Skriptum zur Vorlesung

Numerische Mathematik für Ingenieure, Physiker und Computational Engineering (CE)

P. Spellucci

WS 2007/2008

HINWEIS:

Dieses Skriptum stellt einen ersten Entwurf für den tatsächlichen Inhalt der Vorlesung in einer sehr knappen, sicher nicht buchreifen Form dar. Es soll nicht das Studium der einschlägigen Lehrbücher ersetzen. Für Hinweise auf Fehler, unklare Formulierungen, wünschenswerte Ergänzungen etc. bin ich jederzeit dankbar. Man bedenke jedoch den Zeitrahmen der Veranstaltung, der lediglich 30 Doppelstunden umfasst, weshalb der eine oder andere Punkt wohl etwas zu kurz kommt oder auch einmal ganz wegfallen muss. Die meisten Beweise der Aussagen, sofern sie überhaupt hier Eingang gefunden haben, werden in der Vorlesung nicht vorgerechnet werden und sind eher für einen interessierten Leser gedacht. An Teilen dieses Skriptums haben auch auch meine Doktoranden und Diplomanden mitgewirkt: Alexandra Witzel , Rolf Felkel, Gerald Ziegler und Thomas Laux. Fast alle in diesem Skript beschriebenen Verfahren können mit unserem interaktiven System NUMAWWW

http://numawww.mathematik.tu-darmstadt.de:8081

erprobt werden, ohne dabei selbst Programme erstellen zu müssen. Diese sind im Text durch ein eingerücktes

NUMAWWW

gekennzeichnet. Teile des Textes sind in die Symbole << und >> eingefasst. Diese sind nur als ergänzende Information gedacht und können bei einer ersten Lektüre übergangen werden. Sie werden auch in der Vorlesung nur ganz kurz skizziert.

Inhaltsverzeichnis

-1.1 Literatur

Diese Liste enthält eine Zusammenstellung aktueller elementarer Lehrbücher über das Gesamtgebiet der Numerischen Mathematik, die für diesen Kurs nützlich sind.

1. M. Bollhöfer, V. Mehrmann: *Numerische Mathematik* Vieweg Verlag 2004.
2. Karl Graf Finck von Finckenstein, Jürgen Lehn, Helmut Schellhaas, Helmut Wegmann: *Arbeitsbuch für Ingenieure*. Band II. Teubner 2002 (Kapitel 3)
3. H.G. Roos, H. Schwetlick: *Numerische Mathemtik. Das Grundwissen für jedermann.* Teubner 1999
4. H. Schwetlick, H. Kretzschmar: *Numerische Verfahren für Naturwissenschaftler und Ingenieure.* Fachbuchverlag Leipzig. 1991.
5. F. Weller: *Numerische Mathematik für Ingenieure und Naturwissenschaftler* Vieweg. 1996.

Eine stärkere mathematische Orientierung weisen die folgenden Bände auf:

1. Stoer, J.: Numerische Mathematik I. Stoer,J. und Bulirsch, R.: Numerische Mathematik II. Auch erhältlich als ein Band in Englisch: Introduction to Numerical Analysis. Springer Verlag
2. Quarteroni, A.; Sacco, R.; Saleri, F. : Numerische Mathematik I. Numerische Mathematik II. Springer Verlag. Diese Bände sind ebenfalls in Englisch erhältlich.

Eine gute Einführung in die Behandlung partieller Differentialgleichungen bieten

1. Quarteroni, Alfio; Valli, Alberto Numerical approximation of partial differential equations. (English) Springer Series in Computational Mathematics. 23. Berlin: Springer-Verlag.
2. Knabner, P.; Angermann, L. Numerik partieller Differentialgleichungen. Eine anwendungsorientierte Einfhrung. Berlin: Springer.
3. Großmann, Christian; Roos, Hans-Grg Numerik partieller Differentialgleichungen. Teubner Studienbcher: Mathematik. Stuttgart: B. G. Teubner.

Kapitel 0

Inhaltsübersicht

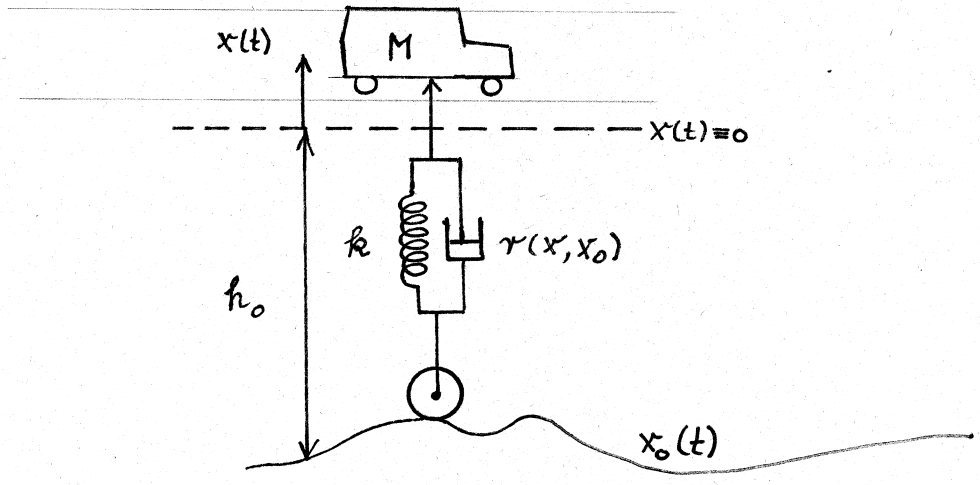
Numerische Mathematik hat die Lösung mathematischer Probleme durch numerisches (Zahlen-)Rechnen zum Thema. Sie ist von grösster Bedeutung für die Ingenieurspraxis, da die wenigsten praktischen Fragestellungen einer analytischen Lösung durch eine geschlossene Formel zugänglich sind. Probleme der Praxis müssen also zunächst erst in ein mathematisches Modell übersetzt werden. Bevor es zu einer numerischen Lösung dieses Problems kommt, muss die Existenz, (lokale) Eindeutigkeit und Stabilität der (einer) Lösung gegen Änderung der Problemdaten (die in der Praxis ja nie exakt bekannt sind) geklärt sein. Wir werden hier stets von einem bereits vorliegenden, wohldefinierten mathematischen Problem ausgehen. Wir beginnen unsere Darstellung mit dem Problem der Annäherung von Funktionen durch Polynome oder stückweise polynomiale Funktionen. Da sich genügend oft differenzierbare Funktionen sehr gut durch solche Funktionen approximieren lassen, andererseits polynomiale Funktionen effizient auswertbar und leicht manipulierbar sind, ist dieser Ansatz grundlegend für die Lösung vieler Probleme, wie z.B. Integration, Differentiation, Lösung von Differentialgleichungen. Es folgt ein Kapitel über die Berechnung bestimmter Integrale. Dieses Problem ist schon für sich alleine praktisch bedeutsam, die entwickelten Methoden spielen aber auch für die Lösung von Differentialgleichungen eine fundamentale Rolle. Es folgt ein Kapitel über die Lösung von Anfangswertproblemen gewöhnlicher Differentialgleichungen. Probleme dieses Typs treten in mannigfacher Form in der Praxis auf, etwa als Gleichungen für Spannungen und Ströme in elektronischen Schaltungen, in der chemischen Reaktionskinetik, als Bewegungsgleichungen diskreter Massensysteme, aber auch als Lösungsschritt bei der Lösung zeitabhängiger partieller Differentialgleichungen. Sodann behandeln wir die Lösung linearer und nichtlinearer Gleichungssysteme durch finite und iterative Verfahren in drei darauffolgenden Kapiteln. Solche Systeme treten regelmässig als Unteraufgaben im Zusammenhang mit der Lösung von Differentialgleichungen auf, aber auch als selbständige Probleme etwa bei der Berechnung statischer Gleichgewichte diskreter Strukturen. Auch das Problem der (linearen) Parameteranpassung wird kurz gestreift. Schliesslich folgt ein Kapitel über Differenzenformeln und ihre Anwendung im numerischen Differenzie-

ren, bei der Lösung von Randwertaufgaben gewöhnlicher Differentialgleichungen und bei den einfachsten Lösungsansätzen für partielle Differentialgleichungen.

Kapitel 1

Interpolation

In diesem Kapitel besprechen wir die einfachsten Methoden zur genäherten Darstellung von Funktionen einer reellen Veränderlichen. Auf den Fall mehrerer Veränderlicher gehen wir nur sehr kurz ein. Wir beschränken uns dabei auf Ansatzfunktionen, die sich zumindest stückweise als Polynome darstellen lassen. Sinn dieser Methoden ist es, "komplizierte Funktionen", deren exakte Berechnung mit endlich vielen arithmetischen Operationen unmöglich ist, oder Funktionen, die nur in Form von diskreten Werten $(x_i, y_i),\ i = 0, \ldots, n$ bekannt sind, durch einfache, leicht manipulierbare Funktionen auf vorgegebenen Teilen ihres Definitionsbereiches so anzunähern, daß die Abweichungen für die Praxis tolerierbar sind. "Leicht manipulierbar" sind nun offensichtlich alle Funktionen, die sich stückweise als Polynome darstellen lassen. Diese Näherungsmethoden werden uns dann später zu Näherungsmethoden für bestimmte Integrale und für Ableitungswerte führen und sind daher auch grundlegend für die numerische Behandlung von gewöhnlichen und partiellen Differentialgleichungen. Die überwiegende Mehrheit technischer Probleme führt auf solche, analytisch nicht lösbaren Aufgaben. Wir beginnen mit einer stark vereinfachten Version eines Problems aus der Automobiltechnik, dem Entwurf eines Federbeins. Die Feder/Dämpfereigenschaften eines solchen Federbeins sollen den Einfluss eines unebenen Fahrbahnprofils auf das Fahrzeug abmildern. Hier berechnen wir zunächst bei gegebener Spezifikation der Feder/Dämpfereigenschaften die Vertikalbewegung des Schwerpunktes, also die Funktion $x(t)$ bei gegebenem Fahrbahnprofil $x_0(t)$ (d.h. implizit auch bei gegebener gleichmässiger Bewegung dieses Schwerpunktes in horizontaler Richtung, siehe Abbildung).



$$M\ddot{x} = -k(x - x_0) - r(x, x_0)(\dot{x} - \dot{x}_0)$$

$$r(x, x_0) = r_0\,(1 + c\,|\dot{x} - \dot{x}_0|)$$

x : Auslenkung des Schwerpunkts vertikal

x_0 : Straßenprofil

$$y_1 = x$$

$$y_2 = \dot{x}$$

$$\dot{y} = \begin{pmatrix} y_2 \\ -\frac{k}{M}(y_1 - x_0) - \frac{r_0}{M}(1 + c\,|y_2 - \dot{x}_0|)(y_2 - \dot{x}_0) \end{pmatrix}$$

$$= F(t, y)$$

Abb 1.1.1 Vertikalbewegung eines Masse/Feder/Dämpfersystems

Wegen der Nichtlinearität des Dämpfergliedes ist diese Differentialgleichung nicht analytisch lösbar. Ausserdem ist in der Praxis x_0 nicht als analytischer Ausdruck, sondern als eine Menge von Messpunkten gegeben, für die erst noch ein analytischer Ausdruck gefunden werden muss. Die gewöhnliche Differentialgleichung zweiter Ordnung schreiben wir dann in der üblichen Weise in ein System erster Ordnung um

$$y' \;=\; F(t, y), \quad y(t_0) = y_0\ .$$

Die häufigste Vorgehensweise zur numerischen Lösung eines solchen Systems besteht in der Umschreibung als Volterra-Integralgleichung

$$y(t) = y(t_0) + \int_{\tau=t_0}^{\tau=t} F(\tau, y(\tau))\mathrm{d}\tau \ ,$$

und der Ersetzung des Integrals durch eine sogenannte Quadraturformel

$$\int_{\tau=t_0}^{\tau=t} F(\tau, y(\tau))\mathrm{d}\tau \ \approx \ \sum_{i=0}^{N} w_i \tilde{F}(t_i) \ ,$$

wobei

$$\tilde{F}(t_i) \ \approx \ F(t_i, y(t_i)) \ .$$

Eine Quadraturformel ist eine Formel, die Polynome eines gewissen maximalen Grades exakt integriert. $\tilde{F}$ entsteht durch Approximation der benötigten Funktionswerte $y(t_i)$ durch Interpolation bereits gegebener Funktionswerte durch ein Polynom und Auswertung dieses Polynoms. Mit den benötigten Rechentechniken werden wir uns in den nächsten drei Kapiteln beschäftigen.

1.1 Polynominterpolation

Dieser Abschnitt beschäftigt sich mit der Interpolation von gegenenen Werten einer Funktion f durch ein Polynom p und der Ersetzung der Auswertung von f durch die des Polynoms.

Aufgabenstellung:
Gegeben seien $n+1$ "Stützpunkte " $(x_i, y_i) \in \mathbb{R}^2, i = 0, \ldots, n$ mit $x_i \neq x_j$ für $i \neq j$.
$\otimes$ Gesucht $p_n \in \prod_n$ (=Menge aller Polynome vom Höchstgrad n) mit

$$p_n(x_i) \ = \ y_i, \quad i = 0, \ldots, n.$$

Man bezeichnet in diesem Zusammenhang die x_i als Stützstellen und die y_i als Stützwerte. Im Prinzip könnte man diese Aufgabe auf die Lösung eines linearen Gleichungssystems zurückführen. Dies wäre aber sehr ungeschickt, sowohl wegen des erhöhten Rechenaufwandes als auch wegen des sehr viel ungünstigeren Einflusses der unvermeidlichen Rundungsfehler. Ein erster einfacher Lösungsweg besteht in der Konstruktion von Polynomen vom genauen Grad n, die an genau einer der Stellen x_j den Wert 1 und an allen anderen Stellen x_k den Wert null haben und deren additiver Überlagerung:

Satz 1.1.1. Interpolationspolynom nach Lagrange: *Die Interpolationsaufgabe $\otimes$ hat genau eine Lösung. Diese kann dargestellt werden als*

$$p_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

mit

$$L_i(x) \ = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Beweis: Offenbar gilt $L_i(x_i) \ = \ 1, \quad L_i(x_k) \ = \ 0 \qquad \text{für} \qquad i \neq k$. Also

$$p_n(x_k) \ = \ \sum_{i=0}^{n} y_i \delta_{ik} = y_k$$

Annahme: $\exists p_n^\star, \quad p_n^{\star\star} \in \Pi_n$ mit $\otimes$, $p_n^\star \not\equiv p_n^{\star\star}$. Dann ist
$\widetilde{p}_n \stackrel{def}{=} p_n^\star - p_n^{\star\star} \not\equiv 0, \in \Pi_n$ und $\widetilde{p}_n(x_j) \ = \ 0, \quad j = 0, \ldots, n.$
Da die x_j paarweise verschieden sind, erhält man einen Widerspruch, denn ein Polynom vom Höchstgrad n hat höchstens n verschiedene Nullstellen oder verschwindet identisch. □

Beispiel 1.1.1.

$$(x_i, y_i) \ = \ \{(-1, 1), \ (0, 1), \ (1, 3)\} \quad n \ = \ 2 \ .$$

Die Lösung nach Lagrange lautet dann explizit ausgeschrieben

$$1\frac{(x-1)x}{(-1-1)(-1)} + 1\frac{(x+1)(x-1)}{(0+1)(0-1)} + 3\frac{(x+1)x}{(1-(-1))(1-0)} \ .$$

Bemerkung 1.1.1. *Dieser Satz besagt u.a., daß man ein Polynom gleichwertig durch seine Taylorentwicklung in 0 oder einen Satz von $n+1$ Funktionswerten repräsentieren kann, oder, anders ausgedrückt, daß die Interpolation von Werten eines Polynoms vom Höchstgrad n durch ein Polynom vom Höchstgrad n dieses Polynom exakt reproduziert.*

Bemerkung 1.1.2. *Die oben eingeführten Polynome $L_i(x)$ heißen die* **Lagrangeschen Grundpolynome**. *Es sind Polynome vom genauen Grad n, die eine Basis von Π_n bilden, denn nach dem obigen Satz kann man jedes Polynom vom Höchstgrad n linear aus diesen kombinieren. Eigentlich müßte man die Abhängigkeit der L_i von n und von $\{x_i\}$ kennzeichnen, doch verzichtet man aus Gründen der Übersichtlichkeit normalerweise darauf.*

In der folgenden Abbildung ist $n = 3$ und $\{x_i\} = \{1, 2, 3, 4\}$.

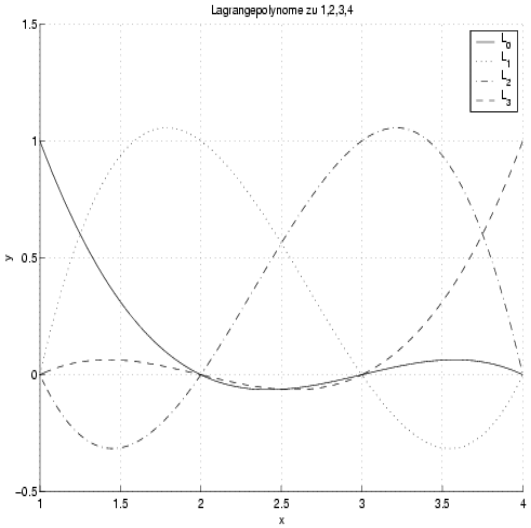


Abb 1.1.2

Das Interpolationspolynom hat den **Höchstgrad** n, nicht den genauen Grad n.
(Beispiel: $y_0 = y_1 = \ldots = y_n = 1 \Rightarrow p_n(x) \equiv 1$)

Für theoretische Zwecke ist diese Darstellung nach Lagrange sehr nützlich. Für das praktische Rechnen jedoch ist der folgende Zugang wesentlich angenehmer.

Wir wählen als Ansatz die **Newtonsche Darstellung** des Interpolationspolynoms

$$p_n(x) = \gamma_0 + \gamma_1(x-x_0) + \gamma_2(x-x_0)(x-x_1) + \cdots + \gamma_n(x-x_0)(x-x_1)\cdot\ldots\cdot(x-x_{n-1}) \quad (1.1)$$

Es folgt nun aus den Interpolationsbedingungen, daß gelten muß

$$\begin{aligned}
\gamma_0 &= y_0 \\
\gamma_1 &= \frac{y_1 - y_0}{x_1 - x_0} \\
\gamma_2 &= \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} \\
&\vdots
\end{aligned}$$

Die Koeffzienten γ_i, die hier auftreten, bezeichnet man mit

$$\gamma_i =: f_{[x_0,\ldots,x_i]}$$

als die i-te **dividierte Differenz** zu den Stützstellen $x_0, \ldots, x_i$. Hierbei ist $\gamma_0 = y_0 = f_{[x_0]}$. Allgemein berechnet sich die dividierte Differenz zu den Stützstellen $x_0, \ldots, x_{i+1}$ rekursiv über

$$f_{[x_0,\ldots,x_{i+1}]} = \frac{f_{[x_1,\ldots,x_{i+1}]} - f_{[x_0,\ldots,x_i]}}{x_{i+1} - x_0}$$

mit der Initialisierung $f_{[x_i]} = y_i,\ i = 0,\ldots,n$.

Beispiel 1.1.2.

$$\begin{aligned} f_{[x_0,x_1]} &= \frac{f_{[x_1]} - f_{[x_0]}}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0} = \gamma_1 \\ f_{[x_0,x_1,x_2]} &= \frac{f_{[x_1,x_2]} - f_{[x_0,x_1]}}{x_2 - x_0} = \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} = \gamma_2 \end{aligned}$$

Es läßt sich nun folgende **allgemeine Rekursion** angeben, die aus den Ausgangsdaten (x_i, y_i) die Werte $\gamma_i = f_{[x_0,\ldots,x_i]}$ berechnet: Hier werden die oben benutzten γ_i als $\gamma_{0,i}$ bezeichnet

Schema der dividierten Differenzen:

$$\gamma_{j,k} = \frac{\gamma_{j+1,k-1} - \gamma_{j,k-1}}{x_{j+k} - x_j} \qquad j = 0,\ldots,n-k, \quad k = 1,\ldots,n$$

mit

$$\gamma_{j,0} = y_j \qquad j = 0,\ldots,n$$

Bemerkung 1.1.3. *$\gamma_{i,k}$ ist der Höchstkoeffizient des Polynoms vom Höchstgrad k, das die Punkte $(x_i, y_i),\ \ldots\ ,(x_{i+k}, y_{i+k})$ interpoliert.* □

Zur Veranschaulichung betrachten wir folgendes

Beispiel 1.1.3. *Gegeben seien die folgenden Daten*

i	0	1	2	3
x_i	-1	0	2	3
y_i	-1	1	5	4

Hier ist also $n = 3$ und wir berechnen

i	x_i	$(k=0)$ $y_i = f_{[x_i]}$	$(k=1)$ $f_{[x_i,x_{i+1}]}$	$(k=2)$ $f_{[x_i,x_{i+1},x_{i+2}]}$	$(k=3)$ $f_{[x_i,x_{i+1},x_{i+2},x_{i+3}]}$
0	-1	$\boxed{-1 = \gamma_0}$	$\frac{1-(-1)}{0-(-1)} = \boxed{2 = \gamma_1}$		
1	0	1	$\frac{5-1}{2-0} = 2$	$\frac{2-2}{2-(-1)} = \boxed{0 = \gamma_2}$	
2	2	5	$\frac{4-5}{3-2} = -1$	$\frac{-1-2}{3-0} = -1$	$\frac{-1-0}{3-(-1)} = \boxed{-\frac{1}{4} = \gamma_3}$
3	3	4			

Als Interpolationspolynom erhalten wir also in diesem Beispiel

$$\begin{aligned} p_3(x) &= \gamma_0 + \gamma_1(x-x_0) + \gamma_2(x-x_0)(x-x_1) + \gamma_3(x-x_0)(x-x_1)(x-x_2) \\ &= -1 + 2\cdot(x+1) + 0\cdot(x+1)(x-0) - \tfrac{1}{4}(x+1)x(x-2) \\ &= -1 + 2(x+1) - \tfrac{1}{4}x(x+1)(x-2) \end{aligned}$$

Beim praktischen Rechnen beachtet man, daß man gemeinsame Klammerterme ausklammern kann, wodurch sich der Aufwand für die Auswertung des Polynoms auf n Multiplikationen und $2n+1$ Additionen reduziert.

$$p_n(x) = \underbrace{(((\ldots(}_{n-1}\gamma_{0,n}\cdot(x-x_{n-1}) + \gamma_{0,n-1})(x-x_{n-2}) + \cdots + \gamma_{0,1})(x-x_0) + y_0$$

Der durch die obige Klammerung angedeutete Algorithmus heißt **"verallgemeinertes Hornerschema"**.

NUMAWWW Interpolation, Polynominterpolation

Das Interpolationspolynom soll uns als Ersatz für die den Daten zugrundeliegende Funktion dienen. Es stellt sich also die Frage, wie groß die Abweichung des Interpolationspolynoms zu den Daten von f von einer gegebenen Funktion $f(x)$ ist, d.h. wir betrachten folgende Problemstellung:

Sei $f(x)$ unbekannt, aber die Daten $y_j = f(x_j)$ für $j = 0, \ldots, n$ gegeben.

Wie groß ist dann $f(x) - p_n(x)$ für $x \neq x_j$?

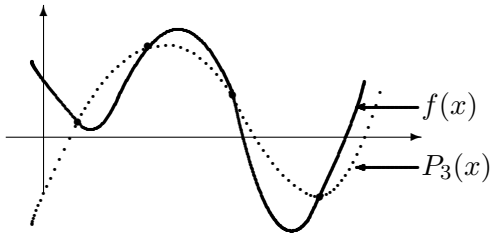


Abb 1.1.3

Wir beobachten:

Die Daten zu den Stützstellen $(x_0, \ldots, x_n)$ liefern ein Interpolationspolynom $p_n(x)$. Für eine beliebigen Stelle $\tilde{x}$ setzen wir $x_{n+1} \stackrel{def}{=} \tilde{x}$ und berechnen das Interpolationspolynom $p_{n+1}(x)$ zu diesen $n+2$ Punkten.

Dann gilt an der Stelle $\tilde{x}$

$$\begin{aligned} f(\tilde{x}) - p_n(\tilde{x}) &= p_{n+1}(\tilde{x}) - p_n(\tilde{x}) \\ &= \gamma_{n+1}(\tilde{x}-x_0)\cdot\ldots\cdot(\tilde{x}-x_n) \\ &= f_{[x_0,\ldots,x_n,\tilde{x}]}(\tilde{x}-x_0)\cdot\ldots\cdot(\tilde{x}-x_n) \end{aligned}$$

Wir müssen nun noch die dividierte Differenz $f_{[x_0,\ldots,x_n,\tilde{x}]}$ durch eine handhabbare Grösse der Funktion f ausdrücken. Dazu gilt

Satz 1.1.2. *Sei $f \in C^{n+1}[a,b]$, d.h. im Intervall $[a,b]$ $n+1$-mal stetig differenzierbar, und $x_i, \ldots, x_{i+k} \in [a,b], k \leq n+1$.*
Dann gilt:

$$f_{[x_i,\ldots,x_{i+k}]} = \frac{f^{(k)}(\xi)}{k!}$$

mit unbekannter Zwischenstelle ξ zwischen den Stützstellen $x_i, \ldots, x_{i+k}$. □

Bemerkung 1.1.4. *Dies bedeutet, daß das Schema der dividierten Differenzen in der k-ten Spalte Werte von $\frac{f^{(k)}}{k!}$ enthält (die Spalten werden von null an numeriert). Die Zwischenstellen sind hierbei unbekannt, aber man kann bei genügend feiner Tabellierung aus den dividierten Differenzen zumindest die Größenordnung der einzelnen Ableitungen ablesen.*

Bemerkung 1.1.5. *Man kann wegen Satz 1.1.2 bei den dividierten Differenzen auch den Fall mehrfacher Argumente zulassen, z.B. $f_{[x_0,x_0,x_0]}$ und dies als Grenzwert für den Fall paarweiser verschiedener, aber (teilweise) gegen gemeinsame Grenzwerte konvergierender Stützstellen interpretieren. Im Grenzfall von $n+1$ zusammenfallenden Argumenten hat man dann das Taylorpolynom der Funktion f.*

Zwei unmittelbare Folgerungen aus der Eindeutigkeit der Lösung der Interpolationsaufgabe sind

Satz 1.1.3. Permutationsinvarianz der dividierten Differenzen *Ist $(j_i, \ldots, j_{i+k})$ eine Permutation von $(i, \ldots, i+k)$, dann gilt*

$$f_{[x_i,\ldots,x_{i+k}]} = f_{[x_{j_i},\ldots,x_{j_{i+k}}]}$$

□

Man darf also die Stützstellen beliebig anordnen, ohne am zugehörigen Interpolationspolynom etwas zu ändern und obwohl einzelne Zwischengrössen sich ändern, ändert sich die "Spitze" eines Teildreiecks im dreieckigen Schema der dividierten Differenzen nicht .

Satz 1.1.4. *Ist $f \in \Pi_k$ und $k < n$, dann $f_{[x_0,\ldots,x_n]} = 0$* □

Bemerkung 1.1.6. *Die Umkehrung von Satz 1.1.4 ist natürlich falsch! Wenn die k^{te} dividierte Differenz von $(x_i, f(x_i)), \ldots, (x_{i+m}, f(x_{i+m}))$ identisch verschwindet, folgt letztlich, daß f auf der Stützstellenmenge mit einem Polynom vom Grad $< k$ übereinstimmt!*
(Beispiel: $f(x) = \sin(\frac{\pi}{2}x)$
mit $x_i = 4i+1, i \in Z$ $f_{[x_i,x_{i+1}]} \equiv 0$,

aber f ist natürlich keine Konstante!)

Beispiel 1.1.4. *Zu 5 Stützpunkten gibt es gwöhnlich ein Polynom vom genauen Grad 4, das diese interpoliert. Mit den Daten $(-1,1),\ (0,1),\ (1,1),\ (3,25),\ (4,61)$ ergibt sich als Schema der dividierten Differenzen*

$$\begin{array}{rrrrrr} -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 4 & 1 & 0 \\ 1 & 1 & 12 & 8 & 0 & 0 \\ 3 & 25 & 36 & 0 & 0 & 0 \\ 4 & 61 & 0 & 0 & 0 & 0 \end{array}$$

es gibt also ein Polynom vom Grad 3, das diese Werte interpoliert.

Wir gelangen nun zu folgender **Fehleraussage**

Satz 1.1.5. *Sei $f \in C^{n+1}[a,b]$ und $x_0,\ldots,x_n \in [a,b]$ seien paarweise verschieden. p_n sei das eindeutig bestimmte Interpolationspolynom vom Höchstgrad n zu $(x_i, f(x_i)),\ i=0,\ldots,n$. Dann gilt für $x \in [a,b]$:*

$$\underbrace{f(x)-p_n(x)}_{\text{Fehler}} = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\cdot\prod_{j=0}^{n}(x-x_j), \quad \textit{mit } \xi_x \in [a,b] \textit{ unbekannt}$$

□

Beispiel 1.1.5. $f(x)=e^x, \quad [a,b]=[0,1], \quad n=5$

$$x_0=0, \quad x_1=0.2, \quad x_2=0.4, \quad x_3=0.6, \quad x_4=0.8, \quad x_5=1$$

$$|f^{(6)}(x)/6!| \le 2.72/720 \ = \ 3.78\cdot 10^{-3}$$

und numerische Maximierung ergibt

$$x\in[0,1] \Rightarrow \left|\prod_{i=0}^{5}(x-x_i)\right| \le 1.1\cdot 10^{-3}$$

Die Fehlerschranke ist also $4.16\cdot 10^{-6}$ während der tatsächliche maximale Fehler $2.65\cdot 10^{-6}$ beträgt.

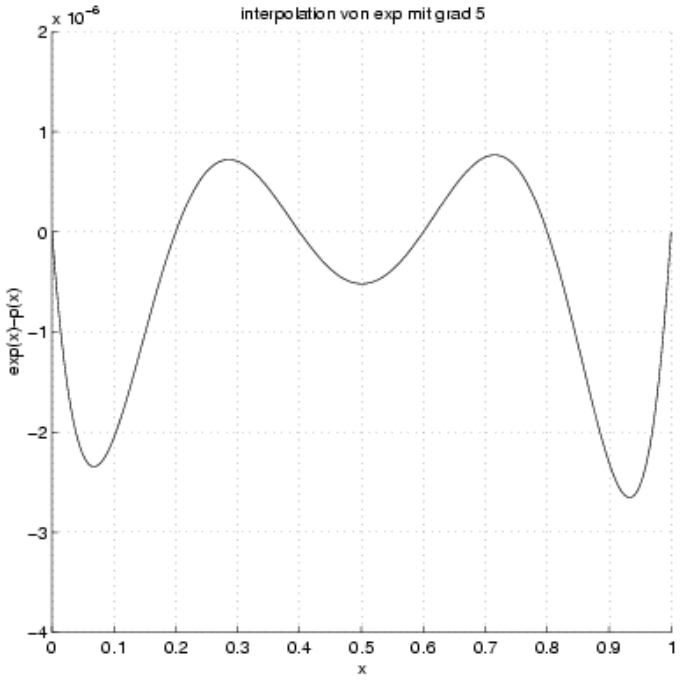


Abb 1.1.4

Das Restglied der Interpolation hat also zwei Faktoren: Das Polynom

$$\prod_{j=0}^{n}(x - x_j)$$

wächst ausserhalb seines Stützstellenintervalls sehr schnell an, ist aber insbesondere in der Mitte des Stützstellenintervalls (bei vernünftiger Anordnung der x_j) recht klein. Der andere Faktor

$$f^{(n+1)}(\xi)/(n+1)!$$

spiegelt die Regularität der zugrundeliegenden Funktion wieder: Dazu gilt folgende Aussage über $\frac{f^{(n)}(x)}{n!},\ x \in [a,b]$:

Lemma 1.1.1. *Sei $f \in C^\infty[a,b]$ für jedes x in $[a,b]$ in eine Potenzreihe entwickelbar mit Konvergenzradius $\geq R$, d.h. R sei der kleinste Abstand einer singulären Stelle von f (mit f aufgefaßt als komplexe Funktion von $z \in \mathbb{C}$) zu irgendeinem Punkt x von $[a,b]$, dann gilt für alle $x \in [a,b]$ und alle n:*

$$\left|\frac{f^{(n)}(x)}{n!}\right| \leq \left(\tfrac{1}{R} + \varepsilon_n(x)\right)^n \quad \textit{mit} \quad \varepsilon_n(x) \to 0.$$

. □

Beispiele:

1. Sei $f(x) = e^x$. Es gibt also keine singuläre Stelle, d.h. $R \in \mathbb{R}$ ist beliebig. Es folgt also für jedes kompakte Intervall $[a, b]$

$$\max_{x\in[a,b]} \left\{ \left| \frac{f^{(n)}(\xi_x)}{n!} \right| \cdot \left| \prod_{j=0}^{n} (x - x_j) \right| \right\} \to 0 \quad \text{für} \quad n \to \infty$$

weil das Produkt kleinergleich $(b - a)^{n+1}$ ist und wir R beliebig groß wählen können.

2. Sei $f(x) = \frac{1}{1+25x^2}$ für $[a, b] = [-1, 1]$

$f(x)$ ist singulär für $z = \pm\frac{1}{5}\mathrm{i}$. Es gilt also $R = \frac{1}{5}$.

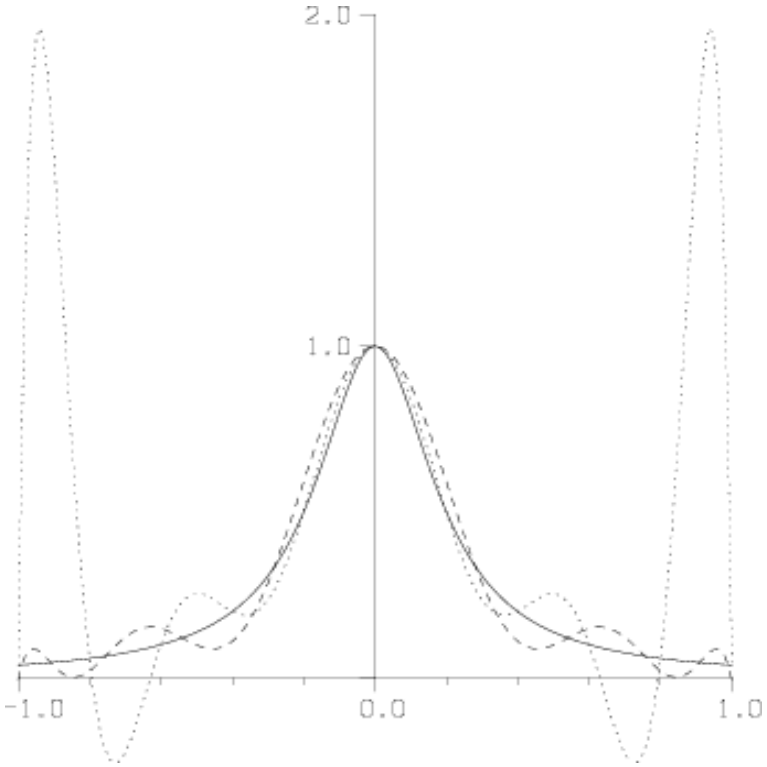
Man kann nun zeigen, daß für jede denkbare Anordnung der Stützstellen x_i stets gilt

$$\max_{x\in[-1,1]} |\prod_{j=0}^{n} (x - x_j)| \geq \frac{1}{2^n}$$

und sogar " $\gg$ " für die äquidistante Einteilung. Dies deutet bereits darauf hin, daß es hier wohl Schwierigkeiten geben wird. Man erhält keine Konvergenzaussage und auch tatsächlich keine Konvergenz im äquidistanten Fall. In der Abbildung unten ist einmal die Interpolierende an äqudistanten Stützstellen (die mit $n \to \infty$ tatsächlich an den Intervallrändern punktweise divergiert) und die Interpolierende an den sogenannten Tschebyscheffabszissen

$$x_i \;=\; \cos(\tfrac{2i+1}{n+1}\tfrac{\pi}{2})$$

dargestellt. Letztere Interpolation ist in diesem Fall mit $n \to \infty$ konvergent, wenn auch recht langsam.



——— $f(x) = \frac{1}{1+25x^2}$ -------- $p_{0,10}(x; (\cos\left(\frac{2i+1}{11} \cdot \frac{\pi}{2}\right), f_i))$

$\cdots\cdots$ $p_{0,10}(x; (-1 + \frac{i}{5}, f_i))$

Abb 1.1.5

Man sollte deshalb in der Praxis n nie sehr groß wählen, sondern eher stückweise in kleinen Intervallen vorgehen. Wählt man eine äquidistante Einteilung der Stützstellen $x_i = x_0 + ih$, dann kann man zeigen daß

$$\max_{x\in[x_0,x_n]} |f(x) - p_n(x)| \;\leq\; \max_{x\in[x_0,x_n]} |f^{(n+1)}(x)|h^{n+1}$$

gilt und auch für die Ableitungen

$$\max_{x\in[x_0,x_n]} |f^{(k)}(x) - p_n^{(k)}(x)| = \mathcal{O}(1)h^{n+1-k} \quad k \;=\; 1,\ldots,n$$

wobei $\mathcal{O}(1)$ vom Maximum der Ableitungen $f^{(n+1)},\ldots,f^{(n+1+k)}$ abhängt. Neben der Funktionsapproximation kann man die Interpolationspolynome also auch zur Approximation von Ableitungen benutzen. Dies werden wir uns später zu Nutze machen.

<<

Wir geben nun einen Überblick über mögliche **Anwendungen der Polynominterpolation:**

1. **Tabellenkonstruktion** Hier besteht die Aufgabe darin, bei vorgegebenem Gültigkeitsbereich $[a, b]$ und vorgegebenem Interpolationsgrad n (in der Regel $n = 1$ oder $n = 3$) eine Gittereinteilung $x_0 = a < x_1 < \ldots < x_N$ so zu konstruieren, daß der Fehler zwischen der gegebenen Funktion f und dem Interpolationspolynom zu den Stellen $x_i,\ x_{i+1}, \ldots, x_{i+n}$ kleiner als eine vorgegebene Schranke ist. Hier hängt i von der gewünschten Auswertestelle x ab, man versucht, x in die Mitte des Stützstellenintervalls zu plazieren.

   Beispiel 1.1.6. Konstruktion einer äquidistanten Tabelle von $\sin x$ mit $x \in [0, \frac{\pi}{2}]$.
   Forderung: Fehler $\leq 5 \cdot 10^{-9}$ bei kubischer Interpolation, dh.
   $\forall x \in [0\ ,\ \frac{\pi}{2}] \quad | \sin x - p_3(x; (x_j, \sin x_j) : j = i, \ldots, i+3) | \leq 5 \cdot 10^{-9}$
   wobei i von x abhängt.
   Forderung an die Schrittweite h : $h = b \cdot 10^{-l}$, $b \in \{1, 2, 5\}$, $l \in \mathbb{N}$ (damit eine vernünftige Tabelle entsteht.)
   Bei äquidistanter Interpolation hat die Funktion

$$\omega(x) \stackrel{def}{=} \prod_{j=i}^{i+3}(x - x_j)$$

   folgendes Aussehen:

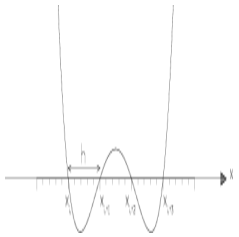


   Abbildung 1.1.6

   Es ist deshalb sinnvoll, die Wahl von x_i (zu gegebenen x) so vorzunehmen, daß $x_{i+1} \leq x \leq x_{i+2}$ (sonst wird $\omega(x)$ unnötig groß). Um den Extremwert von ω einfach bestimmen zu können, drücken wir x und die x_j in der neuen Variablen t aus durch

$$x \stackrel{def}{=} x_i + th, \quad x_j \stackrel{def}{=} x_i + (j - i)h$$

also

$$\omega(x_i + th) \;=\; th(th-h)(th-2h)(th-3h) \;=\; h^4 t(t-1)(t-2)(t-3)$$

Uns interessiert der Extremalwert in $1 \le t \le 2$. Er ergibt sich bei $t = 1.5$, d.h.
$x_{i+1} \le x \le x_{i+2} \;\Rightarrow$
$|\;\omega(x)\;| \le h^4 \;\cdot\; 1.5^2 \cdot 0.5^2 \;=\; h^4 \frac{9}{16}$
Wegen $|\sin^{(4)}(x)| \le 1 \quad (\forall x)$ und $4! = 24$ ergibt sich als Bedingung für h
$\frac{1}{24} \cdot \frac{9}{16} h^4 \;\le\; 5 \cdot 10^{-9}$, *d.h.* $h \;\le\; 2.15 \cdot 10^{-2}$ also $h \;=\; 2 \cdot 10^{-2}$
Damit die Konstruktion von i für alle $x \in [0, \frac{\pi}{2}]$ gelingt, benötigt man als erste Stützstelle $x_{-1} = -2 \cdot 10^{-2}$ und als letzte $x_{80} = 1.60$, die Tabelle erhält also 82 Einträge. Die Tabellengenauigkeit muß natürlich 8 Nachkommastellen betragen. (In der Nähe von $x = 0$ würde der Interpolationsfehler (bei exakten Werten $\sin x_i$) natürlich noch viel kleiner als $5 \cdot 10^{-9}$ wegen $\sin^{(4)}(\xi) \;=\; \sin\xi \;\approx\; 0$. Weil die Tabellenwerte aber gerundet sind, tritt aufgrund der Rundungsfehler auch bei 0 ein Gesamtfehler von $\approx 10^{-9}$ auf. □

2. **Approximation einer Funktion auf einem festem Intervall $[a, b]$ durch ein einziges Polynom :**

   Hierbei erweist sich eine äquidistante Gittereinteilung als unzweckmäßig.

   Abhilfe liefert eine Interpolation an den sogenannten **Tschebyscheffabszissen** (die auch in obiger Abbildung benutzt wurden)

$$x_j^{(n)} = \frac{a+b}{2} + \frac{b-a}{2} \cdot \cos\Big(\frac{(2j+1)\pi}{(n+1)\cdot 2}\Big), \quad j = 0, \ldots, n\,.$$

   Für die damit gebildeten Polynome gilt folgender Satz:

   Satz 1.1.6. *Sei $f \in C^k[a,b]$, $k \ge 2$ fest. Dann gilt für das mit den Tschebyscheff-Abszissen gebildete $p_n(x)$: Es gilt für $n \;\to \infty$*

$$\max_{x\in[a,b]} |p_n(x) - f(x)| \le \mathit{const} \;\cdot \tfrac{\ln n}{n^k} \cdot \max_{x\in[a,b]} |f^{(k)}(x)| \left(\frac{\pi(b-a)}{4}\right)^k \to 0$$

   Hier liegt also für $n \;\to\; \infty$ immer Konvergenz des Interpolationspolynoms gegen die zugrundeliegende Funktion im Sinne der Maximumnorm vor, wenn f wenigstens zweimal stetig differenzierbar ist.

3. **Nullstellenbestimmung und inverse Interpolation:**

   Sei $f:\; [a,b] \to [c,d]$ bijektiv, d.h. $f'(x) \neq 0$ für alle $x \in [a,b]$.

   Weiterhin seien die Daten $(x_i, f(x_i))$ für f gegeben. Dann gilt $y_i = f(x_i) \iff x_i = f^{-1}(y_i)$. Es folgt also, daß die Daten (y_i, x_i) eine Tabelle für die Umkehrfunktion f^{-1} bilden. Es gilt $f(x^*) = 0 \iff x^* = f^{-1}(0)$. D.h. wir nähern $f^{-1}(0)$

durch ein Interpolationspolynom mit den Abszissen $y_0, \ldots, y_n$ und den Ordinaten $x_0, \ldots, x_n$ an.

Eine Auswertung dieses Polynoms an der Stelle $0 = y$ ergibt dann die neue Nullstellennäherung. In diesem Sinne sind $x_0, \ldots, x_n$ "alte" Nullstellennäherungen und $x_{n+1} \stackrel{def}{=} p_n(0)$ die neue Nullstellennäherung.

Beispiel 1.1.7. *Die Funktion* $f(x) = x^3 - 2x - 5$ *hat eine Nullstelle bei* $x^* = 2.0945514815$. *Mit einer Tabelle aus den Werten bei* 1.6, 1.8, 2.0, 2.2 *und inverser Interpolation ergibt sich das Schema dividierter Differenzen*

−4.10400000000000	1.60000000000000	0.14970059880240	−0.01178428700280	0.00107806331605
−2.76800000000000	1.80000000000000	0.11312217194570	−0.00601449213532	0
−1.00000000000000	2.00000000000000	0.08896797153025	0	0
1.24800000000000	2.20000000000000	0	0	0

und als Nullstellennäherung durch Auswertung des Polynoms bei $y = 0$ *der recht gute Näherungswert 2.0927*

NUMAWWW Nichtlineare Gleichungen, Einschachtelungsverfahren

Das Brent-Decker-Verfahren ist im wesentlichen inverse quadratische Interpolation, wobei in gewissen Ausnahmefällen auf lineare Interpolation bzw. Intervallhalbierung zurückgegriffen wird.

4. **Numerische Quadratur** (Kapitel 2)

   Zur näherungsweisen Berechnung des Integrals einer Funktion bestimmen wir zunächst ein Interpolationspolynom, das wir anschliessend einfach integrieren können, d.h. wir bilden $\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b p_n(x)\,\mathrm{d}x$.

5. **Numerische Differentiation** (Kapitel 8)

   Mit $f(x) \approx p_n(x)$ bilden wir $f'(x) \approx p_n'(x)$.

>>

1.2 Stückweise Interpolation in einer Veränderlichen

Aus den Resultaten in Abschnitt 1.1 folgt, daß es zu Approximationszwecken nicht sinnvoll ist, den Interpolationsgrad stark zu vergrößern, um den Approximationsfehler klein zu machen, wenn die zu approximierende Funktion selbst nur geringe Regularitätseigenschaften besitzt oder das Intervall sehr gross ist. Wenn z.B. das Strassenprofil in unserem

Eingangsbeispiel die Länge 1000 (m) hat und je 10m ein Messwert dafür vorliegt, wird man kaum diese 101 Werte durch ein Polynom vom Grad 100 annähern. Stattdessen gehen wir hier so vor, daß bei festgehaltenem Interpolationsgrad eine Einteilung des Ausgangsintervalls in kleine Teilintervalle betrachtet wird. Die einfachste, auf dem Ausgangsintervall noch stetige stückweise polynomiale Approximationsfunktion ist dann der interpolierende Streckenzug: (Abb. 1.2.1)

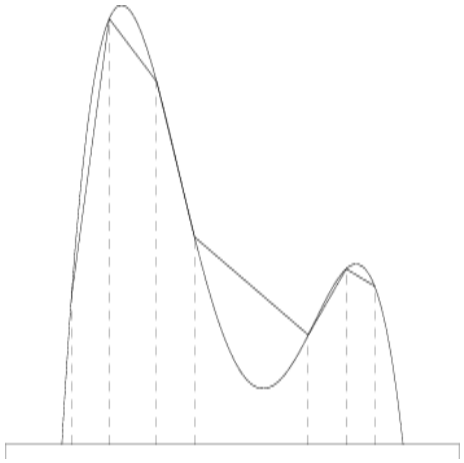


Abbildung 1.2.1

Aufgabenstellung:
Gegeben $x_i,\ y_i = f(x_i) \quad i = 0, \ldots, n+1\ , \quad a = x_0 < \cdots < x_{n+1} \stackrel{def}{=} b$
Gesucht: $s \in C[a,b]\ :\quad s_{\vert[x_i,x_{i+1}]} \in \Pi_1$ für $i = 0, \ldots, n$ und
$s(x_i) = y_i, \quad i = 0, \ldots, n+1$

Diese Aufgabe kann man durch Angabe **geeigneter Basisfunktionen** im Raum

$$\mathcal{S}_1(\mathcal{Z}) \stackrel{def}{=} \{s \in C[a,b] :\ s_{\vert[x_i,x_{i+1}]} \in \Pi_1 \quad \text{für } i = 0, \ldots, n\}$$

unmittelbar lösen: Mit

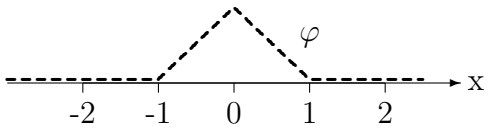
$$\varphi_i(x) \stackrel{def}{=} \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{für } x \in [x_{i-1}, x_i] \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i} & \text{für } x \in [x_i, x_{i+1}] \\ 0 & sonst \end{cases} \qquad \text{“Dachfunktionen”}$$

wird

$$s(x) = \sum_{i=0}^{n+1} y_i \varphi_i(x)\ .$$

Zur Definition von φ_0 und φ_{n+1} benötigt man noch Hilfspunkte x_{-1} und x_{n+2}, die man beliebig $< a$ *bzw.* $> b$ wählen kann. Besonders einfach werden alle Aussagen, wenn man die Einteilung $\mathcal{Z}$ **äquidistant** wählt. Dann kann man mit

$$\varphi(x) = \begin{cases} 0 & x < -1 \\ x+1 & -1 \le x \le 0 \\ 1-x & 0 < x \le 1 \\ 0 & 1 < x \end{cases}$$



schreiben

$$s(x) = \sum_{i=0}^{n+1} y_i \varphi(\frac{x - x_i}{h}) \ .$$

Es gelten dazu die folgenden Konvergenzaussagen:

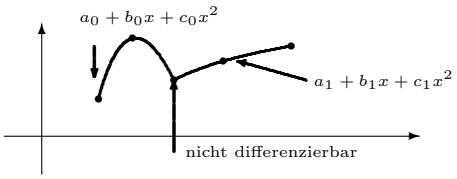
Satz 1.2.1. Konvergenzsatz für stetige stückweise lineare Interpolation
Es sei
$a = x_0 < \cdots < x_{n+1} = b \ , \quad x_i = a + ih$ *mit* $h = (b-a)/(n+1)$ *und* $f \in C^2[a,b]$.
Dann gilt mit $M_2 \stackrel{def}{=} \max\limits_{x \in [a,b]} \mid f''(x) \mid$

(1) $\max\limits_{x \in [a,b]} \mid f(x) - s(x) \mid \le \dfrac{h^2}{8} M_2$

(2) $\max\limits_{x \in [a,b]} \mid f'(x) - s'(x) \mid \le \dfrac{h}{2} M_2$

(Dabei sei definiert:
$s'(x_0) = \lim\limits_{\varepsilon \searrow 0} s'(x_0 + \varepsilon) \ , \ s'(x_{n+1}) = \lim\limits_{\varepsilon \nearrow 0} s'(x_{n+1} + \varepsilon) \ ,$
$s'(x_i) \stackrel{def}{=} \lim\limits_{\varepsilon \searrow 0} (s'(x_i - \varepsilon) + s'(x_i + \varepsilon))/2$ *.)* □

Diese Interpolierende approximiert also sogar auch f' immer noch mit einem Fehler $\mathcal{O}(h)$, obwohl sie selbst an den Stützstellen in der Regel gar nicht differenzierbar ist. Man könnte analog mit höheren Interpolationsgraden verfahren, indem man intervallweise vorgeht, aber dies führt zu Näherungen, bei denen die erste Ableitung in der Regel Sprünge an den Stützstellen aufweist, was unerwünscht ist.



1.3 "Glatte" Interpolation, Spline-Interpolation

Im Folgenden sind die Stützstellen x_i stets als angeordnet anzusehen

$$x_i \ < \ x_{i+1} \quad \forall\, i \ .$$

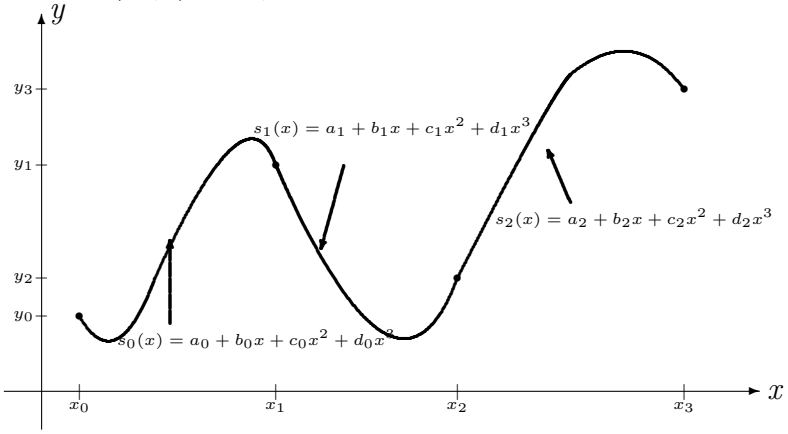
Unser **Ziel** ist es, daß mit der Interpolation durch stückweise polynomiale Funktionen eine differenzierbare Approximation zu erhalten. In einer Veränderlichen ist die sinnvollste Konstruktion ein **kubischer Spline** $S(x)$. Dieser erfüllt

Definition 1.3.1. *Ein interpolierender kubischer Spline S zu den Daten $(x_i, y_i),\ i = 0, \ldots, n+1$ ist gegeben durch die Forderungen*

1. *S ist eine in $[x_0, x_{n+1}]$ zweimal stetig differenzierbare Funktion.*
2. *Die Einschränkung von S auf das Intervall $[x_i, x_{i+1}]$ ist ein Polynom vom Höchstgrad 3, das wir mit S_i bezeichnen:*

$$S_{|[x_i, x_{i+1}]} \ = \ S_i(x) \ \in \ \Pi_3.$$

3. $S_i(x_i) = y_i$,
4. $S_i(x_{i+1}) = y_{i+1}$.



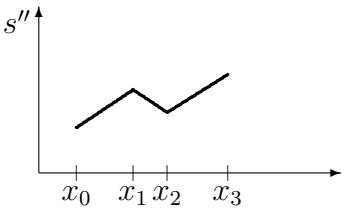
Die Bedingung der zweimaligen stetigen Differenzierbarkeit kann man mit den Teilstücken S_i formulieren als

$$\left.\begin{array}{l} S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \\ S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \end{array}\right\} i = 0, \ldots, n-1 \ .$$

Diese Splines bilden bei festem Gitter $x_0, \ldots, x_{n+1}$ einen Vektorraum. Wir werden noch sehen, daß er die Dimension $n+4$ hat mit den $y_0, \ldots, y_{n+1}$ als Freiheitsgraden und zwei weiteren freien Bedingungen, deren Wahl wir noch diskutieren.

Ein guter Ansatz für $S(x)$ geht aus von $S''(x)$, weil S'' stückweise linear und stetig ist, also über seine Funktionswerte sofort angebbar ist.

Wir setzen $M_i \stackrel{def}{=} S''(x_i)$. (für die elastische Linie sind dies die Biegemomente).



$$\begin{aligned} S_i''(x) &= \tfrac{x-x_i}{x_{i+1}-x_i} \cdot M_{i+1} + \tfrac{x_{i+1}-x}{x_{i+1}-x_i} \cdot M_i \\ \Rightarrow \quad S_i(x) &= \tfrac{1}{6}\Big(\tfrac{(x-x_i)^3}{x_{i+1}-x_i} \cdot M_{i+1} + \tfrac{(x_{i+1}-x)^3}{x_{i+1}-x_i} \cdot M_i \Big) \\ &\quad + c_i(x-x_i) + d_i \end{aligned}$$

c_i und d_i entstehen als freie Integrationskonstante, wobei um der geschickteren Berechnung willen nicht die Form $c_i x + d_i$, sondern die äquivalente $c_i(x-x_i)+d_i$ gewählt wurde. Wir berechnen c_i, d_i aus den Interpolationsforderungen fúr das Stück S_i: $S_i(x_i) = y_i$ und $S_i(x_{i+1}) = y_{i+1}$.

Wir erhalten für $i = 0, \ldots, n$

$$d_i = y_i - h_{i+1}^2 M_i^*$$

und

$$c_i = \frac{y_{i+1} - y_i}{h_{i+1}} - h_{i+1}(M_{i+1}^* - M_i^*)$$

mit $M_i^* = M_i/6$. Es verbleibt die Bestimmung der Momente M_i bzw. M_i^*. Dazu nutzen wir die geforderte Stetigkeit von S' aus, d.h.

$$S_i'(x_{i+1}) \;=\; S_{i+1}'(x_{i+1}) \quad \text{für } i = 0, \ldots, n-1$$

Dies ergibt, die obige Darstellung und die bereits ermittelten c_i und d_i eingesetzt, ein lineares Gleichungssystem zur Bestimmung der M_i (bzw. M_i^*).

Das Resultat ist mit

$$h_{i+1} = x_{i+1} - x_i, \;\; i = 0, \ldots, n$$

ein System mit den Gleichungen

$$\begin{aligned} M_i^* \cdot h_{i+1} + 2(h_{i+1} + h_{i+2}) \cdot M_{i+1}^* + h_{i+2} \cdot M_{i+2}^* & \\ = \quad \frac{y_{i+2} - y_{i+1}}{h_{i+2}} - \frac{y_{i+1} - y_i}{h_{i+1}} & \\ i = 0, 1, 2, \ldots, n-1 & \end{aligned}$$

Gesucht sind die Werte für die sogenannten *Momente* $M_0^*, \ldots, M_{n+1}^*$.

Man hat also n lineare Gleichungen für $n+2$ Unbekannte. Dieses System ist immer lösbar und hat 2 Freiheitsgrade, weil die Matrix den vollen Rang n hat.

Sinnvolle **Zusatzforderungen** , die die Konstruktion eindeutig machen, sind alternativ:

I. $s''(a) = 0,\ s''(b) = 0$ "natürlicher interpolierender kubischer Spline" (Ein dünnes elastisches Lineal (Stahllatte, "Spline" der Schiffsbauer), das in den Punkten $(x_i, y_i)\ \ i = 0, \ldots, n+1$ gelenkig gelagert wird, nimmt bei kleinen $f_{[x_i,x_{i+1}]}$ gerade die Form eines solchen Spline an.)

II. $s'(a) = f'(a),\ \ s'(b) = f'(b)$ "hermitischer interpolierender kubischer Spline"

III. $s'(a) = s'(b),\ \ s''(a) = s''(b)$ mit der **Zusatzvoraussetzung** $f'(a) = f'(b),\ \ $ "periodischer interpolierender kubischer Spline" (Diese Konstruktion ist nur sinnvoll, wenn auch $f(a) = f(b)$. Dann ist wegen der Interpolationsforderung auch $s(a) = s(b)$, d.h. s wird eine periodische Funktion mit Periode $b - a$)

Diese Zusatzbedingungen haben ihre Begründung in der Gleichung

$$\int_a^b (s''(x) - f''(x))^2 \mathrm{dx} \ = \ \int_{\mathrm{a}}^{\mathrm{b}} (\mathrm{f}''(\mathrm{x}))^2 \mathrm{dx} - \int_{\mathrm{a}}^{\mathrm{b}} (\mathrm{s}''(\mathrm{x}))^2 \mathrm{dx} - 2(\mathrm{f}'(\mathrm{x}) - \mathrm{s}'(\mathrm{x}))\mathrm{s}''(\mathrm{x})|_{\mathrm{a}}^{\mathrm{b}}$$

die für jede zweimal stetig differenzierbare Funktion f gilt, die die $y-$Werte interpoliert. Durch sie wird der dritte Term auf der rechten Seite der Gleichung zu null, was bedeutet

$$\int_a^b (s''(x))^2 \mathrm{dx} \ \leq \ \int_{\mathrm{a}}^{\mathrm{b}} (\mathrm{f}''(\mathrm{x}))^2 \mathrm{dx} \ .$$

d.h. der Spline ist die "glatteste" Funktion, die die Daten interpoliert.

Nach Einarbeitung der jeweils 2 Zusatzbedingungen ergibt sich ein lineares Gleichungssystem für die Momente M_i. Dies hat folgende Gestalt:

$$A \cdot \vec{M}^* = B$$

Im Fall I (natürlicher Spline)

$$A = \begin{pmatrix} 2(h_1+h_2) & h_2 & 0 & \cdots & \cdots & \cdots & 0 \\ h_2 & \ddots & \ddots & & & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & & \cdots & \vdots \\ \vdots & \cdots & h_{l-1} & 2(h_{l-1}+h_l) & h_l & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & h_n \\ 0 & \cdots & \cdots & \cdots & 0 & h_n & 2(h_n+h_{n+1}) \end{pmatrix}$$

$$\vec{M}^* = \begin{pmatrix} M_1^\star \\ M_2^\star \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ M_n^\star \end{pmatrix} \qquad B = \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \frac{y_{n+1} - y_n}{h_{n+1}} - \frac{y_n - y_{n-1}}{h_n} \end{pmatrix}$$

$M_0^\star = M_{n+1}^\star = 0$

Im Fall II

$$A = \begin{pmatrix} 2h_1 & h_1 & 0 & \cdots & \cdots & \cdots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \cdots & \cdots & \vdots \\ 0 & h_2 & 2(h_2+h_3) & h_3 & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & h_n & \vdots \\ \vdots & & & & h_n & 2(h_n+h_{n+1}) & h_{n+1} \\ 0 & \cdots & \cdots & \cdots & 0 & h_{n+1} & 2h_{n+1} \end{pmatrix}$$

$$\vec{M}^* = \begin{pmatrix} M_0^\star \\ M_1^\star \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ M_{n+1}^\star \end{pmatrix} \qquad B = \begin{pmatrix} \frac{y_1 - y_0}{h_1} - f'(x_0) \\ \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \vdots \\ \vdots \\ \frac{y_{n+1} - y_n}{h_{n+1}} - \frac{y_n - y_{n-1}}{h_n} \\ f'(x_{n+1}) - \frac{y_{n+1} - y_n}{h_{n+1}} \end{pmatrix}$$

Im Fall III

$$A = \begin{pmatrix} 2(h_1+h_2) & h_2 & 0 & \cdots & \cdots & \cdots & h_1 \\ h_2 & \ddots & \ddots & & & & 0 \\ 0 & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \cdots & h_l & 2(h_l+h_{l+1}) & h_{l+1} & \cdots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & h_{n+1} \\ h_1 & 0 & \cdots & \cdots & \cdots & h_{n+1} & 2(h_1+h_{n+1}) \end{pmatrix}$$

$$\vec{M}^* = \begin{pmatrix} M_1^\star \\ M_2^\star \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ M_{n+1}^\star \end{pmatrix} \qquad B = \begin{pmatrix} \frac{y_2-y_1}{h_2} - \frac{y_1-y_0}{h_1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \frac{y_{n+1}-y_n}{h_{n+1}} - \frac{y_n-y_{n-1}}{h_n} \\ \frac{y_1-y_0}{h_1} - \frac{y_{n+1}-y_n}{h_{n+1}} \end{pmatrix}$$

$M_0^\star = M_{n+1}^\star$

Die Koeffizientenmatrix ist in allen drei Fällen symmetrisch mit nichtnegativen Elementen. Für jede Zeile ist das Diagonalelement grösser als die Summe aller Ausserdiagonalelemente. Eine solche Matrix nennt man strikt diagonaldominant:

$$|a_{i,i}| \; > \; \sum_{j=1, j\neq i}^{n} |a_{i,j}| \quad \forall \; i \; .$$

Wir beweisen

Satz 1.3.1. Invertierbarkeit strikt diagonaldominanter Matrizen *Jede strikt diagonaldominante Matrix ist invertierbar.* □

Beweis: Wir nehmen an, die Behauptung sei falsch. Dann gibt es ein $x^* \neq 0$ mit $Ax^* = 0$. Dieses x^* hat eine betragsmaximale Komponente $x_{i_0}^* \neq 0$. Wir betrachten

nun die Zeile i_0 von Ax^*:

$$\begin{aligned} 0 &= |\sum_{j=1}^{n} a_{i_0,j} x_j^*| \\ &= |x_{i_0}^*||a_{i_0,i_0} + \sum_{j=1, j\neq i_0}^{n} a_{i_0,j} x_j^*/x_{i_0}^*| \\ &\geq |x_{i_0}^*|(|a_{i_0,i_0}| - \sum_{j=1, j\neq i_0}^{n} |a_{i_0,j}|) \\ &> 0 \ . \end{aligned}$$

Dies ist ein Widerspruch, die Annahme also falsch und der Satz bewiesen.
Wir haben somit

Satz 1.3.2. Existenz- und Eindeutigkeitssatz der kubischen Splineinterpolation *Zu beliebigen $a = x_0 < x_1 < \cdots < x_{n+1} = b$ und $y_i,\ i = 0, \ldots, n+1$ existiert genau ein interpolierender kubischer Spline, der eine der Bedingungen I, II, III erfüllt.* □

Bemerkung 1.3.1. *Gelegentlich fordert man statt der bisher besprochenen Zusatzbedingungen auch die "kein Knoten" (not a knot) Bedingung, was bedeutet, daß die dritte Ableitung in x_1 und x_n stetig sein soll. Da die dritte Ableitung stückweise konstant sein muss, bedeutet dies, daß auf $[x_0, x_2]$ bzw. $[x_{n-1}, x_{n+1}]$ mit jeweils einem Polynom dritten Grades interpoliert wird, also für $n = 2$ die Konstruktion auf die gewöhnliche kubische Interpolation reduziert wird.* □

<<

Beispiel 1.3.1. *Gesucht ist der natürliche kubische interpolierende Spline zu den Daten*

x_i	-3	-1	0	1	3
y_i	5	3	7	9	23

Für den natürlichen Spline gilt: $M_0^* = M_4^* = 0$.
Mit $h_1 = h_4 = 2$ und $h_2 = h_3 = 1$ erhalten wir folgendes lineare Gleichungssystem:

$$Ax = b \quad mit$$

$$A = \begin{pmatrix} 2\ (2+1) & 1 & 0 \\ 1 & 2\ (1+1) & 1 \\ 0 & 1 & 2\ (1+2) \end{pmatrix} = \begin{pmatrix} 6 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 6 \end{pmatrix},$$

$$x = \begin{pmatrix} M_1^* \\ M_2^* \\ M_3^* \end{pmatrix}, \quad b = \begin{pmatrix} \frac{7-3}{1} - \frac{3-5}{2} \\ \frac{9-7}{1} - \frac{7-3}{1} \\ \frac{23-9}{2} - \frac{9-7}{1} \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \\ 5 \end{pmatrix}$$

Als Lösung erhält man:
$M_1^* = 1,\ M_2^* = -1,\ M_3^* = 1.$

2. Koeffizienten der Splinefunktionen:

$$\begin{aligned} d_0 &= 5 - 4 \cdot 0 = 5, & c_0 &= \tfrac{3-5}{2} - 2 \cdot (1-0) = -3, \\ d_1 &= 3 - 1 \cdot 1 = 2, & c_1 &= \tfrac{7-3}{1} - 1 \cdot (-1-1) = 6, \\ d_2 &= 7 - 1 \cdot (-1) = 8, & c_2 &= \tfrac{9-7}{1} - 1 \cdot (1+1) = 0, \\ d_3 &= 9 - 4 \cdot 1 = 5, & c_3 &= \tfrac{23-9}{2} - 2 \cdot (0-1) = 9 \end{aligned}$$

3. Bestimmung der Splinefunktionen: Mit den berechneten Werten ergibt sich:

$$\begin{aligned} s_0(x) &= \tfrac{1}{2}[(-1-x)^3 \cdot 0 + (x+3)^3 \cdot 1] - 3(x+3) + 5 \\ s_1(x) &= (0-x)^3 \cdot 1 + (x+1)^3 \cdot (-1) + 6(x+1) + 2 \\ s_2(x) &= (1-x)^3 \cdot (-1) + (x-0)^3 \cdot 1 + 0 \cdot (x-0) + 8 \\ s_3(x) &= \tfrac{1}{2}[(3-x)^3 \cdot 1 + (x-1)^3 \cdot 0] + 9(x-1) + 5 \end{aligned}$$

Für den Spline gilt also:

$$s(x) = \begin{cases} \frac{1}{2}(x+3)^3 - 3(x+3) + 5, & -3 \le x \le -1 \\ -x^3 - (x+1)^3 + 6(x+1) + 2, & -1 < x \le 0 \\ -(1-x)^3 + x^3 + 8, & 0 < x \le 1 \\ \frac{1}{2}(3-x)^3 + 9(x-1) + 5, & 1 < x \le 3 \end{cases}$$

Die folgende Abbildung vergleicht die Polynominterpolation mit der Interpolation durch einen natürlichen kubischen Spline für 19 äquidistante Datenpunkte in $[0,1]$ mit den Ordinatenwerten

$$y = -(0, 0.35, 0.8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0.8, 0.3, 0, 0)$$

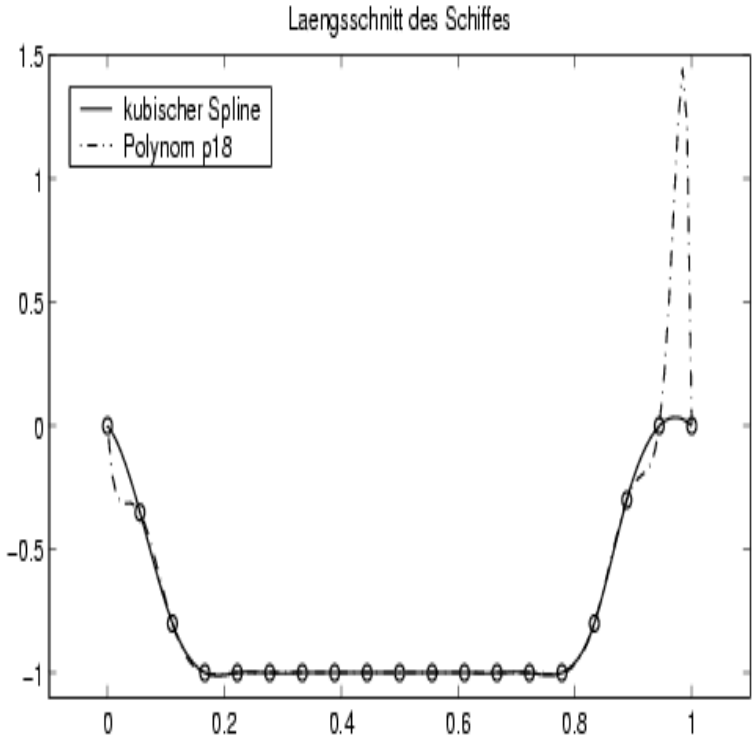


Abb 1.3.1

Während der Spline die Daten im gesamten Bereich sehr gut approximiert, zeigt das interpolierende Polynom grosse Ausschläge an den Intervallenden und diese Lösung wäre hier ganz unbrauchbar.

Eine wichtige Anwendung von periodischen Splines besteht in der Konstruktion geschlossener differenzierbarer Kurven durch vorgegebene Punkte (x_i, y_i),
$i = 0, \ldots, n$. Man legt eine Reihenfolge dieser Punkte auf einem Streckenzug fest, setzt $x_{n+1} \stackrel{def}{=} x_0$, $y_{n+1} \stackrel{def}{=} y_0$, bestimmt die Bogenlänge auf dem Streckenzug:

$$t_0 \stackrel{def}{=} 0, \quad t_i \stackrel{def}{=} t_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \quad i = 1, \ldots, n+1$$

und legt dann periodische kubische Splines s_1 durch (t_i, x_i) und s_2 durch (t_i, y_i), $i = 0, \ldots, n+1$. Die Kurve ist dann in Parameterform $(s_1(t), s_2(t))$,
$0 \le t \le t_{n+1}$ dargestellt. Durch Veränderung der willkürlichen Parameterfestlegung t_i kann man das Aussehen der Kurve beeinflussen. Das folgende Bild zeigt das Resultat einer solchen Konstruktion.

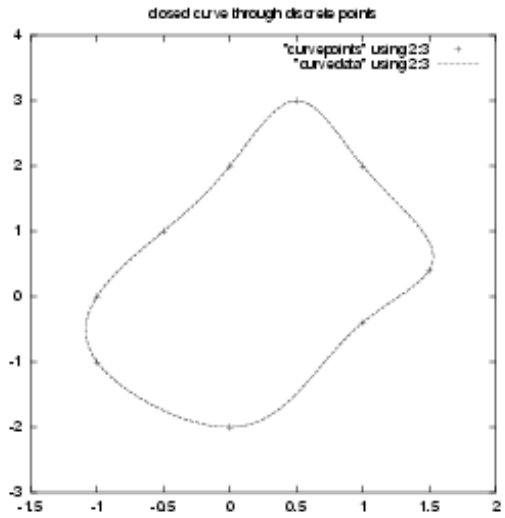


Abb 1.3.2

Beispiel 1.3.2. *Gegeben sind die vier Punkte* $P_0 = (0,0)$, $P_1 = (1,0)$, $P_2 = (1,1)$, $P_3 = (0,1)$. *Es wird* $P_4 := P_0$ *gesetzt. Zur geschickteren Darstellung lassen wir hier t nicht von 0 bis 4, sondern von -2 bis 2 variieren. Die beiden periodischen kubischen Splines* $s_1(t)$ *und* $s_2(t)$ *genügen dabei folgenden Bedingungen:*

t_i	-2	-1	0	1	2
$s_1(t_i)$	0	1	1	0	0
$s_2(t_i)$	0	0	1	1	0

i) *Berechnung von* $s_1(t)$ *(mit* $h_i = 1, i = 0, \ldots, 4$*):*

$$\begin{pmatrix} 4 & 1 & 0 & 1 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} M_1^\star \\ M_2^\star \\ M_3^\star \\ M_4^\star \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}, \quad M_0^\star = M_4^\star$$

$$\Longrightarrow M_0^\star = \frac{1}{4}, M_1^\star = -\frac{1}{4}, M_2^\star = -\frac{1}{4}, M_3^\star = \frac{1}{4}, M_4^\star = \frac{1}{4}$$

Wegen $d_i = y_i - h_{i+1}^2 M_i^\star$ *folgt weiterhin*

$$\Longrightarrow d_0 = -\frac{1}{4}, d_1 = \frac{5}{4}, d_2 = \frac{5}{4}, d_3 = -\frac{1}{4}$$

Und mit

$$c_i = \frac{y_{i+1} - y_i}{h_{i+1}} - h_{i+1}(M_{i+1}^\star - M_i^\star)$$

erhält man

$$\begin{aligned}
&\Longrightarrow \quad c_0 = \frac{3}{2}, c_1 = 0, c_2 = -\frac{3}{2}, c_3 = 0 \\
&\Longrightarrow \quad s_1(t) = \begin{cases} \frac{1}{4}(-1-t)^3 - \frac{1}{4}(t+2)^3 + \frac{3}{2}(t+2) - \frac{1}{4} & , t \in [-2,-1] \\ -\frac{1}{4}(-t)^3 - \frac{1}{4}(t+1)^3 + \frac{5}{4} & , t \in [-1,0] \\ -\frac{1}{4}(1-t)^3 + \frac{1}{4}t^3 - \frac{3}{2}t + \frac{5}{4} & , t \in [0,1] \\ \frac{1}{4}(2-t)^3 + \frac{1}{4}(t-1)^3 - \frac{1}{4} & , t \in [1,2] \end{cases}
\end{aligned}$$

ii) *Berechnung von $s_2(t)$:*
Als neue rechte Seite des Gleichungssystems erhält man $(1,-1,-1,1)^T$. Daraus ergibt sich:

$$\Longrightarrow M_0^\star = \frac{1}{4}, M_1^\star = \frac{1}{4}, M_2^\star = -\frac{1}{4}, M_3^\star = -\frac{1}{4}, M_4^\star = \frac{1}{4}$$

Für die Koeffizienten c_i, d_i erhält man:

$$\begin{aligned}
&d_0 = -\frac{1}{4}, d_1 = -\frac{1}{4}, d_2 = \frac{5}{4}, d_3 = \frac{5}{4} \\
&c_0 = 0, c_1 = \frac{3}{2}, c_2 = 0, c_3 = -\frac{3}{2} \\
&\Longrightarrow \quad s_2(t) = \begin{cases} \frac{1}{4}(-1-t)^3 + \frac{1}{4}(t+2)^3 - \frac{1}{4} & , t \in [-2,-1] \\ \frac{1}{4}(-t)^3 - \frac{1}{4}(t+1)^3 + \frac{3}{2}(t+1) - \frac{1}{4} & , t \in [-1,0] \\ -\frac{1}{4}(1-t)^3 - \frac{1}{4}t^3 + \frac{5}{4} & , t \in [0,1] \\ -\frac{1}{4}(2-t)^3 + \frac{1}{4}(t-1)^3 - \frac{3}{2}(t-1) + \frac{5}{4} & , t \in [1,2] \end{cases}
\end{aligned}$$

□

Zum Abschluß dieses Abschnitts wollen wir einen Satz über die Approximationsgüte der hermitischen kubischen Splines kennenlernen. Dieser besagt, daß der interpolierende kubische hermitische Spline mit seinen Ableitungen die interpolierte Funktion mit ihren ersten drei Ableitungen approximiert mit einer Approximationsgüte $h^4, \ldots, h$.

>>

Satz 1.3.3. Konvergenzsatz für den kubischen hermitischen C^2-Spline
Es sei $f \in C^4[a,b]$, $a = x_0 < \ldots < x_{n+1}$ *und* $h = \max\{x_{i+1} - x_i\}$.
s *sei der hermitische kubische interpolierende Spline zu* $(x_i, f(x_i))$,
$i = 0, \ldots, n+1$
Dann gilt für $j = 0, 1, 2, 3$

$$\max_{x\in[a,b]} |s^{(j)} - f^{(j)}(x)| \leq 2h^{4-j}C_4 \ \ mit \ \ C_4 \stackrel{def}{=} \max_{x\in[a,b]} |f^{(4)}(x)|$$

Dabei sei

$$s'''(x_0) \stackrel{def}{=} \lim_{\epsilon\searrow 0} s'''(x_0+\varepsilon), \quad s'''(x_{n+1}) \stackrel{def}{=} \lim_{\epsilon\searrow 0} s'''(x_{n+1}-\varepsilon)$$

und für $i = 1, \ldots, n$

$$s'''(x_i) \stackrel{def}{=} \lim_{\epsilon\searrow 0} (s'''(x_i+\varepsilon) + s'''(x_i-\varepsilon))/2$$

□

Bemerkung 1.3.2.

1. *Ein analoge Aussage (mit einer anderen Konstanten als 2) gilt für den natürlichen Spline auf jedem abgeschlossenenen Intervall* $[c,d]$ *mit* $a < c < d < b$. *An den Intervallrändern aber ist die Approximation nur* $\mathcal{O}(h^2)$ *genau, was aus der willkürlichen Setzung von* $s''(a)$ *und* $s''(b)$ *resultiert. Ist* f *periodisch mit Periode* $b-a$, *dann gilt die gleiche Aussage auch für den periodischen Spline.*

2. *Man kann diese Konstruktionen auch mit anderen Polynomgraden durchführen. Dann muss man aber teilweise die Interpolationsbedingungen anders wählen oder andere Randbedingungen formulieren. Siehe dazu die Spezialliteratur.* □

1.4 Stückweise polynomiale Interpolation in zwei Veränderlichen

Bei **zwei** freien Veränderlichen x und y betrachten wir einen Bereich $\bar{G} \subset \mathbb{R}^2$ und eine Funktion $f: \bar{G} \to \mathbb{R}$.

Unser Ziel ist es, f durch ein Polynom in zwei Veränderlichen zu approximieren .

Wir betrachten hier zwei Fälle:

1. Sei $\bar{G} = [a,b] \times [c,d]$ ein achsenparalleles Rechteck. Weiterhin sei eine Zerlegung vorgegeben, $a = x_0 < \ldots x_n = b$ und $c = y_0 < \ldots < y_m = d$. Auf dem Gitter der (x_i, y_j) seien die Funktionswerte $f_{i,j} = f(x_i, y_j)$ bekannt.

Als Interpolationspolynom ergibt sich unmittelbar mit Hilfe der Lagrange-Polynome:

$$P_{n,m}(x,y) \stackrel{def}{=} \sum_{i=0}^{n} \sum_{j=0}^{m} f_{i,j} \cdot \underbrace{L_{i,n}(x) \cdot L_{j,m}(y)}_{\text{Lagrange-Polynome}} .$$

Für diese Interpolation kann man eine zum eindimensionalen Fall analoge Fehlerabschätzung beweisen.

Außer für $n, m \leq 2$ ist dieses Vorgehen ungebräuchlich, die entstehenden Funktionen sind sehr "wellig".

Für $n = m = 1$ ist dies der bilineare Ansatz $f_{00} \cdot \frac{x-x_1}{x_0-x_1} \cdot \frac{y-y_1}{y_0-y_1} + \cdots + f_{11} \cdot \frac{x-x_0}{x_1-x_0} \cdot \frac{y-y_0}{y_1-y_0} = \cdots = a + bx + cy + dxy$.
Auf jeder achsenparallelen Geraden ist dies eine affin lineare Funktion, global aber eine hyperbolische Fläche. Will man auf einem grösseren Gebiet arbeiten, wo ein solch niedriger Grad keine ausreichende Genauigkeit liefert, dann kann man dieses Gebiet in kleinere Rechtecke zerlegen und wieder stückweise interpolieren. Die so erzeugte Interpolierende ist dann automatisch stetig, aber in der Regel nicht stetig differenzierbar. (Die Tatsache, dass man achsenparallele Rechtecke vorliegen hat, ist für die Stetigkeit wesentlich (warum?)).

Beispiel 1.4.1. *Die Datenpunkte* $(0,0;1),\ (2,0;2),\ (0,2;4),\ (2,2;8)$ *werden durch die bilineare Funktion*

$$1\frac{(x-2)(y-2)}{4} + 2\frac{x(y-2)}{(-4)} + 4\frac{(x-2)y}{(-4)} + 8\frac{xy}{4}$$

interpoliert.

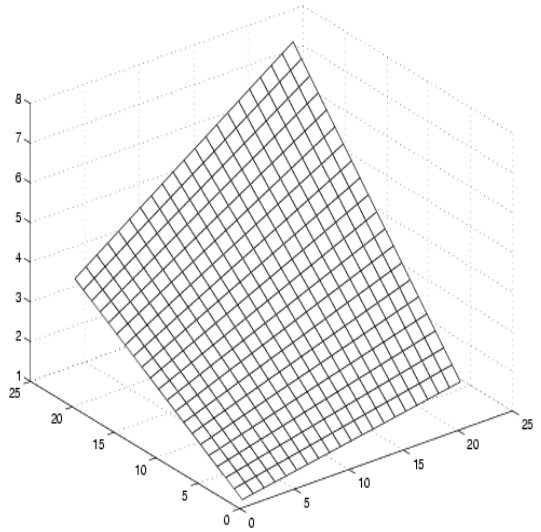


Abb 1.4.1

Die folgende Abbildung zeigt den Interpolationsfehler für die Funktion $\sin(\pi x)\cos(\pi y)$ auf $[-1,1]\times[-1,1]$ bei äquidistanter Interpolation mit $n=4$ und $m=6$

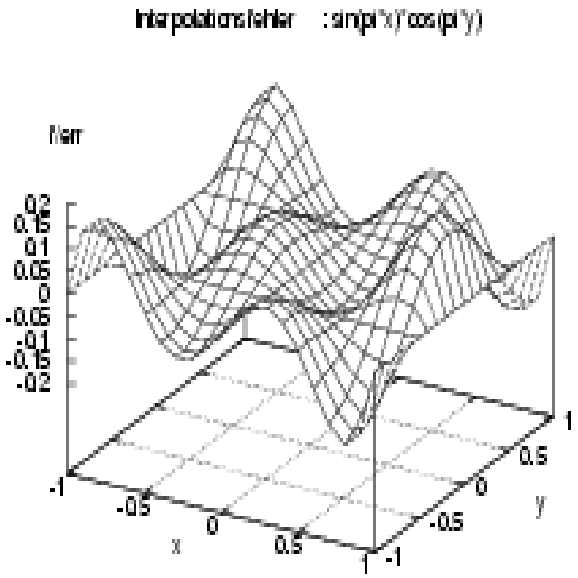


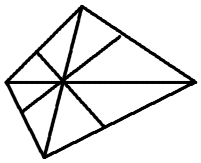
Abb 1.4.2

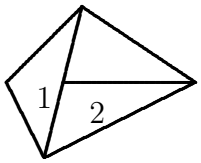
2. Fall Sei nun $\bar{G}$ polygonal berandet. Wir definieren

Definition 1.4.1. *Eine Menge* $\{T_i,\ i = 0, \ldots, N\}$*, wobei* T_i *ein abgeschlossenes Dreieck im* $\mathbb{R}^2$ *ist, heißt* **zulässige Triangulierung** *von* $\bar{G}$*, wenn gilt:*

1. $\bar{G} = \bigcup_{i=0}^{N} T_i$.
2. $T_i \cap T_j = \begin{cases} \emptyset & \\ P_{ij} & \text{(gemeinsamer Eckpunkt von } T_i \text{ und } T_j\text{)} \\ K_{ij} & \text{(gemeinsame vollständige Seite von } T_i \text{ und } T_j\text{)} \end{cases}$

Zur Illustration seien folgende **Beispiele** angegeben:

$\bar{G}$:  zulässig

$\bar{G}$: 

unzulässig, weil 1 und 2 keine vollständige gemeinsame Seite haben.

Die Menge der Ecken der Dreiecke der Triangulierung sei nun mit $\{P_0, \ldots, P_s\}$ bezeichnet. Sie heisst die "Knotenmenge" der Triangulierung. Wir wollen nun auf jedem Dreieck f durch eine affin lineare Funktion approximieren, im Ganzen aber eine stetige Approximation erhalten. Man überlegt sich leicht, daß dies auf einer unzulässigen Triangulierung nicht möglich ist. Die Interpolierende wollen wir wieder in einer Basisdarstellung darstellen, wie im eindimensionalen Fall.

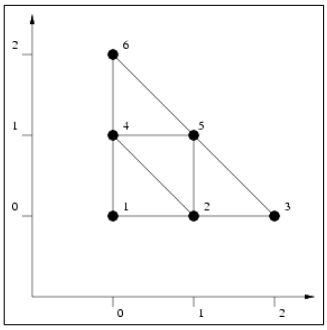
Definition 1.4.2. *Die* **Basisfunktion** *der stetigen stückweise linearen Interpolation zum Knoten* $P_j = (x_j, y_j)$ *der Triangulierung ist definiert durch*

1. $\varphi_j(x, y) = a_{ij} + b_{ij}(x - x_j) + c_{ij}(y - y_j)$*, falls* $(x, y) \in T_i$
2. $\varphi_j \in C(\bar{G})$
3. $\varphi_j(P_j) = 1$ *und* $\varphi_j(P_l) = 0$ *für* $l \neq j$

Wir erhalten die stückweise lineare Interpolierende zu den Daten $(P_j, f(P_j))$ dann in der Form

$$\sum_{j=0}^{s} f(P_j)\varphi_j(x, y)\,.$$

Beispiel 1.4.2. *Wir suchen die Basisfunktion zum Knoten 5 der unten dargestellten Triangulierung.*



Die gesuchte Basisfunktion muß in den Knoten 1,2,3,4,6 verschwinden und im Knoten 5 den Wert 1 annehmen. Daraus ergeben sich mit dem linearen Ansatz

$$\varphi(x,y) = a + bx + cy$$

(wobei a, b und c vom jeweiligen Dreieck abhängen) folgende Bestimmungsgleichungen

- *Dreieck 124 :*

$$\left.\begin{array}{rcrcl} \varphi(P_1) & = & a + b\cdot 0 + c\cdot 0 & = & 0 \\ \varphi(P_2) & = & a + b\cdot 1 + c\cdot 0 & = & 0 \\ \varphi(P_4) & = & a + b\cdot 0 + c\cdot 1 & = & 0 \end{array}\right\} \Rightarrow a = b = c = 0$$

- *Dreieck 235 :*

$$\left.\begin{array}{rcrcl} \varphi(P_2) & = & a + b\cdot 1 + c\cdot 0 & = & 0 \\ \varphi(P_3) & = & a + b\cdot 2 + c\cdot 0 & = & 0 \\ \varphi(P_5) & = & a + b\cdot 1 + c\cdot 1 & = & 1 \end{array}\right\} \Rightarrow a = b = 0, c = 1$$

- *Dreieck 245 :*

$$\left.\begin{array}{rcrcl} \varphi(P_2) & = & a + b\cdot 1 + c\cdot 0 & = & 0 \\ \varphi(P_4) & = & a + b\cdot 0 + c\cdot 1 & = & 0 \\ \varphi(P_5) & = & a + b\cdot 1 + c\cdot 1 & = & 1 \end{array}\right\} \Rightarrow a = -1, b = c = 1$$

- *Dreieck 456 :*

$$\left.\begin{array}{rcrcl} \varphi(P_4) & = & a + b\cdot 0 + c\cdot 1 & = & 0 \\ \varphi(P_5) & = & a + b\cdot 1 + c\cdot 1 & = & 1 \\ \varphi(P_6) & = & a + b\cdot 0 + c\cdot 2 & = & 0 \end{array}\right\} \Rightarrow b = 1, a = c = 0$$

Die Basisfunktion lautet demnach

$$\varphi_5(x,y) = \begin{cases} 0 & in \quad Dreieck\ 124 \\ y & in \quad Dreieck\ 235 \\ x+y-1 & in \quad Dreieck\ 245 \\ x & in \quad Dreieck\ 456 \end{cases}$$

Satz 1.4.1. Existenz und Eindeutigkeit der stetigen stückweise linearen 2D Interpolation *Es sei $\bar{G}$ ein polygonal berandeter Bereich und $\{T_0, \ldots, T_N\}$ eine zulässige Triangulierung mit der Knotenmenge $\{P_0, \ldots, P_s\}$. Dann gibt es eine eindeutig bestimmte, stetige, auf jedem T_i affin-lineare Funktion l mit $l(P_j) = f_j,\ j = 0, \ldots, s$. Hierbei sind f_j beliebig vorgegebene Werte.*
Diese besitzt die **Darstellung**

$$l(x,y) = \sum_{j=0}^{s} f_j \cdot \varphi_j(x,y)\,.$$

$\varphi_j(x,y)$ ist hier die Basisfunktion zum Knoten P_j.

Die Stetigkeit folgt aus der Konstruktion. Die folgende Abbildung zeigt eine solche Konstruktion

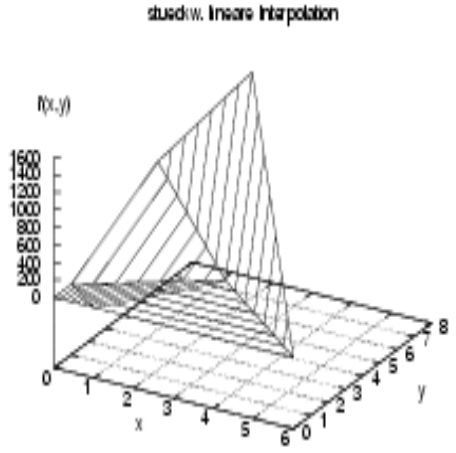


Abb 1.4.3

Satz 1.4.2. *Approximationsgüte der stetigen stückweise linearen 2D-Interpolation: Sei $f \in C^2(\bar{G})$ und $\{T_0, \ldots, T_n\}$ eine zulässige Triangulierung von $\bar{G}$ mit der Knotenmenge $P_0, \ldots, P_s$ und $f_j = f(P_j),\ j = 0, \ldots, s$.*
h sei die Länge der längsten Dreiecksseite und φ der kleinste Dreieckswinkel.
Dann gilt für die oben konstruierte Funktion l die Aussage

$$\max_{(x,y)\in\bar{G}} |f(x,y) - l(x,y)| \ \leq \ c \cdot h^2$$

und

$$\sup_i \left(\sup_{(x,y)\in T_i^0} \left\{ \left| \frac{\partial}{\partial x} f(x,y) - \frac{\partial}{\partial x} l(x,y) \right|, \left| \frac{\partial}{\partial y} f(x,y) - \frac{\partial}{\partial y} l(x,y) \right| \right\} \right) \ \leq \ \frac{ch}{\sin\varphi} \ .$$

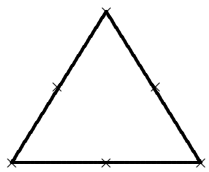
Hierbei ist T_i^0 das Innere von T_i. Und es gilt $c = 8M_2$ mit

$$M_2 = \max_{(x,y)\in\bar{G}} \left\{ \left| \tfrac{\partial^2}{\partial^2 x} f(x,y) \right|, \left| \tfrac{\partial^2}{\partial x \partial y} f(x,y) \right|, \left| \tfrac{\partial^2}{\partial^2 y} f(x,y) \right|, \right\}$$

□

Man muß die Bedingung $\max_i \left\{ \frac{\text{längste Seite}}{\text{kürzeste Seite}}(T_i) \right\} \leq$ const für $h \to 0$ erfüllen, um $\varphi \to 0$ zu verhindern. Dann kann man also aus der stetigen, stückweise linearen Approximation, die ja auf den Knotenverbindungen nicht differenzierbar ist, dennoch brauchbare Approximationen an den Gradienten von f erhalten.

Bemerkung 1.4.1. *Analog kann man für vollständigen Grad 2 vorgehen. Man führt außer den Ecken der Dreiecke noch die Seitenmitten als Interpolationspunkte ein.*



$$a + bx + cy + dx^2 + exy + fy^2$$

Auch diese Interpolationsaufgabe ist eindeutig lösbar.

Für Ansätze höheren Grades benutzt man die Form:

$$\sum_{i=0}^{n} \sum_{j=0}^{i} a_{ij} x^{i-j} y^j$$

Für $n = 3$ nimmt man als Knoten die Drittelungen der drei Dreiecksseiten und zusätzlich den Dreiecksschwerpunkt (10 Freiheitsgrade). Diese Interpolierenden sind über die Dreiecksseiten hinweg stetig, aber nicht notwendig differenzierbar. Um differenzierbare

Interpolanten zu erhalten, muss man ähnlich vorgehen wie bei den Splines im eindimensionalen Fall.

Splinefunktionen lassen sich leicht auf Rechtecksgittern oder kubischen Gittern berechnen. Auf Dreiecksnetzen wird die Berechnung glatter Interpolierenden wesentlich komplizierter, ist aber auch möglich.

In zwei Veränderlichen und mit dem Ziel, eine C^1-Funktion auf einer zulässigen Triangulierung zu erhalten, stellt man fest, daß 21 Freiheitsgrade pro Dreieck notwendig sind. Diese repräsentieren die Daten $f,\ f_x,\ f_y,\ f_{xx},\ f_{xy},\ f_{yy}$ an den drei Ecken eines Dreiecks und die 3 Normalableitungen an den Seitenmitten. Der Ansatz hierzu ist von der Form

$$\sum_{i=0}^{5}\sum_{j=0}^{i} a_{ij}x^j y^{i-j} \ .$$

Es gibt aber einfachere Ansätze für reduzierte Glattheitsanforderungen. Siehe dazu die Spezialliteratur.

1.5 Zusammenfassung

Die Aufgabe der Interpolation von $n+1$ Datenpunkten mit paarweise verschiedenen Abszissen ist stets eindeutig lösbar. Deshalb kann man ein Polynom wahlweise durch seine Koeffizientendarstellung in einer geeigneten (und frei wählbaren) Basis des Π_n oder durch $n+1$ seiner Wertepaare darstellen. Für theoretische Zwecke ist die Darstellung von Lagrange die zweckmässigste. An ihr erkennt man u.a. unmittelbar, daß das Polynom linear von den Ordinatenwerten y_i, aber nichtlinear von den Abszissen x_i abhängt. Für rechnerische Zwecke ist die Darstellung nach Newton die zweckmässigste und auch die effizienteste. Wird nur ein einziger Wert des Interpolationspolynoms gesucht, so ist der Nevillealgorithmus günstig, der diesen Wert direkt rekursiv berechnet ohne das Polynom selbst aufzustellen.

Interpolationspolynome von kleinem oder massvollem Grad (evtl. ≤ 10) liefern sehr gute Annäherungen auf kleinen Intervallen und können zur Funktionsapproximation, zur numerischen Differentiation und Quadratur und auch zur Nullstellenbestimmung benutzt werden. Es ist in der Regel jedoch nicht zweckmässig und oft sogar unsinnig, durch die Erhöhung des Grades eine Genauigkeitssteigerung erzwingen zu wollen. Oft ist dieser Prozess sogar divergent.

Will man ein Interpolationspolynom als Approximation einer Funktion auf einem grösseren Intervall benutzen, so sollte man auf jeden Fall die transformierten Tschebyscheffabszissen benutzen.

Den Interpolationsvorgang kann man leicht ausdehnen auf den Fall, daß auch zusätzlich

Ableitungswerte vorgeschrieben werden, dabei dürfen aber keine "Lücken" in den Ordnungen der Ableitungen auftreten. (Hermite-Interpolation und Verallgemeinerungen).

Auf grossen Intervallen ist in der Regel eine Splineapproximation angebracht, bei der stückweise polynomiale Funktionen so zusammengesetzt werden, daß eine mehrmals differenzierbare Funktion entsteht. Der klassische kubische Spline ist zweimal stetig differenzierbar. Erst durch Vorgabe zweier zusätzlicher Bedingungen wird seine Konstruktion eindeutig. Wählt man dazu Randvorgaben in Form des natürlichen, des hermitischen oder des periodischen Splines, so erhält man sogar eine im Sinne der Minimierung des Quadratintegrals der zweiten Ableitung optimale Konstruktion. Der kubische Spline liefert eine Approximationsgüte vierter Ordnung im maximalen Abszissenabstand h und sogar simultan eine um je eine $h-$Potenz niedrigere Approximation an die erste, zweite und dritte Ableitung (letztere durch eine Treppenfunktion). Auch dieser Spline hängt linear von den Ordinatenwerten, aber nichtlinear von den Abszissen ab. Im Unterschied zur Polynominterpolation die z.B. lokal (in einer Tabelle) durchführbar ist, ist die Splinekonstruktion ein globaler Prozess, man muss eine gekoppeltes System linearer Gleichungen lösen, um die Konstruktion durchzuführen. Die Methoden der eindimensionalen Interpolation lassen sich auf das Mehrdimensionale nur dann unmittelbar übertragen, wenn der Bereich ein Rechteck (oder Quader) ist. Dann genügen nämlich sogenannte Tensorproduktansätze, bei denen die Basis aus den Produkten der Basiselemente der eindimensionalen Interpolation in den verschiedenen unabhängigen Variablen entstehen. Sonst muss man zu anderen Formen der Interpolation übergehen. In der Praxis bewährt ist die Interpolation auf simplizialen Netzen (im Zweidimensionalen also Dreiecksnetze)

Kapitel 2

Numerische Integration (Quadratur)

2.1 Problemstellung und Grundbegriffe

In diesem Kapitel besprechen wir Methoden zur genäherten Berechnung von Werten bestimmter (Riemannscher) Integrale $\int_a^b f(t)\mathrm{d}t$. Eine spezielle Verfahrensklasse lässt auch die direkte Behandlung unendlicher Intervalle zu. Einfache Lösungsansätze für diese Aufgaben bestehen darin, zunächst eine polynomiale oder stückweise polynomiale Approximationsfunktion für f zu bestimmen und dann das Integral des Polynoms bzw. des Splines exakt auszuwerten. Über diese einfachen Ansätze hinaus werden wir hier erheblich effizientere Methoden kennenlernen und auch auf die Möglichkeit der Fehlererfassung eingehen. Der zentrale Gesichtspunkt bei allen diesen Verfahren ist der Wunsch, Formeln zu entwickeln, die einerseits für jede Riemannintegrierbare Funktion bei entsprechendem Aufwand ein Resultat liefern, dessen Genauigkeit nach Belieben gesteigert werden kann, andererseits bei "gutartigen" Integranden hohe Genauigkeit mit nur sehr wenigen Funktionsauswertungen garantieren. Darüberhinaus ist man auch an einer automatisierten Genauigkeitskontrolle interessiert. Als Integralnäherungen betrachten wir Formeln des Typs

$$\int_a^b f(t)\mathrm{d}t \;\approx\; \sum_{i=0}^{N} w_i^{(N)} f(t_i^{(N)}) \;.$$

Dabei heissen $w_i^{(N)}$ die "Gewichte" und $t_i^{(N)}$ die "Knoten" der Formel. Welche dramatischen Unterschiede im Aufwand hier auftreten können, soll folgendes einfache Beispiel belegen.

Beispiel 2.1.1. *Zu berechnen sei*

$$\int_1^2 \tfrac{1}{x}\mathrm{d}x \;=\; \ln 2\ .$$

Wir benutzen die Approximation des Integrals durch eine Riemannsumme, wobei wir den Funktionswert jeweils am linken Intervallende nehmen:

$$\int_1^2 \tfrac{1}{x}\mathrm{d}x \;=\; h\sum_{i=0}^{n-1} \tfrac{1}{1+ih} + R_n \quad \textit{mit } h \;=\; \tfrac{1}{n}\ .$$

Aus der Taylorreihe für $\ln(1+x)$ *erhält man mit dem Leibnizkriterium die obere Schranke und mit einer Abschätzung durch eine geometrische Reihe die untere Schranke in*

$$\tfrac{x^2}{2}(1-\tfrac{x}{3}) \;\le\; x-\ln(1+x) \;\le\; \tfrac{x^2}{2} \textit{ für } x\in[0,1]\ .$$

Dies ergibt wegen

$$\begin{aligned}
-R_n &= \sum_{i=0}^{n-1}\Big\{\tfrac{h}{1+ih} - \int_{1+ih}^{1+(i+1)h} \tfrac{1}{x}\mathrm{d}x\Big\}\\
&= \sum_{i=0}^{n-1}\Big\{\tfrac{h}{1+ih} - (\ln(1+(i+1)h)-\ln(1+ih))\Big\}\\
&= \sum_{i=0}^{n-1}\Big\{\tfrac{h}{1+ih} - \ln(\tfrac{1+(i+1)h}{1+ih})\Big\}\\
&= \sum_{i=0}^{n-1}\Big\{x_i - \ln(1+x_i)\Big\} \quad \textit{mit } x_i \;=\; \tfrac{h}{1+ih}
\end{aligned}$$

wegen $h/2 \;\le\; x_i \;\le\; h$ *und* $nh \;=\; 1$ *die Abschätzung*

$$\tfrac{h}{8}(1-\tfrac{h}{3}) \;\le\; |R_n| \;\le\; \tfrac{h}{2}\ .$$

Man sieht, daß der Fehler also nur wie $1/n$ *gegen null geht und um einen Fehler kleinergleich* 10^{-6} *zu garantieren, benötigt man 500000 Funktionswerte. Wir werden später eine Formel erhalten (Gaußformel), die bereits mit 6 Funktionsauswertungen eine Genauigkeit von* $9\cdot 10^{-8}$ *garantiert.*

Alle Ansätze, die wir hier besprechen, beruhen auf der exakten Quadratur geeigneter Interpolationspolynome für den Integranden. (sogenannte interpolatorische Quadratur) Da ein Polynom durch ein Interpolationspolynom vom gleichen oder höheren Grad exakt reproduziert wird, ergibt sich, daß diese Formeln Polynome des entsprechenden Grades exakt integrieren. Dies erlaubt in Kombination mit Abschätzungen des Approximationsfehlers für den Integranden f durch Polynome eine universelle Fehlerabschätzung:

Satz 2.1.1. universelle Fehlerschranke für Quadraturformeln *Sei die Quadraturformel exakt für alle Polynome vom Grad kleinergleich m, und es gelte $t_k^{(N)} \in [a,b] \quad (\forall k)$.*
Dann gilt für $f \in C[a,b]$

$$|\int_a^b f(t)\mathrm{d}t \; - \; \sum_{i=0}^{N} w_i^{(N)} f(t_i^{(N)})| \; \leq \; (b-a+\sum_{j=0}^{N} |w_j^{(N)}|)\, E_m(f)\,.$$

Hier ist $E_m(f)$ der Fehler der Bestapproximation an f durch Polynome vom Höchstgrad m, d.h.

$$E_m(f) \stackrel{def}{=} \min \{ \max\{|f(x)-p_m(x)| : \; x \; \in \; [a,b]\} : p_m \; \in \; \Pi_m\}$$

und man kennt für E_m eine Schranke:

Satz 2.1.2. Satz von Jackson *Falls $f \in C^k[a,b]$ und $\max\limits_{x\in[a,b]} \mid f^{(k)}(x) \mid \leq M_k \quad k$ fest,*
dann gilt für $m \geq k \geq 1$

$$E_m(f) \; \leq \; M_k \left(\frac{\pi}{2}\right)^k \frac{1}{(m+1)m\ldots\ldots(m+2-k)} \left(\frac{b-a}{2}\right)^k =: C(m,k)/m^k$$

(Beweis in Lehrbüchern der Approximationstheorie). Es ist $C(m,k) = \mathcal{O}(1)$ für $m \to \infty$. □

Das bedeutet für die Praxis, daß solche Formeln gute Werte mit geringem Aufwand liefern wenn

1. $b-a$ nicht groß ist

2. die Formel bei gegebener Knotenzahl $N+1$ Polynome möglichst hohen Grades m exakt integriert . (Man kann zeigen, daß $m \; \leq \; 2N+1$ gilt)

3. der Integrand f Ableitungen besitzt, die grössenordnungsmässig nicht stark anwachsen.

In diesem Zusammenhang wird folgende Definition benutzt:

Definition 2.1.1. *Die Quadraturformel*

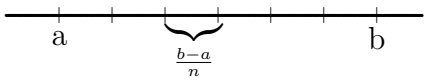
$$Q_{N+1}(f; w^{(N)}, t^{(N)}) := \sum_{j=0}^{N} w_j^{(N)} f(t_j^{(N)})$$

heißt von der **Ordnung** *mindestens k, falls sie alle Polynome vom Grad $\leq k-1$ exakt integriert und von der genauen Ordnung k, wenn es ein Polynom vom Grad k gibt, das nicht von ihr exakt integriert wird.*

Der **Exaktheitsgrad** ist also **Ordnung -1** !

2.2 Newton-Cotes-Quadratur

Zunächst wählen wir zur Berechnung des Interpolationspolynoms $n+1$ äquidistante Stützstellen $x_i^{(n)} := a + i \cdot \frac{b-a}{n}, \quad i = 0, \ldots, n.$



Dann ersetzen wir f durch das Interpolationspolynom auf diesem Gitter, und zwar in der Darstellung von Lagrange

$$f(x) \approx \sum_{i=0}^{n} f(x_i^{(n)}) \cdot L_{i,n}(x), \text{ mit } L_{i,n} := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j^{(n)}}{x_i^{(n)} - x_j^{(n)}}.$$

Dieses Interpolationspolynom wird nun integriert und wir erhalten eine Näherung für das gesuchte Integral

$$\int_a^b f(t)\,\mathrm{d}t \approx \int_a^b \sum_{i=0}^{n} f(x_i^{(n)}) \cdot L_{i,n}(x)\,\mathrm{d}x = \sum_{i=0}^{n} f(x_i^{(n)}) \cdot \underbrace{\int_a^b L_{i,n}(x)\mathrm{d}x}_{=:w_i^{(n)}} .$$

In dieser Formel nennt man die Stützstellen $x_i^{(n)}$ auch Knoten und die $w_i^{(n)} := \int_a^b L_{i,n}(x)\,\mathrm{d}x$ auch Gewichte. Die Gewichte sind vom Integranden unabhängig und können vorab bestimmt werden. Durch Anwendung der Substitutionsregel zur linearen Transformation

von $[a,b]$ auf $[-1,1]$ kann man diese Werte auch unabhängig vom Intervall berechnen und erhält dann eine Darstellung:

$$w_i^{(n)} \;=\; \tfrac{b-a}{2}\tilde{w}_i^{(n)}$$

wo $\tilde{w}_i^{(n)}$ die Gewichte auf $[-1,1]$ sind. Diese haben die Form rationaler Zahlen $Z_{j,n}/D_n$ mit gemeinsamen Hauptnenner D_n und sind für einige Grade unten tabelliert. Man erhält z.B. für

$$n=1 \quad w_0^{(1)} = w_1^{(1)} = \frac{b-a}{2} \textbf{ Trapezregel}$$

und

$$n=2 \quad w_0^{(2)} = w_2^{(2)} = \frac{b-a}{6} \text{ und } w_1^{(2)} = 4\frac{b-a}{6} \textbf{ Simpsonregel } .$$

Diese Art der Konstruktion von numerischen Integrationsformeln liefert bei " kleiner " Intervallbreite $b-a$ und nicht zu großer Knotenanzahl $n+1$ recht gute Näherungen. Für $n \to \infty$ erhält man aber in der Regel ebensowenig eine Konvergenz der Integralnäherung gegen den Integralwert wie die Konvergenz des Interpolationspolynoms gegen die Funktion f. Der Integrationsfehler ergibt sich nämlich aus dem integrierten Interpolationsfehler, der wiederum vom Verhalten der $(n+1)$-ten Ableitung von f im Intervall $[a,b]$ abhängt.

Für die Trapezregel ($n=1$) ergibt sich beispielsweise

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\{f(a)+f(b)\} - \frac{(b-a)^3}{12}f''(\xi) \text{ mit } \xi \in [a,b] \; .$$

Bemerkungen:

1. Die Formeln sind exakt, falls $f \in \Pi_n$. Jedes Polynom vom Grad kleiner gleich n wird also exakt integriert. Sie haben also die Ordnung mindestens $n+1$.

2. Bei der Knotenwahl von $t_i^{(n)} = a + ih, \;\; h$ =Knotenabstand= $\frac{b-a}{n}$
   liegt eine Symmetrie der Knoten zu $\frac{a+b}{2}$ vor. Daraus resultiert eine Symmetrie in den Gewichten, d.h. $w_i^{(n)} = w_{n-i}^{(n)}$ (z.B. 1,4,1 oder 1,3,3,1)
   Daraus folgt ebenfalls, daß die Integrationsformel sogar von der Ordnung $n+2$ ist, falls n gerade ist. Deshalb hat z.B. die Simpsonregel die Ordnung 4, obwohl sie aus der Integration einer Parabel 2. Ordnung hervorgeht.

   Man kann zeigen, daß diese so ermittelte Mindestordnung auch die genaue Ordnung der Formeln ist und daß der Quadraturfehler folgende Gestalt hat. (Details siehe z.B. bei Schmeisser & Schirmeier: Praktische Mathematik)

$$\int_a^b f(t)\mathrm{d}t - \sum_{i=0}^{n} w_i^{(n)} f(t_i^{(n)}) \;=\; \left(\frac{b-a}{2}\right)^{k+1} C_n f^{(k)}(\xi) \quad \text{mit } k = \begin{cases} n+1, n \text{ ungerade} \\ n+2, n \text{ gerade} \end{cases}$$

Die Konstanten C_n sind ebenfalls der untenstehenden Tabelle zu entnehmen.

3. Es gibt auch Quadraturformeln, bei denen zur Interpolation der Funktion f die Randstellen a und b nicht benutzt werden. Das führt zu den *offenen* Newton-Cotes-Formeln, im Gegensatz zu den hier behandelten abgeschlossenen Newton-Cotes-Formeln. Ein Beispiel ist die sogenannte Rechteckregel

$$\int_a^b f(x)\mathrm{d}x \;=\; (b-a)f((a+b)/2) + \frac{1}{24}(b-a)^3 f''(\xi)\ .$$

Tabelle der abgeschlossenen Newton–Cotes–Formeln:
Es ist stets $Z_{j,N} = Z_{N-j,N}$, $w_j^{(N)} = \frac{b-a}{2}\,\frac{Z_{j,N}}{D_N}$, $t_j^{(N)} = a + j(b-a)/N$.

		←		$Z_{j,N}$		→		
N	D_N	$j=0$	1	2	3	4	C_N	*Abl.*
1	1	1	1				$-\frac{2}{3}$	f''
2	3	1	4	1			$-\frac{1}{90}$	f^{IV}
3	4	1	3	3	1		$-\frac{2}{405}$	f^{IV}
4	45	7	32	12	32	7	$-\frac{1}{15120}$	f^{VI}
5	144	19	75	50	50	75	$-\frac{22}{590625}$	f^{VI}
6	420	41	216	27	272	27	$-\frac{1}{3061800}$	f^{VIII}
7	8640	751	3577	1323	2989	2989	$-\frac{334}{1667674575}$	f^{VIII}
8	14175	989	5888	−928	10496	−4540	$-\frac{37}{30656102400}$	f^{X}
9	44800	2857	15741	1080	19344	5778	$-\frac{88576}{114697772870895}$	f^{X}

Beispiel 2.2.1. *Es soll das Integral*

$$\int_0^2 \frac{2}{x^2+4}\,\mathrm{d}x.$$

berechnet werden, und zwar mit Trapezregel und mit der Simpsonregel. Trapezregel:

$$T = \frac{2-0}{2}\big(\frac{2}{4}+\frac{2}{8}\big) = \frac{3}{4} = 0,75$$

Simpsonregel:

$$S = \frac{2-0}{6}\big(\frac{2}{4}+4\frac{2}{5}+\frac{2}{8}\big) = \frac{47}{60} \approx 0,78\bar{3}.$$

Der exakte Wert ist

$$\int_0^2 \frac{2}{x^2+4}\,\mathrm{d}x = \Big[\arctan\frac{x}{2}\Big]_0^2 = \frac{\pi}{4} \approx 0,7854\ldots,$$

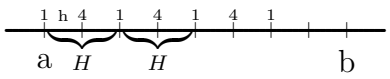
so dass die Simpsonregel hier genauer ist.

2.3 Zusammengesetzte Newton-Cotes-Formeln

Wie im vorigen Abschnitt diskutiert, liefern die Newton-Cotes-Formeln nur für kleine Intervalle und nicht zu große Knotenanzahl gute Näherungen. Dies macht man sich zunutze, indem man ein "grosses" Ausgangsintervall in einzelne kleinere Intervalle aufteilt

und in jedem dieser Teilintervalle das entsprechende Teilintegral mit einer Newton-Cotes-Formel wie oben berechnet.

Beispielsweise ergibt sich bei einer Aufteilung des Intervalls $[a, b]$ in N Teilintervalle der Breite $H = \frac{b-a}{N}$ unter Verwendung der Simpsonregel ($n = 2$) folgende Aufteilung



Für die Fälle Grad n, $n = 1$ und $n = 2$ erhält man so die Resultate mit $h =$ Knotenabstand

$n = 1$: **Zusammengesetzte Trapezregel**

$$\left\| \begin{array}{l} (h = H) \\ T(h) = \frac{h}{2} \cdot \left(f(a) + 2 \cdot \sum_{i=1}^{N-1} f(a + ih) + f(b) \right) \\ \text{Fehler: } -\frac{1}{12} h^2 (b-a) \cdot f''(\xi), \quad \xi \in [a, b] \end{array} \right.$$

$n = 2$: **Zusammengesetzte Simpsonformel**

$$\left\| \begin{array}{l} (h = \frac{H}{2}) \\ S(h) = \frac{h}{3} \cdot \left(f(a) + 4 \cdot \sum_{i=1}^{N} f(a + (2i-1)h) + 2 \cdot \sum_{i=1}^{N-1} f(a + 2ih) + f(b) \right) \\ \text{Fehler: } -\frac{1}{180} h^4 (b-a) \cdot f''''(\xi), \quad \xi \in [a, b] \end{array} \right.$$

Bemerkungen:

1. Für die Ordnung der zusammengesetzten Newton-Cotes Formeln gilt

$$\text{Ordnung} = \begin{cases} n+1 & \text{für } n \text{ ungerade} \\ n+2 & \text{für } n \text{ gerade} \end{cases}$$

(n ist der auf den Teilintervallen benutzte Polynomgrad, nicht die gesamte Knotenanzahl). So ergibt eine genauere Betrachtung des Restgliedes am Beispiel der

zusammengesetzten Simpsonregel:

$$\begin{aligned}\sum_{i=1}^{N}(-\tfrac{1}{90}h^5\cdot f''''(\xi_i)) &= -\tfrac{1}{90}h^5\cdot\underbrace{\sum_{i=1}^{N} f''''(\xi_i)}_{=N\cdot f''''(\hat{\xi})}\\ \text{mit } h=\tfrac{H}{2} &\qquad \overset{\text{nach dem Zwischenwertsatz}}{}\\ &= -\tfrac{1}{90}h^4\cdot\tfrac{1}{2}\tfrac{b-a}{N}\cdot N\cdot f''''(\hat{\xi})\\ &= -\tfrac{1}{180}h^4\cdot(b-a)\cdot f''''(\hat{\xi})\end{aligned}$$

Allgemein gilt: Restglied = const $\cdot\, h^{\text{Ordnung}}\cdot f^{(\text{Ordnung})}(\xi)$, falls f genügend oft differenzierbar ist. Bei niedrigerer Differenzierbarkeit ist entsprechend auch die h-Potenz kleiner ($C^3 \to h^3$ usw.).

2. Der Begriff der Ordnung gibt nicht direkt an, welche Integrationsformel besser ist. Für die Genauigkeit ist nämlich auch das Verhalten der entsprechenden Ableitung entscheidend. So kann durchaus die zusammengesetzte Trapezregel genauer sein als die zusammengestzte Simpsonregel, wenn die vierte Ableitung der Funktion f größere Werte liefert als die zweite.

3. Die zusammengesetzte Trapez- und Simpsonregel konvergieren für $h \to 0$ gegen den gewünschten Integralwert, falls f Riemannintegrierbar ist. Aussagen über die Konvergenzgeschwindigkeit lassen sich aber nur bei differenzierbaren Funktionen angeben.

4. Man kann zeigen, daß für die Integration einer periodischen Funktion über ihre volle Periode (z.B. Bestimmung von Fourierkoeffizienten) die Trapezregel besondere Vorteile hat. Hier gilt

Satz 2.3.1. *Hat f die Periode $b-a$ und ist f auf $[a,b]$ $2m+2$ mal stetig differenzierbar, dann gilt für die zusammengesetzte Trapezregel mit Knotenabstand $h=(b-a)/N$*

$$|\int_a^b f(t)\mathrm{d}t - T(h)| \;\le\; 4(b-a)\Big(\frac{h}{2\pi}\Big)^{2m+2}\max\{|f^{(2m+2)}(x)| \;:\; x \in [a,b]\}$$

Beispiel 2.3.1. *Das folgende Diagramm zeigt die Genauigkeit von zusammengesetzter Trapezregel und zusammengesetzter Simpsonregel für das Integral*

$$\int_{-1}^{1}\frac{1}{10^{-2}+x^2}\;\mathrm{d}x.$$

Die Ableitungen berechnen sich zu

$$f'(x) = \frac{-2x}{(10^{-2}+x^2)^2}, \qquad f''(x) = -2 \cdot \tfrac{10^{-2}-3x^2}{(10^{-2}+x^2)^3},$$
$$f^{(3)}(x) = 24x \cdot \frac{10^{-2}-x^2}{(10^{-2}+x^2)^4}, \quad f^{(4)}(x) = 24 \cdot \tfrac{10^{-4}-10^{-3}x^2+5x^4}{(10^{-2}+x^2)^5},$$

so dass sich die folgenden Maximalwerte ergeben:

$$\max\{|f''(x)|\} = f''(0) = 2 \cdot 10^4, \quad \max\{|f^4(x)|\} = f^4(0) = 24 \cdot 10^6.$$

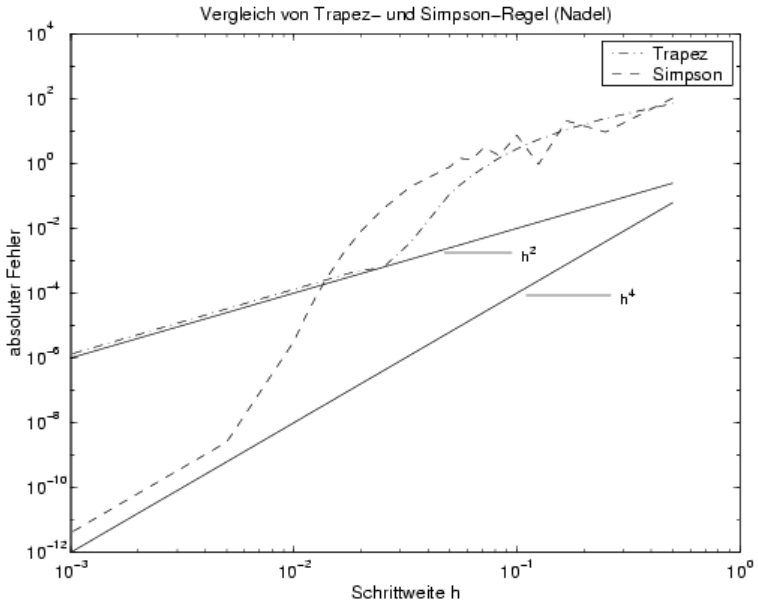


Abb 2.2.1

Man erkennt, daß erst für kleines h die Methode 4. Ordnung tatsächlich der Methode 2. Ordnung überlegen ist.

2.4 Adaptive Quadratur und automatische Kontrolle des Quadraturfehlers

Wenn ein Integral über ein relativ großes Intervall $[a, b]$ numerisch berechnet werden soll, so ist es nicht sinnvoll, eines der bisher besprochenen Verfahren direkt auf $[a, b]$ anzuwenden. Der Quadraturfehler hängt ja vom Verhalten einer der höheren Ableitungen von f ab, und dies kann lokal sehr unterschiedlich sein. So variiert die n–te Ableitung von $\dfrac{x}{x^2-1}$ auf $[1.001, 10]$ zwischen $\frac{1}{2}(-1)^n n!(10^{3n+3} + 2.001^{-3n-3})$ bei $x = 1.001$ und

$\frac{1}{2}(-1)^n n!(11^{-n-1} + 9^{-n-1})$ bei $x = 10$. Entsprechend groß bzw. klein würden in kleinen Teilintervallen die Quadraturfehler. Es ist daher wünschenswert, eine Methode zu besitzen, um eine geeignete Unterteilung des Intervalls zu konstruieren und gleichzeitig den Quadraturfehler zu kontrollieren. Bei genügender Differenzierbarkeit des Integranden gilt für alle bisher und noch im Folgenden besprochenen Quadraturverfahren eine Darstellung des Quadraturfehlers der Form

$$\int_a^b f(t)\mathrm{d}t - \sum_{i=0}^{n} w_i^{(n)} f(t_i^{(n)}) \;=\; c \cdot H^{m+1} + O(H^{m+2}),$$

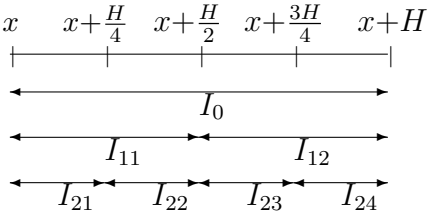
c = Konstante, H = Intervallbreite $= b - a$, m = Ordnung. Z.B. kann man für das Restglied der Simpsonformel

$$-\tfrac{1}{90}(b-a)^5 f^{(4)}(\xi)$$

mit $H = b - a$ auch schreiben

$$-\tfrac{1}{90}H^5 f^{(4)}((a+b)/2) - \tfrac{1}{90}H^5 f^{(5)}(\tilde{\xi})(\xi - (a+b)/2)$$

und der zweite Term ist hierbei $\mathcal{O}(H^6)$. Wir betrachten nun die Anwendung einer solchen Formel auf einer mehrfachen Unterteilung des gleichen Grundintervalls der Länge H



$$I - I_0 = cH^{m+1} + O(H^{m+2})$$
$$I = \int_a^b f(t)\mathrm{d}t$$

Abbildung 2.4.1

Wir stellen uns vor, die Intervallbreite H sei "klein", sodaß Terme der Ordnung $\mathcal{O}(H^{m+2})$ vernachlässigbar sind gegenüber Termen der Ordnung $\mathcal{O}(H^{m+1})$. Wir wenden die gleiche Formel nun weiterhin einmal auf dem Teilintervall $[x,\, x + \frac{H}{2}]$ und auf dem Teilintervall $[x + \frac{H}{2},\, x + H]$ an. Addition beider Werte liefert eine Näherung $I_1 = I_{11} + I_{12}$ für I mit

$$I - I_1 = 2c(\frac{H}{2})^{m+1} + O(H^{m+2})\,.$$

Daher wird

$$I_1 - I_0 = cH^{m+1}(1 - 2^{-m}) + O(H^{m+2})$$

oder

$$cH^{m+1} = \frac{I_1 - I_0}{1 - 2^{-m}} + O(H^{m+2}) = I - I_0 + O(H^{m+2})\,.$$

Wenn der O–Term vernachlässigbar ist (d.h. H "genügend" klein), dann gilt also

$$I - I_0 \approx \frac{I_1 - I_0}{1 - 2^{-m}}\,. \tag{2.1}$$

Man kann das Ergebnis (2.1) nun leicht zur Konstruktion einer geeigneten Intervallunterteilung benutzen. Vorgegeben sei eine Genauigkeitsforderung

$$|I - \sum_{j=0}^{N} I_0^{(j)}| \leq \delta\,,$$

wobei $I_0^{(j)}$ die Integralnäherung auf dem Teilintervall $[x_j,\, x_j + H_j]$ bedeute. Diese Forderung wird sicher erfüllt, wenn

$$|\int_{x_j}^{x_j+H_j} f(t)\,\mathrm{d}t - I_0^{(j)}| \leq \frac{\delta H_j}{b-a}\,,$$

oder, wegen (2.1) approximativ erfüllt, wenn

$$|I_1^{(j)} - I_0^{(j)}| \leq \frac{(1-2^{-m})\delta H_j}{b-a}\,. \tag{2.2}$$

Sei x_j schon konstruiert und $\tilde{H}_j$ eine Vorschlagsschrittweite für H_j (aus dem davorliegenden Schritt, $\tilde{H}_j \leq b - x_j$). Dann berechnet man $I_0^{(j)}$, $I_1^{(j)}$ wie oben beschrieben.
Es ist also

$$I_1^{(j)} - I_0^{(j)} \approx c\tilde{H}_j^{m+1}(1-2^{-m}).$$

Die mit (2.2) maximal verträgliche Schrittweite H_j habe die Form

$$H_j = \kappa \tilde{H}_j.$$

Es soll dann also gelten,

$$c(\kappa\tilde{H}_j)^m \approx \delta/(b-a)$$

d.h.

$$\kappa^m \approx \left|\frac{\delta}{(b-a)c\tilde{H}_j^m}\right| \approx \frac{\delta(1-2^{-m})\tilde{H}_j}{(b-a)|I_1^{(j)} - I_0^{(j)}|}$$

oder

$$\kappa = \left(\frac{\delta(1-2^{-m})\tilde{H}_j}{(b-a)|I_1^{(j)} - I_0^{(j)}|}\right)^{1/m}.$$

Der Faktor κ ist somit berechenbar. Falls $\kappa \geq 1$ wird der Schritt akzeptiert, d.h. $I_1^{(j)}$ als Wert des Teilintegrals auf $[x_j, x_j + \tilde{H}_j]$ akzeptiert und

$$\begin{aligned} x_{j+1} &= x_j + \tilde{H}_j \\ \tilde{H}_{j+1} &= \max\{1, \min\{0.9\kappa, 2\}\}\tilde{H}_j \end{aligned}$$

gesetzt. 2 als maximaler Vergrößerungsfaktor stellt dabei eine Sicherheitsschranke dar. Ist dagegen $\kappa < 1$, wird der laufende Schritt verworfen,

$$\tilde{H}_j \stackrel{def}{=} 0.9\kappa\tilde{H}_j$$

gesetzt und die Berechnung von $I_1^{(j)}, I_0^{(j)}$ wiederholt. (0.9 stellt dabei einen praxistypischen "Sicherheitsfaktor" dar.) Gleichzeitig beachtet man, daß H_j niemals eine (sinnvoll gewählte) obere Schranke überschreitet, (z.B. $\min\{0.1, \frac{b-a}{10}\}$). Die Schrittweitenreduktion muß man natürlich abbrechen, wenn $H_j \approx \varepsilon|x_j|$, $\varepsilon =$ Rechengenauigkeit. In diesem Fall kann man davon ausgehen, daß bei x_j eine Singularität des Integranden vorliegt. Die systematische Anwendung dieser Überlegungen führt uns auf folgenden Algorithmus:

Adaptive Quadratur
Daten: $a,\ b,\ \delta,\ H_{\min},\ H_{\max}$. Integrand: f. Erzeugte Intervalleinteilung: $\{x_k\}$. Integralnäherung: I, Quadraturformel: $int(f, a, b)$.

$$\begin{aligned} k &= 0; \\ x_0 &= a; \\ I &= 0; \\ \tilde{H} &= (b-a)/10; \text{ Versuchsschrittweite} \\ fin &= false; \end{aligned}$$

While not fin

$$\begin{aligned} I_0 &= int(f, x_k, x_k + \tilde{H}); \\ I_1 &= int(f, x_k, x_k + \tilde{H}/2) + int(f, x_k + \tilde{H}/2, x_k + \tilde{H}); \end{aligned}$$

If $I_0 \neq I_1$

$$\kappa = \Big(\frac{(1-2^{-m})\delta\tilde{H}}{(b-a)|I_0 - I_1|}\Big)^{1/m};$$

else $\kappa = 2$;
endif
If $\kappa \geq 1$

$$\begin{aligned} I &= I + I_1; \\ x_{k+1} &= x_k + \tilde{H}; \end{aligned}$$

If $x_{k+1} \geq b$

$$fin = true;$$

end if.

$$\tilde{H} = \min\{H_{\max},\ \max\{1,\ \min\{0.9\kappa, 2\}\}\tilde{H}\} ;$$

If $x_{k+1} + \tilde{H} > b$

$$\tilde{H} = b - x_{k+1};$$

end if .

$$k = k + 1;$$

else

if $\tilde{H} \leq H_{\min}$
stop: Genauigkeit nicht erreichbar;
end if

$$\tilde{H} = 0.9\kappa\tilde{H};$$

endif
end while

Die folgende Abbildung zeigt ein typisches Resultat dieser Vorgehensweise. Bei 0.3 liegt ein sehr steiler "Peak" von f vor. Jede weitere Ableitung von f wächst größenordnungsmäßig um den Faktor 10^4. Solange man auf den Peak zu integriert, wird die Vorschlagsschrittweite ständig reduziert (viele verworfene Schritte). Dahinter wird die Schrittweite allmählich wieder vergrößert, die Steuerung verhält sich "ruhig". Hier wird mit der Simpsonformel als Grundformel und einer Genauigkeitsforderung von $\delta = 10^{-4}$ gearbeitet. Die maximale Schrittweite ist 0.1. Der Integrand ist

$$f(x) = 1/((x-0.3)^2 + 0.001) + 1/((x-0.9)^2 + 0.04) - 6 \text{ auf } [0,1]$$

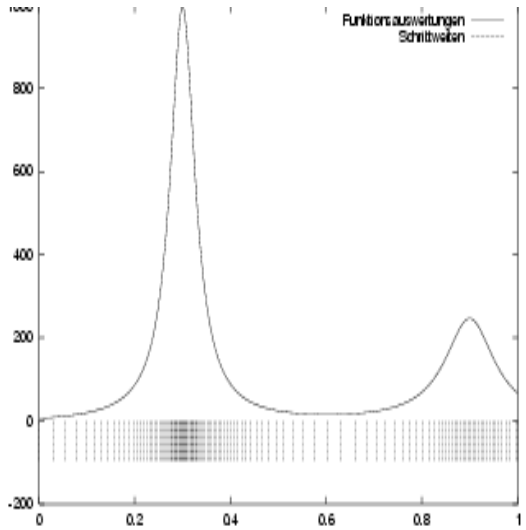


Abb 2.4.2

Beispiel 2.4.1. *Für die Funktion* $f(x) = \frac{1}{1+225x^4}$ *soll mittels adaptiver Quadratur und unter Verwendung der Simpsonformel das Integral in den Grenzen von 0 bis 1 bestimmt werden. Als Vorschlagsschrittweite für den ersten Schritt sei* $\tilde{H}_0 = \frac{1}{4}$ *gegeben. Wir untersuchen, ob diese Schrittweite akzeptabel ist, wenn eine Fehlertoleranz von* $\delta =$

10^{-4} *gefordert wird. Zunächst müssen im Interval* $[0, \frac{1}{4}]$ *mit der Simpsonregel und der summierten Simpsonregel mit zwei Teilintervallen zwei Integralnäherungen bestimmt werden. Die benötigten Funktionswerte sind*

x	*0*	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{3}{16}$	$\frac{1}{4}$
$f(x)$	*1*	*0.996578*	*0.947928*	*0.782416*	*0.532225*

Die Integralnäherungen sind

- *einfache Simpsonregel:*

$$F_1 = \frac{1}{24}\left(f(0) + 4f(\frac{1}{8}) + f(\frac{1}{4})\right) = 0.2218308$$

- *summierte Simpsonregel:*

$$F_2 = \frac{1}{48}\left(f(0) + 4f(\frac{1}{16}) + 2f(\frac{1}{8}) + 4f(\frac{3}{16}) + f(\frac{1}{4})\right) = 0.2196680$$

Mit diesen Werten läßt sich nun κ *berechnen, wobei wir beachten, daß die Simpsonformel die Ordnung* $m = 4$ *besitzt: Der Test auf die Akzeptanz der Schrittweite* $\tilde{H}_0 = 0.25$ *ergibt*

$$\kappa \;=\; \left|\frac{10^{-4}\frac{15}{16}\frac{1}{4}}{1(0.2218308 - 0.2196680)}\right|^{1/4} = 0.3226\ .$$

Die Schrittweite ist also ***nicht*** *akzeptabel. Wir würden einen neuen Versuch starten etwa mit der Intervallbreite 0.075.*

NUMAWWW Quadratur/adaptive Quadratur

2.5 Gauß-Quadratur

Bisher haben wir - zumindest intervallweise - immer äquidistante Knotenabstände benutzt. Dies ist auch die gegebene Vorgehensweise, wenn z.B. die "Daten" $f(x_i)$ nur diskret (z.B. aus Messungen) gegeben sind. Die berechtigte Frage ist nun, ob wir vielleicht bessere Formeln - im Sinne der erzielbaren Ordnung der Integrationsformel - erhalten können, wenn wir die Knoten anders verteilen.
Die Antwort liefert der folgende Satz, der die Knotenverteilung zur Erzielung der optimalen Ordnung angibt:

Satz 2.5.1. *Es gibt genau eine Quadraturformel der Ordnung* $2n+2$ *mit* $n+1$ *Knoten*

$$\int_a^b f(t)\,\mathrm{d}t = \sum_{i=0}^{n} w_i^{(n)} \cdot f(t_i^{(n)}) \quad \textit{für} \quad f \in \Pi_{2n+1}$$

Dabei gilt: $t_i^{(n)}$ *sind die auf* $[a,b]$ *transformierten Nullstellen des* $(n+1)$*-ten Legendre-Polynoms und für* $w_i^{(n)}$*, die Gewichte, gilt*

$$w_i^{(n)} = \int_a^b \underbrace{L_{i,n}(t)}_{\text{Lagrange-Polynom zu } t_i^{(n)}} \mathrm{d}t = \int_a^b L_{i,n}^2(t)\,\mathrm{d}t > 0 \quad i = 1,\ldots,n.$$

Das Restglied hat die Form

$$\frac{2^{2n+3}((n+1)!)^4}{(2n+3)((2n+2)!)^3} \cdot \left(\tfrac{b-a}{2}\right)^{2n+3} \cdot f^{(2n+2)}(\xi)$$

Diese Formel ist nach Gauß benannt. □

Bei der Gauß-Quadratur benötigt man also die Nullstellen der Legendre-Polynome, die wir hier mit P_i bezeichnen. Die Legendre-Polynome sind wie folgt rekursiv definiert

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x \\ P_{n+1}(x) &= \tfrac{2n+1}{n+1} \cdot x \cdot P_n(x) - \tfrac{n}{n+1} \cdot P_{n-1}(x), \quad n \geq 1. \end{aligned}$$

Sie sind *Orthogonalpolynome* zur Gewichtsfunktion 1 auf $[-1,1]$, d.h. es gilt

$$\int_{-1}^{1} P_i(x) \cdot P_j(x)\,\mathrm{d}x = 0 \quad \text{für} \quad i \neq j$$

und sie haben nur einfache reelle Nullstellen $s_i^{(n)} \in]-1,1[$. (siehe Tabelle weiter unten). Die in Satz 2.5.1 genannten auf das Intervall $[a,b]$ transformierten Nullstellen $t_i^{(n)}$ ergeben sich mittels

$$t_i^{(n)} = \tfrac{b-a}{2} \cdot s_i^{(n)} + \tfrac{b+a}{2}$$

Bemerkung:

Das Prinzip der Gauß-Quadratur läßt sich übertragen auf den allgemeineren Fall von Integralen mit Gewichtsfunktion
$\int_I \omega(x) f(x)\,\mathrm{d}x$ und auch auf unendliche Intervalle I. Von der Gewichtsfunktion ω benötigt

man nur, daß sie positiv ist, höchstens abzählbar viele Nullstellen hat und daß die (eventuell uneigentlichen) Integrale

$$\int_I \omega(x)x^k \mathrm{d}x$$

für alle ganzen $k \geq 0$ existieren (endlich sind). An die Stelle der $t_i^{(n)}$ in Satz 2.5.1 treten dann die Nullstellen $z_i^{(n)}$ der entsprechenden Orthogonalpolynome. Man erhält so den Formeltyp

$$\sum_{i=0}^{n} \underbrace{\omega_i}_{\substack{\omega(x) \text{ ist hier versteckt} \\ \omega_i^{(n)} = \int_I \omega(x)\cdot L_{i,n}(x)\,\mathrm{d}x}} \cdot f(z_i^{(n)}).$$

Ein Beispiel (die sogenannte Gauss-Hermite-Interpolation):

$$\int_{-\infty}^{\infty} exp(-x^2)f(x)\mathrm{d}x \;=\; \frac{\sqrt{\pi}}{6}\Big(f(-\frac{\sqrt{\pi}}{6}) + 4f(0) + f(\frac{\sqrt{\pi}}{6})\Big) + \frac{\sqrt{\pi}}{960}f^{(6)}(\xi)\ .$$

Dies ist besonders nützlich bei Funktionen mit integrierbaren Singularitäten wie z.B.

$$\int_0^r \ln(x)f(x)dx \text{ mit glattem } f\ ,\ \text{wo } \omega(x) \;=\; -\ln(x)$$

oder

$$\int_0^r \sqrt{x}f(x)dx \text{ mit glattem } f\ ,\ \text{wo } \omega(x) \;=\; \sqrt{x}$$

Bemerkung 2.5.1. *In der Praxis kann man natürlich das Quadraturrestglied in der Regel nicht mit analytischen Methoden abschätzen. Man verwendet dann gerne ein Paar von Formeln, nämlich eine Gauss-Formel und eine weitere, die die gleichen Knoten und noch einige weitere benutzt, und eine höhere Ordnung hat. Häufig verwendet werden die Gauß-Kronrod-Formeln. Dies sind Paare von Quadraturformeln der Ordnung* $2n$ *und* $3n+2$ *bzw.* $3n+3$*. Die erste ist eine "normale" Gaußformel mit* n *Knoten, die zweite entsteht, indem man zu diesen* n *Knoten* $n+1$ *weitere hinzufügt und verlangt, daß die entstehende Formel die Ordnung* $3n+2$ *hat. Dadurch ergeben sich die Zusatzknoten eindeutig. Die Differenz der beiden Näherungswerte ist dann zugleich eine Schätzung des Quadraturfehlers der ungenaueren Formel. Alle diese Knoten sind innere Knoten und die entstehenden Formeln liefern konvergente Verfahren für jede Riemannintegrierbare Funktion.*

Beispiel 2.5.1. $[a,b] = [0,1]$*,* $n = 1$*, also die Rechteckregel, ergänzt um 2 Knoten ergibt eine Gauss-Kronrod-Formel der Ordnung 6:*

Knoten	*Gauss-Gewicht*	*Kronrod-Gewicht*
0.112701665379258	*0*	*5/18*
0.5	*1*	*4/9*
*0.887298334620741*8	*0*	*5/18*

Beispiel 2.5.2. *Wir berechnen näherungsweise das Integral mit einer Dreipunktformel (also Ordnung 6)*

$$\int_2^3 \frac{e^t}{t}\mathrm{d}t$$

und schätzen den Quadraturfehler ab. Die Gauß-Quadraturformel lautet in der allgemeinen Form

$$\int_a^b f(t)\mathrm{d}t \approx \frac{b-a}{2}\sum_{k=0}^{n}\beta_k^{(n)} f(\bar{x}_k^{(n)}) \quad mit \quad \bar{x}_k^{(n)} = \frac{b-a}{2}x_k^{(n)} + \frac{b+a}{2}.$$

In folgender Tabelle sind die Nullstellen des Legendre-Polynoms zweiter Ordnung $x_k^{(2)}$, die transformierten Stützstellen $\bar{x}_k^{(2)}$, die Gewichte $\beta_k^{(2)}$, die Punktauswertungen von f und die Produkte $\beta_k^{(2)} f(\bar{x}_k^{(2)})$ angegeben.

k	$x_k^{(2)}$	$\bar{x}_k^{(2)}$	$\beta_k^{(2)}$	$f(\bar{x}_k^{(2)})$	$\beta_k^{(2)} f(\bar{x}_k^{(2)})$
0	$-\sqrt{\frac{3}{5}}$	*2.112702*	*0.5555556*	*3.914682*	*2.174823*
1	*0*	*2.5*	*0.8888889*	*4.872998*	*4.331553*
2	$\sqrt{\frac{3}{5}}$	*2.887298*	*0.5555556*	*6.215071*	*3.452817*
				$\frac{1}{2}\Sigma =$	*4.979597*

Die exakte Darstellung des Integral ist

$$\int_2^3 \frac{e^t}{t}\mathrm{d}t = \frac{1}{2}\sum_{k=0}^{2}\beta_k^{(n)} f(\bar{x}_k^{(n)}) + \left(\frac{d^6}{dt^6}\frac{e^t}{t}\right)_{t=\xi}\frac{(3!)^4}{7(6!)^3}.$$

Weil

$$\left|\frac{d^6}{dt^6}\frac{e^t}{t}\right| \leq 190 \text{ für } t \in [2,3]$$

ergibt sich dann eine Fehlerabschätzung von

$$\left|\int_2^3 \frac{e^t}{t}\mathrm{d}t - 4.979597\right| \leq 190\frac{(3!)^4}{7(6!)^3} = 9.42\cdot 10^{-5}.$$

Tabelle von Gauß–Legendre–Formeln:

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{i=0}^{n} w_i^{(n)} f(t_i^{(n)})$$

Es gilt:

$t_i^{(n)} = -t_{n-i}^{(n)}$ und $w_i^{(n)} = w_{n-i}^{(n)}$ $i = 0, ..., n$
Tabelliert sind
$t_i^{(n)}$, $i = n, n-1, ..., \lfloor n/2 \rfloor + 1$ $w_i^{(n)}$, $i = n, n-1, ..., \lfloor n/2 \rfloor + 1$ also, wegen der Antisymmetrie bzw. Symmetrie nur "die rechte Hälfte" der Daten, und zwar in der linken Spalte die Knoten und in der rechten die Gewichte.

$n = 0$

0.00000000000000	2.000000000000000

$n = 1$

0.577350269189626	1.000000000000000

$n = 2$

0.774596669241483	0.555555555555556
0.000000000000000	0.888888888888889

$n = 3$

0.861136311594053	0.347854845137454
0.339981043584856	0.652145154862546

$n = 4$

0.906179845938664	0.236926885056189
0.538469310105683	0.478628670499366
0.000000000000000	0.568888888888889

$n = 5$

0.932469514203152	0.171324492379170
0.661209386466265	0.360761573048139
0.238619186083197	0.467913934572691

$n = 6$

0.949107912342759	0.129484966168870
0.741531185599394	0.279705391489277
0.405845151377397	0.381830050505119
0.000000000000000	0.417959183673469

$n = 7$

0.960289856497536	0.101228536290376
0.796666477413627	0.222381034453374
0.525532409916329	0.313706645877887
0.183434642495650	0.362683783378362

$n = 8$

0.968160239507626	0.081274388361574
0.836031107326636	0.180648160694857
0.613371432700590	0.260610696402935
0.324253423403809	0.312347077040003
0.000000000000000	0.330239355001260

$n = 9$

0.973906528517172	0.066671344308688
0.865063366688985	0.149451349150581
0.679409568299024	0.219086362515982
0.433395394129247	0.269266719309996
0.148874338981631	0.295524224714753

2.6 Uneigentliche Integrale

Integrale mit integrierbaren Randsingularitäten oder über unendlichen Intervallen treten häufig in den Anwendungen auf. Zu ihrer Behandlung gibt es verschiedene Möglichkeiten. Eine besteht darin, die Singularität als Gewichtsfaktor abzuspalten und eine spezielle Gaußformel zu benutzen. Eine zweite in einer additiven Abspaltung eines analytisch integrierbaren Anteils. Diese beiden Methoden erfordern aber eine spezielle Betrachtung des jeweiligen Integranden. Daneben gibt es auch universell einsetzbare Methoden. Eine vielfach bewährte besteht in der Anwendung einer modifizierten Gauß-Quadratur mit automatischer Genauigkeitskontrolle. Grundlage sind die Gauß-Kronrod-Formeln (s.o.). Die Differenz der beiden Näherungswerte ist dann zugleich eine Schätzung des Quadraturfehlers der ungenaueren Formel. Integrale über unendliche Intervalle zerlegt man zunächst in solche über Intervalle der Form $[b, \infty[$ und transformiert dann dieses Intervall mittels der Substitution

$$z = 1/(x - b + 1)$$

auf $]0, 1]$.

NUMAWWW Quadratur: Gauss-Kronrod-Formeln, uneigentliche Integrale

2.7 Bereichsintegrale

In der Praxis stellt sich häufig die Aufgabe, Integrale über höherdimensionale Bereiche zu approximieren. In zwei oder drei Dimensionen sind Adaptationen oder Verallgemeinerungen der Methoden für eine Veränderliche sinnvoll einsetzbar. Ist z.B. B ein Normalbereich im $\mathbb{R}^2$ und $\int_B f(x, y)\,\mathrm{d}(x, y)$ gesucht, dann bietet sich sofort die Darstellung

durch ein iteriertes Integral an. O.B.d.A. sei

$$B = \{(x,y) : \ a \le x \le b, \ \psi_1(x) \le y \le \psi_2(x)\}$$

dann wird

$$\int_B f(x,y)\,\mathrm{d}(x,y) = \int_a^b F(x)\mathrm{d}x.$$

mit

$$F(x) \ = \ \int_{\psi_1(x)}^{\psi_2(x)} f(x,y)\,\mathrm{d}y$$

Entsprechend verwenden wir nun die Quadratur in iterierter Form. Bei festem x benutzt man nun eine gewöhnliche Quadratur bzgl. y auf $[\psi_1(x), \psi_2(x)]$ unter der bereits bekannten Transformation für Knoten und Gewichte und gelangt so zum Wert für $F(x)$. Das Integral über F wird dann wieder standardmässig behandelt. Sind etwa $\tilde{w}_i^{(m)}, t_i^{(m)}$ Gewichte und Knoten der Formel für das Intervall [-1,1], dann ergibt Intervalltransformation die Formel

$$\left.\begin{array}{rcl} w_i^{(m)}(x) & = & \frac{\psi_2(x)-\psi_1(x)}{2}\tilde{w}_i^{(m)} \\ y_i^{(m)}(x) & = & \frac{\psi_2(x)-\psi_1(x)}{2}t_i^{(m)} + \frac{\psi_1(x)+\psi_2(x)}{2} \end{array}\right\} i = 0,\ldots,m.$$

Für das verbleibende Integral wird nun wieder eine Quadraturformel (eventuell eine andere) verwendet und man gelangt schließlich zu einer Formel des Typs

$$\sum_{k=0}^{n} w_k^{(n)} \sum_{i=0}^{m} w_i^{(m)}(x_k^{(n)}) f(x_k^{(n)}, y_i^{(m)}(x_k^{(n)})).$$

Sei z.B. $a = 0, b = 1, \psi_1(x) = 0, \psi_2(x) = 1 + 4x^2$, und die zugrunde liegende Formel in beiden Fällen die Simpsonformel, d.h. $n = m = 2$

$$\begin{array}{rclrclrcl} t_0^{(2)} & = & -1, & t_1^{(2)} & = & 0, & t_2^{(2)} & = & 1, \\ w_0^{(2)} & = & \frac{1}{3}, & w_1^{(2)} & = & \frac{4}{3}, & w_2^{(2)} & = & \frac{1}{3}, \end{array}$$

dann wird

$$\begin{aligned} \int_B f(x,y)\,\mathrm{d}(x,y) \ \approx \ & \tfrac{1}{6}\Big(\tfrac{1}{6}(f(0,0) + 4f(0,\tfrac{1}{2}) + f(0,1)) + \\ & \tfrac{4}{3}(f(\tfrac{1}{2},0) + 4f(\tfrac{1}{2},1) + f(\tfrac{1}{2},2)) + \\ & \tfrac{5}{6}(f(1,0) + 4f(1,\tfrac{5}{2}) + f(1,5))\Big). \end{aligned}$$

Als weiteres Beispiel betrachten wir eine Anwendung der Gauss-Quadratur.

Beispiel 2.7.1. *Es sei der Bereich*

$$B = \{(x, y) : -1 \leq x \leq 1 \ , \ 0 \leq y \leq 1 - x^2\}$$

gegeben. Wir fragen nach der Anzahl und der Plazierung der Quadraturknoten, wenn man eine beliebige affin lineare Funktion auf B exakt integrieren will. Die Umschreibung in ein Doppelintegral ergibt

$$\int_B f(x, y)\mathrm{d}(x, y) = \int_{-1}^{1} \int_{0}^{1-x^2} f(x, y)\mathrm{d}y\mathrm{d}x.$$

Das innere Integral ist ein Integral über eine lineare Funktion in y. Damit ist in y-Richtung ein Knoten in der Intervallmitte $\frac{b+a}{2} = \frac{1-x^2}{2}$ nötig um das Integral exakt zu bestimmen. Das zugehörige Gewicht lautet $b - a = 1 - x^2$.

Durch diese exakte Integration ist das äußere Integral

$$\int_{-1}^{1} \underbrace{(1 - x^2) \cdot f\left(x, \frac{1 - x^2}{2}\right)}_{F(x)} \mathrm{d}x \ .$$

Der Integrand $F(x)$ ist ein Polynom 4. Grades in x und wird folglich durch 3 Knoten in x-Richtung bei $-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}$ exakt integriert. Die Gewichte lauten $\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$.

Durch Anwendung der Gauß-Quadratur auf F ergibt sich nach Einsetzen von f die Quadraturformel:

$$\int_{-1}^{1} F(x)\mathrm{d}x = \frac{5}{9} F\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} F(0) + \frac{5}{9} F\left(\sqrt{\frac{3}{5}}\right) =$$

$$= \frac{5}{9}\left(1 - \left(-\sqrt{\frac{3}{5}}\right)^2\right) f\left(-\sqrt{\frac{3}{5}}, \frac{1 - \left(-\sqrt{\frac{3}{5}}\right)^2}{2}\right) + \frac{8}{9} f\left(0, \frac{1}{2}\right) +$$

$$+ \frac{5}{9}\left(1 - \left(\sqrt{\frac{3}{5}}\right)^2\right) f\left(\sqrt{\frac{3}{5}}, \frac{1 - \left(\sqrt{\frac{3}{5}}\right)^2}{2}\right) =$$

$$= \frac{2}{9} f\left(-\sqrt{\frac{3}{5}}, \frac{1}{5}\right) + \frac{8}{9} f\left(0, \frac{1}{2}\right) + \frac{2}{9} f\left(\sqrt{\frac{3}{5}}, \frac{1}{5}\right)$$

□

Ein Nachteil dieser Vorgehensweise besteht darin, daß die dabei benötigte Anzahl an Funktionsauswertungen sehr schnell sehr groß wird. Es gibt auch spezielle, an die Geometrie angepaßte Formeln. Z.B. ist für das Standarddreieck

$$T_0 = \{(x, y) : \ 0 \leq x \leq 1, \ 0 \leq y \leq 1 - x\}$$

die Schwerpunktregel

$$\int_{T_0} f(x,y)\,\mathrm{d}(x,y) \approx \tfrac{1}{2} f(\tfrac{1}{3},\tfrac{1}{3})$$

exakt für affin lineares f und die Formel von Collatz und Albrecht

$$\int_{T_0} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx B(f(r,r)+f(r,s)+f(s,r)) + C(f(u,u)+f(u,v)+f(v,u))$$

mit

$$r=\tfrac{1}{2},\quad s=0,\quad ,u=\tfrac{1}{6},\quad v=\tfrac{4}{6},\quad ,B=\tfrac{1}{60},\quad C=\tfrac{9}{60},$$

ist für Polynome vom Gesamtgrad ≤ 3 exakt. Mit nur sieben Funktionsauswertungen kann man mit der Formel von Radon

$$\begin{aligned}\int_{T_0} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \quad &\approx \quad Af(t,t)+B(f(r,r)+f(r,s)+f(s,r))+\\ &\qquad C(f(u,u)+f(u,v)+f(v,u))\end{aligned}$$

mit

$$\begin{gathered}A=\tfrac{9}{80},\quad B=(155-\sqrt{15})/2400,\quad C=(155+\sqrt{15})/2400,\\ t=\tfrac{1}{3},\quad u=(6+\sqrt{15})/21,\quad v=(9-2\sqrt{15})/21,\\ r=(6-\sqrt{15})/21,\quad s=(9+2\sqrt{15})/21,\end{gathered}$$

Polynome in x und y vom Gesamtgrad ≤ 5 exakt integrieren. Mit iterierter Gaußquadratur benötigt man dazu bereits 12 Funktionswerte.

Durch Triangulierung von B und Anwendung der Transformationsregel kann man dann beliebige Bereichsintegrale in $\mathbb{R}^2$ annähern: Man summiert die Teilintegrale über die einzelnen Dreiecke auf. Für ein beliebiges Dreieck der Triangulierung benutzt man zur Auswertung die Transformation auf das Standarddreieck

$$\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_{T_0} f(x(\xi,\eta),y(\xi,\eta))|\det\,(A_T)|\,\mathrm{d}\xi\,\mathrm{d}\eta,$$

wobei (x,y) mit (ξ,η) durch die lineare Abbildung

$$\begin{pmatrix} x(\xi,\eta)\\ y(\xi,\eta)\end{pmatrix} = \begin{pmatrix} x_i\\ y_i\end{pmatrix} + A_T \begin{pmatrix}\xi\\ \eta\end{pmatrix}$$

mit der festen, nur von der Geometrie abhängigen Matrix

$$A_T = \begin{pmatrix}(x_j-x_i) & (x_k-x_i)\\ (y_j-y_i) & (y_k-y_i)\end{pmatrix}$$

verknüpft ist. Dabei sind $(x_i,y_i),(x_j,y_j),(x_k,y_k)$ die drei Ecken von T, die (in dieser Reihenfolge) auf (0,0), (1,0), (0,1) abgebildet werden. (Krummlinig berandete Bereiche werden dabei zunächst durch polygonal berandete approximiert.)

Diese Methoden funktionieren auch noch im $\mathbb{R}^3$ gut (s. z.B. S. Stroud: Approximate calculation of multiple integrals), aber in höheren Dimensionen wird der Aufwand untragbar. Hier hilft nur noch die sogenannte Monte-Carlo-Quadratur:

Sind x_i identisch und unabhängig verteilt mit Dichte $g(x)$ auf B, dann gilt für alle $\varepsilon > 0$

$$\mathcal{P}(\Big| \int_B f(x)\,\mathrm{d}x - |B|\tfrac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{g(x_i)} \Big| > \varepsilon) \longrightarrow_{n\to\infty} 0.$$

$\mathcal{P}(\,$ Aussage $)$ bedeutet hier "Wahrscheinlichkeit der Gültigkeit von Aussage". Dabei ist $|B|$ das Volumen von B. Der Erwartungswert des Fehlers in der Monte-Carlo-Approximation ist dabei $\mathcal{O}(\frac{1}{\sqrt{N}})$, unabhängig von der Dimension von B.

2.8 Zusammenfassung

Bei der numerischen Quadratur hat man als zentrale Forderungen die beliebig genaue Approximation des Integrals zumindest für jede stetige Funktion und gleichzeitig die Forderung einer möglichst kleinen Anzahl von Funktionsauswertungen bei gegebener Genauigkeit zumindest für gutartige Funktionen (deren höhere Ableitungen alle existieren und nicht zu schnell anwachsen). Die Formeln bzw. Verfahren müssen dabei vom Integranden unabhängig sein.

Brauchbare Verfahren in diesem Sinne sind die zusammengesetzten abgeschlossenen Newton-Cotes Formeln bis zur Ordnung 8 und die Gauss-Quadratur. Zentraler Begriff in der Quadratur ist der der "Ordnung". Als Merkregel kann man benutzen "die Ordnung einer Quadraturformel ist gleich der Ordnung der Ableitung des Integranden im Quadraturfehler" wenn man beliebig hohe Differenzierbarkeit unterstellt.

Die Gaussformeln liefern die grösstmögliche Ordnung bei gegebener Knotenzahl, die Ordnung ist das Doppelte der Knotenzahl. (Als Knoten bezeichnet man die Abszissen, an denen der Integrand ausgewertet werden muss.) Der Nachteil der Gauss-Formeln besteht darin, daß ihre Anwendung die Auswertbarkeit des Integranden an beliebigen Stellen erfordert.

Auch die Newton-Cotes Formeln ungerader Knotenzahl haben eine Ordnungserhöhung (um eins).

Die Newton-Cotes Formeln erlauben auch die Quadratur von nur tabellarisch gegebenen Funktionen.

Singuläre Integrale führt man durch Substitution in nichtsinguläre über oder benutzt nach Standardtransformation auf ein festes Intervall Gauss-Quadratur sehr hoher Ordnung. Dabei darf eine Singularität nur an den Intervallenden auf treten.

Mehrdimensionale Integrale behandelt man nach Zerlegung in Normalbereiche als iterierte Integrale mit iterierter Quadratur oder wendet Spezialformeln an, die von der

Form des Grundbereichs abhängen. Letzteres ist praktikabel z. B. bei simplizialen Zerlegungen des Bereichs.

Kapitel 3

Anfangswertprobleme gewöhnlicher Differentialgleichungen

3.1 Problemstellung

In diesem Kapitel behandeln wir die numerische Lösung des Anfangswertproblems gewöhnlicher Differentialgleichungen:

$$\begin{array}{rl} \text{Gegeben :} & (t_0, y_0) \ \in [a,b] \times \mathcal{D},\ f \ : \ [a,b] \times \mathcal{D} \ \rightarrow \ \mathbb{R}^n \\ \text{Gesucht :} & y \text{ mit } y' \ = \ f(t,y)\ , \quad y(t_0) \ = \ y_0 \end{array} \tag{3.1}$$

Dabei kann y (und damit f) ein Skalar oder ein Vektor beliebiger Dimension sein. Diese Aufgabenstellung hat mannigfache Anwendungen in der Technik, u.a. in der Robotik, der Fahrdynamik, der Schaltkreisanalyse, der chemischen Reaktionskinetik .

NUMAWWW Differentialgleichungen, gew. Differentialgleichungen

Wir werden stets die Bedingungen des folgenden Existenz- und Eindeutigkeitssatzes (Picard-Lindelöf) voraussetzen:

Satz 3.1.1. *Es sei f in einer Umgebung $\mathcal{U}$ des Punktes (t_0, y_0) bezüglich y Lipschitzstetig, d.h.*

$$||f(t,y_1) - f(t,y_2)|| \ \leq \ L||y_1 - y_2|| \quad \textit{für alle } (t,y_1),\ (t,y_2) \ \in \mathcal{U}\ .$$

mit geeignetem L ($||.||$ sei die euklidische Länge eines Vektors) und bezüglich (t,y) stetig. Dann hat das Problem 3.1 hat genau eine Lösung in einer (eventuell kleineren) Umgebung dieses Punktes. Ist f $p-$fach stetig partiell differenzierbar nach allen Variablen, dann ist die Lösung y $(p+1)-$fach stetig differenzierbar und hängt von den Anfangswerten und eventuellen Parametern in f $p-$fach differenzierbar ab.

Unter den Gültigkeitsbedingungen bilden die Lösungen der Differentialgleichung eine Kurvenschar, die den Raum überdeckt. 2 Kurven der Schar schneiden sich nie. Dies alles gilt lokal. Globale Existenz ist gesichert für den Fall linearer Differentialgleichungen mit beschränkten Koeffizienten.

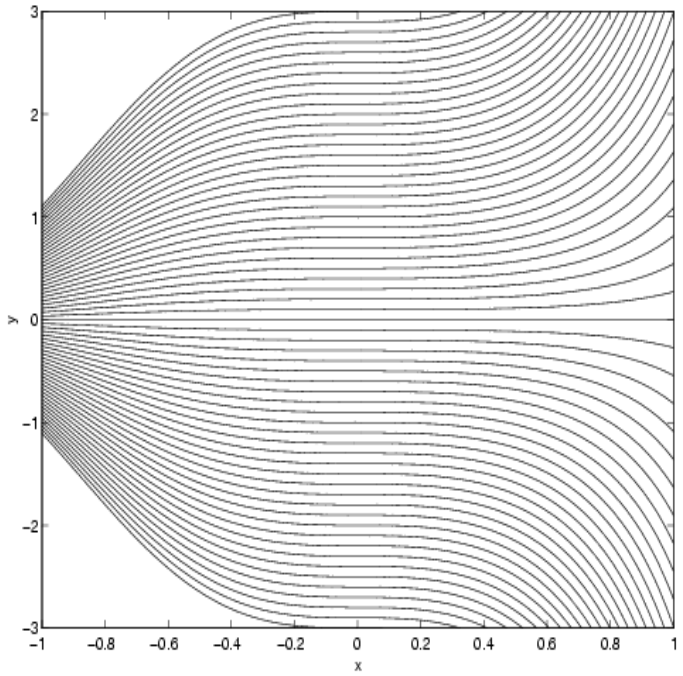


Abb 3.0.1: Lösungsschar einer linearen DGL

Für nichtlineares f existiert die Lösung häufig tatsächlich nur in einem kleinen Intervall um t_0, auch wenn f überall definiert und beliebig oft differenzierbar ist.

Beispiel 3.1.1. $y' = -2ty^2$ *und* $y(t_0) = y_0$
$f(t,y) = -2ty^2 \quad (\in \mathbb{R})$

$$\begin{aligned} |f(t,y_1) - f(t,y_2)| &= 2 \cdot |t| \cdot |{y_1}^2 - {y_2}^2| \\ &\leq 2 \cdot |t| \cdot |y_1 + y_2| \cdot |y_1 - y_2| \\ &\leq 4 \cdot (|t_0| + b) \cdot (|y_0| + \varepsilon) \cdot 2 \cdot |y_1 - y_2| \end{aligned}$$

damit ergibt sich $L = 8 \cdot (|t_0|+b) \cdot (|y_0|+\varepsilon)$ *solange* $|y(t)| \leq |y_0|+\varepsilon$ *und* $t_0 \leq t \leq t_0+b$.
$f(t,y)$ *ist beliebig oft differenzierbar, aber :*

$$\begin{aligned} \tfrac{\mathrm{d}y}{\mathrm{d}t} = -2ty^2 \quad &\Rightarrow \quad \tfrac{\mathrm{d}y}{y^2} = -2t\,\mathrm{d}t \\ \Rightarrow \int_{y_0}^{y} \tfrac{\mathrm{d}\eta}{\eta^2} &= -2 \cdot \int_{t_0}^{t} \tau\,\mathrm{d}\tau = -t^2 + t_0^2 = \tfrac{1}{y_0} - \tfrac{1}{y} \\ \Rightarrow y(t) &= \frac{1}{t^2 + \frac{1}{y_0} - t_0^2} \end{aligned}$$

Die Lösung ist also z.B. für $t_0 = 0,\ y_0 = -1$ *nur für* $|t| < 1$ *beschränkt.*

Die folgende Abbildung zeigt die Lösung des einleitenden Problems aus Kapitel 1 für mehrere Werte des Dämpfungsparameters r_0.

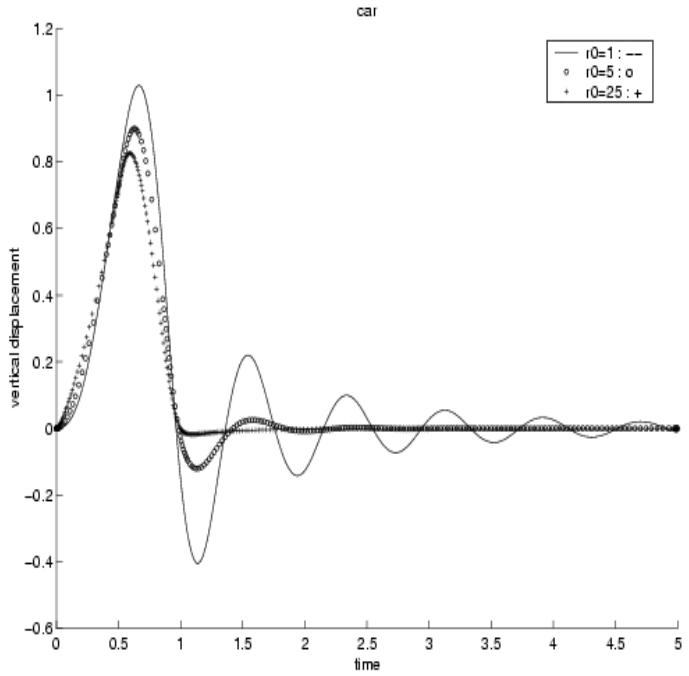


Abb 3.0.2

Unser Ziel wird es sein, die gesuchte Funktion y nur auf einem Gitter $t_0 < t_1 < \ldots < t_N = t_{end}$ anzunähern durch Gitterwerte

$$y_i^h \approx y(t_i)$$

Für theoretische Zwecke nehmen wir hier ein äquidistantes Gitter an, in der Praxis benutzt man aber stets adaptive Gitter, die auf eine zur adaptiven Quadratur analoge Weise erzeugt werden.

Es gibt zwei prinzipielle Möglichkeiten dazu: Wir können versuchen, direkt $y'(t_i)$ mit Hilfe der Differentiation eines Interpolationspolynoms durch Gitterwerte auszudrücken und daraus eine Gleichung für den nächsten Gitterwert zu gewinnen, z.B.

$$y'(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i}$$

mit der resultierenden Verfahrensvorschrift

$$y_{i+1}^h = y_i^h + hf(t_i, y_i^h) \quad \text{Euler vorwärts}$$

oder wir benutzen den Hauptsatz der Differential- und Integralrechnung

$$y(t) = \underbrace{y(t_i)}_{=y_i} + \int_{t_i}^{t} y'(\tau)\,\mathrm{d}\tau = \underbrace{y(t_i)}_{=y_i} + \int_{t_i}^{t} f(\tau, y(\tau))\,\mathrm{d}\tau$$

mit $t = t_{i+1}$. Das Integral wird dann mit einer Quadraturformel näherungsweise bestimmt.
Man unterstellt dabei, daß alles "gut geht", d.h. beim numerischen Rechnen geht man von der Existenz der Lösung im betrachteten Zeitintervall aus.

3.2 Einschrittverfahren (ESV)

Im Folgenden bezeichnet y_i^h den Näherungswert für $y(t_i)$, der mit einem Diskretisierungsverfahren mit Schrittweite $h > 0$ berechnet wurde, d.h. $y_i^h \approx y(t_i)$ Als Einschrittverfahren bezeichnet man Verfahren, die den nächsten Gitterwert y_{i+1}^h nur mit Hilfe von y_i^h (und der Differentialgleichung) bestimmen. Allgemein lautet die Verfahrensvorschrift eines ESV

$$y_{i+1}^h = y_i^h + h \cdot \Phi(t_i, y_i^h, h), \qquad i = 0, 1, \ldots, N-1$$

Die Funktion $\Phi(t_i, y_i^h, h)$ heißt **Schrittfunktion** oder **Inkrementfunktion** Manchmal ist es nützlich, die Inkrementfunktion allgemeiner als

$$\Phi(t_i, y_i^h, y_{i+1}^h, h)$$

zu schreiben. Φ ist unter Umständen eine nur implizit (über ein weiteres Gleichungssystem) definierte Funktion. Diese Einschrittverfahren sind leicht handhabbar und vergleichsweise unempfindlich gegen schnell variierende Lösungen. Bei nicht extrem hohen Genauigkeitsanforderungen sind sie auch genügend effizient.

Beispiel 3.2.1.

1. *Euler (vorwärts)*
   $$\begin{aligned} y_0^h &= y(t_0) \\ y_{i+1}^h &= y_i^h + h \cdot f(t_i, y_i^h) \end{aligned}$$ **explizites Verfahren**
   Dabei wurde das Integral näherungsweise mit der Rechteckregel berechnet, wobei das linke Intervallende als Stützpunkt verwendet wurde: $\int_{t_i}^{t_{i+1}} g(\tau)\,\mathrm{d}\tau \approx h \cdot g(t_i)$.

2. *Euler (rückwärts)*

$$\begin{aligned} y_0^h &= y(t_0) \\ y_{i+1}^h &= y_i^h + h \cdot f(t_{i+1}, y_{i+1}^h) \qquad \textbf{implizites Verfahren} \end{aligned}$$

Dabei wurde das Integral näherungsweise mit der Rechteckregel berechnet, wobei das rechte Intervallende als Stützpunkt verwendet wurde: $\int_{t_i}^{t_{i+1}} g(\tau)\,\mathrm{d}\tau \approx h \cdot g(t_{i+1})$

Die Schrittfunktion $\Phi(t_i, y_i^h, h)$ *ist hier nur implizit definiert. Man muss in jedem Zeitschritt ein lineares oder nichtlineares Gleichungssystem lösen, um die nächste Näherung zu erhalten. Deshalb stellt sich nun die Frage, ob man die Gleichungen lokal eindeutig nach* y_{i+1}^h *auflösen kann. Um diese Frage zu beantworten schreibe man:*
$y_{i+1}^h = y_i^h + h \cdot f(t_i + h, y_{i+1}^h) \iff$
$y_{i+1}^h - y_i^h - h \cdot f(t_i + h, y_{i+1}^h) = 0$ *und fasse dies als nichtlineare Gleichung (Gleichungssystem) für* y_{i+1}^h *auf zu gegebenen Werten* $t_i,\ h,\ y_i^h$

$$F(y_{i+1}^h, y_i^h, t_i, h) \overset{def}{=} y_{i+1}^h - y_i^h - h \cdot f(t_i + h, y_{i+1}^h) = 0$$

$h = 0$ *liefert die Lösung* $y_{i+1}^h = y_i^h$. *Nun kann man den Satz über die impliziten Funktionen auf* F *anwenden und erhält als Bedingung für die lokal eindeutige Auflösbarkeit in einer Umgebung von* $(t_i, y_i^h, 0)$ *für hinreichend kleines* $h > 0$ *die Invertierbarkeit der partiellen Funktionalmatrix von* F *bezüglich der Variablen* y_{i+1}^h, *d.h.*

$$\frac{\partial}{\partial y_{i+1}^h} F(y_i^h, y_{i+1}^h, h) = I - 0 - h \cdot f_y(t_i + h, y_{i+1}^h) \cdot I \text{ muß invertierbar sein .}$$

Dies ist offensichtlich der Fall für genügend kleines h. *Man kann sogar* $h > 0$ *beliebig wählen, wenn* f_y *nur Eigenwerte mit nichtpositiven Realteilen hat.*

Also ist die Gleichung lokal eindeutig nach y_{i+1}^h *lösbar.*

3. *Näherung: Trapezregel. Um das Integral näherungsweise zu berechnen, benutzen wir nun die Trapezregel, womit sich für das Integral ergibt:*

$$\begin{aligned} \int_{t_i}^{t_{i+1}} g(\tau)\,\mathrm{d}\tau &\approx \tfrac{h}{2} \cdot [g(t_i) + g(t_{i+1})] \\ &= \tfrac{h}{2} \cdot [f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))] \end{aligned}$$

Damit erhält man das implizite Verfahren:

$$\begin{aligned} y_0^h &= y(t_0) \\ y_{i+1}^h &= y_i^h + \tfrac{h}{2} \cdot (f(t_i, y_i^h) + f(t_{i+1}, y_{i+1}^h)) \end{aligned}$$

Ersetzt man rechts die implizite Größe y_{i+1}^h nach dem Euler-vorwärts-Verfahren: ${y_{i+1}}^h = {y_i}^h + h \cdot f(t_i, {y_i}^h)$ so erhält man die

4. *Näherung: modifiziertes Eulerverfahren, 1.Verfahren von Heun:*

$$y_{i+1}^h = y_i^h + h \cdot \underbrace{\tfrac{1}{2}\Big(f(t_i, y_i^h) + f(t_i + h, y_i^h + h \cdot f(t_i, y_i^h))\Big)}_{=:\Phi(t_i, y_i^h, h)}$$

Dieses 1. Verfahren von Heun ist nun ein explizites Verfahren, doch es ist nicht linear in f.

Beispiel 3.2.2. *Wir betrachten das Anfangswertproblem $y' = f(x,y), y(a) = y_a$, mit*

$$f(t,y) = -2ty^2 \quad \textit{und} \quad t_0 = 1,\ y_0 = \frac{1}{2}.$$

und approximieren die Lösung mit dem Euler- und mit dem Heun-Verfahren . Die wahre Lösung ist $y(t) = 1/(1+t^2)$. Als Schrittweite wählen wir $h = 0.1$.

a) Euler vorwärts :

$$\begin{array}{rclcrcl} k_1 & = & f(t_i, y_i) & \Rightarrow & k_1 & = & f(1, 0.5) = -0.5 \\ y_{i+1} & = & y_i + h \cdot k_1 & \Rightarrow & y_1 & = & y_a + h \cdot k_1 = 0.5 - 0.05 = 0.45 \end{array}$$

Der Fehler ist $|y_1 - y(1.1)| = 2.489 \cdot 10^{-3}$.

b) Heun :

$$\begin{array}{rclcrcl} k_1 & = & f(t_i, y_i) & \Rightarrow & k_1 & = & -0.5 \\ k_2 & = & f(t_{i+1}, y_i + h \cdot k_1) & \Rightarrow & k_2 & = & -2 \cdot 1.1 \cdot 0.45^2 = -0.4455 \\ y_{i+1} & = & y_i + \frac{h}{2}(k_1 + k_2) & \Rightarrow & y_1 & = & 0.452725 \end{array}$$

Der Fehler ist $|y_1 - y(1.1)| = 2.363 \cdot 10^{-4}$.

□

Im Folgenden werden Runge-Kutta-Verfahren betrachtet, die einfach handhabbar und bei nicht zu hohen Genauigkeitsforderungen auch hinreichend effizient sind. Explizite Runge-Kutta-Verfahren sind die praktisch am häufigsten verwendeten ESV.

Allgemeines Runge-Kutta-Verfahren: Herleitung Wir gehen aus von der Volterra-Integralgleichung

$$y(t+h) = y(t) + \int_t^{t+h} f(\tau, y(\tau))\,\mathrm{d}\tau, \qquad g(\tau) \stackrel{def}{=} f(\tau, y(\tau))$$

1. Schritt: Das Integral wird mit einer Quadraturformel mit Knoten $\alpha_i \in [0,1]$ angenähert; die Gewichte seien $\gamma_i \Rightarrow$

$$\int_t^{t+h} g(\tau)\,\mathrm{d}\tau \approx h \cdot \sum_{i=1}^{m} g(t+\alpha_i h)\gamma_i$$

2. Schritt: da $g(t+\alpha_i h) = f(t+\alpha_i h, y(t+\alpha_i h))$ und sich wieder $y(t+\alpha_i h) = y(t) + \int_t^{t+\alpha_i h} g(\tau)\,\mathrm{d}\tau$ ergibt, erhält man:

$$g(t+\alpha_i h) = f(t+\alpha_i h, \underbrace{y(t) + \int_t^{t+\alpha_i h} g(\tau)\,\mathrm{d}\tau}_{=y(t+\alpha_i h)})$$

Hier sind alle Funktionswerte ausser für $\alpha_i = 0$ noch unbekannt.

3. Schritt: Setze $k_i(t,h) := g(t+\alpha_i h)$

4. Schritt: Das Integral $\int_t^{t+\alpha_i h} g(\tau)\,\mathrm{d}\tau$ berechnet man mit einer Quadraturformel und verwendet als Knoten die gleichen Knoten α_i, wie für das erste Integral "außen", d.h. man verwendet wieder die Werte k_i als Funktionswerte. Dies ist nun eine etwas andere Vorgehensweise als in Kapitel 2 nur insofern, als die Knoten nun auch ausserhalb des Integrationsintervalls liegen dürfen. Die Gewichte für diese "innere" Quadraturformel β_{il} werden entsprechend angepaßt, normalerweise mit dem Ziel, für die Quadratur eine möglichst hohe Ordnung zu erzielen.

Ergebnis: m Gleichungen für die Werte $k_1, \ldots, k_m$ an der Stelle (t_j, y_j^h):

$$k_i(t_j,h) = f(t_j + \alpha_i h, y_j^h + h \cdot \sum_{l=1}^{m} \beta_{il} k_l),$$

wobei β_{il} die Gewichte für die "innere" Integration sind.
Damit ergibt sich das **Runge-Kutta-Verfahren**:

$$y_{j+1}^h \;=\; y_j^h + h \cdot \sum_{i=1}^{m} \gamma_i k_i(t_j,h)$$

Das Runge-Kutta-Verfahren wird also vollständig beschrieben durch folgendes Schema: "Butcher array"

$$\begin{array}{c|cccc} \alpha_1 & \beta_{11} & \cdots & \cdots & \beta_{1m} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ \alpha_m & \beta_{m1} & \cdots & \cdots & \beta_{mm} \\ \hline & \gamma_1 & \cdots & \cdots & \gamma_m \end{array}$$

Beispiele:

Euler (vorwärts):

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad m = 1$$

Euler (rückwärts):

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad m = 1$$

Trapezregel:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad m = 2$$

Heun:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad m = 2$$

Herleitung des Butcher-arrays für die Trapezregel:

$$\begin{aligned} k_1 &= f(t + 0 \cdot h, y + h \cdot (0 \cdot k_1 + 0 \cdot k_2)) = f(t, y) \\ k_2 &= f(t + 1 \cdot h, y + h \cdot (\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2)) = f(t + h, y + h(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2)) \\ y(t + h) &\approx y + h \cdot (\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2) \\ &= y + \tfrac{h}{2}k_1 + \tfrac{h}{2}k_2 \\ &= y + \tfrac{h}{2} \cdot f(t, y) + \tfrac{h}{2} \cdot f(t + h, y + \tfrac{h}{2}k_2 + \tfrac{h}{2}k_1) \\ y_{i+1}^h &= y_i^h + \tfrac{h}{2}(f(t_i, y_i^h) + f(t_i + h, y_{i+1}^h)) \ . \end{aligned}$$

Allgemeine Runge-Kutta-Verfahren sind

- **explizit,** wenn $\beta_{ij} = 0$ für $j \geq i$,
  d.h. $k_1, k_2, \ldots, k_m$ können nacheinander ausgerechnet werden, durch je eine Auswertung von f.
- **implizit,** sonst

"Das" klassische Runge-Kutta-Verfahren ist ein explizites Verfahren für $m = 4$.

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \qquad \left.\begin{aligned} k_1 &= f(t, y) \\ k_2 &= f(t + \tfrac{h}{2}, y + \tfrac{h}{2}k_1) \\ k_3 &= f(t + \tfrac{h}{2}, y + \tfrac{h}{2}k_2) \\ k_4 &= f(t + h, y + hk_3) \end{aligned}\right\} y_{i+1}^h = y_i^h + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Beispiel 3.2.3. *Wir wenden dieses Verfahren auf die gleiche Problemstellung wie im vorangegangenen Beispiel an.*

$$\begin{aligned} k_1 &= f(x_i, y_i) && \Rightarrow\ k_1 &= -0.5 \\ k_2 &= f(x_i + \tfrac{h}{2}, y_i + \tfrac{h}{2} \cdot k_1) && \Rightarrow\ k_2 &= -0.4738125 \\ k_3 &= f(x_i + \tfrac{h}{2}, y_i + \tfrac{h}{2} \cdot k_2) && \Rightarrow\ k_3 &= -0.476428303 \\ k_4 &= f(x_i + h, y_i + h \cdot k_3) && \Rightarrow\ k_4 &= -0.450179419 \\ y_{i+1} &= y_i + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) && \Rightarrow\ y_1 &= 0.45248898289 \end{aligned}$$

Der Fehler ist $|y_1 - y(1.1)| = 2.9511 \cdot 10^{-7}$.

Aufgrund unserer Ergebnisse in Kapitel 2 wissen wir, daß eine minimale Voraussetzung für eine vernünftige Quadraturformal lautet: "Summe der Gewichte gleich Intervalllänge", d.h. hier

$$\boxed{\sum_{j=1}^{m} \beta_{ij} = \alpha_i, \qquad \sum_{i=1}^{m} \gamma_i = 1}$$

Im Folgenden werden nun **Konvergenzaussagen** betrachtet. Diese beschreiben das Verhalten der Lösungsverfahren für $h \to 0$. Zu diesem Zweck betrachten wir nun ein $t \neq t_0$, o.B.d.A. $t > t_0$. Die Stelle t werde nach N Schritten erreicht, d.h. $Nh = t - t_0$. Wenn wir nun h gegen null gehen lassen, geht damit N gegen unendlich. Wir erwarten, daß dann $y_N^h \to y(t)$. Den Fehler zwischen den Vektoren messen wir dabei in einer sogenannten Norm, symbolisch $||.||$. In diesem Kapitel können wir unter der Norm die euklidische Länge eines Vektors verstehen.

Dies wird festgehalten in

Definition 3.2.1. *Ein Einschrittverfahren* $y_{i+1}^h = y_i^h + h \cdot \Phi(t_i, y_i^h, h)$ *heißt* **konvergent,** *falls für* $t \in [t_0, t_0 + b]$, *mit* $b > 0$ *geeignet (abhängig von AWA) gilt:*

$$y_N^h \to y(t), \quad \textit{falls } N \cdot h = t - t_0 \ \textit{ fest } \ \textit{und } h \to 0.$$

Es heißt **konvergent von der Ordnung** p, *falls*

$$\|y_N^h - y(t)\| \leq c \cdot h^p, \quad c = c(t, f) > 0 \ \textit{ geeignet und } h \ \leq \ h_0 > 0$$

für beliebig oft differenzierbare rechte Seite f.

Beispiel: Das klassische Runge Kutta Verfahren ist ein Verfahren, welches konvergent von der Ordnung 4 ist. Der Fehler

$$y(t) - y_N^h \ \text{ mit } Nh \ = \ t - t_0 \ \text{ fest}$$

wird als **globaler Diskretisierungsfehler** bezeichnet. Er ist das Resultat der Auswirkung der N Fehler in den einzelnen Schritten des Verfahrens, die selbst Quadraturfehler oder Differentiationsfehler sind. Ob ein zurückliegender Fehler in den späteren Schritten verstärkt oder wieder gedämpft wird, ist keine Eigenschaft des Verfahrens, sondern hängt von der Differentialgleichung ab. Das Fehlerverhalten des Verfahrens wird beschrieben durch den sogenannten **lokalen Abschneidefehler**. Dieser Fehler entsteht in Schritt i nicht mehr an der Stelle $(t_i, y(t_i))$, sondern irgendwo im Definitionsbereich der Differentialgleichung. Man hat ja im Schritt i die wahre Lösung längst verlassen.

Deshalb wird in der folgenden Definition nicht die Lösung des Anfangswertproblems, sondern die Lösungskurve durch einen beliebigen Punkt (t, z) betrachtet.

Definition 3.2.2. *Der* **lokale Abschneidefehler** *eines Einschrittverfahrens (ESV) an einer beliebigen Stelle (t, z) ist definiert als*

$$\varrho(h, z, t) = \frac{y(t+h) - z}{h} - \Phi(t, z, h),$$

wobei $y(t+h)$ die Lösung der Anfangswertaufgabe $y' = f(t, y), \quad y(t) = z$ an der Stelle $t + h$ ist.

Nach Definition von ϱ ist also

$$y_{i+1}^h \;=\; y_i^h + h \cdot \Phi(t_i, y_i^h, h) = \tilde{y}(t_{i+1}) - h \cdot \varrho(h, y_i^h, t_i)$$

wobei $\tilde{y}$ die Lösung der Anfangswertaufgabe (AWA)

$$y' = f(t, y), \qquad y(t_i) = y_i^h$$

ist. Also ist $h\varrho(h, y_i^h, t_i)$ der an der Stelle (t_i, y_i^h) neu hinzukommende Integrationsfehler, den man auch als **lokalen Diskretisierungsfehler** bezeichnet.
So wie Quadraturverfahren verschiedene Ordnungen haben, haben die daraus abgeleiteten Integrationsverfahren für Differentialgleichungen verschiedene Konvergenzordnungen p (d.h. der globale Diskretisierungsfehler geht wie Ch^p gegen null. Diese hängen zusammen mit der Grössenordnung des Quadraturfehlers auf dem Intervall der Breite h. Zur Erinnerung: In Kapitel 2 hatten wir "Ordnung $p \Rightarrow$ Quadraturfehler ist $\mathcal{O}(\text{Intervallbreite}^{p+1})$". Dies drückt sich aus in

Definition 3.2.3. *Das ESV hat die* **Konsistenzordnung** p**,** *falls*

$$\|\varrho(h, z, t)\| \leq C \cdot h^p \quad \textit{mit} \quad C > 0 \quad \textit{geeignet}$$

$t \in [t_0, t_0 + b]$, z geeignet beschränkt.

Dies bedeutet, daß man von einem ESV nicht nur Konvergenz erwartet, sondern auch daß

$$\Phi(t, y, h) = \frac{y_1^h - y(t)}{h} \longrightarrow y'(t) = f(t, y), \quad y_0^h = y(t)$$

falls $h \to 0$, d.h. daß Φ die 1. Ableitung von y approximiert.

Die Konsistenordnung erhält man, indem man formal die "wahre Lösung" $y(t)$ in die Verfahrensvorschrift einsetzt und diese sowie $y(t+h)$ an der Stelle $(t, y(t))$ nach Taylor entwickelt, wobei man die Ordnung dieser Entwicklung nach der erwarteten Konsistenzordnung wählt. Man faßt dann gleiche $h-$Potenzen zusammen und erhält dann aus der Forderung einer gewissen Konsistenzordnung Gleichungen für die Parameter der

Methode. Dies wird sehr schnell sehr verwickelt und für die allgemeinen Runge-Kutta-Verfahren gibt es dazu eine eigens entwickelte Technik, die der "Butcher-Bäume". Wir begnügen uns hier mit einem einfachen Beispiel.

Beispiel 3.2.4. *Konsistenzordnung des Heun-Verfahrens:*
Zuerst die Taylorentwicklung von y:

$$y(t+h) = y(t) + y'(t)\cdot h + y''(t)\cdot \tfrac{h^2}{2} + y'''(t)\cdot \tfrac{h^3}{6} + \mathcal{O}(h^4)$$

Die Verfahrensvorschrift lautet:

$$y_1^h = y(t) + \tfrac{h}{2}\cdot(\underbrace{f(t,y(t))}_{=y'(t)} + f(t+h, y(t) + h\cdot \underbrace{f(t,y(t))}_{=:k})$$

Die Taylorentwicklung der Funktion f (wobei t und y als unabhängige Variablen zu behandeln sind) lautet

$$\begin{aligned} f(t+h, y+h\cdot k) = \; & f(t,y) + f_t\cdot h + f_y(h\cdot k) \\ & + \tfrac{1}{2}\cdot(f_{tt}\cdot h^2 + 2\cdot f_{ty}\cdot h^2\cdot k + f_{yy}(hk)^2) \\ & + \mathcal{O}(h^3) \end{aligned}$$

Mit den Setzungen $y' = f, \quad y'(t) = f(t,y(t)) = k$ haben wir

$$y''(t) = f_t(t,y(t))\cdot 1 + f_y(t,y(t))\cdot \underbrace{y'(t)}_{f} = f_t + f_y\cdot f = f_t + f_y k$$

und

$$\begin{aligned} y'''(t) &= (y''(t))' = (f_t + f_y\cdot f)' \\ &= f_{tt} + 2f_{ty}\cdot f + f_{yy}\cdot f^2 + f_y\cdot(f_t + f_y\cdot f) \\ &= f_{tt} + 2f_{ty}\cdot f + f_{yy}\cdot f^2 + f_y\cdot f_t + (f_y)^2\cdot f \end{aligned}$$

Dies ergibt

$$y_1^h - y(t+h) = \mathcal{O}(h^3) \stackrel{!}{=} -h\cdot\varrho(h,y(t),t)$$

Also

$$\varrho(h,y(t),t) = \mathcal{O}(h^2) \Rightarrow \quad \textit{Konsistenzordnung 2.}$$

□

Für Einschrittverfahren besteht zwischen Konsistenzordnung und Konvergenzordnung unter minimalen Bedingungen Übereinstimmung:

Satz 3.2.1. *Das ESV sei konsistent von der Ordnung p. Ferner gelte für Φ: Es gibt ein $L > 0$, so daß*

$$\|\Phi(t,z_1,h) - \Phi(t,z_2,h)\| \le L\cdot\|z_1 - z_2\|$$

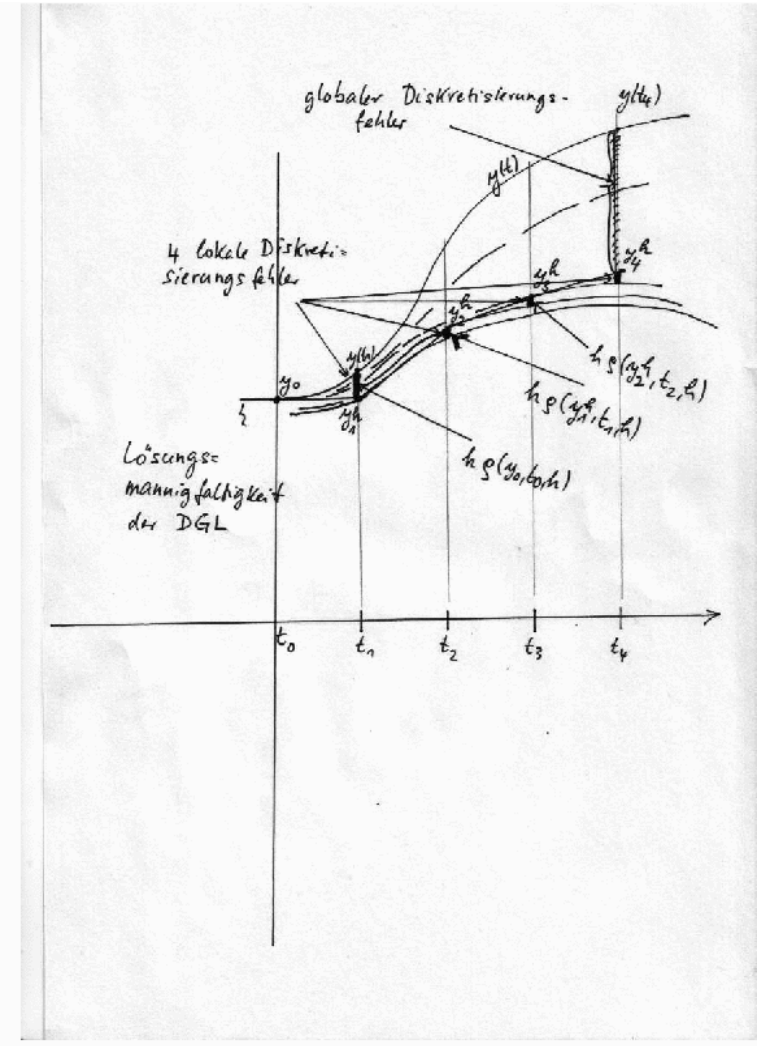
für alle $(t,z_1),(t,z_2)$ in einer geeigneten Umgebung der Lösung $(t,y(t))$, $t \in [t_0, t_0+b]$ und für alle $h \in [0,h_0]$.
Dann ist das Verfahren **konvergent von der Ordnung** p**.**

Die Bedingungen für Φ sind in den praktisch interessanten Fällen (Runge-Kutta) immer erfüllt!

Achtung: In Definition 3.2.1 ist die Konstante (bezüglich h) $c(t)$ unter Umständen riesig! Sie ergibt sich nämlich zu

$$c(t) \;=\; \exp(L \cdot (t - t_0)) \cdot \tilde{C} \;,$$

wobei L die Lipschitzkonstante von f und $\tilde{C}$ eine weitere Konstante ist, die vom maximalen Betrag einiger höherer partieller Ableitungen von f auf einer Umgebung der Lösung abhängt.
Die folgende Abbildung zeigt den Zusammenhang von lokalem und globalem Diskretisierungsfehler :



3.3 Absolute (lineare) Stabilität von ESV

Konsistenz und Konvergenz beschreiben das Verhalten des Diskretisierungsverfahrens für $h \to 0$. In der Praxis rechnet man aber nie mit beliebig kleinen Schrittweiten, ja man will sogar oft über ein sehr grosses Zeitintervall integrieren und dies natürlich mit möglichst grossen Schrittweiten. Dann tritt ein ganz neues Problem auf, nämlich das der Stabilität des Diskretisierungsverfahrens für festes endliches $h > 0$:

Die numerische Näherungslösung soll das Verhalten der exakten Lösung zumindest qualitativ reproduzieren. Um dies zu untersuchen, wendet man die Verfahren auf eine bestimmte Test- oder Modellgleichung an und beurteilt das Verhalten der Lösung. Dabei erhält man die Eigenschaft der linearen Stabilität, wenn die Testgleichung linear ist.

Wir betrachten hier die **Modellgleichung** $y' = \lambda y, \quad y(t_0) = y_0$ mit $\mathrm{Re}\,\lambda < 0$. Diese Modellgleichung kann man sich hervorgegangen denken aus der allgemeineren (und praktisch bedeutsameren)

$$y' \;=\; Ay + g(t)\;, \quad y(t_0) = y_0 \quad (*)$$

mit einer festen diagonalähnlichen Matrix A. D.h.

$$\exists T \text{ invertierbar, mit } T^{-1}AT \;=\; \mathrm{diag}(\lambda_1, \ldots, \lambda_n)\;.$$

Setzt man dann

$$z \;\stackrel{def}{=}\; T^{-1}y$$

dann erhält man das entkoppelte System

$$z_i' \;=\; \lambda_i z_i + (T^{-1}g(t))_i,\; i = 1, \ldots, n\;, \quad z_i(t_0) = z_{i,0}\;.$$

Da eine reine Inhomogenität die Stabilität der Differentialgleichung nicht beeinflusst, kann man sich das System (*) aus der Linearisierung von

$$y' \;=\; F(t, y)\;, \quad y(t_0) \;=\; y_0$$

an einer festen Stelle t, etwa t_0, hervorgegangen denken mit

$$\hat{y}'(t) \;=\; F_y(t_0, y_0)(\hat{y}(t) - y_0) + F(t_0, y_0) + F_t(t_0, y_0)(t - t_0)$$

in der Hoffnung, zumindest in einer kleinen Umgebung von t_0 das Stabilitätsverhalten der nichtlinearen Gleichung durch das der Linearisierung beschreiben zu können. Dies ist allerdings nur in sehr beschränktem Masse der Fall. (H.O. Kreiss hat ein lineares DGL-System mit variablen Koeffizienten $A(t)$ konstruiert, das exponentiell anwachsende Lösungen besitzt, obwohl für jedes t alle Eigenwerte von $A(t)$ negativen Realteil haben.) Da das Abklingverhalten der DGL nicht von einer rein zeitabhängigen Inhomogenität abhängt, beschränkt man sich schliesslich sogar auf den Fall $g \equiv 0$.

Es gibt auch eine Stabilitätstheorie für die Integrationsverfahren bei nichtlinearen DGLen, auf die wir hier aber nicht eingehen können.

Für uns macht es also keinen wesentlichen Unterschied statt des Systems (*) die skalare Gleichung

$$y' \;=\; \lambda y \;,\; y(t_0) \;=\; y_0$$

zu betrachten. In diesem Zusammenhang ist der Fall $\text{Re}\ \lambda < 0$ interessant. Dann gilt $|y(t_i)| \to 0$ falls $t_i \;\to\; \infty$.
Nun stellt sich die Frage, ob für die verschiedenen Verfahren auch die Näherungslösungen sich so verhalten, d.h. $|y_i^h| \stackrel{?}{\longrightarrow} 0$ für endliches $h > 0$ und $i \;\to\; \infty$
Dies ist sicher eine Minimalforderung! Wir untersuchen unsere elementaren Integratoren:

1. **Euler-vorwärts:**

   $y_{i+1}^h = y_i^h + h \cdot (\lambda \cdot y_i^h) = (1 + h\lambda) \cdot y_i^h$
   Damit $|y_i^h| \to 0$ muß also gelten:
   $|1 + \lambda h| < 1$
   $|1 + \lambda h| = \sqrt{(1 + h \cdot \ \text{Re}\ \lambda)^2 + (\ \text{Im}\ \lambda)^2 h^2}$
   $|1 + \lambda h| < 1$ ist erfüllt für reelles $\lambda < 0$, wenn $h|\lambda| < 2$
   Man beachte, daß z.B. für rein imaginäres λ diese Forderung mit diesem Verfahren nie erfüllbar ist.

2. **Euler rückwärts:**

$$\begin{aligned} y_{i+1}^h &= y_i^h + \lambda h \cdot y_{i+1}^h \\ y_{i+1}^h &= \frac{1}{1 - h\lambda} \cdot y_i^h \\ \left|\frac{1}{1 - h\lambda}\right| &= \frac{1}{\sqrt{(1 - h \cdot \ \text{Re}\ \lambda)^2 + (h \cdot \ \text{Im}\ \lambda)^2}} < 1 \end{aligned}$$

   Dies ist erfüllt für alle $h > 0$, falls $\text{Re}\ \lambda < 0$.

3. analog ergibt sich für die **Trapezregel:**
   $y_{i+1}^h \;=\; \frac{1+h\lambda/2}{1-h\lambda/2} y_i^h$
   keine Einschränkung für h

4. **1. Verfahren von Heun:**
   Wir haben $\left|1+\lambda h+\frac{h^2\lambda^2}{2}\right| < 1$ als Forderung an h. Für reelles λ also $-2 \;<\; h\lambda \;<\; 0$.
   Bei den expliziten Verfahren hat man immer eine (u.U. starke) Einschränkung an h.

Einschrittverfahren, angewendet auf die skalare Modellgleichung, ergeben allgemein einen Zusammenhang der Form $y^h_{i+1} = g(h\lambda)y^h_i$. Die Funktion $g(.)$ heisst dabei die **Stabilitäts-** oder **Verstärkungsfunktion:** ,

$$\oplus \qquad y^h_{i+1} = g(h\cdot\lambda)\cdot y^h_i \begin{cases} \text{Euler explizit:} & g(h\lambda) = 1 + h\lambda \\ \text{Heun (Ordnung 2):} & g(h\lambda) = 1 + h\lambda + \frac{h^2\lambda^2}{2} \\ \text{Trapezregel:} & g(h\lambda) = \frac{1+h\cdot\lambda/2}{1-h\cdot\lambda/2} \\ \text{Euler (impl.):} & g(h\lambda) = \frac{1}{1-h\lambda} \\ \text{Klass. Runge-Kutta V.:} & g(h\lambda) = 1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6} + \frac{h^4\lambda^4}{24} \end{cases}$$

Beispiel 3.3.1. *Das folgende Butcher-Schema definiert ein Runge-Kutta-Verfahren:*

$$\begin{array}{c|cc} \frac{1}{2} & +\frac{1}{8} & \frac{3}{8} \\ \frac{3}{4} & 1 & -\frac{1}{4} \\ \hline & +\frac{1}{2} & \frac{1}{2} \end{array}$$

Wir bestimmen die Stabilitätsfunktion dafür: Für die angegebene Testgleichung ergeben sich die folgenden Bestimmungsgleichungen für die Hilfsfunktionen k_1 und k_2:

$$\begin{aligned} k_1 &= \lambda\Big(y^h_i + \frac{1}{8}hk_1 + \frac{3}{8}hk_2\Big), \\ k_2 &= \lambda\Big(y^h_i + hk_1 - \frac{1}{4}hk_2\Big). \end{aligned}$$

Dies führt mit der Setzung $z = h\lambda$ auf das lineare Gleichungssystem

$$\begin{bmatrix} 1 - \frac{1}{8}z & -\frac{3}{8}z \\ -z & 1 - \frac{1}{4}z \end{bmatrix} \cdot \begin{bmatrix} hk_1 \\ hk_2 \end{bmatrix} = \begin{bmatrix} zy^h_i \\ zy^h_i \end{bmatrix},$$

mit der Lösung

$$\begin{aligned} hk_1 &= y^h_i \frac{32z + 20z^2}{32 + 4z - 13z^2}, \\ hk_2 &= y^h_i \frac{32z + 28z^2}{32 + 4z - 13z^2}. \end{aligned}$$

Aus dem Butcher-Schema folgt $y^h_{i+1} = y^h_i + \frac{1}{2}(hk_1 + hk_2)$, so dass sich insgesamt die Evolution

$$y^h_{i+1} = y^h_i \cdot \underbrace{\frac{32 + 36z + 11z^2}{32 + 4z - 13z^2}}_{g(z):=} = y^h_i \cdot g(z).$$

ergibt. □

Definition 3.3.1. *Das ESV erfülle $\oplus$ für $y' = \lambda y$. Dann heißt die Teilmenge G der komplexen Ebene mit:*

$$G = \{z \ : \quad |g(z)| < 1\}$$

das **Gebiet der absoluten Stabilität** *des ESVs.*
Gilt

$$g(z) \ \rightarrow \ 0 \ \textit{für} \ \mathrm{Re}\ (z) \ \rightarrow \ -\infty$$

dann heisst das Verfahren **L-stabil** *. Das ESV heißt A-**stabil,** falls*

$$G \supset \{z \ : \ \mathrm{Re}\ z < 0\}$$

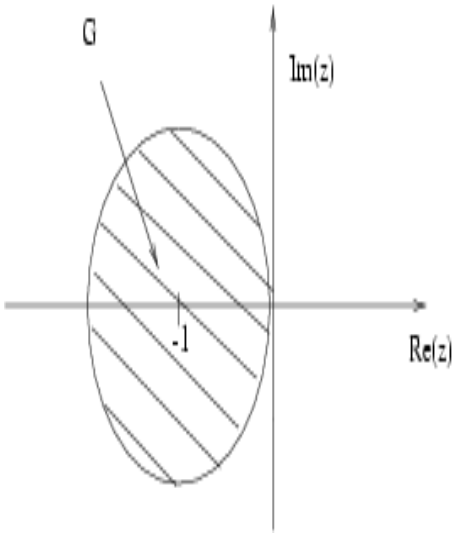
also, falls das Gebiet der absoluten Stabilität die linke Halbebene der komplexen Ebene umfaßt. Das Verfahren heisst A(α) stabil, falls das Gebiet der absoluten Stabilität eine Obermenge des Kegels

$$\{z \ : \ \arg(z) \ \in [-\pi, -\pi + \alpha] \cup [\pi - \alpha, \pi]\}$$

ist.

Man ist natürlich an Verfahren interessiert, die A-stabil sind oder zumindest ein grosses Gebiet absoluter Stabilität besitzen. Wir berechnen nun einige Gebiete der absoluten Stabilität:

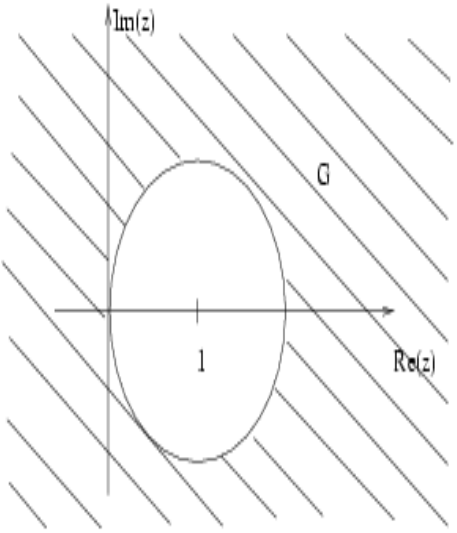
1. **Euler explizit:** $G = \{z \in \mathbb{C} | \quad |1 + z| < 1\}$



Das Verfahren ist nicht A-stabil

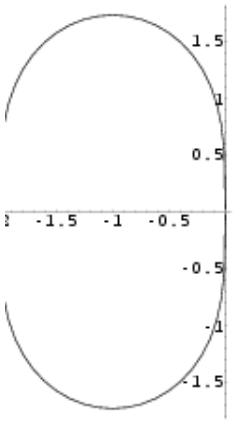
2. **Trapezregel:** Das Verfahren ist A-stabil, da $G = \{z | \text{ Re } z < 0\}$
   $|g(z)| = 1$, falls Re $z = 0$.
   (Die Trapezregel ist gut geeignet für Schwingungsgleichungen)
   Sie hat außerdem die Konsistenz- und Konvergenz-Ordnung 2.

3. **Euler implizit:** $G = \{z \in \mathbb{C} | \quad |1 - z| > 1\}$

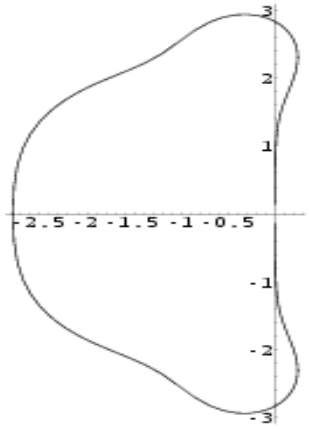


das Verfahren ist also A-stabil und L-stabil. Da das Gebiet der absoluten Stabilität auch die imaginäre Achse (ausser 0) enthält, wird die diskrete Lösung einer reinen Schwingung Re $\lambda = 0$ auch gedämpft, man sagt, das Verfahren erzeuge eine **künstliche Dissipation**.

4. Heun: Stabilität liegt vor im Inneren des Bereiches



5. explizites 4-stufiges Runge-Kutta-Verfahren der Ordnung 4:
   Stabilität im Inneren des Bereiches.



Bei den expliziten RK-Verfahren erhält man als Bedingung der absoluten Stabilität $|\lambda|h < C,\ C \in [1,3]$ i.w.

Die impliziten Verfahren werden verwendet für "steife" Gleichungen. Der Begriff der "Steifheit" ist eher qualitativer als quantitativer Art. Manchmal wird er über den Quotienten aus (absolut genommen) grösstem und kleinstem Eigenwert der Funktionalmatrix von f bezüglich y definiert. Dies ist aber nicht korrekt. Steifheit tritt auch schon bei skalaren Gleichungen auf. Wenn die Lösungsgesamtheit der Differentialgleichung Funktionen enthält, deren Ableitung sehr viel grösser ist als die der gesuchten Lösung des Anfanswertproblems, sodaß man bei der Integration eine viel kleinere Schrittweite benutzen muß als zur korrekten Wiedergabe dieser speziellen Lösung eigentlich erforderlich wäre, dann liegt Steifheit vor. Ein einfaches Beispiel mit dieser Eigenschaft stammt von Gear:

$$y' \ = \ \lambda(y - \exp(-t)) - \exp(-t), \quad y(t_0) \ = \exp(-t_0)$$

hat für jedes λ die Lösung $y(t) = \exp(-t)$. Die Lösungsmannigfaltigkeit hat die Form

$$(y(t_0) - \exp(-t_0)) \exp(\lambda(t - t_0)) + \exp(-t) \ ,$$

Funktionen, die sich für $\mathrm{Re}\ \lambda < 0$ mit wachsendem t alle schnell der Partikulärlösung $\exp(-t)$ annähern. Hier ist die Schrittweite z.B. beim expliziten klassischen Runge-Kutta-Verfahren der Ordnung 4 durch $2.8/|\lambda|$ beschränkt, also bei betragsgrossem λ sehr klein, während $y' \ = \ -y$ mit der Lösung $y(t) \ = \ y_0 \exp(-(t - t_0))$ schon mit der Schrittweite $h = 0.1$ mit hervorragender Genauigkeit integriert werden könnte.

Die folgende Abbildung zeigt die Lösungsschar und die Partikulärlösung dieser DGL für den harmlosen Fall $\lambda = -10$.

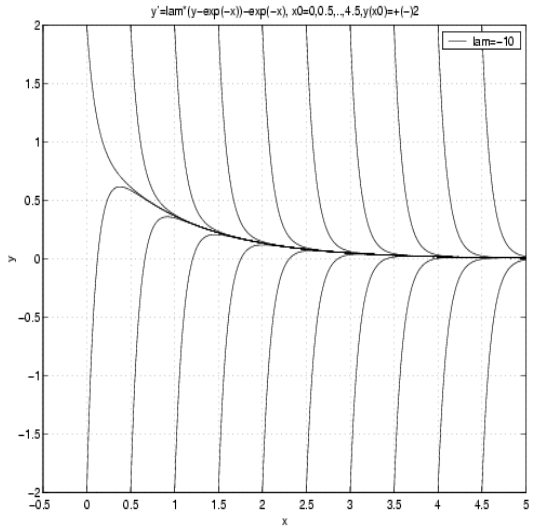


Abb. 3.3.1

Eine häufig benutzte Testgleichung ist auch die van der Pol Gleichung

$$y'' \;=\; \mu(1-y^2)y' - y$$

Hier zeigt die Lösung lokal sehr unterschiedliches Verhalten, in gewissen Bereichen variiert sie langsam, in anderen nimmt die Ableitung extreme Werte an:

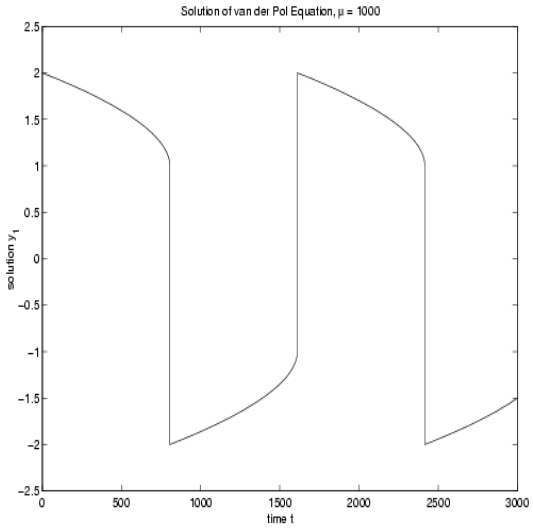


Abb 3.3.2 van der Pol : Lösung

3.4 Schrittweitensteuerung

Wir wenden uns nun der praktischen Anpassung der Gitterweite h an den Lösungsverlauf zu. A priori können wir ja die Schrittweite nicht auf Grund unserer theoretischen Aussagen wählen. Selbst wenn wir in der Lage wären, die relevanten Konstanten abzuschätzen, wäre das Ergebnis viel zu pessimistisch. Die Wahl der Schrittweite geschieht durch Kontrolle des lokalen Abschneidefehlers. Man beachte, daß wir im Folgenden unterstellen, daß die Lösung der Differentialgleichung mindestens $p+2$-mal stetig differenzierbar ist, wenn unser "Grundverfahren", das wir steuern wollen, die Ordnung p hat. In vielen Anwendungen gibt es zumindest lokal Stellen der Nichtglattheit der Lösung (Schalter einer äusseren Kraft, Umkehrung der Bewegungsrichtung bei Bewegung mit Reibung, etc). Unsere Methoden versagen an diesen Stellen. Praktisch bedeutet das, daß die berechnete Schrittweite h extrem klein wird. Alle praktisch eingesetzten Integratoren beruhen jedoch auf diesen Methoden. (In der Praxis "fressen" sie sich dann mit extrem kleinen Schrittweiten "fest"). Man muss deshalb in der Anwendung sorgfältig prüfen, ob eine solche Situation vorliegen kann und z.B. dort die Integration enden lassen und wieder neu starten. Es gibt spezielle software, die das ermöglicht.

Bemerkung 3.4.1. *Aus dem lokalen Fehler kann im allgemeinen nicht auf den globalen Fehler $y_i^h - y(t_i)$ geschlossen werden! Wenn aber die DGL überall "dämpft", dann ist der globale Fehler im Wesentlichen*

$$\int_{t_0}^{t} h\varrho(h, y(\tau), \tau)\, \mathrm{d}\tau \ ,$$

d.h. die Steuerung von ϱ beeinflußt direkt den globalen Fehler.

Das Problem liegt darin, h so zu wählen, daß ϱ klein wird und damit den globalen Fehler günstig beeinflußt, ohne h zu klein zu machen.

Steuerung von h bei vorgegebener Genauigkeitsforderung für den lokalen Diskretisierungsfehler: Die einfacheste Methode ist sicher der Vergleich von zwei Verfahren verschiedener Ordnung: (Φ_1: Ordnung p; Φ_2: Ordnung $p+1$ oder grösser.) pro Schritt. Wir befinden uns im Schritt i an der Stelle (t_i, y_i^h). $\tilde{y}$ sei Lösung der AWA

$$\begin{aligned} \tilde{y}' &= f(x, \tilde{y}) \\ \tilde{y}(t_i) &= y_i^h \end{aligned}$$

(d.h. wir betrachten die Lösung der DGL durch den neuen Anfangswert (t_i, y_i^h).) Dann

gilt nach Definition des lokalen Abschneidefehlers und den Verfahrensvorschriften

$$
\begin{array}{rcl}
\left\{\begin{array}{rcl} y_{i+1}^{[1]h} &=& y_i^h + h\cdot\Phi_1(t_i,y_i^h,h)\\ y_{i+1}^{[2]h} &=& y_i^h + h\cdot\Phi_2(t_i,y_i^h,h)\end{array}\right.\\
y_{i+1}^{[1]h} - \tilde{y}(t_{i+1}) = -h\cdot\varrho_1(h,y_i^h,t_i)\\
-y_{i+1}^{[2]h} + \tilde{y}(t_{i+1}) = \ h\cdot\varrho_2(h,y_i^h,t_i)\\
\hline
y_{i+1}^{[1]h} - \ y_{i+1}^{[2]h} \quad = -h\cdot\varrho_1(h,y_i^h,t_i) + \underbrace{h\cdot\underbrace{\varrho_2(h,y_i^h,t_i)}_{=\mathcal{O}(h^{p+1})}}_{=\mathcal{O}(h^{p+2})\ll h\varrho_1}
\end{array}
$$

falls h genügend klein

$$\Rightarrow \|y_{i+1}^{[1]h} - y_{i+1}^{[2]h}\| \approx \|h_i\cdot\varrho_1(h_i,y_i^h,t_i)\|$$

Von ϱ_1 wird folgende Struktur vorausgesetzt (dies ist in allen relevanten Fällen erfüllt)

$$\varrho_1(h,y,t) = h^p\cdot\Psi_1(t,y) + \mathcal{O}(h^{p+1}).$$

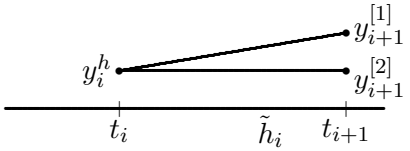
mit einer (unbekannten) Funktion $\Psi_1(t,y)$ Die Steuerung von h erfolgt nun aufgrund einer Forderung an ϱ_1 (lokaler Abschneidefehler) z.B.:

$$\oplus \qquad \|h\cdot\varrho_1(h_i,y_i^h,t_i)\| \le \frac{h\cdot\varepsilon}{t_{\text{Ende}}-t_0}\cdot\|y_i^h\|$$

$y_{i+1}^{[1]h}, y_{i+1}^{[2]h}$ werden nun versuchsweise mit einer "Vorschlagsschrittweite" $h := \tilde{h}_i$ berechnet.

Also erhalten wir

$$
\begin{array}{rcl}
\|\tilde{h}_i^{p+1}\cdot\Psi_1(y_i^h,t_i)\| &\approx& \|y_{i+1}^{[1]h} - y_{i+1}^{[2]h}\|\\
\Rightarrow \|\Psi_1(y_i^h,t_i)\| &\approx& \underbrace{\frac{1}{\tilde{h}_i^{p+1}}\cdot\|y_{i+1}^{[1]h} - y_{i+1}^{[2]h}\|}_{\text{berechenbar}}
\end{array}
$$



Mit der Forderung $\oplus$ ergibt sich:

$$\underbrace{\|h_i^{p+1} \cdot \underbrace{\Big(\frac{1}{\tilde{h}_i^{p+1}} \cdot \|y_{i+1}^{[1]h} - y_{i+1}^{[2]h}\|\Big)}_{=\Psi_1} \|}_{\approx h_i \cdot \varrho_1(h_i, y_i^h, t_i)} \quad \leq \quad \frac{h_i \cdot \varepsilon \cdot \|y_i^h\|}{t_{\text{Ende}} - t_0}$$

$$\Rightarrow h_i \quad \overset{!}{\leq} \quad \tilde{h}_i \cdot \underbrace{\sqrt[p]{\frac{\tilde{h}_i \cdot \varepsilon \cdot \|y_i^h\|}{(t_{\text{Ende}} - t_0) \cdot \|y_{i+1}^{[1]h} - y_{i+1}^{[2]h}\|}}}_{=c_i \text{ berechenbar}}$$

Die Grösse c_i ist also berechenbar. Die Steuerung lautet nun :

$$c_i \geq 1 \Rightarrow h_i \ = \tilde{h}_i \ ,$$

die Schrittweite war klein genug, akzeptiere also diesen Schritt und beginne den nächsten mit einer eventuell vorsichtig vergrösserten Schrittweite (es wäre ja $c_i \gg 1$ denkbar)

$$\tilde{h}_{i+1} \ = \ \max\{1, \min\{2, 0.9c_i\}\} h_i \ .$$

Andernfalls ist

$$c_i \ < \ 1 \ \Rightarrow \ \tilde{h}_i \overset{def}{=} 0.9 c_i \tilde{h}_i$$

also wiederhole den Schritt. Der Faktor 0.9 ist hierbei ein praxistypischer "Sicherheitsfaktor".

Geeignete Paare solcher Verfahren sind z.B. die sogenannten "eingebettenen" Runge-Kutta-Verfahren, bei denen mit dem gleichen Satz von k_i-Werten durch verschiedene Wahl der äusseren Quadraturgewichte γ_j Verfahren verschiedener Ordnung erhältlich sind. Es gibt viele solcher Formelpaare in der Spezialliteratur. z.B.: Bogacki, P.; Shampine, L.F.: An efficient Runge-Kutta $(4, 5)$ pair. Journal on Comput. Math. Appl. 32, No.6, 15-28 (1996).

Beispiel 3.4.1. *Es soll die Verfahrenskombination Euler und Heun zur Schrittweitensteuerung des Heun-Verfahrens auf die DGL $y' = -y^2$ mit Anfangswert $y(0) = 1$ angewandt werden. Hier zur Erinnerung die entsprechenden Butcher-Arrays dazu:*

0	0	0
1	1	0
γ_i	1	
$\tilde{\gamma}_i$	$\frac{1}{2}$	$\frac{1}{2}$

Wir testen, ob die Vorschlagsschrittweite $h = 0.1$ im Punkt $t = 0$ akzeptiert wird, wenn eine Genauigkeit von $\varepsilon = 0.1$ auf dem Intervall $t \in [0, 1]$ gefordert ist. Für $h = 0.1$,

$f = -y^2$ und $y(0) = 1$ ergeben sich die Werte:

$$\begin{array}{rcl} k_1 &=& f(t,y) \\ k_2 &=& f(t+h, y+hk_1) \\ \Phi_1 &=& k_1 \\ \Phi_2 &=& \frac{1}{2}(k_1+k_2) \end{array} \quad \Rightarrow \quad \left\{ \begin{array}{rcl} k_1 &=& f(0,1) = -1 \\ k_2 &=& f(0+0.1, 1+0.1(-1)) = -0.81 \\ \Phi_1 &=& -1 \\ \Phi_2 &=& \frac{1}{2}(-1-0.81) = -0.905. \end{array} \right.$$

Die Näherungslösungen sind dann

$$\begin{array}{rcl} y_1 &=& 1 + 0.1(-1) = 0.9 \\ y_2 &=& 1 + 0.1(-0.905) = 0.9095. \end{array}$$

Wir haben $p = 1$, $y_0^h = 1$ und somit

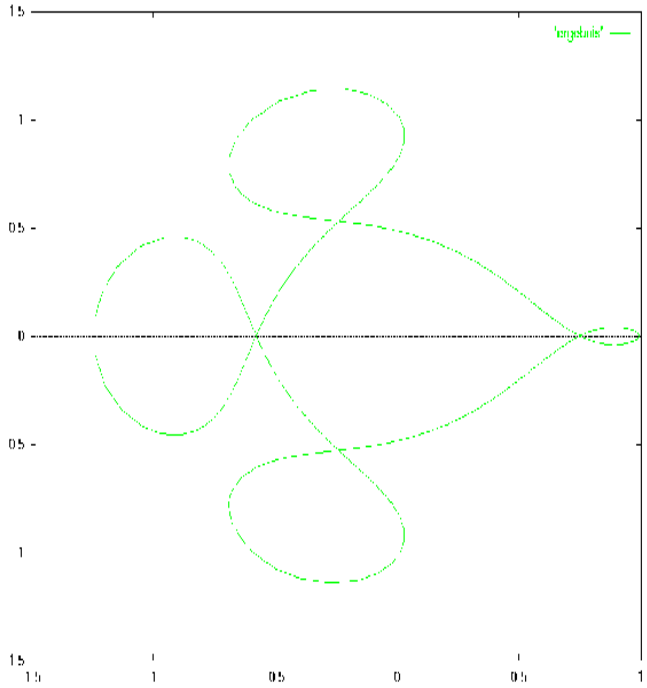
$$c_0 \;=\; \tfrac{0.1 \cdot 0.1 \cdot 1}{1 \cdot 0.0095} = 1.05263$$

der Schritt wird also akzeptiert.

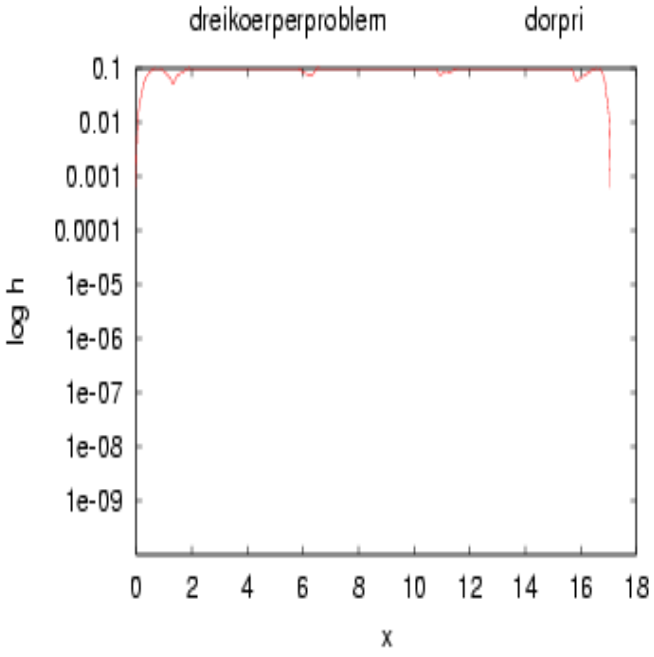
Zur Schätzung des **globalen Fehlers** kennt man verschiedene Methoden. Eine davon ist die Durchführung des eben beschriebenen Verfahrens unabhängig voneinander auf zwei Gittern, von denen das eine immer durch Schrittweitenhalbierung aus dem anderen hervorgeht, d.h. man erzeugt eine zweite Lösung $y_{2i}^{h/2}$ auf dem feineren Gitter gleichzeitig mit y_i^h und akzeptiert die Schritte nur, wenn beide Teilintegrationen unabhängig voneinander akzeptiert würden. Dann nimmt man

$$y_i^h - y_{2i}^{h/2} \text{ als Schätzung für den globalen Diskretisierungsfehler.}$$

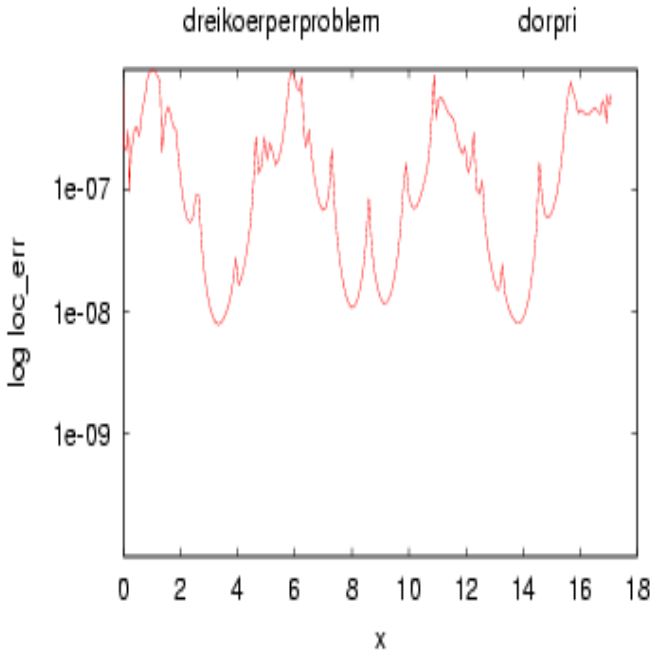
Das Resultat einer solchen Vorgehensweise bei einer speziellen Problemstellung, dem sogenannten eingeschränkten Dreikörperproblem (hier das System Erde-Mond-Raumkapsel) zeigen die nachfolgenden Abbildungen. Hier wurde für den lokalen Abschneidefehler eine Grenze von 10^{-6} angegeben, und die Abbildung zeigt, daß dies auch tatsächlich eingehalten wird. Die Schrittweite variiert stark, vor allem im Anfangsbereich ist sie zunächst viel kleiner. Der globale Fehler aber wächst weit über diesen Wert hinaus. Hier erweist sich die Schätzung des globalen Fehlers als korrekt. Die Verfahrenskombination war hier ein siebenstufiges Runge-Kutta-Verfahren mit den Ordnungen 4 und 5 von Dormand und Prince.



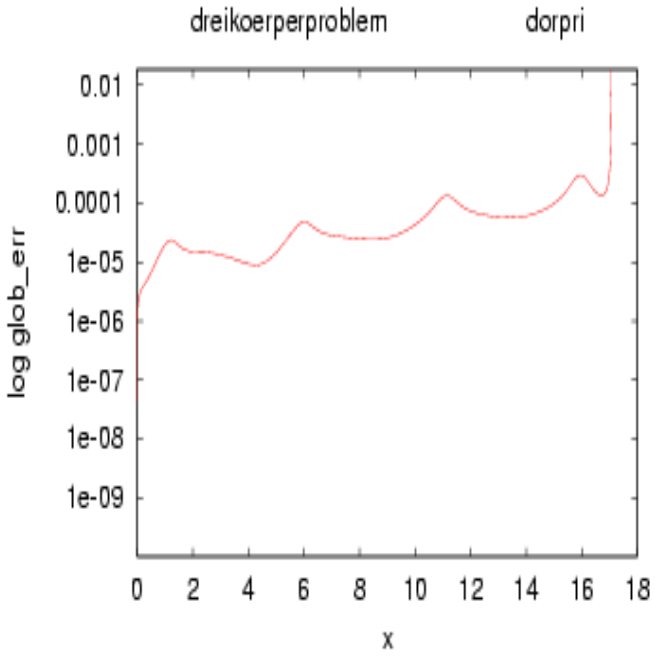
Phasendiagramm der Lösung $y_1(t),\ y_2(t)$



Schrittweiten über Zeitachse



lokaler Diskretisierungsfehler über Zeitachse



globaler Disketisierungsfehler über Zeitachse

NUMAWWW

Die zweite Methode ist die sogenannte **globale Extrapolation**, die allerdings nur beim Integrieren mit fester Schrittweite zulässig ist:

Satz 3.4.1. Φ *sei hinreichend oft differenzierbar bzgl.* h, y *und* t. $\varrho(h, y, t)$ *habe die Darstellung*

$$h \cdot \varrho(h, y, t) = h^{p+1} \cdot \Psi_1(t, y) + \mathcal{O}(h^{p+2})$$

$\{y_i^h\}$ *werde mit der konstanten Schrittweite* h *berechnet.*
Dann gibt es eine Darstellung für y_i^h *der Form:*

$$\begin{aligned} y_N^h &= y(t) + h^p \cdot e_1(t) + \mathcal{O}(h^{p+1}) \quad \textit{mit} \\ t &= t_0 + N \cdot h \quad \textit{fest} \ , \quad e_1(t) \ \textit{geeignete Funktion} \end{aligned}$$

Idee: Berechne y_N^h mit Schrittweite h und $y_{2N}^{h/2}$ mit Schrittweite $h/2$ (dies ergibt den gleichen Endpunkt t_N) und damit

$$\begin{array}{r} y_N^h = y(t_N) + h^p \cdot e_1(t_N) + \mathcal{O}(h^{p+1}) \\ -y_{2N}^{h/2} = -y(t_N) - \frac{1}{2^p} \cdot h^p \cdot e_1(t_N) + \mathcal{O}(h^{p+1}) \\ \hline h^p \cdot e_1(t_N) \approx \frac{1}{1-\frac{1}{2^p}} \cdot (y_N^h - y_{2N}^{h/2}) \approx y_N^h - y(t_N) \end{array}$$

also eine Schätzung des globalen Fehlers in y_N^h.

Man kann also a posteriori den globalen Fehler schätzen, aber nicht a priori steuern (weil man die Dynamik der Differentialgleichung ausser im linearen Fall nicht voraussehen kann).

3.5 Mehrschrittverfahren:

Mehrschrittverfahren (MSV) unterscheiden sich von Einschrittverfahren (ESV) darin, daß zur Berechnung eines neuen Näherungspunktes nicht einer, sondern k schon bekannte Näherungswerte benötigt werden, d.h. die Schrittfunktion Φ ist von k Näherungswerten abhängig:

$$y_{i+1}^h = y_i^h + h \cdot \Phi(t_i, y_i^h, \ldots, y_{i-k+1}^h, h)$$

Die Theorie ist komplizierter und für die asymptotische Stabilität (Stabilität für $h \to 0$) müssen zusätzliche Forderungen gestellt werden, doch man erreicht eine Erhöhung der Ordnung der Verfahren ohne Erhöhung des Aufwandes an Funktionsauswertungen von f. Mit 2 Auswertungen von f kann man brauchbare Verfahren sehr hoher Ordnung (auf Kosten einer Reduktion des Gebietes der absoluten Stabilität) erreichen. (siehe Spezialliteratur). Diese Verfahren haben sich aber nur da bewährt, wo die Lösung der AWA nicht extrem schnell variiert. Über Details informiere man sich in der Spezialliteratur.

3.6 Eigenwertabschätzungen

Kehren wir zurück zu unserer Modellgleichung (*). Offenbar spielen die Eigenwerte von A eine wesentliche Rolle und es ist deshalb nützlich, einfache Lokalisierungssätze für die Eigenwerte zu kennen. Das elementarste in dieser Hinsicht ist wohl

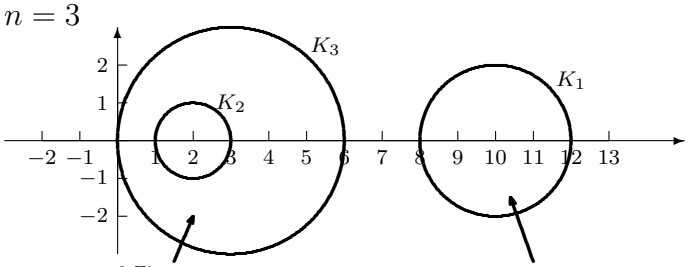
Satz 3.6.1. (Kreisesatz von Gerschgorin) *A sei eine beliebige reelle oder komplexe $n \times n$-Matrix. Dann liegt jeder Eigenwert λ von A in der Vereinigungsmenge der n Kreisscheiben*

$$K_i := \{z \in \mathbb{C} \ : \ |z - \underbrace{a_{ii}}_{\text{Mittelpunkt}} | \leq \underbrace{\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|}_{\text{Radius}} \} .$$

Sind von den n Kreisscheiben s von den übrigen $n-s$ isoliert, dann enthält deren Vereinigung genau s Eigenwerte. Sind also alle n Kreisscheiben isoliert, dann hat A n verschiedene Eigenwerte und jeder von ihnen liegt in genau einem K_i.

Wir betrachten mehrere **Beispiele**:

- $A = \begin{pmatrix} 10 & 1 & -1 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix}$, $n = 3$

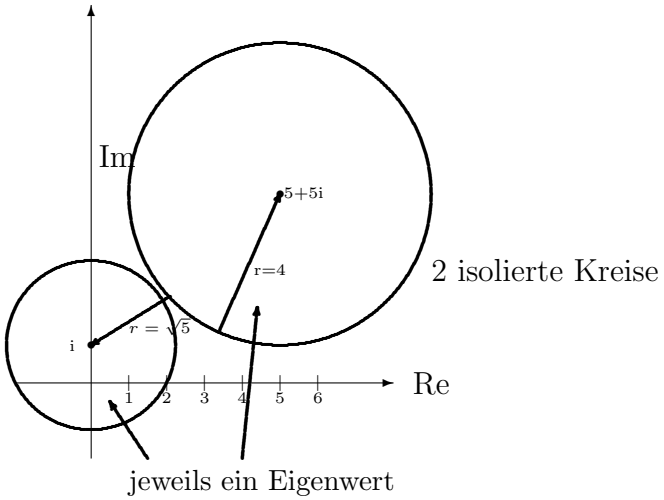


- $A = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}$

Alle Kreise überlagern sich. Und da A reell und symmetrisch ist, sind alle Eigenwerte reell. Folglich gilt $0 \leq \lambda_i \leq 4$ für alle i.

- Man kann Satz 3.6.1 genauso auf A^T anwenden, weil die Eigenwerte von A und A^T die gleichen sind. Dies ergibt weitere Einschließungen für die Eigenwerte.

- $A = \begin{pmatrix} i & 1+2i \\ 4 & 5+5i \end{pmatrix}$



3.7 Zusammenfassung

Anwendung von Formeln der numerischen Differentiation auf eine DGL bzw. der numerischen Quadratur auf die zugehörige Volterra-Integralgleichung liefern Näherungsformeln zur schrittweisen Integration von Anfangswertproblemen gewöhnlicher Differentialgleichungen. Wie bei der Quadratur sind im Prinzip Verfahren beliebig hoher Ordnung konstruierbar. Eine höhere Ordnung erfordert bei den Einschrittverfahren auch stets eine erhöhte Anzahl von Auswertungen der rechten Seite der Differentialgleichung. In der Praxis sind nur Verfahren mit massvoll grosser Ordnung, etwa 4 bis 8, im Einsatz. Explizite Verfahren sind sehr einfach handhabbar, für die sogenannten steifen Gleichungen jedoch ungeeignet, weil dann die verwendbaren Schrittweiten zu klein werden. Mit impliziten Verfahren kann man Eigenschaften wie A- und L-Stabilität erreichen, muss aber als Preis die Lösung von in der Regel nichtlinearen Gleichungssystemen pro Integrationsschritt bezahlen. Methoden dazu werden wir später besprechen. In der Praxis wird die Integration stets in Verbindung mit einer Schrittweitensteuerung durchgeführt. Diese beruht auf einer Schätzung des lokalen Diskretisierungsfehlers, die man z.B. aus dem Vergleich der Resultate von Verfahren verschiedener Ordnung erhält. Der globale Fehler lässt sich jedoch nicht a priori steuern, man kann ihn jedoch kontrollieren, z.B. durch die mehrfache Integration auf verfeinerten kohärenten Gittern.

Kapitel 4

Lösung linearer Gleichungssysteme: Direkte Methoden

4.1 Problemstellung und Einführung

Bemerkung 4.1.1. *In den folgenden Abschnitten betrachten wir die Lösung linearer und nichtlinearer Gleichungssysteme, bei denen die gesuchte Unbekannte ein Vektor ist. Zur Verdeutlichung benutzen wir deshalb für Vektoren eine Notation wie*

$$\vec{x}, \quad \vec{b}, \quad \ldots$$

Wir beschäftigen uns in diesem Kapitel mit der
Aufgabenstellung: Gegeben ist eine $n \times n-$Matrix A und eine Inhomogenität $\vec{b}$. Zu lösen ist

$$A\vec{x} = \vec{b} \, .$$

Gesucht ist also $\vec{x} \in \mathbb{R}^n$. Wir werden immer voraussetzen, daß

$$det(A) \; \neq \; 0 \, .$$

Die Gleichung ist dann eindeutig lösbar.

Der Fall einer singulären Koeffizientenmatrix A ist natürlich auch von (theoretischem) Interesse. Beim Auftreten von Rundungsfehlern in der Rechnung kann jedoch die Singularität bzw. Nichtsingularität einer Matrix nicht mehr in allen Fällen erkannt werden. Dies hängt ab von der Relation zwischen der Rechengenauigkeit und der später in diesem Kapitel definierten "Konditionszahl" der Matrix. Deshalb lassen wir diesen Fall beiseite. Auch die allgemeine Aufgabe mit mehreren rechten Seiten

$$AX = B, \qquad A \in \mathbb{K}^{n \times n}, \qquad X \in \mathbb{K}^{n \times p}, \qquad B \in \mathbb{K}^{n \times p},$$

insbesondere die Aufgabe der Matrixinversion

$$AX = I$$

kann hier eingeordnet werden. Setze dazu $X = (\vec{x}_1, \ldots, \vec{x}_p), \quad B = (\vec{b}_1, \ldots, \vec{b}_p)$:

$$AX = B \qquad \Leftrightarrow \qquad A\vec{x}_i = \vec{b}_i \quad i = 1, ..., p$$

Diese Aufgabe tritt in der Praxis üblicherweise als Teilaufgabe bei der Lösung einer Vielzahl von Problemen auf, z.B. bei der Lösung von Rand- und Randanfangswertaufgaben gewöhnlicher und partieller Differentialgleichungen (Berechnung der Deformation und der Schwingungen von Bauteilen), bei der Schaltkreissimulation, in der chemischen Reaktionskinetik, in der Bildverarbeitung, in der Optimierung etc. Man hat geschätzt, daß etwa 75% der Rechenzeit, die für wissenschaftlich-technische Berechnungen überhaupt aufgewendet wird, auf Kosten der Lösung dieser elementar erscheinenden Aufgabe geht. Ihre zuverlässige und effiziente Behandlung ist daher von grösster Wichtigkeit. n kann in der Praxis durchaus 10^6 oder mehr betragen. "Direkte Methoden" bedeutet in diesem Zusammenhang, daß man einen Lösungsweg wählt, der bei exakter reeller oder komplexer Rechnung die exakte Lösung in einer endlichen Anzahl von elementaren Rechenoperationen $+, -, *, /$ liefert.

Formal erhalten wir $\vec{x} = A^{-1}\vec{b}$. Dies suggeriert als Lösungsweg die explixite Berechnung der inversen Matrix und dann die Matrix-Vektor-Multiplikation mit $\vec{b}$.

Diese formale Lösung ist in der Praxis in der Regel nicht empfehlenswert, ja oft unmöglich, da

1. der Aufwand rechnerisch ungünstig ist,
2. der Speicheraufwand u.U. untragbar ist (in der Praxis ist A oft "dünn besetzt", A^{-1} dagegen voll) und
3. der Rundungsfehlereinfluß besonders ungünstig ist, wenn man A^{-1} explizit berechnet.

Wir wählen einen anderen Lösungsweg. Unser Ziel wird es sein, die Aufgabe auf zwei Teil-Aufgaben mit sogenannten Dreiecksmatrizen zurückzuführen durch eine Faktorisierung

$$PA \;=\; LR$$

mit einer Permutationsmatrix P, einer unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R. Dann wird

$$A^{-1} \;=\; R^{-1}L^{-1}P$$

und

$$A\vec{x} \;=\; \vec{b} \;\Leftrightarrow\; L\vec{z} \;=\; P\vec{b}\,,\; R\vec{x} \;=\; \vec{z}\,.$$

Wir beschäftigen uns daher zunächst mit der entsprechenden Aufgabe im Falle von Dreiecksmatrizen.

4.2 Systeme mit Dreiecksmatrix

Bei solchen Systemen hat man nacheinander n lineare Gleichungen in einer Unbekannten zu lösen, was unmittelbar möglich ist: **Beispiel:**

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 3 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 6 \end{pmatrix}$$

$$\left.\begin{array}{rl} x_1 = 1 & \\ -2x_1 + 2x_2 = 2 & \\ x_1 - 2x_2 + 3x_3 = 6 & \end{array}\right\} \begin{array}{l} \Rightarrow \; x_1 = 1 \\ \Rightarrow \; x_2 = 2 \\ \Rightarrow \; x_3 = 3 \end{array} \Rightarrow \vec{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Ein System der obigen Form mit einer Dreiecksmatrix A nennt man auch ein gestaffeltes System. Für

$$A = L = \text{◺} \quad \text{oder} \quad A = R = \text{◹}$$

erhalten wir

$$\det L = \prod_{i=1}^{n} l_{ii} \quad \text{bzw.} \quad \det R = \prod_{i=1}^{n} r_{ii} \; ,$$

die Invertierbarkit dieser Matrizen ist also trivial überprüfbar, im Gegensatz zum Fall einer allgemeinen quadratischen Matrix.

Die Komponenten der Lösung von

$$L\vec{y} = \vec{b} \quad \text{bzw.} \quad R\vec{z} = \vec{c}$$

berechnen sich als

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}} \quad (i = 1, \ldots, n) \qquad \text{bzw.} \qquad z_{n-i} = \frac{c_{n-i} - \sum_{j=n-i+1}^{n} r_{n-i,j} z_j}{r_{n-i,n-i}} \quad (i = 0, \ldots, n-1)$$

Der Aufwand hierfür ist $\mathcal{O}(n^2)$ an Additionen und Multiplikationen, falls nicht noch zusätzlich spezielle Besetztheitsstrukturen vorliegen (etwa Bandstruktur, vergl. hinten).

Bemerkung 4.2.1. *Wenn ausnahmsweise tatsächlich die explizite Inverse benötigt wird, kann man zur Berechnung der Inversen von Dreiecksmatrizen bzw. Block-Dreiecksmatrizen die spezielle Struktur ebenfalls gewinnbringend ausnutzen, denn die ersten i Spalten der Inversen einer oberen Dreiecksmatrix hängen nur von den ersten i Spalten der Ausgangsmatrix ab (und entsprechend bei den Zeilen für eine untere Dreiecksmatrix). Dies drückt sich aus in den Formeln:*

$$R = \begin{array}{|c|c|}\hline R_{11} & \vec{r} \\ \hline 0\ldots 0 & \varrho \\ \hline \end{array} \quad \textit{bzw.} \quad R = \begin{array}{|c|c|}\hline R_{11} & R_{12} \\ \hline 0 & R_{22} \\ \hline \end{array}$$

$$\Rightarrow$$

$$R^{-1} = \left(\begin{array}{c|c} R_{11}^{-1} & -R_{11}^{-1}\cdot\vec{r}\cdot\frac{1}{\varrho} \\ \hline 0 & \frac{1}{\varrho} \end{array}\right) \quad \textit{bzw.} \quad R^{-1} = \left(\begin{array}{c|c} R_{11}^{-1} & -R_{11}^{-1}\cdot R_{12}\cdot R_{22}^{-1} \\ \hline 0 & R_{22}^{-1} \end{array}\right)$$

Zum Beweis benutzen wir die Tatsache daß gilt:

Falls B: $BA = I_n = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$ erfüllt, dann gilt $B = A^{-1}$.

Z.B.

$$R^{-1}\cdot R = \left(\begin{array}{c|c} R_{11}^{-1} & -R_{11}^{-1}\cdot\vec{r}\cdot\frac{1}{\varrho} \\ \hline 0 & \frac{1}{\varrho} \end{array}\right)\left(\begin{array}{c|c} R_{11} & \vec{r} \\ \hline 0 & \varrho \end{array}\right) = \left(\begin{array}{c|c} R_{11}^{-1}\cdot R_{11} & R_{11}^{-1}\cdot\vec{r} - R_{11}^{-1}\cdot\vec{r}\cdot\frac{1}{\varrho}\cdot\varrho \\ \hline 0 & \frac{\varrho}{\varrho} \end{array}\right)$$

$$= \left(\begin{array}{c|c} I_{n-1} & 0 \\ \hline 0 & 1 \end{array}\right) = I_n$$

Wir können also bei der Berechnung der Inversen R^{-1} einer Dreiecksmatrix die Ausgangsmatrix R sukzessive mit dem Elementen von R^{-1} spaltenweise überschreiben, sinnvollerweise von hinten nach vorne.

Für eine untere Dreiecksmatrix L geht man analog vor. Man stelle sich alles transponiert vor. □

4.3 Dreieckszerlegung einer Matrix Gauss-Algorithmus

Wir verfolgen nun das Ziel, eine allgemeine Matrix in ein Produkt von Dreiecksmatrizen zu zerlegen. Dies ist aber in der einfachen Form

$$A \;=\; LR \text{ mit invertierbarem } L \text{ und } R \text{ für invertierbares } A$$

nicht immer möglich, wie das folgende Beispiel zeigt:

Beispiel 4.3.1. $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$

$$\Rightarrow\ 0 = l_{11} \cdot r_{11} + 0 \cdot 0$$
$$\Rightarrow l_{11} = 0 \quad \textit{oder} \quad r_{11} = 0$$

Nun folgt

$\det L = l_{11} \cdot l_{22} - l_{21} \cdot 0 = l_{11} \cdot l_{22} = 0$ *für* $l_{11} = 0$

Dies ist ein Widerspruch zur Voraussetzung $\det L \neq 0$. *(Für* $r_{11} = 0$ *geht man analog vor.)* □

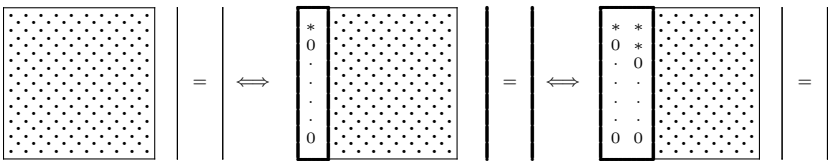
Glücklicherweise gibt es folgenden Ausweg:

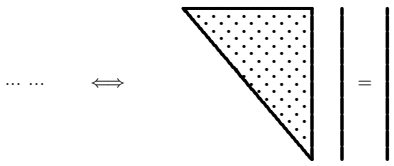
Wenn $\det A \neq 0$, dann existiert immer eine "Zeilentauschmatrix" P (Permutationsmatrix) mit $PA = LR$.

Wie wird nun diese Zerlegung praktisch berechnet?

Sie wird vom Gauß-Algorithmus, dem schon von der Schule bekannten "Einsetzverfahren", mitgeliefert!

Die Idee des Gauß'schen Eliminationsverfahrens besteht darin, ein beliebiges Gleichungssystem mit regulärer $n \times n$–Koeffizientenmatrix in (höchstens) $n-1$ Äquivalenz-Transformationsschritten in ein System mit oberer Dreiecksmatrix zu überführen. Symbolisch





Als Äquivalenztransformationen dienen im i^{ten} Schritt, $i = 1, ..., n-1$:

a) Falls nötig,Vertauschung der Zeile i mit einer der Zeilen $i+1, ..., n$ des Systems

b) (Falls erwünscht, Vertauschung der Spalte i mit einer der Spalten $i+1, ..., n$ des Systems. Dies dient der weiteren Dämpfung der Rundungsfehlereinflüsse. Dies wird aber nur in seltenen Fällen benutzt)

c) Subtraktion von geeigneten Vielfachen der i^{ten} Zeile von den Zeilen $i+1, ..., n$.

Streng formal müsste man diese Systeme also bezeichen als

$$A^{(i)}\vec{x}^{(i)} \;=\; \vec{b}^{(i)}\;, \quad i \;=\; 1, \ldots, n$$

mit

$$A^{(1)} \;=\; A\;,\; \vec{b}^{(1)} \;=\; \vec{b} \quad \text{Ausgangsdaten}$$

und $\vec{x}^{(i)}$ als permutiertem $\vec{x}-$Vektor. Wir verzichten hier darauf und schreiben das System, wie in der Praxis üblich, in ein Schema, das um die Zeilen- und Spaltennummern erweitert wird. Auf die Positionen der erzeugten Nullen schreiben wir die Vielfachen, die zu ihrer Erzeugung notwendig waren. Bei einer Vertauschung werden dann vollständige Zeilen bzw. Spalten vertauscht. Man kann dann an den Vektoren der vertauschten Zeilen- und Spaltennummern die Originalposition und die angewendete Vertauschungsmatrix ablesen. Das Ausgangsschema hat also jetzt die Form
$a_{ij}^{(1)} := a_{ij} \qquad i,j = 1, ..., n \qquad b_i^{(1)} := b_i \qquad i = 1, ..., n$

	1	$\cdots$	$\cdots$	n	
1	$a_{11}^{(1)}$	$\cdots$	$\cdots$	$a_{1n}^{(1)}$	$b_1^{(1)}$
$\vdots$	$\vdots$			$\vdots$	
$\vdots$	$\vdots$			$\vdots$	
$\vdots$	$\vdots$			$\vdots$	
n	$a_{n1}^{(1)}$	$\cdots$	$\cdots$	$a_{nn}^{(1)}$	$b_n^{(1)}$

Im $i-$ten Schritt wollen wir Nullen auf den Positionen $(i+1, i), \ldots, (n, i)$ erzeugen, während die Elemente $(i, i), \ldots, (i, n)$ zu einer Zeile der Matrix R werden. Dazu muss also das Element auf der Position (i, i) ungleich null sein. Man bezeichnet es als "Pivot"-Element (Pivot=Flügelmann, in der Technik: Drehzapfen). Formal genügt hier die Forderung $\neq 0$, aber um den Rundungsfehlereinfluss klein zu halten, muss man hier sehr sorgfältig vorgehen. Dies ist der Punkt, wo die Vertauschungen der Zeilen und Spalten zum Tragen kommen. Wir haben bisher stillschweigend angenommen, daß bei nichtsingulärer Koeffizientenmatrix A die Auswahl von Zeilenvertauschungen (und Spaltenvertauschungen) es stets erlaubt, ein Pivotelement ungleich null zu finden. Dies ist tatsächlich der Fall .

Bemerkung 4.3.1. *Die* **Auswahlregel für die Vertauschungen** *heißt* **Pivotstrategie**. *Folgende Pivotstrategien sind üblich ($\tilde{a}_{i,j}$ bezeichnet die Elemente der $i-$ten Matrix nach den Vertauschungen):*

a) **"Spaltenpivotwahl"** *:* $|\tilde{a}_{k,k}^{(k)}| \stackrel{!}{=} \max\limits_{i \geq k} |a_{i,k}^{(k)}|$

(Zeilenvertauschung; keine Spaltenvertauschungen)
Pivot = ein betragsgrösstes Element der Restspalte

b) "Restmatrix–Pivotwahl": $|\tilde{a}_{k,k}^{(k)}| \stackrel{!}{=} \max_{i,j \geq k} |a_{i,j}^{(k)}|$

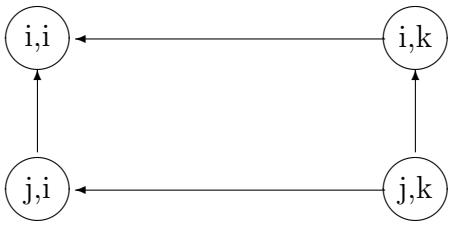
(Zeilen– und Spaltenvertauschungen)
Pivot = ein betragsgrösstes Element der Restmatrix .

Man beachte, daß in den Fällen a) und b) die Multiplikatoren $\tilde{a}_{j,i}^{(i)}/\tilde{a}_{i,i}^{(i)}$ betragsmäßig ≤ 1 sind. Dies bewirkt ein günstiges Rundungsverhalten. Den völligen Verzicht auf Vertauschungen bezeichnet man als "natürliche" Pivotwahl. Dies ist nur bei speziellen Matrizen A durchführbar und gefahrlos. (bzgl. des Rundungsfehlerverhaltens) □

Hat man den Pivot auf der Position (j,k) gewählt (wie bereits gesagt benutzt man meist nur Zeilentausch, dann ist $k = i$) mit $j \geq i$ und $k \geq i$, dann vertauscht man Zeile i mit Zeile j und Spalte i mit Spalte k. Es ist wichtig, die Vertauschung nur in dieser Form, also als "Pärchentausch" vorzunehmen. Nun ist man in der Position, die gewünschten Nullen zu erzeugen. Die dazu notwendigen Multiplikatoren entstehen aus den Quotienten der Koeffizienten auf den Positionen (j,i) und (i,i). Die Umrechnung auf das nächste Teilsystem betrifft dann nur die sogenannte "Restmatrix" (das sind die Elemente mit Index $\geq i+1$). Man merkt sich diese Umrechnung leicht als sogenannte "Rechteckregel"

$$(j,k)_{\text{neu}} \;=\; (j,k)_{\text{alt}} - \frac{(j,i)_{\text{alt}}}{(i,i)_{\text{alt}}}(i,k)_{\text{alt}} \qquad \begin{matrix} i+1 \leq j \leq n \\ i+1 \leq k \leq n \end{matrix}$$

$(j,k)_{\text{neu}}$ = neues Element
$(j,k)_{\text{alt}}$ =altes Element im Restsystem
Quotient =Multiplikator =zugeh. Element Pivotspalte / Pivot
$(i,k)_{\text{alt}}$ = zugeh. Element Pivotzeile



Nach $n-1$ Schritten hat man dann die Dreiecksform erreicht, die gemäss dem vorangegangenen Abschnitt behandelt wird. Man muss dann noch bedenken, daß bei angewendetem Spaltentausch der Lösungsvektor $\vec{x}^{(n)} = \vec{y}$ auch vertauscht ist. Die "richtige" Position der Lösungskomponente liest man aus den vertauschten Spaltennummern ab. Sind diese $\sigma_i,\ i = 1,\ldots,n$ dann gilt

$$x_{\sigma_i} \;=\; y_i$$

wo x_j die Komponenten von $\vec{x}$ und y_j die von $\vec{y}$ sind.

Beispiele

Mit Spaltenpivotsuche :

$$\left[\begin{array}{cccc|c} & 1 & 2 & 3 & \\ 1 & 3 & 4 & 5 & 26 \\ 2 & -3 & 5 & 1 & 10 \\ 3 & 6 & 5 & 18 & 70 \end{array}\right]$$

Zeilentausch 3 gegen 1:

$$\left[\begin{array}{cccc|c} & 1 & 2 & 3 & \\ 3 & 6 & 5 & 18 & 70 \\ 2 & -3 & 5 & 1 & 10 \\ 1 & 3 & 4 & 5 & 26 \end{array}\right]$$

Elimination:

$$\left[\begin{array}{cccc|c} & 1 & 2 & 3 & \\ 3 & 6 & 5 & 18 & 70 \\ 2 & -\mathbf{\frac{1}{2}} & 5-(-3)\cdot 5/6=\frac{15}{2} & 1-(-3)\cdot 18/6=10 & 10-(-3)\cdot 70/6=45 \\ 1 & \mathbf{\frac{1}{2}} & 4-3\cdot 5/6=\frac{3}{2} & 5-3\cdot 18/6=-4 & 26-3\cdot 70/6=-9 \end{array}\right]$$

Zweiter Schritt (keine Vertauschung notwendig)

$$\left[\begin{array}{cccc|c} & 1 & 2 & 3 & \\ 3 & 6 & 5 & 18 & 70 \\ 2 & -\mathbf{\frac{1}{2}} & \frac{15}{2} & 10 & 45 \\ 1 & \mathbf{\frac{1}{2}} & \mathbf{\frac{1}{5}} & -4-10\cdot\frac{3}{2}/\frac{15}{2}=-6 & -9-45\cdot\frac{3}{2}/\frac{15}{2}=-18 \end{array}\right]$$

Und daher

$$\begin{aligned} x_3 &= 3 \\ x_2 &= (45-10\cdot 3)/\tfrac{15}{2} = 2 \\ x_1 &= (70-5\cdot 2-18\cdot 3)/6 = 1\ . \end{aligned}$$

Mit Restmatrixpivotsuche

$$\left[\begin{array}{cccc|c} & 1 & 2 & 3 & \\ 1 & 0 & 1 & -3 & 3 \\ 2 & 1 & 1 & 3 & -4 \\ 3 & 1 & -1 & 3 & 5 \end{array}\right]$$

Als Pivotposition wählen wir (2,3). Das vertauschte System ist

$$\left[\begin{array}{cccc|c} & 3 & 2 & 1 & \\ 2 & 3 & 1 & 1 & -4 \\ 1 & -3 & 1 & 0 & 3 \\ 3 & 3 & -1 & 1 & 5 \end{array}\right]$$

Nach dem ersten Eliminationsschritt haben wir

$$\left[\begin{array}{cccc|c} & 3 & 2 & 1 & \\ 2 & 3 & 1 & 1 & -4 \\ 1 & \mathbf{-1} & 2 & 1 & -1 \\ 3 & \mathbf{1} & -2 & 0 & 9 \end{array}\right]$$

Nur zur Illustration vertauschen wir noch Zeile 2 und 3:

$$\left[\begin{array}{cccc|c} & 3 & 2 & 1 & \\ 2 & 3 & 1 & 1 & -4 \\ 3 & \mathbf{1} & -2 & 0 & 9 \\ 1 & \mathbf{-1} & 2 & 1 & -1 \end{array}\right]$$

und der zweite Eliminationsschritt ergibt

$$\left[\begin{array}{cccc|c} & 3 & 2 & 1 & \\ 2 & 3 & 1 & 1 & -4 \\ 3 & \mathbf{1} & -2 & 0 & 9 \\ 1 & \mathbf{-1} & \mathbf{-1} & 1 & 8 \end{array}\right]$$

und unter Benutzung der vertauschten Spaltennummern ergibt sich

$$\begin{array}{lllll} x_1 & = & y_3 & = & 8 \\ x_2 & = & y_2 & = & -9/2 \\ x_3 & = & y_1 & = & (-4+9/2-8)/3 = -5/2 \end{array}$$

Es gilt zu diesem Algorithmus

Satz 4.3.1. *Es sei A invertierbar. Dann existiert eine Zeilenpermutationsmatrix P, so daß $PA = LR$ faktorisierbar ist. L entsteht aus den im Lauf des Gauß-Algorithmus benutzten und mitvertauschten Multiplikatoren, ergänzt um die Diagonale $(1, \ldots, 1)$. R ist die resultierende obere Dreiecksmatrix und P entsteht, indem man die Zeilen der Einheitsmatrix so vertauscht, wie es der Vektor der vertauschten Zeilennummern angibt.*

Beispiel:

$$\begin{pmatrix} 1 & 2 & 4 \\ 2 & 1 & 1 \\ -1 & 0 & 4 \end{pmatrix} = (a_{ij}^{(1)})$$

1. Schritt: $k = 2$

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 4 \\ -1 & 0 & 4 \end{pmatrix} = (\tilde{a}_{ij}^{(1)})$$

Multiplikatoren: $\begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$

Neue Restmatrix:

$$\begin{aligned} \tfrac{3}{2} &= 2 - \tfrac{1}{2} \cdot 1 = a_{22}^{(2)} \\ \tfrac{7}{2} &= a_{23}^{(2)} \\ \tfrac{1}{2} &= 0 - (-\tfrac{1}{2}) \cdot 1 = a_{32}^{(2)} \\ \tfrac{9}{2} &= 4 - (-\tfrac{1}{2}) \cdot 1 = a_{33}^{(2)} \\ &\Rightarrow \begin{pmatrix} \frac{3}{2} & \frac{7}{2} \\ \frac{1}{2} & \frac{9}{2} \end{pmatrix} \end{aligned}$$

2. Schritt: kein Tausch

Multiplikator: $\frac{\frac{1}{2}}{\frac{3}{2}} = \frac{1}{3}$

Neue Restmatrix: $a_{33}^{(3)} = \frac{9}{2} - \frac{1}{3} \cdot \frac{7}{2} = \frac{10}{3}$

Vertauschte Zeilennummern: $\begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$

$$\Rightarrow P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & \frac{1}{3} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 2 & 1 & 1 \\ 0 & \frac{3}{2} & \frac{7}{2} \\ 0 & 0 & \frac{10}{3} \end{pmatrix}$$

$$LR = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 4 \\ -1 & 0 & 4 \end{pmatrix}, \quad PA = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 4 \\ -1 & 0 & 4 \end{pmatrix}$$

NUMAWWW

Nur für Sonderfälle von "fast singulären" Matrizen und bei nicht zu grosser Dimension wird die Restmatrixpivotsuche angewendet. Ein Spaltentausch in der Matrix entspricht einer Umnumerierung der Unbekannten:
Beispiel:

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -5 \\ -6 \end{pmatrix} \Rightarrow \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} -6 \\ -5 \end{pmatrix}$$

In diesem Fall lautet die Zerlegung $PAQ = LR$.

Q ist Gesamtresultat aller Spaltenvertauschungen, gegeben durch die vertauschten Spaltennummern.

Beispiel: (4 1 3 2) als vertauschte Spaltennummern. Dann ist

$$Q = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Wollen wir nun das System $A\vec{x} = \vec{b}$ lösen, so beachten wir $A\vec{x} = \vec{b} \Longleftrightarrow PA\vec{x} = P\vec{b}$ (rechte Seite mitvertauschen!) Bei der Software-Implementierung stellt man P und Q nicht als Matrizen dar, sondern als Vektoren mit den vertauschten Einträgen $(z_1, \ldots, z_n)$ bzw $(s_1, \ldots, s_n)$. Mit

$$P \hat{=} \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$$

ist

$$P\vec{b} = \begin{pmatrix} b_{z_1} \\ \vdots \\ b_{z_n} \end{pmatrix}$$

Sei $PAQ = LR$. Wir erhalten

$$\begin{aligned} \underbrace{PAQ}_{LR}\underbrace{Q^{-1}\vec{x}}_{:=\vec{y}} &= \underbrace{P\vec{b}}_{:=\vec{c}} \\ L\underbrace{R\vec{y}}_{:=\vec{z}} &= \vec{c} \end{aligned}$$

Wir gehen also in folgenden Schritten vor:

$$\begin{aligned} L\vec{z} &= \vec{c} \quad \text{ergibt } \vec{z} \\ R\vec{y} &= \vec{z} \quad \text{ergibt } \vec{y} \\ Q^{-1}\vec{x} &= Q^T\vec{x} = \vec{y} \quad \text{ergibt } \vec{x}\,. \end{aligned}$$

Die Auflösung $Q^T\vec{x} = \vec{y}$ leistet

$$x_{s_i} = y_i,\ i = 1, \ldots, n$$

Wir erhalten für die Permutationsmatrizen P und Q mittels der Einzelvertauschungen die Darstellung

$$\begin{aligned} P &= P_{n-1} \cdot \ldots \cdot P_1\,;\, P^T = P^{-1} \\ Q &= Q_1 \cdot \ldots \cdot Q_{n-1}\,;\, Q^T = Q^{-1}. \end{aligned}$$

Zur Inversion von A berechnen wir aus $PAQ = LR$

$$\begin{aligned} A &= P^T L R Q^T \qquad (\text{weil } P^{-1} = P^T,\ Q^{-1} = Q^T) \\ A^{-1} &= (Q^T)^T R^{-1} L^{-1} (P^T)^T \quad (\text{weil } (AB)^{-1} = B^{-1} A^{-1}) \\ &= Q R^{-1} L^{-1} P \\ &= Q_1 \dots Q_{n-1} R^{-1} L^{-1} P_{n-1} \dots P_1 \end{aligned}$$

d.h. nach der Berechnung der inversen Dreiecksmatrizen und deren Multiplikation hat man nun die ausgeführten Spaltenvertauschungen in umgekehrter Reihenfolge von links als Zeilenvertauschungen und entsprechend die ursprünglichen Zeilenvertauschungen in umgekehrter Reihenfolge auf die Spalten anzuwenden und erhält damit die Inverse der Ausgangsmatrix.

NUMAWWW lineare Systeme, Matrixinversion

Die explizite Ausführung der Matrixinversion ist aber nur in seltenen Ausnahmefällen wirklich erforderlich.

Die Durchführung der Vertauschungen in der oben beschriebenen Form bedeutet einen nicht unerheblichen Zeitfaktor und wirkt sich unter Umständen auch ungünstig auf die Besetztheitsstruktur der Matrizen L und R aus. Deshalb ist es wichtig, Matrizenklassen zu kennen, bei denen ohne Gefahr für das Rundungsfehlerverhalten auf die Pivotisierung verzichtet werden kann.

Matrizen, bei denen prinzipiell kein Tausch notwendig ist, sind die folgenden:

1. A symmetrisch und positiv definit, d.h.

   $$a_{i,j} = a_{j,i} \quad \text{für} \quad i, j = 1, \dots, n$$

   und

   $$x^T A x > 0 \quad \text{für alle } x \neq 0 \,.$$

2. A strikt diagonaldominant, d.h.

   $$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = 1, \dots, n \,,$$

3. A M-Matrix, d.h. folgende drei Eigenschaften sind gegeben

   $$\begin{aligned} a_{ii} &> 0 \quad \text{für } i = 1, \dots, n, \\ a_{ij} &\leq 0 \quad \text{für } i \neq j \,, \end{aligned}$$

   der betragsgrößte Eigenwert von $D^{-1}(A - D)$ ist im Betrag < 1, $D = \text{diag}\ (a_{11}, \dots, a_{nn})$

Beispiel 4.3.2. • $A = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{pmatrix}$

A ist strikt diagonaldominant.

A ist symmetrisch und positiv definit

• $A = \begin{pmatrix} 3 & -1 & -1 \\ -2 & 3 & -1 \\ -1 & -2 & 3 \end{pmatrix}$

A ist M-Matrix. $D^{-1}(A-D) = \begin{pmatrix} 0 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{2}{3} & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{2}{3} & 0 \end{pmatrix}$

□

Bemerkung 4.3.2. *Es gilt auch : die Inverse einer M-Matrix ist komponentenweise positiv.*

4.4 Gauß-Algorithmus in Spezialfällen

4.4.1 $A = A^T$ reell symmetrisch und positiv definit, Cholesky-Zerlegung, LDL^T-Zerlegung

Definition 4.4.1. *Sei* $A = A^T \in \mathbb{R}^{n\times n}$ *(bzw. im Komplexen* $A = A^H$*, wobei H transponiert und konjugiert komplex bedeutet, also* $A^H = (\bar{A})^T$ *)*
A heißt **positiv definit,** *falls* $\vec{x}^T A\vec{x} > 0$ *für alle* $\vec{x} \in \mathbb{R}^n,\ \vec{x} \neq \vec{0}$ *(bzw.* $\vec{x}^H\ A\ \vec{x} > 0$, $\vec{x} \in \mathbb{C}^n \neq 0$*).*

Bemerkung 4.4.1. *Es gelten folgende äquivalente Aussagen:*

- *A positiv definit,*
- *alle Eigenwerte sind* > 0*,*
- *alle n Hauptabschnittsunterdeterminanten, d.h.* $\det(a_{11}), \det\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, , \ldots, \det A$ *sind positiv.*

Zur Bedeutung des Begriffes "positiv definit": Sei $f(x) := \frac{1}{2}\vec{x}^T A\vec{x} - \vec{b}^T\vec{x} + c : \mathbb{R}^n \to \mathbb{R}$ mit positiv definitem A . Dann beschreibt die "(Hyper-)Fläche" $f(x) = c$ mit geeignetem c die Oberfläche eines "(Hyper-)Ellipsoids" im $\mathbb{R}^n$, für $n = 2$ also eine Ellipse.

Beispiel 4.4.1. $n = 2,\ \vec{b} = 0,\ c = 0$

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

ergibt

$$f(x) \ = \ \tfrac{1}{2}\vec{x}^T Ax \ = \ \tfrac{1}{4}\Big(\frac{(x_1 + x_2)^2}{1/3} + \frac{(x_1 - x_2)^2}{1}\Big) \ .$$

d.h. $f(x) = c > 0$ *ist die Gleichung einer Ellipse mit dem Hauptachsenverhältnis 1:3 und den Hauptachsenrichtungen* $(1, 1)$ *und* $(1, -1)$.

Im Fall einer solchen Matrix erlaubt der Gauß-sche Algorithmus eine erhebliche Vereinfachung. Es gilt nämlich, daß die Anwendung des Gauß-Algoritmhus ohne Vertauschungen möglich ist und eine Zerlegung $A = LR$ liefert mit

$$R = DL^T \text{ und } D = \text{diag } (r_{11}, \ldots, r_{nn}) \ .$$

Beispiel 4.4.2. $A = \begin{pmatrix} 1 & -1 & -2 & -3 \\ -1 & 5 & 8 & -5 \\ -2 & 8 & 29 & -26 \\ -3 & -5 & -26 & 75 \end{pmatrix}$

$$\left(\begin{array}{cccc} 1 & -1 & -2 & -3 \\ \hline -1 & \multicolumn{3}{|c}{\multirow{3}{*}{$\begin{array}{ccc} 4 & 6 & -8 \\ 6 & 25 & -32 \\ -8 & -32 & 66 \end{array}$}} \\ -2 & \multicolumn{3}{|c}{} \\ -3 & \multicolumn{3}{|c}{} \end{array}\right) \quad \textit{Restmatrix wieder symmetrisch}$$

$$\left(\begin{array}{cccc} 1 & -1 & -2 & -3 \\ \hline -1 & \multicolumn{3}{|c}{\multirow{3}{*}{$\begin{array}{ccc} 4 & 6 & -8 \\ \frac{3}{2} & 16 & -20 \\ -2 & -20 & 50 \end{array}$}} \\ -2 & \multicolumn{3}{|c}{} \\ -3 & \multicolumn{3}{|c}{} \end{array}\right) \quad \textit{Restmatrix wieder symmetrisch}$$

$$\left(\begin{array}{cccc} 1 & -1 & -2 & -3 \\ \hline -1 & \multicolumn{3}{|c}{\multirow{3}{*}{$\begin{array}{ccc} 4 & 6 & -8 \\ \frac{3}{2} & 16 & -20 \\ -2 & -\frac{5}{4} & 25 \end{array}$}} \\ -2 & \multicolumn{3}{|c}{} \\ -3 & \multicolumn{3}{|c}{} \end{array}\right)$$

$$\Rightarrow L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -2 & \frac{3}{2} & 1 & 0 \\ -3 & -2 & -\frac{5}{4} & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 25 \end{pmatrix}$$

$$R = \begin{pmatrix} 1 & -1 & -2 & -3 \\ 0 & 4 & 6 & -8 \\ 0 & 0 & 16 & -20 \\ 0 & 0 & 0 & 25 \end{pmatrix}$$

$$A = LDL^T$$

□

Der Beweis dieser Behauptung benutzt

Satz 4.4.1. *Wird der Gauß-sche Algorithmus ohne Vertauschungen bis zum Schritt k ($1 \leq k \leq n$) durchgeführt, dann gilt*

$$\det \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ldots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix} = \textit{k-te Hauptabschnittsunterdeterminante} = \prod_{i=1}^{k} a_{ii}^{(i)} = \textit{Produkt der Pivots } 1 \ldots, k \ .$$

d.h. A positiv definit $\Leftrightarrow a_{ii}^{(i)} > 0, \ i = 1, \ldots, n.$ □

Wir erhalten dann

$$\begin{aligned} A &= LDL^T, \quad \text{wobei} \quad D = \text{diag}\,(a_{11}^{(1)}, \ldots, a_{nn}^{(n)}) \\ D^{\frac{1}{2}} &\stackrel{def}{=} \text{diag}\left(\sqrt{a_{11}^{(1)}}, \ldots, \sqrt{a_{nn}^{(n)}}\right) \quad \text{mit} \quad D^{\frac{1}{2}} D^{\frac{1}{2}} = D \\ \Rightarrow A &= LDL^T = LD^{\frac{1}{2}} D^{\frac{1}{2}} L^T = \tilde{L}\tilde{L}^T \end{aligned}$$

Also eine neue (symmetrische) Form der Dreieckszerlegung. Diese Zerlegung ist nach **Cholesky** benannt (1925). Dazu gilt

Satz 4.4.2. *Genau dann ist A symmetrisch und positiv definit, wenn eine untere Dreiecksmatrix $\tilde{L}$ mit positiven Diagonalelementen existiert, so daß*

$$A = \tilde{L}\tilde{L}^T \text{ Cholesky-Zerlegung .}$$

□

Aus dem Ansatz $A = \tilde{L}\tilde{L}^T$ folgt die Beziehung

$$a_{jk} = \sum_{i=1}^{k} \tilde{l}_{ji}\tilde{l}_{ki} \quad \text{für} \quad k \leq j \quad \text{und} \quad j = 1, \ldots, n.$$

und dies wiederum hat zur Folge, daß kein Element von $\tilde{L}$ grösser werden kann als die Wurzel aus dem grössten Element von A (das notwendig auf der Diagonalen auftritt). Wir berechnen die Elemente von $\tilde{L}$ in der folgenden Reihenfolge:

$$\tilde{l}_{11}, \ldots, \tilde{l}_{n1}, \tilde{l}_{22}, \ldots, \tilde{l}_{n2}, \ldots, \tilde{l}_{nn}$$

durch die Berechnungsvorschriften

$$\begin{aligned} &\text{für } j = 1, \ldots, n: \\ &\qquad \tilde{l}_{jj} = \sqrt{a_{jj} - \sum_{i=1}^{j-1} \tilde{l}_{ji}^2} \\ &\text{für } k = j+1, \ldots, n: \\ &\qquad \tilde{l}_{kj} = (a_{kj} - \sum_{i=1}^{j-1} \tilde{l}_{ji}\tilde{l}_{ki})/\tilde{l}_{jj} \end{aligned}$$

Die Vorteile dieses Verfahrens sind

- Eine Einsparung von Speicherplatz und Rechenzeit (halb so viel wie für den Gauß-Algorithmus) und

- geringere Rundungsfehler bei der "Produktsummenakkumulation" in der Arithmetik-Einheit.

Beispiel 4.4.3.

$$A = \begin{pmatrix} 1 & -1 & -2 & -3 \\ -1 & 5 & 8 & -5 \\ -2 & 8 & 29 & -26 \\ -3 & -5 & -26 & 75 \end{pmatrix}$$

$$\Rightarrow L =$$

$$\begin{pmatrix} 1 & & & \\ -1 & \sqrt{5-(-1)^2}=2 & & \\ -2 & (8-(-2)(-1))/2=3 & \sqrt{29-3^2-(-2)^2}=4 & \\ -3 & (-5-(-3)(-1))/2=-4 & (-26-(-4)3-(-3)(-2))/4=-5 & \sqrt{75-(-5)^2-(-4)^2-(-3)^2}=5 \end{pmatrix} \quad \square$$

NUMAWWW lineare Gleichungssyteme, Choleskyzerlegung

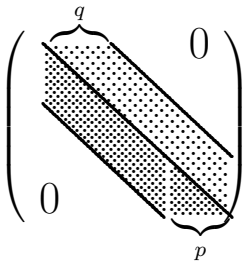
Bemerkung 4.4.2. *Der Cholesky-Algorithmus ist die effizienteste allgemeine Testmethode auf positive Definitheit. Man kann auf das Rechnen mit den Quadratwurzeln verzichten, indem man unter Berücksichtigung der Symmetrie den Gauß-Algorithmus wie gewohnt durchführt und nur die Pivots in einer Diagonalmatrix D (also programmtechnisch in einem Vektor) ablegt und L wie üblich belässt. Dies ergibt dann die sogenannte LDL^T-Zerlegung.*

Bem.: Eine ähnliche symmetrische Zerlegung (jetzt aber mit symmetrischen Zeilen- und Spaltenvertauschungen) gibt es auch für indefinite symmetrische Matrizen. Dabei muss man aber in D auch 2×2 Untermatrizen zulassen, was bedeutet, daß zwei Spalten auf einmal eliminiert werden. Dies ist die sogenannte Bunch-Parlett-Zerlegung. . Die gewöhnliche Gauß-Zerlegung darf man hier nicht benutzen, der Rundungsfehlereinfluss ist dann nicht kontrollierbar.

4.4.2 Schwach besetzte Matrizen

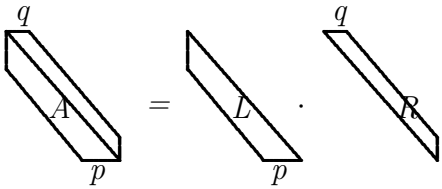
Definition 4.4.2. *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt* (p,q)-**Bandmatrix**, *wenn gilt:*

$$a_{ij} = 0, \quad \textit{falls} \;\; j < i-p \;\; \textit{oder} \;\; j > i+q.$$



Wichtiger Spezialfall: $p = q = 1$: **Tridiagonalmatrix** □

Satz 4.4.3. *Falls $A = L \cdot R$ mit $L = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$ durchführbar ist, dann besitzt für eine (p,q)-Bandmatrix L die Struktur einer $(p,0)$ und R die einer $(0,q)$-Matrix, d.h.*



Speziell für symmetrische positiv definite Matrizen $A = L \cdot L^T$ (Cholesky-Zerlegung): Es genügt, eine "Hälfte" von A zu speichern als $n\times(p+1)$-Matrix und L kann ganz in diesem Bereich abgelegt werden. Dies bedeutet eine erhebliche Einsparung an Speicherplatz und Rechenaufwand für solche Matrizen.

Beispiel 4.4.4.

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \longrightarrow \boxed{\begin{matrix} * & 2 \\ -1 & 2 \\ -1 & 2 \\ -1 & 2 \end{matrix}} \quad \textit{gespeichert als } 4\times 2\textit{-Matrix}$$

Allgemein kann man eine (p,q)-Bandmatrix als eine $n \times (p+q+1)$-Rechtecksmatrix speichern mit der Indexabbildung

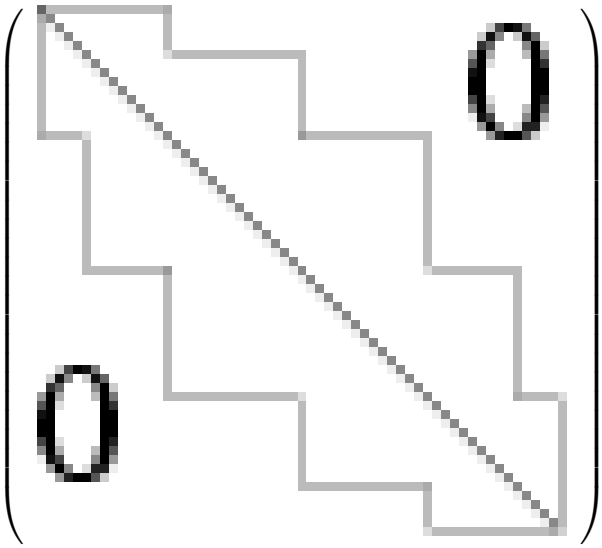
$$a_{i,j} \;\; \longrightarrow \;\; \tilde{a}_{i,j-i+p+1} \;\; ,$$

die Diagonale der ursprünglichen Matrix steht also in Spalte $p+1$. (Einige Elemente links oben und rechts unten bleiben so undefiniert, was aber nicht stört.) Wenn man keine Vertauschungen benötigt, kann man den Gauss'schen Algorithmus nun ganz in diesem Rechtecksfeld ablaufen lassen.

Aufwand: (für $p = q$) von $\mathcal{O}(np^2)$ Multiplikationen/Additionen (statt $\frac{n^3}{3} + \mathcal{O}(n^2)$ für den allgemeinen Fall).

Bei solchen Matrizen wendet man niemals Spaltentausch an! Zeilentausch bewirkt Verbreiterung der Bandbreite von R auf $(0, p+q)$.
Neben den Bandmatrizen treten in vielen Anwendungen noch allgemeinere "dünn besetzte" Matrizen auf (engl: sparse matrices). ,

Definition 4.4.3. *Sei $A = A^T$. Ferner gelte $a_{ij} = 0$ für $j < k(i),\ i = 1, \ldots, n$ mit $k(i) = 1$ für $a_{i,1} \neq 0$.*
Dann heißt $(k(i), i)$ die **Einhüllende** *von A (nur unteres Dreieck).*



□

Dazu gilt

Satz 4.4.4. *Der Gaußsche Algorithmus ohne Pivotisierung erhält die Einhüllende einer symmetrischen Matrix.*
Aber: Innerhalb der Einhüllenden werden Nullen in der Regel zerstört!

Englischer Begriff dazu : "**fill in**" Das Ausmaß des "fill in" hängt von der Numerierung der Gleichungen und Unbekannten ab.

(Spezielle Strategien: siehe Spezialliteratur)

Beispiel 4.4.5. *Dreieckszerlegung einer Dreibandmatrix mit natürlicher Pivotwahl*

$$A = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & & & \\ -1 & 1 & -1 & & \\ & -1 & 1 & -1 & \\ & & -1 & 1 & -1 \\ & & & -1 & 1 \end{pmatrix}$$

Dreieckszerlegung ohne Vertauschungen

□

Man beachte, daß die Inverse einer Bandmatrix in der Regel voll besetzt ist, die explizite Inversion wäre hier ein grober Kunstfehler. Ein kleines Beispiel für "fill in" unter Erhaltung der Bandstruktur:

Beispiel 4.4.6. *Gegeben sei eine positiv definite und symmetrische Matrix. Das untere Dreieck dieser Matrix habe folgende Besetztheitsstruktur (wegen der Symmetrie wird immer nur das untere bzw. obere Dreieck gespeichert)*

$$\left(\begin{array}{ccc|ccc} \times & \times & \times & \times & & \\ \times & \times & & & \times & \\ \times & & \times & & & \times \\ \hline \times & & & \times & & \\ & \times & & & \times & \\ & & \times & & & \times \end{array}\right).$$

Dabei steht $\times$ für einen Eintrag ungleich Null. Wir führen nun symbolisch den Cholesky-Algorithmus durch:

$$\begin{aligned} l_{11} &= \sqrt{a_{11}} = \sqrt{\times} = \times \\ l_{21} &= \frac{1}{l_{11}} a_{21} = \frac{1}{\times} \times = \times \\ l_{31} &= \frac{1}{l_{11}} a_{31} = \frac{1}{\times} \times = \times \\ l_{41} &= \frac{1}{l_{11}} a_{41} = \frac{1}{\times} \times = \times \\ l_{51} &= \frac{1}{l_{11}} a_{51} = \frac{1}{\times} \bigcirc = \bigcirc \\ l_{61} &= \frac{1}{l_{11}} a_{61} = \frac{1}{\times} \bigcirc = \bigcirc \end{aligned}$$

$$
\begin{aligned}
l_{22} &= \sqrt{a_{22} - l_{21}^2} = \sqrt{\times - \times^2} = \times \\
l_{32} &= \frac{1}{l_{22}}(a_{32} - l_{21}l_{31}) = \frac{1}{\times}(\bigcirc - \times\times) = \times \\
l_{42} &= \frac{1}{l_{22}}(a_{42} - l_{21}l_{41}) = \frac{1}{\times}(\bigcirc - \times\times) = \times \\
l_{52} &= \frac{1}{l_{22}}(a_{52} - l_{21}l_{51}) = \frac{1}{\times}(\times - \times\bigcirc) = \times \\
l_{62} &= \frac{1}{l_{22}}(a_{62} - l_{21}l_{61}) = \frac{1}{\times}(\bigcirc - \times\bigcirc) = \bigcirc
\end{aligned}
$$

$$
\begin{aligned}
l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2} = \sqrt{\times - \times^2 - \times^2} = \times \\
l_{43} &= \frac{1}{l_{33}}(a_{43} - l_{31}l_{41} - l_{32}l_{42}) = \frac{1}{\times}(\bigcirc - \times\times - \times\times) = \times \\
l_{53} &= \frac{1}{l_{33}}(a_{53} - l_{31}l_{51} - l_{32}l_{52}) = \frac{1}{\times}(\bigcirc - \times\bigcirc - \times\times) = \times \\
l_{63} &= \frac{1}{l_{33}}(a_{63} - l_{31}l_{61} - l_{32}l_{62}) = \frac{1}{\times}(\times - \times\bigcirc - \times\bigcirc) = \times
\end{aligned}
$$

$$
\begin{aligned}
l_{44} &= \sqrt{a_{44} - l_{41}^2 - l_{42}^2 - l_{43}^2} = \sqrt{\times - \times^2 - \times^2 - \times^2} = \times \\
l_{54} &= \frac{1}{l_{44}}(a_{54} - l_{41}l_{51} - l_{42}l_{52} - l_{43}l_{53}) = \frac{1}{\times}(\bigcirc - \times\bigcirc - \times\times - \times\times) = \times \\
l_{64} &= \frac{1}{l_{44}}(a_{64} - l_{41}l_{61} - l_{42}l_{62} - l_{43}l_{63}) = \frac{1}{\times}(\bigcirc - \times\bigcirc - \times\bigcirc - \times\times) = \times
\end{aligned}
$$

$$
\begin{aligned}
l_{55} &= \sqrt{a_{55} - l_{51}^2 - l_{52}^2 - l_{53}^2 - l_{54}^2} \\
&= \sqrt{\times - \bigcirc^2 - \times^2 - \times^2 - \times^2} = \times \\
l_{65} &= \frac{1}{l_{55}}(a_{65} - l_{51}l_{61} - l_{52}l_{62} - l_{53}l_{63} - l_{54}l_{64}) \\
&= \frac{1}{\times}(\bigcirc - \bigcirc\bigcirc - \times\bigcirc - \times\times - \times\times) = \times \\
l_{66} &= \sqrt{a_{66} - l_{61}^2 - l_{62}^2 - l_{63}^2 - l_{64}^2 - l_{65}^2} \\
&= \sqrt{\times - \bigcirc^2 - \bigcirc^2 - \times^2 - \times^2 - \times^2} = \times
\end{aligned}
$$

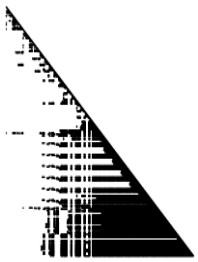
Damit hat L die Struktur

$$
\left(\begin{array}{ccc|ccc}
\times & & & & & \\
\times & \times & & & & \\
\times & \times & \times & & & \\
\hline
\times & \times & \times & \times & & \\
\bigcirc & \times & \times & \times & \times & \\
\bigcirc & \bigcirc & \times & \times & \times & \times
\end{array}\right).
$$

Ein Teil der Besetztheitsstrukur ist also verloren gegangen, die Bandstruktur ist jedoch erhalten geblieben.

Und hier ein Eindruck von der Struktur allgemeiner dünnbesetzter Matrizen: die Nichtnullelemente sind als schwarze Punkte gekennzeichnet.

Eine dünn besetzte Matrix und ihre Cholesky-Zerlegung



4.5 Störeinfluß bei der Lösung linearer Gleichungssysteme

In diesem Abschnitt beschäftigen wir uns mit der folgenden Fragestellung:

Vorgelegt seien

$$A\vec{x} = \vec{b}$$

sowie ein gestörtes System der Form

$$\tilde{A}\tilde{\vec{x}} = \tilde{\vec{b}}$$

mit $A - \tilde{A}$ "klein" und $\vec{b} - \tilde{\vec{b}}$ "klein".

Es stellt sich nun die folgende Frage: Wie "klein" ist $\vec{x} - \tilde{\vec{x}}$? Diese Frage ist von grösster praktischer Bedeutung, da sehr häufig die Koeffizienten eines Gleichungssystems selbst bereits berechnete Rundungs- oder Approximations-Fehler behaftete Grössen sind. Es stellt sich heraus, daß eine einfache Kennzahl, die sogenannte "Konditionszahl" der Matrix, diesen Störeinfluss beschreibt. Man kann häufig ein gestelltes Problem in verschiedener Weise als lineares Gleichungssystem formulieren und man wird dann natürlich den Weg wählen, der zur kleinsten Konditionszahl führt. Die Unterschiede hierin können riesig sein.

Beispiel 4.5.1. *Gegeben seien*

$$A = \begin{pmatrix} 1 & -2 & 0 & 0 \\ 1 & -2.1 & -4 & 0 \\ 0 & -0.1 & -4.01 & -8 \\ 0 & 0 & -0.01 & -8.001 \end{pmatrix} = L \cdot R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 0 & 0 \\ 0 & -0.1 & -4 & 0 \\ 0 & 0 & -0.01 & -8 \\ 0 & 0 & 0 & -0.001 \end{pmatrix}$$

$$\vec{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow \vec{x} = \begin{pmatrix} 8001 \\ 4000 \\ -100 \\ 0 \end{pmatrix}$$

Sei nun $\tilde{\vec{x}}$ gesucht, sodass mit $(\tilde{A} = A)$

$$\tilde{A}\tilde{\vec{x}} = A\tilde{\vec{x}} \overset{def}{=} \tilde{\vec{b}} = \vec{b} + \begin{pmatrix} 10^{-8} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \tilde{\vec{x}} = \begin{pmatrix} 8001.6.... \\ 4000.32... \\ -100.08 \\ 10^{-5} \end{pmatrix}$$

Der Fehler von 10^{-8} in $\tilde{\vec{b}}$ hat sich auf $6 \cdot 10^{-1}$ in $\tilde{\vec{x}}$ vervielfacht.

Dieses abschreckende Resultat hat eine einfache Erklärung:

$$
\begin{array}{l}
A\tilde{\vec{x}} - \vec{b} = \vec{r} \\
A\vec{x} - \vec{b} = 0 \\
\hline
\end{array}
$$

$$
\begin{array}{l}
A\tilde{\vec{x}} - A\vec{x} \;=\; \vec{r} \\
A(\tilde{\vec{x}} - \vec{x}) = \vec{r} \\
\tilde{\vec{x}} - \vec{x} = A^{-1}\vec{r} \\
\tilde{\vec{x}} = \vec{x} + A^{-1}\vec{r} \\
A^{-1} = R^{-1} \cdot L^{-1} = \begin{pmatrix} 64. & \cdot 10^6 & \cdots \\ & \vdots & \cdots \end{pmatrix} \\
\tilde{\vec{x}} - \vec{x} \;=\; \begin{pmatrix} 64. & \cdot 10^6 & \cdots \\ & \vdots & \cdots \end{pmatrix} \begin{pmatrix} 10^{-8} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.64... \\ \vdots \\ \vdots \\ \vdots \end{pmatrix}
\end{array}
$$

□

Die formale Behandlung dieser Frage wird sehr einfach, wenn man die Störungen nicht komponentenweise betrachtet, sondern auf ein pauschales "Grössenmaß" reduziert, nämlich auf Normbetrachtungen. Dabei ist "Norm" eine geeignete Verallgemeinerung des Begriffs der euklidischen Länge eines Vektors. Dafür benutzen wir in Zukunft das Symbol $\|.\|$. Die euklidische Länge schreiben wir als

$$\|\vec{x}\|_2 \;=\; \sqrt{\sum_{i=1}^{n} |x_i|^2}$$

Definition 4.5.1. *Eine Abb.:* $\|\cdot\|: \mathbb{K}^n \to \mathbb{R}_+ \quad (\mathbb{K} \in \{\mathbb{C}, \mathbb{R}\})$ *heißt Vektornorm auf* $\mathbb{K}^n$*, falls sie folgenden Gesetzen genügt:*

(V1) $\forall \vec{x} \in \mathbb{K}^n: \ ||\vec{x}|| \geq 0 \quad ||\vec{x}|| = 0 \Leftrightarrow \vec{x} = 0.$ **Definitheit**

(V2) $\forall \alpha \in \mathbb{K}, \forall \vec{x} \in \mathbb{K}^n: \quad ||\alpha\vec{x}|| = |\alpha| \ ||\vec{x}||$ **Homogenität**

(V3) $\forall \vec{x}, \vec{y} \in \mathbb{K}^n: \quad ||\vec{x} + \vec{y}|| \leq ||\vec{x}|| + ||\vec{y}||$ **Dreiecksungleichung**

□

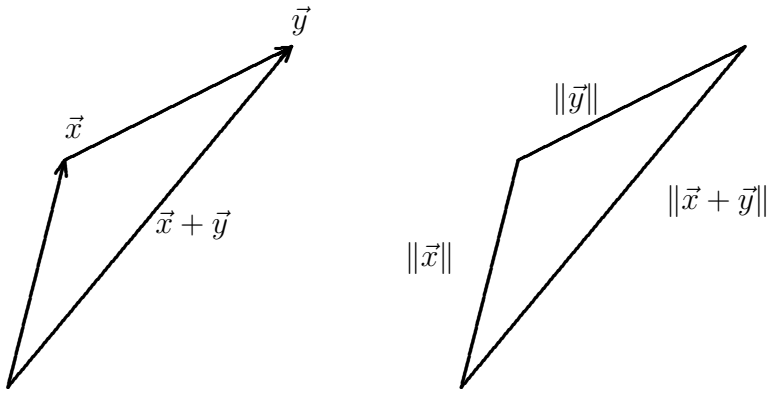


Abbildung 5.5.1

Beispiel 4.5.2.

$$\|\vec{x}\|_p := (\sum_{i=1}^{n} |x_i|^p)^{1/p} \quad 1 \leq p < \infty, \quad p = 2 \quad \textit{euklidische Norm}$$
$$\|\vec{x}\|_\infty := \max_{i=1,\ldots,n} |x_i| \qquad \textit{Maximumnorm}$$

(Beweis der Normeigenschaften elementare Übungsaufgabe) □

Aus (V3) leitet man (wie bei der Betragsfunktion) her die

zweite Dreiecksungleichung
(V4) $||\vec{x} + \vec{y}|| \geq | \ ||\vec{y}|| - ||\vec{x}|| \ | \quad (\forall \vec{x}, \vec{y} \in \mathbb{K}^n)$

Normen sind stetige Funktionen auf $\mathbb{R}^n$ bzw. $\mathbb{C}^n$. Zu zwei beliebigen Vektornormen $||.||$ und $||.||^*$ gibt es stets zwei Konstanten $C_1,\ C_2$ (die von diesen Normen abhängen) mit

$$C_1||\vec{x}|| \ \leq \ ||\vec{x}||^* \ \leq \ C_2||\vec{x}|| \quad \text{für alle } \vec{x}\ .$$

Man sagt, in einem endlich dimensionalen Raum seien alle Normen topologisch gleichwertig. Da die Menge aller $n \times m$–Matrizen über $\mathbb{K}$ einen linearen Vektorraum der Dimension nm bildet, kann man auch hierfür Normen einführen, die den Gesetzen

(V1)–(V3) aus Def. 5.5.1 genügen. Für das praktische Arbeiten sind diese Eigenschaften jedoch noch nicht ausreichend, weil man ja auch Normen von Matrizenprodukten durch Normen der Faktoren ausdrücken können will. Dies führt zu

Definition 4.5.2. *Eine Abb.* $\|\cdot\| \quad \mathbb{K}^{n\times n} \to \mathbb{R}_+$ *heißt* **Matrixnorm** *auf* $\mathbb{K}^{n\times n}$, *falls gilt:* $\forall\, A, B \in \mathbb{K}^{n\times n}, \quad \forall\, \alpha \in \mathbb{K}$*:*

(M1) $||A|| \geq 0, \quad A = 0 \Leftrightarrow ||A|| = 0$

(M2) $||\alpha A|| = |\alpha| \;\; ||A||$

(M3) $||A + B|| \leq ||A|| + ||B||$

(M4) $||AB|| \leq ||A|| \;\; ||B||$ **(Submultiplikativität)**

□

Bemerkung 4.5.1. *In (M4) benötigen wir die Relation "*$\leq$*", denn "=" kann nicht gelten wegen*

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \;\; B = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow \;\; AB = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

und $||AB|| = 0, \;\; ||A||||B|| > 0.$ □

Bemerkung 4.5.2. *Die Einschränkung auf* $n \times n$*–Matrizen in Def.* 5.5.2. *ist dadurch bedingt, daß man Normen dimensionsabhängig definieren kann. (vgl. Bsp)* **Bei vielen praktisch wichtigen Normen bleiben jedoch (M1)–(M4) gültig, wenn für** A **und** B **beliebige verknüpfbare Rechteckmatrizen stehen**. □

Ist A eine $n \times n$ Matrix und $x \in \mathbb{K}^n$ d.h. $Ax \in \mathbb{K}^n$, dann können wir folgende Normen betrachten:

$$||A\vec{x}||, \qquad\qquad ||A||, \qquad\qquad ||\vec{x}||$$

$$\|\cdot\| \text{ Norm auf } \mathbb{K}^n \quad \|\cdot\| \text{ Matrixnorm auf } \mathbb{K}^{n\times n}$$

Ein für die Praxis sinnvoller Zusammenhang (interpretiere x als $n \times 1$ Matrix) ist offensichtlich

$$||A\vec{x}|| \leq ||A|| \;\; ||\vec{x}||$$

Diese Überlegung führt zu

Definition 4.5.3. *Die Matrixnorm* $\|\cdot\|$ *auf* $\mathbb{K}^{n\times n}$ *heißt* <u>*verträglich*</u> *mit der Vektornorm* $\|\cdot\|$ *auf* $\mathbb{K}^n$ *, falls*

(M5) $\quad ||A\vec{x}|| \leq ||A|| \;\; ||\vec{x}|| \qquad \forall \vec{x} \in \mathbb{K}^n$ □

Satz 4.5.1. *Ist $\|\cdot\|$ eine Vektornorm auf $\mathbb{K}^n$, dann wird durch die Definition*

$$||A|| := \max_{||\vec{x}||=1} ||A\vec{x}||$$

eine Matrixnorm eingeführt, die (M1)–(M5) erfüllt. Man bezeichnet sie als die **der Vektornorm zugeordnete Matrixnorm**.

Es gibt Matrixnormen, die mit einer Vektornorm verträglich sind, ohne ihr zugeordnet zu sein, z.B. die Kombination

$$||\vec{x}|| \;=\; \max\{|x_i|\} \;\text{ und }\; ||A|| \;=\; n\max\{|a_{i,j}|\}$$

und

$$||\vec{x}|| \;=\; (\sum_{i=1}^{n} |x_i|^2)^{1/2} \;\text{ und }\; ||A|| \;=\; (\sum_{i,j=1}^{n} |a_{i,j}|^2)^{1/2} \text{ Frobeniusnorm .}$$

Die einer Vektornorm zugeordnete Matrixnorm ist also über eine Maximierungsaufgabe definiert. In einigen wichtigen Fällen kann man diese Maximierungsaufgabe explizit lösen. Dazu gilt

Satz 4.5.2. *Es gilt*

1. $$||A||_\infty \overset{def}{=} \max_{\vec{x}\neq 0} \frac{||A\vec{x}||_\infty}{||\vec{x}||_\infty} \;=\; \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{i,j}|$$

2. $$||A||_1 \overset{def}{=} \max_{\vec{x}\neq 0} \frac{||A\vec{x}||_1}{||\vec{x}||_1} \;=\; \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{i,j}|$$

3. $$||A||_2 \overset{def}{=} \max_{\vec{x}\neq 0} \frac{||A\vec{x}||_2}{||\vec{x}||_2} \;=\; \max\{\sqrt{\sigma_i} :\; \sigma_i \text{ Eigenwerte von } A^TA\}$$

□

Diese Ergebnisse erklären, weshalb diese Normen auch als Zeilensummennorm, Spaltensummennorm und Spektralnorm bezeichnet werden.

Satz 4.5.3. *Ist $||.||$ eine Vektornorm und T eine feste invertierbare Matrix, dann ist auch*

$$||\vec{x}||_T \overset{def}{=} ||T\vec{x}||$$

eine Vektornorm und die zugeordnete Matrixnorm ist

$$||A||_T \;=\; ||TAT^{-1}||$$

□

Da solche Normen in Fehlerabschätzungen auftreten, ist es natürlich, nach Normen zu suchen, für die die zugeordnete Matrixnorm einer Matrix möglichst klein ist. Dazu gilt

Satz 4.5.4. *1. Ist $||.||$ eine einer beliebigen Vektornorm zugeordnete Matrixnorm, dann gilt für jede Matrix A*

$$\max_i\{|\lambda_i| \ : \lambda_i \text{ ein Eigenwert von } A\} \ \leq \ ||A|| \ .$$

2. Zu jeder Matrix B und jedem $\varepsilon > 0$ gibt es eine (in der Regel von B und ε abhängende) Norm $||.||_{B,\varepsilon}$ mit

$$||B||_{B,\varepsilon} \ \leq \ \max_i\{|\lambda_i| \ : \ \lambda_i \text{ ein Eigenwert von } B\} + \varepsilon$$

□

Beispiel 4.5.3. *Sei*

$$A \ = \ \begin{pmatrix} 0.8 & 10000 \\ 0 & 0.7 \end{pmatrix}$$

und $\varepsilon = 10^{-3}$. Es ist

$$||A||_1 \ = \ 10000.7, \ ||A||_\infty \ = 10000.8, \ ||A||_2 \ = \ 10000.0000565$$

aber

$$\lambda_1 \ = \ 0.8 \quad \lambda_2 \ = \ 0.7 \ .$$

Man setze

$$||x|| \ = \ max\{|x_1|, 10^7|x_2|\} \ = \ ||diag(1, 10^7)x||_\infty$$

Wegen Satz 4.5.3 ist dann

$$||A|| \ = \ \left|\left|\begin{pmatrix} 0.8 & 10^{-3} \\ 0 & 0.7 \end{pmatrix}\right|\right|_\infty \ = \ 0.801$$

Man nennt

$$\max_i\{|\lambda_i| \ : \ \lambda_1, \ldots, \lambda_n \text{ die Eigenwerte von } B\}$$

den **Spektralradius** von B und benutzt dafür das Symbol $\varrho(B)$.

Definition 4.5.4.

$$\varrho(B) \ \stackrel{def}{=} \ \max\{|\lambda| \ : \lambda \text{ ein Eigenwert von } B\} \ .$$

Der obige Satz besagt, daß keine Matrixnorm den Spektralradius unterbieten kann. Wir kommen nun zur Anwendung dieser Begriffe auf unsere Fragestellung.

Satz 4.5.5. Banach perturbation Lemma *Es sei $\|\cdot\|$ eine Vektornorm auf $\mathbb{R}^n$ bzw. $\mathbb{C}^n$. Als Matrixnorm auf $\mathbb{R}^{n\times n}$ bzw. $\mathbb{C}^{n\times n}$ werde die zugeordnete Matrixnorm verwendet. Falls $H \in \mathbb{C}^{n\times n}$ und*

$$\|H\| < 1$$

dann ist $I + H$ regulär und es gilt

(i) $\|(I+H)^{-1}\| \leq 1/(1-\|H\|)$

(ii) $\|(I+H)^{-1} - I\| \leq \|H\|/(1-\|H\|)$

□

Dieser Satz ist sehr nützlich. So hat man z.B. bei Verfahren zur Lösung von gewöhnlichen und partiellen Differentialgleichungen häufig lineare Systeme mit einer Matrix

$$I + (\Delta t)A$$

zu lösen, wobei Δt ein Zeitinkrement ist und A die Jacobimatrix einer vektorwertigen Funktion. Der Satz besagt hier, daß solch ein System für genügend kleine Zeitschritte immer eindeutig lösbar ist.

Beispiel 4.5.4.

$$A = \begin{pmatrix} 1.0 & 0.3 & 0.3 & 0.4 \\ -0.2 & 1.0 & 0.0 & 0.2 \\ 0.2 & -0.6 & 1.0 & 0.1 \\ 0.4 & 0.0 & 0.6 & 1.0 \end{pmatrix}$$

Hier ist offenbar $||.||_\infty$ ungeeignet, während $||.||_1$ den Wert $||H||_1 = 0.9$ ergibt, die Matrix ist also invertierbar, ihre Inverse hat eine 1-Norm ≤ 10.

$$A = \begin{pmatrix} 3 & 900 \\ 0.004 & 4 \end{pmatrix}$$

Hier bringen wir A durch Multiplikation mit einer Diagonalmatrix aus den reziproken Diagonalelementen auf die gewünschte Gestalt:

$$diag(\tfrac{1}{3}, \tfrac{1}{4})A = \begin{pmatrix} 1 & 300 \\ 0.001 & 1 \end{pmatrix}$$

A ist offenbar genau dann invertierbar, wenn die Matrix auf der rechten Seite dies ist. Nun können wir keine der "Standardnormen" benutzen. Wählen wir aber

$$||\vec{x}|| \stackrel{def}{=} \max\{|x_1|, 600|x_2|\} = ||diag(1,600)\vec{x}||_\infty ,$$

dann wird die zugeordnete Matrixnorm zu

$$||diag(1,600)(.)(diag(1,600))^{-1}||_\infty$$

und die so transformierte Matrix

$$\begin{pmatrix} 1 & 0.5 \\ 0.6 & 1 \end{pmatrix}$$

erlaubt nun die Anwendung des Satzes. □

Wir gelangen nun zum allgemeinen Störungssatz für lineare Gleichungssysteme:

Satz 4.5.6. Störungssatz für lineare Gleichungssyteme *Sei* $A \in \mathbb{K}^{n\times n}$ *regulär,* $\vec{b} \neq 0$, $\vec{b} \in \mathbb{K}^n$, $\tilde{A} \in \mathbb{K}^{n\times n}$, $\tilde{\vec{b}} \in \mathbb{K}^n$. *Es gelte in der der Vektornorm* $\|\cdot\|$ *zugeordneten Matrixnorm*

$$\|A^{-1}\|\,\|\tilde{A} - A\| < 1$$

Ferner sei $\vec{x} := A^{-1}\vec{b}$. *Dann ist* $\tilde{A}$ *invertierbar und für die eindeutig bestimmte Lösung* $\tilde{\vec{x}}$ *von* $\tilde{A}\tilde{\vec{x}} = \tilde{\vec{b}}$ *gilt*

$$\frac{\|\tilde{\vec{x}} - \vec{x}\|}{\|\vec{x}\|} \leq \text{cond}_{\|\cdot\|}(A) \left(\frac{\|\tilde{\vec{b}} - \vec{b}\|}{\|\vec{b}\|} + \frac{\|\tilde{A} - A\|}{\|A\|} \right) \frac{1}{1 - \text{cond}_{\|\cdot\|}(A) \frac{\|\tilde{A}-A\|}{\|A\|}} \tag{4.1}$$

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

Definition 4.5.5. *Die Größe* $\text{cond}_{\|\cdot\|}(A) := \|A\|\,\|A^{-1}\|$ *heißt die* **Konditionszahl** *der Matrix bezüglich der Gleichungslösung in der Norm* $\|\cdot\|$. □

Bemerkung 4.5.3. *Es gilt stets* $\text{cond}_{\|\cdot\|}(A) \geq \varrho(A)\varrho(A^{-1}) \geq 1$.
(vgl.Satz 4.5.4). Falls $\text{cond}_{\|\cdot\|}(A) \gg 1$, *dann besagt dies, daß schon geringe Fehlereinflüsse (in der Matrix A oder z.B. Rundungsfehlereinflüsse bei der Gleichungsauflösung, die man so deuten kann, als wäre die Ausgangsmatrix A abgeändert worden bei anschließender exakter Rechnung) eine starke Veränderung der Lösung des Gleichungssystems hervorrufen können. Man sagt dann, das Gleichungssystem sei "schlecht konditioniert".* □

Beispiel 4.5.5. *Wir betrachten die Matrix*

$$A = \begin{pmatrix} 0.99 & 0.98 \\ 0.98 & 0.97 \end{pmatrix}$$

sowie dem Vektor

$$\vec{b} = \begin{pmatrix} -1.97 \\ -1.95 \end{pmatrix}$$

Durch Störung gehe daraus hervor das System mit

$$\tilde{A} = \begin{pmatrix} 0.990005 & 0.979996 \\ 0.979996 & 0.970004 \end{pmatrix}$$

sowie dem Vektor

$$\tilde{\vec{b}} = \begin{pmatrix} -1.969967 \\ -1.950035 \end{pmatrix}.$$

Die Lösung des Ausgangssystems ist

$$\left[\begin{array}{cc|c} 0.99 & 0.98 & -1.97 \\ 0.98 & 0.97 & -1.95 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 0.99 & 0.98 & -1.97 \\ 0 & -0.0001 & 0.0001 \end{array}\right] \longrightarrow \tilde{\vec{x}} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

Die Lösung des gestörten Systems dagegen

$$\left[\begin{array}{cc|c} 0.990005 & 0.979996 & -1.969967 \\ 0.979996 & 0.970004 & -1.950035 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 0.990005 & 0.979996 & -1.969967 \\ 0 & -8.4191 \cdot 10^{-5} & 1.5535 \cdot 10^{-5} \end{array}\right]$$
$$\longrightarrow \tilde{\vec{x}} = \begin{pmatrix} -1.8072 \\ -0.18458 \end{pmatrix}$$

Es ist

$$A^{-1} = \begin{pmatrix} -9.7 \cdot 10^3 & -9.8 \cdot 10^3 \\ -9.8 \cdot 10^3 & -9.9 \cdot 10^3 \end{pmatrix}$$

und daher gilt

$$\begin{aligned} ||A||_\infty &= 1.97 \\ ||A^{-1}||_\infty &= 1.97 \cdot 10^4 \\ cond_{||.||_\infty}(A) = ||A||_\infty \, ||A^{-1}||_\infty &= 3.8809 \cdot 10^4. \end{aligned}$$

Nach der Fehlerformel oben gilt:

$$\begin{aligned} \frac{||\tilde{\vec{x}} - \vec{x}||_\infty}{||\vec{x}||_\infty} &\leq cond_{||.||_\infty}(A) \left(\frac{||\tilde{\vec{b}} - \vec{b}||_\infty}{||\vec{b}||_\infty} + \frac{||\tilde{A} - A||_\infty}{||A||_\infty} \right) \frac{1}{1 - cond_{||.||_\infty}(A) \frac{||\tilde{A} - A||_\infty}{||A||_\infty}} \\ &= 3.8809 \cdot 10^4 \cdot \left(\frac{3.5 \cdot 10^{-5}}{1.97} + \frac{9 \cdot 10^{-6}}{1.97} \right) \cdot \frac{1}{1 - 3.8809 \cdot 10^4 \cdot \frac{9 \cdot 10^{-6}}{1.97}} \\ &\leq 1.0014 \end{aligned}$$

Der tatsächlich aufgetretene relative Fehler ist

$$\frac{||\tilde{\vec{x}} - \vec{x}||_\infty}{||\vec{x}||_\infty} = 0.8155$$

Also eine ganz realistische Aussage.

4.5.1 Rundungsfehlereinfluß beim Gauß-Algorithmus:

Wir nehmen an, daß der Gauß-Algorithmus (aus Abschnitt 5.2) mit Pivottechnik mit r-stelliger Gleitpunktarithmetik ausgeführt wird. ε bezeichne den elementaren Rundungsfehler (dezimal: $\frac{1}{2} \cdot 10^{-r} = \varepsilon$). Dann kann man zeigen, daß für die berechnete Lösung $\vec{x}_\varepsilon$ folgendes gilt:

Satz 4.5.7. *Es gibt eine Matrix E, so daß $(A+E)\vec{x}_\varepsilon = \vec{b}$, mit*

$$|(E)_{ij}| \leq (2n^3 + 2n^2) \cdot g \cdot \max_{i,j} |a_{ij}| \cdot \varepsilon$$

wobei

$$g = \frac{\max\limits_{k \leq i,j \leq n} |a_{ij}{}^{(k)}|}{\max\limits_{1 \leq i,j \leq n} |a_{ij}|}$$

d.h. g hängt von der Pivotstrategie ab! □

Bemerkung 4.5.4. *Die Pivotstrategien haben den Sinn, die Grösse g im vorstehenden Satz unabhängig von A beschränkt zu halten. Es gilt $g \leq 2^{n-1}$ bei Spaltenpivotisierung, in der Regel aber ist g (erfahrungsgemäß) sogar ≤ 10. Man beachte, daß man g bei der praktischen Rechnung mitkontrollieren kann.*

Diese Überlegungen zeigen, daß es von grosser Wichtigkeit ist, die Konditionszahl einer Matrix wenigstens grössenordnungsmässig mit geringem Aufwand schätzen zu können. Die Anwendung der formalen Definition wie im vorausgegangenen Beispiel führt ja zu einem Vielfachen des Aufwandes, der für das Gleichungssystem selbst erforderlich ist, insbesondere wenn A dünn besetzt ist. Tatsächlich ist dies möglich. Die Idee besteht darin, künstlich eine zusätzliche rechte Seite zu konstruieren, die $||A^{-1}z||$ bezüglich z mit $||z|| = 1$ nahezu maximiert, wobei die Berechnung ja nur die Lösung von $Ay = z$ erfordert. (Spezialliteratur, "Konditionsschätzer").

4.6 Lineare Ausgleichsrechnung, QR-Zerlegung

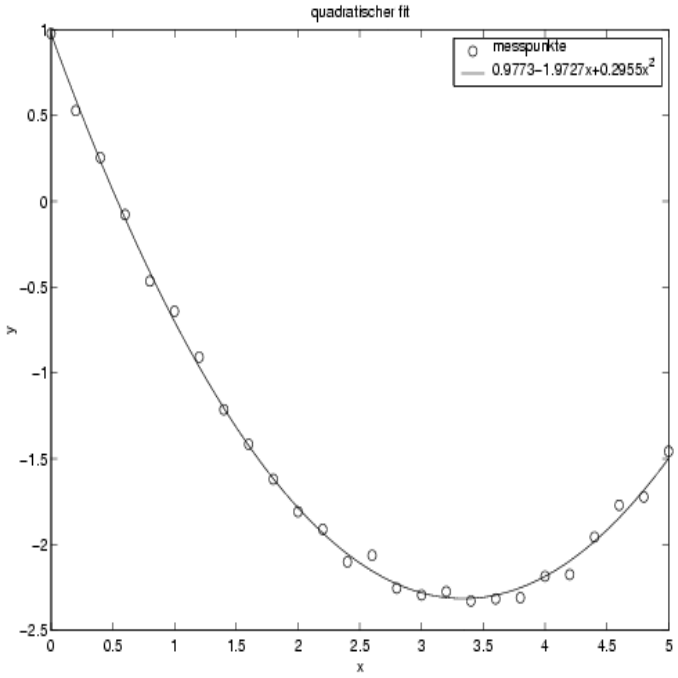
Gegeben: Datenpunkte (t_j, y_j), $j = 1, \ldots, M$ Ferner seien vorgegeben "Ansatzfunktionen" $\varphi_1(t), \ldots, \varphi_n(t)$, z.B. $\varphi_i(t) = t^{i-1}$.

Es sei $M \geq n$. (In der Regel ist $M \gg n$.)

Gesucht sind Koeffizienten $(x_1, \ldots, x_n)$, so daß

$$\left\{ \sum_{j=1}^{M} (y_j - (\sum_{i=1}^{n} x_i \cdot \varphi_i(t_j)))^2 \right\} = \min_{x_1, \ldots, x_n}$$

d.h. die Summe der Abweichungsquadrate der Ordinaten ist zu minimieren.



Mit den Setzungen

$$r_j = y_j - \sum_{i=1}^{n} x_i \varphi_i(t_j) \text{ also } \vec{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_M \end{pmatrix} \text{ Residuenvektor}$$

ist also folgendes Problem zu lösen

$$\sum_{j=1}^{M} {r_j}^2 = \|\vec{r}\|_2^2 \stackrel{!}{=} \min_{x_1,\ldots,x_n} .$$

4.6.1 Lösungsansatz mittels Differentialrechnung: Gauß-sche Normalgleichungen

Man setze

$$s(x_1,\ldots,x_n) := \sum_{j=1}^{M} (y_j - (\sum_{i=1}^{n} x_i \cdot \varphi_i(t_j)))^2 \quad (= \|\vec{r}\|_2^2)$$

(s ist also die "Fehlerquadratsumme"). Das notwendige Extremalkriterium lautet:

$$\frac{\partial}{\partial x_k} s(x_1,\ldots,x_n) = 0 \quad \text{für} \quad k = 1,\ldots,n$$

Mit Hilfe der Kettenregel ergibt sich die partielle Ableitung

$$\frac{\partial}{\partial x_k} s(x_1, \ldots, x_n) \;=\; \sum_{j=1}^{M} 2 \underbrace{(y_j - (\sum_{i=1}^{n} x_i \cdot \varphi_i(t_j)))}_{=r_j} (-\varphi_k(t_j)) = 0 \quad \text{für} \quad k = 1, \ldots, n$$

Setzt man

$$\vec{\varphi}_k \;=\; \begin{pmatrix} \varphi_k(t_1) \\ \vdots \\ \varphi_k(t_M) \end{pmatrix}$$

dann ergibt sich diese partielle Ableitung aus

$$\begin{aligned} -2 \cdot \textstyle\sum_{j=1}^{M} r_j \cdot \varphi_k(t_j) = 0 \quad &| : (-2) \\ \Leftrightarrow \qquad & \sum_{j=1}^{M} r_j \cdot \varphi_k(t_j) = 0 \\ \Leftrightarrow \qquad & \vec{r}^T \cdot \vec{\varphi}_k = 0 \quad \text{für} \quad k = 1, \ldots, n \end{aligned}$$

Wir führen die Matrix aus den Ansatzfunktionen, ausgewertet auf dem Gitter der $t_1, \ldots, t_M$, ein:

$$\Phi \;=\; (\vec{\varphi}_1, \ldots, \vec{\varphi}_n) \in \mathbb{R}^{M \times n} \; .$$

Dann liest sich die notwendige Extremalbedingung als

$$\Phi^T \vec{r} = 0 \in \mathbb{R}^n$$

Mit

$$\vec{y} \;=\; \begin{pmatrix} y_1 \\ \vdots \\ y_M \end{pmatrix} \quad \vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad \text{und } \Phi^T \vec{r} = 0$$

erhalten wir

$$\begin{aligned} \vec{r} \;=\; \vec{y} - \Phi\vec{x} &\Rightarrow (\vec{y} - \Phi\vec{x})^T \Phi = \vec{0} && |()^T \\ &\Leftrightarrow \Phi^T(\vec{y} - \Phi\vec{x}) = \vec{0} && | + \Phi^T\Phi\vec{x} \\ &\Leftrightarrow \boxed{\Phi^T\Phi\vec{x} = \Phi^T\vec{y}} && \text{Gauß'sches Normalgleichungssystem} \end{aligned}$$

Die Matrix $\Phi^T\Phi \in \mathbb{R}^{n \times n}$ ist symmetrisch und $\vec{x} \in \mathbb{R}^n$, $\Phi^T\vec{y} \in \mathbb{R}^n$. Falls die Spalten $\vec{\varphi}_1, \ldots, \vec{\varphi}_n$ von Φ linear unabhängig sind, dann gilt:

$$\Phi\vec{u} = \vec{0} \Leftrightarrow \vec{u} = \vec{0} \; .$$

Dann ist $\Phi^T\Phi$ positiv definit, d.h. $\vec{u}^T(\Phi^T\Phi)\vec{u} > 0 \quad \text{für} \quad \vec{u} \neq \vec{0}$.
Beweis:

$$\begin{aligned} \vec{u}^T(\Phi^T\Phi)\vec{u} &= (\vec{u}^T\Phi^T)(\Phi\vec{u}) = (\Phi\vec{u})^T(\Phi\vec{u}) = \|\Phi\vec{u}\|_2^2 > 0 \\ &\Leftrightarrow \Phi\vec{u} \neq \vec{0}\ \forall \vec{u} \neq \vec{0}\ . \end{aligned}$$

$\vec{x}$ ergibt $\vec{r}$ als "optimales Residuum" mit

$$\Phi^T\vec{r} = \vec{0} \text{ d.h. } \vec{r} \perp \{\vec{\varphi}_1, \ldots, \vec{\varphi}_n\}\ .$$

D.h. das **optimale Residuum ist orthogonal zum Bildraum von** Φ, (der Menge aller Linearkombinationen der Spalten von Φ.). Wir betrachten die 2. Ableitung:

$$\frac{\partial^2}{\partial x_l \partial x_k} s(x_1, \ldots, x_n) = 2\sum_{j=1}^{M} \varphi_l(t_j)\varphi_k(t_j) = 2(\Phi^T\Phi)_{lk}\ , \quad \Phi^T\Phi \text{ positiv definit}$$

d.h. in diesem Fall – $\varphi_1, \ldots, \varphi_n$ linear unabhängig, d.h. Rang $\Phi = n$ – ist auch das hinreichende Optimalkriterium erfüllt.

Zur Erinnerung: Ist $s: \mathbb{R}^n \to \mathbb{R}$, $\nabla s = (\frac{\partial s}{\partial x_k})_{k=1,\ldots,n} = 0$ und $\nabla^2 s = \left(\frac{\partial^2 s}{\partial x_l \partial x_k}\right)_{k,l=1,\ldots,n}$ positiv definit, so ist x eine strenge lokale Minimalstelle von s.

Beispiel:

t	1	2	4	7
y	4.1	6.9	12.8	23

, $\quad M = 4$

$\varphi_1(t_j) = 1, \quad \varphi_2(t_j) = t_j,$ also $n = 2$

$$\begin{aligned} \Phi &= (\varphi_1(t_j), \varphi_2(t_j)) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \\ 1 & 7 \end{pmatrix} \Rightarrow \Phi^T\Phi = \begin{pmatrix} 4 & 14 \\ 14 & 70 \end{pmatrix} \\ \Phi^T\vec{y} &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 7 \end{pmatrix} \begin{pmatrix} 4.1 \\ 6.9 \\ 12.8 \\ 23 \end{pmatrix} = \begin{pmatrix} 46.8 \\ 230.1 \end{pmatrix} \\ \vec{x} &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (\Phi^T\Phi)^{-1}\Phi^T\vec{y} \\ &= \tfrac{1}{84} \begin{pmatrix} 70 & -14 \\ -14 & 4 \end{pmatrix} \begin{pmatrix} 46.8 \\ 230.1 \end{pmatrix} = \begin{pmatrix} 0.65 \\ 3.157 \end{pmatrix} \end{aligned}$$

Die gesuchte Funktion ist also $\underline{3.157t + 0.65}$.

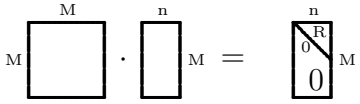
Dieser eigentlich elegante Zugang leidet unter einem Problem: Die Matrix $\Phi^T\Phi$ ist oft sehr schlecht konditioniert. Die Rundungsfehler bei der Aufstellung der Normalgleichungen haben einen großen Einfluß auf die berechnete Lösung.

Wir stellen uns folgende Frage:

Kann man den Cholesky-Faktor L^T der Zerlegung $\Phi^T\Phi = LL^T$ (mit $L =$ ) direkt berechnen, ohne $\Phi^T\Phi$ zu bilden? Die Antwort lautet "ja" und der nächste Abschnitt zeigt den Weg dazu. Man kann dann die schlechte Kondition des Normalgleichungssystems weitgehend vermeiden.

4.6.2 QR-Zerlegung

Annahme: Gegeben sei eine unitäre $M \times M$ Matrix Q, so daß $Q \cdot \Phi = \binom{R}{0} \in \mathbb{R}^{M\times n}$.



Dann folgt

$$\Phi = Q^T \begin{pmatrix} R \\ 0 \end{pmatrix} \text{ , weil } Q \text{ unitär, gilt } Q^{-1} = Q^T$$

und

$$\Phi^T\Phi = (R^T|0) \underbrace{QQ^T}_{I_M} \begin{pmatrix} R \\ 0 \end{pmatrix} = R^T R\,.$$

Bemerkung 4.6.1. *Mit $\Phi^T\Phi = LL^T$ (Cholesky) folgt hieraus nun $R = DL^T$, wobei D eine Diagonalmatrix mit Elementen ± 1 (im Reellen) ist.* □

Definition 4.6.1. *Sei $Q \in \mathbb{R}^{M\times M}$ unitär und $\Phi \in \mathbb{R}^{M\times n}$ mit $M \geq n$. Eine Zerlegung der Form*

$$Q\Phi = \begin{pmatrix} R \\ \cdots \\ 0 \end{pmatrix} \in \mathbb{R}^{M\times n}$$

*mit einer oberen Dreiecksmatrix $R \in \mathbb{R}^{n\times n}$ nennen wir QR-**Zerlegung**. Für $M = n$ gilt $Q\Phi = R$.*

□

Q wird konstruktiv in n Schritten gebildet. Falls $M = n$ benötigt man $n-1$ Schritte. Bei diesem Rechengang wird Q aber nicht explizit aufgestellt, weil dies den Aufwand nur unnötig vergrössern würde. Stattdessen konstruiert man Q als ein Produkt von n bzw. $n-1$ einfachen unitären Matrizen.

Wir wenden uns zunächst dem Spezialfall $n = 1$ zu:

Sei $\Phi = (\vec{\varphi}_1) \in \mathbb{R}^{M\times 1}$. Dann erreichen wir $Q \cdot \Phi = \begin{pmatrix} * \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}$ mit $\begin{array}{rcl} * & = & R \in \mathbb{R}^{1\times 1} \\ |*| & = & \|\vec{\varphi}_1\|_2 \end{array}$

wenn wir Q als geeignete Spiegelung wählen. $Q = I_n - \frac{2}{\vec{u}^T\vec{u}}\vec{u}\vec{u}^T$ beschreibt eine Spiegelung an der Hyperebene H im $\mathbb{R}^M$ mit Normalenvektor $\vec{u}$:

Denn

$$\begin{array}{rrcl} \vec{x} = \lambda\vec{u}: & Q\vec{x} & = & (I_n - \frac{2}{\vec{u}^T\vec{u}}\vec{u}\vec{u}^T)\vec{x} \\ & & = & \vec{x} - \frac{2}{\vec{u}^T\vec{u}}\vec{u}(\vec{u}^T\vec{x}) \\ & & = & \vec{x} - \frac{2}{\vec{u}^T\vec{u}}(\vec{u}^T\vec{x})\vec{u} \\ & & = & \lambda\vec{u} - \frac{2}{\vec{u}^T\vec{u}}(\vec{u}^T\lambda\vec{u})\vec{u} \\ & & = & \lambda\vec{u} - 2\lambda\vec{u} \\ & & = & -\lambda\vec{u} = -\vec{x} \\ \vec{x}\perp\vec{u}: & \vec{u}^T\vec{x} & = & 0 \\ & \Rightarrow Q\vec{x} & = & \vec{x} \end{array}$$

Man bezeichnet diese Matrizen auch als

Householdermatrix:

$$U = I - \frac{2}{\vec{u}^H\vec{u}}\vec{u}\vec{u}^H$$

(benannt nach A.S. Householder, der sie zuerst in diesem Zusammenhang benutzte.)

Zu einer gegebenen Spalte $\vec{x}$ wollen wir nun solch eine Spiegelung, d.h. $\vec{u}$ konstruieren, die diese in ein Vielfaches des ersten Koordinateneinheitsvektors überführt.

Ist $\vec{x}$ gegeben, so kann man ein solches $\vec{u}$ sofort angeben: $x = (x_1, \ldots, x_n)^T$

$$\vec{u} = \begin{pmatrix} (|x_1| + \|\vec{x}\|_2)\sigma \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Dabei ist σ das verallgemeinerte Vorzeichen von x_1:

$$\sigma = \text{sign}_0(z) \stackrel{\text{def}}{=} \begin{cases} 1 & z = 0 \\ z/|z| & \text{sonst.} \end{cases}$$

Beispiel 4.6.1. *Wir nehmen*

$$\vec{x} \;=\; (-8,3,1,5,-1)^T \;.$$

Dann leistet offenbar

$$\vec{u} \;=\; (-18,3,1,5,-1)^T$$

das Gewünschte, denn

$$\vec{u}^T\vec{u} \;=\; 360,\; \vec{u}^T\vec{x} \;=\; 180, \quad \textit{also} \quad \vec{x} - \frac{2}{\vec{u}^T\vec{u}}(\vec{u}^T\vec{x})\vec{u} \;=\; (10,0,0,0,0)^T \;.$$

Diese Methode wird nun systematisch auf Φ angewendet: Die erste Transformation U_1 transformiert die erste Spalte von Φ auf ein Vielfaches des 1. Einheitsvektors. U_1 wird auf alle Spalten von Φ angewendet und auf den Vektor der Messwerte $\vec{y}$. Danach wird die gleiche Vorgehensweise wiederholt, jetzt mit den Komponenten $2,\ldots,M$ der zweiten Spalte von $U_1\Phi$. Dies definiert U_2. Auch U_2 wird auf alle übrigen Spalten von $U_1\Phi$ angewendet und auf $U_1\vec{y}$ usw.

Allgemein lautet der Algorithmus:

QR-Zerlegung :

$$
\begin{aligned}
i &= 1, ..., n'; \qquad n' := \begin{cases} n-1 & \text{falls } m = n \\ n & \text{sonst} \end{cases} \\
j &= i, ..., n \\
\vec{\phi}_i^{(i)} &= \begin{pmatrix} \tilde{\tilde{\phi}}_i^{(i)} \\ \cdots \\ \tilde{\phi}_i^{(i)} \end{pmatrix} \qquad \tilde{\phi}_i^{(i)} \in \mathbb{R}^{m-i+1} \\
\vec{y}^{(i)} &= \begin{pmatrix} \tilde{\tilde{y}}^{(i)} \\ \cdots \\ \tilde{y}^{(i)} \end{pmatrix} \\
\beta_i &:= \frac{2}{\hat{\vec{u}}_i^T \hat{\vec{u}}_i} \\
\hat{\vec{u}}_i^T &:= (\text{sign}_0(\vec{\phi}_{ii}^{\,i})(|\vec{\phi}_{ii}^{\,i}| + \|\tilde{\vec{\phi}}_i^{(i)}\|_2), \vec{\phi}_{i+1,i}^{\,i}, ..., \vec{\phi}_{m,i}^{\,i}) \\
&\quad \text{falls } \tilde{\vec{\phi}}_i^{(i)} \neq 0. \text{ Sonst setze } \beta_i = 0, \quad \hat{\vec{u}}_i = 0, \text{ d.h. } U_i = I \\
\tilde{\tilde{\vec{\phi}}}_j^{(i+1)} &= \tilde{\tilde{\vec{\phi}}}_j^{(i)} \text{ erste } i-1 \text{ Zeilen ändern sich nicht} \\
\tilde{\vec{\phi}}_j^{(i+1)} &= \tilde{\vec{\phi}}_j^{(i)} - \beta_i (\hat{\vec{u}}_i^T \tilde{\vec{\phi}}_j^{(i)}) \hat{\vec{u}}_i \\
\tilde{\tilde{y}}^{(i+1)} &= \tilde{\tilde{y}}^{(i)} \\
\tilde{y}^{(i+1)} &= \tilde{y}^{(i)} - \beta_i (\hat{\vec{u}}_i^T \tilde{y}^{(i)}) \hat{\vec{u}}_i
\end{aligned}
$$

Hier werden im Schritt i die Spalten in zwei Teilspalten zerlegt: Die erste Teilspalte, gekennzeichnet durch die Doppel-Tilde, ändert sich nicht (die ersten $i-1$ Zeilen des Systems bleiben ungeändert) und die zweite Teilspalte (Tilde) wird mit der Householdermatrix multipliziert, wobei deren Struktur explizit ausgenutzt wird.

Beispiel 4.6.2.

$$
\Phi = \begin{pmatrix} -3 & -5 \\ 2 & 2 \\ 2 & -2 \\ 2 & 2 \\ 2 & -2 \end{pmatrix}
$$

Wir setzen

$$
\vec{u}_1 = \begin{pmatrix} -(3+5) \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}
$$

Durch $\vec{u}_1$ wird U_1 gegeben:

$$\begin{aligned} U_1 &= I - \frac{2}{\vec{u}_1^T\vec{u}_1}\cdot\vec{u}_1\vec{u}_1^T \\ U_1\vec{x} &= \vec{x} - \frac{2}{\vec{u}_1^T\vec{u}}(\vec{u}_1\vec{u}_1^T)\vec{x} \\ &= \vec{x} - \frac{2}{\vec{u}_1^T\vec{u}_1}(\vec{u}_1^T\vec{x})\cdot\vec{u}_1 \end{aligned}$$

wobei $\vec{x}$ für die Spalten von Φ steht!

$$\Rightarrow U_1\Phi = \begin{pmatrix} 5 & 3 \\ 0 & 0 \\ 0 & -4 \\ 0 & 0 \\ 0 & -4 \end{pmatrix} \qquad \vec{u}_1^T\begin{pmatrix} -5 \\ 2 \\ -2 \\ 2 \\ -2 \end{pmatrix} = 40 = \vec{u}_1^T\begin{pmatrix} -3 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

$$i=2:\ \hat{\vec{u}}_2 = \begin{pmatrix} \sqrt{32} \\ -4 \\ 0 \\ -4 \end{pmatrix} \Rightarrow \vec{u}_2 = \begin{pmatrix} 0 \\ \sqrt{32} \\ -4 \\ 0 \\ -4 \end{pmatrix} \Rightarrow U_2 = \begin{pmatrix} 1 & 0\,0\,0\,0 \\ 0 & \\ 0 & \hat{U}_2 \\ 0 & \\ 0 & \end{pmatrix}$$

mit $\hat{U}_2 = I - \frac{2}{\hat{u}_2^T\hat{u}_2}\hat{u}_2\hat{u}_2^T$

$$R = \begin{pmatrix} 5 & 3 \\ 0 & -\sqrt{32} \end{pmatrix}, \quad \begin{array}{l} Q = U_2U_1 \text{ implizit gegeben durch.} \\ \vec{u}_1, \vec{u}_2 \end{array}$$

□

Schliesslich hat man

$$U_n\ldots U_1(\Phi\vec{x}-\vec{y}) \;=\; \left(\begin{pmatrix} R \\ O \end{pmatrix}\vec{x} - \begin{pmatrix} \vec{c}_1 \\ \vec{c}_2 \end{pmatrix}\right)$$

worin $\vec{c}_1$ die ersten n Komponenten der transformierten rechten Seite sind. Die Lösung der Ausgleichsaufgabe bestimmt sich dann aus

$$R\vec{x} \;=\; \vec{c}_1$$

und die Länge des optimalen Residuenvektors ist $||\vec{c}_2||_2$. Mit allgemeinen Bezeichnungen haben wir

Satz 4.6.1. QR Zerlegung und Anwendung *Es sei* $A \in \mathbb{R}^{m\times n}$ *mit* $m \geq n$. *Dann existiert eine orthonormale Matrix* $Q \in \mathbb{R}^{m\times m}$ *mit* $QA = \begin{pmatrix} R \\ \cdots \\ 0 \end{pmatrix}$, R $n \times n$ *obere Dreiecksmatrix. Ist* A *vom Rang* n, *dann ist* R *invertierbar und die Aufgabe:*
Bestimme $\vec{x}^*$:

$$\|A\vec{x}^* - \vec{b}\|_2^2 \leq \|A\vec{x} - \vec{b}\|_2^2 \quad \text{für alle } \vec{x} \in \mathbb{R}^n$$

besitzt eine eindeutig bestimmte Lösung $\vec{x}^*$, *die sich aus*

$$R\vec{x}^* = \vec{c}_1$$

errechnet, wo $Q\vec{b} = \begin{pmatrix} \vec{c}_1 \\ \cdots \\ \vec{c}_2 \end{pmatrix}$ *mit* $\vec{c}_1 \in \mathbb{R}^n$.
(R ist in diesem Falle regulär) □

Beispiel 4.6.3.

```
(A,b)=   [  -4     1 ,  4.5 ]
         [   2     2 , -1.0 ]
         [   2     2 ,  2.0 ]
         [   1     1 , -1.5 ] ;
 u_1= [ -9 , 2 , 2 , 1 ]' ;

 U_1 = I - (2/(u_1'u_1))*u_1*u_1' ;

 U_1(A,b) = [ 5.0000    1.0000 , -3.5000 ]
            [ 0.0000    2.0000 ,  0.7778 ]
            [ 0.0000    2.0000 ,  3.7778 ]
            [ 0.0000    1.0000 , -0.6111 ] ;

 u_2= [ 0 , 5 , 2 , 1 ]' ;

 U_2= I - (2/(u_2'*u_2))*u_2*u_2' ;

 U_2*U_1*(A,b) = [ 5.0000    1.0000 , -3.5000 ]
                 [ 0.0000   -3.0000 , -2.8333 ]
                 [ 0.0000    0.0000 ,  2.3333 ]
                 [ 0.0000    0.0000 , -1.3333 ] ;

 x2= (-2.8333)/(-3.0000)= 0.9444 ;
 x1= (-3.5000 - 0.9444)/5.0000 = -0.8889 ;
```

```
Residuenlaenge = 2.6874
```

□

NUMAWWW lineare Gleichungssyteme, QR-Zerlegung

Bemerkung: Die Methode der kleinsten Quadrate ist natürlich nicht auf Messdaten mit einem "freien" Parameter (im Beispiel t) beschränkt. Man kann sie wörtlich auch auf ganz allgemeine Ansätze

$$y_i \;=\; a_1\phi_1(\xi_i,\eta_i,\ldots)+\ldots+a_n\phi_n(\xi_i,\eta_i,\ldots)\;,\; i=1,\ldots N$$

anwenden, wobei $\xi,\eta,\ldots$ die "Messstellen" repräsentieren.

4.7 Zusammenfassung

Das Standardverfahren zur Lösung linearer Gleichunggsysteme ist der Gauss'sche Algorithmus. Um den Einfluss von Rundungsfehlern auf die berechnete Lösung unter Kontrolle zu halten, ist die Anwendung von Pivotisierungsregeln unerlässlich, mit Ausnahme spezieller Matrizen, insbesondere der hermitisch positiv definiten. Dieser Algorithmus erzeugt eine Faktorisierung

$$PA \;=\; LR$$

bzw.

$$PAQ \;=\; LR$$

mit Permutationsmatrizen P und Q (die durch Permutationsvektoren repräsentiert werden) und einer unteren Dreiecksmatrix L mit Diagonale $(1,\ldots,1)$, engl "unit lower triangular", und einer oberen Dreiecksmatrix R. Auf der Diagonalen von R stehen dann die Pivotelemente. Es ist daher

$$\det(A) \;=\; \pm\det R \;=\; \pm\prod_{i=1}^{n}\rho_{i,i}$$

Diese Zerlegung ersetzt die Information über A gleichwertig und erlaubt z.B. die spätere Lösung von Gleichungssystemen mit A bei beliebiger rechter Seite b. Ist A hermitisch und positiv definit, dann kann man zweckmässig die Cholesky-Zerlegung

$$A \;=\; LL^H$$

mit einer unteren Dreiecksmatrix L mit positiven reellen Diagonalelementen verwenden. Der Gauss'sche Algorithmus erlaubt die Berücksichtigung von Besetztheitsstrukturen (Bandstruktur, Hessenbergstruktur, auch "sparsity"). Der Fehlereinfluss bei der

Anwendung dieses Algorithmus oder allgemeiner von Datenfehlern in Matrix und Inhomogenität wird beschrieben durch die sogenannte "Konditionszahl" der Matrix. Wir haben (etwas vergröbert) die Aussage "normrelativer Fehler in der Lösung kleinergleich Summe der normrelativen Fehler in Matrix und rechter Seite, multipliziert mit der Konditionszahl". Die Konditionszahl ist stets grössergleich eins und oft sehr gross gegen eins. Lineare Ausgleichsaufgaben kann man über die Normalgleichungen mit Hilfe der Choleskyzerlegung lösen. Wegen des u.U. sehr verstärkten Fehlereinflusses sollte man aber besser den Weg über die QR-Zerlegung der Ansatzmatrix gehen. Die QR-Zerlegung vermittelt zugleich eine Berechnung von Orthogonalbasen von Bildraum $\mathcal{R}(A)$ und Kern $\mathcal{N}(A^H)$.

Kapitel 5

Lösung nichtlinearer Gleichungen und Gleichungssysteme

5.1 Problemstellung

In diesem Kapitel betrachten wir die **Aufgabenstellung:**

Gegeben ist eine nichtlineare Abbildung $F: \mathbb{R}^n \to \mathbb{R}^n$ bzw. $F: U \subset \mathbb{R}^n \to \mathbb{R}^n$, wobei U das Definitionsgebiet von F ist.

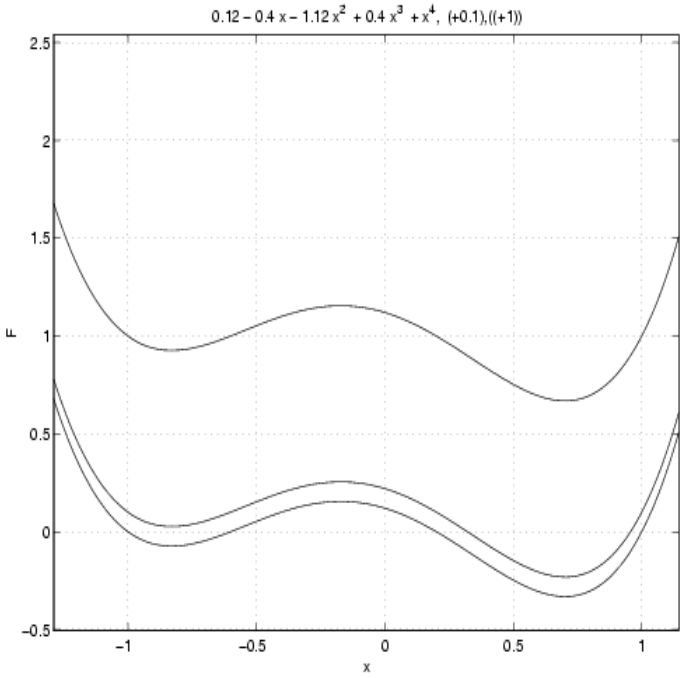
Gesucht ist **ein** $x^* \in \mathbb{R}^n$ bzw. $x^* \in U$ mit $F(x^*) = 0$.

Dieser Aufgabe sind wir bereits bei den impliziten Verfahren zur Lösung nichtlinearer Anfangswertprobleme begegnet. Die Existenz einer Lösung werden wir in der Regel voraussetzen, doch werden wir auch Sätze kennenlernen, bei denen allein aus der "Kleinheit" von $\|F(x_0)\|$ und gewissen Voraussetzungen an die Jacobi–Matrix $\mathcal{J}_F(x)$ auf die Existenz einer Lösung geschlossen werden kann. Einige einfache Beispiele sollen zuerst die möglichen Schwierigkeiten andeuten, mit denen man im nichtlinearen Fall rechnen muß.

Beispiel 5.1.1.

$$n = 1,\ F(x) = 0.12 - 0.4x - 1.12x^2 + 0.4x^3 + x^4$$

Es gibt vier reelle Lösungen $x = \pm 1, 0.2, -0.6$ *für* $F(x) = 0$*, zwei reelle Lösungen für* $F(x) + 0.1 = 0$*, nämlich* $0.9566, 0.3210$ *und keine reelle Lösung für* $F(x) + 1 = 0$*. Die Anzahl der Lösungen ändert sich also unstetig mit den Koeffizienten.*



Beispiel 5.1.2.

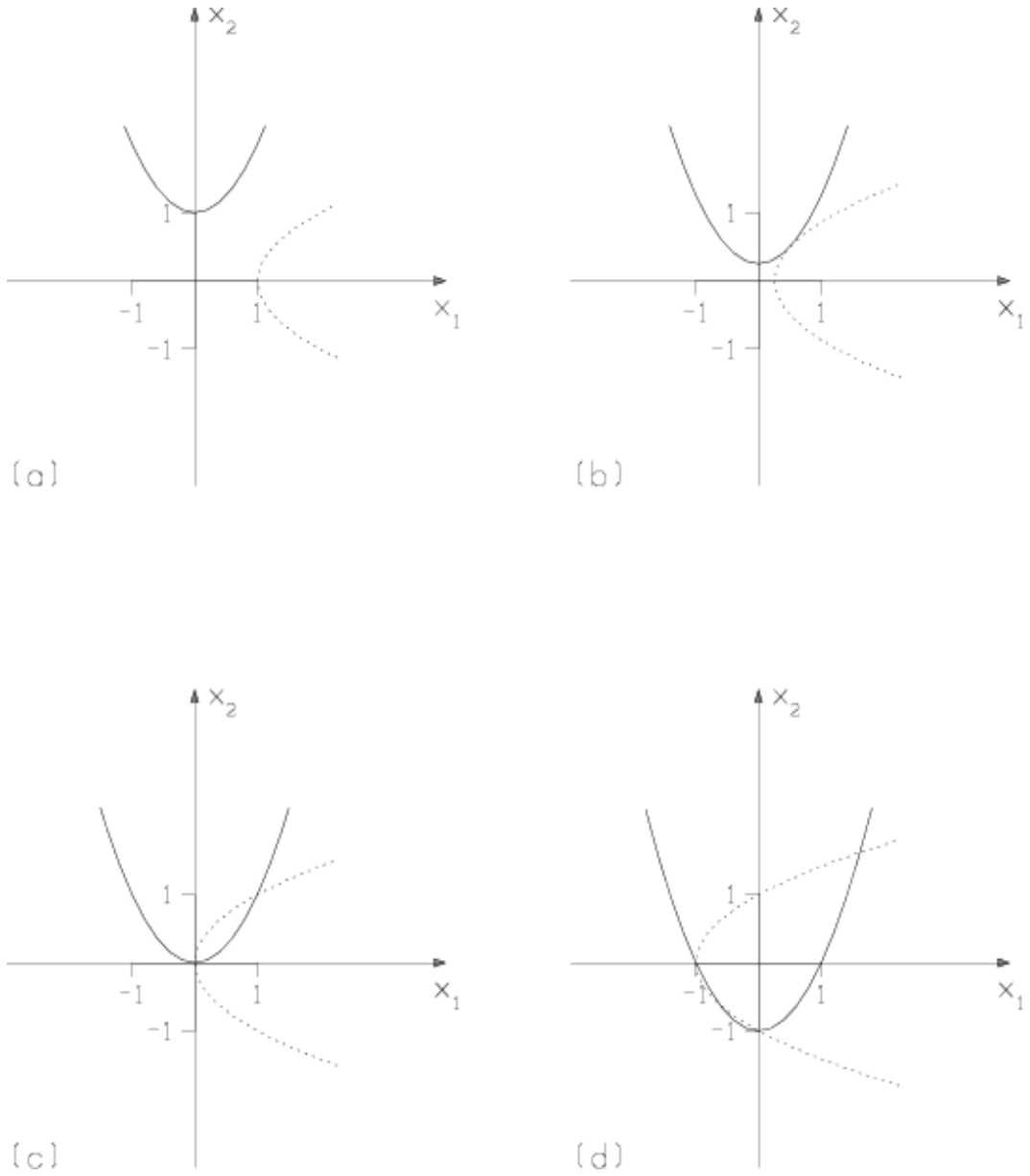
$$n = 2,\ F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \end{pmatrix} = \begin{pmatrix} {x_1}^2 - x_2 + \alpha \\ -x_1 + {x_2}^2 + \alpha \end{pmatrix}$$

(a) $\alpha = 1$*: keine Lösung.*

(b) $\alpha = \frac{1}{4}$*: genau eine Lösung* $x_1 = x_2 = \frac{1}{2}$*.*

(c) $\alpha = 0$*: zwei Lösungen* $x_1 = x_2 = 0$ *und* $x_1 = x_2 = 1$*.*

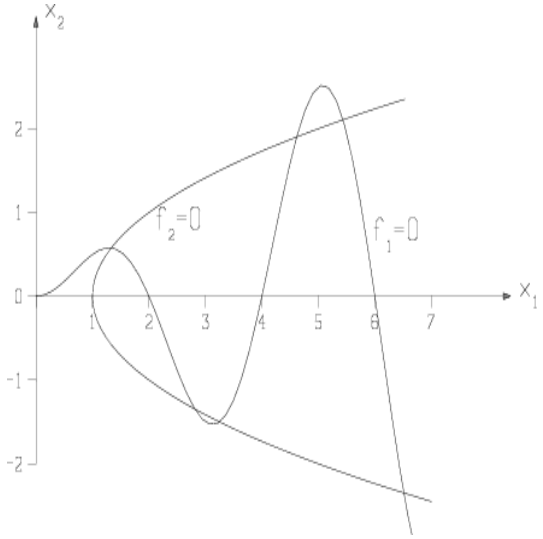
(d) $\alpha = -1$*: vier Lösungen* $x_1 = -1$,$x_2 = 0$ *und* $x_1 = 0$,$x_2 = 1$ *sowie* $x_1 = x_2 = \frac{1}{2}(1 \pm \sqrt{5})$*.*



Beispiel 5.1.3.

$$n = 2,\ F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} x_1 \sin(\frac{1}{2}\pi x_1) - x_2 \\ x_2^2 - x_1 + 1 \end{pmatrix}$$

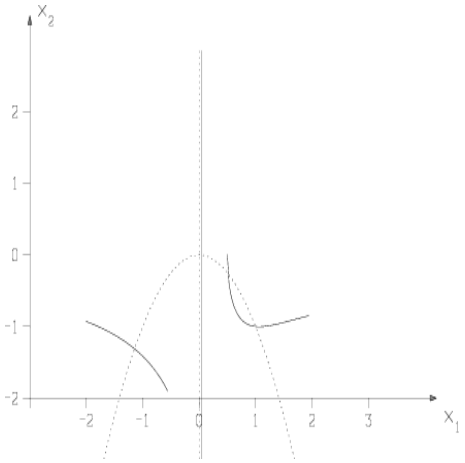
$F(x) = 0$ besitzt in $\mathbb{R}^2$ abzählbar unendlich viele Lösungen.



Beispiel 5.1.4.

$$n = 2,\ F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \end{pmatrix} = \begin{pmatrix} \ln(2x_1^2 - x_1 + 1) - x_1^2 x_2^2 \ln(2) \\ x_2 x_1 + x_1^3 \end{pmatrix}$$

Es gibt hier drei isolierte Lösungen $x_1 = 1, x_2 = -1$*;* $x_1 = 0.512307608\ldots, x_2 = -0.26245908\ldots$ *und* $x_1 = -1.14497278, x_2 = -1.3109626$ *sowie das Lösungskontinuum* $x_1 = 0$*,* $x_2 \in \mathbb{R}$*.*



Im allgemeinen ist also weder die Existenz noch die Anzahl eventueller Lösungen bekannt.

Ein einfaches Anwendungsbeispiel:

Beispiel 5.1.5. *Ein an den Enden gelenkig gelagerter Balken der Länge* l *sei einer ebenen Belastung ausgesetzt.* EI *sei die Biegesteifigkeit und* $M(x)$ *das Biegemoment.*

Dann lautet das Randwertproblem zur Berechnung der Durchbiegung $y(x)$

$$y''(x) \;=\; -\frac{M(x)}{EI}(1+(y')^2)^{3/2},\; 0 \;<\; x \;<\; l\;, y(0)=y(l)=0\;.$$

Mit den Ersetzungen

$$y''(x_i) \;\hat{=}\; \frac{y_{i+1}-2y_i+y_{i-1}}{h^2}$$

und

$$y'(x_i) \;\hat{=}\; \frac{y_{i+1}-y_{i-1}}{2h}$$

wo $x_i = ih,\; i=0,\ldots,N+1,\; h=l/(N+1)$ erhalten wir das nichtlineare System in $y_1,\ldots,\; y_N$

$$y_{i+1}-2y_i+y_{i-1}+h^2\frac{M(x_i)}{EI}\Big(1+(\frac{y_{i+1}-y_{i-1}}{2h})^2\Big)^{3/2} \;=\; 0\;, i=1,\ldots,N$$

Beispiel 5.1.6. Parameteridentifikation *Kommen wir zurück zum einleitenden Beispiel des ersten Kapitels. Dort wurde das Schwingungsverhalten eines Feder -Dämpfer -Systems betrachtet bei gegebenen Koeffizienten $M,\; k,\; r_0,\; c$. In Kapitel 3 haben wir Methoden kennengelernt, um diese nichtlineare Differentialgleichung numerisch zu lösen. In der Praxis ist ein bestimmtes Schwingungsverhalten als Wunsch vorgegeben (d.h. die gewünschte Form der Lösung $x_{ref}(t)$ der DGL ist vorgegeben) und nun sollen bei gegebenem M die Parameter $k,\; r_0$ und c so bestimmt werden, daß diese Form auch (möglichst gut) erreicht wird. Dazu wird diese vorgegebene Funktion an N Stellen τ_j mit der Lösung der DGL verglichen, indem man die Fehlerquadratsumme*

$$f(k,r_0,c) \;\stackrel{def}{=}\; \sum_{j=1}^{N}(x_{ref}(\tau_j)-x(\tau_j;k,r_0,c))^2$$

betrachtet. Wir haben hier für $x(t)$ die Abhängigkeit von den variablen Parametern (durch die DGL) explizit kenntlich gemacht. Man muss also die Lösung der DGL zumindest an den Stellen τ_j annähern. Normalerweise benutzt man aber für die Integration ein sehr viel feineres Gitter und sorgt nur dafür, daß das Gitter der τ_j in diesem enthalten ist. Nun minimiert man $f(k,r_0,c)$ bezüglich der Parameter, d.h. man löst das nichtlineare Gleichungssystem

$$F(\vec{x}) \;=\; \nabla f(\vec{x}) \;=\; 0\;, \quad \vec{x} \;=\; (k,r_0,c)^T\;.$$

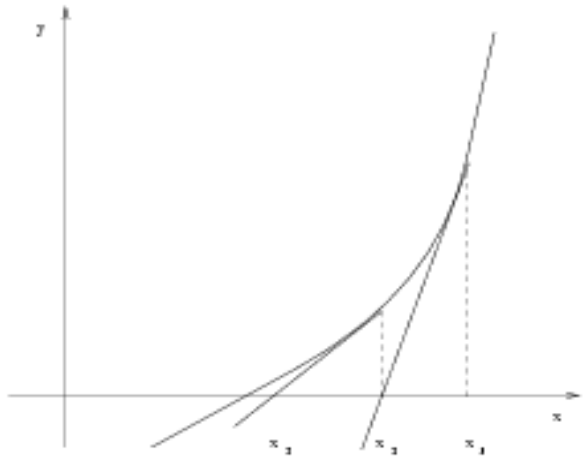
Jede Auswertung von F erfordert also die Integration eines Differentialgleichungssystems, und zwar auch noch die der sogenannten Variationsdifferentialgleichung, das ist das Resultat der Differentiation des DGL-Systems nach den drei Parametern. Letzteres kann man häufig gar nicht explizit leisten und muß dann zur Berechnung des Gradienten auf numerische Differentiation zurückgreifen.

5.2 Das Newton-Verfahren

Anschauliche Herleitung des Newton-Verfahrens für $n = 1$

Sei $x^{(k)}$ eine Näherung für x^* und $x^{(k+1)}$ die Nullstelle der Tangente an $(x, f(x))$ im Punkt $(x^{(k)}, f(x^{(k)}))$
Die Tangentengleichung lautet $y = f(x^{(k)}) + (x - x^{(k)}) \cdot f'(x^{(k)})$ und $x^{(k+1)}$ ist die Lösung von $0 \stackrel{!}{=} f(x^{(k)}) + (x^{(k+1)} - x^{(k)}) \cdot f'(x^{(k)})$.

Unter der Voraussetzung $f'(x^{(k)}) \neq 0$ folgt nun $x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$.



Beispiel 5.2.1. *Für $f(x) = x^2 - a$ folgt als Verfahrensvorschrift*

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

Ist nun $x^{(0)} \geq \sqrt{a}$, dann folgt hier immer monotone Konvergenz und es gilt sogar die sogenannte "quadratische Konvergenz"

$$\frac{|x^{(k+1)} - \sqrt{a}|}{\sqrt{a}} \;\leq\; \tfrac{1}{2}\Big(\frac{|x^{(k)} - \sqrt{a}|}{\sqrt{a}}\Big)^2$$

Grob gesagt verdoppelt sich die Anzahl gültiger Stellen pro Schritt.

Beispiel 5.2.2. *In den folgenden Tabellen ist die Iterationsfolge des Newtonverfahrens zur Lösung der Gleichung*

$$x \exp(x) \;=\; w \;=\; 3 \;, \quad \textit{manchmal bezeichnet als Lambert\_w}(w)$$

für verschiedene Startwerte $x^{(0)}$ angegeben. Man erkennt die typische quadratische Konvergenz und auch, wie stark die Effizienz des Lösungsverfahrens vom Startwert abhängt.

w = 3 , x0 = 0

0.0000000e+00	-3.0000000e+00
3.0000000e+00	5.7256611e+01
2.2873403e+00	1.9527347e+01
1.6841987e+00	6.0746838e+00
1.2641778e+00	1.4754181e+00
1.0801095e+00	1.8092443e-01
1.0505752e+00	3.9050276e-03
1.0499092e+00	1.9340224e-06
1.0499089e+00	4.7517545e-13
1.0499089e+00	4.4408921e-16

w = 3 , x0 = 30

3.0000000e+01	3.2059424e+14
2.9032258e+01	1.1787733e+14
2.8065556e+01	4.3340066e+13
2.7099961e+01	1.5934264e+13
2.6135548e+01	5.8580869e+12
2.5172400e+01	2.1535699e+12
2.4210608e+01	7.9166085e+11
2.3250274e+01	2.9100049e+11
2.2291511e+01	1.0695954e+11
2.1334445e+01	3.9310894e+10
2.0379219e+01	1.4446734e+10
1.9425993e+01	5.3086581e+09
1.8474950e+01	1.9505279e+09
1.7526298e+01	7.1658014e+08
1.6580275e+01	2.6321697e+08
1.5637157e+01	9.6669390e+07
1.4697264e+01	3.5495807e+07
1.3760969e+01	1.3030527e+07
1.2828716e+01	4.7821520e+06
1.1901030e+01	1.7544270e+06
1.0978544e+01	6.4337473e+05
1.0062031e+01	2.3581140e+05
9.1524422e+00	8.6372808e+04
8.2509719e+00	3.1609344e+04
7.3591533e+00	1.1554597e+04
6.4790112e+00	4.2169523e+03
5.6133346e+00	1.5353712e+03
4.7661995e+00	5.5689467e+02

3.9440528e+00	2.0062125e+02
3.1580693e+00	7.1293995e+01
2.4292344e+00	2.4572268e+01
1.7979210e+00	7.8541984e+00
1.3329356e+00	2.0547044e+00
1.1006833e+00	3.0889578e-01
1.0517696e+00	1.0914058e-02
1.0499115e+00	1.5069618e-05
1.0499089e+00	2.8841374e-11
1.0499089e+00	4.4408921e-16

Betrachtung des Falles $n = 2$

Zu lösen ist nun $F(\vec{x}^*) = 0$ mit $F = (F_1, F_2)^T \ : \ \mathbb{R}^2 \to \mathbb{R}^2$. Eine geometrische Vorstellung dazu ist die folgende: $z = F_i(x, y)$ mit reellen $x,\ y$ und z beschreibt eine Fläche im $\mathbb{R}^3$.

1. Fläche: $z = F_1(x_1, x_2)$
2. Fläche: $z = F_2(x_1, x_2)$

$z = 0$ ergibt jeweils die Spur der Flächen in der (x_1,x_2)-Ebene. Gegeben sei eine "Näherungslösung" $\vec{x}^{(0)}$ für $\vec{x}^*$.

In $\vec{x}^{(0)}$ wird jede Fläche durch ihre Tangentialebene ersetzt.

Die Gleichung der Tangentialebene an $z = F_1(x_1, x_2)$ im Punkt $(x_1^{(0)}, x_2^{(0)})$ lautet

$$z \ = \ F_1(x_1^{(0)}, x_2^{(0)}) + \frac{\partial}{\partial x_1} F_1(x_1^{(0)}, x_2^{(0)}) \cdot (x_1 - x_1^{(0)}) + \frac{\partial}{\partial x_2} F_1(x_1^{(0)}, x_2^{(0)}) \cdot (x_2 - x_2^{(0)}) \, .$$

Für die zweite Funktion folgt analog

$$z \ = \ F_2(x_1^{(0)}, x_2^{(0)}) + \frac{\partial}{\partial x_1} F_2(x_1^{(0)}, x_2^{(0)}) \cdot (x_1 - x_1^{(0)}) + \frac{\partial}{\partial x_2} F_2(x_1^{(0)}, x_2^{(0)}) \cdot (x_2 - x_2^{(0)}) \, .$$

(Bemerkung: Formal beschreibt $z = a + b(x - x_0) + c(y - y_0)$ eine Ebene im $\mathbb{R}^3$.)

Wir berechnen die gemeinsame Schnittgerade der beiden Tangentialebenen.

Ihr Durchstoßpunkt mit der Ebene $z = 0$ ist dann der nächste Näherungspunkt.
Mit $\vec{d} := \vec{x}^{(1)} - \vec{x}^{(0)}$ erhält man das Gleichungssystem für $\vec{d}$ in der Form

$$\vec{0} = F(x_1^{(0)}, x_2^{(0)}) + \mathcal{J}_F(\vec{x}^{(0)}) \cdot \vec{d}.$$

wobei $\mathcal{J}_F$ die **Jacobi-Matrix von** F mit den Elementen

$$\boxed{(\mathcal{J}_F(\vec{x}))_{ij} = \tfrac{\partial}{\partial x_j} F_i(\vec{x})}$$

ist, j ist der Spalten- und i der Zeilenindex.

Beispiel 5.2.3. *Das nichtlineare System*

$$\begin{aligned} x_1^3 + x_2 - \tfrac{1}{2} &= 0\ , \\ x_1^2 - x_2^2 &= 0 \end{aligned}$$

hat eine Lösung in der Nähe von $(0.5, 0.5)^T$*. Mit diesem Startwert wird*

$$F_1(\vec{x}^{(0)}) \ = \ \tfrac{1}{8}\ , F_2(\vec{x}^{(0)}) \ = \ 0$$

und

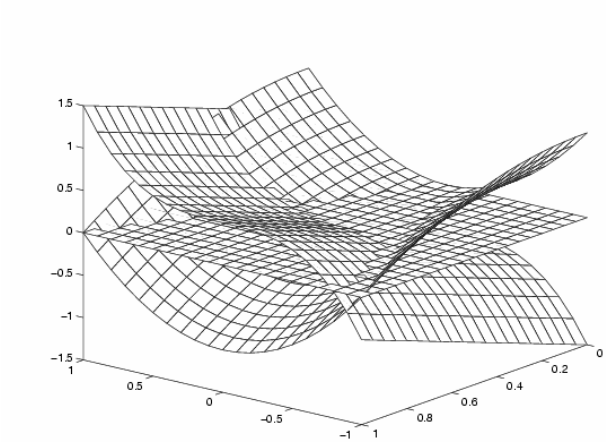
$$\mathcal{J}_F(\vec{x}) \ = \ \begin{pmatrix} 3x_1^2 & 1 \\ 2x_1 & -2x_2 \end{pmatrix}\ , \ \textit{also} \quad \mathcal{J}_F(\vec{x}^{(0)}) \ = \ \begin{pmatrix} \frac{3}{4} & 1 \\ 1 & -1 \end{pmatrix}$$

und somit

$$x_1 \ = \ \tfrac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \ - \ \frac{1}{-\frac{3}{4} - 1}\begin{pmatrix} -1 & -1 \\ -1 & \frac{3}{4} \end{pmatrix}\begin{pmatrix} \frac{1}{8} \\ 0 \end{pmatrix} \ = \ \tfrac{3}{7}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Als neuen F-Wert erhalten wir $F(\vec{x}^{(1)}) \ = \ (.0072886296, 0)^T$.

Die folgende Abbildung zeigt die beiden Flächen F_1 und F_2 zusammen mit der Ebene $z = 0$. Eine Lösung des Systems entspricht dem gemeinsamen Schnittpunkt dieser drei Flächen.



Zusammenfassend lautet das **Newton-Verfahren** für $F(\vec{x}^*) = \vec{0}$ mit $\vec{x}^{(k)}$ als der k-ten Näherung für $\vec{x}^*$

$$\boxed{\begin{array}{l} \mathcal{J}_F(\vec{x}^{(k)}) \cdot \vec{d}^{(k)} = -F(\vec{x}^{(k)}) \text{ zu lösen für } \vec{d}^{(k)} \\ \vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{d}^{(k)}\ . \end{array}}$$

Als formale Voraussetzungen haben wir $F \in C^1(U)$, $\vec{x}^* \in U$, U offen (mit C^1 als Menge der einmal stetig nach allen Variablen partiell differenzierbaren Funktionen) sowie $\mathcal{J}_F(\vec{x})$ invertierbar auf U.

Für theoretische Untersuchungen schreibt man $\vec{x}^{(k+1)}$ als Funktion von $\vec{x}^{(k)}$ in der Form

$$\begin{aligned} \vec{x}^{(k+1)} &= \vec{x}^{(k)} - (\mathcal{J}_F(\vec{x}^{(k)})^{-1}) \cdot F(\vec{x}^{(k)}) \\ &\stackrel{def}{=} \Phi(\vec{x}^{(k)}) \end{aligned}$$

Ein Verfahren dieser Form heißt **direkte Iteration** oder auch **Picard-Iteration**.

Das Nullstellenproblem $F(\vec{x}^*) = 0$ wurde also umgewandelt in ein **Fixpunktproblem** der Gestalt $\vec{x}^* = \Phi(\vec{x}^*)$.

In den Anwendungen hat man oftmals nichtlineare Systeme der Gestalt

$$A(\vec{x})\vec{x} \;=\; \vec{b}$$

mit einer invertierbaren Matrix $A(\vec{x})$. Bei Praktikern beliebt ist dann ein Ansatz der Form

$$A(\vec{x}^{(k)})\vec{x}^{(k+1)} \;=\; \vec{b}$$

sodaß $\vec{x}^{k+1}$ aus einem linearen Gleichungssystem erhalten werden kann. Formal (nicht rechentechnisch) schreibt man dies als

$$\vec{x}^{(k+1)} \;=\; (A(\vec{x}^{(k)})^{-1}\vec{b}$$

und hat wieder die Form der direkten Iteration. Die Konvergenzeigenschaften dieser Lösungszugänge kann man mit Hilfsmitteln des folgenden Abschnitts untersuchen.

5.3 Konvergenzaussagen

In diesem Abschnitt diskutieren wir die Konvergenzbedingungen für Verfahren des Typs

$$\vec{x}^{(k+1)} = \Phi(\vec{x}^{(k)}) \qquad \textbf{(stationäre Einstellenverfahren)}$$

Definition 5.3.1. *Jede Lösung von $\vec{x}^* = \Phi(\vec{x}^*)$ heißt ein* **Fixpunkt** *von Φ.* □

Unter einschränkenden Bedingungen an die Iterationsfunktion Φ kann man Existenz und Eindeutigkeit eines Fixpunktes von Φ sowie die Konvergenz des Iterationsverfahrens für alle Startwerte aus einem gewissen Bereich beweisen. In der Regel ist dieser Bereich eine "Kugel" um einen vorgegebenen Wert.

Satz 5.3.1. Banach'scher Fixpunktsatz, vereinfachte Version: *Es sei $\mathcal{D} \subset \mathbb{R}^n$, $\Phi : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^n$ $\mathcal{D}_0 \subset \mathcal{D}$ sei abgeschlossen und konvex. Φ sei stetig differenzierbar auf $\mathcal{D}$. $\|\cdot\|$ sei eine geeignet gewählte Norm auf $\mathbb{R}^n$. Es gelte*

(i)

$$\sup_{\vec{x}\in\mathcal{D}_0} \|\mathcal{J}_\Phi(\vec{x})\| \stackrel{def}{=} L < 1 \qquad \begin{array}{l}\textit{"Kontraktionseigenschaft"}\\ \textit{L: "Lipschitzkonstante"}\end{array}$$

(ii) *Mit einem geeignet gewählten $\vec{y}^{(0)} \in \mathcal{D}_0$ gelte: Die "Kugel"*

$$\mathcal{K} \stackrel{def}{=} \{\vec{x} : \|\vec{x} - \vec{y}^{(0)}\| \leq \delta\}$$

liege ganz in $\mathcal{D}_0$, wo

$$\delta \stackrel{def}{=} \frac{1}{1-L}||\vec{y}^{(0)} - \Phi(\vec{y}^{(0)})||$$

Dann gilt:

(i) *Es gibt genau einen Fixpunkt $\vec{x}^*$ von Φ in $\mathcal{D}_0$, der sogar in $\mathcal{K}$ liegt.*

(ii) *Für jeden Startwert $\vec{x}^{(0)}$ aus $\mathcal{K}$ konvergiert das Iterationsverfahren gegen diesen Fixpunkt.*

(iii) *Es gilt dabei*

$$||\vec{x}^{(k+1)} - \vec{x}^*|| \leq L||\vec{x}^{(k)} - \vec{x}^*|| \;\forall\; k$$

und

$$||\vec{x}^{(k+1)} - \vec{x}^*|| \leq \frac{L}{1-L}||\vec{x}^{(k+1)} - \vec{x}^{(k)}|| \leq \frac{L^{k+1}}{1-L}||\vec{x}^{(1)} - \vec{x}^{(0)}||$$

□

L beschreibt also die Fehlerreduktion pro Schritt, gemessen in der gewählten Norm. Häufig muss man die Norm erst noch konstruieren, weil die einfachen Normen die Kontraktionsbedingung nicht ergeben. Aus den Sätzen des vorausgegangenen Abschnitts sehen wir auch, daß jedenfalls die Jacobimatrix von Φ im betrachteten Bereich nie-

mals einen Spektralradius grösser als 1 haben darf, weil sonst die Konstruktion der Norm nicht gelingen kann. Natürlich konvergiert das Verfahren in jeder Norm, wenn der Nachweis für eine Norm gelingt. Konkret muß man in der Praxis wie folgt vorgehen:

1. Schritt: Wahl eines möglichst guten Startwertes $\vec{x}^{(0)}$ (durch "sinnvolles Probieren", bei $n = 1$ z.B. Tabellieren von Φ oder unter Ausnutzung analytischer Eigenschaften von Φ).

2. Schritt: Wahl von $\mathcal{D}_0$ als Teil des Definitionsbereiches von Φ. $\mathcal{D}_0$ sollte symmetrisch zu $\vec{x}^{(0)}$ liegen, um L möglichst gut abschätzen zu können (vgl. vorstehenden Satz). Wenn die Konstruktion von $L < 1$ mit $\mathcal{D}_0 = \mathbb{R}^n$ möglich ist, dann ist alles weitere trivial erfüllt mit $\mathcal{D} = \mathcal{D}_0 = \mathbb{R}^n$. Es folgt die Berechnung von L.

3. Schritt: Überprüfung der Voraussetzung an δ mit $\vec{y}^{(0)} = \vec{x}^{(0)}$. Falls sie erfüllt ist, ist alles bewiesen.

Fällt im 2. Schritt $L \geq 1$ aus, dann kann dies zwei Gründe haben:

a) $\mathcal{D}_0$ ist zu groß gewählt worden bzw. $\vec{y}^{(0)}$ ist zu schlechte Näherung.

b) Die Iterationsfunktion Φ ist ungeeignet z.B.:

$$\begin{aligned} x^3 - x - 5 = 0 \;\Leftrightarrow\; (1)\quad & x = x^3 - 5 = \Phi_1(x) \\ (2)\quad & x = \sqrt{3}x + 5 = \Phi_2(x) \\ (3)\quad & x = \frac{5}{x^2 - 1} = \Phi_3(x) \end{aligned}$$

Zur Bestimmung der Lösung bei ≈ 1.9 ist nur die zweite Formel geeignet! Man muß dann eine andere Iterationsvorschrift zu konstruieren versuchen.

Wenn der Test $\mathcal{K} \subset \mathcal{D}_0$ versagt, dann ist normalerweise $\vec{y}^{(0)}$ eine zu schlechte Näherung.

Beispiel 5.3.1. $n = 2$, $\Phi(x_1, x_2) = \begin{pmatrix} 3{x_1}^2 - x_2 + 0.001 \\ \frac{1}{2000}x_1 + 4{x_2}^3 - 0.002 \end{pmatrix}$, $\mathcal{D} = \mathbb{R}^2$. *Vermutung: Ein Fixpunkt liegt nahe bei* 0.

Bilde Jacobi-Matrix:

$$\mathcal{J}_\Phi(x_1, x_2) = \begin{pmatrix} 6x_1 & -1 \\ \frac{1}{2000} & 12{x_2}^2 \end{pmatrix}$$

Wähle $\|\vec{x}\| := \max\{|x_1|; 10|x_2|\}$

$$\begin{aligned} \|\mathcal{J}_\Phi(x_1, x_2)\| &= \left\| \begin{pmatrix} 6x_1 & -\frac{1}{10} \\ \frac{1}{200} & 12{x_2}^2 \end{pmatrix} \right\|_\infty \\ &= \max\{6|x_1| + \tfrac{1}{10}; \tfrac{1}{200} + 12{x_2}^2\} \end{aligned}$$

Wir schätzen

$$\mathcal{D}_0 \;=\; \{\vec{x}:\; |x_1| \le \tfrac{1}{10},\; |x_2| \le \tfrac{1}{5}\}$$

Damit folgt nun

$$\begin{aligned}
L = 0.7 &= \sup_{\vec{x}\in D_0} \|\mathcal{J}_\Phi(\vec{x})\| \\
\vec{y}^{(0)} = 0 \Rightarrow \vec{y}^{(1)} &= \begin{pmatrix} 0.001 \\ -0.002 \end{pmatrix} = \Phi(\vec{y}^{(0)}) \\
\|\vec{y}^{(0)} - \vec{y}^{(1)}\| &= \max\{|-0.001|; 10\cdot|0.002|\} = 0.02 \\
\delta = \frac{1}{1-L}\cdot\|\Phi(\vec{y}^{(0)}) - \vec{y}^{(0)}\| &= \frac{1}{1-0.7}\cdot 0.02 = \frac{0.02}{0.3} = 0.0\bar{6} \\
\mathcal{K} \;\stackrel{def}{=}\; \{\vec{x} : \|\vec{x} - \vec{y}^{(0)}\| &= \max\{|x_1|; 10|x_2|\} \le 0.0\bar{6}\} \;\subset\; \mathcal{D}_0\ .
\end{aligned}$$

Es existiert also genau ein Fixpunkt $\vec{x}^*$ von Φ auf $\mathcal{K}$ und für jeden Startwert aus $\mathcal{K}$ ist $\{\vec{x}^{(k)}\}$ konvergent gegen $\vec{x}^*$. Der Fixpunkt ist sogar in $\mathcal{D}_0$ eindeutig. □

Wenn die Existenz eines Fixpunktes schon anderweitig sichergestellt ist, dann kann folgender wesentlich schwächere Satz zur Konvergenzuntersuchung herangezogen werden:

Satz 5.3.2. Satz von Ostrowski:
Sei $\Phi : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^n$ im Fixpunkt $\vec{x}^*$ differenzierbar. Es gelte

$$\varrho := \varrho(\mathcal{J}_\Phi(\vec{x}^*)) < 1$$

Dann existiert ein $\varepsilon_1 > 0$ und eine geeignete Norm $\|\cdot\|$, so daß mit

$$\mathcal{K} \stackrel{def}{=} \{\vec{x} \in \mathbb{R}^n :\; \|\vec{x} - \vec{x}^*\| \le \varepsilon_1\}$$

gilt: $\forall\, \vec{x}^{(0)} \in \mathcal{K} :\;\; \vec{x}^{(i+1)} = \Phi(\vec{x}^{(i)})$ ist wohldefiniert $(\forall i)$ und $\lim\limits_{i\to\infty} \vec{x}^{(i)} = \vec{x}^*$.
Da auf $\mathbb{R}^n$ alle Normen gleichwertig sind bedeutet dies, daß das Verfahren für hinreichend gute Startwerte konvergiert. □

Eine Konvergenzaussage für das Newtonverfahren (im allgemeinen vektorwertigen Fall) kann man mit Hilfe dieses Satzes von Ostrowski herleiten. Wegen der speziellen Struktur des Verfahrens kann man aber auch direkt vorgehen: Die Iterationsfunktion ist

$$\Phi(\vec{x}) = \vec{x} - (\mathcal{J}_F(\vec{x}))^{-1}\cdot F(\vec{x})$$

Wir setzen hier voraus, daß F zweimal stetig partiell ableitbar ist, daß eine Lösung $\vec{x}^*$ des Nullstellenproblems existiert und daß dort $\mathcal{J}_F(\vec{x})$ invertierbar ist. Wir betrachten nun die Differenz

$$\vec{x}^{(k+1)} - \vec{x}^* \;=\; \Phi(\vec{x}^{(k)}) - \Phi(\vec{x}^*) \quad (\diamond)$$

und wollen zeigen, daß für $\vec{x}^{(0)} - \vec{x}^*$ genügend klein $\vec{x}^{(k)} - \vec{x}^* \ \rightarrow \ 0$. Dazu entwickeln wir die rechte Seite von ($\diamond$) nach Taylor bis zum zweiten Glied. Wenn F zweimal stetig differenzierbar ist, dann ist nach unserer Annahme Φ in einer Umgebung von $\vec{x}^*$ einmal stetig differenzierbar und es gilt

$$\Phi(\vec{x}^{(k)}) - \Phi(\vec{x}^*) \ = \ \mathcal{J}_\Phi(\vec{x}^*)(\vec{x}^{(k)} - \vec{x}^*) + o(\|\vec{x}^{(k)} - \vec{x}^*\|)$$

Dabei ist $o(x)$ das Landausymbol klein-o, d.h. $o(x)/x \rightarrow 0$ für $x \rightarrow 0$. Wir müssen also zunächst $\mathcal{J}_\Phi(\vec{x})$ berechnen. Wir erleichtern uns die Arbeit, indem wir die Jacobimatrix spaltenweise berechnen: die $j-$te Spalte der Jacobimatrix ist die partielle Ableitung von Φ nach x_j, d.h.

$$\mathcal{J}_\Phi(\vec{x}) \cdot \vec{e}^{(j)} = \frac{\partial}{\partial x_j}\Phi(\vec{x}) \ .$$

Sei $A(\vec{x}) \in \mathbb{R}^{n\times n}$, $\vec{b}(\vec{x}) \in \mathbb{R}^n$. Wir betrachten nun eine partielle Ableitung des Matrix-Vektorproduktes:

$$\begin{aligned}
\vec{c}(\vec{x}) &= A(\vec{x}) \cdot \vec{b}(\vec{x}), \quad \vec{c}(\vec{x}) \in \mathbb{R}^n \\
c_k(\vec{x}) &= \sum_{l=1}^{n} \underbrace{a_{kl}(\vec{x})}_{\in\mathbb{R}} \cdot \underbrace{b_l(\vec{x})}_{\in\mathbb{R}} \\
\frac{\partial}{\partial x_j} c_k(\vec{x}) &= \sum_{l=1}^{n} \Big(\Big(\frac{\partial}{\partial x_j} a_{kl}(\vec{x}) \Big) \cdot b_l(\vec{x}) + a_{kl}(\vec{x}) \cdot \Big(\frac{\partial}{\partial x_j} b_l(\vec{x}) \Big) \Big) \\
\frac{\partial}{\partial x_j} \vec{c}(\vec{x}) &= \Big(\frac{\partial}{\partial x_j} A(\vec{x}) \Big) \cdot \vec{b}(\vec{x}) + A(\vec{x}) \cdot \Big(\frac{\partial}{\partial x_j} \vec{b}(\vec{x}) \Big) \\
A(\vec{x}) &= \mathcal{J}_F(\vec{x}))^{-1}, \quad \vec{b}(\vec{x}) = F(\vec{x}) \text{ beim Newtonverfahren} \\
\frac{\partial}{\partial x_j}\Phi(\vec{x}) &= \underbrace{\frac{\partial}{\partial x_j}\vec{x}}_{\vec{e}^{(j)}} - \Big(\frac{\partial}{\partial x_j} A(\vec{x}) \Big) \cdot \underbrace{\vec{b}(\vec{x})}_{=F=0} - \underbrace{A(\vec{x})}_{\mathcal{J}_F(\tilde{x})^{-1}} \underbrace{\Big(\frac{\partial}{\partial x_j} \vec{b}(\vec{x}) \Big)}_{\mathcal{J}_F(\vec{x})\vec{e}^{(j)}} = 0 \quad \text{für } \vec{x} = \vec{x}^* \text{ und alle } j
\end{aligned}$$

Falls $F(\vec{x}^*) = 0$ und $\mathcal{J}_F(\vec{x}^*)$ regulär ist, ist also $\mathcal{J}_\Phi(\vec{x}^*)$ die Nullmatrix.
Dies in die obige Taylorentwicklung eingesetzt und die Definiton von $o(.)$ benutzt ergibt sogar die Aussage

$$\lim_{k\rightarrow\infty} \frac{\|\vec{x}^{(k+1)} - \vec{x}^*\|_\infty}{\|\vec{x}^{(k)} - \vec{x}^*\|_\infty} \ \rightarrow \ 0 \ .$$

Folgerung:

Satz 5.3.3. *Sei $F(\vec{x}^*) = 0$ und F zweimal stetig differenzierbar auf einer Umgebung von $\vec{x}^*$. $\mathcal{J}_F(\vec{x}^*)$ sei invertierbar. Dann gibt es eine Umgebung $\mathcal{V}$ von $\vec{x}^*$, so daß das Newton-Verfahren für jedes $\vec{x}^{(0)} \in \mathcal{V}$ gegen $\vec{x}^*$ superlinear konvergiert.*

Bemerkung 5.3.1. *Mit einer anderen Beweistechnik kann man zeigen, daß für zweimal stetig differenzierbares F mit invertierbarer Jacobi-Matrix das Newton-Verfahren bereits lokal quadratisch konvergiert, d.h.*

$$||\vec{x}^{(k+1)} - \vec{x}^*|| \;\leq\; C||\vec{x}^{(k)} - \vec{x}^*||^2 \quad \textit{mit einer geeigneten Konstanten } C$$

Schon einfache Beispiele zeigen, daß das Newton–Verfahren auch dann nicht notwendig für beliebige Startwerte konvergiert, wenn die Gleichung $F(x) = 0$ nur genau eine Lösung besitzt.
(z.B. $F(x) = \text{arctg}x, \quad |\text{arctg}(x_0)| \geq 2|x_0|/(1 + x_0^2)$) Man benötigt also in der Regel tatsächlich gute Startwerte. Eine Ausnahme bilden konvexe und konkave Funktionen ($n = 1$) mit reellen Nullstellen, wenn x_0 grösser als die grösste oder kleiner als die kleinste Nullstelle ist. Ebenso Polynome mit nur reellen Nullstellen, wenn der Startwert ausserhalb des Nullstellenbereiches liegt.

Eine ähnliche Aussage wie in obigem Satz gilt auch für das **vereinfachte Newton-Verfahren:**

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - (\mathcal{J}_F(\vec{x}^{(0)}))^{-1} \cdot F(\vec{x}^{(k)}) \; .$$

Der **Vorteil** dieses Verfahrens liegt darin, daß nur eine einzige Jacobi-Matrix berechnet werden muss. Mittels einer Dreieckszerlegung berechnet man dann

$$\begin{aligned} P \cdot \mathcal{J}_F(\vec{x}^{(0)}) &= L \cdot R \quad \text{(Gauß)} \\ L \cdot (R \cdot \vec{d}^{(k)}) &= -PF(\vec{x}^{(k)}) \\ \vec{x}^{(k+1)} &= \vec{x}^{(k)} + \vec{d}^{(k)} \end{aligned}$$

Der **Nachteil** des vereinfachten Newtonverfahrens ist die langsamere (nur lineare) Konvergenz, wobei die Konvergenzgeschwindigkeit (i.w. beschrieben durch die Lipschitzkonstante L) von der Güte der Startnäherung abhängt. In vielen Anwendungen besitzt man jedoch gute Startwerte und benötigt keine sehr hohe Endgenauigkeit in der Lösung. Dann ist das vereinfachte Newtonverfahren die angemessenere Lösung. (z.B. bei impliziten Integratoren für Anfangswertprobleme steifer Systeme).

Leider ist das Newton-Verfahren in der Regel tatsächlich nur für sehr gute Startwerte $\vec{x}^{(0)}$ konvergent , so daß man sich nach geeigneten Methoden umsehen muß, um die Konvergenz zu "globalisieren". Eine solche, oft mit gutem Erfolg angewandte Methode ist die "Dämpfung" der Korrektur. Statt $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{d}^{(k)}$ wählt man

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \sigma_k \vec{d}^{(k)} \; ,$$

wobei die sogenannte Schrittweite $\sigma_k \in]0, 1]$ so gewählt wird, daß $\|F(\vec{x}^{(j)})\|$ eine streng monoton fallende Folge ist. Genauer verlangt man, daß

$$\|F(\vec{x}^{(j+1)})\|^2 \leq (1 - \delta\sigma_j)\|F(\vec{x}^{(j)})\|^2 \tag{5.1}$$

gilt mit $0 < \delta < \frac{1}{2}$ fest gewählt. Man kann dann z.B. σ_j maximal in der Folge $\{1, \frac{1}{2}, \frac{1}{4}, \cdots\}$ wählen, so daß (5.1) gilt. Die Norm muß dabei so gewählt sein, daß $\|.\|^2$ eine C^2-Funktion ist, z.B.

$$\|\vec{x}\| = \|A\vec{x}\|_2$$

mit festem regulärem A. Für $A = (\mathcal{J}_F(\vec{x}^*))^{-1}$ würde das die (lokal) monotone Abnahme des Fehlers $\|\vec{x}^{(j)} - \vec{x}^*\|_2$ bedeuten. Man kann zeigen, daß diese Modifikation für jedes $\vec{x}^{(0)}$, das die Bedingung

$$\|F(\vec{x})\| \leq \|F(\vec{x}^{(0)}\| \Rightarrow \mathcal{J}_F(\vec{x}) \quad \text{invertierbar}$$

erfüllt, gegen eine Nullstelle von F konvergiert. Ferner wird ab einer gewissen Schrittzahl automatisch $\sigma_j = 1$, d.h. das Verfahren erhält schliesslich die quadratische Konvergenz.

Deuflhard empfiehlt, A variabel als $(\mathcal{J}_F(\vec{x}^{(k)}))^{-1}$ zu wählen. Dafür gilt der Beweis jedoch nicht. In der Praxis hat sich die Vorgehensweise allerdings oft sehr bewährt.

NUMAWWW Nichtlineare Gleichungssysteme

Beispiel 5.3.2. *Zu minimieren sei die Funktion*

$$f(x_1, x_2) = \left(x_1^2 + x_2^2\right)\left(x_1^2 + (x_2 - 1)^2\right).$$

Um Extremstellen von f zu finden, bestimmt man Nullstellen des Gradienten

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1(x_1^2 + (x_2-1)^2) + 2x_1(x_1^2 + x_2^2) \\ 2x_2(x_1^2 + (x_2-1)^2) + 2(x_2-1)(x_1^2 + x_2^2) \end{pmatrix}$$

mit Hilfe des Newton–Verfahrens. Für verschiedene Startwerte

$$\vec{x}^{(0)} \in \{\begin{pmatrix} 0 \\ 0.72 \end{pmatrix}, \begin{pmatrix} 0.1 \\ 0.72 \end{pmatrix}, \begin{pmatrix} 0.1 \\ 0.73 \end{pmatrix}, \begin{pmatrix} 0.2 \\ -0.2 \end{pmatrix}, \begin{pmatrix} 0 \\ -0.2 \end{pmatrix}\}$$

wird das Newtonverfahren durchgeführt. Die Jacobi–Matrix von $F(\vec{x}) = \nabla f(\vec{x})$ ist gegeben durch

$$\mathcal{J}_F(\vec{x}) = \begin{pmatrix} 12x_1^2 + 2(x_2-1)^2 + 2x_2^2 & 4x_1(x_2-1) + 4x_1x_2 \\ 4x_1(x_2-1) + 4x_1x_2 & 4x_1^2 + 2(x_2-1)^2 + 8x_2(x_2-1) + 2x_2^2 \end{pmatrix}.$$

Die Iterationsfolge lautet für die ersten beiden Startwerte

k	$\vec{x}^{(k)}$	$-F(\vec{x}^{(k)})$	$\mathcal{J}_F(\vec{x}^{(k)})$	$\vec{d}^{(k)}$
0	$\begin{pmatrix} 0 \\ 0.72 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0.1774 \end{pmatrix}$	$\begin{pmatrix} 1.1936 & 0 \\ 0 & -0.4192 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.4232 \end{pmatrix}$
1	$\begin{pmatrix} 0 \\ 0.2968 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.1696 \end{pmatrix}$	$\begin{pmatrix} 1.1652 & 0 \\ 0 & -0.5045 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0.3363 \end{pmatrix}$
2	$\begin{pmatrix} 0 \\ 0.6331 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0.1236 \end{pmatrix}$	$\begin{pmatrix} 1.0708 & 0 \\ 0 & -0.7875 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.1570 \end{pmatrix}$
3	$\begin{pmatrix} 0 \\ 0.4761 \end{pmatrix}$			
$\vdots$				
∞	$\begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$			

k	$\vec{x}^{(k)}$	$-F(\vec{x}^{(k)})$	$\mathcal{J}_F(\vec{x}^{(k)})$	$\vec{d}^{(k)}$
0	$\begin{pmatrix} 0.1 \\ 0.72 \end{pmatrix}$	$\begin{pmatrix} -0.1234 \\ 0.1686 \end{pmatrix}$	$\begin{pmatrix} 1.3136 & 0.1760 \\ 0.1760 & -0.3792 \end{pmatrix}$	$\begin{pmatrix} -0.0323 \\ -0.4596 \end{pmatrix}$
1	$\begin{pmatrix} 0.0677 \\ 0.2604 \end{pmatrix}$	$\begin{pmatrix} -0.0845 \\ -0.1802 \end{pmatrix}$	$\begin{pmatrix} 1.2847 & -0.1297 \\ -0.1297 & -0.2925 \end{pmatrix}$	$\begin{pmatrix} -0.0034 \\ 0.6175 \end{pmatrix}$
2	$\begin{pmatrix} 0.0643 \\ 0.8779 \end{pmatrix}$	$\begin{pmatrix} -0.1021 \\ 0.1558 \end{pmatrix}$	$\begin{pmatrix} 1.6208 & 0.1944 \\ 0.1944 & 0.7301 \end{pmatrix}$	$\begin{pmatrix} -0.0915 \\ 0.2378 \end{pmatrix}$
3	$\begin{pmatrix} -0.0272 \\ 1.1156 \end{pmatrix}$			
$\vdots$				
∞	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$			

Für die drei anderen Startwerte ergibt sich

k	$\vec{x}^{(k)}$	$F(\vec{x}^{(k)})$
0	$\begin{pmatrix} 0.1 \\ 0.73 \end{pmatrix}$	$\begin{pmatrix} 0.1252 \\ -0.1721 \end{pmatrix}$
1	$\begin{pmatrix} 0.0807 \\ 0.1898 \end{pmatrix}$	$\begin{pmatrix} 0.1138 \\ 0.1827 \end{pmatrix}$
2	$\begin{pmatrix} -0.1736 \\ -1.1002 \end{pmatrix}$	$\begin{pmatrix} -1.973 \\ -14.984 \end{pmatrix}$
3	$\begin{pmatrix} -0.0987 \\ -0.604 \end{pmatrix}$	$\begin{pmatrix} -0.5836 \\ -4.3191 \end{pmatrix}$
4	$\begin{pmatrix} -0.0468 \\ -0.291 \end{pmatrix}$	$\begin{pmatrix} -0.1643 \\ -1.1954 \end{pmatrix}$
5	$\begin{pmatrix} -0.0155 \\ -0.109 \end{pmatrix}$	$\begin{pmatrix} -0.039 \\ -0.2949 \end{pmatrix}$
$\vdots$		
∞	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	

k	$\vec{x}^{(k)}$	$F(\vec{x}^{(k)})$
0	$\begin{pmatrix} 0.2 \\ -0.2 \end{pmatrix}$	$\begin{pmatrix} 0.642 \\ -0.784 \end{pmatrix}$
1	$\begin{pmatrix} 0.0591 \\ -0.0758 \end{pmatrix}$	$\begin{pmatrix} 0.1382 \\ -0.1956 \end{pmatrix}$
2	$\begin{pmatrix} 0.0077 \\ -0.015 \end{pmatrix}$	$\begin{pmatrix} 0.0158 \\ 0.0315 \end{pmatrix}$
$\vdots$		
∞	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	

k	$\vec{x}^{(k)}$	$F(\vec{x}^{(k)})$
0	$\begin{pmatrix} 0 \\ -0.2 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.672 \end{pmatrix}$
1	$\begin{pmatrix} 0 \\ -0.0623 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.1488 \end{pmatrix}$
2	$\begin{pmatrix} 0 \\ -0.009 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -0.0185 \end{pmatrix}$
$\vdots$		
∞	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	

Beispiel 5.3.3. *Das folgende Beispiel zeigt die Iterationsfolge für das gedämpfte Ne-*

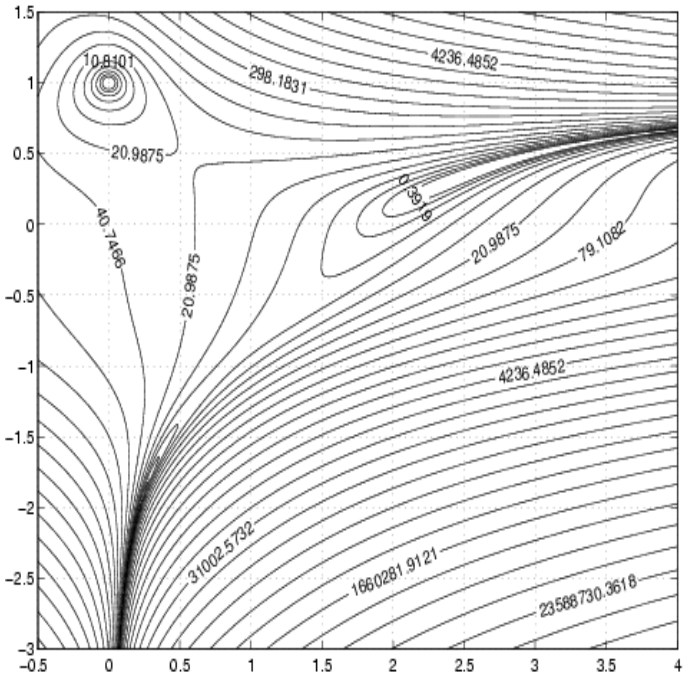
tonverfahren zur Lösung des Nullstellenproblems

$$
\begin{aligned}
f_1(\vec{x}) &= -(1.5 - x_1(1-x_2))(1-x_2) - (2.25 - x_1(1-x_2^2))(1-x_2^2) - \\
&\quad (2.625 - x_1(1-x_2^3))(1-x_2^3) = 0 \\
f_2(\vec{x}) &= (1.5 - x_1(1-x_2))x_1 + (2.250 - x_1(1-x_2^2))2x_1x_2 + \\
&\quad (2.6250 - x_1(1.00 - x_2^3))3.0x_1x_2^2 = 0
\end{aligned}
$$

mit dem Startwert $x^{(0)} = (3, -0.5)$*. Man erkennt, daß zuerst stark gedämpft werden muss und erst in der Schlussphase die schnelle Konvergenz des Verfahrens zum Tragen kommt. Obwohl es hier nur zwei Variablen gibt (mit drei Lösungen) ist dies bereits ein recht schwieriges Testbeispiel, bei dem das Verfahren für schlechtere Startwerte in der Regel völlig versagt. In der hier benutzten Variante des gedämpften Newtonverfahrens (NLEQ1 aus der CODELIB, siehe Kapitel8) wird die Schrittweite* σ *nicht durch die einfache Halbierungsmethode, sondern durch ein Interpolationsverfahren bestimmt.*

```
        it       norm(f)          norm(x)       sigma
         0      0.845E+01        0.319E+00
         1      0.836E+01    *   0.316E+00      0.01000
         2      0.422E+01    *   0.135E+00      0.55071
         3      0.354E+01    *   0.370E+00      0.18228
         4      0.273E+01    *   0.465E+00      0.27921
         5      0.220E+01    *   0.843E+00      0.23244
         6      0.167E+01    *   0.157E+01      0.26868
         7      0.256E+00    *   0.154E+01      0.43700
         8      0.292E+00    *   0.257E+00      1.00000
         9      0.288E-02    *   0.396E-02      1.00000
        10      0.410E-03    *   0.899E-04      1.00000
        11      0.256E-07    *   0.979E-07      1.00000
solution x
              0.3000000000E+01   0.4999999999E+00
function values
 f[ 1]=-0.8398033310730623E-10 f[ 2]= 0.1063837142220906E-09
```

Hier folgt der Höhenlinienplot für $||F(x)||_2^2$.



5.4 Einschachtelungsverfahren

Ein schwerwiegender Nachteil der bisher besprochenen Verfahren ist die normalerweise nur lokale Konvergenz. Ist bei einer reellwertigen stetigen Funktion einer reellwertigen Veränderlichen ein Intervall $[a, b]$ bekannt mit $F(a)F(b) \leq 0$ (d.h. auf Grund des Zwischenwertsatzes enthält $[a, b]$ mindestens eine Nullstelle), dann kann man eine der Nullstellen x^* von F auf $[a, b]$ mit global konvergenten Einschachtelungsverfahren finden. Hierbei wird eine Folge $\{[a_k, b_k]_{k\in\mathbb{N}}\}$ konstruiert mit

$a =: a_0 \leq a_1 \leq a_2 \leq \cdots \qquad \cdots \leq b_2 \leq b_1 \leq b_0 := b$

und $\lim\limits_{k\to\infty} a_k = x^*$ oder $\lim\limits_{k\to\infty} b_k = x^*$.

Das einfachste Verfahren ist die **Intervallhalbierungsmethode** (Bisektion).

$k = 0, 1, 2,$

$$t_k := (a_k + b_k)/2$$

$$a_{k+1} := \begin{cases} a_k & \text{falls } F(a_k)F(t_k) < 0 \\ t_k & \text{sonst} \end{cases} \qquad b_{k+1} := \begin{cases} t_k & \text{falls } F(a_k)F(t_k) \leq 0 \\ b_k & \text{sonst} \end{cases}$$

Hier gilt offensichtlich $|x^* - t_k| \leq 2^{-k-1}(b - a) \quad (\forall k)$

Die folgenden Verfahren unterscheiden sich nur in der Konstruktion des "Testpunktes" t_k.

Bei der **Regula falsi** benutzt man die Nullstelle der Sekante durch
$(a_k, F(a_k))$ $(b_k, F(b_k))$ als neuen Testpunkt t_k, d.h.

$$t_k := a_k - F(a_k)\frac{b_k - a_k}{F(b_k) - F(a_k)}$$

Nachteil der Regula falsi ist, daß eines der Intervallenden gewöhnlich "stehen bleibt", d.h. $a_{k_0} \equiv a_k \quad (\forall k \geq k_0)$ oder $b_{k_0} \equiv b_k \quad (\forall k \geq k_0)$. Die Konvergenzgeschwindigkeit ist oft langsamer als bei der Bisektion. Die folgende Modifikation, der **Illinois–Algorithmus**, schafft hier Abhilfe:

$$t_k := \begin{cases} t_{k-1} - F(t_{k-1})\dfrac{t_{k-1} - t_{k-2}}{F(t_{k-1}) - F(t_{k-2})} & \text{falls } F(t_{k-1})F(t_{k-2}) < 0 \\ t_{k-1} - F(t_{k-1})\dfrac{t_{k-1} - t_{k-3}}{F(t_{k-1}) - F(t_{k-3})/2} & \text{falls } F(t_{k-1})F(t_{k-2}) > 0 \\ & \text{und } F(t_{k-1})F(t_{k-3}) < 0 \\ (a_k + b_k)/2 & \text{sonst} \end{cases}$$

Die Modifikation bewirkt, daß $\lim\limits_{k\to\infty} a_k = \lim\limits_{k\to\infty} b_k = x^*$ und

$$|t_{k+3} - x^*| \leq C|t_k - x^*|^3$$

mit einer geeigneten Konstanten C (die von F', F'', F''' abhängt), falls $F \in C^3 \quad (a, b)$. Der Beweis dieser letzten Abschätzung ist allerdings bereits ziemlich diffizil.

Eine weitere, sehr erfolgreiche Methode beruht auf der Kombination des Einschachtelungsprinzips mit der inversen quadratischen Interpolation. Dies ist das Brent-Decker-Verfahren. In gewissen Ausnahmefällen, die insbesondere bei sehr starker Änderung der Funktionswerte auftreten können (sodaß der Testpunkt zu nahe an einem der "alten" Testpunkte liegt) wird dabei noch auf die Regula falsi bzw. die Bisektion zurückgegriffen.

Beispiel 5.4.1. *Für das Problem der Bestimmung der Lambertfunktion an der Stelle w, das ist die Lösung des Problems*

$$x\exp(x) \;=\; w$$

mit $w = 3$ und $[a, b] = [0, 3]$ ergibt sich mit diesen Verfahren mit dem Abbruchkriterium $b_k - a_k \;\leq\; 10^{-10}$

```
bisektion erfordert     36 funktionswerte
letzte nullstellennaeherung ist t= 0.1049908895103727E+01
mit funktionswert                = 0.8181989244113175E-09
```

```
illinois-algorithmus erfordert     14 funktionswerte
letzte nullstellennaeherung ist t= 0.1049908894964040E+01
mit funktionswert                  = 0.5228022875725102E-15
regula falsi erfordert     97 funktionswerte
letzte nullstellennaeherung ist t= 0.1049908893861318E+01
mit funktionswert                  =-0.6459073855630271E-08
brent decker erfordert     12 funktionswerte
letzte nullstellennaeherung ist t= 0.1049908894957791E+01
mit funktionswert                  =-0.3659975729604359E-10
```

Bemerkung 5.4.1. *Man kennt inzwischen auch brauchbare Übertragungen der Bisektion auf nichtlineare Gleichungssysteme (bedeutsam wegen der globalen Konvergenz) Details siehe z.B. Moore, R.E.;Jones, S.T.: Safe starting regions for iterative methods. SIAM J.Numer. Anal.* $14, (1977),\ 1051 - 1065$ □

5.5 Zusammenfassung

Nichtlineare Gleichungen und Gleichungssysteme haben nicht notwendig eine Lösung und u.U. aber auch viele Lösungen. Die hier dargestellten Methoden begnügen sich sämtlich mit der Bestimmung nur einer Lösung. Alle diese Verfahren sind iterativ. Das am häufigsten angewendete Verfahren, das Newtonverfahren , konvergiert für eine Nullstelle mit regulärer Jacobimatrix lokal superlinear und im Fall einer zweimal stetig differenzierbaren Funktion lokal von zweiter Ordnung. Es erfordert die Kenntnis eines "hinreichend guten" Startwertes. Durch Kontrolle der monotonen Abnahme von $\|F\|$ und deren Erzwingung durch Verkürzung des Korrekturschrittes ("gedämpftes Newtonverfahren") kann man den Konvergenzbereich in der Regel vergrössern. Gelegentlich gelingt es auch auf andere Weise, ein Nullstellenproblem in ein äquivalentes Fixpunktproblem umzuwandeln, auf das der Banach'sche Fixpunktsatz anwendbar ist. Dieser Satz garantiert unter den Bedingungen der Kontraktions- und der Selbsabbildungseigenschaft auf einem Bereich die Existenz und Eindeutigkeit eines Fixpunktes in diesem Bereich. Die Kontraktionsbedingung ist erfüllt, wenn die Jacobimatrix der Iterationsfunktion Φ auf diesem Bereich in einer geeigneten Norm durch eine Konstante kleiner als 1 beschränkt werden kann. Für die Selbstabbildungseigenschaft gibt es Tests, die diese Eigenschaft wenigstens auf einem Teilbereich (der dann notwendig den Fixpunkt enthält) garantieren. Gelegentlich kann man aber die Selbstabbildung auch direkt nachweisen. Eine notwendige, aber nicht hinreichende Bedingung für die Kontraktion ist es, daß die Eigenwerte der Jacobimatrix von Φ betragsmässig stets kleiner als eins sind. Ist die Existenz eines Fixpunktes schon anderweitig gesichert, dann ist die Bedingung "Spektralradius der Jacobimatrix im Fixpunkt kleiner als eins" hinreichend für lokale Konvergenz (Satz von Ostrowski).

Kapitel 6

Elementare Iterationsverfahren für lineare Gleichungssysteme hoher Dimension

6.1 Lineare Systeme: Splittingverfahren

In der Praxis stellt sich oft das Problem, lineare Systeme von sehr großer Dimension lösen zu müssen. In diesen Fällen ist die Koeffizientenmatrix in der Regel auch schwach besetzt (sparse). Aus Rechenzeit- und Speicherplatzgründen ist eine direkte Lösung des Systems dann u.U. nicht sinnvoll, so daß auf iterative Verfahren zurückgegriffen wird.

Zu lösen sei also

$$A\vec{x} = \vec{b}$$

mit einer invertierbaren Matrix $A \in \mathbb{R}^{n\times n}$ sowie der Lösung $\vec{x}^*$.

Wir wählen folgende Interpretation:

Gesucht ist ein Nullstelle $\vec{x}^*$ von $F(\vec{x}) = A\vec{x} - \vec{b}$, d.h.

$$F(\vec{x}) = \vec{0}$$

Dieses Nullstellenproblem wandeln wir nun um in ein Fixpunktproblem der Form

$$\vec{x} = \Phi(\vec{x}).$$

Hier wählen wir $\Phi(\vec{x})$ affin linear, d.h. Φ hat die Gestalt $\Phi(\vec{x}) = G\vec{x} + \vec{g}$.

Es soll also gelten

$$A\vec{x}^* = \vec{b} \Longleftrightarrow \vec{x}^* = G\vec{x}^* + \vec{g}$$

Wir definieren nun die Folge $\{\vec{x}^{(k)}\}$ mit Hilfe der direkten Iteration

$$\vec{x}^{(k+1)} = G\vec{x}^{(k)} + \vec{g} \; . \tag{6.1}$$

Dies alles macht natürlich nur dann einen Sinn, wenn diese Folge gegen den Fixpunkt $\vec{x}^*$ konvergiert. Wegen der Linearität des Problems sind hier sehr viel weitgehendere Aussagen als im vorausgegangenen Kapitel möglich.

Ein hinreichendes und notwendiges Kriterium für die Konvergenz liefert

Satz 6.1.1. *Das Verfahren 6.1 konvergiert genau dann für beliebige $\vec{x}^{(0)}$ gegen einen Fixpunkt $\vec{x}^* = G\vec{x}^* + \vec{g}$, wenn $\varrho(G) < 1$.* □

Beweis:

1. Sei $\varrho(G) \geq 1$. Wähle $\vec{x}^{(0)} = \vec{x}^* + \vec{v}$, wobei $\vec{v}$ Eigenvektor zu einem Eigenwert λ von G ist mit $|\lambda| \geq 1$:

$$\begin{array}{rcl} \vec{x}^{(k+1)} & = & G\vec{x}^{(k)} + \vec{g} \\ \vec{x}^* & = & G\vec{x}^* + \vec{g} \\ \hline \vec{x}^{(k+1)} - \vec{x}^* & = & G(\vec{x}^{(k)} - \vec{x}^*) \end{array}$$

$$\begin{array}{l} \vec{x}^{(0)} = \vec{x}^* + \vec{v} \Rightarrow G(\vec{x}^{(0)} - \vec{x}^*) = G\vec{v} = \lambda \cdot \vec{v} \\ \Rightarrow \vec{x}^1 - \vec{x}^* = G(\vec{x}^{(0)} - \vec{x}^*) = \lambda \cdot \vec{v} \\ \xrightarrow[\text{Induktion}]{\text{vollständige}} \vec{x}^{(k)} - \vec{x}^* = \lambda^k \cdot \vec{v} \end{array}$$

$$\left.\begin{array}{l} |\lambda| \geq 1 \Rightarrow |\lambda^k| \geq 1 \\ \vec{v} \neq \vec{0} \end{array}\right\} \text{ keine Konvergenz!}$$

2. Sei $\varrho(G) < 1$. Dann existiert $\|.\|$ mit $\|G\| < 1$. Da hier die Existenz des Fixpunktes auf Grund der Voraussetzung an A und die äquivalente Umformung in ein Fixpunktproblem schon gegeben ist, können wir abschätzen:

$$||\vec{x}^{(k+1)} - \vec{x}^*|| \; \leq \; ||G||||\vec{x}^{(k)} - \vec{x}^*|| \; \leq \ldots \; ||G||^{k+1}||\vec{x}^{(0)} - \vec{x}^*||$$

woraus die Konvergenz unmittelbar ersichtlich ist.

□

Es stellt sich nun die Frage nach der Konstruktion von G.

Ein erster einfacher Ansatz ist der folgende.

Wir zerlegen die Matrix A additiv in der Form

$$A = \begin{pmatrix} \ddots & & -U \\ & D & \\ -L & & \ddots \end{pmatrix}$$

mit

$$\begin{aligned} D &= \text{diag}\,(a_{ii}) && \text{(Diagonale von } A \text{)} \\ -u_{ij} &= \begin{cases} a_{ij} & \text{falls } j > i \\ 0 & \text{sonst} \end{cases} && \text{(strikter oberer Dreiecksanteil)} \\ -l_{ij} &= \begin{cases} a_{ij} & \text{falls } j < i \\ 0 & \text{sonst} \end{cases} && \text{(strikter unterer Dreiecksanteil)} \end{aligned}$$

Dann ist

$$A\vec{x} = (D - L - U)\vec{x}.$$

Setzen wir zusätzlich noch $a_{ii} \neq 0 \quad \forall i$ voraus, so können wir verschiedene Verfahren herleiten:

1. Ansatz:

$$\begin{aligned} D\vec{x}^{(k+1)} &= (L+U)\vec{x}^{(k)} + \vec{b} \\ \vec{x}^{(k+1)} &= \underbrace{D^{-1}(L+U)}_{G}\vec{x}^{(k)} + \underbrace{D^{-1}\vec{b}}_{\vec{g}} \\ &= G\vec{x}^{(k)} + \vec{g} \end{aligned}$$

   Jacobi- oder Gesamtschrittverfahren

2. Ansatz:

$$\begin{aligned} (-L+D)\vec{x}^{(k+1)} &= U\vec{x}^{(k)} + \vec{b} \\ \vec{x}^{(k+1)} &= (-L+D)^{-1}U\vec{x}^{(k)} + (-L+D)^{-1}\vec{b} \\ &= G\vec{x}^{(k)} + \vec{g} \end{aligned}$$

   Gauß-Seidel- oder Einzelschrittverfahren

3. Ansatz:

$$\vec{x}_i^{(k+1)} \overset{def}{=} \omega\vec{x}_{i,\text{Einzelschritt}}^{(k+1)} + (1-\omega)\vec{x}_i^{(k)}$$

   ergibt vektoriell

$$\begin{aligned} \vec{x}^{(k+1)} &= (-\omega L + D)^{-1}(\omega U + (1-\omega)D)\vec{x}^{(k)} + (-\omega L + D)^{-1}\omega\vec{b} \\ &= G(\omega)\vec{x}^{(k)} + \vec{g}(\omega) \end{aligned}$$

   SOR-Verfahren $(0 < \omega < 2)$

Das Verfahren wird auch als Überrelaxationsverfahren bezeichnet für $\omega > 1$. Der Fall $\omega < 1$ macht in diesem linearen Fall keinen praktischen Sinn. Somit

1.	Gesamtschrittverfahren:	$G = D^{-1}(L+U),\ \vec{g} = D^{-1}\vec{b}$
2.	Einzelschrittverfahren:	$G = (D-L)^{-1}U,\ \vec{g} = (D-L)^{-1}\vec{b}$
3.	SOR-Verfahren:	$G = (D-\omega L)^{-1}(\omega U + (1-\omega)D),\ \vec{g} = \omega(D-\omega L)^{-1}\vec{b}$

Die Iterationsmatrix G des SOR-Verfahrens wird in der Literatur oft mit $B(\omega)$ bezeichnet. Eine explizite Darstellung von G und $\vec{g}$ ist nur für die Theorie von Bedeutung. In der Praxis wird auf folgende Komponentenschreibweise zurückgegriffen.

Wir betrachten die Einträge in der i-ten Zeile des Systems: Beim SOR-Verfahren stehen links die Koeffizienten a_{ii} bzw. $\omega\, a_{ij}$ für $j < i$ und rechts steht $(1-\omega)a_{ii}$, $-\omega\, a_{ij}$ für $j > i$ sowie $\omega\, b_i$.

Allgemein gilt nun

$$\sum_{j=1}^{i-1} \omega a_{ij} \underbrace{x_j^{(k+1)}}_{\text{bereitsbekannt}} + a_{ii} \underbrace{x_i^{(k+1)}}_{\text{unbekannt}} = (1-\omega)a_{ii}x_i^{(k)} + \sum_{j=i+1}^{n} (-\omega a_{ij} \underbrace{x_j^{(k)}}_{\text{bekannt}}) + \omega b_i, \quad i = 1, \ldots, n$$

und es folgt

$$\begin{aligned} x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \tfrac{\omega}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} + a_{ii}x_i^{(k)} - a_{ii}x_i^{(k)}) \\ &= x_i^{(k)} + \tfrac{\omega}{a_{ii}} \underbrace{(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)})}_{=-F_i(x_1^{(k+1)},\ldots,x_{i-1}^{(k+1)},x_i^{(k)},\ldots,x_n^{(k)})} \\ &= x_i^{(k)} - \tfrac{\omega}{a_{ii}} F_i(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, \ldots, x_n^{(k)}) \end{aligned}$$

$F_i(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, \ldots, x_n^{(k)})$ ist der Einsetzfehler von $(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, \ldots, x_n^{(k)})$ in der i-ten Gleichung von $F(\vec{x}) = A\vec{x} - \vec{b} = \vec{0}$. Man hat also zur Berechnung der neuen $i-$ten Komponente den Einsetzfehler der letzten Näherung in der $i-$ten Gleichung zu ermitteln und mit dem Faktor ω/a_{ii} gewichtet vom Wert der laufenden $i-$ten Komponente der Näherung abzuziehen.

Man beachte, daß das SOR-Verfahren mit $\omega = 1$ das Einzelschrittverfahren liefert. Für das Gesamtschrittverfahren rechnet man analog

$$x_i^{(k+1)} = x_i^{(k)} - \tfrac{1}{a_{ii}}(\sum_{j=1}^{n} a_{ij}x_j^{(k)} - b_i) = x_i^{(k)} - \tfrac{1}{a_{ii}} F_i(x^{(k)})$$

Hier kann man also die n Korrekturen unabhängig voneinander, z.B. parallel, berechnen. Im zweidimensionalen erlauben die Verfahren eine einfache graphische Deutung: die neue $i-$te Komponente ist so gewählt, daß die $i-$te Gleichung bei sonst unveränderten übrigen Komponenten exakt erfüllt ist im Falle $\omega = 1$. Für anderes ω muss man die dazu notwendige Änderung mit ω multiplizieren. Beim Gauß-Seidel-Verhran bzw. SOR-Verfahren hat man schon den nächsten "Zwischenpunkt" und nach Durchlauf der n Gleichungen den nächsten Punkt. Beim Jacobi-Verfahren werden erst alle Korrekturen einzeln gebildet und dann gleichzeitig auf den alten Punkt angewendet.

Beispiel 6.1.1. *Gegeben sei das lineare Gleichungssystem $Ax = b$ mit*

$$A = \begin{pmatrix} 5 & -4 \\ 1 & -2 \end{pmatrix} \quad \textit{und} \quad b = \begin{pmatrix} 9 \\ -1 \end{pmatrix}.$$

Mit dem Startvektor $x^{(0)} = (-6, -6)^T$ führen wir jeweils drei Schritte des Jacobi- und des Gauß-Seidel-Verfahrens aus. Die Zerlegung der Matrix A in $D - L - U$ ergibt für das **Jacobi-Verfahren** *die Iterationsvorschrift*

$$\begin{aligned} x^{(k+1)} &= D^{-1}(L+U)x^{(k)} + D^{-1}b \\ &= \begin{pmatrix} \frac{1}{5} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \left(\begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 4 \\ 0 & 0 \end{pmatrix} \right) x^{(k)} + \begin{pmatrix} \frac{1}{5} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 9 \\ -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & \frac{4}{5} \\ \frac{1}{2} & 0 \end{pmatrix} x^{(k)} + \begin{pmatrix} \frac{9}{5} \\ \frac{1}{2} \end{pmatrix} \end{aligned}$$

Damit ergibt sich die Iterationsfolge

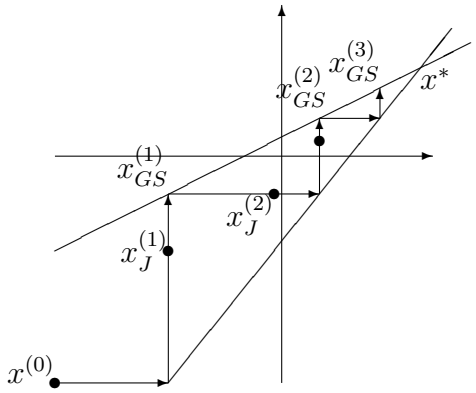
$$\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \quad \begin{pmatrix} -3 \\ -2.5 \end{pmatrix}, \quad \begin{pmatrix} -\frac{1}{5} \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ \frac{2}{5} \end{pmatrix}, \quad \dots$$

Das **Gauß-Seidel-Verfahren** *ist gegeben durch*

$$\begin{aligned} x^{(k+1)} &= D^{-1}(Lx^{(k+1)} + Ux^{(k)} + b) \\ x_1^{(k+1)} &= \tfrac{1}{5}(9 + 4x_2^{(k)}) \\ x_2^{(k+1)} &= \tfrac{1}{2}(1 + x_1^{(k+1)}) \end{aligned}$$

Damit ergibt sich die Iterationsfolge

$$\begin{pmatrix} -6 \\ -6 \end{pmatrix}, \quad \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \frac{13}{5} \\ \frac{9}{5} \end{pmatrix}, \quad \dots$$



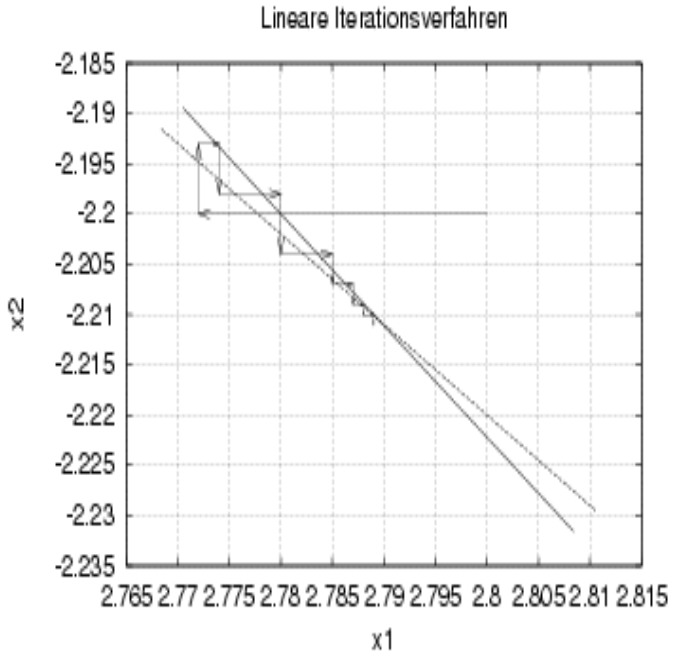
Beispiel 6.1.2. *Hier folgt die Darstellung der Iteration für das SOR-Verfahren mit*

$$A = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 0.8 \\ 0.3 \end{pmatrix} \quad \omega = 1.39286$$

Die Iterationsfolge ist

k	x(1)	x(2)	||r||
0	2.800000	-2.200000	0.2828427E-01
1	2.772143	-2.200000	0.9351765E-02
2	2.772143	-2.192936	0.2493744E-02
3	2.774232	-2.192936	0.3916963E-02
4	2.774232	-2.198330	0.4528347E-02
5	2.780172	-2.198330	0.4176263E-02
6	2.780172	-2.203658	0.3462886E-02
7	2.784518	-2.203658	0.2702011E-02
8	2.784518	-2.207012	0.2027286E-02
9	2.787015	-2.207012	0.1480038E-02
10	2.787015	-2.208825	0.1058975E-02
11	2.788307	-2.208825	0.7460880E-03
12	2.788307	-2.209732	0.5192620E-03
13	2.788936	-2.209732	0.3578318E-03
14	2.788936	-2.210165	0.2445744E-03
15	2.789232	-2.210165	0.1660158E-03
16	2.789232	-2.210365	0.1120303E-03
17	2.789366	-2.210365	0.7521769E-04
18	2.789366	-2.210455	0.5027890E-04
19	2.789427	-2.210455	0.3347840E-04
20	2.789427	-2.210495	0.2221513E-04
21	2.789453	-2.210495	0.1469601E-04
22	2.789453	-2.210513	0.9695118E-05
23	2.789465	-2.210513	0.6380080E-05

24	2.789465	-2.210521	0.4189082E-05
25	2.789470	-2.210521	0.2744849E-05
26	2.789470	-2.210524	0.1795149E-05
27	2.789472	-2.210524	0.1172012E-05
28	2.789472	-2.210525	0.7639617E-06



Dies sind also 14 Schritte, da auch die Zwischenwerte tabelliert sind. Zunächst scheint die Genauigkeitkeit sich sogar (in der Maximumnorm) zu verschlechtern. Das Gauß-Seidel-Verfahren benötigt für die gleiche Genauigkeit bereits 37 Schritte. Für gewisse Matrizen kann das SOR-Verfahren die Konvergenz ganz erheblich beschleunigen, wenn ω optimal gewählt ist.

<<

Im Folgenden sollen **hinreichende Konvergenzkriterien** für die obengenannten Verfahren angegeben werden, und zwar solche, die sich sehr einfach nachprüfen lassen. Dies ergibt Konvergenzaussagen für einige ganz spezielle Matrizenklassen. Die Beweise dieser Sätze sind teilweise sehr kompliziert und tragen nichts zum Verständnis der Verfahren bei. Wir lassen sie daher beiseite (man kann sie in der Spezialliteratur oder in dem Skriptum "Einführung in die Numerische Mathematik" (für Mathematiker) finden.)

Definition 6.1.1. *Eine Matrix A heißt* **strikt diagonaldominant**, *falls*

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \ldots, n \ .$$

Beispiel 6.1.3. $A = \begin{pmatrix} -3 & 1 & 1 \\ 1 & 3 & -1 \\ -1 & 1 & 3 \end{pmatrix}$ *ist strikt diagonaldominant.*

Satz 6.1.2. *Sei A strikt diagonaldominant. Dann konvergieren das Gesamtschritt-, das Einzelschritt- und das SOR-Verfahren für $0 < \omega \leq 1$.* □

Für das Gesamtschrittverfahren gilt im obigen Fall $\|G\|_\infty = \max_i(\frac{1}{|a_{ii}|} \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|) < 1$. Deshalb ist auch der Spektralradius von $G < 1$. Strikte Diagonaldominanz tritt nicht sehr häufig auf. Häufig trifft man aber auf Matrizen des folgenden Typs:

Definition 6.1.2. *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt* **reduzibel**, *wenn es eine Permutationsmatrix P gibt mit*

$$P^T AP = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} \text{ mit quadratischen Matrizen } B_{11}, B_{22}.$$

Ist dies nicht der Fall, so heißt A **irreduzibel**.

Für reduzibles A ist

$$A\vec{x} = \vec{b} \Longleftrightarrow P^T AP \underbrace{P^T\vec{x}}_{\vec{y}} = P^T\vec{b}$$

und das System zerfällt in 2 kleine Systeme.

Definition 6.1.3. *A heißt* **irreduzibel diagonaldominant**, *wenn A irreduzibel ist und zusätzlich gilt*

1. $|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \qquad i = 1, \ldots, n$
2. $\exists i_0 \ : \ |a_{i_0 i_0}| > \sum_{\substack{j=1 \\ j\neq i_0}}^{n} |a_{i_0,j}|$

Satz 6.1.3. *Falls A irreduzibel diagonaldominant ist, so gilt die Aussage von Satz 6.1.2 ungeändert.*

Die direkte Überprüfung auf Irreduzibilität nach der Definition erweist sich als sehr unhandlich, so daß wir ein einfaches hinreichendes und notwendiges **Kriterium für Irreduzibilität**

angeben.

Definition 6.1.4. *Der einer Matrix $A \in \mathbb{R}^{n\times n}$ zugeordnete* **gerichtete Graph** $G(A)$ *ist wie folgt definiert:*

1. *$G(A)$ besteht aus n Knoten P_i. (Man kann z.B. P_i mit a_{ii} identifizieren).*
2. *Eine* **gerichtete Kante** *verbindet P_i mit P_j genau dann, wenn $a_{ij} \neq 0$ (für alle $i, j \in \{1, \ldots, n\}$.).*
3. *Ein* **gerichteter Weg** *ist die Aneinanderfügung gerichteter Kanten.*
4. *$G(A)$ heißt* **zusammenhängend**, *wenn es für jedes Indexpaar (i, j) mit $i \neq j$ einen gerichteten Weg von P_i nach P_j gibt.*

Satz 6.1.4. *Eine Matrix A ist irreduzibel genau dann, wenn der zugehörige gerichtete Graph $G(A)$ zusammenhängend ist.*

Beispiel 6.1.4. $n = 4$

Sei $A = \begin{pmatrix} 1 & 1 & 0 & 2 \\ 0 & 4 & 0 & 1 \\ -1 & 3 & 0 & 8 \\ 2 & 0 & 1 & -7 \end{pmatrix}$. *Der gerichtete Graph $G(A)$ ist zusammenhängend und damit ist A irreduzibel.*

Zum Beispiel ist eine Tridiagonalmatrix mit Nebendiagonalelementen $\neq 0$ stets irreduzibel.

Bisher haben wir Konvergenzresultate für das SOR-Verfahren nur für $0 < \omega \leq 1$. Dies ist aber eigentlich uninteressant, weil in den praxisrelevanten Fällen die Konvergenzgeschwindigkeit für $\omega = 1$ grösser ist als für $\omega < 1$.

Es gilt aber:

Satz 6.1.5. *Sei $A = A^T$ positiv definit. Dann konvergiert das SOR-Verfahren für $0 < \omega < 2$.* □

Andererseits ist klar, daß man von vorneherein eine Einschränkung $0 < \omega < 2$ hat wegen

Satz 6.1.6. *Falls $\omega \in \mathbb{R},\ \omega \notin]0, 2[$, dann divergiert das SOR-Verfahren.* □

Definition 6.1.5. *Eine Matrix A heißt* **L-Matrix**, *falls*

$$a_{ii} > 0, \quad i = 1, \ldots, n, \quad a_{ij} \leq 0 \quad \text{für } i \neq j$$

gilt.
Eine Matrix A heißt **M-Matrix**, *falls*

$$a_{ii} > 0, \quad i = 1, \ldots, n, \quad a_{ij} \leq 0 \quad \text{für } i \neq j.$$

und zusätzlich die Inverse A^{-1} existiert und $(A^{-1})_{ij} \geq 0$ für alle i, j.

Bemerkung 6.1.1. *Die Bezeichnung M-Matrix kommt von "monoton". Sei $0 < \vec{x} \leq \vec{y} \in \mathbb{R}^n$, komponentenweise z.B.*

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \leq \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}$$

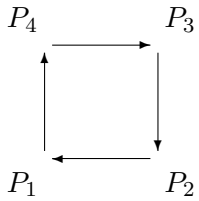
Dann gelten mit den Beziehungen $A\vec{u} = \vec{x},\ A\vec{v} = \vec{y},\ A:$ M-Matrix:

$$\begin{aligned} \vec{y} &\geq \vec{x} \Rightarrow \\ \sum_{j=1}^{n} (A^{-1})_{ij} y_j &\geq \sum_{j=1}^{n} (A^{-1})_{ij} x_j \Rightarrow \\ A^{-1}\vec{y} &\geq A^{-1}\vec{x} \Rightarrow \\ \Rightarrow \vec{v} &\geq \vec{u} \end{aligned}$$

Beispiel 6.1.5. *Im folgenden Beispiel werden an Hand von drei Matrizen die hier definierten speziellen Matrixeigenschaften noch einmal diskutiert:*

1. $$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

   - *Irreduzibilität: Der gerichtete Graph, der der Matrix A zugeordnet werden kann hat die Gestalt*

     Damit ist die Matrix irreduzibel, da es für je zwei beliebige Punkte P_1 und P_2 immer einen Weg von P_1 nach P_2 gibt.

- *Die Matrix A ist* **nicht** *diagonaldominat, und somit auch* **nicht** *strikt oder irreduzibel diagonaldominant, da die Diagonalelemente der Matrix kleiner sind als die Summe der Einträge in der entsprechenden Zeile:*

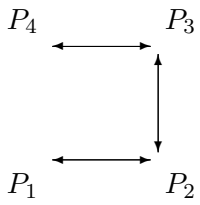
$$0 < 1$$

- *A ist* **weder** *eine L-* **noch** *eine M-Matrix.*

2.

$$B = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

- *Irreduzibilität: Der gerichtete Graph, der der Matrix B zugeordnet werden kann hat die Gestalt*



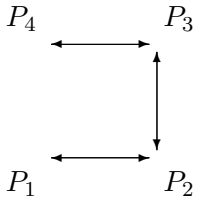
Damit ist die Matrix B irreduzibel.

- *B ist diagonaldominant, wegen* $2 \geq 1 + 1$ *und* $2 > 1$,
- *und sie ist irreduzibel diagonaldominant.*
- *Aber die Matrix B ist* **nicht** *strikt diagonaldominant, da in der 2. Zeile nicht die echte Ungleichung gilt.*
- *B ist eine L-Matrix*
- *Folglich ist B eine M-Matrix.*

3.

$$C = \begin{pmatrix} 2 & -2 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -2 & 2 \end{pmatrix}$$

- *Irreduzibilität: Der gerichtete Graph, der der Matrix C zugeordnet werden kann hat die Gestalt*



  Damit ist die Matrix C irreduzibel.
- *C ist diagonaldominant, wegen $2 \geq 1+1$ und $2 \geq 2$,*
- *aber sie ist* **nicht** *irreduzibel diagonaldominant, weil nie die Ungleichheit gilt.*
- *Aus dem gleichen Grund ist die Matrix C* **nicht** *strikt diagonaldominant.*
- *C ist eine L-Matrix*
- *C ist* **keine** *M-Matrix, da C singulär ist.*

Der folgende Satz liefert eine Charakterisierung der M-Matrizen:

Satz 6.1.7. *A ist eine M-Matrix genau dann, wenn A eine L-Matrix ist und $\varrho(D^{-1}(L+U)) < 1$ gilt, d.h. das Gesamtschrittverfahren konvergiert für $A\vec{x} = \vec{b}$ mit $\vec{b}$ beliebig.* □

Als Folgerung aus diesem Satz ergibt sich sofort

Lemma 6.1.1. *Ist A eine L-Matrix und irreduzibel-diagonaldominant oder eine L-Matrix und strikt diagonaldominant, dann ist A eine M-Matrix.* □

Satz 6.1.8. *Eine symmetrische M-Matrix ist positiv definit.*

Konvergenzaussage für das SOR-Verfahren

Satz 6.1.9. *(Varga) Ist A eine irreduzible M-Matrix, dann ist $\varrho((D-\omega L)^{-1}(\omega U+(1-\omega)D))$ monoton fallend in $0 < \omega \leq \omega_0$ mit $\omega_0 > 1$.* □

Es besteht hier das Problem, daß ω_0 unbekannt ist.

>>

Zusammenfassend erhalten wir folgende Tabelle

Hinreichende Kriterien der Konvergenz der drei Verfahren

Matrix A	Jacobi	Gauß-Seidel	SOR
strikt diagonaldominant	konvergent	konvergent	konvergent für $0 < \omega \leq 1$
irreduzibel diagonaldominant	konvergent	konvergent	konvergent für $0 < \omega \leq 1$
M-Matrix	konvergent	konvergent	konvergent für $0 < \omega \leq 1$
irreduzible M-Matrix	konvergent	konvergent	konvergent für $0 < \omega \leq \omega_0$ mit $\omega_0 > 1$
$A = A^T$ positiv definit	nicht immer konvergent	konvergent	konvergent für $0 < \omega < 2$

Für das SOR-Verfahren stellt sich die Frage, ob man $\varrho((D-\omega L)^{-1}(\omega U+(1-\omega)D))$ als Funktion von ω bestimmen kann. Das ist nur in einem ganz speziellen Fall möglich. Wir gehen hier nicht darauf ein. Die Konvergenzgeschwindigkeit dieser einfachen Verfahren hängt entscheidend von der Eigenwertverteilung von A ab, im symmetrisch positiv definiten Fall von der Konditionszahl in $||.||_2$. Und zwar ist hier

$$\varrho(G) = 1 - const/cond(A)$$

und nur für SOR mit optimalem ω in einem Spezialfall

$$\varrho(G) = 1 - const/sqrt(cond(A))$$

Für sich alleine genommen spielen diese Verfahren heute keine Rolle mehr. Sie bilden aber die Grundlage für die äusserst effizienten sogenannten "Mehrgittermethoden" zur Lösung von linearen Gleichungssystemen, die aus der Diskretisierung von Differential- und Integralgleichungssystemen hervorgehen, deren Dimension und Kondition an die Feinheit des Diskretisierungsgitters gekoppelt ist. Dies beruht darauf, daß sie die hochfrequenten Fehlerkomponenten in $Ax - b$ schnell, die "glatten" Fehlerkomponenten aber nur langsam mit den oben angegebenen konditionsabhängigen Konvergenzfaktoren dämpfen. Für eine Behandlung dieser wichtigen Verfahrensgruppe ist hier nicht der Raum.

6.2 Krylov-Unterraum-Methoden

Der wesentliche Aufwand bei obigen einfachen Iterationsverfahren ist pro Iterationsschritt die Auswertung der Residuen

$$F_i(\vec{x}) \;=\; (A\vec{x}-\vec{b})_i \text{ für } i \;=\; 1,\ldots,n$$

(teilweise mit wechselndem Argument), i.w. pro Iterationsschritt eine Matrix–Vektor–Multiplikation. Die Konvergenz dieser Verfahren ist sehr langsam, im Allgemeinen benötigt man Schrittzahlen $\gg \; n$ für gute Genauigkeit. Deshalb ist klar, daß diese Verfahren nur für dünn besetzte Matrizen Sinn machen, wo eine Matrix-Vektor-Multiplikation wesentlich weniger als n^2 Operationen erfordert. Es gibt eine ganz andere Klasse von Verfahren, bei denen man mit einem Aufwand von n Matrix-Vektor-Multiplikationen und einem damit vergleichbar geringen Zusatzaufwand bereits (in exakter Rechnung) die exakte Lösung des Systems erreicht. Im Fall einer positiv definiten symmetrischen Matrix kann man benutzen, daß

$$A\vec{x}-\vec{b} \;=\; \nabla f(\vec{x}) \text{ mit } f(\vec{x}) \;=\frac{1}{2}\vec{x}^T A\vec{x}-\vec{b}^T\vec{x}$$

ist und kann f auf Unterräumen wachsender Dimension minimieren. Ist V_k eine Basis ($n\times k$-Matrix) der $k-$ten Mannigfaltigkeit (also kommt pro Schritt eine Spalte hinzu), dann lautet die Darstellung der Lösung explizit

$$\vec{x}^{(k)} = \vec{x}^{(0)} - V_k((V_k)^T AV_k)^{-1})V_k^T\nabla f(\vec{x}^{(0)})$$

und man erkennt, daß dies besonders einfach zu berechnen ist, wenn gilt

$$(V_k)^T AV_k \quad \text{diagonal .}$$

Dies führt auf die Idee, f längs sogenannter A-orthogonaler Richtungen zu minimieren.

Definition 6.2.1. *Sei A positiv definit. Ein System von Vektoren $\vec{p}^{(i)},\ i=0,1,\ldots,n-1$ mit $\vec{p}^{(i)}\neq\vec{0}$ für alle i*
heißt A-orthogonal (oder A-konjugiert), falls

$$\vec{p}^{(i)T}A\vec{p}^{(j)}=0 \quad \textit{für } i\neq j.$$

□

Bemerkung 6.2.1. *Mit $A=LL^T$ folgt, daß die Vektoren $L^T\vec{p}^{(i)}$ im üblichen Sinne orthogonal sind.*

$A=LL^T$: Cholesky-Zerlegung.

Bemerkung 6.2.2. *Die Eigenvektoren von A bilden ein Orthogonalsystem, das zugleich A-orthogonal ist.*

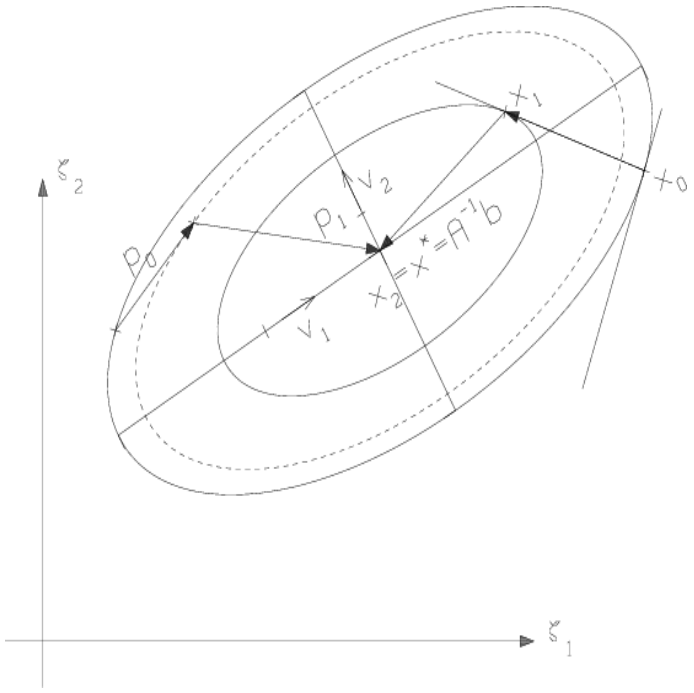
In diesem Zusammenhang gilt

Satz 6.2.1. *Sei* $f(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} - \vec{b}^T\vec{x}$ *und* $\vec{p}^{(0)}, \ldots, \vec{p}^{(n-1)}$ *sei ein A-orthogonales Vektorsystem.* $\vec{x}^{(0)}$ *sei beliebig. Definiert man* $\vec{x}^{(k+1)}$ *durch*

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \sigma_k \vec{p}^{(k)} \quad mit \quad \sigma_k = \frac{(\nabla f(\vec{x}^{(k)}))^T \cdot \vec{p}^{(k)}}{(\vec{p}^{(k)})^T A \vec{p}^{(k)}} \; für \; k = 0, \ldots, n-1$$

dann ist $\nabla f(\vec{x}^{(n)}) = A\vec{x}^{(n)} - \vec{b} = 0$, *d.h.* $\vec{x}^{(n)}$ *löst* $A\vec{x} = \vec{b}$. □

In der folgenden Abbildung sind $p_0,\ p_1$ A-orthogonal, $v_1,\ v_2$ die beiden Eigenvektoren von A sowohl orthogonal wie A-orthogonal und $x_0,\ x_1,\ x_2$ eine vom im Folgenden beschriebenen cg-Verfahren erzeugte Folge.



σ_k in obiger Formel ist die "optimale" Schrittweite, die $\phi(\sigma) \stackrel{def}{=} f(\vec{x}^{(k)} - \sigma\vec{p}^{(k)})$ bezüglich σ minimiert:

Falls $f(\vec{x}^{(k)} - \sigma \cdot \vec{p}^{(k)})$ minimal bzgl. σ, dann gilt:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\sigma} f(\vec{x}^{(k)} - \sigma \cdot \vec{p}^{(k)}) = 0 &= \nabla f(\vec{x}^{(k)} - \sigma \cdot \vec{p}^{(k)})^T \cdot (-\vec{p}^{(k)}) \\ &= (A(\vec{x}^{(k)} - \sigma \cdot \vec{p}^{(k)}) - \vec{b})^T \cdot (-\vec{p}^{(k)}), \end{aligned}$$

also $\sigma = \sigma_k = \frac{(A\vec{x}^{(k)} - \vec{b})^T \cdot \vec{p}^{(k)}}{\vec{p}^{(k)T} \cdot A\vec{p}^{(k)}}$ (> 0 hier nach Konstruktion von $\vec{p}^{(k)}$), also gerade obige Formel.

Nun zur **praktischen Bestimmung der Suchrichtungen** $\vec{p}^{(j)}$:

Wir verfolgen die folgende Idee.

Wir setzen $\vec{p}^{(0)} := \nabla f(\vec{x}^{(0)})$.

Falls $\vec{p}^{(k)}$ schon berechnet ist, dann bestimmt man $\vec{p}^{(k+1)}$ aus dem Ansatz

$$\vec{p}^{(k+1)} = \nabla f(\vec{x}^{(k+1)}) + \sum_{j=0}^{k} \beta_{k+1,j} \cdot \vec{p}^{(j)}$$

und zwar so, daß $\vec{p}^{(k+1)}$ A-orthogonal zu $\vec{p}^{(0)}, \ldots, \vec{p}^{(k)}$ ist. D.h. wir multiplizieren von links mit $(\vec{p}^{(j)})^T A$ und erhalten unter Ausnutzung der $A-$Orthogonalität der schon berechneten $\vec{p}^{(j)}$ die Bedingung

$$\beta_{k+1,j} \;=\; \frac{(\vec{p}^{(j)})^T A \nabla f(\vec{x}^{(k+1)})}{(\vec{p}^{(j)})^T A \vec{p}^{(j)}} \;.$$

Als Resultat erhalten wir unter der Voraussetzung, daß $\vec{x}^{(k+1)}$ wie in Satz 6.2.1 konstruiert ist, nach einiger Rechnung die einfache Rekursion

$$\begin{aligned} \vec{p}^{(0)} &= \nabla f(\vec{x}^{(0)}) \qquad (6.2) \\ \vec{p}^{(k+1)} &= \nabla f(\vec{x}^{(k+1)}) + \frac{\|\nabla f(\vec{x}^{(k+1)})\|_2^2}{\|\nabla f(\vec{x}^{(k)})\|_2^2}\, \vec{p}^{(k)}, \quad k = 0, \ldots, n-1 \end{aligned}$$

(D.h. in obigem Ansatz ergeben sich mit Ausnahme von $\beta_{k+1,k}$ alle $\beta_{k+1,j}$ zu null.)

Die Rekursion bricht ab für $\nabla f(\vec{x}^{(k+1)}) = 0$ ($\Rightarrow \vec{p}^{(k+1)} = 0$). Dann ist man bereits fertig und es gilt $A\vec{x}^{(k+1)} = \vec{b}$. Der Unterraum V_k mit dem man hier arbeitet, hat die Form

$$V_k \;=\; \text{span}\{A^j(A\vec{x}^{(0)} - \vec{b}) \;:\; 0 \;\leq\; j \leq\; k\}$$

Man nennt dies (wegen des Zusammmenhangs mit einem anderen, von Krylov stammenden Verfahren) einen **Krylov-Unterraum**

cg-Verfahren:
Sei $\vec{x}^{(0)}$ beliebig. Für $k = 0, \ldots, n-1$ berechne

1. $\vec{r}^{(k)} := A\vec{x}^{(k)} - \vec{b}$
2. $\delta_k := \|\vec{r}^{(k)}\|^2$ Stop, falls $\delta_k = 0$.
3. $\vec{p}^{(k)} = \begin{cases} \vec{r}^{(k)} & \text{für } k = 0 \\ \vec{r}^{(k)} + \frac{\delta_k}{\delta_{k-1}} \vec{p}^{(k-1)} & \text{für } k > 0 \end{cases}$
4. $\sigma_k = \frac{\vec{r}^{(k)T} \vec{p}^{(k)}}{\vec{p}^{(k)T} A \vec{p}^{(k)}}$
5. $\vec{x}^{(k+1)} = \vec{x}^{(k)} - \sigma_k \cdot \vec{p}^{(k)}$

Beispiel 6.2.1. $n = 2, \quad A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \vec{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\begin{aligned}
\Rightarrow \vec{p}^{(0)} &= \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \delta_0 = 1, \quad \sigma_0 = \tfrac{1}{2}, \quad \vec{x}^{(1)} = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}, \\
A\vec{x}^{(1)} - \vec{b} &= A(\vec{x}^{(0)} - \sigma_0 \cdot \vec{p}^{(0)}) - \vec{b} = \underbrace{A\vec{x}^{(0)} - \vec{b}}_{=\vec{r}^{(0)}} - \sigma_0 \underbrace{A\vec{p}^{(0)}}_{\substack{\text{bereits für} \\ \sigma_0 \text{ berechnet}}} \\
A\vec{x}^{(1)} - \vec{b} &= \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} = \vec{r}^{(1)} \\
\vec{p}^{(1)} &= \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} + \frac{\frac{1}{4}}{1} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{2} \end{pmatrix} \\
A\vec{p}^{(1)} &= \begin{pmatrix} 0 \\ \frac{1}{4} \end{pmatrix} \\
\vec{p}^{(1)T} A \vec{p}^{(1)} &= \tfrac{1}{8}, \quad \sigma_1 = 2 \\
\vec{x}^{(2)} &= \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} - 2 \begin{pmatrix} -\frac{1}{4} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}
\end{aligned}$$

Das Ergebnis des cg-Verfahrens ist ein endlicher Algorithmus

Nachteile des cg-Verfahrens sind:

- die große Rundungsempfindlichkeit, d.h. die berechneten $\vec{p}^{(k)}$ verlieren völlig ihre A-Orthogonalität durch Rundungsfehler.
- Das Verfahren ist sehr konditionsabhängig, sogar abhängig von der Verteilung der Eigenwerte. Eine ungünstige Situation liegt vor, wenn es viele große und nur wenige kleine Eigenwerte gibt.

Die folgende Abbildung zeigt den Verlauf des normierten Fehlers $||\vec{x}^{(k)} - \vec{x}^*||/||\vec{x}^*||$ für eine Matrix der Dimension 100 mit der Konditionszahl 10^5 mit der Darstellung

$$A \;=\; V \mathrm{diag}(\; \alpha \cdot i^2 + \beta \;:\; 1 \le i \le n) V^T$$

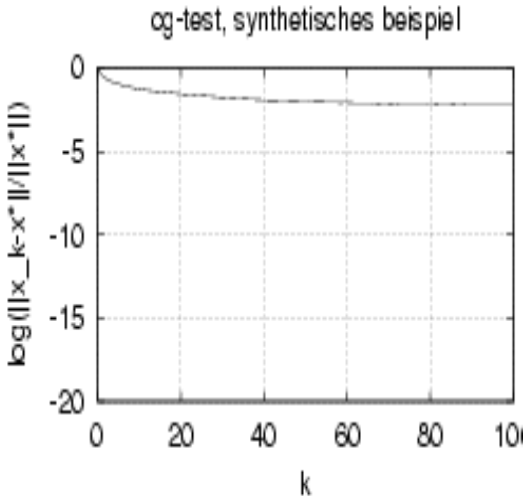
mit der orthonormalen Eigenvektormatrix

$$V_{i,j} \;=\; \sqrt{\tfrac{2}{n+1}} \sin(\tfrac{i \cdot j \cdot \pi}{n+1})$$

und

$$\alpha \;=\; \frac{10^5 - 1}{n^2 - 1}, \quad \beta \;=\; 1 - \alpha$$

und einer künstlich erzeugten rechten Seite, für die die exakte Lösung bekannt ist. Nach $n = 100$ Schritten ist der Fehler keineswegs null geworden, er nimmt vielmehr nur sehr langsam ab.



Man kann diese Nachteile aber durch geeignete Transformation des Gleichungssystems abschwächen, eine sogenannte Präkonditionierung. Wir können hier nicht darauf eingehen und verweisen auf die Spezialliteratur.

NUMAWWW linerare Gleichungssytemen, cg-Verfahren

Leider ist es nicht möglich, ein ähnlich effizientes iteratives Verfahren wie das cg-Verfahren für allgemeine Matrizen anzugeben. Verschiedene Ansätze sind möglich. Ein interessanter Ansatz ist der,

$$f(\vec{x}) \stackrel{def}{=} ||A\vec{x} - \vec{b}||_2^2$$

auf einem Unterraum $\mathcal{V}_k = \text{span}\{A^0(A\vec{x}^{(0)} - \vec{b}), \ldots, A^k(A\vec{x}^{(0)} - \vec{b})\}$ zu minimieren, (das ist das Verfahren der generalisierten minimalen Residuen GMRES), aber dies ergibt keine "kurze" Rekursion für die Korrektur, vielmehr ist der Aufwand von Schritt k $\mathcal{O}(n(k+p))$, wobei p die durchschnittliche Anzahl von Elementen ungleich null in einer Zeile von A ist. Wenn man statt des vollen Krylovunterraums z.B. immer nur die letzten m Vektoren (m fest) zur Definition des Unterraums benutzt, bekommt man Konvergenzprobleme.

6.3 Zusammenfassung

Für spezielle Matrizenklassen kann man iterative Verfahren zur Lösung des linearen Gleichungssystems $A\vec{x} = \vec{b}$ angeben. Diese sind strukturell wesentlich einfacher als die direkten Verfahren und benötigen als zusätzlichen Speicheraufwand nur einige Vektoren der gleichen Länge wie $\vec{x}$. Der wesentliche Aufwand eines Schrittes dieser Verfahren ist stets eine Matrix-Vektor-Multiplikation. Der Nachteil dieser Verfahren besteht in ihrer

oft langsamen, konditionsabhängigen Konvergenzgeschwindigkeit. Durch spezielle, aber problemabhängige Massnahmen kann man die Kondition verbessern. Ein schnelles und universell einsetzbares Verfahren dieser Art gibt es jedoch nicht.

Kapitel 7

Eigenwertprobleme

7.1 Vorbemerkung

Dieses Kapitel beschäftigt sich mit der Lösung folgender **Aufgabenstellung:**

Gesucht ist ein Paar $(\lambda, \vec{x})$ mit $\vec{x} \neq \vec{0}$, so daß gilt: $A\vec{x} = \lambda \cdot \vec{x}$.

Dabei ist A eine reelle oder komplexe $n \times n$-Matrix .

$\vec{x}$ heißt **Eigenvektor** (Rechtseigenvektor) und λ heißt **Eigenwert**. Diese Problemstellung taucht z.B. bei der Lösung von Schwingungsaufgaben auf.

Als **notwendige Bedingung** an die Eigenwerte erhalten wir

$$\boxed{\det\,(A - \lambda I_n) = 0}$$

D.h. wir suchen die Nullstellen eines Polynoms vom Grad n in λ der Form

$$p(\lambda) \;=\; (-1)^n \cdot \lambda^n + (-1)^{n-1} \cdot \lambda^{n-1} \cdot \underbrace{\text{spur } A}_{=\sum\limits_i a_{ii}} + \cdots + \det A\,.$$

Es stellt sich nun die Frage, ob eine numerische Berechnung der Eigenwerte auf obige Art und Weise sinnvoll ist. Dazu betrachten wir folgendes

Beispiel 7.1.1. *Sei $n = 20$ und $\lambda \in \{1, 2, \ldots, 20\}$. Der Koeffizient bei $-\lambda^{19}$ ist $spur A = 210$. Eine Störung um ε in diesem Koeffizienten ergibt nun eine Störung um $\varepsilon \cdot 20^{19}$ bei $\lambda = 20$ im Polynomwert. Um sicherzustellen, daß die Nullstellen des gestörten Polynoms wenigstens noch reell bleiben, ergibt sich als Forderung ungefähr $\varepsilon < 10^{-17}$, Diese Forderung ist in der Praxis schwer erfüllbar, sie verlangt 20-stellige Rechnung.*

Dieses Phänomen gilt allgemein. Die Nullstellen eines Polynoms sind u.U. sehr empfindlich gegen Störungen in den Koeffizienten des Polynoms in der Standarddarstellung. Dagegen kan man leicht zeigen, daß die Eigenwerte einer symmetrischen Matrix unter symmetrischen Störungen sich um nicht mehr als die Norm der Störmatrix ändern. Konsequenterweise suchen wir nach völlig anderen Methoden zur Berechnung von Eigenwerten und Eigenvektoren. Die umgekehrte Vorgehensweise, die Nullstellen eines Polynoms aus den Eigenwerten der zugeordneten Frobeniusbegleitmatrix

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & 0 & 1 & & \vdots \\ & & \ddots & \ddots & 0 \\ & & & \ddots & 1 \\ -\alpha_0 & -\alpha_1 & & & -\alpha_{k-1} \end{pmatrix} .$$

zu bestimmen, ist aber überaus erfolgreich bei Anwendung der QR-Iteration, s.u. Diese Matrix A hat das charakteristische Polynom

$$\psi(\lambda) = det(A - \lambda I) = (-1)^k(\lambda^k + \alpha_{k-1}\lambda^{k-1} + \cdots + \alpha_0) .$$

7.2 Eigenwertabschätzungen

Wir wollen uns hier groben Eigenwerteingrenzungen und der Berechnung von Eigenwertschätzungen zu gegebenen Eigenvektornäherungen beschäftigen. In der Praxis sind meistens Eigenvektornäherungen das primär Gegebene.

Bereits bekannt ist uns die folgende Aussage:

Ist λ ein Eigenwert von A, dann gilt $|\lambda| \leq \|A\|$ für jede einer Vektornorm zugeordnete Matrixnorm. Genauere Aussagen lieferte uns der Satz 3.6.1 Diese Abschätzungen sind jedoch zur genaueren Eingrenzung eher ungeeignet. Seien genäherte Eigenvektoren gegeben, d.h.

$$\vec{x} \approx \vec{u} \quad \text{mit} \quad A\vec{u} = \lambda\vec{u}, \quad \vec{u} \neq \vec{0}$$

dann definieren wir die zugehörigen **Eigenwertnäherungen** durch den sogenannten **Rayleighquotienten**

$$\boxed{R(\vec{x}, A) = \frac{\vec{x}^T A\vec{x}}{\vec{x}^T\vec{x}}}$$

Bemerkung: Im Komplexen setzt man $R(\vec{x}, A) = \frac{\vec{x}^H A\vec{x}}{\vec{x}^H\vec{x}}$.

Eine Fehlerabschätzung erhalten wir aus

Satz 7.2.1. *Sei A eine diagonalähnliche Matrix, d.h. es gebe eine invertierbare Matrix T mit*

$$T^{-1}AT = \text{diag}\,(\lambda_1, \ldots, \lambda_n)\,.$$

Weiterhin sei $\vec{x} \neq \vec{0}$ beliebig, aber $A\vec{x} \neq \vec{0}$. Dann gilt

(i) Es existiert ein Eigenwert $\lambda \neq 0$ von A mit

$$\left|\frac{\lambda - R(\vec{x}, A)}{\lambda}\right| \leq \frac{\|A\vec{x} - R(\vec{x}, A)\cdot\vec{x}\|_2}{\|A\vec{x}\|_2} \cdot \text{cond}_{\,\|.\|_2}(T)$$

(ii) Für $A = A^H$ kann man $\text{cond}_{\,\|.\|_2}(T) = 1$ in (i) wählen

(iii) Ist A hermitisch, d.h. $A = A^H$, mit $A\vec{u} = \lambda\vec{u}$ und $\vec{x}$ als einer Näherung für $\vec{u}$, dann ist

$$|R(\vec{x}, A) - \lambda| \leq 2 \cdot \|A\|_2 \cdot \left\| \frac{\vec{x}}{\|\vec{x}\|_2} - \frac{\vec{u}}{\|\vec{u}\|_2} \right\|_2^2,$$

d.h. der Fehler im Rayleighquotienten ist quadratisch klein in den Fehlern der Eigenvektornäherung.

□

Bemerkungen: Eine hermitische Matrix A ist immer diagonalähnlich (Spektralsatz). T kann dann immer unitär gewählt werden und es ist folglich $\text{cond}_{\,\|.\|_2}(T) = 1$.

Eine beliebige Matrix A mit n verschiedenen Eigenwerten ist ebenfalls diagonalähnlich, hier kann aber $\text{cond}_{\,\|.\|_2}(T) >> 1$ sein. Im nichtdiagonalähnlichen Fall aber ist das Störverhalten der Eigenwerte viel ungünstiger:

Beispiel 7.2.1. *Die Matrix $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ ist nicht diagonalähnlich. Aus $(1-\lambda)^2 = 0$ folgt nämlich $\lambda_1 = \lambda_2 = 1$, aber Rang $(A - \lambda I) = 1$ für alle λ. Die Matrix $B = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}$ ist wegen $(1-\lambda)^2 = \varepsilon \Rightarrow \lambda_{1,2} = \pm\sqrt{\varepsilon} + 1$ aber diagonalähnlich. Eine ε-Störung in der Matrix hat sich hier zu einer $\sqrt{\varepsilon}$-Störung in den Eigenwerten verstärkt. Bei einer diagonalähnlichen Matrix kann dies nicht auftreten.*

7.3 Berechnung von Eigenvektornäherungen

Das **von Mises-Verfahren** wird zur Bestimmung des **betragsdominanten Eigenwertes**, d.h. λ mit $|\lambda| = \varrho(A)$ einer Matrix A eingesetzt. Die Berechnung erfolgt über den zugehörigen Eigenvektor.

Als **Voraussetzungen** formulieren wir:

- A sei diagonalähnlich und es gelte
- $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, wobei λ_i die Eigenwerte von A sind.

Die **Verfahrensvorschrift** für einen geeigneten Startvektor $\vec{x}^{(0)} \neq \vec{0}$ lautet dann für $k = 1, 2, \ldots$:

$$\boxed{\vec{x}^{(k)} = A\vec{x}^{(k-1)}}$$

Mit Normierung von $\vec{x}^{(k)}$ ergibt sich folgender Algorithmus:

Sei $\vec{x}^{(0)} \neq 0$ geeignet mit $\|\vec{x}^{(0)}\|_2 = 1$.

Berechne für $k = 1, 2, \ldots$:

$$\begin{aligned} \tilde{\vec{x}}^{(k)} &= A\vec{x}^{(k-1)} \\ \varrho_{k-1} &= (\vec{x}^{(k-1)})^H \cdot \tilde{\vec{x}}^{(k)} \quad (= R(\vec{x}^{(k-1)}, A)) \\ \vec{x}^{(k)} &= \frac{\tilde{\vec{x}}^{(k)}}{\|\tilde{\vec{x}}^{(k)}\|_2} \end{aligned}$$

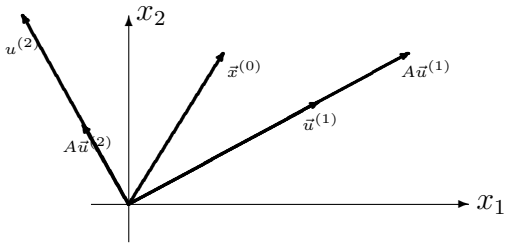
Wir erhalten als **Konvergenzaussage**

Satz 7.3.1. *Sei A diagonalähnlich mit genau einem Eigenwert von maximalem Betrag, d.h. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$.*
$\vec{x}^{(0)}$ habe einen Anteil in Richtung von $\vec{u}^{(1)}$, wobei $\vec{u}^{(1)}$ Eigenvektor zu λ_1 ist, d.h. $A\vec{u}^{(1)} = \lambda_1\vec{u}^{(1)}$. Es gilt also $\vec{x}^{(0)} = \sum_{i=1}^{n} \alpha_i \vec{u}^{(i)}$, $\alpha_1 \neq 0$.
Dann gilt die Konvergenzaussage:

$$\begin{aligned} |\varrho_k - \lambda_1| &\leq const \left|\frac{\lambda_2}{\lambda_1}\right|^k \\ \|v_k \cdot \vec{x}^{(k)} - \vec{u}^{(1)}\|_2 &\leq const \left|\frac{\lambda_2}{\lambda_1}\right|^k \end{aligned}$$

(Vorzeichennormierung: $|v_k| = 1$)

$n =$ 2:



$\vec{x}^{(0)} \neq \alpha_2\vec{u}^{(2)}$ für alle α_2 verlangt!

Die Voraussetzungen an $\vec{x}^{(0)}$ sind "schwach" und werden durch eingeschleppte Rundungsfehler immer erfüllt.

Bemerkung 7.3.1. *Eine Konvergenzaussage gilt auch, wenn λ_1 mehrfach auftritt, und auch, wenn A nicht diagonalähnlich ist. Hat aber A zwei verschiedene betragsmaximale Eigenwerte, dann konvergiert das Verfahren nicht (siehe Übung).*

Wenn A symmetrisch ist ($A = A^T$), dann gilt genauer:

$$\vec{x}^{(k)} = \cos\varphi_k \cdot \vec{u}^{(1)} + \sin\varphi_k \cdot \vec{r}^{(k)}; \quad \|\vec{r}^{(k)}\|_2 = 1;$$
$$\vec{r}^{(k)T}\vec{u}^{(1)} = 0 \quad \text{mit } |\tan\varphi_k| \leq \left|\frac{\lambda_2}{\lambda_1}\right| \cdot |\tan\varphi_{k-1}|$$

<<

Beweisskizze für Satz 7.3.1 Im Folgenden ist "Faktor" der Normierungsfaktor auf Länge eins, der für die Konvergenz in der Richtung irrelevant ist. Voraussetzung: $\exists T: \; T^{-1}AT = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} =: \Lambda$

$$\Rightarrow AT = T\Lambda$$
$$T = (\vec{u}^{(1)}, \ldots, \vec{u}^{(n)})$$
$$AT = A(\vec{u}^{(1)}, \ldots, \vec{u}^{(n)}) = (\vec{u}^{(1)}, \ldots, \vec{u}^{(n)})\Lambda = T\Lambda$$
$$(A\vec{u}^{(1)}, \ldots, A\vec{u}^{(n)}) = (\vec{u}^{(1)}\lambda_1, \ldots, \vec{u}^{(n)}\lambda_n)$$
$$\iff \qquad A\vec{u}^{(i)} = \vec{u}^{(i)}\lambda_i$$
$$A = T\Lambda T^{-1} = (\lambda_1\vec{u}^{(1)}, \ldots, \lambda_n\vec{u}^{(n)}) \begin{pmatrix} \vec{v}^{(1)H} \\ \cdots \\ \vec{v}^{(n)H} \end{pmatrix} = \underbrace{\sum_{i=1}^{n} \lambda_i \vec{u}^{(i)}\vec{v}^{(i)H}}_{\text{“Spektralzerlegung von } A\text{”}}$$

$$\left(\text{mit } T^{-1} = \begin{pmatrix} \vec{v}^{(1)H} \\ \cdots \\ \vec{v}^{(n)H} \end{pmatrix}\right)$$

$$
\begin{aligned}
T^{-1}A &= \Lambda T^{-1} = \begin{pmatrix} \lambda_1 \cdot \vec{v}^{(1)H} \\ \cdots \\ \lambda_n \cdot \vec{v}^{(n)H} \end{pmatrix} \Rightarrow \\
\vec{v}^{(i)H}A &= \lambda_i \vec{v}^{(i)H} \quad \text{mit } \vec{v}^{(i)H} = \text{“Linkseigenvektor”} \\
\vec{x}^{(k+1)} &= \text{Faktor } \cdot A \cdot \vec{x}^{(k)} = \cdots = \text{Faktor} \cdot \ldots \cdot \text{Faktor} \cdot A^{k+1} \cdot \vec{x}^{(0)} \\
&= \text{Faktor} \cdot (T\Lambda T^{-1})^{k+1} \cdot \vec{x}^{(0)} \\
&= \text{Faktor} \cdot (T\Lambda T^{-1})(T\Lambda T^{-1}) \cdot \ldots \cdot (T\Lambda T^{-1}) \cdot \vec{x}^{(0)} \\
&= \text{Faktor} \cdot T\Lambda^{k+1}T^{-1} \cdot \vec{x}^{(0)} \\
&\quad \text{mit } \vec{x}^{(0)} = \sum_{i=1}^{n} \alpha_i \cdot \vec{u}^{(i)} = T \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\
&\quad \Rightarrow T^{-1}\vec{x}^{(0)} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\
\vec{x}^{(k+1)} &= \underbrace{\text{Faktoren}}_{\text{Normierung auf } \|.\|_2=1} \cdot T \cdot \Lambda^{k+1} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\
&= \text{Faktoren} \cdot \sum_{i=1}^{n} \vec{u}^{(i)} \cdot \lambda_i^{k+1} \cdot \alpha_i \\
&= \underbrace{\text{Faktoren} \cdot \lambda_1^{k+1}}_{\text{Faktor}} \Big(\underbrace{\alpha_1}_{\neq 0} \vec{u}^{(1)} + \underbrace{\sum_{i=2}^{n} \underbrace{\vec{u}^{(i)}}_{\|.\|_2=1} \underbrace{\left(\frac{\lambda_i}{\lambda_1}\right)^{k+1}}_{\leq \left|\frac{\lambda_2}{\lambda_1}\right|^{k+1}} \cdot \alpha_i}_{\text{Fehlervektor}} \Big)
\end{aligned}
$$

□

>>

Ist A nicht diagonalähnlich, dann muß man zum Konvergenzbeweis die Jordan-Normalform benutzen. Es folgt, daß der Fehler $\leq \frac{const}{k}$ ist, die Konvergenz ist dann also extrem langsam. Aber auch im diagonalähnlichen Fall ist die Konvergenz schlecht, wenn $\left|\frac{\lambda_2}{\lambda_1}\right| \approx 1$, d.h. wenn die Eigenwerte nicht gut separiert sind.

Das **Wielandt-Verfahren** dient der **Bestimmung des betragskleinsten Eigenwerts** einer Matrix A.

Wir setzen voraus, daß A regulär ist. Denn wäre A nicht regulär, dann würde ein $\vec{x} \neq \vec{0}$ mit $A\vec{x} = 0 = 0 \cdot \vec{x}$ existieren, d.h. 0 wäre der betragskleinste Eigenwert und $\vec{x}$ ein zugehöriger Eigenvektor.

Wir stellen zunächst fest, daß λ_i genau dann Eigenwert von A ist, falls $\frac{1}{\lambda_i}$ Eigenwert von A^{-1} ist, denn es gilt $A\vec{x} = \lambda\vec{x} \iff \vec{x} = \lambda A^{-1}\vec{x} \iff \frac{1}{\lambda}\vec{x} = A^{-1}\vec{x}$.

Weiterhin gilt, daß $A-\mu I_n$ die Eigenwerte $\lambda_i-\mu$ hat wegen $\det(A-\lambda I_n)=\det(A-\mu I_n-(\lambda-\mu)I_n)$. $(A-\mu I_n)^{-1}$ hat also die Eigenwerte $\frac{1}{\lambda_i-\mu}$.

Das von Mises-Verfahren für $(A-\mu I_n)^{-1}$ liefert formal folgendes Teilproblem: Berechne $\tilde{\vec{x}}^{(k+1)}=(A-\mu I_n)^{-1}\vec{x}^{(k)}$. Wegen der Äquivalenz der letzten Gleichung zu $(A-\mu I_n)\tilde{\vec{x}}^{(k+1)}=\vec{x}^{(k)}$ kann man diesen Schritt über eine LR-Zerlegung der Form $P(A-\mu I_n)=LR$ oder eine QR-Zerlegung $Q(A-\mu I_n)=R$ durchführen.

Wir haben also ein Gleichungssystem für $\tilde{\vec{x}}^{(k+1)}$ zu lösen. Hier ist der Rundungsfehlereinfluß beim Gauß-Algorithmus mit Pivotisierung bzw. der QR-Zerlegung harmlos in seiner Wirkung auf die berechnete Richtung. Weiterhin überträgt sich die Aussage von Satz 7.3.1 von A auf $(A-\mu I_n)^{-1}$. Wir kommen nun zur Wahl von $\vec{x}^{(0)}$. In der Regel ist es nicht leicht, eine gute Startnäherung zu raten. Im Falle einer nicht zerfallenden oberen Hessenbergmatrix A, d.h. es gilt $a_{i,j}=0$ für $j<i-1$ mit $a_{i,i-1}\neq 0$ für $i=2,\ldots,n$, die das Aussehen $A=\begin{pmatrix} * & \cdot & \cdots & \cdots & \cdot \\ * & * & \ddots & & \vdots \\ & * & * & \ddots & \vdots \\ 0 & & * & * & \cdot \\ & & & * & * \end{pmatrix}$ mit $*\neq 0$ hat, stellen wir die Behauptung auf, daß die Wahl von

$$\vec{x}^{(0)}=\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

immer geeignet ist im Sinne der Voraussetzungen von Satz 7.3.1. Denn Satz 7.3.1 liefert

$$\alpha_1=(T^{-1}\vec{x}^{(0)})_1=(\vec{e}^{(1)})^T\cdot T^{-1}\vec{x}^{(0)}=\vec{v}^{(1)H}\vec{x}^{(0)}\ ,$$

$\vec{v}^{(1)H}$ ist Linkseigenvektor zu λ_1.

Mit $\vec{x}^{(0)}=\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ ist $\vec{v}^{(1)H}\vec{x}^{(0)}$ die 1. Komponente von $\vec{v}^{(1)H}$.

Wir zeigen nun die Zwischenbehauptung, daß kein Linkseigenvektor einer nicht zerfallenden oberen Hessenbergmatrix die erste Komponente $=0$ haben kann. Dazu genügt es, den Fall $n=3$ zu betrachten.

Mit der Annahme $(0, \bar{v}_2, \bar{v}_3) \begin{pmatrix} * & * & * \\ \otimes & * & * \\ 0 & \otimes & * \end{pmatrix} = \lambda(0, \bar{v}_2, \bar{v}_3)$ folgt $\bar{v}_2 \cdot \otimes = \lambda \cdot 0 = 0$

$\otimes \neq 0 \;\Rightarrow\; \bar{v}_2 = 0$

$$(0, 0, \bar{v}_3) \cdot \begin{pmatrix} * \\ * \\ \otimes \end{pmatrix} = \lambda \bar{v}_2 = \lambda \cdot 0 = 0 = \bar{v}_3 \cdot \otimes$$

$\otimes \neq 0 \Rightarrow \bar{v}_3 = 0$

$\Rightarrow \vec{v}^H = (\bar{v}_1, \bar{v}_2, \bar{v}_3) = \vec{0} \Rightarrow \vec{v} = \vec{0}$ Widerspruch! ($\vec{0}$ ist kein Eigenvektor)

Nun kann **jede** Matrix durch unitäre Ähnlichkeitstransformationen auf obere Hessenberggestalt gebracht werden. Dazu benutze man etwa $n-2$ Householdertransformationen von rechts und links, die so berechnet werden, daß sie die Elemente unterhalb des Subdiagonalelements in den Spalten $1, \ldots, n-2$ in null überführen. Damit kann man also jedes Eigenwertproblem auf das einer nichtzerfallenden oberen Hessenberform zurückführen und hat insoweit das Problem der Startvektorwahl beseitigt. Im hermitischen Fall erhält man automatisch eine hermitische Dreibandmatrix.

Wenn man in der Lage ist, einen Eigenvektor zu berechnen, dann kann man auch das vollständige Eigenwert- Eigenvektorproblem lösen. Dies skizzieren wir im folgenden Abschnitt.

7.4 Ein Verfahren zur Bestimmung aller Eigenwerte und Eigenvektoren

Ist ein Paar $(\lambda, \vec{x})$ mit $A\vec{x} = \lambda\vec{x}$ mit $\vec{x} \neq \vec{0}$ gefunden, dann kann die Bestimmung der übrigen Paare wieder mit dem Wielandt-Verfahren nach vorheriger Matrix-Transformation erfolgen:

Man bestimmt die Householder-Matrix U_1 mit $U_1\vec{x} = \begin{pmatrix} \|\vec{x}\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

O.B.d.A. sei $\|\vec{x}\|_2 = 1$. Dann folgt $U_1\vec{x} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.

Es gilt $U_1 = U_1^H = U_1^{-1}$. Wir setzen $B := U_1^H A U_1$ und stellen die folgende Behauptung auf:

$$B = \left(\begin{array}{c|c} \lambda_1 & * \cdots \cdots * \\ \hline 0 & \\ \vdots & \underbrace{\tilde{B}}_{(n-1)\times(n-1)} \\ & \\ 0 & \end{array}\right)$$

d.h. $B\vec{e}^{(1)} = \lambda_1 \vec{e}^{(1)}$

$\iff U_1^H A U_1 \vec{e}^{(1)} = \lambda_1 \vec{e}^{(1)}$

$\iff AU_1\vec{e}^{(1)} = \lambda_1 U_1 \vec{e}^{(1)} \qquad (\ U_1\vec{x}^{(1)} = \vec{e}_1 \iff \vec{x}^{(1)} = U_1^{-1}\vec{e}^{(1)} = U_1\vec{e}^{(1)}\ )$

$\iff A\vec{x}^{(1)} = \lambda_1 \vec{x}^{(1)}$

□

Aber

$$\begin{aligned} \det\,(B-\lambda I_n) &= \det\,(U_1^H A U_1 - \lambda I_n) = \det\,(U_1^H A U_1 - \lambda U_1^H U_1) \\ &= \det\,(U_1^H (A-\lambda I_n) U_1) = \det\,U_1^H \cdot \det\,(A-\lambda I_n)\cdot \det\,(U_1) \\ &= \det\,(A-\lambda I_n) \end{aligned}$$

und

$$\det\,(B-\lambda I_n) = \det \left(\begin{array}{c|c} \lambda_1-\lambda & * \cdots \cdots * \\ \hline 0 & \\ \vdots & \tilde{B}-\lambda I_{n-1} \\ 0 & \end{array}\right) = (\lambda_1-\lambda)\cdot \det\,(\tilde{B}-\lambda I_{n-1})$$

d.h. man kann das gleiche Verfahren für $\tilde{B}$ anwenden usw. Für die Praxis wesentlich ist nun, daß man diese Rechenschritte auch mit genäherten Eigenvektoren ausführen und schliesslich sie sogar vollständig mit der Iteration im Wielandtverfahren mischen kann. Dies führt zum **QR-Verfahren** zur Bestimmung aller Eigenwerte und Eigenvektoren.

Dieses Verfahren ist formal definiert durch die Vorschrift

$$\begin{aligned} A_0 &= A \\ k &= 0,1,\ldots \\ A_k - \mu_k I &= Q_k R_k \qquad \text{QR-Zerlegung} \\ A_{k+1} &= R_k Q_k + \mu_k I \end{aligned}$$

mit geeignet gewählten μ_k.

Für eine hermitische Dreibandmatrix kann man eine Wahl von μ_k angeben, sodaß immer Konvergenz und in der Regel sogar superkubische Konvergenz eintritt. Der Beweis benutzt die Tatsache, daß jeder Schritt dieses Verfahrens als ein Wielandtiterationsschritt, jetzt aber mit variablem Shift pro Schritt, gedeutet werden kann. Auch für den nichthermitischen Fall kennt man Shifttechniken, die praktisch immer zum Erfolg führen. Die Iteration erhält die Hessenberg- und Dreibandform, was für ihre Effizienz wesentlich ist. Details siehe in der Spezialliteratur.

NUMAWWW Eigenwertprobleme

Kapitel 8

Differenzenformeln, numerisches Differenzieren, Zweipunktrandwertaufgaben

8.1 Differenzenformeln

In vielen Anwendungen ist man gezwungen, den Wert der Ableitung einer Funktion numerisch zu berechnen, z.B. weil die formale Differentiation unmöglich oder zu kompliziert ist. Einige mögliche Vorgehensweisen haben wir bereits kennengelernt, nämlich die Differentiation eines Interpolationspolynoms oder die eines interpolierenden Splines. Eine andere Möglichkeit besteht in der direkten Auswertung von Differenzenquotienten. Im eindimensionalen Fall erhält man alle üblichen Formeln durch Differentiation von Interpolationspolynomen zu gegebenen Gitterwerten.

NUMAWWW Interpolation/Numerische Differentiation

So ist z.B. der symmetrische Differenzenquotient

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

der Wert der Ableitung der Parabel durch $(x-h, f(x-h)), (x, f(x)), (x+h, f(x+h))$ an der Stelle x und entsprechend der Differenzenquotient zweiter Ordnung

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

die zweite Ableitung des gleichen Polynoms. Wählt man als Auswertungsstelle nicht die Intervallmitte, erhält man andere Formeln, so z.B.

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

als Ableitung der interpolierenden Parabel zu $(x, f(x)), (x+h, f(x+h), (x+2h, f(x+2h))$ an der Stelle x. Für die vierte Ableitung erhält man analog

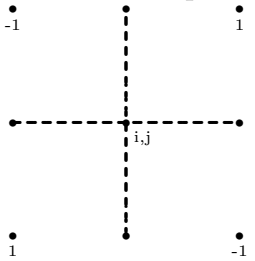
$$\frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} \;=\; f^{(4)}(x) + \mathcal{O}(h^2)$$

für 6-mal stetig differenzierbares f.

Hat man Funktionen mehrerer Veränderlicher, so kann man bei partiellen Ableitungen nach nur einer der Veränderlichen, also etwa $u_x,\ u_{xx},\ u_{xxx}$ die gleiche Methode zur Herleitung von Formeln benutzen. Bei gemischten Ableitungen benutzt man dagegen besser die Methode der Taylorreihenentwicklung. Etwa für u_{xy}:

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

Hierbei steht $u_{i,j}$ für $u(x_i, y_j)$ und wir nehmen an, daß wir auf einem quadratischen Gitter arbeiten: $x_i = x_0 + ih,\ y_j = y_0 + jh$.



Anwendung der **Taylor-Formel** (alle Terme auf der rechten Seite werden in (x_i, y_j) ausgewertet) führt auf :

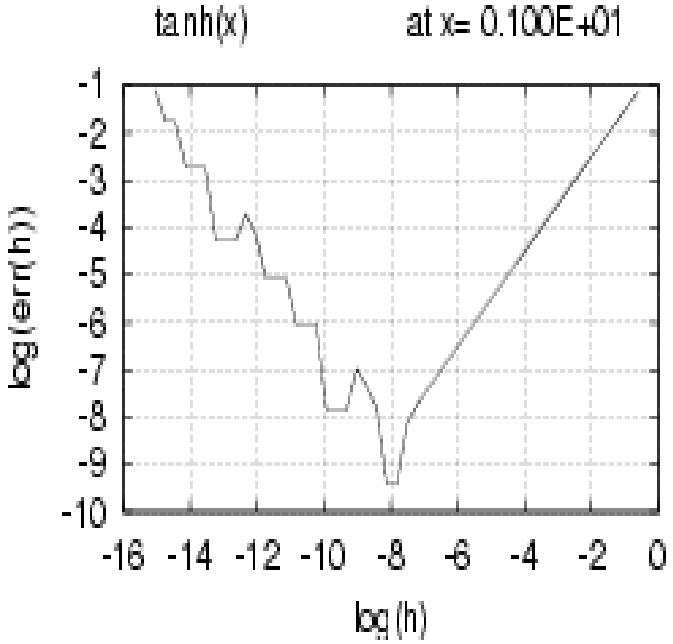
$$\begin{array}{c|l}
+ & u_{i+1,j+1} = \quad u + (u_x)\cdot h + (u_y)\cdot h + \frac{1}{2}h^2\Big((u_{xx}) + 2(u_{xy}) + (u_{yy})\Big) + \\
 & \qquad + \frac{1}{6}h^3\Big(u_{xxx} + 3u_{xxy} + 3u_{yyx} + u_{yyy}\Big) + \mathcal{O}(h^4) \\
+ & u_{i-1,j-1} = \quad u + (u_x)\cdot(-h) + (u_y)\cdot(-h) + \frac{1}{2}h^2\Big((u_{xx}) + 2(u_{xy}) + (u_{yy})\Big) \\
 & \qquad - \frac{1}{6}h^3\Big(u_{xxx} + 3u_{xxy} + 3u_{yyx} + u_{yyy}\Big) + \mathcal{O}(h^4) \\
+ & -u_{i+1,j-1} = \quad -u - (u_x)\cdot h + (u_y)\cdot h - \frac{1}{2}h^2\Big((u_{xx}) - 2(u_{xy}) + (u_{yy})\Big) \\
 & \qquad - \frac{1}{6}h^3\Big(u_{xxx} - 3u_{xxy} + 3u_{yyx} - u_{yyy}\Big) + \mathcal{O}(h^4) \\
+ & -u_{i-1,j+1} = \quad -u + (u_x)\cdot h - (u_y)\cdot h - \frac{1}{2}h^2\Big((u_{xx}) - 2(u_{xy}) + (u_{yy})\Big) \\
 & \qquad - \frac{1}{6}h^3\Big(-u_{xxx} + 3u_{xxy} - 3u_{yyx} + u_{yyy}\Big) + \mathcal{O}(h^4) \\
\hline
 & \qquad \{.....\} = 4h^2(u_{xy})_{ij} + 4\mathcal{O}(h^4) \\
 & \frac{1}{4h^2}\{.....\} = (u_{xy})_{ij} + \mathcal{O}(h^2)
\end{array}$$

Diese Formeln werden z.B. benutzt, um Differentialgleichungsprobleme in algebraische Gleichungen für Näherungswerte der Lösung auf Gitterpunkten zu überführen. Dabei sind die Funktionswerte selbst zu bestimmende Unbekannte.

8.2 Numerisches Differenzieren

Man kann die oben angegebenen Formeln auch , und dies geschieht häufig, benutzen, um aus gegebenen oder berechneten Funktionswerten Ableitungswerte explizit zu berechnen. Wird im angenäherten Wert nur eine sehr geringe Genauigkeit gefordert, dann ist die Anwendung aller dieser Formeln unkritisch. Will man aber eine hohe Genauigkeit erzielen, so steht man vor erheblichen Problemen, wie das folgende Beispiel zeigt:

Beispiel 8.2.1. $f(x) = \tanh(x)$, $f'(x) \approx \frac{f(x+h)-f(x)}{h} = f_{[x,x+h]}$.
$\overline{x} = 1.0$, $f'(\overline{x}) = 0.41997434161403$, Rechengenauigkeit $\varepsilon = 5 \cdot 10^{-16}$ (53-stellige binäre Gleitpunktrechnung).



Man erkennt, daß der Fehler zunächst bis etwa $h = \sqrt{\varepsilon}$ linear fällt, um danach genauso schnell wieder anzuwachsen. □

Offensichtlich gibt es beim Differenzenquotienten eine optimale Schrittweite, nach deren Unterschreiten der Fehler wieder anwächst. Dies ist natürlich ein reiner Rundungsfehlereffekt, denn bei exakter Rechnung gilt

$$f'(x) = \lim_{h \to 0} f_{[x,x+h]} \,.$$

Eine Rundungsfehleranalyse macht dies sofort klar. Auf die Details wollen wir hier nicht eingehen. Es ergibt sich, daß für den gewöhnlichen Vorwärts- oder Rückwärtsdifferenzenquotienten die optimale Schrittweite in der Grössenordnung $\sqrt{\varepsilon}$ liegt mit einer optimalen Genauigkeit ebenfalls von $\sqrt{\varepsilon}$, für den symmetrischen Differenzenquotienten ist

die optimale Schrittweite von der Grössenordnung $\varepsilon^{1/3}$ mit einer optimalen Genauigkeit in der Grössenordnung $\varepsilon^{2/3}$ und ganz allgemein für eine Näherungsformel der Ordnung p für eine $q-$te Ableitung ergibt sich die optimale Schrittweite aus einer Gleichung der Form

$$-C_1 \frac{q\varepsilon}{h^{q+1}} + C_2 p h^{p-1} \;=\; 0$$

worin C_1 und C_2 Konstanten sind, die von den Gewichten der Formel und einer höheren Ableitung der Funktion f in der Nähe von x abhängen, und ε die Rechengenauigkeit bezeichnet. Die optimale Schrittweite ist also von der Grössenordnung $\varepsilon^{\frac{1}{p+q}}$ und die optimale Genauigkeit $\varepsilon^{\frac{p}{p+q}}$. Im Prinzip kann man also durch grosses p die Genauigkeit bis nahe an die Rechengenauigkeit steigern, benötigt dafür aber eine hohe Anzahl von Funktionswerten, z.B. für eine 1. Ableitung und Ordnung 6 6 Funktionswerte (bei symmetrischer Anordnung zu x).

8.3 Zweipunktrandwertaufgaben

Wir wenden uns nun Randwertproblemen gewöhnlicher Differentialgleichungen zu. Wir betrachten zunächst Differentialgleichungen zweiter Ordnung für y und Bedingungen an y an zwei verschiedenen Stellen. Bei diesen Differentialgleichungsproblemen bezeichnet die "freie" Variable normalerweise eine räumliche Grösse, weshalb wir hier wieder die Variable x dafür verwenden.

Aufgabenstellung: Gesucht ist eine Funktion y, die

$$y'' \;=\; f(x, y, y') \quad \text{mit } f:\ [a,b] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}\,,$$

sowie die Randbedingungen

$$\begin{aligned} g_1(y(a), y(b), y'(a), y'(b)) &= 0 \\ g_2(y(a), y(b), y'(a), y'(b)) &= 0 \end{aligned}$$

erfüllt.

Ein **einfaches Modell** ist hierbei

$$y'' = f(x, y, y') \quad \text{für } a < x < b$$

mit $y(a) = 0 = y(b)$.

Man hat hier das Problem, daß es keine allgemeinen Existenz- oder Eindeutigkeitsaussagen gibt und auch nicht geben kann, wie das folgende Beispiel zeigt.

Beispiel 8.3.1.

$$y'' + exp(y+1) \;=\; 0, \quad x \;\in\; [0,1], \quad y(0) \;=\; y(1) \;=\; 0$$

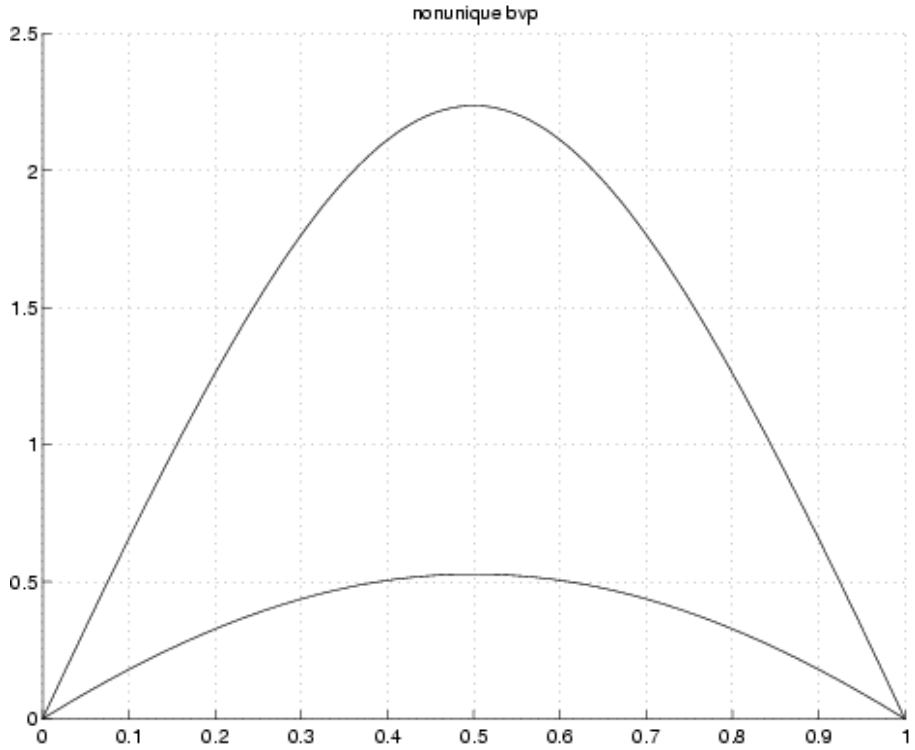
mit der Lösung

$$y(x) \;=\; -2\ln\left(\frac{\cosh((x-1/2)\theta/2)}{\cosh(\theta/4)}\right)$$

wo θ die beiden Lösungen von

$$\theta \;=\; \sqrt{2e}\cosh(\theta/4)$$

bezeichnet. □



In der Praxis geht man davon aus, daß das vorgelegte Problem lösbar ist und im nichtlinearen Fall, in dem f nichtlinear von y oder y' abhängt auch davon, daß man eine genügend gute Startnäherung hat.

Nun bestimmen wir eine **Diskretisierung** der obigen Aufgabe in mehreren Schritten:

1. Schritt: Festlegen eines Gitters mit Gitterpunkten $x_i = a + i \cdot h$ für $i = 0, \ldots, N$ und $h = \frac{b-a}{N}$. Wir schreiben nun

$$y''(x_i) = f(x_i, y_i, y_i')$$

mit den Setzungen $y_i := y(x_i)$, $y_i' := y'(x_i)$.

2. Schritt: Ersetzen der Ableitungen durch Differenzenquotienten, d.h. durch Ableitungen von Interpolationspolynomen.

$$\begin{aligned} y_i' &= \frac{y_{i+1} - y_{i-1}}{2h} + \mathcal{O}(h^2) \\ y_i'' &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \mathcal{O}(h^2) \end{aligned}$$

3. Schritt: Bezeichnet y_i^h die Näherung für y_i, so erhalten wir unter Vernachlässigung der $\mathcal{O}$-Terme

$$\oplus \quad \frac{y_{i+1}^h - 2y_i^h + y_{i-1}^h}{h^2} = f(x_i, y_i^h, \frac{y_{i+1}^h - y_{i-1}^h}{2h}), \quad 1 \le i \le N-1$$

sowie $y_0^h = y(a) = 0, \quad y_N^h = y(b) = 0.$

Die Gleichung $\oplus$ ist nichtlinear, wenn f nichtlinear in y, y' ist.

Beispiel für $N = 4$, d.h. $i = 0, 1, 2, 3, 4$:

Vernachlässigung von $\mathcal{O}(h^2)$-Termen entspricht dem Übergang von y_i zu Approximationen y_i^h. Wir erhalten

$$\begin{aligned} \text{i} = 1: \quad \frac{y_2^h - 2y_1^h + y_0^h}{h^2} &= f\Big(x_1, y_1^h, \frac{y_2^h - y_0^h}{2h}\Big) \quad \text{mit } y_0^h = 0 \\ \text{i} = 2: \quad \frac{y_3^h - 2y_2^h + y_1^h}{h^2} &= f\Big(x_2, y_2^h, \frac{y_3^h - y_1^h}{2h}\Big) \\ \text{i} = 3: \quad \frac{y_4^h - 2y_3^h + y_2^h}{h^2} &= f\Big(x_3, y_3^h, \frac{y_4^h - y_2^h}{2h}\Big) \quad \text{mit } y_4^h = 0 \end{aligned}$$

Es ergibt sich also ein System zur Bestimmung von $\vec{y}^h = \begin{pmatrix} y_1^h \\ \vdots \\ y_3^h \end{pmatrix}$.

Dies ist ein gekoppeltes System, welches – wenn f nichtlinear in y oder y' ist – ebenfalls nichtlinear ist. Für diesen Fall einer skalaren Differentialgleichung zweiter Ordnung gibt es noch Existenz- und Eindeutigkeitsaussagen:

Satz 8.3.1.
Sei $f \in C^1([a,b]\times\mathbb{R}\times\mathbb{R})$. *Ferner gelte* $|\frac{\partial}{\partial s}f(x,r,s)| \le B$ *für alle* $x\in[a,b]$, $r,s\in\mathbb{R}$ *sowie* $0 \le \frac{\partial}{\partial r}f(x,r,s)$ *für alle* $t\in[a,b]$, $r,s\in\mathbb{R}$.
Dann ist die Randwertaufgabe (RWA) eindeutig lösbar. Das diskrete System ist eindeutig lösbar, falls $0 < h \le \min\left\{\frac{2}{(B+4)^2}, b-a\right\}$. *Ist* f *linear, so genügt* $h \le \min\left\{\frac{2}{B}, b-a\right\}$.
Für die diskrete Lösung $\vec{y}^h$ *gilt weiterhin*

$$\oplus \quad |y(x_i) - y_i^h| \le C\cdot h^2$$

mit einer geeigneten Konstanten C *(Verfahren 2. Ordnung).*
Falls $f \in C^{2m+2}([a,b]\times\mathbb{R}\times\mathbb{R})$, *dann gilt mit geeigneten Funktionen* $e_i(x)$*:*

$$\oplus\oplus \quad y_i^h = y(x_i) + e_1(x_i)h^2 + \cdots + e_m(x_i)h^{2m} + \mathcal{O}(h^{2m+2})\,.$$

Beispiel 8.3.2.

- $y'' = (x^2+1)\cdot y' + y - \arctan y,\ x\in[0,1]$
  Es ist also $f(x,r,s) = (x^2+1)\cdot s + r - \arctan r$. *Daraus folgt nun* $\frac{\partial}{\partial s}f(x,r,s) = x^2+1 \to B = 2$ *sowie* $\frac{\partial}{\partial r}f(x,r,s) = 1 - \frac{1}{1+r^2} = \frac{r^2}{1+r^2} \ge 0$ *für alle* r.
- $y'' = \frac{3}{2}y^2$ *für* $x \in [0,1]$, $y(0)=4,\ y(1)=1$.
  Hier ist $f(x,r,s) = \frac{3}{2}r^2$ *und* $B=0$, *aber* $\frac{\partial}{\partial r}f(x,r,s) = 3r$ *ist nicht* ≥ 0 *für alle* r*!*
  Der Satz ist nicht anwendbar und die Eindeutigkeit ist auch nicht gegeben.

□

Beispiel 8.3.3. *Wir diskretisieren die Randwertaufgabe*

$$y'' + y' + y = 0,\ x\in[0,1] \quad \textit{mit} \quad y'(0) = 1 \textit{ und } y(1) = 0,$$

mittels obiger finiter Differenzen: Die Diskretisierung der DGl erfolgt mit dem zentralen Differenzenquotienten

$$\underbrace{\frac{1}{h^2}\left(y_{i+1} - 2y_i + y_{i-1}\right)}_{=y''(x_i)+\mathcal{O}(h^2)} + \underbrace{\frac{1}{2h}\left(y_{i+1} - y_{i-1}\right)}_{=y'(x_i)+\mathcal{O}(h^2)} + y_i = 0$$

für $i = 0,1,\ldots,N$. *Problematisch ist nur die Approximation im linken Randpunkt* $x_0 = 0$, *da dort von Neumann-Randbedingungen gegeben sind. Um die Ordnung 2 zu erhalten, wird ein fiktiver Punkt* x_{-1} *bzw.* y_{-1} *angefügt und die Randableitung mit dem zentralen Differenzenquotienten*

$$\underbrace{\frac{1}{2h}\left(y_1 - y_{-1}\right)}_{=y'(x_0)+\mathcal{O}(h^2)} = 1.$$

approximiert. Das entstehende Gleichungssystem hat zunächst $N+2$ Unbekannte $y_{-1}, \ldots, y_N$

$$\left(\begin{array}{c|c|c|c|c} -\frac{1}{2h} & 0 & \frac{1}{2h} & 0 & \cdots \\ \hline -\frac{1}{2h}+\frac{1}{h^2} & -\frac{2}{h^2}+1 & \frac{1}{2h}+\frac{1}{h^2} & \ddots & \\ \hline 0 & -\frac{1}{2h}+\frac{1}{h^2} & -\frac{2}{h^2}+1 & \frac{1}{2h}+\frac{1}{h^2} & \\ \hline & & \ddots & \ddots & \ddots \\ \hline & \cdots & 0 & -\frac{1}{2h}+\frac{1}{h^2} & -\frac{2}{h^2}+1 \end{array}\right) \left(\begin{array}{c} y_{-1} \\ y_0 \\ y_1 \\ \vdots \\ y_N \end{array}\right) = \left(\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array}\right)$$

Durch geeignete Kombination der ersten beiden Zeilen wird dann y_{-1} eliminiert. Auf diesem Wege entsteht wieder ein System mit Tridiagonalmatrix aber nur noch $N+1$ Unbekannten $y_0, \ldots, y_N$

$$\left(\begin{array}{c|c|c|c|c} -\frac{2}{h^2}+1 & \frac{2}{h^2} & 0 & \cdots & \\ \hline -\frac{1}{2h}+\frac{1}{h^2} & -\frac{2}{h^2}+1 & \frac{1}{2h}+\frac{1}{h^2} & \ddots & \\ \hline & \ddots & \ddots & \ddots & \\ \hline & \cdots & 0 & -\frac{1}{2h}+\frac{1}{h^2} & -\frac{2}{h^2}+1 \end{array}\right) \left(\begin{array}{c} y_0 \\ y_1 \\ \vdots \\ y_N \end{array}\right) = \left(\begin{array}{c} \frac{2}{h}-1 \\ \vdots \\ 0 \end{array}\right).$$

□

Wir wenden uns nun der Frage zu, welche Bedeutung die Voraussetzungen an h in diesem Satz haben. Die i-te Zeile der Jacobi-Matrix des entstandenen (nicht)linearen Systems hat folgende Form:

$$\begin{aligned} & \tfrac{1}{h^2}(0,\ldots,0,1,-2,1,0,\ldots,0) - (0,\ldots,0,\Big(\frac{\partial}{\partial s}f\Big)\Big(-\frac{1}{2h}\Big),\Big(\frac{\partial}{\partial r}f\Big)\cdot 1,\Big(\frac{\partial}{\partial s}f\Big)\Big(\frac{1}{2h}\Big),0,\ldots,0) \\ = \;& \tfrac{1}{h^2}(0,\ldots,0,1+\tfrac{h}{2}\cdot\Big(\frac{\partial}{\partial s}f\Big),-2-h^2\cdot\Big(\frac{\partial}{\partial r}f\Big),1-\tfrac{h}{2}\cdot\Big(\frac{\partial}{\partial s}f\Big),0,\ldots,0)\,. \end{aligned}$$

Die Voraussetzungen an f und h in obigem Satz ergeben nun, daß die Jacobimatrix unabhängig von den für y eingesetzten Werten immer invertierbar ist, d.h. daß das nichtlineare System für genügend kleines h eindeutig lösbar ist. Die Jacobi-Matrix von oben ist unsymmetrisch, wenn

$$\Big(\frac{\partial}{\partial s}f\Big) \;\neq\; 0\,,$$

denn die Bedingung der Symmetrie würde erfordern, daß $-1+\frac{h}{2}\cdot r_i = -1-\frac{h}{2}\cdot r_{i+1}$ gilt mit

$$r_i \;=\; \Big(\frac{\partial}{\partial s}f\Big)(x_i, y_i^h, \frac{y_{i+1}^h - y_{i-1}^h}{2h})$$

Für $h \to 0$ würde das aber $r \equiv 0$ erfordern.

Eine **Randwertaufgabe in selbstadjungierter Form** hat folgende Gestalt:

$$-\tfrac{\partial}{\partial x}\Big(a(x)\cdot\tfrac{\partial}{\partial x}y\Big)+c(x)\cdot y=g(x),\qquad \text{mit } a(x)\geq\alpha>0\,.$$

Hier ist nun eine symmetrische Diskretisierung möglich. Wir erhalten:

1. Schritt:

$$\begin{aligned}\tfrac{\partial}{\partial x}\Big(a(x)\cdot\tfrac{\partial}{\partial x}y(x)\Big) &= \tfrac{\partial}{\partial x}(a(x)\cdot y'(x))_{|_{x=x_i}}\\ &= \frac{a(x_i+\frac{h}{2})\cdot y'(x_i+\frac{h}{2})-a(x_i-\frac{h}{2})\cdot y'(x_i-\frac{h}{2})}{h}+\mathcal{O}(h^2)\end{aligned}$$

2. Schritt:

$$\begin{aligned}y'(x_i+\frac{h}{2}) &= \frac{y(x_{i+1})-y(x_i)}{h}+\mathcal{O}(h^2)\\ y'(x_i-\frac{h}{2}) &= \frac{y(x_i)-y(x_{i-1})}{h}+\mathcal{O}(h^2)\end{aligned}$$

Einsetzen der Formeln aus dem zweiten Schritt in die des ersten Schrittes liefert nun eine symmetrische Matrix, denn die Koeffizienten der **Gesamtformel**

$$-\tfrac{\partial}{\partial x}(a(x)\cdot\tfrac{\partial}{\partial x}y(x))=-\tfrac{1}{h^2}(a(x_i+\tfrac{h}{2})\cdot(y_{i+1}-y_i)-a(x_i-\tfrac{h}{2})\cdot(y_i-y_{i-1}))$$

bei den entsprechenden Unbekannten lauten nun

$$\begin{aligned}y_{i-1} &: -\tfrac{1}{h^2}\cdot a(x_i-\tfrac{h}{2})\\ y_i &: \tfrac{1}{h^2}\cdot(a(x_i+\tfrac{h}{2})+a(x_i-\tfrac{h}{2}))\\ y_{i+1} &: -\tfrac{1}{h^2}\cdot a(x_i+\tfrac{h}{2})\,.\end{aligned}$$

Die Diskretisierung des Terms $c(x)y$ liefert nur Beiträge zur Diagonalen.

Durch genauere Taylorentwicklung erhält man auch hier einen **Gesamtfehler** der Form $\mathcal{O}(h^2)$.

Beispiel 8.3.4. *Gegeben sei die Randwertaufgabe in selbstadjungierter Form*

$$-((1+x^3)y')'+xy=0,\ x\in[0,1]\quad \textit{mit}\quad y(0)=y(1)=1.$$

Die Diskretisierung führen wir einmal mit der hier angegebenen speziellen Diskretisierung durch und vergleichen mit der Standardvorgehensweise, bei der die DGL ausdifferenziert und dann die "normale" Diskretisierung durchgeführt wird:

1. *Das Resultat der symmetrischen Diskretisierung ist*

$$-\frac{1}{h^2}\left((1+(x_i+\frac{h}{2})^3)(y_{i+1}-y_i)-(1+(x_i-\frac{h}{2})^3)(y_i-y_{i-1})\right)+x_iy_i=0$$

mit $i=1,2,3$ und $y_0=y_4=1$.

Benötigt werden die Zahlenwerte

x_i	$x_i \pm \frac{h}{2}$	$\frac{1+(x_i \pm \frac{h}{2})^3}{h^2}$
0.25	*0.125*	*16.03125*
0.5	*0.375*	*16.84375*
0.75	*0.625*	*19.90625*
	0.875	*26.71875*

Sortiert man in den einzelnen Gleichungen nach y_i und bringt die Randwerte auf die rechte Seite, so ergibt sich das lineare Gleichungssystem

$$\begin{pmatrix} 33.125 & -16.84375 & 0 \\ -16.84375 & 37.25 & -19.90625 \\ 0 & -19.90625 & 47.375 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 16.03125 \\ 0 \\ 26.71875 \end{pmatrix}.$$

Man kann zeigen, daß die Matrix dieses Systems positiv definit ist.

2. *Nach dem Ausdifferenzieren lautet die DGL*

$$-(1+x^3)y'' - 3x^2y' + xy = 0.$$

Die erste und zweite Ableitung werden durch die jeweiligen zentralen Differenzenquotienten approximiert und es ergibt sich

$$-\frac{1+(x_i)^3}{h^2}(y_{i+1} - 2y_i + y_{i-1}) - \frac{3(x_i)^2}{2h}(y_{i+1} - y_{i-1}) + x_i y_i = 0$$

mit $i = 1, 2, 3$ und $y_0 = y_4 = 1$. Benötigt werden nun die Zahlenwerte

x_i	$\frac{1+(x_i)^3}{h^2}$	$\frac{3(x_i)^2}{2h}$
0.25	*16.25*	*0.375*
0.5	*18*	*1.5*
0.75	*22.75*	*3.375*

Sortiert man in den einzelnen Gleichungen wieder nach y_i und bringt die Randwerte auf die rechte Seite, so ergibt sich jetzt das lineare Gleichungssystem

$$\begin{pmatrix} 32.75 & -16.625 & 0 \\ -16.5 & 36.5 & -19.5 \\ 0 & -19.375 & 46.25 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 15.875 \\ 0 \\ 26.125 \end{pmatrix}.$$

Es handelt sich immer noch um eine invertierbare Matrix. Die Symmetrie ist aber verloren gegangen.

□

Allgemeine lineare Randbedingungen werden entweder wegtransformiert oder direkt diskretisiert.

Die **1. Möglichkeit** ist eine Diskretisierung wie bisher und die Verwendung einseitiger Differenzenquotienten für y'. Für das Beispiel

$$\begin{array}{l} -y'' = f(x,y,y'),\ y'(0)=0,\ y(1)=1, \\ h=\frac{1}{N+1},\ x_i = ih,\ y_{N+1}=1, \\ \frac{y_1^h - y_0^h}{h} = 0, \quad N+1 \text{ Unbekannte } y_0^h,\ldots,y_N^h \end{array}$$

ist der Gesamtfehler von der Form $\mathcal{O}(h^2)$, obwohl der Fehler im Randwert von der Form $\mathcal{O}(h)$ ist.

Ist die Lösung der Differentialgleichung (bzw. ihre Ableitung) in a bzw. b nicht singulär, dann gibt es bessere Möglichkeiten:

1. Man verschiebt das Gitter um $\frac{h}{2}$, wobei $h = \frac{1}{N+1}$. $\frac{y_1^h - y_0^h}{h} = 0$ als Ersatz für $y'(0)=0$ jetzt einen Fehler der Form $\mathcal{O}(h^2)$ hat. Einen Dirichletrandwert ersetzt man dabei durch den Mittelwert der beiden benachbarten Gitterwerte, ebenfalls mit einem Fehler $\mathcal{O}(h^2)$.

2. Es werden "fiktive Punkte" eingeführt, d.h. $x_i = ih,\ i = -1,\ldots,N+1$. Man wählt dann $\frac{y_1^h - y_{-1}^h}{2h} = 0$ und benutzt die Differentialgleichung auch in $i=0$. Aus der zweiten Randbedingung folgt dann $y_{N+1}^h = 1$.

Bemerkung 8.3.1. *In der Praxis treten viel allgemeinere Randwertaufgaben auf, z.B. in der Form*

$$y' \ = \ F(x,y) \quad \textit{mit } F : [a,b] \times \mathcal{D} \ \to \ \mathbb{R}^n$$

worin $\mathcal{D}$ ein Gebiet des $\mathbb{R}^n$ ist, mit den zugehörigen n Randbedingungen in der Form eines nichtlinearen Gleichungssystems

$$R(y(a), y(b)) \ = \ 0 \ .$$

Oft hat man sogar noch Bedingungen an Zwischenstellen im Inneren des Intervalls $[a,b]$ und Mischungen von Differentialgleichungen verschiedener Ordnung. Dann sind die obigen einfachen Vorgehensweisen nicht mehr möglich. Eine Vorgehensweise findet man im folgenden Abschnitt.

8.3.1 Kollokationsmethoden

Wir kehren zurück zur allgemeinen Zweipunktrandwertaufgabe

$$y' \ = \ F(x,y), \quad x \ \in]a,b[, \quad R(y(a),y(b)) \ = \ 0 \ .$$

Auch hier können wir sinnvolle Differenzenformeln direkt ansetzen. Ein Beispiel ist die implizite Mittelpunktregel

$$\frac{y_{i+1}^h - y_i^h}{h} \;=\; F(x_i + \tfrac{h}{2}, \frac{y_i^h + y_{i+1}^h}{2}) \;,\; i \;=\; 0, \ldots, N-1$$

mit $h = (b-a)/N$ oder die Trapezregel

$$\frac{y_{i+1}^h - y_i^h}{h} \;=\; \tfrac{1}{2}(F(x_i, y_i^h) + F(x_{i+1}, y_{i+1}^h)) \;.$$

Zusammen mit der Randbedingung entsteht also ein

Gleichungssystem für $y_0^h, \ldots, y_N^h$.

Dieses ist nichtlinear, wenn F oder R nichtlinear in y sind. In Abhängigkeit von der Jacobi-Matrix von F bezüglich y bzw. der von R können wieder die gleichen Probleme auftreten, die wir schon bei den Gleichungen zweiter Ordnung diskutiert haben. Diese beiden Formeln ergeben sich als Spezialfälle einer Vorgehensweise, die als Kollokationsmethode bezeichnet wird. Die Idee dieser Methoden ist die folgende: man ersetzt die gesuchte Lösung der DGL lokal (d.h. zwischen zwei Gitterpunkten x_i und x_{i+1}) durch eine "einfache" Funktion $\varphi(x)$ und legt die Parameter, die diese Funktion beschreiben, dadurch fest, daß man fordert, daß diese an gewissen Punkten die Differentialgleichung (exakt) erfüllt. In der Regel ist φ ein Polynom vom Grad s, das man durch die $s+1$ Forderungen

$$\begin{aligned} \varphi(x_i) &= y_i^h \\ \varphi'(x_i + \alpha_j h) &= F(x_i + \alpha_j h, \varphi(x_i + \alpha_j h)),\; j \;=\; 1, \ldots, s \end{aligned}$$

festlegt. Man setzt dann

$$y_{i+1}^h \;=\; \varphi(x_i + h) \;.$$

Die Koeffzienten α_j erfüllen dabei

$$0 \;\leq\; \alpha_1 \;<\; \alpha_2 \;<\; \ldots \;<\alpha_s \;\leq\; 1 \;.$$

In den obigen Beispielen ist $s = 1$ und $\alpha_1 = 1/2$ bzw. $s = 2$ und $\alpha_1 = 0,\; \alpha_2 = 1$. Durch die Interpolationsforderungen ist das Polynom φ eindeutig bestimmt. Nach der Formel von Lagrange ergibt sich

$$\varphi'(x_i + \tau h) \;=\; \sum_{j=1}^{s} L_j(x_i + \tau h) K_j$$

mit

$$K_j \;\stackrel{def}{=}\; \varphi'(x_i + \alpha_j h)$$

und

$$L_j(x_i + \tau h) \;=\; \prod_{l=1, l\neq j}^{s} \frac{\tau - \alpha_l}{\alpha_j - \alpha_l} \; .$$

Man beachte, daß L_j unabhängig von h und x_i ist. Integration von φ' ergibt dann

$$\begin{aligned} \varphi(x_i + \alpha_j h) - \varphi(x_i) &= h \sum_{l=1}^{s} \beta_{j,l} K_l \\ \varphi(x_{i+1}) - \varphi(x_i) &= h \sum_{l=1}^{s} \gamma_l K_l \end{aligned}$$

wobei

$$\begin{aligned} \beta_{j,l} &= \int_0^{\alpha_j} L_l(\tau) \mathrm{d}\tau \; , \\ \gamma_l &= \int_0^{1} L_l(\tau) \mathrm{d}\tau \; . \end{aligned}$$

Setzen wir dies und die Definition der K_l in die Bestimmungsgleichungen wieder ein, so erhalten wir die Gleichungen

$$\begin{aligned} K_j &= F(x_i + \alpha_j h, y_i^h + h \sum_{l=1}^{s} \beta_{j,l} K_l), \; j \;=\; 1, \ldots, s \\ y_{i+1}^h &= y_i^h + h \sum_{l=1}^{s} \gamma_l K_l \; . \end{aligned}$$

Bei gegebenem y_i^h sind also zunächst die Gleichungen für die K_l zu lösen, danach kann man y_{i+1}^h unmittelbar angeben. Man beachte, daß die $\beta_{k,l}$ und die γ_l nur von den gewählten Knoten α_j abhängen.

Ein solches Verfahren bezeichnet man als $s-$stufiges Runge-Kutta-Verfahren. Durch die obigen Festlegungen der $\beta_{i,j}$ und γ_j hat das Verfahren automatisch die Mindestordnung s. Man beachte, daß durch die Wahl der α_j, also der Knoten, das ganze Verfahren bereits eindeutig festgelegt ist. Im Zusammenhang mit einer Randwertaufgabe erhalten wir also simultane Gleichungen für alle y_i^h und die Werte $K_l(i)$, die ja auch noch von der Gitterstelle abhängen. Sinnvolle Wahlen der α_j sind hier solche, die symmetrisch zur Intervallmitte liegen und eine möglichst hohe Ordnung besitzen. Es bieten sich dazu also die Gaussknoten (bezogen auf $[0, 1]$) (dies ergibt die Ordnung $2s$) und die sogenannten Lobatto-Knoten an. Diese haben stets $\alpha_1 = 0, \alpha_s = 1$ und die noch freien Knoten werden so gewählt, daß die Ordnung maximal wird, also $2s - 2$.

Beispiel 8.3.5. *Gauss - Runge- Kutta - Verfahren mit 2 Stufen der Ordnung 4:*

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Beispiel 8.3.6. *Gauss - Lobatto - Runge- Kutta - Verfahren mit 3 Stufen der Ordnung 4:*

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Die implizite Mittelpunktregel ist zugleich die einfachste Gauss-Regel und die Trapezregel die einfachste Lobatto-Quadratur. Es gilt zur Konvergenz dieser Verfahren der

Satz 8.3.2. *Die Randwertaufgabe sei lokal eindeutig lösbar und F und R hinreichend oft differenzierbar. Dann konvergiert die Kollokationsmethode mit den Gaussknoten und s Stufen von der Ordnung $2s$ und die mit s Stufen und den Lobatto-Knoten von der Ordnung $2s-2$.*

Für den Beweis siehe z.B. bei Ascher und Petzold. Diese Kollokationsmethoden kann man auch bei viel allgemeineren Problemen (Mehrpunktprobleme, wo auch an Zwischenstellen Vorschriften für die Funktion gegeben sind, Gleichungen höherer Ordnung, Gleichungen mit algebraischen Nebenbedingungen) einsetzen und es gibt gute Software dafür (z.B. `COLDAE, COLNEW` von Ascher und Bader in der `NETLIB`).

Kapitel 9

Elliptische Randwertprobleme in zwei freien Veränderlichen

9.0 Klassifizierung der semilinearen partiellen DGLen 2. Ordnung

Wir betrachten die **allgemeine Differentialgleichung**

$$\oplus \quad -\left(a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy}\right) \;=\; f(x,y,u,u_x,u_y) \quad \text{für } (x,y) \in G \subset \mathbb{R}^2\,.$$

f darf hier von x,y und zusätzlich von u,u_x, u_y abhängen, aber die Koeffizienten a, b, c nur von x, y. Eine solche Gleichung bezeichnet man als semilinear . Ihr sogenannter Typ (s.u.) hängt nicht von der Lösung u ab, wie dies der Fall wäre, wenn u oder eine der partiellen Ableitungen von u in den Koeffzienten $a,\ b,\ c$ aufträte.

Definition 9.0.1. *Eine Differentialgleichung der obigen Form $\oplus$ heißt*

$$\left\{\begin{array}{l}\textbf{elliptisch}\\ \textbf{parabolisch}\\ \textbf{hyperbolisch}\end{array}\right\} \textit{ auf } G,\ \textit{wenn die Matrix } B = \begin{pmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y)\end{pmatrix}$$

$$\textit{auf } G \left\{\begin{array}{l}\textit{negativ oder positiv definit}\\ \textit{singulär , aber nicht die Nullmatrix}\\ \textit{indefinit}\end{array}\right\} \textit{ ist.}$$

Beispiele:

- Physikalisch beschreiben elliptische Probleme Gleichgewichtszustände.

  Als Modellproblem betrachten wir die Poisson-Gleichung der Form

  $$-\Delta u = f(x,y) \text{ mit } \Delta u := u_{xx} + u_{yy}\,.$$

  Diese beschreibt die Verformung einer elastischen Membran unter verteilter Last f.

  Die Matrix $B = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ ist negativ definit.

- Physikalisch beschreiben parabolische Probleme Diffusionsvorgänge.

  Wir betrachen die Wärmeleitungsgleichung der Form

  $$u_t = u_{xx} + g(x,t)\,,$$

  die die Wärmeleitung in einem isolierten Stab beschreibt, bei dem der Wärmetransport nur über die Stabenden erfolgt.

  Die Matrix $B = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$ ist singulär.

- Als Modellproblem eines hyperbolischen Problems betrachten wir die Wellengleichung (schwingende Saite) der Form

  $$u_{tt} = c^2 u_{xx} \text{ mit } c \neq 0\,.$$

  Hier ist $B = \begin{pmatrix} -c^2 & 0 \\ 0 & 1 \end{pmatrix}$. B ist indefinit.

Zu der Differentialgleichung treten stets noch Bedingungen an u oder die Normalableitung von u auf dem Rand oder einem Teil des Randes von G hinzu, sowie weitere Glattheits- und Kompatibilitätsbedingungen an die Koeffizienten und Inhomogenitäten. Es muss sichergestellt sein, daß das Problem wohlgestellt ist im Sinne von Hadamard, d.h. die Lösung existiert, ist eindeutig und hängt stetig von den Daten ab, die das Problem beschreiben. Dabei ist Stetigkeit bezüglich einer geeigneten Funktionennorm definiert.

9.1 Differenzenverfahren

Wir beschäftigen uns nun mit dem Modellproblem eines elliptischen Randwertproblems, konkret mit dem Poisson-Problem. Dieses hat auf dem Einheitsquadrat die Gestalt

$$-\Delta u = f(x,y,u,u_x,u_y) \quad \text{für } (x,y) \in]0,1[\times]0,1[$$

Als Zusatzbedingung muß man hier Randbedingungen auf dem **gesamten** Rand von G vorgeben.

Ein **Beispiel** für mögliche Randbedingungen ist die Forderung $u_{|\partial G} = 0$, d.h. $u(x,y) = 0$, falls x oder $y \in \{0,1\}$. u beschreibt die Verformung einer elastischen Membran, die am Rand eingespannt ist.

Prinzipiell gibt es zur approximativen Lösung verschiedene Möglichkeiten. Als 1. Lösungsansatz betrachten wir **Differenzenverfahren** und gehen folgendermaßen vor:

1. Wir betrachten die DGL nur auf einem Gitter (x_i, y_j), $1 \le i,j \le N$ mit $x_i = ih$, $y_j = jh$ und $h = \frac{1}{N+1}$. Wir wählen als Bezeichnungen $u_{ij} := u(x_i, y_j)$, $(u_x)_{i,j} = u_x(x_i, y_j)$, usw. .

2. Nun ersetzen wir die partiellen Ableitungen durch Differenzenquotienten. $\frac{\partial}{\partial x}$ entspricht einer Ableitung in x bzw. i-Richtung. Entsprechend gehört $\frac{\partial}{\partial y}$ zu einer Ableitung in y bzw. j-Richtung. Wir verwenden nun

$$\begin{aligned}(u_x)_{ij} &= \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \mathcal{O}(h^2)\,,\\ (u_{xx})_{ij} &= \tfrac{1}{h^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}) + \mathcal{O}(h^2)\,,\\ (u_{yy})_{ij} &= \tfrac{1}{h^2}(u_{i,j+1} - 2u_{ij} + u_{i,j-1}) + \mathcal{O}(h^2)\,.\end{aligned}$$

   Einsetzen in die DGL unter Vernachlässigung der $\mathcal{O}(h^2)$-Terme ergibt ein Gleichungssystem zur Bestimmung einer Näherungslösung u^h_{ij} für u_{ij}.

$$\begin{cases} -\frac{1}{h^2} \cdot \Big(u^h_{i+1,j} + u^h_{i,j+1} + u^h_{i-1,j} + u^h_{i,j-1} - 4u^h_{i,j}\Big) \\ \qquad = f\Big(x_i, y_j, u^h_{i,j}, \frac{u^h_{i+1,j} - u^h_{i-1,j}}{2h}, \frac{u^h_{i,j+1} - u^h_{i,j-1}}{2h}\Big), \quad 1 \le i,j \le N \end{cases}$$

   mit

$$u^h_{0,j} \equiv 0, \quad u^h_{i,0} \equiv 0, \quad u^h_{N+1,j} \equiv 0, \quad u^h_{i,N+1} \equiv 0\,.$$

   Wir erhalten N^2 Gleichungen mit N^2 Unbekannten. Ist f linear in u, u_x, u_y, so ist auch das Gleichungssystem linear. Die linke Seite der Gleichung wird repräsentiert durch einen sogenannten Differenzenstern:

gemeinsamer Faktor: $\frac{1}{h^2}$
Differenzenstern:

$$\begin{array}{ccccc} & & -1 & & \\ & & \vdots & & \\ -1 & \cdots & 4 & \cdots & -1 \\ & & \vdots & & \\ & & -1 & & \end{array}$$

Andere Randbedingungen kann man analog behandeln:

Wir betrachten dazu die Forderung, daß $\frac{\partial}{\partial n}u = c(x,y)$ auf einem Teilstück des Randes von G ist.

$$\vec{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}. \quad \frac{\partial}{\partial \vec{n}}u = u_x \cdot n_1 + u_y \cdot n_2 = (\nabla u)^T \cdot \vec{n}\,.$$

Hier bezeichnet $\frac{\partial}{\partial n}u$ die Richtungsableitung in Richtung der äußeren Normalen. In Falle des Einheitsquadrates wird diese Forderung also zu einer Forderung an u_x oder u_y auf dem Randstück.

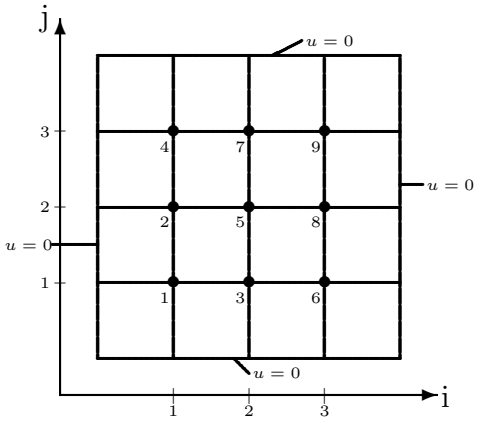
Wir führen nun fiktive Punkte ein, diskretisieren die Randableitung und benutzen die DGL auch am Rand, um eine zusätzliche Gleichung für die fiktiven Punkte zu erhalten.

Die Struktur der Matrix (bzw. bei nichtlinearem f der Jacobi-Matrix des nichtlinearen Systems) hängt von der Numerierung der Unbekannten ab.

1. Regel: Die Gleichung für die k-te Unbekannte wird im System auch als k-te Gleichung geführt.

Eine **Schrägzeilennumerierung** führt auf eine konsistent geordnete, symmetrische, irreduzibel-diagonaldominante L-Matrix, also eine positiv definite M-Matrix.

Wir betrachten dazu das **Beispiel:**



$N = 3, \quad h = \frac{1}{4}$

$\Rightarrow$ 9×9-Matrix
block-tridiagonal:
Diagonalblöcke: $4I$,
L-Matrix, irreduzibel diagonaldominant

Die entstehende Matrix hat nun folgende Gestalt:

$$A = \left(\begin{array}{ccc|ccc|ccc} 4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 4 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 & -1 & 0 & 0 & 0 \\ \hline 0 & -1 & 0 & 4 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 4 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 4 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 4 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 4 \end{array}\right)$$

Wird hingegen eine **zeilenweise Numerierung** zugrundegelegt, wobei sich die Gleichungsnummer aus $k = (i-1)N + j$ berechnet, so erhalten wir ein Gleichungssystem $A\vec{x} = \vec{b}$ mit der Matrix

$$A = \begin{pmatrix} T & -I & & \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & T \end{pmatrix} \quad \text{und} \quad T := \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{pmatrix}.$$

Hierbei ist $I = I_N$ und $A = \mathbb{R}^{N^2 \times N^2}$.

Jede **andere Numerierung** entspricht einer Abbildung $\vec{x} \mapsto P\vec{x}$ mit einer Permutationsmatrix P.

Wir betrachten dann das System $AP^TP\vec{x} = \vec{b}$ mit $P^T = P^{-1}$, numerieren die Gleichungen um (Zeilentausch) und erhalten das System $PAP^TP\vec{x} = P\vec{b}$.

Bei zeilenweiser Numerierung hat A die Bandbreite $2N + 1$. In der Praxis löst man ein System mit A mit dem Cholesky-Verfahren, solange der Speicherbedarf dies erlaubt. Das Cholesky-Verfahren erzeugt aber totales "fill in", d.h. die Matrix L aus dem Verfahren hat die Gestalt

$$L = \begin{pmatrix} & & & 0 \\ & & \iddots & \\ & \iddots & \iddots & \\ 0 & & & \end{pmatrix}$$

und benötigt $(N+1) \times N^2$ Speicherplätze.

Deshalb benutzt man für größeres N (oder 3D-Probleme) iterative Methoden, wie z.B. das präkonditionierte cg-Verfahren oder das SOR-Verfahren mit ω_{optimal}.

Bemerkung: Falls f von u_x oder u_y abhängt, dann wird die Matrix A unsymmetrisch!

Als Beispiel betrachten wir

$$-\Delta u + au_x + bu_y = g(x,y)\,,$$

es ist also $f(x,y,u,u_x,u_y) = g(x,y) - au_x - bu_y$.

Bei zeilenweiser Numerierung und unter Verwendung von

$$(u_x)_{ij} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad (u_y)_{ij} \approx \frac{u_{i,j+1} - u_{i,j-1}}{2h}$$

hat die i-te Zeile der Matrix die Gestalt:

$$\tfrac{1}{h^2}(0,\ldots,0,-1-b\cdot\tfrac{h}{2},\underbrace{0,\ldots,0}_{N-2},-1-a\cdot\tfrac{h}{2},4,-1+a\cdot\tfrac{h}{2},\underbrace{0,\ldots,0}_{N-2},-1+b\cdot\tfrac{h}{2},0,\ldots,0)$$

Jetzt muß $h < \frac{2}{\max\{|a|,|b|\}}$ gelten, sonst hat man keine L-Struktur.

Satz 9.1.1. *Für die obige Aufgabe ($\oplus$) mit $b(x,y) = 0$ und mit ihrer Diskretisierung gilt folgende Aussage:*
Hängt f nicht von u, u_x, u_y ab, dann besitzt die diskretisierte Aufgabe für alle $h > 0$ eindeutige Lösungen $\vec{u}^h$ und es gibt eine Konstante $c > 0$, so daß

$$|u(x_i, y_j) - u^h_{i,j}| \leq c \cdot h^2,$$

falls zusätzlich für die wahre Lösung $u \in C^4(\bar{G})$ mit $\bar{G} := [0,1] \times [0,1]$ gilt.
Hängt f nicht nur von x und y ab, so benötigt man weitere Voraussetzungen:

1. $\frac{\partial}{\partial u} f \leq 0, \quad \left|\frac{\partial}{\partial u_x} f\right|, \left|\frac{\partial}{\partial u_y} f\right| \leq B,$
2. *h sei hinreichend klein.*

Als eine **andere Form der Problemstellung** betrachten wir nun die sogenannte **selbstadjungierte Form**.

$$-\frac{\partial}{\partial x}\Big(a_1(x,y)\frac{\partial}{\partial x}u\Big) - \frac{\partial}{\partial y}\Big(a_2(x,y)\frac{\partial}{\partial y}u\Big) + c(x,y)u = g(x,y) \tag{9.1}$$

mit $(x,y) \in G$, $a(x,y) > 0$, $b(x,y) > 0$, $c(x,y) \geq 0$.

Eine **passende Diskretisierung** läßt sich mittels der Näherungsformel

$$\begin{aligned} -\frac{1}{h^2}\Big((a_1)_{i+\frac{1}{2},j}(u^h_{i+1,j} - u^h_{i,j}) - (a_1)_{i-\frac{1}{2},j}(u^h_{i,j} - u^h_{i-1,j}) + \\ (a_2)_{i,j+\frac{1}{2}}(u^h_{i,j+1} - u^h_{i,j}) - (a_2)_{i,j-\frac{1}{2}}(u^h_{i,j} - u^h_{i,j-\frac{1}{2}}) + h^2 c_{i,j} u^h_{i,j}\Big) \\ = \quad g_{i,j} \end{aligned}$$

angeben. Im Falle von homogenen Dirichletranddaten ist die entstehende Matrix A nun immer symmetrisch positiv definit.
Hierfür gilt eine zum vorstehenden Satz analoge Aussage, wegen der speziellen Form der Problemstellung gilt die Aussage aber für alle Diskretisierungsschrittweiten.

Bemerkung 9.1.1. *Falls die partiellen Ableitungen von f nach der vierten und fünften Variablen (entsprechend u_x und u_y) betragsmässig sehr gross werden, ist das Problem konvektionsdominiert. Dann liefert der obige Satz sehr restriktive Bedingungen an die Gitterweite h. Für solche Fälle gibt es spezielle Diskretisierungen, siehe dazu die Spezialliteratur.*

Zum Abschluß geben wir noch eine Differenzenapproximation für u_{xy} an: Der in 9.1 betrachtete Differenzenstern hat den Nachteil, die M-Matrix Struktur zu zerstören. Geeigneter ist deshalb ein allgemeinerer Differenzenstern. Wir legen einen Neun-Punkte-Stern auf einem quadratischen Gitter zugrunde. Dann kann man durch Taylorentwicklung bis zur Ordnung $\mathcal{O}(h^4)$ zeigen, daß der allgemeine Neun-Punkte-Stern der Konsistenzordnung 2 für u_{xy} die Form

$$\frac{1}{h^2}\begin{bmatrix} -\frac{1}{4}-\frac{1}{6}\alpha & \frac{1}{3}\alpha & \frac{1}{4}-\frac{1}{6}\alpha \\ \frac{1}{3}\alpha & -\frac{2}{3}\alpha & \frac{1}{3}\alpha \\ \frac{1}{4}-\frac{1}{6}\alpha & \frac{1}{3}\alpha & -\frac{1}{4}-\frac{1}{6}\alpha \end{bmatrix}$$

hat mit beliebigem reellen α. $\alpha = 0$ ergibt die oben erwähnte Näherung. Mit $\alpha = 3/2$ erhalten wir

$$\frac{1}{2h^2}\begin{bmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

und mit $\alpha = -3/2$

$$\frac{1}{2h^2}\begin{bmatrix} 0 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

Da $\partial_1\partial_2 u$ in der DGL mit $2b(x,y)$ multipliziert wird und man die M-Matrixstruktur erhalten will, benutzt man für $b(x,y) < 0$ den ersten und für $b(x,y) \geq 0$ den zweiten. Damit wird im Ganzen der Differentialoperator

$$-(a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy})$$

ersetzt durch den Differenzenstern

$$\frac{1}{h^2}\begin{bmatrix} b^- & -c+|b| & -b^+ \\ -a+|b| & 2(a+c-|b|) & -a+|b| \\ -b^+ & -c+|b| & b^- \end{bmatrix}.$$

Hier ist $b^+ = \max\{0,b\}$ und $b^- = \min\{0,b\}$. Dieser Differenzenstern hat die M-Struktur, wenn

$$a > |b| \text{ und } c > |b| \tag{9.2}$$

auf G. Die gleichmässige Elliptizität bedeutet aber

$$a > \alpha > 0\,,\ c > \alpha > 0\,,\ \ ac - b^2 > \alpha > 0$$

auf G. Unter dieser Bedingung kann man durch eine Umskalierung von x oder y erreichen, daß (9.2) erfüllt ist. Die Diskretisierung der ersten Ableitungen erfolgt wieder

durch den zentralen Differenzenquotienten. Dann ergibt sich

Satz 9.1.2. *Der Differentialoperator sei gleichmässig elliptisch auf G und die Variablen x, y seien so skaliert, daß (9.2) gilt. Falls dann h hinreichend klein ist, dann gilt für die obige Diskretisierung*

1. *Das Verfahren ist konsistent von zweiter Ordnung*
2. *Die Jacobimatrix des (nicht)linearen Systems ist eine M-Matrix*
3. *Das Verfahren ist konvergent von der Ordnung 2*

□

9.2 Ritzsches Verfahren und die Methode der finiten Elemente

Wir gehen nun von der selbstadjungierten Form aus und betrachten die **Aufgabenstellung:**

$$-\frac{\partial}{\partial x}\Big(a(x,y)\frac{\partial}{\partial x}u\Big)-\frac{\partial}{\partial y}\Big(b(x,y)\frac{\partial}{\partial y}u\Big)+c(x,y)u=g(x,y) \text{ für } (x,y)\in G \quad u_{|\partial G}=0. \tag{9.3}$$

mit $a(x,y)\geq\alpha>0$, $b(x,y)\geq\beta>0$, $c(x,y)\geq 0$ und $a,b\in C^1(\bar{G})$, $c\in C^0(\bar{G})$. $\bar{G}$ ist hierbei ein Bereich, für den der Gaußsche Integralsatz anwendbar ist, z.B. kann $\bar{G}$ eine endliche Vereinigung von Normalbereichen sein.

Der sogenannte **Ritzsche Ansatz** geht nun folgendermaßen vor:

- Wir wählen Ansatzfunktionen $\varphi_1,\ldots,\varphi_N\in C^2(\bar{G})$, die die Randbedingungen erfüllen, d.h. es gilt $\varphi_{j|\partial G}=0$ für $j=1,\ldots,N$.
- Wir machen den Ansatz $u^h:=\sum_{i=1}^{N}\alpha_i\varphi_i$ mit $\alpha_1,\ldots,\alpha_N$ als unbekannten Koeffizienten.

  Gesucht sind die Koeffizienten α_i.
- Wir setzen obigen Ansatz in die Differentialgleichung ein.
- Dann multiplizieren wir jeweils mit φ_j, $j=1,\ldots,N$ und erhalten so N Gleichungen.

- Anschließend werden beide Seiten über G integriert und man erhält

$$\begin{aligned}\sum_{i=1}^{N}\alpha_i\Big\{&\int_G\Big(-\tfrac{\partial}{\partial x}\Big(a(x,y)\tfrac{\partial}{\partial x}\varphi_i(x,y)\Big)\Big)\cdot\varphi_j(x,y)\,d(x,y)\\&+\int_G\Big(-\tfrac{\partial}{\partial y}\Big(b(x,y)\tfrac{\partial}{\partial y}\varphi_i(x,y)\Big)\Big)\cdot\varphi_j(x,y)\,d(x,y)\\&+\int_G\varphi_i(x,y)\cdot c(x,y)\cdot\varphi_j(x,y)\cdot d(x,y)\Big\}\\=&\int_G\varphi_j(x,y)\cdot g(x,y)\cdot\,\mathrm{d}(x,y),\quad j=1,\ldots,N\,.\end{aligned}$$

- Man gewinnt somit N Gleichungen für die N gesuchten Koeffizienten $\alpha_1,\ldots,\alpha_N$.

Bemerkung 9.2.1. *Gaußscher Integralsatz*
$\int\limits_G$ *div* $F\,\mathrm{d}(x,y)=\oint\limits_{\partial G}F^T\vec{n}\,\mathrm{d}s,\quad F=\binom{F_1}{F_2}$ *mit der äußeren Normalen* $\vec{n}$ *auf* G.

Wir betrachten weiter das folgende Funktional:

$$I(v)\;=\;\int_G\Big(a(x,y)\cdot(\underbrace{\tfrac{\partial}{\partial x}v}_{v_x(x,y)})^2+b(x,y)\cdot(\tfrac{\partial}{\partial y}v)^2+c(x,y)\cdot v^2-2g\cdot v\Big)\,\mathrm{d}(x,y)$$

und formulieren folgendes Minimierungsproblem:

Bestimme v, so daß

$$I(v)\;\stackrel{!}{=}\;\min_{v\in V}\tag{9.4}$$

mit $V=\{v|v\in C^2(G)\cap C^0(\bar{G}),\ v_{|\partial G}=0\}$. Hat man auf einem Teil des Randes ∂G_1 von Neumann- oder Robin-Bedingungen

$$\frac{\partial u}{\partial n}+\phi(x,y,u)\;=\;0\;\text{ für }(x,y)\;\in\partial G_1$$

dann kommt zum Integral I ein Term

$$\int_{\partial G_1}h(x(s),y(s),u(x(s),y(s)))\mathrm{ds}$$

hinzu mit s als Kurvenparameter auf dem Rand und

$$\frac{\partial}{\partial u}h(x,y,u)\;=\;\phi(x,y,u)\;.$$

Als Beispiel betrachten wir $G =]0,1[\times]0,1[$, $\varphi_1(x,y) = x(1-x)\cdot y(1-y)$ und $V = \{\varphi_1, x\varphi_1, y\varphi_1, x^2\varphi_1, xy\varphi_1, y^2\varphi_1, \ldots,$ und alle konvergenten unendlichen Linearkombinationen davon$\}$.
(Konvergenz im Sinne der Norm $(\int_G \{u^2 + u_x^2 + u_y^2\}\, d(x,y))^{1/2}$)

Satz 9.2.1. *Unter den obigen Voraussetzungen an G und an a, b, c, g ist u genau dann Lösung der Randwertaufgabe (9.3), wenn u Lösung der Minimierungsaufgabe (9.4) ist.*

Wir betrachten ein Beispiel:

Es sei $a \equiv b \equiv 1, \quad c \equiv 0, \quad g \equiv 1$ und damit also

$$-u_{xx} - u_{yy} = 1 \quad \text{mit } u = 0, \quad \text{falls } x \in \{0,1\} \text{ oder } y \in \{0,1\}\,.$$

Das obige Funktional hat nun die Gestalt

$$I(v) \;=\; \int_0^1 \int_0^1 \Big((v_x)^2 + (v_y)^2 - 2v \Big)\, \mathrm{d}x\ \mathrm{d}y = \min_v$$

Wir wählen $N = 3$ und $\varphi_1 = x(1-x)\cdot y(1-y)$, $\varphi_2 = \varphi_1 x$, $\varphi_3 = \varphi_1 y$. Mit dem Ansatz $v := \alpha_1\varphi_1 + \alpha_2\varphi_2 + \alpha_3\varphi_3$ folgt nun als endlichdimensionales Minimierungsproblem

$$\begin{aligned} I(\alpha_1,\alpha_2,\alpha_3) \;&=\; \int_0^1 \int_0^1 \Big((\alpha_1\varphi_{1,x} + \alpha_2\varphi_{2,x} + \alpha_3\varphi_{3,x})^2 + (\alpha_1\varphi_{1,y} + \alpha_2\varphi_{2,y} + \alpha_3\varphi_{3,y})^2 \\ &\qquad -2(\alpha_1\varphi_1 + \alpha_2\varphi_2 + \alpha_3\varphi_3) \Big)\, \mathrm{d}x\, \mathrm{d}y \\ &\overset{!}{=}\; \min_{\alpha_1,\alpha_2,\alpha_3}\,. \end{aligned}$$

Eine **notwendige Extremalbedingung**, die **hier** aufgrund der Voraussetzungen auch hinreichend ist, ist nun

$$\tfrac{\partial}{\partial\alpha_1} I = 0, \quad \tfrac{\partial}{\partial\alpha_2} I = 0, \quad \tfrac{\partial}{\partial\alpha_3} I = 0\,.$$

Wir erhalten also

$$\begin{aligned} &\tfrac{1}{2}\tfrac{\partial}{\partial\alpha_1} I(\alpha_1,\alpha_2,\alpha_3) \\ =\;& \int_0^1 \int_0^1 \Big((\alpha_1\varphi_{1,x} + \alpha_2\varphi_{2,x} + \alpha_3\varphi_{3,x})\cdot\varphi_{1,x} + (\alpha_1\varphi_{1,y} + \alpha_2\varphi_{2,y} + \alpha_3\varphi_{3,y})\cdot\varphi_{1,y} - \varphi_1 \Big)\, \mathrm{d}x\, \mathrm{d}y \\ =\;& 0 \end{aligned}$$

und entsprechend

$$\begin{aligned} \tfrac{1}{2}\tfrac{\partial}{\partial\alpha_2} I(\alpha_1,\alpha_2,\alpha_3) \;&=\; \int_0^1 \int_0^1 ((\ldots)\cdot\varphi_{2,x} + (\ldots)\cdot\varphi_{2,y} - \varphi_2)\, \mathrm{d}x\, \mathrm{d}y \;=\; 0\,, \\ \tfrac{1}{2}\tfrac{\partial}{\partial\alpha_3} I(\alpha_1,\alpha_2,\alpha_3) \;&=\; \int_0^1 \int_0^1 ((\ldots)\cdot\varphi_{3,x} + (\ldots)\cdot\varphi_{3,y} - \varphi_3)\, \mathrm{d}x\, \mathrm{d}y \;=\; 0\,. \end{aligned}$$

Zum Zusammenhang zwischen der Randwertaufgabe und der Variationsaufgabe beachten wir folgende Gleichungen:

$$\begin{aligned}
&\int_0^1 \int_0^1 \underbrace{(-u_{xx}\varphi_1 - u_{yy}\varphi_1)}\, \mathrm{d}x\, \mathrm{d}y \\
= &\int_0^1 \int_0^1 (-\mathrm{div} \begin{pmatrix} u_x\varphi_1 \\ u_y\varphi_1 \end{pmatrix} + u_x\varphi_{1x} + u_y\varphi_{1,y})\, \mathrm{d}x\, \mathrm{d}y \\
= &\int_0^1 \int_0^1 (u_x\varphi_{1,x} + u_y\varphi_{1,x})\, \mathrm{d}(x,y)\,.
\end{aligned}$$

Denn mit Hilfe des Gaußschen Integralsatz gilt

$$\int_G \mathrm{div} \begin{pmatrix} u_x\varphi_1 \\ u_y\varphi_1 \end{pmatrix} = \oint_{\partial G} \begin{pmatrix} u_x\varphi_1 \\ u_y\varphi_1 \end{pmatrix} \vec{n}\, \mathrm{d}s = 0\,, \text{ weil } \varphi_1 = 0 \text{ auf } \partial G$$

und analog für φ_2, φ_3 usw. Am Ende bleibt ein Minimierungsproblem einer Funktion von $\alpha_1, \ldots, \alpha_N$, die in der Form

$$\tfrac{1}{2}\vec{\alpha}^T A\vec{\alpha} - \vec{b}^T\vec{\alpha} + \mathrm{const}$$

geschrieben werden kann, mit einer symmetrischen positiv definiten Matrix A.

Die Lösung dieser Aufgabe kann man durch Lösen des linearen Gleichungssystems

$$A\vec{\alpha} = \vec{b} \text{ mit } \vec{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} \text{ und } \vec{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}$$

bestimmen.

Hierbei ist

$$b_i = \int_G g\varphi_i\, \mathrm{d}x\, \mathrm{d}y$$

$A = (a_{ji})$ mit

$$a_{ji} = \int_G (a_1\varphi_{ix}\varphi_{jx} + a_2\varphi_{iy}\varphi_{jy} + c\varphi_i\varphi_j)\, \mathrm{d}x\, \mathrm{d}y\ .$$

Im allgemeinen Fall stellt sich bei dieser Vorgehensweise das **Problem**, daß die Konstruktion der φ_i, die die Randbedingungen exakt erfüllen und die nötige Differenzierbarkeitsordnung besitzen, praktisch unmöglich ist. Außerdem ist das Gleichungssystem häufig extrem schlecht konditioniert (schon bei kleinem N).

Wir betrachten folgende **Abhilfe:** Betrachte eine Obermenge von V, die einfacher darstellbar ist, und andererseits die Erfüllung der Randbedingungen erlaubt.

Folgende "Obermenge" des Funktionenraumes V ist geeignet

$$H_0^1(G) = \{v | v \in C^0(\bar{G}),\ v_{|\partial G} = 0,\ v \text{ stückweise einmal stetig partiell differenzierbar und } \int\limits_G (v^2 + v_x^2 + v_y^2)\,\mathrm{d}x\,\mathrm{d}y < \infty \text{ sowie alle Funktionen, die als Grenzwerte solcher Funktionen darstellbar sind, wobei die Konvergenz in der Norm } (\int\limits_G (v^2 + v_x^2 + v_y^2)\,\mathrm{d}x\,\mathrm{d}y)^{1/2} \text{ betrachtet wird } \}$$

Dies ist selbst ein (vollständiger) Vektorraum. Man bezeichnet diesen Raum als **Sobolevraum** und die angegebene Norm als die zugehörige Sobolevnorm.

Beispiel: G sei polygonal berandet. Betrachte **alle** zulässigen Triangulierungen von G mit der Zusatzbedingung "**minimaler Winkel** in allen Dreiecken $\geq \psi_0 > 0$", oder, äquivalent: der Quotient aus Umkreis- und Inkreisradius ist für alle Dreiecke beschränkt durch eine Konstante $C > 0$ auch für beliebig feine Diskretisierung. Eine solche Triangulierung nennt man **quasiuniform**. Weiterhin betrachten wir dazu die stetigen, stückweise linearen Funktionen. Diese sind Elemente von $H_0^1(G)$ und jedes Element von $H_0^1(G)$ kann durch Linearkombinationen dieser Funktionen beliebig genau approximiert werden.

Es gilt nun

$$\min_{v \in V} I(v) = \min_{v \in H_0^1(G)} I(v)\,.$$

Idee: Wähle Elemente $\varphi_1, \ldots, \varphi_N$ aus $H_0^1(G)$ und löse das Problem

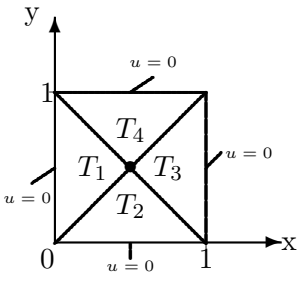
$$I\Big(\sum_{i=1}^{N} \alpha_i \varphi_i\Big) = \min_{\alpha_1, \ldots, \alpha_N}\,.$$

Ist also G polygonal berandet, so wird G trianguliert. Die inneren Knoten seien mit $P_1, \ldots, P_N$ bezeichnet. Man betrachtet die Basisfunktionen $\varphi_1, \ldots, \varphi_N$ der stetigen stückweise linearen Interpolation auf der Triangulierung zu den inneren Knoten.

So erhält man die einfachste Methode der finiten Elemente. Analog kann man mit biquadratischen Ansätzen usw. verfahren.

Als **Beispiel** betrachten wir

$$\begin{aligned} -u_{xx} - (1 + x^2)u_{yy} + xy \cdot u &= xy^2 \text{ auf } G =]0,1[\times]0,1[\\ u_{|\partial G} &= 0 \end{aligned}$$



1 innerer Knoten: $\binom{0.5}{0.5}$

$$\varphi_1(x,y) = \begin{cases} 2x & \text{auf } T_1 \\ 2-2x & \text{auf } T_3 \\ 2y & \text{auf } T_2 \\ 2-2y & \text{auf } T_4 \end{cases}$$

$u \mapsto \alpha_1\varphi_1$

$$\int_0^1\int_0^1 \Big(1\cdot(\alpha_1\varphi_{1,x})^2 + (1+x^2)\cdot(\alpha_1\varphi_{1,y})^2 + xy(\alpha_1\varphi_1)^2 - (2xy^2)(\alpha_1\varphi_1)\Big)\,\mathrm{d}x\,\mathrm{d}y \stackrel{!}{=} \min_{\alpha_1}$$

$\frac{\partial}{\partial\alpha_1}I = 0$:

$$\int_0^1\int_0^1 \Big(2\cdot(\alpha_1\varphi_{1,x})\cdot\varphi_{1,x} + 2(\alpha_1\varphi_{1,y})\cdot(1+x^2)\cdot\varphi_{1,y} + 2xy(\alpha_1\varphi_1)\cdot\varphi_1 - 2xy^2\varphi_1\Big) = 0$$

$$\alpha_1 = \frac{\int_0^1\int_0^1 xy^2\varphi_1(x,y)\,\mathrm{d}x\,\mathrm{d}y}{\int_0^1\int_0^1((\varphi_{1,x})^2 + (1+x^2)\cdot(\varphi_{1,y})^2 + xy\cdot(\varphi_1)^2)\,\mathrm{d}x\,\mathrm{d}y}$$

$\int_0^1\int_0^1 = \int_{T_1} + \int_{T_2} + \int_{T_3} + \int_{T_4}$ Darstellung von φ_1 einsetzen und α_1 ausrechnen!

Für großes N ist A dünnbesetzt, symmetrisch positiv definit und für die Konditionszahl gilt $\mathrm{cond}(A) = \mathcal{O}(\frac{1}{h^2})$. Man kann Konvergenz der so erzeugten Näherungen im Quadratmittel von $\mathcal{O}(h^2)$ beweisen. Da man leicht mit höheren Polynomgraden auf Dreiecksnetzen arbeiten kann (solange man keine Glattheit fordert, was hier ja auch nicht erforderlich ist), kann man leicht höhere Konvergenzordnung erzielen, was mit finiten Differenzen schwierig wird. Bei Praktikern besonders beliebt ist der vollständige quadratische Ansatz auf krummlinigen (parabolischen) Dreiecksnetzen, die sogenannten isoparametrischen quadratischen Elemente. Diese erlauben eine gute Anpassung an krummlinige Ränder und bieten gute Genauigkeit, ohne allzu grosse Elementmatrizen zu erzeugen. (Im $\mathbb{R}^3$ sind die zugehörigen Elementmatrizen auf Quadern immerhin schon von der Dimension 60×60).

Kapitel 10

Parabolische Randanfangswertaufgaben

Als **Modellproblem** betrachten wir die Wärmeleitung in einem isolierten Stab:

$$u_t \;=\; ku_{xx}, \qquad \text{für } x \in]0,1[\text{ und } t > 0 \tag{10.1}$$

mit dem Randbedingungen

$$\begin{aligned} u(0,t) &= 0 \\ u(1,t) &= 0 \end{aligned}$$

und der Anfangsbedingung

$$u(x,0) \;=\; \varphi(x)\,.$$

k ist der Wärmeleitkoeffizient, φ die Temperaturverteilung zur Zeit $t = 0$ und die Randbedingungen besagen, daß die Stabenden auf der konstanten Temperatur null gehalten werden. Damit man eine klassische Lösung hat, muss noch $\varphi(0) = \varphi(1) = 0$ gelten. Die sinnvollste numerische Lösungsmethode ist hier die sogenannte vertikale **Linienmethode**. Zunächst bleibt t kontinuierlich und wir führen eine **Semidiskretisierung** bezüglich des Ortes durch. Wir betrachten die Differentialgleichung nur auf "Linien" $(x_i, t),\ t > 0$.

Mit $x_i = i \cdot \Delta x$, $i = 0, \ldots, N+1$ und $\Delta x = \frac{1}{N+1}$ approximieren wir $u(x_i, t)$.

Wir setzen nun

$$v_i(t) \;:=\; u(x_i,t), \quad 0 \le i \le N+1\,.$$

Aufgrund der Randbedingungen gilt

$$v_0(t) \equiv 0, \quad v_{N+1}(t) \equiv 0$$

und zu bestimmen sind also

$$v_1(t), \ldots, v_N(t)\,.$$

Aus der Wärmeleitungsgleichung (10.1) folgt nun

$$u_t(x_i,t) \;=\; k \cdot u_{xx}(x_i,t)$$

und durch Taylorentwicklung erhält man

$$u_{xx}(x_i,t) \;=\; \frac{\overbrace{u(x_{i+1},t)}^{v_{i+1}(t)} - 2\overbrace{u(x_i,t)}^{v_i(t)} + \overbrace{u(x_{i-1},t)}^{v_{i-1}(t)}}{(\Delta x)^2} + \mathcal{O}((\Delta x)^2)\,.$$

Die Vernachlässigung von $\mathcal{O}((\Delta x)^2)$ ergibt einen Übergang von $v_i(t)$ zu $v_i^h(t)$ mit $h \hat{=} \Delta x$ und wir erhalten

$$\begin{aligned} \dot{v}_i^h(t) &= \frac{k}{(\Delta x)^2} \cdot (v_{i+1}^h(t) - 2v_i^h(t) + v_{i-1}^h(t)), \quad 1 \le i \le N\,, \\ v_0^h(t) &\equiv v_{N+1}^h(t) \equiv 0\,. \end{aligned}$$

Dieses System ist nun ein gekoppeltes System linearer gewöhnlicher Differentialgleichungen der Form

$$\dot{\vec{v}}^h = A\vec{v}^h \text{ mit } \vec{v}^h := \begin{pmatrix} v_1^h \\ \vdots \\ v_N^h \end{pmatrix}$$

und $\vec{v}^h(0) = \begin{pmatrix} \varphi(x_1) \\ \vdots \\ \varphi(x_N) \end{pmatrix}$. Wir haben also ein gewöhnliches lineares Anfangswertproblem für $\vec{v}^h(t)$ als Funktion von t mit der folgenden Matrix A zu lösen.

$$A = \tfrac{k}{(\Delta x)^2} \cdot \begin{pmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix}.$$

Einschub:

Wir betrachten ein Anfangswertproblem der Gestalt

$$\dot{\vec{z}} = A\vec{z} \text{ mit } \vec{z}_0 = \vec{z}(0)\,.$$

Wenn A diagonalähnlich mit den einfachen Eigenwerten $\lambda_1, \ldots, \lambda_N$ und Eigenvektoren $\vec{u}_1, \ldots, \vec{u}_N$ ist, dann gilt für die Lösung $\vec{z}(t)$ des gewöhnlichen Anfangswertproblems:

$$\vec{z}(t) = \sum_{i=1}^{N} \vec{u}_i \cdot \exp(\lambda_i \cdot t) \cdot \alpha_i$$

und weiterhin

$$\vec{z}_0 = \vec{z}(0) = (\vec{u}_1, \ldots, \vec{u}_N) \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}, \text{ d.h. } \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = (\vec{u}_1, \ldots, \vec{u}_N)^{-1} \cdot \vec{z}_0 \,.$$

Die Eigenwerte der obigen Matrix A sind

$$\lambda_i = \tfrac{k}{(\Delta x)^2} \cdot (-2 \cdot (1 - \cos(\tfrac{i\pi}{N+1}))) \,, \quad 1 \le i \le N$$

Mit der Reihenentwicklung für $\cos(x)$ folgt

$$1 - \cos(x) = 1 - (1 - \tfrac{x^2}{2} + \tfrac{x^4}{24} \mp \cdots) = \tfrac{x^2}{2} - \tfrac{x^4}{24} \pm \cdots \,, \quad x = \tfrac{i\pi}{N+1}$$

und deshalb

$$\lambda_1 \approx \frac{-k}{(\Delta x)^2} \cdot \frac{2\pi^2}{(N+1)^2 \cdot 2} = -k\pi^2$$

sowie

$$\lambda_N \approx -\frac{4k}{(\Delta x)^2} \,.$$

d.h. $\vec{v}^h$ geht mit t exponentiell gegen 0, der Abfall ist wie $\exp(-k\pi^2 t)$, aber es gibt auch extrem schnell abklingende Lösungskomponenten für die höheren λ_i.

Beispiel: Für $k = 1$ und $\Delta x = 0.01$ ist $\lambda_{100} \approx -40000$.

Wir sagen ein Differentialgleichungssystem ist **steif**, wenn die Zeitkonstanten λ_i in einem großen Bereich variieren, und die Lösungsmannigfaltigkeit des Systems Komponenten hat, deren Ableitungen sehr groß werden, obwohl die Lösung der Anfangswertaufgabe "glatt" ist.

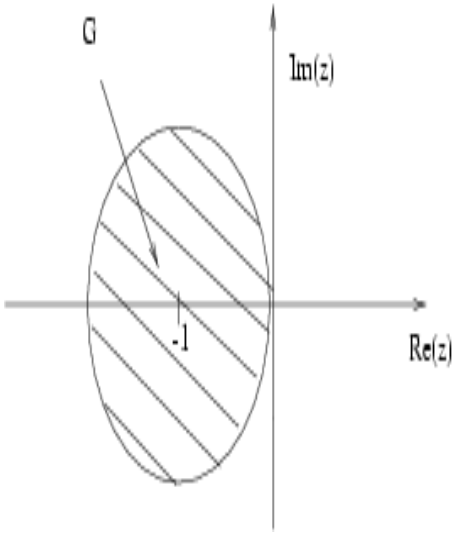
Bei solchen Differentialgleichungssystemen hat man beim numerischen Integrieren **Stabilitätsprobleme**. Die einfachste Methode zur Integration einer gewöhnlichen Differentialgleichung ist sicher das **explizite Euler-Verfahren** (siehe Kapitel 4)

$$\dot{\vec{y}} \;=\; F(t, \vec{y}) : \qquad \vec{y}^h_{N+1} \;=\; \vec{y}^h_N + h \cdot F(t_N, \vec{y}^h_N) \,.$$

und erhalten hier also mit $h = \Delta t$, $\vec{y} = \vec{v}^h$ die Vorschrift

$$\vec{v}^{h,\Delta t}_{N+1} = \vec{v}^{h,\Delta t}_N + \Delta t \cdot A \vec{v}^{h,\Delta t}_N \,.$$

Zur Erinnerung: das Gebiet der absoluten Stabilität des expliziten Euler-Verfahrens ist das folgende:



$\Delta t \lambda_i$ muß innerhalb des Stabilitätsgebietes liegen. Hier heißt das also $\lambda_N \cdot \Delta t > -2$. Und es folgt $\Delta t < \frac{2}{\frac{4k}{(\Delta x)^2}} = \frac{1}{2k} \cdot (\Delta x)^2$, d.h. wir müssen mit unnatürlich kleinen Zeitschritten integrieren, auch wenn die wahre Lösung u schon ganz glatt ist (bei Nullrandbedingungen praktisch null).

Für das **implizite Euler-Verfahren**, das in dieser Anwendung auch "vollimplizites Verfahren" heißt, haben wir die Verfahrensvorschrift:

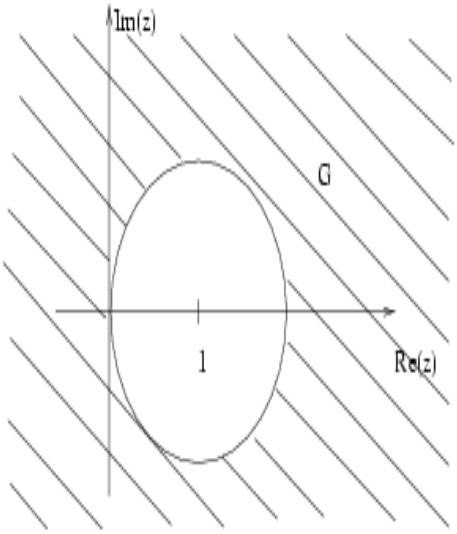
$$\vec{v}_{n+1}^{h,\Delta t} = \vec{v}_n^{h,\Delta t} + \Delta t \cdot A \vec{v}_{n+1}^{h,\Delta t} \, .$$

Da wir diese Vorschrift auch in der Form

$$(I - \Delta t \cdot A) \vec{v}_{n+1}^{h,\Delta t} = \vec{v}_n^{h,\Delta t}$$

schreiben können, ist hier ein lineares Gleichungssystem mit einer positiv definiten Tridiagonalmatrix $\tilde{A} := I - \Delta t \cdot A$ mit Eigenwerten ≥ 1 zu lösen.

Das **Gebiet der absoluten Stabilität** für dieses Verfahren hat folgendes Aussehen



Und konsequenterweise ist das Verfahren für **jedes** Δt absolut stabil.

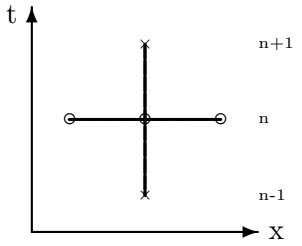
Hier ist eine **Schrittweitensteuerung** sinnvoll. Da $\vec{v}^h$ mit wachsendem t immer "glatter" wird, kann Δt dann immer mehr vergrößert werden.

Der Nachteil dieses Verfahrens besteht darin, daß die Konvergenz des obigen Verfahrens nur von der Ordnung 1 in Δt ist.

Der **Gesamtfehler** kann folgendermaßen abgeschätzt werden:

$$\frac{1}{\sqrt{N}} \cdot \left\| \vec{v}_n^{h,\Delta t} - \begin{pmatrix} u(x_1, n \cdot \Delta t) \\ \vdots \\ u(x_N, n \cdot \Delta t) \end{pmatrix} \right\|_2 \leq \text{ const } \cdot (\Delta t + (\Delta x)^2)$$

Eine naheliegende Idee wäre nun, folgende Formel zu verwenden $\vec{u}_t(x_i, t) = \frac{\vec{u}(x_i, t+\Delta t) - \vec{u}(x_i, t-\Delta t)}{2\Delta t} + \mathcal{O}((\Delta t)^2)$.



Man gelangt nun für die Differentialgleichung $\dot{y} = F(t, y)$ zur **expliziten Mittelpunktregel** der Form

$$y_{n+1}^h = y_{n-1}^h + 2\Delta t F(t_n, y_n^h) \,.$$

Leider hat dieses Verfahren **kein** Gebiet der absoluten Stabilität in der negativen Halbebene und ist somit hier **völlig unbrauchbar**.

Trapezregel:

Für $\dot{y} = F(t, y)$ hat die Trapezregel die Gestalt

$$y^h_{n+1} = y^h_n + \tfrac{h}{2} \cdot (F(t_n, y^h_n) + F(t_{n+1}, y^h_{n+1}))\,.$$

In unserem Fall hat man also ein lineares Gleichungssystem der Form

$$(I - \tfrac{\Delta t}{2} A) \cdot \vec{v}^{h,\Delta t}_{n+1} = (I + \tfrac{\Delta t}{2} A) \cdot \vec{v}^{h,\Delta t}_n\,.$$

zu lösen.

Die Trapezregel bezeichnet man in diesem Zusammenhang auch als **Crank-Nicholson-Verfahren**. Für den **Gesamtfehler** gilt nun

$$\tfrac{1}{\sqrt{N}} \cdot \left\| \vec{v}^{h,\Delta t}_n - \begin{pmatrix} u(x_1, n \cdot \Delta t) \\ \vdots \\ u(x_N, n \cdot \Delta t) \end{pmatrix} \right\|_2 \leq \text{const} \cdot \left((\Delta t)^2 + (\Delta x)^2 \right) \quad \text{für } n \cdot \Delta t \leq T\,.$$

Die Trapezregel ist bekanntlich A-stabil und ihre Verstärkungsfunktion ist

$$g(z) = \frac{1 + z/2}{1 - z/2}$$

wobei z für $h\lambda = \Delta t \lambda$ steht.

Aber: Hier ist $h \hat{=} \Delta t$, $\lambda \hat{=} \lambda_i$, $\lambda_i \in [-\frac{4k}{(\Delta x)^2}, \approx -k\pi^2]$ und nach unitärer Transformation

$$\vec{\omega}^{h,\Delta t} := V^T \cdot \vec{v}^{h,\Delta t}, \quad V = \text{ Eigenvektormatrix von } A$$

wird

$$\omega^{h,\Delta t}_{i,n+1} = \frac{1 + \frac{\Delta t}{2} \cdot \lambda_i}{1 - \frac{\Delta t}{2} \cdot \lambda_i} \cdot \omega^{h,\Delta t}_{i,n}, \qquad \text{für } i = N \text{ also } \frac{1 + \frac{\Delta t}{2} \cdot (-\frac{4k}{(\Delta x)^2})}{1 - \frac{\Delta t}{2} \cdot (-\frac{4k}{(\Delta x)^2})}$$

als Faktor ≈ -1, falls Δt nicht extrem klein ist. Dies bedeutet, daß die numerische Lösung von einer oszillierenden , sehr langsam abklingenden Komponente überlagert wird, was unphysikalisch ist. Es sollte $\frac{2\Delta t \cdot k}{(\Delta x)^2} < 1$ sein, damit diese Oszillationen keine Rolle spielen. Besser ist es, für diese Probleme das sogenannte BDF2-Verfahren zu benutzen. Man erhält es, indem man die Gitterfunktion $\vec{v}^{h,\Delta t}_n$ auf drei Zeitschichten $n+1,\ n,\ n-1$ durch eine Parabel zweiter Ordnung interpoliert, diese Parabel differenziert und den Wert der Ableitung an der Stelle t_{n+1} gleich der rechten Seite der Differentialgleichung setzt:

$$\vec{v}^{h,\Delta t}_{n+1} \;=\; \frac{4}{3}\vec{v}^{h,\Delta t}_n - \frac{1}{3}\vec{v}^{h,\Delta t}_{n-1} + \frac{2}{3}\frac{\Delta t}{(\Delta x)^2} k A \vec{v}^{h,\Delta t}_{n+1}\,.$$

Dieses Verfahren ist also auch implizit, hat die Ordnung 2 und ist A-stabil und L-stabil (d.h. die Stabilitätsfunktion $g(z)$ erfüllt $g(z) \to 0$ für $\text{Re}\,(z) \to -\infty$.)

Man kann bezüglich des Raumes (x) auch eine Semidiskretisierung mittels finiter Elemente durchführen. Man gelangt dann schliesslich zu einer gewöhnlichen Differentialgleichung für die Koeffizienten $\vec{a}(t)$ in dem Ansatz

$$u^h(x,t) \;=\; \sum_{i=1}^{N} a_i(t)\varphi_i(x)$$

der Form

$$M\dot{\vec{a}} \;=\; K\vec{a} + \vec{g}$$

mit den symmetrischen und positiv definiten Matrizen M und K, deren Elemente sich aus den L_2- bzw. Sobolev-Skalarprodukten der Ansatzfunktionen φ_i berechnen. Im Fall stückweise linearer stetiger Interpolation kann man M durch sogenanntes "lumping" durch eine Diagonalmatrix ersetzen, sonst muss man in jedem Zeitschritt lineare Gleichungssysteme mit M bzw. $M - \Delta t \cdot K$ lösen. (M wird niemals invertiert, weil man sonst die Dünnbesetztheit verlöre.) Weil man hier ohnehin stets implizit zeitdiskretisieren wird, ist dies aber keine besondere Erschwernis. Mit Ansätzen höheren Grades kann man dann leicht auch bessere Konsistenzordnungen als 2 erreichen.

Kapitel 11

Hyperbolische Differentialgleichungen

Wir beginnen mit typischen **Beispielen**.

1. Hyperbolische Einzeldifferentialgleichung 2. Ordnung:

   $$a\, u_{xx} + 2b\, u_{xy} + c\, u_{yy} = f(x, y, u, u_x, u_y) \quad \text{(Wellengleichung)}$$

   Für eine hyperbolische Differentialgleichung ist die Matrix $\left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$ indefinit, aber regulär. So könnten wir z.B. $a = 1$, $c = -1$ und $b = 0$ wählen.

2. Hyperbolische Systeme 1. Ordnung:

   $$\tfrac{\partial}{\partial y} u = A(x, y, u) \cdot \tfrac{\partial}{\partial x} u + g(x, y, u)$$

   Hierbei ist A eine reelle $n \times n$-Matrix mit n verschiedenen reellen Eigenwerten $\neq 0$.

Wir haben hier immer eine Kopplungsbedingung an die Diskretisierung der beiden Variablen. Der Grund dafür ist der Lösungscharakter der Aufgaben.

Sinnvolle Problemstellungen sind reine Anfangs- oder Anfangs-Randwertaufgaben.

Als Beispiel betrachten wir die **Wellengleichung** der Form

$$u_{tt} = c^2 u_{xx}, \text{ für } x \in \mathbb{R}, t > 0$$

mit der Wellenausbreitungsgeschwindigkeit c. Als Anfangswerte wählen wir $u(x, 0) = f(x),\ u_t(x, 0) = g(x)$. Die allgemeine Lösung läßt sich nach d'Alembert angeben :

$$u(x, t) = \tfrac{1}{2}(f(x + ct) + f(x - ct)) + \tfrac{1}{2c} \cdot \int_{x-ct}^{x+ct} g(\tau) \mathrm{d}\tau\,.$$

Diese Lösungsformel besagt, daß die Lösung u an der Stelle (x,t) nur von den Anfangsvorgaben auf dem Intervall $[x-ct, x+ct]$ abhängt, dem sogenannten analytischen Abhängigkeitsbereich. Ein weiteres Beispiel bildet die **Konvektionsgleichung**

$$u_t = a \cdot u_x, \text{ für } x \in \mathbb{R}, t > 0\,.$$

Anfangswerte seien hier durch $u(x,0) = h(x)$ mit $x \in \mathbb{R}$ und einer Funktion $h \in C^1(\mathbb{R})$ gegeben. Die Lösung ist hier von der Form $u(x,t) = h(x+at)$.

Die sogenannten **Charakteristiken** sind hier von der Form $at + x = \text{const}$. Auf einer Charakteristik gilt $u(x,t) = h(x+at) = h(\text{const}) = \text{const}$, d.h. die Lösung $u(x,t)$ ist auf einer Charakteristik konstant und der analytische Abhängigkeitsbereich ist ein einziger Punkt. Ist also der Anfangswert für u auf der $x-$Achse als eine Funktion u_0 gegeben, dann wird $h = u_0$.

Man kann die Wellengleichung auf ein hyperbolisches System erster Ordnung und anschliessende Quadratur auf folgende Art und Weise zurückführen:

Wir setzen $\vec{v} = \binom{v_1}{v_2} := \binom{u_t}{u_x}$ und erhalten mit

$$\tfrac{\partial}{\partial t}\vec{v} = \begin{pmatrix} u_{tt} \\ u_{tx} \end{pmatrix} \text{ und } \tfrac{\partial}{\partial x}\vec{v} = \begin{pmatrix} u_{xt} \\ u_{xx} \end{pmatrix}$$

ein hyperbolisches System der Form

$$\tfrac{\partial}{\partial t}\vec{v} = \begin{pmatrix} 0 & c^2 \\ 1 & 0 \end{pmatrix} \cdot \tfrac{\partial}{\partial x}\vec{v}\,,$$

da $u_{xt} = u_{tx}$ für $u \in C^2(\mathbb{R}^2)$ gilt. Ist $\vec{v}$ bekannt, kann man dann u durch Quadratur berechnen. Eine **Entkopplung** dieses Systems in zwei Konvektionsgleichungen kann man auf folgende Art und Weise durchführen.

Die Eigenwerte der Matrix $\binom{0\ c^2}{1\ 0}$ ergeben sich aus der Bedingung $\lambda^2 - c^2 = 0$ als $\lambda_{1,2} = \pm c$. Das heißt wiederum, daß die Steigungen der Charakteristiken $\frac{1}{\lambda_i}$, $i = 1,2$ sind, also $\pm\frac{1}{c}$.

Als Eigenvektoren haben wir

$$\begin{pmatrix} c \\ 1 \end{pmatrix} \quad \text{wegen} \quad \begin{pmatrix} 0 & c^2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} c \\ 1 \end{pmatrix} = \begin{pmatrix} c^2 \\ c \end{pmatrix} = c \cdot \begin{pmatrix} c \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} -c \\ 1 \end{pmatrix} \quad \text{wegen} \quad \begin{pmatrix} 0 & c^2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} -c \\ 1 \end{pmatrix} = \begin{pmatrix} c^2 \\ -c \end{pmatrix} = -c \cdot \begin{pmatrix} -c \\ 1 \end{pmatrix}$$

Als Transformationsmatrix wählen wir nun

$$T = \begin{pmatrix} c & -c \\ 1 & 1 \end{pmatrix}$$

und erhalten insgesamt

$$T^{-1}AT = \begin{pmatrix} c & 0 \\ 0 & -c \end{pmatrix}.$$

Setzen wir nun $\vec{\omega} := T^{-1}\vec{v}$ so ergibt sich ein entkoppeltes System der Form

$$\frac{\partial}{\partial t}\vec{\omega} = \begin{pmatrix} c & 0 \\ 0 & -c \end{pmatrix} \cdot \frac{\partial}{\partial x}\vec{\omega}$$

beziehungsweise

$$\begin{aligned} \frac{\partial}{\partial t}\omega_1 &= c \cdot \frac{\partial}{\partial x}\omega_1\,, \\ \frac{\partial}{\partial t}\omega_2 &= -c \cdot \frac{\partial}{\partial x}\omega_2\,. \end{aligned}$$

Durch jeden Punkt $\binom{x}{t}$ gehen zwei Charakteristiken der Wellengleichung mit den Steigungen $\frac{1}{c}$ bzw. $-\frac{1}{c}$.

Analog kann man nun für ein System $\frac{\partial}{\partial y}\vec{u} = A \cdot \frac{\partial}{\partial x}\vec{u}$ (mit $A \in \mathbb{R}^{n\times n}$ fest) vorgehen.

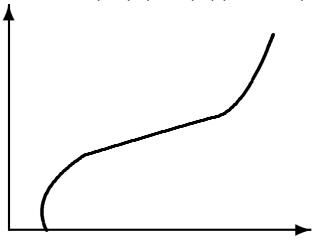
Hat A n verschiedene reelle Eigenwerte, dann ist das System entkoppelbar mittels der Eigenvektoren. Es gibt dann n Charakteristiken mit den Steigungen $\frac{1}{\lambda_i}$ wobei $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von A sind.

Die Charakteristiken spielen u.a. bei der Betrachtung der Existenz- und Eindeutigkeitsfrage für die Lösung einer hyperbolischen AWA eine Rolle. Sind die Anfangswerte auf einer Kurve Γ in der (x,t)-Ebene vorgegeben, die in keinem Punkt eine Charakteristik tangiert, so ist die Anfangswertaufgabe lokal eindeutig lösbar.

Als **Beispiel** betrachten wir

$$u_t = \frac{\partial}{\partial x}(\varphi(u(x,t))) \text{ mit } \varphi \in C^1(\mathbb{R}), 0 < \varphi_0 < -\varphi'(z) < \varphi_1$$

und $u(x(s),t(s)) = g(s)$ auf Γ gegeben. Bei der Konvektionsgleichung gilt $\varphi(z) = az$.



Sei nun s_0 gegeben. Gesucht ist u in einer Umgebung von $\binom{x(s_0)}{t(s_0)}$. Wir nehmen an, daß $u(x(s_0),t(s_0))$ bekannt sei!

Gesucht ist dann eine **Reihenentwicklung** für $u(x,t)$ der Form

$$u(x,t) = \underbrace{u(\underbrace{x(s_0)}_{=:x_0}, \underbrace{t(s_0)}_{=:t_0})}_{=:P_0} + u_x(P_0)\cdot(x-x_0) + u_t(P_0)\cdot(t-t_0) + \cdots$$

Wir bilden nun

$$\frac{\partial}{\partial s}\Big(u(x(s),t(s))\Big) \;=\; u_x \cdot x' + u_t \cdot t' = g'(s)$$

und erhalten mit der Differentialgleichung

$$u_t - \varphi'(u) \cdot u_x \;=\; 0$$

das System

$$\begin{pmatrix} x' & t' \\ -\varphi'(u) & 1 \end{pmatrix} \begin{pmatrix} u_x \\ u_t \end{pmatrix} \;=\; \begin{pmatrix} g'(s) \\ 0 \end{pmatrix}$$

Unter der Voraussetzung, daß Γ nicht charakteristisch ist, d.h. der Tangentenvektor von Γ hat mit der Normalen zur Charakteristik ein Skalarprodukt $\neq 0$, d.h. $x' + \varphi'(u) \cdot t' \neq 0$ (bei der Konvektionsgleichung mit $\varphi(z) = az$ entsprechend $x' + at' \neq 0$), kann man nun u_x und u_t auf Γ berechnen, wenn dort u gegeben ist. Mit diesen Werten kann man dann auch alle weiteren Ableitungen u_{xx} usw. bestimmen, indem man obiges 2×2-System weiter differenziert. D.h. das Anfangswertproblem ist in diesem Fall eindeutig lösbar, wobei die Lösung zumindest in einer kleinen Umgebung von Γ existiert. (Natürlich muß man noch die Konvergenz der Potenzreihe beweisen, was aber möglich ist. Dabei wird die Analytizität von Γ, φ und g vorausgesetzt.)

Wir wenden uns im Folgenden der **Diskretisierung** hyperbolischer Probleme zu.

Als erstes Modellproblem betrachten wir die **Konvektionsgleichung**

$$u_t \;=\; au_x \text{ mit } a < 0 \text{ für } 0 \leq x \leq x_l$$

mit Anfangsbedingung

$$u(x,0) \;=\; h(x) \text{ für } 0 \leq x \leq x_l \text{ mit } h(0) = 0$$

und der Randbedingung

$$u(0,t) \;=\; 0\,.$$

Die Lösung ist von der Form $u(x,t) = h(x+at)$. $a < 0$ bedeutet einen Transport nach rechts. Dadurch sind wir in der Lage, die Lösung tatsächlich auf ganz $[0, x_l] \times \mathbb{R}_+$ zu berechnen. Für variables a hat man keine geschlossene Lösung. Wir nehmen an, daß wir die Lösung noch nicht kennen, und legen zunächst ein Gitter mit Gitterpunkten (x_i, t_j) und $x_i = i \cdot \Delta x$, $i \in \mathbb{Z}$ sowie $t_j = j \cdot \Delta t$, $j \in \mathbb{N} \cup \{0\}$ fest.

Die **Courant-Friedrichs-Levy (CFL)**–Bedingung lautet folgendermaßen:

Das numerische Abhängigkeitsintervall muß das (von den Charakteristiken bestimmte) analytische Abhängigkeitsintervall umfassen (hier nur der Punkt $[x_i + at_j]$). Das numerische Abhängigkeitsintervall ist dabei die bestimmt durch alle Gitterpunkte $(x_i, 0)$,

von denen der berechnete Wert $u^h_{i,j}$ abhängt. An diesem Beispiel sieht man sofort, daß andersfalls Konvergenz nicht eintreten kann. Wir verwenden die Entwicklungen

$$\begin{aligned} u_t(x_i,t_j) &= \frac{u(x_i,t_{j+1})-u(x_i,t_j)}{\Delta t}+\mathcal{O}(\Delta t)\,, \\ u_x(x_i,t_j) &= \begin{cases} \frac{u(x_i,t_j)-u(x_{i-1},t_j)}{\Delta x}+\mathcal{O}(\Delta x) & \text{für } a<0 \\ \frac{u(x_{i+1},t_j)-u(x_i,t_j)}{\Delta x}+\mathcal{O}(\Delta x) & \text{für } a>0 \end{cases} \end{aligned}$$

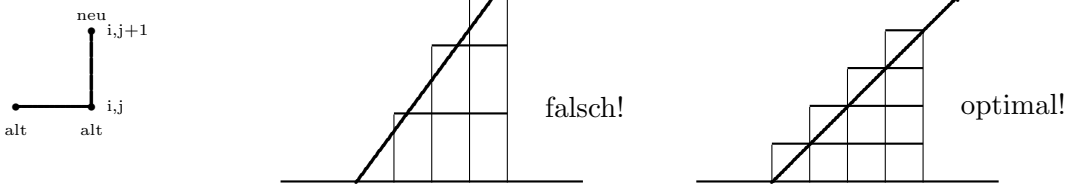
Um die CFL-Bedingung zu erfüllen, dieskretisieren wir

$$\begin{aligned} u^h_{i,j+1} &= u^h_{i,j}+a\cdot\tfrac{\Delta t}{\Delta x}\cdot(u^h_{i,j}-u^h_{i-1,j}) \quad (a<0)\,, \\ u^h_{i,j+1} &= u^h_{i,j}+a\cdot\tfrac{\Delta t}{\Delta x}\cdot(u^h_{i+1,j}-u^h_{i,j}) \quad (a>0)\,. \end{aligned}$$

mit der Einhaltung der Bedingung

$$\frac{\Delta t}{\Delta x} \le \frac{1}{|a|}\,.$$

Für $a<0$ sieht das graphisch so aus:



Ideal wäre $\left|\frac{a\cdot\Delta t}{\Delta x}\right|=1 \quad \left(\frac{a\cdot\Delta t}{\Delta x}=-1\right)$, denn dann wäre

$$u^h_{i,j+1}=u^h_{i-1,j}=\cdots=u^h_{i-j-1,0}=u_{i-j-1,0}$$

und das Verfahren damit exakt. (Dies funktioniert natürlich nur in diesem einfachen Fall.) Obige Diskretisierung kann man offenbar auch bei variablem $a=a(x)$ benutzen, wobei nun die CFL-Begingung mit $1/\sup\{|a(x)|\}$ einzuhalten ist.

Ein **scheinbar besseres** Verfahren arbeitet mit den Approximationen

$$\begin{aligned} u_t(x_i,t_j) &\approx \frac{u(x_i,t_{j+1})-u(x_i,t_j)}{\Delta t}\,, \\ u_x(x_i,t_j) &\approx \frac{u(x_{i+1},t_j)-u(x_{i-1},t_j)}{2\Delta x} \quad \text{(Fehler: } \mathcal{O}((\Delta x)^2) \end{aligned}$$

und lautet

$$u^h_{i,j+1} = u^h_{i,j}+a\cdot\frac{\Delta t}{2\Delta x}\cdot(u^h_{i+1,j}-u^h_{i-1,j})\,.$$

Wichtig ist folgende

Bemerkung 11.0.2. *Die CFL-Bedingung ist (im Zusammenhang mit Konsistenz) notwendig, aber nicht hinreichend für die Konvergenz des Verfahrens.*

Dieses Verfahren ist instabil und daher nicht konvergent! Ersetzt man aber darin $u^h_{i,j}$ durch den Mittelwert von $u^h_{i-1,j}$ und $u^h_{i+1,j}$, dann gelangt man zum **Verfahren von Friedrichs**

$$u^h_{i,j+1} = \tfrac{1}{2}(u^h_{i+1,j} + u^h_{i-1,j}) + a \cdot \tfrac{\Delta t}{2\Delta x}(u^h_{i+1,j} - u^h_{i-1,j})$$

Dies ist konvergent von 1. Ordnung, d.h. es gilt

$$|u(x_i, t_j) - u^h_{i,j}| \leq c(T) \cdot \Delta x,$$

falls $\left| a \cdot \frac{\Delta t}{\Delta x} \right| \leq 1, \quad |j \cdot \Delta t| \leq T$ gilt. c hängt auch noch von den höheren Ableitungen von u ab. Diese Aussage gilt auch bei Anwendung der Verfahrens auf Systeme der Form

$$\vec{u}_t \;=\; A(x) \cdot \vec{u}_x, \quad \vec{u}(x,0) = g(x) \tag{11.1}$$

mit einer symmetrischen Matrix $A \in \mathbb{R}^n \times \mathbb{R}^n$.

Das Verfahren von Friedrichs lautet in diesem Fall

$$\vec{u}^h_{i,j+1} = \tfrac{1}{2}(\vec{u}^h_{i+1,j} + \vec{u}^h_{i-1,j}) + \frac{\Delta t}{2\Delta x} \cdot A(x_i) \cdot (\vec{u}^h_{i+1,j} - \vec{u}^h_{i-1,j})$$

und ist konvergent von 1. Ordnung in Δx, falls $\|A(x)\| \cdot \frac{\Delta t}{\Delta x} \leq 1$ für alle x gilt.

Das **Lax-Wendroff-Richtmyer-Verfahren** für ein System der Gestalt (11.1) lautet

$$\begin{aligned} \vec{u}^h_{i,j+1} \;=\; & \vec{u}^h_{i,j} + \tfrac{\Delta t}{2\Delta x} A(x_i) \cdot (\vec{u}^h_{i+1,j} - \vec{u}^h_{i-1,j}) \\ & + \tfrac{1}{2}\left(\tfrac{\Delta t}{\Delta x}\right)^2 \cdot A(x_i) \cdot (A(x_i + \tfrac{\Delta x}{2}) \cdot (\vec{u}^h_{i+1,j} - \vec{u}^h_{i,j}) - A(x_i - \tfrac{\Delta x}{2}) \cdot (\vec{u}^h_{i,j} - \vec{u}^h_{i-1,j})) \end{aligned}$$

Dieses Verfahren ist für glatte Lösungen konvergent von zweiter Ordnung mit einem Fehler der Form $c(T) \cdot (\Delta x)^2$, falls $\|A(x)\| \cdot \frac{\Delta t}{\Delta x} \leq 1$ für alle x und $|j \cdot \Delta t| \leq T$ gilt.

Als zweites Modellproblem wenden wir uns nun der direkten Diskretisierung der **Wellengleichung** zu. Dieses Problem läßt sich so beschreiben:

$$u_{tt} = c^2 \cdot u_{xx}$$

mit den Anfangsbedingungen

$$u(x,0) = f(x) \text{ und } u_t(x,0) = g(x) \text{ für } 0 \leq x \leq 1$$

sowie im einfachsten Fall mit den Randbedingungen

$$u(0,t) = u(1,t) = 0 \text{ für } t > 0\,.$$

Die **allgemeine Lösung** lautet dann im Falle $g(x) \equiv 0$

$$u(x,t) = \tfrac{1}{2} \cdot (f_{\text{per}}(x-ct) + f_{\text{per}}(x+ct)) \ .$$

wobei f_{per} die ungerade periodische Fortsetzung von f ist.

Wir approximieren nun mithilfe von

$$\begin{aligned} u_{tt}(x_i,t_j) &\approx \tfrac{1}{(\Delta t)^2}\Big(u(x_i,t_{j+1}) - 2u(x_i,t_j) + u(x_i,t_{j-1})\Big) \quad \text{Fehler: } \mathcal{O}((\Delta t)^2) \\ u_{xx}(x_i,t_j) &\approx \tfrac{1}{(\Delta x)^2}\Big(u(x_{i+1},t_j) - 2u(x_i,t_j) + u(x_{i-1},t_j)\Big) \quad \text{Fehler: } \mathcal{O}((\Delta x)^2) \end{aligned}$$

Einsetzen in $u_{tt} = c^2 \cdot u_{xx}$ liefert nun

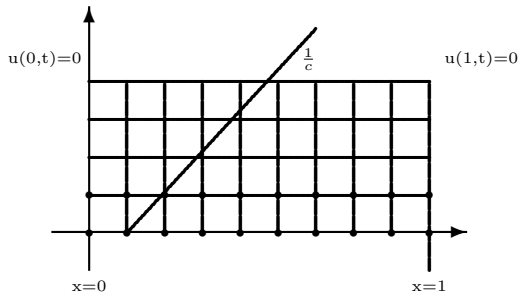
$$u^h_{i,j+1} = 2u^h_{i,j} - u^h_{i,j-1} + \left(\tfrac{c\Delta t}{\Delta x}\right)^2 \cdot (u^h_{i+1,j} - 2u^h_{i,j} + u^h_{i-1,j}) \, .$$

Weiterhin verwenden wir

$$\begin{aligned} j = 0: &\quad u^h_{i,0} = u(x_i,0) = f(x_i) \quad \text{(Anfangsauslenkung)} \\ j = 1: &\quad u^h_{i,1} = u^h_{i,0} + \Delta t \cdot u_t(x_i,0) \quad \text{Fehler: } \mathcal{O}((\Delta t)^2) \, . \end{aligned}$$

wobei die Anfangsgeschwindigkeit $u_t(x,0) = g(x)$ gegeben ist.

Die CFL-Bedingung liefert die Forderung $|c\frac{\Delta t}{\Delta x}| \leq 1$.



Das Verfahren konvergiert von zweiter Ordnung in Δx, falls $|c\frac{\Delta t}{\Delta x}| \leq 1$ gilt. Wie bei der Wärmeleitungsgleichung kann man auch bei diesem Randanfangswertproblem die Methode der **Semidiskretisierung** anwenden. Als Modellproblem betrachten wir wieder die **schwingende Saite**:

$$u_{tt} = c^2 \cdot u_{xx} \text{ mit } u(x,0) = f(x) \text{ und } u_t(x,0) = g(x) \text{ für } 0 \leq x \leq 1$$

und

$$u(0,t) = u(1,t) = 0 \text{ für alle } t > 0 \text{ sowie } f(0) = f(1) = 0.$$

Falls $g(x) \equiv 0$ ist, dann gilt

$$u(x,t) = \tfrac{1}{2}(f_{\text{per}}(x+ct) + f_{\text{per}}(x-ct))\,,$$

wobei f_{per} die ungerade periodische Fortsetzung von f ist (siehe oben). Mit der Fourierentwicklung der Anfangsauslenkung f in der Form

$$f(x) = \sum_{j=1}^{\infty} a_j \cdot \sin(\pi j x)$$

folgt nun für beliebiges $t > 0$

$$u(x,t) = \tfrac{1}{2} \sum_{j=1}^{\infty} a_j \cdot \Big(\sin(\pi j(x+ct)) + \sin(\pi j(x-ct))\Big)\,.$$

D.h. die Lösung ist eine ungedämpfte Schwingung. Hier wird ein Integrationsverfahren benötigt, das die Amplituden einer Schwingung weder verstärkt noch dämpft. Es sollte (**"neutral stabil"**) sein. Es darf aber auch keine großen Phasenfehler erzeugen.

Wir führen zunächst eine **Semidiskretisierung** durch und setzen

$$\vec{u} \;:=\; \begin{pmatrix} u(x_1,t) \\ \vdots \\ u(x_N,t) \end{pmatrix}, \qquad x_i = i \cdot \Delta x, \quad 0 \le i \le N+1, \quad \Delta x = \tfrac{1}{N+1}\,.$$

sowie

$$\ddot{\vec{u}}(t) \;:=\; \begin{pmatrix} u_{tt}(x_1,t) \\ \vdots \\ u_{tt}(x_N,t) \end{pmatrix}$$

wobei $u_{tt}(x_i,t) \approx c^2 \cdot \frac{1}{(\Delta x)^2} \cdot (u(x_{i+1},t) - 2u(x_i,t) + u(x_{i-1},t))$.

Dann folgt also

$$\ddot{\vec{u}}^h \;=\; A\vec{u} + \underbrace{\tfrac{c^2}{(\Delta x)^2} \begin{pmatrix} u(x_0,t) \\ 0 \\ \vdots \\ 0 \\ u(x_{N+1},t) \end{pmatrix}}_{=:\vec{h}(t)} \qquad t > 0$$

mit den Anfangswerten

$$\vec{u}^h(0) = \begin{pmatrix} u(x_1,0) \\ \vdots \\ u(x_N,0) \end{pmatrix} \text{ und } \dot{\vec{u}}^h(0) = \begin{pmatrix} u_t(x_1,0) \\ \vdots \\ u_t(x_N,0) \end{pmatrix}\,.$$

Mit $y := \vec{u}^h$ haben wir nun die Form einer speziellen gewöhnlichen DGL zweiter Ordnung

$$y'' \;=\; f(t,y), \qquad y(0)=y_0, \quad y'(0)=y_0' \text{ bekannt}\,.$$

Als **spezielle Integrationsverfahren** bieten sich also Lösungsverfahren für diesen Typ an. Die einfachste Version ist das **Verfahren von Störmer:**

$$y_{n+1}^h = 2y_n^h - y_{n-1}^h + (\Delta t)^2 \cdot f(t_n, y_n^h)\,.$$

Dies ergibt in unserem Fall wieder die obige Standard-Diskretisierung.

Eine weitere Möglichkeit ist eine **Transformation auf ein System 1. Ordnung:**

Mit den Setzungen

$$\begin{aligned} \vec{v}^h &:= \dot{\vec{u}}^h\,, \\ \dot{\vec{v}}^h &:= \ddot{\vec{u}}^h = A\vec{u}^h + \vec{h}(t)\,, \\ \vec{\omega}^h &:= \begin{pmatrix} \vec{u}^h \\ \vec{v}^h \end{pmatrix} \end{aligned}$$

erhalten wir das System erster Ordnung

$$\dot{\vec{\omega}}^h \;=\; \begin{pmatrix} \dot{\vec{u}}^h \\ \dot{\vec{v}}^h \end{pmatrix} = \begin{pmatrix} 0 & I \\ A & 0 \end{pmatrix} \begin{pmatrix} \vec{u}^h \\ \vec{v}^h \end{pmatrix} + \begin{pmatrix} \vec{0} \\ \vec{h}(t) \end{pmatrix}$$

mit den bekannten Anfangswerten

$$\vec{\omega}^h(0) \;=\; \begin{pmatrix} \vec{u}^h(0) \\ \vec{v}^h(0) \end{pmatrix}\,.$$

Die **Lösung** dieser semidiskretisierten Gleichung ist nun durch die Eigenwerte und Eigenvektoren von $\left(\begin{smallmatrix} 0 & I \\ A & 0 \end{smallmatrix}\right)$ gegeben. Aus

$$\begin{pmatrix} 0 & I \\ A & 0 \end{pmatrix} \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \lambda \cdot \begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix}$$

folgt nun

$$\vec{y} = \lambda\vec{x} \text{ und } A\vec{x} = \lambda\vec{y} = \lambda^2\vec{x}\,.$$

Also gilt $\lambda = \pm\sqrt{\mu}$ mit μ als Eigenwerten von A, wobei

$$A = \tfrac{c^2}{(\Delta x)^2} \cdot \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{pmatrix}$$

Die Eigenwerte von A lauten aber

$$\mu_i = \frac{c^2}{(\Delta x)^2}(-2(1-\cos\frac{i\pi}{N+1}))\ ,$$

sie sind reell und < 0. Also ist $\lambda = \pm\sqrt{\mu}$ rein imaginär, was bedeutet, daß es sich um eine ungedämpfte Schwingung mit hochfrequenten Anteilen handelt.

Die **Modellgleichung** für das entkoppelte System ist jetzt also

$$z' = \mathrm{i}\omega z, \quad z(0) = z_0 \qquad \text{mit der Lösung: } z = z_0 \cdot e^{\mathrm{i}\omega t} \quad \omega \in \mathbb{R}$$

Das Diskretisierungsverfahren für die semidiskretisierte Gleichung muß neutral stabil sein, d.h. **der Rand des Gebietes** der absoluten Stabilität muß ein Stück um den Nullpunkt auf der imaginären Achse als Teil besitzen.

Man kann also über die Verwendbarkeit einiger Verfahren folgende Aussagen treffen:

Euler implizit:	ungeeignet	
Trapezregel:	geeignet	$\Delta t\|\lambda\|$ beliebig,
Euler explizit:	ungeeignet	
Runge-Kutta 4. Ordnung:	bedingt geeignet	$(\Delta t\|\lambda\| \leq 1)$.

Es ist aus dieser Liste also nur die Trapezregel problemlos anwendbar, aber großes Δt ergibt hier große Phasenfehler, sowohl bei der Trapezregel wie bei anderen neutral stabilen Verfahren, sodaß man schliesslich auch hier auf eine Schrittweitenkopplung zwischen Δt und Δx zurückkommt.

Man kann bezüglich des Raumes (x) auch eine Semidiskretisierung mittels finiter Elemente durchführen. Man gelangt dann schliesslich zu einer gewöhnlichen Differentialgleichung für die Koeffizienten $\vec{a}(t)$ in dem Ansatz

$$u^h(x,t) \ = \ \sum_{i=1}^{N} a_i(t)\varphi_i(x)$$

der Form

$$M\ddot{\vec{a}} \ = \ K\vec{a} + \vec{g}$$

mit den symmetrischen und positiv definiten Matrizen M und K, deren Elemente sich aus den L_2- bzw. Energie-Skalarprodukten der Ansatzfunktionen φ_i berechnen. Im Fall stückweise linearer stetiger Interpolation kann man M durch sogenanntes "lumping" durch eine Diagonalmatrix ersetzen, sonst muss man in jedem Zeitschritt lineare Gleichungssysteme mit M bzw. $M - \Delta t \cdot K$ lösen. (M wird niemals invertiert, weil man sonst die Dünnbesetztheit verlöre.) Mit Ansätzen höheren Grades kann man dann leicht auch bessere Konsistenzordnungen als 2 erreichen. Es ist nicht sinnvoll, hier auf ein System 1. Ordnung umzuschreiben. Es gibt spezielle Integratoren für diesen Fall, etwa das Verfahren von Newmark, hier sogleich für den allgemeineren Fall formuliert:

$$M\ddot{\vec{y}} + C\dot{\vec{y}} + Kvy = \vec{f}(t) \qquad \begin{array}{l} M, C, K \text{ symm. } n \times n \text{ Matrizen,} \\ M, K \quad \text{pos.def.} \end{array}$$

(die man u.a. bei der Semidiskretisierung einer linearen hyperbolischen DGL 2. Ordnung mit einem Term u_t erhält)

$$\begin{aligned}(M+\gamma\Delta tC+\beta\Delta t^2K)\vec{u}^h_{n+1}+(-2M+(1-2\gamma)\Delta tC+(\tfrac{1}{2}+\gamma-2\beta)(\Delta t)^2K)\vec{u}^h_n\\ +(M+(\gamma-1)\Delta C+(\tfrac{1}{2}-\gamma+\beta)(\Delta t)^2K)\vec{u}^h_{n-1}\\ =(\Delta t)^2\Big((\tfrac{1}{2}-\gamma+\beta)\vec{f}(t_{n-1})+(\tfrac{1}{2}+\gamma-2\beta)\vec{f}(t_n)+\beta\vec{f}(t_{n+1})\Big)\end{aligned}$$

Im Fall $C=0$ und $2\beta\geq\gamma\geq\frac{1}{2}$ ist die Methode uneingeschränkt neutral stabil.

□

Kapitel 12

Die Methode der finiten Volumen

Die bisher besprochenen Methoden zur numerischen Lösung partieller Differentialgleichungen kann man kurz so zusammenfassen: Bei der Methode der finiten Differenzen (FD) werden die Differentialgleichung und die Randbedingungen nur noch auf einem diskreten Gitter betrachtet. Die auftretenden Ableitungswerte werden durch Differenzenausdrücke (also durch Ableitungswerte von Interpolationspolynomen) angenähert. Diese Näherungsformeln werden in die Differentialgleichung und die Randbedingungen eingesetzt. Es entsteht ein Gleichungssystem für Näherungswerte für die Werte der gesuchten Lösung auf dem Gitter. Dieses Gleichungssystem ist linear oder nichtlinear je nachdem ob die Differentialgleichung und/oder die Randwerte die Lösung u nur linear oder auch nichtlinear enthalten. Bei der Methode der finiten Elemente (FEM) wird das Differentialgleichungsproblem zunächst in ein Integralkriterium überführt (Differentialgleichung als notwendiges und eventuell auch hinreichendes Extremalkriterium eines Variationsproblems oder allgemeiner schwache Form des Problems). Dabei wird über das gesamte Gebiet integriert, auf dem die DGL gelten soll. Sodann wird die kontinuierliche Lösung u durch eine finite Linearkombination gewählter Ansatzfunktionen approximiert und diese Approximation wird anstelle von u benutzt. Dies erfordert gewisse Glattheitseigenschaften des Ansatzes, damit die Integrale überhaupt wohldefiniert sind. Die Integrale werden dann analytisch oder numerisch berechnet. Es bleibt ein lineares oder nichtlineares Gleichungssytem für die Koeffizienten dieser Linearkombination.

Die Methode der finiten Volumen nimmt eine Mittelstellung zwischen diesen beiden Ansätzen ein. Sie wird aber ausschliesslich für Differentialgleichungen in Divergenzform benutzt. Man zerlegt zunächst den Bereich in kleine Teilbereiche und integriert die Differentialgleichung über diese. Das Integral über die Divergenz wird mittels des Gauss'schen Satzes in ein Linienintegral der Normalableitung über den Rand der Teilbereiche umgeformt. Dieses Linienintegal wird durch eine Quadraturformel angenähert. Dabei werden Werte der Normalableitung der gesuchten Lösung an gewissen Stellen an diesen Rändern benötigt. Diese Werte werden wiederum durch Differenzenausdrücke mit Werten der Lösung auf Punkten im Innern der Teilbereiche ausgedrückt. Bei den

Randbedingungen diskretisiert man, sofern nicht schon die natürlichen Randbedingungen vorliegen, wie bei der Differenzenmethode. Es entsteht ein lineares oder nichtlineares Gleichungssystem für Näherungswerte der Lösung auf dem Gitter der inneren Punkte. Man gewinnt zwei Vorteile: Die Gebietseinteilung ist wesentlich flexibler handhabbar als bei der Differenzenmethode und die diskrete Lösung "erbt" automatisch gewisse Erhaltungseigenschaften der kontinuierlichen Lösung, z.B. die Erhaltung des totalen Flusses. Da die Approximation der Normalableitung gefordert ist, ist man aber daran gebunden, die Verbindungslinien zwischen den inneren Punkten auf den Teilbereichen orthogonal zu den Rändern dieser Teilbereiche zu wählen. Die Konvergenztheorie der Methode ist noch nicht so weit entwickelt wie bei FD oder FEM. Die Methode erfreut sich jedoch insbesondere in der numerischen Strömungsmechanik grosser Beliebtheit. Ein kleines Beispiel soll die Methode erläutern. Wir betrachten wiederum die Poissongleichung auf dem Einheitsquadrat mit homogenen Dirichletranddaten:

$$-\Delta\, u \;=\; f \quad \text{auf }]0,1[\times]0,1[$$

mit

$$u = 0 \;\; \text{falls } x \in \{0,1\} \;\; \text{oder } y \in \{0,1\}\ .$$

Sei

$$Q \;=\; [0,1]\times[0,1]\ .$$

Wir überdecken Q mit einem um $h/2$ versetzten quadratischen Gitter mit den Gitterpunkten $(\tilde{x}_i, \tilde{y}_j)$ mit

$$\tilde{x}_i \;=\; -h/2 + ih\ , \tilde{y}_j \;=\; -h/2 + jh\ , 0 \le i,j \le N+1$$

und $h = 1/N$. Als Teilbereiche wählen wir die Quadrate

$$Q_{i,j} \;=\; [\tilde{x}_i, \tilde{x}_{i+1}]\times[\tilde{y}_j, \tilde{y}_{j+1}]\ .$$

Als uns interessierendes Gitter der inneren Punkte nehmen wir die Mittelpunkte (Schwerpunkte) der $Q_{i,j}$, also $x_i = ih,\ y_j = jh,\ i,j = 0,\ldots,N$. Wegen der DGL ist

$$\int_{Q_{i,j}} f\mathrm{d}x\mathrm{d}y \;=\; -\int_{Q_{i,j}} \mathrm{div}(\mathrm{grad}u)\mathrm{d}x\mathrm{d}y \;=\; -\int_{\partial Q_{i,j}} \mathrm{grad}u\cdot\vec{n}\mathrm{d}s$$

Für das Integral auf der linken Seite benutzen wir die Schwerpunktregel

$$\int_{Q_{i,j}} f\mathrm{d}x\mathrm{d}y \;\approx\; h^2 f_{i,j}\ .$$

Für das Linienintegral benutzen wir die zusammengesetzte Rechteckregel

$$\int_{\partial Q_{i,j}} \mathrm{grad}u\cdot\vec{n}\mathrm{d}s \approx\; h\Big(-(u_y)_{i,j-\frac{1}{2}} + (u_x)_{i+\frac{1}{2},j} + (u_y)_{i,j+\frac{1}{2}} - (u_x)_{i-\frac{1}{2},j}\Big)\ .$$

Für die partiellen Ableitungen an den Kantenmittelpunkten, also den um $h/2$ versetzten Gitterpunkten, nehmen wir den zentralen Differenzenquotienten als Näherung

$$\begin{aligned}
-(u_y)_{i,j-\frac{1}{2}} &= -(u_{i,j} - u_{i,j-1})/h + \mathcal{O}(h^2) \ , \\
(u_x)_{i+\frac{1}{2},j} &= (u_{i+1,j} - u_{i,j})/h + \mathcal{O}(h^2) \ , \\
(u_y)_{i,j+\frac{1}{2}} &= (u_{i,j+1} - u_{i,j})/h + \mathcal{O}(h^2) \ , \\
-(u_x)_{i-\frac{1}{2},j} &= (u_{i,j} - u_{i-1,j})/h + \mathcal{O}(h^2) \ .
\end{aligned}$$

Dies in die Rechtecksregel eingesetzt und Ausnutzung der bekannten Randwerte $u_{0,j} = u_{N+1,j} = u_{i,0} = u_{i,N+1} = 0,\ i,j = 0,\ldots,N+1$ ergibt hier dasselbe Gleichungssytem

$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} \ = \ h^2 f_{i,j},\ 1 \ \leq \ i,j \ \leq \ N$$

wie die Standarddiskretisierung mit dem Differenzenverfahren. Aber hier hätten wir ohne Schwierigkeiten auch eine nichtäquidistante Rechtecksgittereinteilung benutzen können.

Kapitel 13

Zugang zu numerischer Software und anderer Information

Don't reenvent the wheel! Für die Standardaufgaben der Numerischen Mathematik gibt es inzwischen public domain Programme sehr guter Qualität, sodass es oft nur notwendig ist, mehrere solcher Module zusammenzufügen, um ein spezifisches Problem zu lösen. Hier wird eine Liste der wichtigsten Informationsquellen angegeben.

13.1 Softwarebibliotheken

In der Regel findet man im Netz bereits vorgefertigte Softwarelösungen, die meisten davon für akademischen Gebrauch kostenfrei: Die bei weitem grösste und wichtigste Quelle ist die

NETLIB

Dies ist eine Sammlung von Programmbibliotheken in f77, f90 , c, c++ für alle numerischen Anwendungen:

http://www.netlib.org/

Man kann nach Stichworten suchen ("search") und bekommt auch Informationen aus dem NaNet (Numerical Analysis Net)

Die Bibliotheken findet man unter "browse repository".

Die wichtigsten Bibliotheken sind:

1. amos, specfunc, cephes: spezielle Funktionen

2. ellpack : elliptische Randwertprobleme
3. fftpack : fast fourier transform
4. fitpack , dierckx: Spline Approximation und Interpolation
5. lapack clapack, lapack90
   : die gesamte numerische lineare Algebra (voll besetzte und band-matrizen) inklusive Eigenwertprobleme und lineare Ausgleichsrechnung in sehr guter Qualität
6. linpack , eispack : die Vorläufer von lapack. Einige der Verfahren aus diesen Bibiliotheken wurden jedoch nicht in lapack übernommen.
7. lanz, lanczos : einige Eigenwerte/Eigenvektoren grosser dünn besetzter symmetrischer Matrizen
8. pdes/cwa : hyperbolische Erhaltungsgleichungen
9. templates : Iterationsverfahren für lineare Gleichungssysteme.
10. toms: Transactions on mathematical software. Sammlung von Algorithmen für verschiedene Aufgaben, sehr gute Qualität u.a. auch automatische Differentiation, Arithmetik beliebiger Genauigkeit, mehrere Optimierungscodes, Nullstellenbestimmung, cubpack (Kubatur), partielle Differentialgleichungen
11. linalg: Iterative Verfahren für lineare Systeme, sonstige lineare Algebra
12. quadpack: Quadratur (bestimmte Integrale, 1-dimensional)
13. ode, odepack: numerische Integration von gewöhnlichen DGLen, auch Randwertaufgaben
14. fishpack: Helmholtzgleichung mit Differenzenverfahren
15. opt,minpack1: Optimierungssoftware (nur ein kleiner Teil, s.u.)
16. slatec: eine eigenständige Bibiliothek mit vielen wichtigen Lösern, u.a. ein Simplexverfahren für grosse dünn besetzte Probleme.

Daneben

http://elib.zib.de/

Dort gibt es auch Bibliotheken, teilweise mit guten Eigenentwicklungen der Gruppe um P. Deuflhard (die Codelib, mit Codes für das gedämpfte Newton- und Gauss-Newtonverfahren, dem System Kaskade zur Lösung elliptischer Gleichungen etc), sowie sonstige weitere Verweise. Die Programme aus Hairer-Norsett-Wanner (Integration gewöhnlicher DGLen I,II) findet man bei

http://www.unige.ch/math/folks/hairer

13.2 Information über Optimierungssoftware

unter

http://plato.la.asu.edu/guide.html

Dort findet man eine vollständige Liste von frei verfügbarer Software für fast alle Bereiche der Optimierung und viele weitere Verweise.

13.3 Suchen nach software

Wenn der Name des Programmmoduls bekannt ist, kann man mit

xarchie

suchen, sonst benutzt man sinnvollerweise zuerst den Dienst

http://math.nist.gov/HotGAMS/

Dort öffnet sich ein Suchmenü, wo man nach Problemklassen geordnet durch einen Entscheidungsbaum geführt wird bis zu einer Liste verfügbarer software (auch in den kommmerziellen Bibliotheken IMSL und NAG). Falls der code als public domain vorliegt, wird er bei "Anklicken" sofort geliefert.

13.4 Andere wichtige Quellen

Wichtig für Ingenieursanwendungen: Die Finite-Element-Resources web-page von Ian MacPhedran:

http://www.engr.usask.ca/∼ macphed/finite/fe_resources/fe_resources.html

Dort gibt es viele links, auch zu freiem FEM-Code, u.a. das Felt - System .

Ebenso:

http://www.dealii.org

mit C++code fuer adaptive finite Element-Berechnungen in 1D, 2D und 3D. Die Lösung grosser, auch unsymmetrischer Eigenwertprobleme leistet ARPACK

http://www.caam.rice.edu/software/ARPACK

Software für C++ findet man unter

http://oonumerics.org/oon/

Software in C oder C++ ist in der Liste aus

ftp://ftp.math.psu.edu/pub/FAQ/numcomp-free-c

zu finden. Software vielfältiger Form für die schnelle Fouriertransformation findet man ausser in der Netlib auch unter

http://theory.lcs.mit.edu/∼ fftw

13.5 Hilfe bei Fragen

Hat man Fragen, z.B. nach Software, Literatur oder auch zu spezifischen mathematischen Fragestellungen, kann man in einer der News-groups eine Anfrage plazieren. Häufig bekommt man sehr schnell qualifizierte Hinweise. Zugang zu Newsgroups z.B. über

xrn

. mit "subscribe" . Die wichtigsten News-Groups sind hier

sci.math.num-analysis
sci.op-research

Es gibt natürlich auch im Bereich Informatik bzw. Software und Ingenieurwissenschaften eine Fülle solcher News-groups.

Im xrn-Menu kann man mit "post" ein Anfrage abschicken und dabei die Zielgruppe frei wählen.

Kapitel 14

Notation, Formeln

$$\|x\| = (x^T x)^{1/2} \qquad \text{euklidische Vektornorm, Länge von } x \qquad (l_2\text{-Norm})$$

$\Big\{$ Andere gebräuchliche Längenmaße:

$$\|x\|_\infty = \max_i |x_i| \qquad \text{Maximumnorm} \qquad (l_\infty\text{-Norm})$$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \text{Betragssummennorm} \qquad (l_1\text{-Norm}) \Big\}$$

1. Ist f eine vektorwertige Funktion von n Veränderlichen x, $f: \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^m$, so bezeichnet

$$\mathcal{J}_f(x) = \begin{pmatrix} \frac{\partial}{\partial x_1} f_1 & , \ldots\ldots, & \frac{\partial}{\partial x_n} f_1 \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_1} f_m & , \ldots\ldots, & \frac{\partial}{\partial x_n} f_m \end{pmatrix}$$

   die **Jacobimatrix** von f (Funktionalmatrix).

   Jacobimatrix: Zeilennummer $\hat{=}$ Funktionsnummer
   Spaltennummer $\hat{=}$ Variablennummer

2. Der **Gradient** ist stets die transponierte Jacobimatrix:

$$\nabla f(x) = (\mathcal{J}_f(x))^T$$

   Gradient: Zeilennummer $\hat{=}$ Variablennummer
   Spaltennummer $\hat{=}$ Funktionsnummer
   Der Gradient einer skalaren Funktion ist also hier ein Spaltenvektor.

3. Für eine skalare Funktion $f: \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ bezeichnet

$$\nabla^2 f(x) = \left(\frac{\partial^2}{\partial x_i \partial x_j} f(x) \right) = \Big((\nabla \nabla^T) f \Big)(x)$$

die **Hessematrix** von f.

Für vektorwertige Funktionen kommt diese Konstruktion nur im Zusammenhang mit der Taylorentwicklung vor, d.h. als Vektor

$$\begin{bmatrix} d^T\nabla^2 f_1(x)d \\ \vdots \\ d^T\nabla^2 f_m(x)d \end{bmatrix} =: \underbrace{d^T\Big(\nabla^2 f(x)\Big)d}_{\text{symbolisch, nur für } m=1 \text{ echte Matrix-Vektor-Notation}}$$

mit einem Inkrementvektor $d \in \mathbb{R}^n$

Intervall im $\mathbb{R}^n$:

$$[x^0, x^0 + d] = \{x^0 + td, \quad 0 \le t \le 1\}$$

Hilfsmittel

Mittelwertsätze

$f \in C^1(\mathcal{D})$

$$\begin{aligned} f(x^0+d) &= f(x^0) + \nabla f(x^0)^T d + o(\|d\|) \quad {}^{*)} \\ f(x^0+d) &= f(x^0) + \nabla f(x^0+\vartheta d)^T d \quad \textbf{falls } f \textbf{ skalar, } 0<\vartheta<1 \\ f(x^0+d) &= f(x^0) + \Big(\underbrace{\int_0^1 \nabla f(x^0+td)^T \,\mathrm{d}t}_{\text{Integral ist komponentenweise zu nehmen}} \Big) d \end{aligned}$$

Taylorentwicklung

$f \in C^2(\mathcal{D}),\ x^0 \in \mathcal{D}$

$$\begin{aligned} f(x^0+d) &= f(x^0) + \nabla f(x^0)^T d + \tfrac{1}{2} d^T \nabla^2 f(x^0) d + o(\|d\|^2) \quad {}^{*)} \\ f(x^0+d) &= f(x^0) + \nabla f(x^0)^T d + \tfrac{1}{2} d^T \nabla^2 f(x^0+\vartheta d) d \quad \textbf{falls } f \textbf{ skalar} \\ f(x^0+d) &= f(x^0) + \nabla f(x^0)^T d + d^T \Big(\int_0^1 (1-t)\nabla^2 f(x^0+td)\,\mathrm{d}t \Big) d \end{aligned}$$

${}^{*)}$ $o(\cdot)$ Landau-Symbol (klein-o)

$o(1)$ bezeichnet eine Größe, die bei einem (in der Regel implizit definierten) Grenzübergang gegen null geht. $o(\|d\|^k)$ bezeichnet eine Größe, die schneller gegen null geht als $\|d\|^k$, d.h.

$$o(\|d\|^k)/\|d\|^k \to 0 \quad \text{für } d \to 0.$$

$\mathcal{O}(h^n)$ bezeichnet eine Grösse mit

$$\mathcal{O}(h^n) \ \le \ Ch^n$$

für einen definierten Grenzübergang von h, hier in der Regel $h \to 0$. $\mathcal{O}(1)$ eine beschränkte Grösse usw.

Taylorformel allgemeiner: Ist f eine $k - mal$ stetig partiell ableitbare Funktion des Vektors x dann gilt

$$\begin{aligned} f(x+h) &= f(x) + \sum_{i=1}^{n} (\frac{\partial}{\partial x_i} f)(x) h_i + \\ & \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\frac{\partial^2}{\partial x_i \partial x_j}) f(x)\ h_i h_j + \\ & \quad \cdots \quad \cdots \\ & \quad \frac{1}{k!} \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \Big((\frac{\partial^k}{\partial x_{i_1} \ldots \partial x_{i_k}}) f(x + \theta_{f,h}) \Big) \prod_{j=1}^{k} h_{i_j} \end{aligned}$$

Ist f ein Vektorfeld, dann muss diese Taylorformel komponentenweise auf die einzelnen Komponentenfunktionen angewendet werden. Den letzten Summanden könnte man mit $\mathcal{O}(||h||^k)$ abkürzend angeben.

Formeln, Rechnen mit $\frac{d}{d\sigma}$, ∇ bei Vektorfunktionen:

$\frac{d}{d\sigma} f(x - \sigma d)_{|\sigma=0} = -(\nabla f(x)^T) d$

$\frac{d^2}{(d\sigma)^2} f(x - \sigma d)_{|\sigma=0} = d^T \nabla^2 f(x) d$

$\nabla(f(x)g(x)) = g(x)\nabla f(x) + f(x)\nabla g(x)$

$\nabla(f(g(x))) = \nabla g(x) \nabla f(y)_{|y=g(x)}$

Insbesondere für: $f(y) = y^T y$ ergeben sich

$\nabla(\|g(x)\|^2) = 2(\nabla g(x)) g(x)$

$\nabla^2(\|g(x)\|^2) = 2(\nabla g(x))(\nabla g(x))^T + 2 \sum_{i=1}^{m} g_i(x) \nabla^2 g_i(x)$ für $g: \ \mathbb{R}^n \to \mathbb{R}^m$

Differentiation einer inversen Matrix nach einem Parameter:

$\frac{d}{d\sigma}(A(\sigma))^{-1} = -(A(\sigma))^{-1} \Big(\frac{d}{d\sigma} A(\sigma) \Big) (A(\sigma))^{-1}$

Index