lautet dann

$$(n+1)L_{n+1}(x) = (2n+1)xL_n(x) - nL_{n-1}(x)$$

Sie wird auch zur numerisch stabilen Berechnung von Funktionswerten genutzt. Durch entsprechendes Skalieren der Legendrepolynome mit $\binom{2n}{n} 2^{-n}$ gewinnt man daraus eine Formel der Form (4.9) mit $\delta_i = 0$ und $\beta_i = \frac{i}{\sqrt{4i^2-1}}$.

Gewöhnlich benutzt man die Integrationsformeln (speziell die Legendreformeln) nicht für das gesamte Intervall $[a,b]$ für große n, sondern man unterteilt das Intervall $[a,b]$ in eine Reihe kleinerer Intervalle und wendet die Formel (für kleines bzw. mittleres n) auf jedes der Teilintervalle an.

4.3 Integration im Mehrdimensionalen

Bisher haben wir uns mit der Integration auf dem Interval (a,b) beschäftigt. Viele praktische Probleme erfordern aber die näherungsweise Berechnung von

$$\mathcal{I}(f) = \int_\Omega f(x)\, \mathrm{d}x \quad \text{mit } \Omega \subset \mathbb{R}^d \text{ beschänkt, } f : \Omega \mapsto \mathbb{R}.$$

Falls $d = 2$ ist, dann kann Ω oftmals in Dreiecke oder Vierecke zerlegt werden. Sei deshalb o.B.d.A. Ω ein Dreieck oder Viereck. Zunächst transformiert man Ω mit einer affin linearen oder bilinearen Transformation auf ein Referenzgebiet, d.h.

- auf das Quadrat $Q = [-1,1]^2$ bei Vierecken,
- auf das Referenzdreieck D mit den Ecken $(-1,-1)$, $(1,-1)$ und $(0,1)$ bei Dreiecken.

Wir betrachten zunächst $\Omega = Q$. Dann ist

$$\mathcal{I}(f) = \int_{-1}^{1}\int_{-1}^{1} f(x,y)\, \mathrm{d}x\, \mathrm{d}y.$$

Wir wenden nun die 1D-Quadraturformel $\mathcal{I}^n(\cdot)$ (Gauß oder Newton-Cotes Formel) in beide Koordinatenrichtungen an und erhalten

$$\begin{aligned}
\mathcal{I}(f) = \int_{-1}^{1}\int_{-1}^{1} f(x,y)\, \mathrm{d}x\, \mathrm{d}y &\approx \int_{-1}^{1} \sum_{i=0}^{n} f(x_i,y)\alpha_i\, \mathrm{d}y \\
&= \sum_{i=0}^{n} \alpha_i \int_{-1}^{1} f(x_i,y)\, \mathrm{d}y \\
&\approx \sum_{i=0}^{n}\sum_{j=0}^{n} f(x_i,y_j)\alpha_i\alpha_j,
\end{aligned}$$

wobei x_i, y_i die Stützstellen sind.

Damit erhalten wir eine entsprechende Formel für das Quadrat.

Sei nun $\Omega = D$. Für kleines n, d.h. $n \leq 3$ lassen sich entsprechende Gaußformeln explizit konstruieren, Manchmal benötigt man jedoch Formeln, welche exakt für $\mathcal{P}_n$ mit großem n sind. Es gibt jedoch dafür kein Konstruktionsprinzip für allgemeines n. Als Alternative nutzt man dann die Duffy-Transformation, die das Dreieck D mit der Substitution $z = \frac{2x}{1-y}$ auf das Einheitsquadrat abbildet, d.h.

$$\begin{aligned}
\int_D f(x,y)\,\mathrm{d}(x,y) &= \int_{-1}^{1}\int_{\frac{y-1}{2}}^{\frac{1-y}{2}} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{-1}^{1}\int_{-1}^{1} f\left(\frac{z(1-y)}{2}, y\right)\,\mathrm{d}z\frac{1-y}{2}\,\mathrm{d}y \\
&= \int_{-1}^{1}\int_{-1}^{1} g(y,z)\,\mathrm{d}y\,\mathrm{d}z
\end{aligned}$$

mit $g(y,z) = f\left(\frac{z(1-y)}{2}, y\right)\frac{1-y}{2}$. Nun wendet man eine Quadraturformel auf $g(y,z)$ an und erhält eine gute Quadraturformel für hohe Genauigkeiten.

Kapitel 5

Approximation

Im Gegensatz zur Interpolation untersucht die Approximation die Annäherung einer gegebenen Funktion f in einer Norm $\|\cdot\|$, d.h.

$$\text{Suche } u = \text{argmin}_{v\in\mathbb{M}}\|v - f\|, \tag{5.1}$$

wobei $f \in \mathbb{V}$ eine beliebige Funktion und $\mathbb{M} \subset \mathbb{V}$ ein (endlichdimensionaler) Teilraum ist. Falls ein Skalarprodukt mit $\|\cdot\|_{\mathbb{V}}^2 = \langle\cdot,\cdot\rangle$ exisitiert, d.h. $\mathbb{V}$ ist ein Hilbertraum, dann läßt sich die gesuchte Funktion u einfach berechnen. Sei Pf die orthogonale Projektion von f bzgl. $\langle\cdot,\cdot\rangle$, d.h.

$$\langle Pf - f, v\rangle = 0 \quad \forall v \in \mathbb{M}. \tag{5.2}$$

Dann ist wegen (5.2)

$$\begin{aligned}
\|u - f\|^2 &= \|u - Pf + Pf - f\|^2 \\
&= \langle u - Pf + Pf - f, u - Pf + Pf - f\rangle \\
&= \langle u - Pf, u - Pf\rangle + \langle Pf - f, Pf - f\rangle \\
&= \|u - Pf\|^2 + \|Pf - f\|^2.
\end{aligned}$$

Da der zweite Summand unabhängig von u ist und $\|\cdot\|^2 \geq 0$, folgt $\|Pf - f\| \leq \|u - f\|$ für alle $u \in \mathbb{M}$. Damit ist $\|u - f\|$ minimal, falls $u - Pf = 0$, d.h. $u = Pf$.

Sei nun $[\Phi] = [\phi_1, \ldots, \phi_n]$ eine Basis von $\mathbb{M}$. Dann ist

$$u = [\Phi]\underline{u} = \sum_{i=1}^{n} u_i\phi_i \quad u_i \in \mathbb{R}.$$

Aus (5.2) folgt nun

$$\begin{aligned}
\langle\sum_{j=1}^{n} u_j\phi_j, \phi_i\rangle &= \langle f, \phi_i\rangle \quad \forall i = 1, \ldots, n \\
\Leftrightarrow \sum_{j=1}^{n}\langle\phi_j, \phi_i\rangle u_j &= \langle f, \phi_i\rangle \quad \forall i = 1, \ldots, n \\
\Leftrightarrow G\underline{u} &= \underline{f} \quad \text{mit } G = [\langle\phi_j, \phi_i\rangle]_{i,j=1}^{n},\ \underline{f} = [\langle f, \phi_i\rangle]_{i=1}^{n},
\end{aligned} \tag{5.3}$$

d.h. es ist ein lineares Gleichungssystem zu lösen. Die Matrix G ist stets symmetrisch und positiv definit und heißt Gramsche Matrix.

Beispiel: Wir setzen $\langle u, v\rangle = \int_0^1 u(x)v(x)\,\mathrm{d}x$ und wählen den Raum der Polynome maximal $n-1$-ten Grades als Approximationsraum, d.h. $\mathbb{M} = \mathcal{P}_{n-1}$, mit der monomialen Basis $[\Phi] = [1, x, x^2, \ldots, x^{n-1}]$. Eine einfache Rechnung liefert nun die Gramsche Matrix

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}.$$

Diese Matrix, auch unter dem Namen Hilbertmatrix bekannt, ist aber extrem schlecht konditioniert, z.B. für $n = 5$ liegt die Konditionszahl bei $4.7 \cdot 10^5$, für $n = 10$ schon bei $1.6 \cdot 10^{13}$. Die Basis der Monome ist also ungeeignet.

Zur Vermeidung der Lösung komplizierter Gleichungssysteme kann man eine orthogonale Basis $[\Psi] = [\psi_1, \ldots, \psi_n]$ wählen, d.h. $\langle \psi_i, \psi_j\rangle = \delta_{ij} c_i$. Dann ist $G = \operatorname{diag}[c_i]_{i=1}^n$ und aus (5.3) folgt dann

$$u = Pf = \sum_{i=1}^n \frac{1}{c_i}\langle \psi_i, f\rangle \psi_i = \sum_{i=1}^n \frac{\langle f, \psi_i\rangle}{\langle \psi_i, \psi_i\rangle}\psi_i. \tag{5.4}$$

- Wir betrachten nun wieder $\mathbb{M} = \mathcal{P}_{n-1}$, d.h. die Polynomapproximation, jedoch im gewichteten Skalarprodukt

  $$\langle u, v\rangle = \int_a^b \omega(x)u(x)v(x)\,\mathrm{d}x.$$

  Für verschiedene Gewichte haben wir im Kapitel Integration bereits orthogonale Polynome kennengelernt.

- Ein weiterer Approximationsraum ist der Raum der trigoniometrischen Polynome $\mathbb{M} = \operatorname{span}[1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx]$ mit dem Skalarprodukt

  $$\langle u, v\rangle = \int_{-\pi}^{\pi} u(x)v(x)\,\mathrm{d}x.$$

  Aufgrund der bekannten Orthogonalitätsbeziehungen

  $$\begin{aligned} \int_{-\pi}^{\pi} \cos nx \cos mx\,\mathrm{d}x &= \delta_{m,n}\pi, \quad m^2 + n^2 > 0, \\ \int_{-\pi}^{\pi} \sin nx \sin mx\,\mathrm{d}x &= \delta_{m,n}\pi, \\ \int_{-\pi}^{\pi} \cos nx \sin mx\,\mathrm{d}x &= 0 \end{aligned}$$

  ist die Basis $[1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots, \cos nx, \sin nx]$ orthogonal. Mit (5.4) folgt dann

  $$\begin{aligned} u(t) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\,\mathrm{d}x + \frac{1}{\pi}\sum_{k=1}^n \left(\int_{-\pi}^{\pi} f(x)\cos kx\,\mathrm{d}x\right)\cos kt \\ &\quad + \frac{1}{\pi}\sum_{k=1}^n \left(\int_{-\pi}^{\pi} f(x)\sin kx\,\mathrm{d}x\right)\sin kt. \end{aligned} \tag{5.5}$$

Die Beziehung (5.5) heißt abgebrochene Fourierreihe von f. Fourierreihen spielen bei der Approximation von periodischen Strömen in der Elektrotechnik eine wichtige Rolle.

In allgemeinen Banachräumen ist die Berechnung einer Bestapproximation meist sehr schwierig. In wenigen Spezialfällen lassen sich trotzdem Lösungen gewinnen. Exemplarisch sei der Raum der stetigen Funktionen mit der Norm

$$\|u\|_{C^0} = \max_{x\in[-1,1]} |u(x)|$$

betrachtet. Wir suchen für $f(x) = x^n$ die Bestapproximation in $\mathcal{P}_{n-1}$ bezüglich $\|\cdot\|_{C^0}$. Dazu nutzt man den folgenden Satz.

Satz 5.1. *Sei $T_n(x) = \cos(n \arccos x)$ das n-te Tschebyscheff-Polynom und $\mathcal{P}_{n,1}$ die Menge der Polynome maximal n-ten Grades mit führendem Koeffizienten 1. Dann gilt*

$$\min_{p_n\in\mathcal{P}_{n,1}} \max_{x\in[-1,1]} |p_n(x)| = \max_{x\in[-1,1]} \frac{1}{2^{n-1}}|T_n(x)| = \frac{1}{2^{n-1}}.$$

Beweis. Das n-te Tschebyscheffpolynom besitzt die Extremalstellen $x_k^n = \cos\frac{k\pi}{n}$, $k = 0,\dots,n$ mit $T_n(x_k^n) = (-1)^k$, d.h. das Polynom $\frac{1}{2^{n-1}}T_n(x)$ erfüllt die Behauptung. Angenommen es gibt ein weiteres Polynom $q \in \mathcal{P}_{n,1}$ mit

$$|q(x)| < \frac{1}{2^{n-1}} \quad \forall x \in [-1,1]. \tag{5.6}$$

Dann gilt wegen (5.6)

$$\begin{aligned} q(x_k^n) &< \frac{1}{2^{n-1}}T_n(x_k^n) = \frac{1}{2^{n-1}} \quad k = 2m \\ q(x_k^n) &> \frac{1}{2^{n-1}}T_n(x_k^n) = -\frac{1}{2^{n-1}} \quad k = 2m+1. \end{aligned}$$

Das Polynom $R(x) = q(x) - \frac{1}{2^{n-1}}T_n(x)$ nimmt daher an den Stellen $1 = x_0^n > x_1^n > x_2^n > \ldots > x_n^n = -1$ alternierende Vorzeichen ein, was auf die Existenz von mindestens n verschiedenen Nullstellen schließen läßt. Da $q(x)$ und $\frac{1}{2^{n-1}}T_n(x)$ den führenden Koeffizienten 1 haben, ist $R \in \mathcal{P}_{n-1}$. Damit kann R nur das Nullpolynom sein, was ein Widerspruch ist. □

Mittels des Satzes folgt nun

$$\min_{u\in\mathcal{P}_{n-1}} \|u - f\|_{C^0(-1,1)} = \frac{1}{2^{n-1}} \quad \text{für} \quad u(x) = x^n - \frac{1}{2^{n-1}}T_n(x) \in \mathcal{P}_{n-1}.$$

Eine weitere Anwendung dieses Satzes liegt in der theoretischen Analyse des Verfahrens des konjugierten Gradienten, siehe Vorlesungen Optimierung und Numerik partieller Differentialgleichungen.

Kapitel 6

Nichtlineare Gleichungssysteme

Gegeben sei eine nichtlineare Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$ (*nichtlinear* bedeutet *nicht notwendigerweise linear*). Gesucht ist ein Nullstelle $\bar{x}$ von F, d.h. ein Punkt $\bar{x} \in \mathbb{R}^n$ mit $F(\bar{x}) = 0$, d.h.

$$\begin{aligned} F_1(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) &= 0, \\ F_2(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) &= 0, \\ &\vdots \\ F_n(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) &= 0. \end{aligned}$$

6.1 Das Newtonverfahren

Das Newtonverfahren ist ein iteratives Verfahren. Wir gehen davon aus, dass wir im $k-$ten Iterationsschritt eine Näherung $x^{(k)}$ für die gesuchte Nullstelle zur Verfügung haben und suchen einen neuen Iterationspunkt der Form $x^{(k+1)} = x^{(k)} + p^{(k)}$. Idealerweise würden wir $p^{(k)}$ so wählen, dass $F(x^{(k)} + p^{(k)}) = 0$, was natürlich normalerweise nicht möglich ist. Wir ersetzen daher F in einer Umgebung von $x^{(k)}$ durch eine geeignete Näherung $\tilde{F}^{(k)}(p) \approx F(x^k + p)$ und wählen p^k, sodass $\tilde{F}^{(k)}(p^{(k)}) = 0$, wir suchen also eine Nullstelle der Approximation.

Unter der Annahme dass F stetig differenzierbar ist, gilt

$$F(x^{(k)} + p) = F(x^{(k)}) + F'(x^{(k)})p + \mathcal{O}(\|p\|) \tag{6.1}$$

und eine geeignete Näherung ist durch die affin lineare Funktion

$$\tilde{F}^{(k)}(p) := F(x^{(k)}) + F'(x^{(k)})p$$

gegeben. Wir werden $p^{(k)}$ dann so wählen, dass

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

Ist die Jacobimatrix $F'(x^{(k)}$ regulär ist, besitzt dieses Gleichungssytem eine eindeutige Lösung

$$p^{(k)} = -F'(x^{(k)})^{-1}F(x^{(k)}) \quad \text{(Newtonrichtung)}$$

und der neue Iterationspunkt lautet

$$x^{(k+1)} = x^{(k)} + p^{(k)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}) \tag{6.2}$$

Achtung: Natürlich wird im allgemeinen **nicht!!!** die Inverse $F'(x^{(k)})^{-1}$ berechnet, sondern $p^{(k)}$ als Lösung des linearen Gleichungssytems $F'(x^{(k)})p = -F(x^{(k)})$.

Die Iterationsvorschrift (6.2) heißt Newtonverfahren. Es gelten die folgenden Konvergenzeigenschaften:

Satz 6.1. *Sei $F : \mathbb{R}^n \to \mathbb{R}^n$ einmal stetig differenzierbar, $\bar{x}$ sei eine Nullstelle von F (also $F(\bar{x}) = 0$) und die Jacobimatrix $F'(\bar{x})$ sei regulär. Dann gibt es eine Umgebung U von $\bar{x}$, sodass für jeden Startpunkt $x^{(0)} \in U$ aus dieser Umgebung das Newtonverfahren (6.2) für alle k wohldefiniert ist und die Iterationsfolge (x^k) gegen $\bar{x}$ konvergiert. Für jede Vektornorm bzw. zugehörige Operatornorm gilt die Fehlerabschätzung*

$$\|x^{(k+1)} - \bar{x}\| \leq \frac{2\|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)}{1 - \|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)}\|x^{(k)} - \bar{x}\| \tag{6.3}$$

wobei $\omega(r) := \sup\{\|F'(x) - F'(\bar{x})\| \,|\, \|x - \bar{x}\| \leq r\}$.

Beweis. Sei $\|\cdot\|$ eine beliebige, aber feste Vektornorm. Da $F'(\cdot)$ stetig in $\bar{x}$, gilt $\lim_{r\to 0}\omega(r) = 0$ und daher gibt es einen Radius $R > 0$ mit $\omega(R) < 1/(3\|F'(\bar{x})^{-1}\|) \Rightarrow$

$$\tau := \frac{2\|F'(\bar{x})^{-1}\|\omega(R)}{1 - \|F'(\bar{x})^{-1}\|\omega(R)} < 1$$

Wir zeigen nun mit Induktion, dass für Startpunkte $x^{(0)} \in U_R(\bar{x}) := \{x \,|\, \|x - \bar{x}\| < R\}$ das Newtonverfahren wohldefiniert ist und die Fehlerabschätzung

$$\|x^{(k)} - \bar{x}\| \leq \tau^k\|x^{(0)} - \bar{x}\| \leq \tau^k R < R$$

gilt, woraus wegen $\tau < 1$ sofort $\lim_k x^{(k)} = \bar{x}$ folgt. Sei also $x^{(k)} \in U_R(\bar{x})$. Dann gilt $\|F'(\bar{x})^{-1}\|\|F'(x^{(k)}) - F'(\bar{x})\| < \frac{1}{3} < 1$ und nach dem Satz über benachbarte Inverse ist $F'(x^{(k)})$ regulär und

$$\|F'(x^{(k)})^{-1}\| \leq \frac{\|F'(\bar{x})^{-1}\|}{1 - \|F'(\bar{x})^{-1}\|\|F'(x^{(k)}) - F'(\bar{x})\|} \leq \frac{\|F'(\bar{x})^{-1}\|}{1 - \|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)} \tag{6.4}$$

Damit ist $x^{(k+1)}$ wohldefiniert und es gilt

$$\begin{aligned}
x^{(k+1)} - \bar{x} &= x^{(k)} - \bar{x} - F'(x^{(k)})^{-1}(F(x^{(k)}) - \underbrace{F(\bar{x})}_{=0}) \\
&= F'(x^{(k)})^{-1}\left(F'(x^{(k)})(x^{(k)} - \bar{x}) - (F(x^{(k)}) - F(\bar{x}))\right) \\
&= F'(x^{(k)})^{-1}\left(F'(x^{(k)}) - \int_0^1 F'(\bar{x} + t(x^{(k)} - \bar{x}))\,\mathrm{d}t\right)(x^{(k)} - \bar{x}) \\
&= F'(x^{(k)})^{-1}\left(F'(x^{(k)}) - F'(\bar{x}) - \int_0^1 F'(\bar{x} + t(x^{(k)} - \bar{x})) - F'(\bar{x})\,\mathrm{d}t\right)(x^{(k)} - \bar{x})
\end{aligned}$$

Unter Berücksichtigung von (6.4) erhalten wir die gewünschte Abschätzung (6.3) durch

$$\begin{aligned}
&\|x^{(k+1)} - \bar{x}\| \\
&\leq \|F'(x^{(k)})^{-1}\| \left(\|F'(x^{(k)}) - F'(\bar{x})\| + \int_0^1 \|F'(\bar{x} + t(x^{(k)} - \bar{x})) - F'(\bar{x})\| \, \mathrm{d}t \right) \|x^{(k)} - \bar{x}\| \\
&\leq \frac{\|F'(\bar{x})^{-1}\|}{1 - \|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)} (\omega(\|x^{(k)} - \bar{x}\|) + \int_0^1 \omega(\|x^{(k)} - \bar{x}\|) \, \mathrm{d}t) \|x^{(k)} - \bar{x}\| \\
&\leq \frac{2\|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)}{1 - \|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)} \|x^{(k)} - \bar{x}\|
\end{aligned}$$

Um den Beweis zu vollenden, beachten wir dass $\omega(\cdot)$ monoton wachsend ist und daher

$$\frac{2\|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)}{1 - \|F'(\bar{x})^{-1}\|\omega(\|x^{(k)} - \bar{x}\|)} \leq \frac{2\|F'(\bar{x})^{-1}\|\omega(R)}{1 - \|F'(\bar{x})^{-1}\|\omega(R)} = \tau$$

gilt, woraus mit der Induktionsannahme $\|x^{(k+1)} - \bar{x}\| \leq \tau \|x^{(k)} - \bar{x}\| \leq \tau^{k+1} \|x^{(0)} - \bar{x}\|$ folgt. □

6.1.1 Konvergenzgeschwindigkeit

Bei numerischen Verfahren ist nicht nur wichtig, *ob* eine erzeugte Iterationsfolge gegen eine Lösung konvergiert, sondern auch, *wie schnell* die Konvergenz ist.

Definition 6.1. *Sei $(\gamma^{(k)})$ eine gegen $\bar{\gamma}$ konvergente reelle Zahlenfolge.*

1. *$(\gamma^{(k)})$ heißt (mindestens)* Q-linear *konvergent, wenn*

$$Q_1(\gamma^{(k)}) := \limsup_{k \to \infty} \frac{|\gamma^{(k+1)} - \bar{\gamma}|}{|\gamma^{(k)} - \bar{\gamma}|} < 1.$$

2. *$(\gamma^{(k)})$ heißt* Q-superlinear *konvergent, wenn $Q_1(\gamma^{(k)}) = 0$.*

3. *$(\gamma^{(k)})$ heißt (mindestens)* Q-quadratisch *konvergent, wenn*

$$Q_2(\gamma^{(k)}) := \limsup_{k \to \infty} \frac{|\gamma^{(k+1)} - \bar{\gamma}|}{|\gamma^{(k)} - \bar{\gamma}|^2} < \infty.$$

Bemerkung:

1. $(\gamma^{(k)})$ ist Q-linear konvergent

$$\Leftrightarrow \bigvee_{\substack{c<1 \\ k_0}} \bigwedge_{k \geq k_0} |\gamma^{(k+1)} - \bar{\gamma}| \leq c|\gamma^{(k)} - \bar{\gamma}|$$

2. $(\gamma^{(k)})$ ist Q-quadratisch konvergent

$$\Leftrightarrow \bigvee_{C \in \mathbb{R}} \bigwedge_{k} |\gamma^{(k+1)} - \bar{\gamma}| \leq C|\gamma^{(k)} - \bar{\gamma}|^2$$

3. Eine Q-quadratisch konvergente Folge ist superlinear konvergent

4. $Q_1(\gamma^{(k)})$ (bzw. $Q_2(\gamma^{(k)})$) heißt *Q-Konvergenzfaktor* zur Ordnung 1(bzw. 2), der Buchstabe Q bezieht sich auf das Wort "Quotient".

5. Ist $(x^{(k)}) \subset \mathbb{R}^n$ eine gegen $\bar{x}$ konvergente Folge, so heißt $(x^{(k)})$ Q-linear (bzw. Q-superlinear bzw. Q-quadratisch) konvergent bezüglich einer Norm $\|\cdot\|$, wenn die Zahlenfolge $\|x^{(k)} - \bar{x}\|$ Q-linear (bzw. Q-superlinear bzw. Q-quadratisch) konvergent ist.

Wir betrachten nun die Konvergenzgeschwindigkeit des Newtonverfahrens.
Offensichtlich gilt wegen $\lim_{r\to 0} \omega(r) = 0$

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - \bar{x}\|}{\|x^{(k)} - \bar{x}\|} = 0 \quad \text{(Q-superlineare Konvergenz)}$$

Ist $F'(\cdot)$ in einer Umgebung von $\bar{x}$ Lipschitz stetig mit Konstante L, so gilt $\omega(r) \leq Lr$ und daher für alle $x^{(k)}$ hinreichend nahe bei $\bar{x}$

$$\|x^{(k+1)} - \bar{x}\| \leq \frac{2\|F'(\bar{x})^{-1}\|L}{1 - \|F'(\bar{x})^{-1}\|L\|x^{(k)} - \bar{x}\|}\|x^{(k)} - \bar{x}\|^2 \leq \underbrace{C}_{4\|F'(\bar{x})^{-1}\|L} \|x^{(k)} - \bar{x}\|^2$$

(Q-quadratische Konvergenz). Dies ist insbesonders der Fall, wenn F in $\bar{x}$ zweimal stetig differenzierbar ist.

Neben der Q-Konvergenz ist auch noch die sogenannte *R–Konvergenz* von Bedeutung.

Definition 6.2. *Sei $(\gamma^{(k)})$ eine gegen $\bar{\gamma}$ konvergente reelle Zahlenfolge.*

1. $(\gamma^{(k)})$ *heißt (mindestens)* R-linear *konvergent, wenn*

$$R_1(\gamma^{(k)}) := \limsup_{k\to\infty} |\gamma^{(k+1)} - \bar{\gamma}|^{\frac{1}{k}} < 1.$$

2. $(\gamma^{(k)})$ *heißt* R-superlinear *konvergent, wenn* $R_1(\gamma^k) = 0$.

3. $(\gamma^{(k)})$ *heißt (mindestens)* R-quadratisch *konvergent, wenn*

$$R_2(\gamma^k) := \limsup_{k\to\infty} |\gamma^{(k+1)} - \bar{\gamma}|^{\frac{1}{2^k}} < 1.$$

Eine Folge $(\gamma^{(k)})$ konvergiert also genau dann R-linear (R-quadratisch) gegen $\bar{\gamma}$, wenn es eine Zahl $c > 0$ und ein $q \in [0, 1)$ gibt, sodass

$$|\gamma^{(k)} - \bar{\gamma}| \leq cq^k \quad (|\gamma^{(k)} - \bar{\gamma}| \leq cq^{2^k})$$

1. $R_1(\gamma^{(k)})$ (bzw. $R_2(\gamma^{(k)})$) heißt *R-Konvergenzfaktor* zur Ordnung 1(bzw. 2), der Buchstabe R bezieht sich auf das Wort "Root".

2. Im $\mathbb{R}^n$ ist R-Konvergenz unabhängig von der verwendeten Norm

3. Q-lineare (Q-quadratische) Konvergenz impliziert R-lineare (R-quadratische) konvergenz

6.1.2 Das gedämpfte Newtonverfahren

Satz 6.1 liefert nur die lokale Konvergenz, d.h., der Startwert muss hinreichend nahe bei einer Nullstelle $\bar{x}$ mit regulärer Jacobimatrix $F'(\bar{x})$ sein. Diesen Nachteil wollen wir jetzt beheben.

Beim Newtonverfahren wird die Newtonrichtung $p^{(k)}$ als Nullstelle der Linearisierung $\tilde{F}^{(k)}(p)$ gesucht, diese Linearisierung approximiert $F(x^{(k)}+p)$ aber nur mit einer Genauigkeit von $\mathcal{O}(\|p\|)$. Ist also $p^{(k)}$ groß, so ist die Linearisierung entsprechend ungenau und die Newtonrichtung liefert eventuell keine Annäherung an eine Nullstelle.

Beim gedämpften Newtonverfahren bewegt man sich nun in Newtonrichtung mit einer Schrittweite $\alpha^{(k)} \in (0,1]$, sodass

$$\tilde{F}^{(k)}(\alpha^{(k)}p^{(k)}) = (1-\alpha^{(k)})F(x^{(k)}) \approx F(x^{(k)}+\alpha^{(k)}p^{(k)})$$

gilt.

Wir erhalten die folgenden Iterationsvorschrift für den k-ten Iterationsschritt des gedämpften Newtonverfahrens.

1. Berechne Newtonrichtung $p^{(k)}$ als Lösung des linearen Gleichungssystems $F'(x^{(k)})p = -F(x^{(k)})$.
2. Bestimme $\alpha^{(k)}$ als das 1.Element der Folge $\{1, \frac{1}{2}, \frac{1}{4}, 0.1, 0.033, 0.01, 0.001, 0.0001, \ldots\}$ (oder einer anderen Nullfolge), sodass

   $$\|F(x^{(k)}+\alpha^{(k)}p^{(k)})\| \leq (1-\mu\alpha^{(k)})\|F(x^{(k)})\|,$$

   wobei der Parameter $\mu \in (0,1)$ gegeben ist, z.B. $\mu = 0.1$
3. $x^{(k+1)} := x^{(k)} + \alpha^{(k)}p^{(k)}$

Die Schrittweite $\alpha^{(k)}$ ist wohldefiniert: für beliebiges $\epsilon > 0$ gibt es eine Zahl $\alpha_\epsilon > 0$, sodass

$$\|F(x^{(k)}+\alpha p^{(k)}) - F(x^{(k)}) - \alpha F'(x^{(k)})p^{(k)}\| \leq \epsilon\alpha$$

für $\alpha \in [0, \alpha_\epsilon)$. Damit gilt auch

$$\|F(x^{(k)}+\alpha p^{(k)})\| \leq (1-\alpha)\|F(x^{(k)})\| + \epsilon\alpha$$

für $\alpha \in [0, \alpha_\epsilon)$ und die Existenz der Schrittweite $\alpha^{(k)}$ folgt sofort mit $\epsilon = (1-\mu)\|F(x^{(k)})\|$ und beliebigem $\alpha^{(k)} < \alpha_\epsilon$.

Satz 6.2. *Sei $F : \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar und sei $(x^{(k)})$ eine durch das gedämpfte Newtonverfahren erzeugte Iterationsfolge. Ist die Folge $(p^{(k)})$ der Newtonrichtungen beschränkt, so gilt*

$$\lim_{k\to\infty} F(x^{(k)}) = 0.$$

Insbesonders ist jeder Häufungspunkt der Iterationsfolge $(x^{(k)})$ eine Nullstelle von F.

Beweis. 1. Die Folge $\|F(x^{(k)})\|$ ist monoton fallend und nichtnegativ, also konvergent gegen $\gamma \geq 0$. Wir zeigen $\gamma = 0$ per Widerspruchsbeweis, wir nehmen also an, dass $\gamma > 0$.

2. Wegen $0 < \gamma \leq \|F(x^{(k+1)})\| \leq \prod_{i=0}^{k}(1-\mu\alpha^{(i)})\|F(x^{(0)})\|$ folgt

$$-\infty < \ln\gamma - \ln\|F(x^{(0)})\| \leq \sum_{i=1}^{\infty}\ln(1-\mu\alpha^{(i)}) \leq \sum_{i=0}^{\infty} -\mu\alpha^{(i)}$$

und daher $\sum_{i=0}^{\infty}\alpha^{(i)} =: A < \infty$ und insbesonders auch $\lim_k \alpha^{(k)} = 0$

3. Mit $P := \sup_i \|p^{(i)}\|$ folgt

$$\|x^{(k+1)} - x^{(0)}\| = \|\sum_{i=0}^{k}\alpha^{(i)}p^{(i)}\| \leq \sum_{i=0}^{k}\alpha^{(i)}\|p^{(i)}\| \leq AP,\ \forall k$$

und daher liegen alle Elemente der Folgen $(x^{(k)})$ und $(x^{(k)}+p^{(k)})$ sowie deren Verbindungsstrecken in der kompakten Menge $K := \{x \,|\, \|x - x^{(0)}\| \leq (A+1)P\}$

4. Die stetige Funktion $F'(x)$ ist auf der kompakten Menge K gleichmäßig stetig $\Rightarrow$

$$\exists\delta > 0 \forall x, y \in K, \|x-y\| \leq \delta : \|F'(x) - F'(y)\| \leq \frac{(1-\mu)\gamma}{P}$$

Damit folgt für $\alpha < \frac{\delta}{P}$

$$\begin{aligned}
&\|F(x^{(k)}+\alpha p^{(k)}) - F(x^{(k)}) - \alpha F'(x^{(k)})p^{(k)}\| \\
&= \|\int_0^1 F'(x^{(k)}+t\alpha p^{(k)})\alpha p^{(k)}\,\mathrm{d}t - \alpha F'(x^{(k)})p^{(k)}\| \\
&= \|\int_0^1 F'(x^{(k)}+t\alpha p^{(k)}) - F'(x^{(k)})\,\mathrm{d}t\alpha p^{(k)}\| \leq \frac{(1-\mu)\gamma}{P}\alpha P \leq (1-\mu)\|F(x^{(k)})\|\alpha
\end{aligned}$$

und daher

$$\|F(x^{(k)}+\alpha p^{(k)})\| \leq \|F(x^{(k)}) + \alpha F'(x^{(k)})p^{(k)}\| + (1-\mu)\|F(x^{(k)})\|\alpha = (1-\mu\alpha)\|F(x^{(k)})\|.$$

Unsere Schrittweitenwahl impliziert nun, dass die Folge $\alpha^{(k)}$ nach unten durch eine positive Zahl beschränkt ist im Widerspruch zu $\lim_k \alpha^{(k)} = 0$.

□

Bemerkung: Die Beschränktheit von $\|F'(x^{(k)})^{-1}\|$ ist sicherlich eine hinreichende Bedingung für die Beschränktheit der Folge $(p^{(k)})$. Allerdings ist Satz 6.2 auch in Situationen anwendbar, in denen diese Bedingung nicht erfüllt ist und insbesonders $F'(\bar{x})$ nicht regulär ist.

6.2 Varianten des Newtonverfahrens

Ist $x^{(k)}$ eine Näherung für eine Nullstelle von F, so wird beim Newtonverfahren eine verbesserte Näherung durch Lösen des linearen Ersatzproblems

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

erzeugt. Ein entscheidender Nachteil ist dabei, dass die Jakobimatrix $F'(x^{(k)})$ bekannt sein muss. Manchmal ist jedoch die Auswertung von $F'(x^{(k)})$ sehr aufwändig oder sogar unmöglich

und wir diskutieren nun Varianten, um die exakte Berechnung der Jakobimatrix zu vermeiden. Dabei soll uns an Stelle der exakten Jacobimatrix nur eine Näherung $A^{(k)}$ zur Verfügung stehen und wir betrachten das Ersatzproblem

$$F(x^{(k)}) + A^{(k)}(x - x^{(k)}) = 0.$$

Das resultierende Iterationsverfahren hat dann die Gestalt

$$x^{(k+1)} = x^{(k)} + p^{(k)} \text{ mit } p^{(k)} = -{A^{(k)}}^{-1} F(x^{(k)}) \tag{6.5}$$

Z.B. wird beim sogenannten *vereinfachten Newtonverfahren* $A^{(k)} = A^0$ gesetzt, wobei A^0 eine feste Approximation für $F'(x^{(0)})$ ist. Bei dieser Methode ist der numerische Aufwand natürlich bedeutend geringer als beim Newtonverfahren: Man erspart sich die Berechnung der Jacobimatrix und in jedem Schritt ist jeweils ein Gleichungssystem mit der gleichen Iterationsmatrix zu lösen, d.h. wenn man z.B. ein direktes Verfahren verwendet, ist nur **eine** Zerlegung der Iterationsmatrix erforderlich.

Weiters kann es bei sehr großen Problemen vorkommen, dass wir das Gleichungssystem

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

mittels eines Iterationsverfahren nur näherungsweise lösen (*inexaktes Newtonverfahren*).

6.2.1 Konvergenzanalye

Wir gehen nun davon aus, dass für unsere Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} + p^{(k)} \tag{6.6}$$

die berechnete Suchrichtung $p^{(k)} \approx -{A^{(k)}}^{-1} F(x^{(k)})$ die Bedingung

$$\|F'(x^{(k)})p^{(k)} + F(x^{(k)})\| \leq \eta^{(k)} \|F(x^{(k)})\| \tag{6.7}$$

erfüllt.

Satz 6.3. *Sei $F : \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar, $F(\bar{x}) = 0$ und $F'(\bar{x})$ regulär. Dann gibt es für jedes $\eta \in [0, 1)$ eine Umgebung U_η von $\bar{x}$, sodass für jeden Startpunkt $x^{(0)} \in U_\eta$ die durch* (6.6), (6.7) *erzeugte Folge $(x^{(k)})$ gegen $\bar{x}$ konvergiert, falls $\eta^{(k)} \leq \eta$, $\forall k$.*

Weiters gilt

$$Q_1(\|x^{(k)} - \bar{x}\|_*) = \limsup_{k\to\infty} \frac{\|x^{(k+1)} - \bar{x}\|_*}{\|x^{(k)} - \bar{x}\|_*} \leq \limsup_{k\to\infty} \eta^{(k)}$$

wobei $\|x\|_* := \|F'(\bar{x})x\|$.

Beweis. Sei $\omega_*(R) := \sup\{\|F'(x) - F'(\bar{x})\|_* \mid \|x - \bar{x}\|_* \leq R\}$, wobei per Definition der Operatornorm für eine $n \times n$ Matrix A

$$\|A\|_* = \sup\{\|F'(\bar{x})Ax\| \mid \|F'(\bar{x})x\| \leq 1\} = \|F'(\bar{x})AF'(\bar{x})^{-1}\|$$

gilt. Insbesonders ist also $\|F'(\bar{x})^{-1}\|_* = \|F'(\bar{x})^{-1}\|$.

Für eine beliebige reguläre Matrix A folgt aus

$$F'(\bar{x})A^{-1} = I + (F'(\bar{x}) - A)F'(\bar{x})^{-1}F'(\bar{x})A^{-1}$$

die Beziehung

$$(1 - \|F'(\bar{x})^{-1}\|\|F'(\bar{x}) - A\|_*)\|F'(\bar{x})A^{-1}\| \leq 1. \tag{6.8}$$

Für beliebiges aber festes $\eta \in [0, 1)$ sei nun $U_\eta := \{x \mid \|x - \bar{x}\|_* < R_\eta\}$, wobei R_η so gewählt wird, dass $\|F'(\bar{x})^{-1}\|\omega_*(R_\eta) < \frac{1}{3}$ und

$$\tau := \frac{1 + \|F'(\bar{x})^{-1}\|\omega_*(R_\eta)}{1 - \|F'(\bar{x})^{-1}\|\omega_*(R_\eta)}\eta + \frac{2\|F'(\bar{x})^{-1}\|\omega_*(R_\eta)}{1 - \|F'(\bar{x})^{-1}\|\omega_*(R_\eta)} < 1$$

Sei nun $x^{(k)} \in U_\eta$ für $k \geq 0$. Wegen

$$\begin{aligned} F(x^{(k)}) &= F'(\bar{x})(x^{(k)} - \bar{x}) + \int_0^1 (F'(\bar{x} + t(x^{(k)} - \bar{x})) - F'(\bar{x}))(x^{(k)} - \bar{x})\, \mathrm{d}t \\ &= F'(\bar{x})(x^{(k)} - \bar{x}) \\ &\quad + F'(\bar{x})^{-1} \int_0^1 F'(\bar{x})(F'(\bar{x} + t(x^{(k)} - \bar{x})) - F'(\bar{x}))F'(\bar{x})^{-1}F'(\bar{x})(x^{(k)} - \bar{x})\, \mathrm{d}t \end{aligned} \tag{6.9}$$

folgt

$$\|F(x^{(k)})\| \leq (1 + \|F'(\bar{x})^{-1}\|\omega_*(\|x^{(k)} - \bar{x}\|_*))\|x^{(k)} - \bar{x}\|_*.$$

Weiters ist

$$x^{(k+1)} - \bar{x} = F'(x^{(k)})^{-1}(F'(x^{(k)})p^{(k)} + F(x^{(k)})) + (x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}) - \bar{x})$$

und mit $\tilde{x}^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)})$ folgt aus Satz 6.1, angewandt auf $\|cdot\|_*$, sowie mit (6.8) angewandt auf $A = F'(x^{(k)})$

$$\begin{aligned} \|x^{(k+1)} - \bar{x}\|_* &= \|F'(\bar{x})(x^{(k+1)} - \bar{x})\| \leq \|F'(\bar{x})F'(x^{(k)})^{-1}\|\eta^{(k)}\|F(x^{(k)})\| + \|\tilde{x}^{(k+1)} - \bar{x}\|_* \\ &\leq \frac{1 + \|F'(\bar{x})^{-1}\|\omega_*(\|x^{(k)} - \bar{x}\|_*)}{1 - \|F'(\bar{x})^{-1}\|\omega_*(\|x^{(k)} - \bar{x}\|_*)}\eta^{(k)}\|x^{(k)} - \bar{x}\|_* \\ &\qquad + \frac{2\|F'(\bar{x})^{-1}\|\omega_*(\|x^{(k)} - \bar{x}\|_*)}{1 - \|F'(\bar{x})^{-1}\|\omega_*(\|x^{(k)} - \bar{x}\|_*)}\|x^{(k)} - \bar{x}\|_* \\ &\leq \tau\|x^{(k)} - \bar{x}\|_* < R_\eta \end{aligned}$$

und daher $x^{(k+1)} \in U_\eta$ sowie Q-lineare Konvergenz der Folge $(x^{(k)})$ bezuüglich $\|\cdot\|_*$ gegen $\bar{x}$, wobei sich die Behauptung über den Konvergenzfaktor auch sofort aus obiger Abschätzung ergibt. □

Aus (6.9) ergibt sich auch

$$\lim_{k\to\infty} \frac{\|x^{(k)} - \bar{x}\|_* - \|F(x^{(k)})\|}{\|x^{(k)} - \bar{x}\|_*} = 0$$

und daher

$$Q_1(\|F(x^{(k)})\|) = \limsup_{k\to\infty} \frac{\|F(x^{(k+1)})\|}{\|F(x^{(k)})\|} = \limsup_{k\to\infty} \frac{\|x^{(k+1)} - \bar{x}\|_*}{\|x^{(k)} - \bar{x}\|_*} \leq \limsup_{k\to\infty} \eta^{(k)} \leq \eta$$

Analog zu Satz 6.2 ergibt sich globale Konvergenz für das gedämpfte Newtonverfahren. Allerdings ist für die Durchführbarkeit des gedämpften Newtonverfahrens wesentlich, dass $\eta^{(k)} \leq \eta < 1 - \mu$ gilt, da dann wegen

$$\begin{aligned}\limsup_{\alpha \downarrow 0} \frac{\|F(x^{(k)} + \alpha p^{(k)})\| - (1-\alpha)\|F(x^{(k)})\|}{\alpha} &\leq \limsup_{\alpha \downarrow 0} \frac{\|F(x^{(k)} + \alpha p^{(k)}) - (1-\alpha)F(x^{(k)})\|}{\alpha} \\ &= \|F'(x^{(k)})p^{(k)} + F(x^{(k)})\| \\ &\leq \eta\|F(x^{(k)})\| < (1-\mu)\|F(x^{(k)})\|\end{aligned}$$

für alle k ein $\alpha_0^{(k)} > 0$ existiert, sodass

$$\|F(x^{(k)} + \alpha p^{(k)})\| < (1-\alpha)\|F(x^{(k)})\| + \alpha(1-\mu)\|F(x^{(k)})\| = (1-\mu\alpha)\|F(x^{(k)})\| \ \forall \alpha \in [0, \alpha_0^{(k)}]$$

Ist $F'(x^{(k)})$ nicht bekannt, so können wir natürlich nicht a-priori überprüfen, ob die Bedingung (6.7) erfüllt ist. Bemerken wir jedoch während der Liniensuche, dass die Schrittweite sehr klein wird, so können wir ausnutzen, dass $F'(x^{(k)})p^{(k)} \approx (F(x^{(k)} + \alpha p^{(k)}) - F(x^{(k)}))/\alpha$ für kleine α gilt und damit (6.7) überprüfen und gegebenenfalls eine neue Suchrichtung $p^{(k)}$ berechnen.

Wir untersuchen nun Iterationsvorschriften der Form (6.5), d.h. $F'(x^{(k)})$ ist nicht bekannt sondern nur eine Näherung $A^{(k)}$, aber das resultierende Gleichungssystem wird exakt gelöst.

Es gilt

$$\begin{aligned}x^{(k+1)} - \bar{x} &= x^{(k)} - \bar{x} - {A^{(k)}}^{-1}(F(x^{(k)}) - F(\bar{x})) \\ &= (I - {A^{(k)}}^{-1}F'(\bar{x}))(x_k - \bar{x}) - R^{(k)}(x_k - \bar{x})\end{aligned}$$

mit

$$R^{(k)} := {A^{(k)}}^{-1} \int_0^1 F'(\bar{x} + t(x^{(k)} - \bar{x})) - F'(\bar{x}) \, \mathrm{d}t$$

Falls $\|{A^{(k)}}^{-1}\|$ gleichmäßig beschränkt ist und $x^{(k)}$ gegen $\bar{x}$ kovergiert, gilt $R^{(k)} \to 0$. Damit folgt lineare Konvergenz, falls $\limsup \|I - {A^{(k)}}^{-1}F'(\bar{x})\| < 1$. Insbesonders ergibt sich superlineare Konvergenz, falls $I - {A^{(k)}}^{-1}F'(\bar{x}) \to 0$. Eine hinreichende Bedingung dafür ist $\lim_k A^{(k)} - F'(\bar{x}) = \lim_k A^{(k)} - F'(\bar{x}) = 0$.

Approximiert man also $F'(x^{(k)}$ durch Differenzenquotienten mit immer kleiner werdenden Schrittweiten (z.B. proportional $\|F(x^{(k)})\|$) (siehe nächstes Kapitel), so kann man superlineare bzw. sogar quadratische Konvergenz erreichen.

Wie bereits oben erwähnt, wird beim sogenannten *vereinfachten Newtonverfahren* $A^{(k)} = A^0$ gesetzt, wobei A^0 eine feste Approximation für $F'(x^{(0)})$ ist. Falls der Startpunkt $x^{(0)}$ hinreichend nahe bei einer Nullstelle $\bar{x}$ (mit regulärer Jacobimatrix) liegt und A^0 hinreichend genau $F'(x^{(0)})$ (und damit auch $F'(\bar{x})$) approximiert, so folgt lineare Konvergenz:

$$\limsup_{k \to \infty} \frac{\|x^{(k+1)} - \bar{x}\|}{\|x^{(k)} - \bar{x}\|} \leq \|I - {A^0}^{-1}F'(\bar{x})\| < 1$$

6.2.2 Quasi-Newtonverfahren

Die oben angeführte Bedingung $\lim_k A^{(k)} - F'(\bar{x}) = \lim_k A^{(k)} - F'(\bar{x}) = 0$ ist zwar hinreichend für superlineare Konvergenz, aber nicht notwendig:

Satz 6.4. *Sei $F : \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar, $F(\bar{x}) = 0$ und $F'(\bar{x})$ regulär. Sei weiters $(x^{(k)})$ eine gemäss (6.5) erzeugte Iterationsfolge, die gegen $\bar{x}$ konvergiert. Dann konvergiert $(x^{(k)})$ genau dann superlinear gegen $\bar{x}$, wenn*

$$\lim_{k\to\infty} \frac{\|(A^{(k)} - F'(x^{(k)}))p^{(k)}\|}{\|p^{(k)}\|} = \lim_{k\to\infty} \frac{\|(A^{(k)} - F'(\bar{x}))p^{(k)}\|}{\|p^{(k)}\|} = 0. \tag{6.10}$$

Beweis. Sei $r^{(k)} := (F'(x^{(k)}) - A^{(k)})p^{(k)} = F'(x^{(k)})p^{(k)} + F(x^{(k)})$. Falls (6.10) gilt, so folgt wegen $\|p^{(k)}\| \leq \|F'(x^{(k)})^{-1}\|\|F'(x^{(k)})p^{(k)}\|$ und $\|F'(x^{(k)})^{-1}\| \to \|F'(\bar{x})^{-1}\|$ auch

$$\lim_{k\to\infty} \frac{\|r^{(k)}\|}{\|F'(x^{(k)})p^{(k)}\|} = 0.$$

Aus $\|F'(x^{(k)})p^{(k)}\| \leq \|r^{(k)}\| + \|F(x^{(k)}\|$ folgt

$$\frac{\|r^{(k)}\|}{\|F(x^{(k)})\|} \leq \frac{\|r^{(k)}\|/\|F'(x^{(k)})p^{(k)}\|}{1 - \|r^{(k)}\|/\|F'(x^{(k)})p^{(k)}\|}$$

für hinreichend großes k und daher auch

$$\lim_{k\to\infty} \frac{\|r^{(k)}\|}{\|F(x^{(k)})\|} = 0$$

Die superlineare Konvergenz folgt nun aus Satz 6.3.

Konvergiert umgekehrt $(x^{(k)})$ superlinear gegen $\bar{x}$, so folgt

$$\begin{aligned} 0 &= \lim_{k\to\infty} \frac{\|x^{(k+1)} - \bar{x}\|}{\|x^{(k)} - \bar{x}\|} = \lim_{k\to\infty} \frac{\|x^{(k+1)} - \bar{x}\|}{\|x^{(k)} - \bar{x}\|} \geq \lim_{k\to\infty} \frac{\|F'(x^{(k)})(x^{(k+1)} - \bar{x})\|}{\|F'(x^{(k)})\|\|x^{(k)} - \bar{x}\|} \\ &= \lim_{k\to\infty} \frac{\|F'(x^{(k)})p^{(k)} + F'(x^{(k)})(x^{(k)} - \bar{x})\|}{\|F'(x^{(k)})\|\|x^{(k)} - \bar{x}\|} \end{aligned}$$

Aus (6.9) ergibt sich $\|F(x^{(k)}) - F'(x^{(k)})(x^{(k)} - \bar{x})\| \leq 2\omega(\|x^{(k)} - \bar{x}\|)\|x^{(k)} - \bar{x}\|$ und daher auch

$$0 = \lim_{k\to\infty} \frac{\|F'(x^{(k)})p^{(k)} + F(x^{(k)})\|}{\|x^{(k)} - \bar{x}\|} = \lim_{k\to\infty} \frac{\|(F'(x^{(k)}) - A^{(k)})p^{(k)}\|}{\|x^{(k)} - \bar{x}\|}.$$

Beachten wir nun noch, dass aus supelinearer Konvergenz die Beziehung $\|x^{(k)} - \bar{x}\|/\|x^{(k)} - x^{(k+1)}\| \to 1$ folgt, so ergibt sich (6.10). □

Um superlineare Konvergenz zu erhalten, muss also $A^{(k)}$ die Jacobimatrix $F'(x^{(k)})$ nur entlang der Richtung $p^{(k)}$ mit wachsender Genauigkeit approximieren. Bei den sogenannten *Quasi-Newton*-Verfahren wird eine neue Approximation $A^{(k+1)}$ so bestimmt, dass die sogenannte Quasi–Newtonbedingung (Sekantenbedingung) erfüllt ist:

$$A^{(k+1)}(x^{(k+1)} - x^{(k)}) = F(x^{(k+1)}) - F(x^{(k)}) \ (\approx F'(x^{(k+1)})(x^{(k+1)} - x^{(k)}))$$

Eine einfache Möglichkeit dazu ist die sogenannte *Broydenmethode*:

$$A^{(k+1)} = A^{(k)} + v^{(k)}u^{(k)^T}$$

mit $u^{(k)} = x^{(k+1)} - x^{(k)}$ und $v^{(k)} = (F(x^{(k+1)}) - F(x^{(k)}) - A^{(k)}u^{(k)})/\|u^{(k)}\|_2^2$. Es kann lokale superlineare Konvergenz gezeigt werden. Überdies existieren sogenannte Updateformeln, die ausnützen, dass sich $A^{(k+1)}$ und $A^{(k)}$ nur durch eine Rang-1 Matrix unterscheiden und daher eine Zerlegung mit wenig Aufwand berechnet werden kann.

Kapitel 7

Numerische Differentiation

Wir betrachten hier die folgenden beiden Problemstellungen:

- Gegeben $F : \mathbb{R}^n \to \mathbb{R}^m$ und $\bar{x} \in \mathbb{R}^n$, berechne die Jacobimatrix $F'(\bar{x})$
- Gegeben $f : \mathbb{R}^n \to \mathbb{R}$ und $\bar{x} \in \mathbb{R}^n$, berechne den Gradient $\nabla f(\bar{x})$ und/oder die Hessematrix $\nabla^2 f(\bar{x})$

Numerische Differentiation ist insbesonders wichtig, wenn keine explizite Darstellung der Funktion vorliegt, sondern z.B. die Funktionswerte nur durch ein Computerprogramm ("black box") ausgewertet werden können, wobei eine Funktionsauswertung sehr zeitaufwändig sein kann.

7.1 Der skalare Fall

Wir betrachten in diesem Abschnitt numerische Differentiation für eine Funktion $f : \mathbb{R} \to \mathbb{R}$. Grundlage sind meist sogenannte Differenzenquotienten für eine Schrittweite $h > 0$:

1. *Vorwärtsdifferenzenquotient:*

$$f'(\bar{x}) \approx \Delta_F(f,h) := \frac{f(\bar{x}+h) - f(\bar{x})}{h}$$

2. *Rückwärtsdifferenzenquotient:*

$$f'(\bar{x}) \approx \Delta_B(f,h) := \frac{f(\bar{x}) - f(\bar{x}-h)}{h}$$

3. *Zentraler Differenzenquotient:*

$$f'(\bar{x}) \approx \Delta_C(f,h) := \frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h}$$

4. *Zentraler Differenzenquotient 2.Ordnung*

$$f''(\bar{x}) \approx \Delta_C^2(f,h) := \frac{f(\bar{x}+h) + f(\bar{x}-h) - 2f(\bar{x})}{h^2}$$

Für hinreichend oft differenzierbare Funktionen f erhält man mittels den Darstellungen

$$f(\bar{x}+h) = f(\bar{x}) + f'(\bar{x})h + \frac{1}{2}f''(\bar{x})h^2 + \frac{1}{6}f'''(\bar{x})h^3 + \ldots + \frac{1}{k!}f^{(k)}(\bar{x})h^k + \frac{1}{(k+1)!}f^{(k+1)}(\xi)h^{k+1}$$

$$f(\bar{x}-h) = f(\bar{x}) - f'(\bar{x})h + \frac{1}{2}f''(\bar{x})h^2 - \frac{1}{6}f'''(\bar{x})h^3 + \ldots + \frac{1}{k!}f^{(k)}(\bar{x})(-h)^k + \frac{1}{(k+1)!}f^{(k+1)}(\eta)(-h)^{k+1},$$

$\xi \in [\bar{x}, \bar{x}+h]$, $\eta \in [\bar{x}-h, \bar{x}]$, die Approximationsfehler

$$\Delta_F(f,h) - f'(\bar{x}) = \frac{1}{2}f''(\xi)h,$$

$$\Delta_B(f,h) - f'(\bar{x}) = -\frac{1}{2}f''(\eta)h,$$

$$\Delta_C(f,h) - f'(\bar{x}) = \frac{1}{12}(f'''(\xi) + f'''(\eta))h^2,$$

$$\Delta_C^2(f,h) - f''(\bar{x}) = \frac{1}{24}(f^{(4)}(\xi) + f^{(4)}(\eta))h^2$$

Für Extrapolation ist interessant, dass bei den zentralen Differenzenquotienten die Entwicklungen

$$\begin{aligned}\Delta_C(f,h) - f'(\bar{x}) &= \frac{1}{6}f'''(\bar{x})h^2 + \frac{1}{5!}f^{(5)}(\bar{x})h^4 + \ldots + \frac{1}{(2m-1)!}f^{(2m-1)}(\bar{x})h^{2m-2} \\ &\quad + \frac{1}{2(2m+1)!}(f^{(2m+1)}(\xi) + f^{(2m+1)}(\eta))h^{2m}\end{aligned}$$

$$\begin{aligned}\Delta_C^2(f,h) - f''(\bar{x}) &= \frac{2}{4!}f^{(4)}(\bar{x})h^2 + \frac{2}{6!}f^{(6)}(\bar{x})h^4 + \ldots + \frac{2}{(2m)!}f^{(2m)}(\bar{x})h^{2m-2} \\ &\quad + \frac{1}{(2m+2)!}(f^{(2m+2)}(\xi) + f^{(2m+2)}(\eta))h^{2m}\end{aligned}$$

nur gerade Potenzen von h auftreten.

Numerisches Differenzieren ist ein schlecht konditioniertes Problem: Ist nämlich $\hat{f}$ eine zur Verfügung stehende Näherung für f und

$$\max|\hat{f}(x) - f(x)| \leq \epsilon_A,$$

$$\max|f^{(k)}(x)| \leq M_k$$

wobei das Maximum über alle x "nahe" bei $\bar{x}$ genommen wird, so gelten die Fehlerschranken

$$|\Delta_F(\hat{f},h) - f'(\bar{x})| \leq |\Delta_F(\hat{f},h) - \Delta_F(f,h)| + |\Delta_F(f,h) - f'(\bar{x})| \leq \frac{2\epsilon_A}{h} + \frac{M_2 h}{2}$$

$$|\Delta_C(\hat{f},h) - f'(\bar{x})| \leq |\Delta_C(\hat{f},h) - \Delta_C(f,h)| + |\Delta_C(f,h) - f'(\bar{x})| \leq \frac{\epsilon_A}{h} + \frac{M_3 h^2}{6}$$

$$|\Delta_C^2(\hat{f},h) - f''(\bar{x})| \leq |\Delta_C^2(\hat{f},h) - \Delta_C^2(f,h)| + |\Delta_C({}^2f,h) - f''(\bar{x})| \leq \frac{4\epsilon_A}{h^2} + \frac{M_4 h^2}{12}$$

Der Datenfehler wird also für kleines h sehr stark verstärkt, für großes h hat man dagegen einen großen Verfahrensfehler. Man versucht nun, die Schrittweite h so zu wählen, dass die Fehlerschranke minimiert wird:

$$h_F = 2\sqrt{\frac{\epsilon_A}{M_2}} \;\Rightarrow\; |\Delta_F(\hat{f}, 2\sqrt{\frac{\epsilon_A}{M_2}}) - f'(\bar{x})| \leq 2\sqrt{M_2\epsilon_A}$$

$$h_C = \sqrt[3]{\frac{3\epsilon_A}{M_3}} \;\Rightarrow\; |\Delta_C(\hat{f}, \sqrt[3]{\frac{3\epsilon_A}{M_3}}) - f'(\bar{x})| \leq \sqrt[3]{\frac{9M_3}{8}\epsilon_A^2}$$

$$h_{C^2} = \sqrt[4]{\frac{48\epsilon_A}{M_4}} \;\Rightarrow\; |\Delta_C^2(\hat{f}, \sqrt[4]{\frac{48\epsilon_A}{M_4}}) - f''(\bar{x})| \leq \sqrt{\frac{4M_4}{3}\epsilon_A}$$

Die Genauigkeit, mit der man erste Ableitungen berechnen kann, ist bei Verwendung des Vorwärtsdiffernzenquotienten bei $\mathcal{O}(\epsilon_A^{\frac{1}{2}})$, bei Verwendung des zentralen Differenzenquotienten dagegen bei $\mathcal{O}(\epsilon_A^{\frac{2}{3}})$, in keinem Fall jedoch proportional zu ϵ_A. Zur Erreichung dieser Genauigkeiten muss man jedoch die Schrittweite geschickt wählen. Der nachfolgende Algorithmus zeigt, wie dies geschehen kann, wobei wir die folgende Konditionszahl verwenden:

$$cond_{C^2}(\Phi, h) := \frac{4\epsilon_A}{h^2|\Phi|}$$

Algorithmus 7.1. *geeignete Schrittweite für den Vorwärtsdifferenzenquotienten*
Input: $\hat{f}$, $\bar{x}$, ϵ_A, *Konstanten* $\eta, \omega \in \mathbb{R}_+$ *und* $K \in \mathbb{N}$
Output: *Schrittweiten* h_F, h_Φ, *Näherungen* $\varphi := \Delta_F(\hat{f}, h_F) \approx f'(\bar{x})$, $\Phi := \Delta_C^2(\hat{f}, h_\Phi) \approx f''(\bar{x})$, *Fehlerflag* $flag$ *(*$flag = 0$ *bedeutet erfolgreiche Berechnung)*

{ $h_\Phi = 20 * (\eta + |\bar{x}|) * \sqrt{\epsilon_A/(\omega + |f(\bar{x})|)}$, $\Phi = \Delta_C^2(\hat{f},\ h_\Phi,)$, $flag = 0$, $k = 0$;
  `if` $(cond_{C^2}(\Phi, h_\phi) < 0.001)$
  { `do`
      $h_{old} = h_\Phi$, $\Phi_{old} = \Phi$, $h_\Phi = h_\Phi/10$, $\Phi = \Delta_C^2(\hat{f},\ h_\Phi)$, $k = k + 1$;
    `while` $(cond_{C^2}(\Phi, h_\Phi) < 0.001$ `&&` $k \leq K)$;
    `if` $(cond_{C^2}(\Phi, h_\Phi) < 0.001)$
      $flag = 1$;
    `else if` $(cond_{C^2}(\Phi, h_\Phi) > 0.1)$
      $h_\Phi = h_{old}$, $\Phi = \Phi_{old}$;
  }
  `else if` $(cond_{C^2}(\Phi, h_\Phi) > 0.1)$
  { `do`
      $h_\Phi = 10 * h_\Phi$, $\Phi = \Delta_C^2(\hat{f},\ h_\Phi)$, $k = k + 1$;
    `while` $(cond_{C^2}(\Phi, h_\Phi) > 0.1$ `&&` $k \leq K)$;
    `if` $(cond_{C^2}(\Phi, h_\Phi) > 0.1)$
      $h_F = 10 * h_\Phi$, $\varphi = \Delta_F(\hat{f}, h_F)$, $flag = 2$, `return`;
  }
  $h_F = 2 * \sqrt{\epsilon_A/|\Phi|}$, $\varphi = \Delta_F(\hat{f}, h_F)$;
}

In diesem Algorithmus wird zuerst eine Näherung Φ für die 2.Ableitung $f''(\bar{x})$ berechnet, wobei man sich mit einer Genauigkeit von 1–3 signifikanten Ziffern zufrieden gibt. Dafür versucht man die entsprechende Schrittweite h_Φ möglichst klein zu halten, um einen kleinen

Verfahrensfehler zu erhalten. Die Wahl des Startwerts für h_Φ ergibt sich aus der Annahme, dass eine Beziehung der Form

$$|f''(x)| = \mathcal{O}(\frac{\omega + |f(x)|}{(\eta + |x|)^2})$$

gilt (in der Praxis haben sich die Werte $\omega = \eta = 0.1$ gut bewährt). Für K empfiehlt sich eine Wahl von $K = 5$.

Für diesen Algorithmus wird noch der Rundungsfehler ϵ_A benötigt. Eine Möglichkeit, diesen Fehler aus den Funktionswerten abzuschätzen, besteht folgendermaßen: Für eine kleine Schrittweite h (z.B. $h = 10^{-5}(0.1 + |\bar{x}|)$ berechnet man die Funktionswerte $\hat{f}(x_i)$, wobei $x_i = \bar{x} + ih$, $i = 0, \ldots, p$ und schreibt diese Werte in die 1.Spalte einer Differenzentabelle:

$$\begin{array}{llll}
\Delta^0\hat{f}_0 := \hat{f}(x_0) & \Delta^1\hat{f}_0 := \Delta^0\hat{f}_1 - \Delta^0\hat{f}_0 & \ldots & \Delta^k\hat{f}_0 := \Delta^{k-1}\hat{f}_1 - \Delta^{k-1}\hat{f}_0 \\
\Delta^0\hat{f}_1 := \hat{f}(x_1) & \Delta^1\hat{f}_1 := \Delta^0\hat{f}_2 - \Delta^0\hat{f}_1 & \ldots & \Delta^k\hat{f}_1 := \Delta^{k-1}\hat{f}_2 - \Delta^{k-1}\hat{f}_1 \\
\vdots & & \ddots & \\
\Delta^0\hat{f}_{p-1} := \hat{f}(x_{p-1}) & \Delta^1\hat{f}_{p-1} := \Delta^0\hat{f}_p - \Delta^0\hat{f}_{p-1} & & \\
\Delta^0\hat{f}_p := \hat{f}(x_p) & & &
\end{array}$$

Unter der Annahme, dass $\hat{f}(x_i) = f(x_i) + \epsilon_A\delta_i$, wobei δ_i unabhängige Zufallsvariable aus $(-1, 1)$ sind, sind die Elemente der k-ten Spalte (k hibnreichend groß, z.B. $k \geq 4$) ungefähr gleich in Größe, aber besitzen alternierendes Vorzeichen. Es kann dann die folgende Formel herangezogen werden:

$$\epsilon_A \approx \frac{\max_i |\Delta^k\hat{f}_i|}{\sqrt{\binom{2k}{k}}}.$$

7.2 Der allgemeine Fall

Für eine vektorwertige Funktion $F : \mathbb{R}^n \to \mathbb{R}^m$ nähert man die Jacobimatrix $F'(\bar{x})$ ebenfalls durch Differenzenquotienten an. Die i-te Spalte ist

$$F'(\bar{x})e_i \approx \frac{F(\bar{x} + h_ie_i) - F(\bar{x})}{h_i} \text{ bzw. } F'(\bar{x})e_i \approx \frac{F(\bar{x} + h_ie_i) - F(\bar{x} - h_ie_i)}{2h_i}.$$

Normalerweise ist es hier nicht möglich, eine Schrittweite h_i zu finden, sodass für alle Komponenten von F der Fehler möglichst klein wird. Bei einigen Anwendungen, z.B. beim Netonverfahren zur Lösung nichtlinearer Gleichungen, ist dies aber gar nicht nötig und eine Wahl von

$$h_i \approx (0.1 + |x_i|)\sqrt{\epsilon_r} \text{ bzw. } h_i \approx (0.1 + |x_i|)\sqrt[3]{\epsilon_r}$$

für den Vorwärts- bzw. zentralen Differenzenquotienten liefert sehr gute Resultate für das Newtonverfahren (obwohl die Jacobimatrix nicht sonderlich genau approximiert wird). Hier ist ϵ_r eine relative Genauigkeit und in der Praxis zwischen 10^{-16} und 10^{-7}.

Für die Berechnung der Hessematrix $H := \nabla^2 f(\bar{x})$ mittels Funktionswerten können sowohl Vorwärtsdifferenzenquotienten

$$H_{ij} \approx \frac{f(\bar{x} + h_ie_i + h_je_j) - f(\bar{x} + h_ie_i) - f(\bar{x} + h_je_j) + f(\bar{x})}{h_ih_j}$$

als auch zentrale Differenzenquotienten

$$H_{ij} \approx \frac{f(\bar{x} + h_i e_i + h_j e_j) - f(\bar{x} + h_i e_i - h_j e_j) - f(\bar{x} - h_i e_i + h_j e_j) + f(\bar{x} - h_i e_i - h_j e_j)}{4 h_i h_j}$$

herangezogen werden.

7.2.1 Dünnbesetzte Matrizen

Ist die Jacobimatrix *dünn besetzt (sparse)*, wie es bei vielen praktischen Anwendungen der Fall ist, so benötigt man zur Berechnung der Jacobimatrix manchmal sehr viel weniger als n Differenzenquotienten, wenn das Besetztheitsmuster (sparsity pattern) der Jacobimatrix bekannt ist. Sei dazu S eine $m \times n$ Matrix mit Elementen aus $\{0, 1\}$, wobei

$$s_{ij} = 0 \;\Rightarrow\; \forall x : F'_{ij}(x) = 0,$$

d.h. es müssen nur mehr die Elemente $F'_{ij}(\bar{x})$ berechnet werden, für die $s_{ij} = 1$ (*NNE (NZE): Nicht Null Element (Non Zero Element)*). Betrachten wir nun eine allgemeine Richtung $d \in \mathbb{R}^n$ und gilt

$$\forall i : \sum_{\substack{j \\ d_j \neq 0}} s_{ij} \leq 1,$$

dann gilt $F_i(\bar{x} + d) - F_i(\bar{x}) \approx F'_{ij}(\bar{x}) d_j, \forall i, \forall j : d_j \neq 0$. Wir versuchen nun also die Spalten so in Mengen $C_1, \ldots, C_p$ zu partitionieren, sodass für zwei Spalten j_1, j_2 in der gleichen Menge C_k die Beziehung $s_{ij_1} s_{ij_2} = 0, \forall i$ gilt.

Dazu betrachten wir einen Graphen $G = (V, E)$, dessen Knoten V den Spalten entsprechen und und zwei Knoten genau dann durch eine Kante verbunden sind, wenn die entsprechenden Spalten in der gleichen Zeile jeweils ein Nicht-Null- Element besitzen:

$$\{j_1, j_2\} \in E \;\Leftrightarrow\; \exists i : s_{ij_1} s_{ij_2} = 1$$

Ziel ist nun, die Knoten mit möglichst wenig Farben $C_1, \ldots, C_p$ so zu färben, dass jede Kante zwischen verschiedenfärbigen Knoten verläuft. Dann kann die Jacobimatrix mit p Differenzenquotienten $F(\bar{x} + d^k) - F(\bar{x})$ approximiert werden, wobei $d^k_j \neq 0$ genau dann wenn der der Spalte j entsprechende Knoten mit Farbe C_k gefärbt wurde.

Leider ist dieses Problem mit vernünftigem Rechenaufwand nur für "kleine" Graphen exakt lösbar, man behilft sich dabei mit der folgenden *sequential coloring heuristic*:

1. Numeriere die Knoten geeignet mit $v_1, \ldots, v_n$
2. Für $k = 1, 2, \ldots, n$ ordne v_k die Farbe mit der niedrigst möglichen Nummer zu.

Verschiedene Numerierungen ergeben normalerweise unterschiedliche Färbungen. Die folgenden beiden Numerierungen sind normalerweise effizient:

1. *Smallest last ordering:* Seien $v_{k+1}, \ldots, v_n$ bereits gewählt und V_k derjenige Graph, den man durch Streichen der bereits gewählten Knoten $v_{k+1}, \ldots, v_n$ und aller mit diesen Knoten inzidierenden Kanten erhält. v_k wird dann in V_k als ein Knoten mit minimalem Grad (minimaler Anzahl inzidierender Kanten) gewählt.

2. *Incidence degree ordering:* Seien $v_1, \ldots, v_{k-1}$ bereits gewählt. Dann wird v_k als derjenige Knoten gewählt, der mit möglichst vielen Knoten aus $v_1, \ldots, v_{k-1}$ benachbart (d.h. durch eine Kante verbunden) ist.

Zur Berechnung dünnbesetzter Hessematrizen über Gradienten kann die Symmetrie vorteilhaft ausgenutzt werden, siehe Literatur

Kapitel 8

Literatur

- Deuflhard, Peter; Hohmann, Andreas: Numerische Mathematik. I. Eine algorithmisch orientierte Einfhrung. Walter de Gruyter Co., Berlin, 1993.
- Golub, Gene H., Van Loan, Charles F.: Matrix computations. Third edition. Johns Hopkins University Press, Baltimore, MD, 1996.
- Schwarz, H.: Numerische Mathematik. B. G. Teubner, Stuttgart, 1986.
- Stoer, Josef: Numerische Mathematik. 1. Springer-Verlag, Berlin, 1994.
- Stoer, J., Bulirsch, R.: Einfhrung in die numerische Mathematik. II. Heidelberger Taschenbücher, Band 114. Springer-Verlag, Berlin-New York, 1973.