Numerische Mathematik

für Informatiker

skriptreif im Wintersemester 2002 jeweils mitten in der Nacht
an die Tafeln der Hörsäle 13 und 11 geschrieben von

Prof. Dr. Peter Kunkel

- http://www.mathematik.uni-leipzig.de/numerik/numerik.html
- mailto:kunkel@mathematik.uni-leipzig.de

Dieses Skript wurde erstellt von Xuân Baldauf
<xuan--numerik-skript--sem04@studium.baldauf.org>
mit freundlicher Unterstützung von Matthias Quasthoff für die Quellen zum Schließen der Lücken, die durch Abwesendheit und Windows-Abstürze entstanden waren.

0 Vorbemerkungen

Zeil der Numerischen Mathematik ist die Entwicklung und Bewertung von Verfahren zur (näherungsweisen) zahlenmäßigen Lösung von mathematischen Aufgabenstellungen

Problematik

Problematisch ist

- Die Endlichkeit der Menge der darstellbaren Zahlen (man hat meist nur endlich viele Dezimalstellen für eine Zahl zur Verfügung).
- Die Endlichkeit der Anzahl der Rechenoperationen (es nützt nichts, wenn ein Rechner unendlich lange rechnen muss, um ein Ergebnis zu erhalten).

Beispiel 0.1

Das Polynom

(0.1) $p(x)=x^2-2$

besitzt die positive Nullstelle $\sqrt{2}$. Diese Aussage ist insofern leer, da das Symbol „Wurzel aus zwei“ gerade dadurch „Nullstelle des Polynoms $p(x)=x^2-2$ “ definiert ist. Die Frage nach der Größe von $\sqrt{2}$ könnte so präzisiert werden, dass man eine Dezimalbruchentwicklung von $\sqrt{2}$ angeben soll. Da aber $\sqrt{2}$ nicht rational ist, ist diese Dezimalbruchentwicklung weder abbrechend noch periodisch.

Man muss sich also mit einer rationalen Approximation von $\sqrt{2}$ zufrieden geben. Ein Taschenrechner liefert zum Beispiel

(0.2) $\sqrt{2}\approx 1.41423562$

Die Frage ist nun, wie man (oder der Rechner) überhaupt eine solche Approximation bestimmen kann. Anforderungen an so eine Methoden wären

- Effizienz (das heißt, dass das Resultat „schnell“ verfügbar sein soll) und
- Genauigkeit (das heißt, dass das Resultat eine „gute“ Approximation an den gesuchten Wert darstellt).

Beispiel 0.2

Gegeben sei eine Funktion

(0.3) $$f:[-1,1]\to\mathbb{R} \text{ mit } f(x)=\begin{cases}\dfrac{1-\sqrt{1-x^2}}{x^2} & \text{für } x\neq 0\\ \dfrac{1}{2} & \text{für } x=0\end{cases}$$

Wegen

$$\frac{1-\sqrt{1-x^2}}{x^2}=\frac{\left(1-\sqrt{1-x^2}\right)\cdot\left(1+\sqrt{1-x^2}\right)}{x^2\cdot\left(1+\sqrt{1-x^2}\right)}$$
$$=\frac{1-\left(1-x^2\right)}{x^2\cdot\left(1+\sqrt{1-x^2}\right)} \qquad \text{für } x\neq 0$$
$$=\frac{1}{1+\sqrt{1-x^2}}$$

kann man f auch durch

(0.4) $$f(x)=\frac{1}{1+\sqrt{1-x^2}}$$

darstellen.

Berechnet man $\forall(i=1,2,\ldots):\left(f\left(10^{-i}\right)\right)$ mit den beiden Darstellungen (z.B. auf einem Taschenrechner), so erhält man folgende Resultate:

i	$f(10^{-1})$ nach (0.3)	$f(10^{-1})$ nach (0.4)
0	1.000'000'000'0	1.000'000'000'0
1	0.501'256'289'4	0.501'256'289'3
2	0.500'012'510'0	0.500'012'500'6
3	0.500'001'000'0	0.500'000'125'0
4	0.500'100'000'0	0.500'000'001'3
5	0.510'000'000'0	0.500'000'000'0
6	0.000'000'000'0	0.500'000'000'0

Offensichtlich ist die Berechnung von $f(x)$ für $x\approx 0$ auf der Basis von (0.3) weniger geeignet als auf der Basis von (0.4).

Die Frage ist nun, wie man dieses Phänomen erklären und vermeiden kann.

1 Rechnerarithmetik und Fehleranalyse

1.1 Arithmetik der Maschinenzahlen

Wegen der Endlichkeit des Rechners kann man auf ihm nicht mit ganz $\mathbb{R}$ arbeiten. Stattdessen muss man sich mit einer endlichen Teilmenge davon zufrieden geben. Im Folgenden soll als Basis für weitere Überlegungen ein Modell für eine Rechenrarithmetik entwickelt werden. Tatsächliche Rechner können zum Teil davon abweichen.

Satz 1.1: Zahlendarstellung zu einer Basis

Sei $b\in(\mathbb{N}\setminus\{0,1\})$. Jedes $x\in\mathbb{R}$ besitzt eine Darstellung

(1.1) $$x=\pm\sum_{i=1}^{\infty}\left(d_i\cdot b^{r-i}\right) \text{ mit } \left(\forall(i):\left(d_i\in\{0,1,\ldots,b-1\}\right)\right)\wedge\left(r\in\mathbb{Z}\right)$$

Beweis

Sei $x_0\in\mathbb{R}$ und ohne Einschränkung $x_0\geqslant 0$. Dann $\exists(r\in\mathbb{Z}):\left(0\leqslant x_0<b^r\right)$. Setzt man $d_1=\left\lfloor\frac{x_0}{b^{r-l}}\right\rfloor$ [die linkeste Stelle der Zahl zur Basis b] wobei $\lfloor x\rfloor=max\left(\{l\in\mathbb{Z}\,|\,l\leqslant x\}\right)$ [also wobei $\lfloor x\rfloor$ die Abrundung von x ist], so ist $d_1\in\{0,\ldots,n-1\}$. [Die linkeste Stelle der Zahl ist kleiner als die Basis.]

Aus $d_1\leqslant\frac{x_0}{b^{r-1}}\leqslant d_1+1$ folgt $d_1\cdot b^{r-1}\leqslant x_0\leqslant d_1\cdot b^{r-1}+b^{r-l}$. Daraus folgt für $x_1=x_0-d_1\cdot b^{r-1}$ die Ungleichung $0\leqslant x_1<b^{r-l}$. [Die Zahl abzüglich der linkesten Stelle hat an dieser Stelle eine 0, ist also damit kleiner.]

Nach s Schritten erhält man so die Darstellung

$$x_0=\sum_{i=1}^{s}\left(d_i\cdot b^{r-1}\right)+x_s$$

mit $d_i\in\{0,1,\ldots,b-1\}$ und $0\leqslant x_s<b^{r-s}$.
Die Behauptung folgt dann daraus wegen $b^{r-s}\to 0$ für $s\to\infty$.

q.e.d.

Bemerkung 1.2: normalisierte Darstellung

Alternativ kann ein $x\in\mathbb{R}$ statt gemäß (1.1) auch in der Form

(1.2) $$x=v\cdot m\cdot b^e \text{ mit } x\in\{-1,+1\} \text{ und } m=\sum_{i=1}^{\infty}\left(d_i\cdot b^{r-e-i}\right)$$

(v ist das Vorzeichen, m ist die Mantisse, b ist die Basis, e ist Exponent) dargestellt werden. Fordert man zunächst, dass $e=r$ ist, so gilt $m\in[0,1[$. Fordert man außerdem $(x\neq 0)\Rightarrow(d_1\neq 0)$, so gilt sogar $m\in\left[\frac{1}{b},1\right[$. Im letzteren Fall spricht man von einer **normalisierte**n **Darstellung** von x.

Auf einem Rechner muss entscheidbar sein, wann zwei Zahlen gleich sind. Eine wichtige Frage ist daher, in wie weit normalisierte Darstellungen von reellen Zahlen eindeutig sind.

Lemma 1.3: Eindeutigkeit

Jedes $x \in \mathbb{R}_{-}^{+}$ besitzt höchstens zwei normalisierte Darstellungen der Form (1.1). Besitzt x zwei verschiedene Darstellungen, so ist eine abbrechend, das heißt: der Form

(1.3) $$x = \pm \sum_{i=1}^{s} \left(d_i \cdot b^{r-i} \right) \text{ mit } d_1 \neq 0$$

Die andere Darstellung ist gegeben durch

(1.4) $$x = \pm \sum_{i=1}^{s} \left(d_i \cdot b^{r-1} \right) \mp b^{r-s} \pm \sum_{i=s+1}^{\infty} \left(d_i \cdot (b-1)^{r-i} \right)$$

wobei diese eventuell noch nicht normalisiert ist.

Beweis

Sei ohne Einschränkung $x > 0$ mit den Darstellungen

$$\sum_{i=1}^{\infty} \left(d_i \cdot b^{r-i} \right) = \sum_{i=1}^{\infty} \left(e_i \cdot b^{r-i} \right)$$

wovon ohne Einschränkung die erste normalisiert sei. Weiterhin sei

$$s = \min \left(\left\{ i \in \mathbb{N} \middle| d_i \neq e_i \right\} \right).$$

Ist die zweite Darstellung nicht normalisiert, so gilt $s = 1$ und

$$d_1 > 0 = e_1$$

Sind beide Darstellungen normalisiert, so können wir ohne Einschränkung annehmen, dass $d_s > e_s$.

Damit folgt

$$\begin{aligned} b^{r-s} &\leqslant \left(d_s - e_s \right) \cdot b^{r-s} \\ &= \sum_{i=1}^{s} \left(\left(d_i - e_i \right) \cdot b^{r-i} \right) \\ &= \sum_{i=s+1}^{\infty} \left(\left(e_i - d_i \right) \cdot b^{r-i} \right) \\ &\leqslant \left(b-1 \right) \cdot \sum_{i=s+1}^{\infty} \left(b^{r-i} \right) \\ &= \left(b-1 \right) \cdot b^{r-s-1} \cdot \sum_{i=0}^{\infty} \left(b^{-1} \right) \\ &= \left(b-1 \right) \cdot b^{r-s-1} \cdot \frac{b}{b-1} \\ &= b^{r-s} \end{aligned}$$

Daraus folgt, dass oben überall Gleichheitszeichen gelten, das heißt:

$$d_s = e_s + 1$$

sowie

$$\forall (i = s+1, s+2, \ldots) : \left(e_i - d_i = b-1 \right)$$

beziehungsweise

$$\forall (i = s+1, s+2, \ldots) : \left(\left(e_i = b-1 \right) \wedge \left(d_i = 0 \right) \right)$$

q.e.d.

Die Idee ist nun, mit endlichen (normalisierten) Darstellungen der Form (1.1) zu arbeiten.

Definition 1.4: normalisierte Fließkommazahlen

Die auf einem (idealisierten) Rechnerverfügbaren Zahlen (sogenannte Maschinenzahlen) seien gegeben durch

(1.5) $$M_{b,l}=\left\{\pm\sum_{i=1}^{l}\left(d_i\cdot b^{r-i}\right)\middle|\left(\forall(i=1,\ldots,l):\left(d_i\in\{0,\ldots,b-1\}\right)\right)\wedge\left(d_1\neq 0\right)\wedge\left(r\in\mathbb{Z}\right)\right\}$$

wobei $\left(b\in\mathbb{N}\setminus\{0,1\}\right)\wedge\left(l\in\mathbb{N}\right)$. Man nennt $M_{b,l}$ die Menge der **normalisierte**n **Fließkommazahlen** zur Basis b mit Mantissenlängen l.

Bemerkung 1.5

Bei einem elektronischen Rechner ist $b=2$ oder $b=16$, bei Handrechnung ist $b=10$. Außerdem ist bei einem Rechner der Exponent r zusätzlich beschränkt gemäß

(1.6) $$r_{min}\leqslant r\leqslant r_{max}$$

Die dadurch auftretenden Randeffekte wie Exponentenüberlauf oder Exponentenunterlauf spielen in unseren Untersuchungen keine Rolle und werden deshalb hier nicht berückstichtigt. Wir arbeiten also in unserem Modell mit einer unendlichen Menge $M_{b,l}$ von **Maschinenzahlen**.

Um mit einem gegebenen $x\in\mathbb{R}$ auf dem Rechner arbeiten zu können, muss x durch ein $\tilde{x}\in M_{b,l}$ ersetzt werden. Man spricht von **Rundung**.

Definition 1.6: Rundung

Eine Abbildung $float:\mathbb{R}\to M_{b,l}$ mit

1. $\forall(x\in M_{b,l}):\left(float(x)=x\right)$
2. $\forall\left(\left(x,y\in\mathbb{R}\right)\wedge\left(x\leqslant y\right)\right):\left(float(x)\leqslant float(y)\right)$

heißt **Rundung** (bezüglich $M_{b,l}$).

Für b=10 kennt man die sogenannte kaufmännische Rundung (auch „5/4-Rundung“). Sie kann folgendermaßen verallgemeinert werden:

Lemma 1.7 kaufmännische Rundung

Sei $x\in\mathbb{R}_-^+$ mit normalisierter (im mehrdeutigen Fall abbrechender) Darstellung

(1.7) $$x=\pm\sum_{i=0}^{\infty}\left(d_i\cdot b^{r-i}\right)\ \text{mit}\ d_1\neq 0\,.$$

Durch

(1.8) $$float(x)=\begin{cases}\pm\sum_{i=1}^{l}\left(d_i\cdot r^{r-i}\right) & \text{für } d_{l+1}\in\left\{0,..,\left\lceil\frac{b-1}{2}\right\rceil\right\}\\ \pm\sum_{i=1}^{l}\left(d_i\cdot r^{r-i}\right)\pm b^{r-l} & \text{für } d_{l+1}\in\left\{\left\lceil\frac{b+1}{2}\right\rceil,\ldots,1\right\}\end{cases}$$

sowie $float(0)=0$ ist eine Rundung $float:\mathbb{R}\to M_{b,l}$ gemäß Definition 1.6 gegeben, die außerdem symmetrisch ist, das heißt die Eigenschaft

(1.9) $$\forall(x\in\mathbb{R}):\left(float(-x)=-float(x)\right)$$

besitzt.

Beweis

Zunächst muss gezeigt werden, dass $\forall(x\in\mathbb{R}):\left(float(x)\in M_{b,l}\right)$. Das ist offensichtlich bis auf den zweiten Fall in (1.8). Ist $d_l\leqslant b-2$, so gilt:

$$float(x)=\pm\sum_{i=1}^{l-1}(d_i\cdot b^{r-i})\pm(d_l+1)\cdot b^{r-l}\in M_{b,l}$$

Ist aber $d_l=b-1$, so hat man zunächst nur

$$float(x)=\pm\sum_{i=1}^{l-1}(d_i\cdot b^{r-i})\pm b^{r-l+1}$$ (Wobei b^{r-l+1} ein Übertrag ist)

Das ist aber genau der zweite in (1.8) mit um eins verkürzter Mantisse. Induktiv ergibt sich also entweder $float(x)\in M_{b,l}$ durch den ersten Schritt oder schließlich $float(x)=\pm b^r\in M_{b,l}$ im zweiten Schritt.

Die Projektionseigenschaft und die Symmetrie ergeben sich direkt aus der Definition. Die Monotonie ergibt sich aus der Beobachtung, dass im Inneren des Intervalls

$$I=\left[\sum_{i=1}^{l}(d_i\cdot b^{r-i}),\sum_{i=1}^{l}(d_i\cdot b^{r-i})+b^{r-l}\right]$$

keine Maschinenzahl liegt und alle x mit

$$x\in\left[\sum_{i=1}^{l}(d_i\cdot b^{r-i}),\sum_{i=1}^{l}(d_i\cdot b^{r-i})+\left[\frac{b+1}{2}\right]\cdot b^{r-l-1}\right[$$

auf die linke Intervallgrenze beziehungsweise alle x mit

$$x\in\left[\sum_{i=1}^{l}(d_i\cdot b^{r-i})+\left[\frac{b+1}{2}\right]\cdot b^{r-l-1},\sum_{i=1}^{l}(d_i\cdot b^{r-i})+b^{r-l}\right[$$

auf die rechte Intervallgrenze gerundet werden.

Bemerkung 1.8

Ist b gerade, so wird bei obiger Rundung zur nächstliegenden Maschinenzahl gerundet. Ist diese eindeutig, so wird zur betragsgrößeren Maschinenzahl gerundet.

Offensichtlich wird bei jeder Rundung (falls nicht zufällig eine Maschinenzahl vorliegt) ein Fehler gemacht, ein so genannter **Rundungsfehler**. Es ist deshalb wichtig zu wissen, wie dieser Fehler aussieht beziehungsweise wie er abgeschätzt werden kann.

Lemma 1.9

Ist b gerade, so gilt für die durch (1.8) definierte Rundung die Abschätzung

(1.10) $$|float(x)-x|\leqslant\frac{1}{2}\cdot b^{-l+1}\cdot|x|$$

Man kann obige Formel durch $x\neq 0$ dividieren, was zeigt, dass Rundungsfehler relative Fehler sind.

Beweis

Sei ohne Einschränkung $x>0$ und damit $float(x)>0$. Unter Verwendung von (1.7) und (1.8) erhält man folgende Abschätzungen:

1. Fall: $d_{l+1} \leqslant \left\lceil \frac{b-1}{2} \right\rceil = \frac{b}{2} - 1$. Dann gilt:

$$\begin{aligned} 0 &\leqslant x - float(x) \\ &= \sum_{i=1}^{\infty} (d_i \cdot b^{r-i}) - \sum_{i=1}^{l} (d_i \cdot b^{r-i}) \\ &= \sum_{i=1+1}^{\infty} (d_i \cdot b^{r-i}) \\ &\leqslant \left(\frac{b}{2} - 1\right) \cdot b^{r-l-1} + (b-1) \cdot \sum_{i=l+2}^{\infty} (b^{r-i}) \\ &= \left(\frac{b}{2} - 1\right) \cdot b^{r-l-1} + (b-1) \cdot b^{r-l-2} \cdot \frac{b}{b-1} \\ &= \frac{1}{2} \cdot b^{r-l} \end{aligned}$$

2. Fall: $d_{l+1} \geqslant \left\lceil \frac{b+1}{2} \right\rceil = \frac{b}{2}$. Dann gilt:

$$\begin{aligned} 0 &\geqslant x - float(x) \\ &= \sum_{i=1}^{\infty} (d_i \cdot b^{r-i}) - \sum_{i=1}^{l} (d_i \cdot b^{r-i}) - b^{r-l} \\ &= \sum_{i=l+1}^{\infty} (d_i \cdot b^{r-1}) - b \cdot b^{r-l-1} \\ &= (d_{l+1} - b) \cdot b^{r-l-1} + \sum_{i=l+2}^{\infty} (d_i \cdot b^{r-1}) \\ &\geqslant \left(\frac{b}{2} - b\right) \cdot b^{r-l-1} \\ &= -\frac{1}{2} \cdot b^{r-l} \end{aligned}$$

In beiden Fällen erhält man also

$$|float(x) - x| \leqslant \frac{1}{2} \cdot b^{r-l} = \frac{1}{2} \cdot b^{r-1-l+1} = \frac{1}{2} \cdot b^{r-1} \cdot b^{-l+1}$$

Die Behauptung folgt daraus wegen $|x| \geqslant b^{r-1}$.

q.e.d.

Definition 1.10: Maschinengenauigkeit

Sei b gerade. Die Größe

(1.11) $$eps = \frac{1}{2} \cdot b^{-l+1}$$

heißt (die zu obiger Rundung gehörige) **Maschinengenauigkeit**. (Sie ist der maximale relative Fehler, der bei einer Rundung überhaupt vorkommen kann.)

Bemerkung 1.11

Die Abschätzung

(1.12) $$\forall \left(x \in \mathbb{R}\right) : \left(|float(x) - x| \leqslant eps \cdot |x|\right)$$

impliziert für ein gegebenes $x \in \mathbb{R}$ mit zugehörigem $\tilde{x} = float(x)$ die Existenz von $\epsilon \in \mathbb{R}$ mit

(1.13) $\tilde{x}=x\cdot(1+\epsilon)$. Es gilt damit: $|\epsilon|\leq eps$
[ϵ ist hier der konkrete relative Fehler der Rundung]

Beweis

Für $x=0$ ist die Behauptung trivial, etwa mit der Wahl $\epsilon=0$.
Für $x\neq 0$ setzt man

$$\epsilon=\frac{\tilde{x}-x}{x}\ .$$

Damit gilt sofort $\tilde{x}=x\cdot(1+\epsilon)$. Außerdem folgt mit (1.12):

$$|\epsilon|=\frac{|\tilde{x}-x|}{|x|}=\frac{|float(x)-x|}{|x|}\leq eps\ .$$

„Wir betrachten nun folgende Problematik: Maschinenzahlen sind bezüglich der Grundrechenarten nicht abgeschlossen. Das heißt: Es gibt Maschinenzahlen, deren Summen keine Maschinenzahl ist."

Nachdem wir ein Modell für die auf einem Rechner verwendbaren Zahlen zur Verfügung haben, wollen wir natürlich mit diesen Zahlen rechnen. Die Problematik dabei ist, dass man schon bei den Grundrechenarten nicht erwarten kann, dass die Verknüpfung zweier Maschinenzahlen wieder eine Maschinenzahl ist.

Als Modell für die Grundrechenarten werwenden wir

(1.14)
$$\begin{array}{ll}(a) & x\oplus y=float(x+y)\\ (b) & x\ominus y=float(x-y)\\ (c) & x\odot y=float(x\cdot y)\\ (d) & x\oslash y=float(x/y)\end{array}$$

entsprechend für die sogenannten Standardfunktionen

(1.15) $(funktion)(x)=float(funktion(x))$

etwa mit $funktion\in\{\sqrt{\ },\sin,\cos,\tan,\arctan,\exp,\log,\ldots\}$.
Zusammen mit (1.4) und (1.8) haben wir damit ein Modell eines Rechners (idealer Rechner) aufgestellt.

Bemerkung 1.12

Man beachte, dass sich nicht alle Eigenschaften der Verknüpfungen von $\mathbb{R}$ auf $M_{b,l}$ übertragen. So ist die Rechneraddition $\oplus$ zwar kommutativ, aber nicht assoziativ.

Beispiel 1.13

Sei $b=10$ und $l=2$. Gegeben seien:

$x=104$, $y=4.22$, $z=3.86$

Diese werden gerundet zu:

$\tilde{x}=0.10\cdot 10^{3}$, $\tilde{y}=0.42\cdot 10^{1}$, $\tilde{z}=0.39\cdot 10^{1}$

Damit erhält man statt $x+y+z=112.08$ folgendes:

$$
\begin{aligned}
(\tilde{x} \oplus \tilde{y}) \oplus \tilde{z} &= float(1.042 \cdot 10^3) \oplus 0.39 \cdot 10^1 \\
&= 0.10 \cdot 10^3 \oplus 0.39 \cdot 10^1 \\
&= float(0.1039 \cdot 10^3) \\
&= 0.10 \cdot 10^3
\end{aligned}
$$

beziehungsweise

$$
\begin{aligned}
\tilde{x} \oplus (\tilde{y} \oplus \tilde{z}) &= 0.10 \cdot 10^3 \oplus float(0.81 \cdot 10^1) \\
&= 0.10 \cdot 10^3 \oplus 0.81 \cdot 10^1 \\
&= float(0.1081 \cdot 10^3) \\
&= float(112.08) \\
&= 0.11 \cdot 10^3
\end{aligned}
$$

(Dies ist gerade das gerundete Ergebnis der genauen Rechnung).

Die Assoziativität (Beliebigkeit der Reihenfolge der Ausführung) geht durch die Rundung in den Zwischenschritten verloren.

„Wir wissen, dass die Maschinengenauigkeit die Form $\frac{1}{2} \cdot b^{l-1}$ hat. Dies ist auf einem Rechner die kleinste Zahl, die, wenn zu eins addiert, eine Zahl größer eins ergibt."

Algorithmus 1.14

Auf einem idealen Rechner (mit geradem b) gilt:

(1.16) $$eps = min\left(\left\{x \in M_{b,l} \middle| (x > 0) \wedge (1 \oplus x > 1)\right\}\right)$$

Für $b = 2$ gilt: $eps = 2^{-l}$. Man kann eps deshalb mit dem folgenden Algorithmus bestimmen:

(1.17)
```
eps = 1.0;
while (1.0+eps>1.0) {
  eps = 0.5*eps;
}
eps = 2.0*eps;
```

Bemerkung 1.15: IEEE-Standard 754

Das vorstehend entwickelte Modell eines idealen Rechners, insbesonder (1.14) und (1.15) zusammen mit (1.13), bildet die Grundlage für die Entwicklung und Beurteilung von numerischen Verfahren.
Tatsächliche Rechner weichen davon nicht nur bezüglich der Endlichkeit der Menge der darstellbaren Zahlen in unterschiedlichem Maß davon ab. Dies betrifft vor allem Großrechner (etwa von IBM oder Cray) zum Beispiel bezüglich Rundung beziehungsweise Genauigkeit der Grundrechenarten. Im Jahr 1985 wude vom Institute of Electrical end Electronics Engineers ein Standard herausgegeben, dem zumindest alle gängigen Workstations und (mit geringen Abweichungen) PCs genügen. Dabei werden zwei Zahlenformate fest vorgeschrieben und Mindestanforderungen für zwei zugehörige erweiterter Formate festgelegt. Diese Forderungen werden auf den letztgenannten Rechnerarchitekturen durch das Vorhandensein von drei Zahlenformaten erfüllt. (Vergleiche: Tabelle 1.1)

Format	*single*	*double*	*extended (Workstation)*	*extended (PC)*
Größe in Byte	4	8	16	10
Basis b	2	2	2	2
Länge l der Mantisse	24	53	113	64
hidden bit	ja	ja	ja	nein
Bits für Exponenten	8	11	15	15

Table 1.1: Zahlenformate

Die Mantisse bei Binärzahlen!=0 fängt immer mit 1 an. Aus diesem Grund wird diese führende '1' im „hidden bit“ als implizit vorausgesetzt. Damit wird ein Bit frei, was für das Vorzeichen verwendet wird.

Der Standard fordert für die Grundformate single und double die Eigenschaften (1.14) sowie (1.15) für die Wurzelfunktion.

In Abweichung von (1.8) wird für ein x, was genau zwischen zwei Maschinenzahlen liegt, auf diejenige von beiden gerundet, deren letzte Ziffer in der Mantisse null ist. Daneben schreibt der Standard drei weitere wählbare Rundungsformen:
- Runden in Richtung 0 (Abschneiden)
- Runden in Richtung $+\infty$
- Runden in Richtung $-\infty$

Im ersten Fall gilt (1.12) wie im Modell mit $eps=2^{-l}$, wenn man von $M_{2,l}$ ausgeht.
Bei den drei anderen Rundungen gilt (1.12) mit $eps=2^{-l+1}$.

„Was für Konsequenzen hat diese Art der Zahlendarstellung für uns, wenn wir tatsächlich wissenschaftliche oder technische Probleme lösen wollen? Rundungsfehler können den Nutzen einer Computer-Berechnung vereiteln.“

1.2 Grundlegende Begriffe

Ein zu lösendes Problem ist dadurch gekennzeichnet, dass zu gegebenen Daten x ein Ergebnis y zu berechnen ist, wobei der Zusammenhang zwischen x und y entweder explizit durch
(1.18) $y=f(x),\ f:X\to Y$
oder implizit durch
(1.19) $g(x,y)=0,\ g:X\times Y\to Z$
beschrieben wird. Die Mengen X,Y,Z sind im folgenden typischerweise Teilmengen irgendwelcher $\mathbb{R}^n$ (Mengen von n-Tupeln, deren Elemente reelle Zahlen sind).

„Forderung: Das Ergebnis muss eindeutig sein.“
„Wir müssen berücksichtigten, dass mit dem Rechner Rundungsfehler auftreten.“

Um ein Ergebnis y zahlenmäßig angeben zu können, muss dieses eindeutig durch die Problemstellung festgelegt sein. („Der implizite Fall muss umformbar sein in explizite Fälle.“)

Da wir außerdem eventuell schon bei der Eingabe der Daten x auf dem Rechner durch Rundung Fehler machen, muss sich das gestellte Problem unter Störung gutartig verhalten. („Das y soll bei leicht abgeändertem x „nicht wackeln und nicht springen“.“)

Definition 1.16: Wohlgestelltheit eines Problems

Ein Problem (1.18) oder (1.19) heißt **wohlgestellt**, wenn es zu jedem x genau eine Lösung $y=f(x)$ mit $y\in Y$ gibt und die Zuordnung $x \overset{f}{\mapsto} y$ stetig ist.

Definition: Norm

Für ein wohlgestelltes Problem möchte man ein Maß dafür, wie stark sich Fehler in den Daten auf Fehler in der Lösung auswirken. Um Fehler messen zu können, benötigen wir Normen. Im $\mathbb{R}^n$ verwendet als Verallgemeinerung des Betrags in $\mathbb{R}$ etwa

(1.20)
$$\text{(a) } \|x\|_1=\sum_{i=1}^{n}|x_i| \quad \text{(Summennorm)}$$
$$\text{(b) } \|x\|_2=\left(\sum_{i=1}^{n}|x_i|^2\right)^{\frac{1}{2}} \quad \text{(euklidische Norm)}$$
$$\text{(c) } \|x\|_\infty=\max_{i=i,\ldots,n}(|x_i|) \quad \text{(Maximumnorm)}$$
$$x=(x_1,\ldots,x_n)^T$$

Eine Norm $\|\cdot\|$ erfüllt die Eigenschaften:

$\|x\|>0\ \forall\ x\neq 0$

$\|\alpha\cdot x\|=|\alpha|\cdot\|x\|\ \forall\ x\in\mathbb{R}^n\wedge\alpha\in\mathbb{R}$

$\|x+y\|\leq\|x\|+\|y\|\ \forall\ x,y\in\mathbb{R}^n$

Definition 1.17: Absoluter|relativer Fehler

Sei $\tilde{x}\in\mathbb{R}^n$ eine Näherung an $x\in\mathbb{R}^n$. Sei außerdem eine Norm $\|\square\|$ auf $\mathbb{R}^n$ gegeben. Dann heißt $\|\tilde{x}-x\|$ der zugehörige absolute Fehler von $\tilde{x}$.

Ist $x\neq 0$, so heißt $\frac{\|\tilde{x}-x\|}{\|x\|}$ der zugehörige relative Fehler von $\tilde{x}$.

Man beachte, dass Rundungsfehler wegen (1.10) bzw. $\frac{|float(x)-x|}{|x|}\leq eps$ für $x\neq 0$ relative Fehler sind.

Datum: 06.11.2002

Wiederholung:

x: Daten

y: Ergebnis

(1.18) $y=f(x),\ f:X\to X$

(1.19) $g(x,y)=0, g:X\times Y\to Z$

Definition 1.16 (Wohlgestelltheit)

$$(\text{genau eine Lösung } y) \wedge (x \overset{f}{\mapsto} y \text{ ist stetig})$$

Definition 1.18: Kondition eines Problems

Sei durch $f: X \to Y$ ein wohlgestelltes Problem beschrieben. Dann heißt.

(1.21) $$\kappa = \sup_{\substack{x_1, x_2 \in X \\ x_1 \neq x_2}} \left(\frac{\|f(x_2) - f(x_1)\|_Y}{\|x_2 - x_1\|_X} \right), \text{ wobei } \kappa \in [0, \infty]$$

Kondition des Problems.

Man beachte, dass κ von der Wahl der Normen auf X und Y abhängt.

Bemerkung 1.19

Sind $X, Y \subseteq \mathbb{R}$ mit $\|\square\|_X = \|\square\|_Y = |\square|$ (Ist das gleichbedeutend mit: $\forall (X, Y \subseteq \mathbb{R}): \forall (x \in X): \forall (y \in Y): ((x = y) \Leftrightarrow (\|x\| = \|y\| = |x| = |y|))$?) und ist f stetig differenzierbar, so gilt nach dem Mittelwertsatz der Differentialrechnung:

(1.22) $$\forall (x \in X): (\kappa = \sup(|f'(x)|))$$

[Das bedeutet also, dass κ der größtmöglichste Anstieg der Funktion im untersuchten Definitionsbereich ist.]

Bemerkung 1.20: gut|schlecht konditioniertes Problem

Um die Kondition eines Problems zu beurteilen, seien x die wahren Daten, $\tilde{x}$ die verfälschten Daten, jeweils mit Resultaten $y = f(x)$ und $\tilde{y} = f(\tilde{x})$. Ist $\epsilon = \|\tilde{x} - x\|_X$ und benötigt man das Resultat mit einer Genauigkeit $\tau > 0$, so kann man wegen (1.21) nur im Fall

(1.23) $$\tau \geqslant \kappa \cdot \epsilon$$

mit der Einhaltung der geforderten Genauigkeit rechnen. Man sagt in diesem Fall, das Problem sei **gut konditioniert**. Andernfalls sagt man, das Problem sei **schlecht konditioniert**. Will man im Extremfall bei Eingabefehlern $\epsilon = eps$ noch die Größenordnung des Resultats erfassen ($\tau = 1$), muss gelten:

(1.24) $$\kappa \leqslant \frac{1}{eps}$$

Beispiel 1.21

Das durch (0.3) mit $X = [-1,1]$ und $Y = \mathbb{R}$ gegebene Problem ($f(x) = \begin{cases} \frac{1-\sqrt{1-x^2}}{x^2} & \text{für } x \neq 0 \\ \frac{1}{2} & x = 0 \end{cases}$)

ist wohlgestellt.

Für $\epsilon > 0$ hinreichend klein gilt:

$$\frac{1}{\epsilon}\left(f(1)-f(1-\epsilon)\right)=\frac{1}{\epsilon}\left(1-\frac{1-\sqrt{1-(1-\epsilon)^2}}{(1-\epsilon)^2}\right)$$
$$=\frac{1-2\cdot\epsilon+\epsilon^2-1+\sqrt{2\cdot\epsilon-\epsilon^2}}{\epsilon\cdot(1-\epsilon)^2}$$
$$=\frac{-2+\epsilon^2+\sqrt{\frac{2}{\epsilon}-1}}{(1-\epsilon)^2}$$

das heißt:

$$\frac{1}{\epsilon}\left(f(1)-f(1-\epsilon)\right)\to\infty \text{ für } \epsilon\to 0$$

Damit ist $\kappa=\infty$ und das Problem ist schlecht konditioniert.

Wählt man stattdessen $X=\left[-\frac{1}{2},\frac{1}{2}\right]$, so hat $f: X\to Y$ wegen

$\sqrt{1-x^2}=1-\frac{1}{2}\cdot x^2-\sum_{\kappa\geqslant 2}\left(\frac{(2\cdot\kappa-3)!!}{\kappa!\cdot 2^\kappa}\cdot x^{2\cdot\kappa}\right)$ mit Konvergenzradius $R=1$,

(Dabei ist $x!!$ die Doppelfakultät $1\cdot 3\cdot 5\cdot 7\cdot 9\cdot\ldots\cdot x$ oder $2\cdot 4\cdot 6\cdot 8\cdot 10\cdot\ldots\cdot x$)

das heißt: f ist differenzierbar auf X mit der Ableitung:

$$f(x)=\frac{1}{2}+\sum_{\kappa\geqslant 2}\left(\frac{(2\cdot\kappa-3)!!}{\kappa!\cdot 2^\kappa}\cdot x^{2\cdot\kappa-2}\right)$$
$$f'(x)=\sum_{\kappa\geqslant 2}\left(\frac{(2\cdot\kappa-2)\cdot(2\cdot\kappa-3)!!}{\kappa!\cdot 2^\kappa}\cdot x^{2\cdot\kappa-3}\right)$$

Also ist f' monoton und punktsymmetrisch und es gilt mit (1.22):

$$\kappa=f'\left(\frac{1}{2}\right)=\sum_{\kappa\geqslant 2}\left(\left(\frac{2\cdot\kappa-2}{\kappa\cdot 2}\cdot\frac{2\cdot\kappa-3}{(\kappa-1)\cdot 2}\cdot\ldots\cdot\frac{1}{1\cdot 2}\right)\cdot\left(\frac{1}{2}\right)^{2\cdot\kappa-3}\right)\leqslant\sum_{\kappa\geqslant 2}\left(\left(\frac{1}{2}\right)^{2\cdot\kappa-3}\right)$$

Das heißt, dass das so modifizierte Problem (mit eingeschränktem Definitionsbereich) jetzt gut konditioniert ist.

Die obige Untersuchung besagt, dass die Funktionswerte $f(x)$ von f aus Beispiel (0.2) in der Nähe der Null nicht sensitiv gegenüber Fehlern in x sind, was die Auswertung mit Hilfe von (0.4) bestätigt. Die Probleme bei der Verwendung der Darstellung (0.3) kommen also nicht von einer schlechten Kondition des Problems, sondern müssen vom gewählten Rechenweg herrühren.

(„Wohlgestelltheit und Kondition hängen vom tatsächlichen Problem (der Funktion selbst) ab, während konkrete Rechnerimplementationen zusätzliche Fehler einbringen können.“)

Im Normallfall kann man nicht erwarten, dass man die Zuordnung $x\overset{f}{\mapsto}y$ auf dem Rechner in einem Schritt ohne Erzeugen von Zwischenergebnissen durchführen kann. [Wenn wir also mit Zwischenergebnissen arbeiten, so können wir diese Tatsache formalisieren:]

Definition 1.22: Algorithmus

Ein Algorithmus ist eine endliche Folge von Rechenvorschriften bestehend aus Elementaroperationen (Grundrechenarten und Auswertungen von Standardfunktionen), die angibt, wie y aus x zu bestimmen ist, das heißt eine Zerlegung einer Funktion $f: X \to Y$ entsprechend

(1.25) $f = \varphi_l \circ \varphi_{l-1} \circ \ldots \circ \varphi_1$

mit

(1.26) $\forall (i = 1, \ldots, l): \left(\phi_i : X_i \to X_{i+1}\right)$

wobei $X_1 = X$ und $X_{l+1} = Y$ und jedes durch eine Elementaroperation erzeugte Zwischenergebnis als Komponente des Bildes eines φ_i auftritt.

Man beachte, dass es zu f verschiedene Algorithmen geben kann, wie etwa im Beispiel (0.2) basierend auf den Darstellungen (0.3) und (0.4).
Da man mit jedem φ_i Zwischenresultate und damit Rundungsfehler erzeugt, muss man auch die Auswirkungen dieser Fehler auf das Resultat berücksichtigen.

(„Man versteht unter einem guten Algorithmen einen, der nicht mehr kaputt macht als die Kondition kaputt macht. Ein schlechter Algorithmus wäre einer, der durch die Rundungsfehler zwischen den Ergebnissen das Endergebnis ganz kaputt macht.“)

Diese Auswirkungen sollten nicht wesentlich größer sein als die der unvermeidbaren Fehler. Zu diesen gehören:

- Eingabefehler
- Fehler in den Daten
- Rundung der Daten
- Rundung des Resultats

Definition 1.23: Stabilität eines Algorithmus

Zu einem Algorithmus (1.25) seien die sogenannten Restabbildungen ψ_i definiert durch

(1.27) $\forall (i = 1, \ldots, l): \left(\psi_i = \phi_l \circ \phi_{l-1} \circ \ldots \circ \phi_i \text{ mit } \psi_i : X_i \to Y\right)$

mit den jeweiligen Konditionen κ_i. Gilt

(1.28) $\kappa_i \leqslant (c+1) \cdot \max\left(\kappa, 1\right)$

wobei c die Anzahl der im Algorithmus durchzuführenden Elementaroperation ist, so spricht man von einem **stabilen Algorithmus**.

(„Um einen Algorithmus zu beurteilen, so schau ich mir die Kondition der Restabbildungen an. “)

Davon unabhängig kann die Zahl

$$\frac{\max\limits_{i=1,\ldots,l}\left(\kappa_i\right)}{\max\left(\kappa, 1\right)}$$

als Stabilitätsmaß eines Algorithmus angesehen werden. („Je kleiner, desto besser.“)

1.3 Differentielle Fehleranalyse

Wie man an Beispiel 1.21 schon gesehen hat, sind Berechnungen basierend auf (1.21) nicht leicht durchzuführen. Geht man davon aus, dass Fehler klein sind, so bieten sich Linearisierungstechniken an, das heißt: man vernachlässigt quadratische Terme in den Fehlern. Man spricht von **differentieller Fehleranalyse**.

„Fehler, die man auf dem Rechner gemacht, sind normalerweise Rundungsfehler. Rundungsfehler sind normalerweise ganz klein. Wenn ich jetzt die Auswirkung von Fehlern untersuche, dann kann ich bei kleinen Fehlern quadratische Terme vernachlässigen, weil, wenn Fehler klein sind, Quadrate von Fehlern noch kleiner sind."

Sei $f: X \to Y$ mit $X, Y \subseteq \mathbb{R}$ differnzierbar, so gilt für $x \in \mathbb{R}$ und gestörtes $\tilde{x}$ mit dazugehörigen Resultaten $y = f(x)$ und $\tilde{y} = f(\tilde{x})$:

$$\begin{aligned} \tilde{y} - y &= f(\tilde{x}) - f(x) \\ &= f(x + (\tilde{x} - x)) - f(x) \\ &\doteq f'(x) \cdot (\tilde{x} - x) \qquad \text{(Taylor-Entwicklung)} \end{aligned} \tag{1.29}$$

Das spezielle Gleichheitszeichen mit dem Punkt drüber $\doteq$ bedeutet, dass quadratische Anteile vernachlässigt werden.

Mit $\|\square\|_X = \|\square\|_Y = |\square|$ erhält man speziell:

$$\kappa = |f'(x)| \qquad \text{(\textbf{differentielle Kondition} bezüglich des absoluten Fehlers)} \tag{1.30}$$

Wählt man stattdessen $\|\square\|_X = \frac{|\square|}{|x|}$ und $\|\square\|_Y = \frac{|\square|}{|y|}$, falls $(x \neq 0) \wedge (y \neq 0)$, so liefert (1.29):

$$\frac{\tilde{y} - y}{y} \doteq \frac{x \cdot f'(x)}{y} \cdot \frac{\tilde{x} - x}{x} \tag{1.31}$$

bzw.

$$\kappa = \left| \frac{x \cdot f'(x)}{f(x)} \right| \qquad \text{(\textbf{differentielle Konditionen} bezüglich des relativen Fehlers)} \tag{1.32}$$

Man beachte, dass Rundungsfehler gemäß (1.10) relative Fehler sind.

Ist $X \subseteq \mathbb{R}^n$, so hat die Jacobimatrix $f'(x)$ die Form

$$f'(x) = \left[\frac{\partial f}{\partial x_1}(x) \cdot \ldots \cdot \frac{\partial f}{\partial x_n}(x) \right] \tag{1.33}$$

[oder etwa $f'(x) = \left[\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right]$?] mit $x = (x_1, \ldots, x_n)^T$

und (1.29) wird mit $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)^T$ zu

$$\tilde{y} - y \doteq \sum_{j=1}^{n} \left(\frac{\partial f}{\partial x_j}(x) \cdot (\tilde{x}_j - x_j) \right) \tag{1.34}$$

Damit gilt in erster Näherung

$$|\tilde{y} - y| \leq \sum_{j=1}^{n} \left(\left| \frac{\partial f}{\partial x_j}(x) \right| \cdot |\tilde{x}_j - x_j| \right) \tag{1.35}$$

bzw.

(1.36) $$\left|\frac{\tilde{y}-y}{y}\right| \leq \sum_{j=1}^{n} \left(\left| \frac{x_j}{y} \cdot \frac{\partial f}{\partial x_j}(x) \right| \cdot \left| \frac{\tilde{x}_j - x_j}{x_j} \right| \right)$$

und die entsprechenden Faktoren beschreiben die differntielle Kondition bezüglich des absoluten bzw. relativen Fehlers der einzelnen Komponenten von x.
Für $Y \subseteq \mathbb{R}^m$ braucht man nur entsprechend die einzelnen Komponenten von $y \in Y$ zu betrachten.
Entsprechend kann man die differentielle Fehleranalyse zur Untersuchung der Stabilität eines Algorithmus verwenden, Man muss nur nach Definition (1.23) statt f die Restabbildungen Ψ_i betrachten.

Sei dazu $X_i \subseteq \mathbb{R}^{n_i}$, $x_1 = x$ sowie $x_{i+1} = \phi_i(x_i)$, $i = 1, \ldots, l$, außerdem $x_i = ((x_i)_1, (x_i)_2, \ldots (x_i)_{n_i})^T$.
Damit sind die maßgeblichen differentiellen Konditionen bezüglich des relativen Fehlers entsprechend (1.35) gegeben durch

(1.37) $$\forall (i = 1, \ldots, l) : \forall (j = i, \ldots, n_i) : \left(\kappa_{ij} = \left| \frac{x_{ij}}{y} \cdot \frac{\partial \psi_i}{\partial x_{ij}}(x_i) \right| \right)$$

Beispiel 1.24 Konditionen von Addition|Subtraktion

Sei $y = x_1 + x_2$, das heißt $f : \mathbb{R}^2 \to \mathbb{R}$ mit $f(x_1, x_2) = x_1 + x_2$, so gilt:
$\tilde{y} - y = (\tilde{x}_1 + \tilde{x}_2) - (x_1 + x_2) = (\tilde{x}_1 - x_1) + (\tilde{x}_2 - x_2)$
und damit gilt für den absoluten Fehler:
$|\tilde{y} - y| \leq |\tilde{x}_1 - x_1| + |\tilde{x}_2 - x_2| = \|\tilde{x} - x\|_1$
bzw. für den relativen Fehler: (bei $y, x_1, x_2 \neq 0$)
$$\frac{|\tilde{y} - y|}{|y|} \leq \frac{|x_1|}{|x_1 + x_2|} \cdot \frac{|\tilde{x}_1 - x_1|}{|x_1|} + \frac{|x_2|}{|x_1 + x_2|} \cdot \frac{|\tilde{x}_2 + x_2|}{|x_2|} \leq max \left\{ \frac{|x_1|}{|x_1 + x_2|}, \frac{|x_2|}{|x_1 + x_2|} \right\} \left(\frac{|\tilde{x}_1 - x_1|}{|x_1|} + \frac{|\tilde{x}_2 + x_2|}{|x_2|} \right)$$

Für $\|\cdot\|_Y = \|\cdot\|_1$ und $\|\cdot\|_Y = |\cdot|$ (absoluter Fehler) hat man wegen $\kappa = 1$ ein gut konditioniertes Problem.

Wählt man $\|\cdot\|_X$ stattdessen $\|(z_1, z_2)^T\|_X = \left|\frac{z_1}{x_1}\right| + \left|\frac{z_2}{x_2}\right|$ (relativer Fehler), so liegt nur dann ein gut konditioniertes Problem vor, wenn $|x_1 + x_2|$ nicht wesentlich kleiner ist als $\max\{|x_1|, |x_2|\}$. (Also wenn es einen großen betragsmäßigen Unterschied zwischen x_1 und x_2 gibt oder sie das selbe Vorzeichen haben. Sind sie etwa gleichen Betrags und haben sie unterschiedliche Vorzeichen, so liegt die Summe innerhalb des Intervalls $x_1..x_2$. Da der Fehler proportional zum Intervall ist, die Größenordnung von x_1+x_2 aber sinkt, steigt damit der relative Fehler.)

Andernfalls hat man ein schlecht konditioniertes Problem (sogenannte **subtraktive Auslöschung**). Das gleiche folgt auch mit differentieller Fehleranalyse.

Beispiel 1.25: Kondition der Multiplikation

Sei $y = x_1 \cdot x_2$, das heißt $f : \mathbb{R}^2 \to \mathbb{R}$ mit $f(x_1, x_2) = x_1 \cdot x_2$. Für die differentielle Kondition bezüglich des relativen Fehlers (bei $x_1, x_2 \neq 0$) erhält man nach (1.36)

$$\left|\frac{x_1}{y}\cdot\frac{\partial f}{\partial x_1}(x)\right| = \left|\frac{x_1}{x_1\cdot x_2}\cdot x_2\right| = 1,$$

(wegen Kommutativität der Multiplikation gilt für x_1 das selbe wie für x_2)

$$\left|\frac{x_2}{y}\cdot\frac{\partial f}{\partial x_2}(x)\right| = \left|\frac{x_2}{x_1\cdot x_2}\cdot x_1\right| = 1$$

Die Multiplikation ist also bezüglich des relativen Fehlers (im differentiellen Sinn) gut konditioniert.

Unter Verwendung von Bemerkung 1.12 kann man auch folgendermaßen schließen.

Sei $|\tilde{x}_i - x_1| \leq \mathrm{eps}(|x_i|),\ i=1,2$

das heißt: gebe es ϵ_i mit $|\epsilon_i| \leq \mathrm{eps}$, sodass

$$\tilde{x}_i = x_i\cdot(1+\epsilon_i),\ i=1,2$$

Dann gilt:

$$\tilde{y} = \tilde{x}_1\cdot\tilde{x}_2 = x_1\cdot(1+\epsilon_1)\cdot x_2\cdot(1+\epsilon_2) \doteq x_1\cdot x_2\cdot(1+\epsilon_1+\epsilon_2) = y\cdot(1+\epsilon_1+\epsilon_2)$$

und damit

$$\frac{|\tilde{y}-y|}{|y|} \doteq |\epsilon_1+\epsilon_2| \leq |\epsilon_1|+|\epsilon_2| \leq 2\cdot eps$$

Dabei entsprechen die obigen Konditionszahlen gerade den Vorfaktoren von $|\epsilon_1|$ und $|\epsilon_2|$.

Ein entsprechendes Resultat gilt auch für die Division, siehe Übungen.

Beispiele 1.26: Stabilität der Exponentiation

Zur Berechnung von Potenzen a^x für $a, x \in \mathbb{R},\ a>0$ kann man basierend auf der Darstellung

$$a^x = \exp(x\cdot\log(e))$$

den folgenden Algorithmus verwenden

$$x_1 = \begin{pmatrix} x \\ a \end{pmatrix} = \begin{pmatrix} (x_1)_1 \\ (x_1)_2 \end{pmatrix},\ \phi_1 : x_1 \to \begin{pmatrix} (x_1)_1 \\ \log((x_1)_2) \end{pmatrix},\ \psi_1 : x_1 \to \exp((x_1)_1, \log((x_1)_2))$$

$$x_2 = \begin{pmatrix} x \\ \log(a) \end{pmatrix} = \begin{pmatrix} (x_2)_1 \\ (x_2)_2 \end{pmatrix},\ \phi_2 : x_2 \to (x_2)_1\cdot(x_2)_2,\ \psi_2 i : x_2 \to \exp((x_2)_1\cdot(x_2)_2)$$

$$x_3 = x\cdot\log(a),\ \phi_3 : x_3 \to \exp(x_3),\ \psi_3 : x_3 \to \exp(x_3)$$

Für die differentiellen Konditionen des relativen Fehlers gemäß (1.37) erhält Mantisse

$$(\kappa_1)_1=\left|\frac{x_{11}}{y}\cdot\frac{\partial\psi_1}{\partial(x_1)_1}(x_1)\right|=\left|\frac{(x_1)_1}{y}\cdot\exp((x_1)_1\cdot\log((x_1)_2))\cdot\log((x_1)_2)\right|=|x\cdot\log(a)|$$
$$(\kappa_1)_2=\left|\frac{x_{12}}{y}\cdot\frac{\partial\psi_1}{\partial(x_1)_2}(x_1)\right|=\left|\frac{(x_1)_2}{y}\cdot\exp((x_1)_1\cdot\log((x_1)_2))\cdot\frac{(x_1)_1}{(x_1)_2}\right|=|x|$$
$$(\kappa_2)_1=\left|\frac{x_{21}}{y}\cdot\frac{\partial\psi_2}{\partial(x_2)_1}(x_2)\right|=\left|\frac{(x_2)_1}{y}\cdot\exp((x_2)_1\cdot(x_2)_2)\cdot(x_2)_2\right|=|x\cdot\log(a)|$$
$$(\kappa_2)_2=\left|\frac{x_{22}}{y}\cdot\frac{\partial\psi_2}{\partial(x_2)_2}(x_2)\right|=\left|\frac{(x_2)_2}{y}\cdot\exp((x_2)_1\cdot(x_2)_2)\cdot(x_2)_1\right|=|x\cdot\log(a)|$$
$$(\kappa_3)=\left|\frac{x_3}{y}\cdot\frac{\partial\psi_3}{\partial x_3}(x_3)\right|=\left|\frac{x_3}{y}\cdot\exp(x_3)\right|=|x\cdot\log(a)|$$

Datum: 13.11.2002

$$a^x=e^{x\cdot\log(a)}$$

$x_1=\begin{pmatrix}x\\a\end{pmatrix}$	$\kappa_{11}=\lvert x\cdot\log(a)\rvert$	$k_{ij}=\left\lvert\frac{x_{ij}}{y}\cdot\frac{\partial\cdot\psi_i}{\partial\cdot x_{ij}}(x_i)\right\rvert$
	$\kappa_{12}=\lvert x\rvert$	
$x_2=\begin{pmatrix}x\\\log(a)\end{pmatrix}$	$\kappa_{21}=\lvert x\cdot\log(a)\rvert$	$i=1,\dots,l$
	$\kappa_{22}=\lvert x\cdot\log(a)\rvert$	$j=1,\dots,n_i$
$x_3=x\cdot\log(a)$	$\kappa_3=\lvert x\cdot\log(a)\rvert$	

Dabei beschreiben κ_{11} und κ_{12} die differentielle Kondition des Problems.
Wegen $\kappa_{21}=\kappa_{22}=\kappa_3=\kappa_{11}$ ist der Algorithmus stabil.

Beispiel 1.27

Der auf (0.3) aufbauende Algorithmus zur Berechnung von y=f(x) kann folgendermaßen geschrieben werden:

$x_1=x$	$\phi_1: x_1\mapsto {x_1}^2$	$\psi_1: x_1\mapsto\frac{1-\sqrt{1-{x_1}^2}}{{x_1}^2}$
$x_2=x^2$	$\phi_2: x_2\mapsto\begin{pmatrix}1-x_2\\x_2\end{pmatrix}$	$\psi_2: x_2\mapsto\frac{1-\sqrt{1-x_2}}{x_2}$
$x_3=\begin{pmatrix}1-x\\x\end{pmatrix}$	$\phi_3: x_3\mapsto\begin{pmatrix}\sqrt{x_{31}}\\x_{32}\end{pmatrix}$	$\psi_3: x_3\mapsto\frac{1-\sqrt{x_{31}}}{x_{32}}$
$x_4=\begin{pmatrix}\sqrt{1-x^2}\\x^2\end{pmatrix}$	$\phi_4: x_4\mapsto\begin{pmatrix}1-x_{41}\\x_{42}\end{pmatrix}$	$\psi_4: x_4\mapsto\frac{1-x_{41}}{x_{42}}$
$x_4=\begin{pmatrix}1-\sqrt{1-x^2}\\x^2\end{pmatrix}$	$\phi_5: x_5\mapsto\frac{x_{51}}{x_{52}}$	$\psi_5: x_5\mapsto\frac{x_{51}}{x_{52}}$

Für $x\in[-1,1]$ mit $|x|$ hinreichend kleine erhält man in erster Näherung:

$$x_1 = x$$
$$x_2 = x^2$$
$$x_3 = \begin{pmatrix} 1-x^2 \\ x^2 \end{pmatrix}$$
$$x_4 = \begin{pmatrix} 1-\frac{1}{2}\cdot x^2 \\ x^2 \end{pmatrix}$$
$$x_5 = \begin{pmatrix} \frac{1}{2}\cdot x^2 \\ x^2 \end{pmatrix}$$
$$y = \frac{1}{2}$$

Wegen $\psi_1 = f$ liefert die Reihenentwicklung aus Beispiel 1.21:

$$\psi_1(x_2) \doteq \frac{1}{2} + \frac{1}{8}\cdot x_1^{\,2},\ \psi_2(x_2) \doteq \frac{1}{2} + \frac{1}{8}\cdot x_2$$

beziehungsweise

$$\frac{\partial \psi_1}{\partial x_1}(x_1) \doteq \frac{1}{4}\cdot x_1,\ \frac{\partial \psi_2}{\partial x_2}(x_2) \doteq \frac{1}{8}$$

und damit für die differentiellen Konditionen des relativen Fehlers

$$\kappa_1 = \left| \frac{x_1}{y}\cdot\frac{\partial \psi_1}{\partial x_1}(x_1) \right| \doteq \left| \frac{x_1}{y}\cdot\frac{1}{4}\cdot x_1 \right| \doteq \frac{1}{2}\cdot x^2$$
$$\kappa_2 = \left| \frac{x_2}{y}\cdot\frac{\partial \psi_2}{\partial x_2}(x_2) \right| \doteq \left| \frac{x_2}{y}\cdot\frac{1}{8} \right| \doteq \frac{1}{4}\cdot x^2$$
$$\kappa_{31} = \left| \frac{x_{31}}{y}\cdot\frac{\partial \psi_3}{\partial x_{31}}(x_3) \right| = \left| \frac{x_{31}}{y}\cdot\frac{1}{x_{32}}\cdot\frac{1}{2\cdot\sqrt{x_{31}}} \right| \doteq \frac{1}{x^2}$$
$$\kappa_{32} = \left| \frac{x_{32}}{y}\cdot\frac{\partial \psi_3}{\partial x_{32}}(x_3) \right| = \left| \frac{x_{32}}{y}\cdot\frac{1-\sqrt{x_{31}}}{x_{32}^{\,2}} \right| \doteq \left| \frac{x^2}{\frac{1}{2}}\cdot\frac{\frac{1}{2}\cdot x^2}{x^4} \right| = 1$$
$$\kappa_{41} = \left| \frac{x_{41}}{y}\cdot\frac{\partial \psi_4}{\partial x_{41}}(x_4) \right| = \left| \frac{x_{41}}{y}\cdot\frac{1}{x_{42}} \right| \doteq \frac{2}{x^2}$$
$$\kappa_{42} = \left| \frac{x_{42}}{y}\cdot\frac{\partial \psi_4}{\partial x_{42}}(x_4) \right| = \left| \frac{x_{42}}{y}\cdot\frac{1-x_{41}}{x_{42}^{\,2}} \right| \doteq 1$$
$$\kappa_{51} = \left| \frac{x_{51}}{y}\cdot\frac{\partial \psi_5}{\partial x_{51}}(x_5) \right| = \left| \frac{x_{51}}{y}\cdot\frac{1}{x_{52}} \right| = 1$$
$$\kappa_{52} = \left| \frac{x_{52}}{y}\cdot\frac{\partial \psi_5}{\partial x_{52}}(x_5) \right| = \left| \frac{x_{52}}{y}\cdot\frac{x_{51}}{x_{52}^{\,2}} \right| = 1$$

Der vorliegende Algorithmus ist also für x mit $|x|$ hinreichend klein instabil ($\kappa_{31}, \kappa_{41} \gg \kappa_1$). Insbesondere werden Fehler bei der Berechnung bzw. Rundung von x_{31} und x_{41} extrem verstärkt. Dies ist zurückzuführen auf die anschließende Subtraktion mit Auslöschung führender Stellen.

2 Lineare Gleichungssysteme

Gesucht sei $x \in \mathbb{R}^n$ mit

(2.1) $A \cdot x = b$

wobei $A \in \mathbb{R}^{n,n}$ nichtsingulär und $b \in \mathbb{R}^n$.

Zum Messen von Fehlern bei Matrizen verwenden wir zu gegebener Vektornorm durch

(2.2) $$\|A\| = \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{\|A \cdot x\|}{\|x\|} = \sup_{\|x\|=1} \|A \cdot x\|$$

definierte sogenannte Operatornorm. Neben

(2.3)
(a) $\forall (A \in \mathbb{R}^{n,n} \text{ mit } A \neq 0) : (\|A\| > 0)$
(b) $\forall (A \in \mathbb{R}^{n,n}) : \forall (\alpha \in \mathbb{R}) : (\|\alpha \cdot A\| = |\alpha| \cdot \|A\|)$
(c) $\forall (A \in \mathbb{R}^{n,n}) : \forall (B \in \mathbb{R}^{n,n}) : (\|A + B\| \leq \|A\| + \|B\|)$

haben diese noch die Eigenschaft

(2.4)
(a) $\forall (A \in \mathbb{R}^{n,n}) : \forall (B \in \mathbb{R}^{n,n}) : (\|A \cdot B\| \leq \|A\| \cdot \|B\|)$ (Submultiplikativität)
(b) $\forall (A \in \mathbb{R}^{n,n}) : \forall (x \in \mathbb{R}^n) : (\|A \cdot x\| = \|A\| \cdot \|x\|)$ (Verträglichkeit)

Zu den Vektoren (1.20) gehören die Operatornormen

(2.5)
(a) $\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|$ (Spaltensummennorm)
(b) $\|A\|_2 = \max \left\{ \sqrt{\lambda} \,\middle|\, \lambda \text{ ist Eigenwert von } A^T \cdot A \right\}$ (Spektralnorm)
(c) $\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|$ (Zeilensummennorm)

2.1 Kondition

Für A nichtsingulär besitzt (2.1) die eindeutige Lösung

(2.6) $x = A^{-1} \cdot b$

Wegen der Stetigkeit der Inversenbildung und der Matrixvektormultiplikation ist das Problem wohlgestellt. In erster Näherung gilt für ein gestörtes Problem:

$$\begin{aligned} x + \Delta x &= (A + \Delta A)^{-1} \cdot (b + \Delta b) \\ &= [A \cdot (I + A^{-1} \cdot \Delta A)]^{-1} \cdot (b + \Delta b) \\ &= (I + A^{-1} \cdot \Delta A)^{-1} \cdot A^{-1} \cdot (b + \Delta b) \\ &\doteq (I - A^{-1} \cdot \Delta A) \cdot A^{-1} \cdot (b + \Delta b) \\ &\doteq A^{-1} \cdot b + A^{-1} \cdot \Delta b - A^{-1} \cdot \Delta A \cdot A^{-1} \cdot b \end{aligned}$$

Das heißt:

(2.6') $\Delta x = A^{-1} \cdot \Delta b - A^{-1} \cdot \Delta A \cdot x$

Mit einer Vektornorm und zugehöriger Operatornorm gilt

(2.7) $\|\Delta x\| \dot{\leq} \|A^{-1}\| \cdot \|\Delta b\| + \|A^{-1}\| \cdot \|x\| \cdot \|\Delta A\|$

und

(2.8) $$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|A \cdot x\|}{\|x\|} \cdot \frac{\|\Delta b\|}{\|b\|} + \frac{\|A^{-1}\| \cdot \|x\| \cdot \|A\|}{\|x\|} \cdot \frac{\|\Delta A\|}{\|A\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right)$$ (relativer Fehler)

Die Größe

(2.9) $\text{cond}_{\|\cdot\|}(A)=\|A\|\cdot\|A^{-1}\|$

heißt Kondition der Matrix A und beschreibt die (differentielle) Verstärkung des relativen Fehlers der Daten A und b. Im Fall (2.5) schreibt man auch

(2.10) $\text{cond}_p(A)=\|A\|_P\cdot\|A^{-1}\|_P$ für $p\in\{1,2,\infty\}$

Datum 22.11.2002

2.2 Gauß-Elimination

Schreibt man (2.1) ausführlicher als

$$\begin{aligned} a_{11}\cdot x_1+a_{12}\cdot x_2+\ldots+a_{1n}\cdot x_n&=b_1\\ a_{21}\cdot x_1+a_{22}\cdot x_2+\ldots+a_{2n}\cdot x_n&=b_2\\ &\vdots\\ a_{n1}\cdot x_1+a_{n2}\cdot x_2+\ldots+a_{nn}\cdot x_n&=b_n \end{aligned} \tag{2.11}$$

So kann man durch Vertauschen von Zeilen und Umbennen der Größen stets erreichen, dass $a_{11}\neq 0$ ist (sonst wäre A singulär). Die Idee der Gauß-Elimination ist dann, durch Addieren von Vielfachen der ersten Gleichung zu den anderen, aus diesen x_1 zu eliminieren. Man erhält

$$\begin{aligned} a_{11}\cdot x_1+a_{12}\cdot x_2+\ldots+a_{1n}\cdot x_n&=b_1\\ a_{22}\cdot x_2+\ldots+a_{2n}\cdot x_n&=b_2\\ &\vdots\\ a_{n2}\cdot x_2+\ldots+a_{nn}\cdot x_n&=b_n \end{aligned} \tag{2.12}$$

Dabei ändert sich die Lösung nicht. Sukzessives Anwenden liefert liefert nach n-1 Schritten ein System der Form:

$$\begin{aligned} r_{11}\cdot x_1+r_{12}\cdot x_2+\ldots+r_{1n}\cdot x_n&=b_1\\ r_{22}\cdot x_2+\ldots+r_{2n}\cdot x_n&=b_2\\ \ddots\quad&\vdots\\ r_{nn}\cdot x_n&=b_n \end{aligned}$$

mit $\forall(i=1,\ldots,n.):(r_{ii}\neq 0)$, Daraus kann man x beginnend mit x_n leicht bestimmen (sogenannte Rückwärtssubstitution).

Algorithmus 2.1: Gauß-Elimination mit Zeilentausch

Der folgende Algorithmus erzeugt (2.13) aus (2.11):

Schleife i=1,...,n:
Suche ein Element $a_{pi}\neq 0$ mit $p\in\{i,\ldots,n\}$ (sogenannte Pivotsuche)
Sind alle $a_{pi}\neq 0$, so ist A singulär $\rightarrow$ stop . Andernfalls vertausche i-te und p-te Zeile
Schleife k=i+1,...,n:
Setze $l_{ki}=\frac{a_{ki}}{a_{ii}}$ Setze $\forall(j=i+1,\ldots,n):(a_{kj}=a_{kj}-l_{ki}\cdot a_{ij})$ Setze $b_k=b_k-l_{ki}\cdot b_i$

Dabei sind die $\forall(1\leqslant i\leqslant j\leqslant n):(r_{ij})$, durch die entsprechenden letzten Werte von a_{ij} und die $\forall(1\leqslant i\leqslant n):(y_i)$, durch die entsprechenden letzten Werte von b_i gegeben.

Der Aufwand in (2.14) beträgt im wesentlichen eine Operation (Addition+Multiplikation) pro Berechnung eines Wertes a_{kj} in der innersten Schleife, das heißt: der Gesamtaufwand beträgt in erster Näherung:

$$(n-1)^2+(n-2)^2+\ldots+4+1=\sum_{i=1}^{n-1}(i^2)\doteq\frac{1}{3}\cdot n^3 \text{ Operationen.}$$

Die gebräuchlichste Pivotsuche in (2.14) ist gegeben durch die Bedingung

(2.15) $|a_{pi}|=\max_{k=i,\ldots,n}(|a_{ki}|)$ (Spaltenpivotsuche)

Es gilt dann:

(2.16) $|l_{ki}|\leqslant 1$

Eine solche Pivotsuche sucht sich also die Zeile als „Referenzzeile“ (Zeile, die nicht geändert wird, aber auf andere Zeilen draufaddiert wird), die das betragsmäßig größte Pivot-Element zu bieten hat. [Der Grund ist vermutlich, dass beim Verrechnen mit den anderen möglichen Pivot-Elementen aus anderen Zeilen kein so großer Fehler durch herunterskalieren entsteht, als er durch heraufskalieren entstehen könnte.]

Hat man Algorithmus 2.1 erfolgreich durchgeführt, so erhält man die Lösung x folgendermaßen:

Algorithmus 2.2 (Rückwärtssubstitution)

Im Anschluss an (2.14) ist folgendes durchzuführen:

(2.17)

Schleife i=n,...,1:	
	Setze $b_i=b_i-a_{ij}\cdot x_j,\ j=n,\ldots,i+1$
	Setze $x_i=\frac{b_i}{a_{ii}}$

Der Aufwand in (2.17) beträgt in erster Näherung

$$0+1+\ldots+(n-1)=\sum_{i=1}^{n}(i-1)\doteq\frac{1}{2}\cdot n^2 \text{ Operationen}$$

Bemerkung 2.3

Will man mehrere Gleichungssysteme mit gleicher Matrix A und verschiedenen Seiten b lösen, so ist es in Algorithmus 2.1 besser, die Berechnungen betreffend A und b zu entkoppeln. Zusätzlich muss man sich dann die vorgenommenen Zeilentausche merken, während man die Werte l_{ki} auf den Speicherplatz a_{ki} ablegen kann. Auf diese Weise zerfällt Algorithmus 2.1 in zwei Teile (sogenannte Zerlegung der Matrix und sogenannte Vorwärtssubstitution zur entsprechenden Behandlung von b).

In Programmbibliotheken findet man deshalb üblicherweise Routinen zur Zerlegung von A (sogenannt: FACTOR) sowie zur Berechnung einer Lösung zu gegebenem b und zerlegtem A (sogenannt: SOLVE), vergleiche Übungsblatt.

Bemerkung 2.4

Den Übergang von (2.11) auf (2.12) lässt sich auch beschreiben durch

(2.18) $$A = E \cdot A$$

wobei

(2.19) $$E = \begin{bmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{bmatrix}$$

Betrachtet man

(2.20) $$E^{-1} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & & & 1 \end{bmatrix}$$

so gilt

(2.21) $$\begin{aligned} \operatorname{cond}_\infty(A') &= \|A'\|_\infty \cdot \|A'^{-1}\|_\infty = \|E \cdot A\|_\infty \cdot \|A^{-1} \cdot E^{-1}\|_\infty \\ &\leqslant \|E\|_\infty \cdot \|A\|_\infty \cdot \|A^{-1}\|_\infty \cdot \|E^{-1}\|_\infty \\ &= \operatorname{cond}_\infty(E) \cdot \operatorname{cond}_\infty(A) \end{aligned}$$

Wobei im Fall (2.15) wegen (2.16) gilt:

(2.22) $$\operatorname{cond}_\infty(E) = \|E\|_\infty \cdot \|E^{-1}\|_\infty \leqslant 2 \cdot 2 = 4$$

Die Kondition des zu lösenden Gleichungssystems verschlechtert sich also in jedem Schritt der Gauß-Elimination eventuell um einen Faktor 4. In diesem Sinn ist Gauß-Elimination selbst mit Spaltenpivotsuche (in der angegeben Form natürlich) instabil. In praktischen Problemen tritt diese Instabilität allerdings normalerweise nicht auf.

Datum: 27.11.2002

[...wegen Abwesendheit nicht direkt mitgeschrieben sondern abgeschrieben von Matthias Quasthof...]

Bemerkung 2.5

Beschreibt $\Pi \in \mathbb{R}^{n,n}$ die im Algorithmus 2.1 gegebenen $A \in \mathbb{R}^{n,n}$ vorzunehmenden Zeilentausche und setzt man $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & \ddots & 0 & 0 \\ \vdots & & \ddots & 0 \\ l_{n1} & \dots & l_{nn-1} & 1 \end{bmatrix}$, $R = \begin{bmatrix} r_{11} & \dots & \dots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & r_{nn} \end{bmatrix}$, so gilt $\Pi \cdot A = L \cdot R$ (eine sogenannte Dreieckszerlegung).
[

Beispiel

Sei $A = \begin{pmatrix} 4 & -2 & 6 \\ -2 & 2 & -7 \\ 6 & -7 & 29 \end{pmatrix}$. Dann gilt:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & -2 & 6 \\ -2 & 2 & -7 \\ 6 & -7 & 29 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 & -7 & 29 \\ -2 & 2 & -7 \\ 4 & -2 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 & -7 & 29 \\ 0 & \frac{6}{3} - \frac{7}{3} & \frac{-21}{3} + \frac{29}{3} \\ 0 & \frac{-6}{3} - \frac{2}{3} \cdot (-7) & \frac{18}{3} - \frac{2}{3} \cdot 29 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ \frac{2}{3} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 & -7 & 29 \\ 0 & -\frac{1}{3} & \frac{8}{3} \\ 0 & \frac{8}{3} & \frac{-40}{3} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 & -7 & 29 \\ 0 & \frac{8}{3} & \frac{-40}{3} \\ 0 & -\frac{1}{3} & \frac{8}{3} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 & -7 & 29 \\ 0 & \frac{8}{3} & \frac{-40}{3} \\ 0 & -\frac{1}{3} & \frac{8}{3} \end{pmatrix}$$

$$\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}\cdot A=\begin{pmatrix}1&0&0\\\frac{2}{3}&1&0\\-\frac{1}{3}&-\frac{1}{8}&1\end{pmatrix}\cdot\begin{pmatrix}6&-7&29\\0&\frac{8}{3}&\frac{-40}{3}\\0&0&1\end{pmatrix}=\Pi\cdot A=L\cdot R$$

]

2.3 Cholesky-Zerlegung

Eine wichtigte Klasse von Matrizen im $\mathbb{R}^{n,n}$ sind die symmetrischen, positiv-definiten Matrizen, das heißt: Matrizen $A\in\mathbb{R}^{n,n}$ mit:
(2.23)
1. Symmetrie: $A^T=A$
2. positive Definitheit: $\forall((x\in R^n)\wedge(x\neq 0)):(x^T\cdot A\cdot x>0)$

Ein Zeilentausch, wie er bei der Gauß-Elimination auftreten kann, würde die Symmetrie sofort zerstören. Als Folge können wichtige Eigenschaften, die auf der Symmetrie basieren, verloren gehen. Andererseits könnte man versuchen, die Symmetrie auszunutzen, um einen effizienteren Algorithmus zu erhalten.

Satz 2.6: Cholesky-Zerlegung

Zu jeder symmetrisch, positiv-definiten Matrix $A\in\mathbb{R}^{n,n}$ gibt es eine untere Dreiecksmatrix $L\in\mathbb{R}^{n,n}$ mit
(2.24) $A=L\cdot L^T$

Beweis

- für $n=1$: Die Behauptung ist trivial. Sei $A=(a_{11})$ mit $a_{11}>0$. Dann ist $L=(\sqrt{a_{11}})$.
- für $n\geq 1$: Sei die Behauptung richtig und sei $A\in\mathbb{R}^{n+1,n+1}$ symmetrisch und positiv definiert. Partitioniert man A folgendermaßen: $A=\begin{bmatrix}A'&v\\v^T&c\end{bmatrix}$ mit $A'\in R^{n,n}$, so ist A' ebenfalls symmetrisch und positiv definiert.
  Nach Induktionsvoraussetzung gilt: $\exists(L'\in\mathbb{R}^{n,n}):(A'=L'\cdot L'^T)$
  Mit dem Ansatz $A=\begin{bmatrix}L'\cdot L'^T&v\\v^T&c\end{bmatrix}=\begin{bmatrix}L'&0\\l^T&d\end{bmatrix}\cdot\begin{bmatrix}L'^T&l\\0&d\end{bmatrix}=\begin{bmatrix}L'\cdot L'^T&L'\cdot l\\l^T\cdot L^T&l^T\cdot l+d^2\end{bmatrix}$ erhält man die Bedingungen
  - $L'\cdot l=v$
  - $l^T\cdot t+d^2=c$

  Da positiv-definite Matrizen insbesondere nichtsingulär sind, ist mit A' auch L' nichtsingulär und damit $l=L'^{-1}\cdot v$.
  Dies folgt aber aus:

$$
\begin{aligned}
0 &< \begin{pmatrix} -l^T \cdot L'^{-1} & 1 \end{pmatrix} \cdot \begin{bmatrix} L' \cdot L'^T & v \\ v^T & c \end{bmatrix} \cdot \begin{pmatrix} -(L'^{-1})^T \cdot l \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} -l^T \cdot L'^{-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ c - l^T \cdot l \end{pmatrix} \\
&= c \cdot l^T \cdot l
\end{aligned}
$$

Zur Berechnung der Cholesky-Zerlegung beachtet man

$$a_{ij} = \sum_{k=1}^{\min(i,j)} (l_{ik} \cdot l_{jk}) \tag{2.25}$$

wovon man wegen der Symmetrie zum Beispiel nur $i \geq j$ beachten muss. Es gilt also

$$\begin{aligned} &\text{(a) } a_{jj} = l_{j1}^2 + l_{j2}^2 + \ldots + l_{jj}^2 \\ &\text{(b) } a_{ij} = l_{i1} \cdot l_{j1} + l_{i2} \cdot l_{j2} + \ldots + l_{ij} \cdot l_{jj} \end{aligned} \tag{2.26}$$

beziehungsweise

$$\begin{aligned} &\text{(a) } l_{jj}^2 = a_{jj} - l_{j1}^2 - \ldots - l_{jj-1}^2 \\ &\text{(b) } \forall (i > j): \left(l_{jj}^2 = \frac{a_{ij} - l_{i1} \cdot l_{j1} - \ldots - l_{ij-1} \cdot l_{jj-1}}{l_{jj}} \right) \end{aligned} \tag{2.27}$$

und man kann L spaltenweise berechnen.

Algorithmus 2.7: Cholesky-Zerlegung

Der folgende Algorithmus berechnet für ein gegebenes symmetrisches, positiv definiertes $A \in \mathbb{R}^{n,n}$ die Zerlegung (2.24):

Schleife j=1,...,n:	
	Setze $s = a_{jj} - l_{j1}^2 - \ldots - l_{jj-1}^2$ Ist $s \leq 0$, so ist A nicht positiv definiert. Setze $l_{jj} = \sqrt{s}$
	Schleife i=j+1,...,n:
	Setze $l_{ij} = \frac{a_{ij} - l_{i1} \cdot l_{j1} - \ldots - l_{ij-1} \cdot l_{jj-1}}{l_{jj}}$

Der Aufwand in (2.28) beträgt in erster Näherung

$$\sum_{j=1}^{n} ((n-j) \cdot j) = \sum_{j=1}^{n} (n \cdot j - j^2) = n \cdot \left(\sum_{j=1}^{n} (j) - \sum_{j=1}^{n} (j^2) \right) = \frac{1}{2} \cdot n^3 - \frac{1}{3} \cdot n^3 = \frac{1}{6} \cdot n^3$$

Operationen sowie n Quadratwurzeln. Zur Lösung eines zugehörigen linearen Gleichungssystems fällt dann jeweils eine Vorwärts- und Rückwärtssubstitution an.

[Es ergeben sich also folgende Gleichungen zur Berechnung der Cholesky-zerlegten Matrix l:

- $l_{11} = \sqrt{a_{11}}$ (Linke obere Zelle)
- $\forall (i = 2, \ldots, n): \left(l_{i1} = \frac{a_{i1}}{l_{11}} \right)$ (Rest der ersten Spalte)

- $\forall(i=2,\ldots,n):\left(l_{ii}=\sqrt{a_{ii}-\sum_{j=1}^{i-1}\left(l_{ij}{}^2\right)}\right)$ (Zellen in der Hauptdiagonale)
- $l_{ij}=\frac{1}{l_{jj}}\cdot\left(a_{ij}-\sum_{s=1}^{j-1}\left(l_{is}\cdot l_{js}\right)\right)$ (Reste aller weiterer Spalten)

Beispiel

Sei $A=\begin{pmatrix}4&-2&6\\-2&2&-7\\6&-7&29\end{pmatrix}$. Dann ist

$$L=\begin{pmatrix}2&&0&0\\-1&&1&0\\3&\frac{1}{1}\cdot(-7-(3\cdot(-1)))&?\end{pmatrix}=\begin{pmatrix}2&0&0\\-1&1&0\\3&-4&\sqrt{29-9-16}\end{pmatrix}=\begin{pmatrix}2&0&0\\-1&1&0\\3&-4&2\end{pmatrix}$$

]

2.4 QR-Zerlegung

Mit Sicht auf Bemerkung 2.4 wäre für (2.1) eine Lösungsverfahren wünschenswert, das durch Eliminationsmatrizen E mit $\mathrm{cond}_\infty(E)=1$ beschrieben werden kann.
Matrizen mit dieser Eigenschaft beschreiben aber leider nur Zeilentausche und Vorzeichenwechsel, das heißt: Man kann damit nicht das gegebene lineare Gleichungssystem vereinfachen.
Hingegen ist die Klasse der Matrizen E mit $\mathrm{cond}_2(E)=1$ hinreichend groß.

Definition 2.8 (orthogonale Matrizen)

Eine Matrix $Q\in\mathbb{R}^{n,n}$ heißt orthogonal, wenn
(2.28) $Q^T\cdot Q=$ Einheitsmatrix

Offensichtlich ist ein Orthogonales Q invertierbar mit $Q^{-1}=Q^T$. Damit gilt auch $Q\cdot Q^T=I$ und außerdem $\|Q\|_2=\|Q^T\|_2=\|Q^{-1}\|_2=1$ nach (2.5b), also
(2.29) $\forall((Q\in\mathbb{R}^{n,n})\wedge(Q \text{ ist orthogonal})):(\mathrm{cond}_2(Q)=1)$
Wegen $\|Q\cdot x\|_2{}^2=x^t\cdot Q^T\cdot Q\cdot x=x^T\cdot x=\|x\|_2{}^2$ gilt:
(2.30) $\forall(x\in\mathbb{R}^n):\forall((Q\in\mathbb{R}^{n,n})\wedge(Q \text{ ist orthogonal})):(\|Q\cdot x\|_2=\|x\|_2)$

Anmerkung des Mitschreibers:
- Für orthogonale Matrizen A gilt: $\det(A)=\pm1$
- Für orthogonale Matrizen A gilt: $A^T=A^{-1}$ ($A^{-1}\cdot A=E=$ Einheitsmatrix)
- Beispiel: $\begin{pmatrix}0&1\\1&0\end{pmatrix}\cdot\begin{pmatrix}0&1\\1&0\end{pmatrix}=\begin{pmatrix}1&0\\0&1\end{pmatrix}$

- für Gauß-Elimination gilt:
  $A' = E \cdot A,\ \mathrm{cond}_\infty(E) \leq 4$
- für orthogonale Matrizen gilt:
  $Q^T \cdot Q = I,\ \mathrm{cond}_2(Q) = 1$
  $Q^{-1} = Q^T,\ \|Q\|_2 = \|Q^T\|_2 = \|Q^{-1}\|_2 = 1$
  $\forall (x): (\|Q \cdot x\|_2 = \|x\|_2)$

Definition 2.9: Householder-Transformation

Matrizen $P \in \mathbb{R}^{n,n}$ der Form

(2.32) $\forall (w \in \mathbb{R}^n): \forall (w^T \cdot w = 1): (P = I - 2 \cdot w \cdot w^T)$

heißen **Housholder-Transformationen**.

Offensichtlich ist eine Housholder-Transformation P-symmetrisch und wegen

$$P^T \cdot P = (I - 2 \cdot w \cdot w^T) \cdot (I - 2 \cdot w \cdot w^T) = I - 4 \cdot w \cdot w^T + 4 \cdot w \cdot w^T \cdot w \cdot w^T = I$$

orthogonal.

Die Frage ist nun, ob man P so wählen kann, dass es wie E in Bemerkung 2.4 in der ersten Spalte von A unterhalb des ersten Eintrags Nullen erzeugt.

Bezeichnet $z \in \mathbb{R}^n,\ z \neq 0$ die erste Spalte von A, so ist also ein $w \in \mathbb{R}^n$ mit $w^T \cdot w = 1$ gesucht, sodass

(2.33) $P \cdot z = (I - w \cdot w^T) \cdot z = \alpha \cdot e_1,\ e_1 = (1,0,\ldots,0)^T$

Aus der Forderung, dass P orthogonal ist, folgt wegen (2.31) sofort

(2.34) $\alpha = \pm\sigma,\ \sigma = \|z\|_2 = \sqrt{z^T \cdot z}$

Weiterhin folgt aus (2.33)

$$z - 2 \cdot w \cdot w^T \cdot z = \alpha \cdot e_1$$

beziehungsweise

$$2 \cdot (w^T \cdot z) \cdot w = z - \alpha \cdot e_1$$

Zusammen mit $w^T \cdot w = 1$ ergibt sich daraus

(2.35) $$w = \frac{z - \alpha \cdot e_1}{\|z - \alpha \cdot e_1\|_2}$$

Um Auslöschung in $z - \alpha \cdot e_1$ zu vermeiden, wählt man das Vorzeichen von α entsprechend.

(2.36) $\mathrm{signum}(a) = -\mathrm{signum}(z_1),\ z = (z_1,\ldots,z_n)^T$

Damit gilt:

$$\begin{aligned} \|z - \alpha \cdot e_1\|_2^2 &= (z_1 - \alpha)^2 + z_2^2 + \ldots + z_n^2 \\ &= z_1^2 + 2 \cdot |z_1| \cdot |\alpha| + \alpha^2 + z_2^2 + \ldots + z_n^2 \\ &= 2 \cdot \sigma^2 + 2 \cdot \sigma \cdot |z_1| \\ &= 2 \cdot \sigma \cdot (\sigma + |z_1|) \end{aligned}$$

beziehungsweise

(2.37) $P = I + \beta \cdot u \cdot u^T$

mit

(2.38) (a) $\beta = \frac{-1}{\sigma \cdot (\sigma + |z_1|)}$

(b) $u = z - \alpha \cdot e_1$

(Man kann sich also die Division durch β sparen, womit man Fehlerquellen umgeht.)

Satz 2.10 (QR-Zerlegung)

Zu jeder nichtsingulären Matrix $A \in \mathbb{R}^{n,n}$ gibt es eine orthogonale Matrix $Q \in \mathbb{R}^{n,n}$ und eine nichtsinguläre obere Dreiecksmatrix $R \in \mathbb{R}^{n,n}$, sodass

(2.39) $A = Q \cdot R$

[Anmerkung des Mitschreibers: Eine obere Dreiecksmatrix ist eine Matrix $A = (a_{ij})$ mit $\forall (i > j) : (a_{ij} = 0)$. Das heißt also, dass alle Elemente unterhalb der Diagonalen gleich 0 sind.]

Beweis

Setzt man für $A = (a_1, \ldots, a_n)$ in der obigen Konstruktion zunächst $z = a_1 \neq 0$ ein, so erhält man mit dem zugehörigen P

$$\begin{aligned} P \cdot A &= P \cdot (a_1, \ldots, a_n) \\ &= (P \cdot a_1, \ldots, P \cdot a_n) \\ &= \begin{bmatrix} \alpha & * \\ 0 & * \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} \ldots r_{1n} \\ 0 & A' \end{bmatrix} \end{aligned}$$

Per Induktion besitzt A' eine QR-Zerlegung $A' = Q' \cdot R'$, das heißt:

$$P \cdot A = \begin{bmatrix} r_{11} & r_{12} \ldots r_{1n} \\ 0 & Q' \cdot R' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & Q' \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} \ldots r_{1n} \\ 0 & R' \end{bmatrix}$$

Mit

$$Q = P^T \cdot \begin{bmatrix} 1 & 0 \\ 0 & Q' \end{bmatrix} = P \cdot \begin{bmatrix} 1 & 0 \\ 0 & Q' \end{bmatrix}, \begin{bmatrix} r_{11} & r_{12} \ldots r_{1n} \\ 0 & R' \end{bmatrix}$$

erhält man daraus eine Zerlegung der Form (2.39).

Beachtet man, dass man $P \cdot y$ nie durch Berechnen der Housholder-Transformation P nach (2.37) mit ausschließender Matrix-Vektor-Multiplikation mit y bestimmen sollte, sondern wegen

$$P \cdot y = (I + \beta \cdot u \cdot u^T) \cdot y = y + \beta \cdot u \cdot (u^T \cdot u)$$

erst den Skalar $\beta = \beta \cdot (u^T \cdot y)$ und dann die Summe $y + \gamma \cdot u$ von Vektoren berechnent sollte, so erhält man den folgenden Algorithmus zur Lösung von (2.1):

Algorithmus 2.11

Der folgende Algorithmus berechnet die Lösung von (2.1) basierend auf einer QR-Zerlegung

von A

(2.40)

Schleife i=1,...,n:		
	Setze	$\tau = a_{ii}^{\ 2} + \ldots + a_{ni}^{\ 2};\ \sigma = \sqrt{\tau}$
	Ist	$\sigma = 0$, so ist A singulär ($\rightarrow$ stop)
	Setze	$\alpha = \begin{cases} \sigma & \text{für } a_{ii} \leqslant 0 \\ -\sigma & \text{für } a_{ii} > 0 \end{cases}$
	Setze	$d_i = \alpha;\ \beta = \frac{1}{\alpha \cdot a_{ii} - \tau};\ a_{ii} = a_{ii} - \alpha$
	Schleife k=i+1,...,n:	
		Setze $\gamma = \beta \cdot (a_{ii} \cdot a_{ik} + \ldots + a_{ni} \cdot a_{nk})$
		Setze $\forall (j = i, \ldots, n) : \left(a_{jk} = a_{jk} + \gamma \cdot a_{ji} \right)$
	Setze	$\gamma = \beta \cdot (a_{ii} \cdot b_i + \ldots + a_{ni} \cdot b_n)$
	Setze	$\forall (j = i, \ldots, n) : \left(b_j = b_j + \gamma \cdot a_{ji} \right)$
Schleife i=n,...,1:		
	Setze	$\forall (j = n, \ldots, i+1) : \left(b_i = b_i - a_{ij} \cdot x_j \right)$
	Setze	$x_i = \frac{b_i}{d_i}$

Die Transformationen $P_i = I + \beta_i \cdot u_i \cdot u_i^T$ werden dabei in der Form gespeichert , dass die u_i als Spalten im unteren Dreieck von A abgelegt sind und $\beta_i = \frac{1}{\alpha_i \cdot a_{ii} - \tau_i}$ sich durch $\alpha_i = d_i$; $\tau_i = \alpha_i^{\ 2}$; $a_{ii} = a_{ii} + x_i$ zurückgewinnen lässt.

Der obige Algorithmus benötigt in erster Näherung

$$\sum_{i=1}^{n} 2 \cdot (n-1)^2 \doteq 2 \cdot \sum_{i=1}^{n} i^2 = \frac{2}{3} \cdot n^3$$

Dazu kommen n Quadratwurzeln.

2.5 Lineare Ausgleichsrechnung

Gegeben seien Messungen (t_i, f_i), mit $t_i, f_i \in \mathbb{R}$, $i=1,\ldots,m$. Zu gegebener Funktion $\forall((j=1,\ldots,n)\wedge(n\leqslant m)):(\phi_j:\mathbb{R}\to\mathbb{R})$ sollen x_j so bestimmt werden, dass $f:\mathbb{R}\to\mathbb{R}$ definiert durch

(2.41) $$f(t)=\sum_{j=1}^{n}\left(x_j\cdot\phi_j(t)\right)$$

die Messungen möglichst gut beschreibt.

Als Maß dafür verwenden wir, dass die Abweichung $r\geqslant 0$ mit

(2.42) $$r^2=\sum_{i=1}^{m}\left(w_i^2\cdot\left(f(t_i)-f_i\right)^2\right)=\sum_{j=1}^{m}\left(\sum_{j=1}^{n}\left(w_i\cdot\left(x_j\cdot\phi_j(t_i)-f_i\right)\right)\right)^2$$

zu gegebenem $w_i>0$, $i=1,\ldots,m$ möglichst klein sein soll. Dabei erlauben die w_i noch eine unterschiedliche Gewichtung der einzelnen Messungen. Setzt man

(2.43) a. $A=(a_{ij})$, $a_{ij}=w_i\cdot\phi_j(t_i)$
b. $b=(b_i)$, $b_i=w_i\cdot f_i$

so ist also zu $A\in\mathbb{R}^{m,n}$ und $b\in\mathbb{R}^m$ ein $x\in\mathbb{R}^n$ gesucht mit

(2.44) $$\min\left(\left\{\forall(x):\left(\|A\cdot x-b\|_2^{\,2}\right)\right\}\right)=\|A\cdot x-b\|_2^{\,2}$$

Ein typisches Resultat ist auch ein hinreichende Bedingung für die Wohlgestelltheit des Problems beinhaltet, ist der folgende Satz:

Satz 2.12

Die Spalten von $A\in\mathbb{R}^{m,n}$ seien linear unabhängig. Dann besitzt das Problem (2.44) eine eindeutige Lösung $x\in\mathbb{R}^n$. Diese ist gegeben als Lösung des linearen Gleichungssystems

(2.45) $A^T\cdot A\cdot x=A^T\cdot b$ (Normalgleichung)

wobei $A^T\cdot A$ symmetrisch und positiv definiert ist.

Leider ist die Berechnung von x mittels (2.45), etwa durch Cholesky-Zerlegung von $A^T\cdot A$, nicht stabil. Als alternative Grundlage zur Entwicklung eines Algorithmus verenden wir deshalb folgende Eigenschaft:

Satz 2.13

Zu jeder Matrix $A\in\mathbb{R}^{m,n}$ mit linear unabhängigen Spalten gibt es eine orthogonale Matrix $Q\in\mathbb{R}^{m,m}$ und eine nichtsinguläre obere Dreiecksmatrix $R\in\mathbb{R}^{n,n}$, sodass

(2.46) $$A=Q\cdot\begin{bmatrix}R\\0\end{bmatrix}$$

Damit gilt:

$$\begin{aligned}\|A\cdot x-b\|_2^2 &= \left\|Q\cdot\begin{bmatrix}R\\0\end{bmatrix}\cdot x-b\right\|_2^2\\ &= \left\|Q^T\cdot Q\cdot\begin{bmatrix}R\\0\end{bmatrix}\cdot x-Q^T\cdot b\right\|_2^2\\ &= \left\|I\cdot\begin{bmatrix}R\\0\end{bmatrix}\cdot x-Q^T\cdot b\right\|_2^2\\ &= \left\|\begin{bmatrix}R\\0\end{bmatrix}\cdot x-\begin{bmatrix}c\\d\end{bmatrix}\right\|_2^2\\ &= \|R\cdot x-c\|_2^2+\|d\|_2^2\end{aligned}$$

$$Q^T\cdot b=\begin{bmatrix}c\\d\end{bmatrix}\begin{matrix}-n\\-m-n\end{matrix}$$

was gerade für

(2.47) $x=R^{-1}\cdot c$

minimal wird.

Algorithmus 2.14

Man modifiziert Algorithmus 2.11 dahingehend, dass die Schleifen bei der Berechnung der Größen τ, γ, a_{jk} und b_j bis m statt bis n laufen. Die Abfrage $\gamma=0$ bedeutet $\sigma=0$ bedeutet dann, dass die Spalten von A nicht linear abhängig sind.

3 Interpolation

Bei der (linearen) Interpolation werden zu gegebenen Daten $\forall(i=0,\ldots,n):\left((t_i,f_i)\right.$ mit $\left.(t_i,f_i\in\mathbb{R})\right)$, mit p aarweise verschiedenen t_i (das heißt mit $\forall(i\neq j):(t_i\neq t_j)$) [(t_i,f_i) sind also Punkte in einem zweidimensionalen Koordinatensystem] und Funktionen $\forall(j=0,\ldots,n):\left(\phi_j:\mathbb{R}\to\mathbb{R}\right)$ Koeffizienten $x_j\in\mathbb{R}$ gesucht mit

(3.1) $$\forall(i=0,\ldots,n):\left(\sum_{j=0}^{n}\left(x_j\cdot\phi_j(t_i)\right)=f_i\right)$$

Gehören die Daten etwa zu einer Funktion f (das heißt: $\forall(i):\left(f(t_i)=f_i\right)$), so kann man dann mit der sogenannten Interpolierenden $\tilde{f}$, definiert durch

(3.2) $$\tilde{f}(t)=\sum_{j=0}^{n}\left(x_j\cdot\phi_j(t)\right)$$

näherungsweise Funktionswerte von f zwischen den Stützstellen t_i bestimmen. Man denke dabei an Funktionen f, deren Auswertung sehr teuer ist.

[x_j scheint hier die Rolle eines Auswählers zu spielen: Es gebe Funktionen ϕ_j, die irgendwie gestaltet seien. Durch geschickte Kombination dieser Funktionen (also durch geschicktes Setzen von x_j) sollte es möglich sein, den Graphen dieser kombinierten Funktion durch die Punkte (t_i,f_i) zu leiten. Geht die kombinierte Funktion durch alle diese Punkte („Stützstellen“), so wird angenommen, dass sie zwischen diesen Stellen ebenfalls sinnvolle Werte annimmt. Sind die Stützstellen Punkte aus einem Graphen einer bekannten Funktion f,

so wird angenommen, dass die kombinierte Funktion auch an den Stellen, die keine Stürztstellen sind, Werte annimmt, die den Werten der original-Funktion f entspricht. Diese kombinierte Funktion heißt hier $\tilde{f}$ und allgemein Interpolationsfunktion. Der Funktionswert von $\tilde{f}$ an der Stelle t ist eine Linearkombination mit der Basis $(\phi_0(t), \phi_1(t), \ldots, \phi_n(t))$. Eine Besonderheit ist, dass die Koeffizienten x_i für jede dieser Linearkombinationen gleich sind.]

Setzt man

(3.3) a. $A=(a_{ij})$, $a_{ij}=\phi_j(t_i)$
b. $b=(b_i)$, $b_i=f_i$

so ist (3.1) äquivalent zum linearen Gleichungssystem $A\cdot x=b$. Im folgenden sollen für speziell gewählte ϕ_i effiziente Methoden vorgestellt werden.

3.1 Polynom-Interpolation

Gesucht ist hier ein Polynom p vom Grad höchstens n (man schreibt $p\in\Pi_n$) sodass

(3.4) $\forall(i=0,\ldots,n):(p(t_i)=f_i)$

Satz 3.1

Gegeben seien $\forall(i=0,\ldots,n):(t_i, f_i\in\mathbb{R})$ mit $\forall(i\neq j):(t_i\neq t_j)$.
Dann gibt es genau ein Polynom $p\in\Pi_n$ mit (3.4)

Beweis

- Existenz:
  Die durch
  $$\forall(j=0,\ldots,n):\left(l_j(t)=\prod_{\substack{t=0\\k\neq j}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right)\right)$$
  definierten Funktionen l_j sind offensichtlich Polynome in Π_n (sogenannte Lagrange-Polynome). Für diese gilt:
  $$\forall(i=0,\ldots,n):\forall(j=0,\ldots,n):\left(l_j(t_i)=\delta_{ij}=\begin{cases}1 & \text{für } i=j\\ 0 & \text{für } i\neq j\end{cases}\right)$$
  [Damit wirken diese Lagrange-Polynome als „Selektoren“ für eine bestimmte Stützstelle. An allen Stützstellen, die nicht der durch i ausgewählten entsprechen, ist das Polynom 0 und wählt damit die Werte dieser Stützstellen nicht aus. An der Stützstelle t_i ist es 1 und wählt damit den Wert dieser Stützstelle aus. Es liefert idealerweise ein Maß dafür, wie stark die Auswahl mit den einzelnen Stützstellen korreliert. Leider - das werden wir noch sehen - gilt das nur für Stützstellen selbst, jedoch nicht für den Bereich zwischen den Stützstellen.]

  Damit genügt p_n definiert durch
  $$p(t)=\sum_{j=0}^{n}(f_j\cdot l_j(t))$$
  den geforderten Eigenschaften.
- Eindeutigkeit:
  Seien $p_1, p_2\in\Pi_n$ mit $\forall(i=0,\ldots,n):(p_1(t_i)=p_2(t_i)=f_i)$.

Dann gilt:
$\forall (i=0,\ldots,n):\left((p_2-p_1)(t_i)=0\right)$
das heißt: das Polynom $p_2-p_1\in\Pi_n$ besitzt die $n+1$ Nullstellen $t_i,(i=0,\ldots,n)$. Ein Polynom vom Grad höchstens n mit mindesten $n+1$ Nullstellen muss aber das Nullpolynom sein, das heißt: $p_2-p_1=0$ bzw. $p_1=p_2$.

Zur Klausur:
- Es dürfen keine programmierbaren Taschenrechner benutzt werden.
- Es darf jegliche Literatur auf Papier (incl. Skripte und Übungsaufgaben) in unbeschränkter Menge mitgebracht werden.

Datum: 11.12.2002

Interpolations-Problem

- Gegeben sei
  - Punkte des Graphen: $\forall (i=0,\ldots,n):\left((t_i,f_i)\right)$
  - beliebige Funktionen: $\forall (j=0,\ldots,n):\left(\phi_j:\mathbb{R}\to\mathbb{R}\right)$.
- Gesucht sei x_j mit $\sum_{j=0}^{n}\left(x_j\cdot\phi_j(t_i)=f_i\right)$
  - Skalierungsfaktoren (für jede der beliebigen Funktion einen), der das „Gesamtbild der Funktionen“ (die Summe der skalierten Funktionen) so hinbekommt, dass sie durch die Punkte des Graphen gehen.

Dieses Problem ist äquivalent mit dem Problem:
$A\cdot x=b$
$A=(a_{ij})$, $a_{ij}=\phi_j(t_i)$
$b=(b_i)$, $b_i=f_i$

Polynominterpolation:
Gesucht sei ein Polynom $p\in\Pi_n$ mit:
- $p(t_i)=f_i$ [Der Graph des Polynoms geht durch die bekannten Punkte]
- $p(t)=\sum_{j=0}^{n}\left(f_j\cdot l_j(t)\right)$ [An jeder Stützststelle werde der Funktionswert nur für diese Stützstelle ausgewählt. Zwischen den Stützstellen gelte die „Selektivität“ des Lagrange-Polynoms für jede Stützstelle.]
  - wobei $l_j(t)=\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right)$ [Lagrange-Polynom, siehe oben]

Um (3.4) in die Form (3.1) zu bringen, müssen wir in Π_n eine Basis auswählen. Von dieser Wahl wird der Aufwand, die Matrix a zu berechnen und das zugehörige Gleichungssystem zu lösen, abhängen. Darüber hinaus ist es wichtig, die zugehörige Darstellung (3.2) des

Interpolationspolynoms effizient auswerten zu können.

Beispiel 3.2

Mögliche Basen sind z.B.:

1. Lagrange-Basis:
   $\phi_j(t)=l_j(t)$, $A=(l_j(t_i))=(\delta_{ij})=I=$Einheitsmatrix
2. Monom-Basis:
   $\phi_j(t)=t^j$, $A=(t_i^{\,j})$ (sogenannte Vandermonde-Matrix)
3. Newton-Basis:
   $\phi_j(t)=w_j(t)=\prod_{k=0}^{j-1}(t-t_k)$, $A=(w_j(t_i))=\left(\prod_{k=0}^{j-1}(t_i-t_k)\right)$
   (Dies ist eine untere Dreiecksmatrix wegen $\forall(i<j):(w_j(t_i)=0)$, weil in diesen Fällen der Faktor $t_i-t_i=0$ im Produkt auftritt.)

Lemma 3.3

Gegeben seien $\forall(i=0,\ldots,n):(t_i, f_i\in\mathbb{R})$ mit $\forall(i\neq j):(t_i\neq t_j)$.
Bezeichnet $\forall(l\in[0,\ldots,n]):\forall(k\in[0,\ldots,n-l]):(P(t,(t_k,\ldots,t_{k+l}))\in\Pi_l)$, das eindeutig bestimmte Interpolationspolynom zu den Knoten $t_k,\ldots,t_{k+l}$ [l ist die Länge der Teilliste der Stützstellen, k ist die Nummer der ersten betrachteten Stützstelle], so gilt:

$$P(t,(t_k,\ldots,t_{k+l}))=\frac{(t_k-t)\cdot P(t,(t_{k+1},\ldots,t_{k+l}))-(t_{k+l}-t)\cdot P(t,(t_k,\ldots,t_{k+l-1}))}{t_k-t_{k+l}} \tag{3.5}$$

Beweis

Bezeichnet $Q(t)$ die rechte Seite von (3.5), so gilt $Q\in\Pi_l$ wegen $P(t,(t_{k+1},\ldots,t_{k+l})), P(t,(t_k,\ldots,t_{k+l-1}))\in\Pi_{l-1}$.
Für $i=k+1,\ldots,k+l-1$ (also die Mitte der Teilliste der Stützstellen) ist

$$Q(t_i)=\frac{(t_k-t_i)\cdot f_i-(t_{k+l}-t_i)\cdot f_i}{t_k-t_{k+l}}=f_i$$

außerdem

$$Q(t_k)=\frac{-(t_{k+l}-t_k)\cdot f_k}{t_k-t_{k+l}}=f_k \qquad \text{(der Anfang der Teilliste)}$$

$$Q(t_{k+l})=\frac{(t_k-t_{k+l})\cdot f_{k+l}}{t_k-t_{k+l}}=f_{k+l} \qquad \text{(das Ende der Teilliste)}$$

also

$$Q(t)=P(t,(t_k,\ldots,t_{k+l})) \qquad \text{(für die gesamte Teilliste)}$$

Algorithmus 3.4: Verfahren von Aitken|Neville

Schreibt man (3.5) in der Form

$$P(t,(t_k,\ldots,t_{k+l})) = P(t,(t_k,\ldots,t_{k+l-1}))-\frac{t-t_k}{t_{k+l}-t_k}\cdot(P(t,(t_k,\ldots,t_{k+l-1})-P(t,(t_{k+1},\ldots,t_{k+l})))) \tag{3.6}$$

so ergibt sich der folgende Algorithmus zur Berechnung von $P(t,(t_0,\ldots,t_n))$ für gegebene $t\in\mathbb{R}$:

(3.7)

Setze $\forall(k=0,\ldots,n):\left(P_{k,0}=f_k\right)$
Schleife l=1,...,n:
Für $k=0,\ldots,n-l$:   Setze $P_{k,l}=P_{k,l-1}-\frac{t-t_k}{t_{k+l}-t_k}\cdot\left(P_{k,l-1}-P_{k+1,l-1}\right)$

Das Ergebnis steht dann in $P_{0,n}$. Die Berechnung kann auf einem Vektor V ausgeführt werden, indem man $P_{k,l}$ nach V_{k+l} schreibt und die innere Schleife über k rückwärts laufen lässt.

Der Aufwand in (3.7) beträgt in erster Näherung:

$$\sum_{l=1}^{n}\left(2\cdot(n-l)\right)=2\cdot\sum_{l=0}^{n-1}(l)=n^2 \quad \text{(Multiplikationen|Divisionen)}$$

Beispiel 3.5

Zu $n=3$ seien folgende Daten gegeben:

$i=$	0	1	2	3
$t_i=$	-1	0	1	2
$f_i=P_{i,0}=$	-2	-1	2	19

Für $t=-2$ liefert Algorithmus 3.4

$$P_{0,1}=P_{0,0}-\frac{t-t_0}{t_1-t_0}\cdot\left(P_{0,0}-P_{1,0}\right)=(-2)-\frac{(-2)-(-1)}{0-(-1)}\cdot\left((-2)-(-1)\right)=-3$$

entsprechend $P_{1,1}=-7$, $P_{2,1}=-49$. Weiterhin ist

$$P_{0,2}=P_{0,1}-\frac{t-t_0}{t_2-t_0}\cdot\left(P_{0,1}-P_{1,1}\right)=(-3)-\frac{(-2)-(-1)}{1-(-1)}\cdot\left((-3)-(-7)\right)=-1$$

entsprechend $P_{1,2}=35$, sowie

$$P_{0,3}=P_{0,2}-\frac{t-t_0}{t_3-t_0}\cdot\left(P_{0,2}-P_{1,2}\right)=(-1)-\frac{(-2)-(-1)}{2-(-1)}\cdot\left((-1)-35\right)=-13$$

Will man das Interpolationspolynom öfters auswerten, so verwendet man die Newton-Basis.

Definition 3.6 (dividierte Differenzen)

Die sogenannte dividierten Defferenzen sind rekursiv definiert durch

(3.8)
a. $\forall(k=0,\ldots,n):\left((\text{dividierte Differenz an der Stelle } t_k \text{ der Funktion } f)=f[t_k]=f_k\right)$

b. $\forall(l=1,\ldots,n):\forall(k=0,\ldots,n-l):\left(f[t_k,\ldots,t_{k+l}]=\frac{f[t_k,\ldots,t_{k+l-1}]-f[t_{k+1},\ldots,t_{k+l}]}{t_k-t_{k+l}}\right)$

Lemma 3.7

Für $k=0,\dots,n-l$, $l=1,\dots,n$ gilt:

$$P(t,(t_k,\dots,t_{k+l}))=$$
$$f[t_k]+f[t_k,t_k+1]\cdot(t-t_k)+f[t_k,t_{k+1},t_{k+2}]\cdot(t-t_k)\cdot(t-t_{k+1})+\dots \tag{3.9}$$
$$+f[t_k,\dots,t_{k+l}]\cdot\left((t-t_k)\cdot\dots\cdot(t-t_{k+l-1})\right)$$

Beweis:

Wegen

$$P(t,(t_k,\dots,t_{k+l}))=P(t,(t_k,\dots,t_{k+l-1}))+f[t_k,\dots,t_{k+l}]\cdot\left((t-t_k)\cdot\dots\cdot(t-t_{k+l-1})\right)$$

genügt es zu zeigen, dass $f[t_k,\dots,t_{k+l}]$ der Koeffizient der höchsten Potenz von t (nämlich t^l) ist. Die Behauptung ist trivial für $l=0$. Sei nun $l\geqslant 1$. Nach Induktionsvoraussetzung ist $f[t_{k+1},\dots,t_{k+l}]$ der Keffizient der höchsten Potenz von t in $P(t,(t_{k+1},\dots,t_{k+l}))$, entsprechend $f[t_k,\dots,t_{k+l-1}]$ der in $P(t,(t_k,\dots,t_{k+l-1}))$. Wegen (3.5) ist dann der höchste Koeffizient von t in $P(t,(t_{k+1},\dots,t_{k+l}))$ gegeben durch

$$\frac{-f[t_{k+1},\dots,t_{k+l}]-f[t_k,\dots,t_{k+l-1}]}{t_k-t_{k+l}}=f[t_k,\dots,t_{k+l}]$$

Letzteres wegen (3.8).b.

[...nicht besucht..., kopiert von Matthias Quasthoff]

Algorithmus 3.8

Der folgende Algorithmus berechnet die dividierten Differenzen für $P(t,(t_0,\ldots,t_n))$:

(3.10)

Setze $\forall(k=0,\ldots,n):\left(D_{k,0}=f_k\right)$
Schleife l=1,...,n:
$\forall(k=0,\ldots,n-l)$:
Setze $D_{k,l}=\frac{D_{k,l-1}-D_{k+1,l}}{t_k-t_{k+1}}$

Benötigt werden die Werte $\forall(l=0,\ldots,n):\left(D_{0,l}=f\left[t_0,\ldots,t_l\right]\right)$. Entsprechend Algorithmus 3.4 kann man $D_{k,l}$ nach V_{k+l} speichern, um mit einem Vektor auszukommen. Dabei ist die innere Schleife k rückwärts zu durchlaufen.

Der Aufwand in (3.10) beträgt in erster Näherung

$\sum_{l=1}^{n}(n-1)\doteq\frac{1}{2}\cdot n^2$ Divisionen.

Algorithmus 3.9 (Horner-Schema)

Naives Vorgehen bei der Auswertung eines Polynoms

(3.11) $p(t)=a_0+a_1\cdot t+a_2\cdot t^2+\ldots+a_{n-1}\cdot t^{n-1}+a_n\cdot t^n$

in der Darstellung bezüglich der Monom-Basis führt auf den Algorithmus

(3.12)

Setze $s=a_0$, $r=1$
Schleife $i=1,\ldots,n$:
Setze $r=r\cdot t$, $s=s+a_i\cdot r$
Setze $p(t)=s$

wobei $2\cdot n$ Multiplikationen benötigt werden. Schreibt man (3.11) stattdessen in der Form

(3.13) $p(t)=a_0+t\cdot(a_1+t\cdot(a_2+\ldots+t\cdot(a_{n-1}+t\cdot a_n)\ldots))$

so erhält man das folgende Horner-Schema

(3.14)

Setze $s=a_n$
Schleife $i=1,\ldots,n$:
Setze $s=s\cdot t+a_{n-i}$
Setze $p(t)=s$

welches nur noch n Multiplikationen benötigt.

[Beispiel

Sei $p(t)=-\frac{11}{2}\cdot t^2+\frac{21}{2}\cdot t+3=a_2\cdot t^2+a_1\cdot t^1+a_0\cdot t^0$. Wenn man nun $p(t)$ für $t=-1$ auswerten möchte, dann ist

- $s_2=-\frac{11}{2}$
- $s_1=s_2\cdot t-a_1=-\frac{11}{2}\cdot(-1)-\left(-\frac{21}{2}\right)=\frac{11}{2}+\frac{21}{2}=\frac{32}{2}=16$
-

Entsprechendes Vorgehen für

(3.15) $p(t)=V_0\cdot(1)+V_1\cdot\left((t-t_0)\right)+V_2\cdot\left((t-t_0)\cdot(t-t_1)\right)+\ldots+V_n\cdot\left((t-t_0)\cdot(t-t_1)\cdot\ldots\cdot(t-t_{n-1})\right)$

liefert das modifizierte Horner-Schema

(3.16)

Setze $s=V_n$
Schleife $i=1,\ldots,n$:
Setze $s=s\cdot(t-t_{n-i})+V_{n-i}$
Setze $p(t)=s$

Algorithmus 3.10

Für die Daten aus Beispiel 3.5 erhält man mit Algorithmus 3.8, ausgehend von $\forall(i):\left(D_{i,0}=f_i\right)$:

$D_{0,1}=\frac{D_{0,0}-D_{1,0}}{t_0-t_1}=\frac{(-2)-(-1)}{(-1)-0}=\frac{-1}{-1}=1$ sowie entsprechend $D_{1,1}=3$, $D_{2,1}=17$.

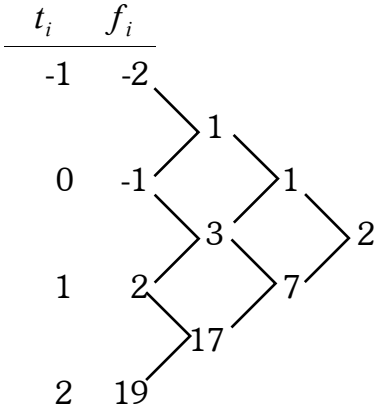
Weiterhin ist

$D_{0,2}=\frac{D_{0,1}-D_{1,1}}{t_0-t_2}=\frac{1-3}{(-1)-1}=1$ sowie entsprechend $D_{1,2}=7$

sowie

$D_{0,3}=\frac{D_{0,2}-D_{1,2}}{t_0-t_3}=\frac{1-7}{(-1)-2}=2$

Zur Vereinfachung der Handrechnung kann man auch das Schema



verwenden. Für $t=-2$ liefert das modifizierte Horner-Schema dann

$s=V_3=2$
$i=1:\quad s=s\cdot(t-t_2)+V_2=2\cdot(-2-1)+1=-5$
$i=2:\quad s=s\cdot(t-t_1)+V_1=-5\cdot(-2-0)+1=11$
$i=3:\quad s=s\cdot(t-t_1)+V_0=11\cdot(-2-(-1))+(-2)=-13$

beziehungsweise als Rechenschema

2	1	0	-1	t_i
-4	-3	-2	-1	$t-t_i$
2	1	1	-2	V_i

	-6	10	-11
2	-5	11	-13

Es bleibt die Frage, wie gut ein Interpolationspolynom festgelegt durch
(3.17) $\forall(i=0,\dots,n):\left(p(t_i)=f(t_i),\ p\in\Pi_n\right)$
eine gegebene Funktion f approximiert.

Satz 3.11

Sei $\bar{t}\in\mathbb{R}$ und $f:I\to\mathbb{R}$ $n+1$-mal stetig differenzierbar, wobei I das kleinste Interval mit $\bar{t},t_0,\dots,t_n\in I$ sei. Dann gibt es ein $\tau\in I$ mit

(3.18) $$f(\bar{t})-p(\bar{t})=\frac{f^{n+1}(\tau)}{(n+1)!}\cdot w(\bar{t}),\quad w(t)=\prod_{k=0}^{n}(t-t_k)$$

Beweis

Wegen $\forall(i=0,\dots,n):\left(\left(p(t_i)=f(t_i)\right)\wedge\left(w(t_i)=0\right)\right)$ ist die Behauptung richtig für die Wahl $\forall(i=0,\dots,n):\left(\bar{t}=t_i\right)$. Sei also im folgenden $\forall(i=0,\dots,n):\left(\bar{t}\neq t_i\right)$. Dann ist $w(\bar{t})\neq 0$ und es gibt ein $C\in\mathbb{R}$ mit

$$f(\bar{t})-p(\bar{t})-C\cdot w(\bar{t})=0$$

Damit besitzt die Funktion $g=f-p-C\cdot w$ in I die $n+2$ Nullstellen $\bar{t},t_0,\dots,t_n$. Nach dem Satz von Rolle besitzt $\dot{g}$ dann mindestens $n+1$ Nullstellen in I, entsprechend $\ddot{g}$ mindestens n Nullstellen usw. Also besitzt $\overset{(n+1)\cdot}{g}$ (die $n+1$ te Ableitung von g) mindestens eine Nullstelle $\tau\in I$. Wegen $p(n+1)=0$ gilt dafür

$$\overset{(n+1)\cdot}{g}(\tau)=\overset{(n+1)\cdot}{f}(\tau)-C\cdot(n+1)!=0$$

das heißt

$$C=\frac{\overset{(n+1)\cdot}{f}(\tau)}{(n+1)!}$$

Bemerkung 3.12

Für größere n besitzen Lagrange-Polynome l_j zwischen den Stützstellen t_i zum Teil Funktionswerte von sehr großem Betrag. Damit kann der Interpolationsfehler (3.18) sehr groß werden.

[Berechnung von Lagrange-Polynomen

$$\tilde{x}=L(x)=f_0\cdot L_0(x)+f_1\cdot L_1(x)+f_2\cdot L_2(x)+\ldots+f_n\cdot L_n(x)=\sum_{i=0}^{n}\left(f_i\cdot L_i(x)\right) \text{ wobei}$$

$$L_i(x)=\frac{\prod_{\substack{a=0\\a\neq i}}^{n}(x-x_a)}{\prod_{\substack{a=0\\a\neq i}}^{n}(x_i-x_a)}=\prod_{\substack{a=0\\a\neq i}}^{n}\left(\frac{x-x_a}{x_i-x_a}\right) \text{ (Lagrange-Polynom)}$$

Beispiel

Für gegebene x_0, x_1, x_2 ist:

$$L_0(x)=\frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2},\ L_1(x)=\frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2},\ L_2(x)=\frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1}$$

Seien nun beispielsweise $x_0=0$, $x_1=1$, $x_2=2$, $y_0=3$, $y_1=8$, $y_2=2$. Dann ist

$$L_0(x)=\frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2}=\frac{x-1}{0-1}\cdot\frac{x-2}{0-2}=\frac{1}{2}\cdot(x-1)\cdot(x-2)$$

$$L_1(x)=\frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2}=\frac{x-0}{1-0}\cdot\frac{x-2}{1-2}=-x\cdot(x-2)$$

$$L_2(x)=\frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1}=\frac{x-0}{2-0}\cdot\frac{x-1}{2-1}=\frac{1}{2}\cdot x\cdot(x-1)$$

Damit ist

$$\begin{aligned}\tilde{x}=L(x)&=f_0\cdot L_0(x)+f_1\cdot L_1(x)+f_2\cdot L_2(x)\\&=3\cdot\frac{1}{2}\cdot(x-1)\cdot(x-2)+8\cdot(-x\cdot(x-2))+2\cdot\frac{1}{2}\cdot x\cdot(x-1)\\&=\frac{3}{2}\cdot(x-1)\cdot(x-2)-8\cdot x\cdot(x-2)+x\cdot(x-1)\\&=\frac{3}{2}\cdot(x^2-2\cdot x-x+2)-8\cdot x^2+16\cdot x+x^2-x\\&=\frac{3}{2}\cdot x^2-3\cdot x-\frac{3}{2}x+3-8\cdot x^2+16\cdot x+x^2-x\\&=-5{,}5\cdot x^2+10{,}5\cdot x+3\end{aligned}$$

]

3.2 Spline-Interpolation

Um Effekte der Polynom-Interpolation bei größerem Grad n wie in Bemerkung 3.12 angedeutet zu vermeiden, kann man versuchen, Polynome geringeren Grades stückweise zu hinreichend glatten Funktionen zusammenfügen. Man nennt solche Funktionen Splines.

Definition 3.12 (kubische Splines)

Gegeben sei eine Unterteilung

(3.19) $a=t_0<t_1<\ldots<t_n=b$

des Intervalls $[a,b]$. Eine Funktion $s:[a,b]\to\mathbb{R}$ heißt (zu (3.19) gehöriger) kubischer Spline, wenn $s\in C^2([a,b],\mathbb{R})$ und $s|[t_i,t_{i+1}]$ (Die Funktion s mit auf $[t_i,t_{i+1}]$ eingeschränkten Definitionsbereich) für jedes $i=0,\ldots,n-1$ auf $[t_i,t_{i+1}]$ mit e inem kubischen Polynom übereinstimmt.

Satz 3.14

Gegeben seien $t_i,f_i\in\mathbb{R}$, $i=0,\ldots,n$ mit $\forall(i\neq j):(t_i\neq t_j)$.
Fordert manneben den Interpolationsbedingungen

(3.20) $\forall(i=0,\ldots,n):(s(t_i)=f_i)$

noch eine der drei Bedingungen

(3.21)
a. $\ddot{s}(t_0)=0$, $\ddot{s}(t_n)=0$ (natürlicher Spline)
b. $\dot{s}(t_0)=\dot{f}_0$, $\dot{s}(t_n)=\dot{f}_n$ mit $\dot{f}_0,\dot{f}_n\in\mathbb{R}$ (eingespannter Spline)
c. $\dot{s}(t_0)=\dot{s}(t_n)$, $\ddot{s}(t_0)=\ddot{s}(t_n)$ (periodischer Spline)

so ist dadurch genau ein kubischer Spline festgelegt.

Im folgenden betrachten wir nur natürliche Splines, das heißt (3.20) mit (3.21.a).
Zur Konstruktion von s setzen wir

(3.22) $\forall(i=0,\ldots,n-1):(h_i=t_{i+1}-t_i)$

und machen auf $[t_i,t_{i+1}]$ den Ansatz

(3.23)
$$s_i(t)=a_i+b_i(t-t_i)+\frac{1}{2}\cdot c_i\cdot(t-t_i)^2+\frac{1}{6}\cdot d_i\cdot(t-t_i)^3$$
$$s_i=s|[t_i,t_{i+1}]\ (\forall(t\in[t_i,t_{i+1}]):(s_i(t)=s(t)))$$

Zu erfüllen sind die Bedingungen

(3.24)
a. $\forall(i=0,\ldots,n-1):(s_i(t_i)=f_i)$, $\forall(i=0,\ldots,n-1):(s_i(t_{i+1})=f_{i+1})$
b. $\forall(i=0,\ldots,n-2):(\dot{s}_i(t_{i+1})=\dot{s}_{i+1}(t_{i+1}))$
c. $\forall(i=0,\ldots,n-2):(\ddot{s}_i(t_{i+1})=\ddot{s}_{i+1}(t_{i+1}))$
d. $\ddot{s}_0(t_0)=0$, $\ddot{s}_{n-1}(t_n)=0$

wobei

(3.25)
a. $\dot{s}_i(t)=b_i+c_i\cdot(t-t_i)+\frac{1}{2}\cdot d_i\cdot(t-t_i)^2$
b. $\ddot{s}_i(t)=c_i+d_i\cdot(t-t_i)$

Aus (3.24.a) ergibt sich sofort

(3.26) $\forall(i=0,\ldots,n-1):(a_i=f_i)$

und aus (3.24.c)

(3.27) $\forall(i=0,\ldots,n-2):(c_{i+1}=c_i+d_i\cdot h_i)$

oder (mit $c_0=c_n=0$ entsprechend (3.24.d))

(3.28) $$\forall(i=0,\ldots,n-1):\left(d_i=\frac{c_{i+1}-c_i}{h_i}\right)$$

Damit folgt aus (3.24.a)

$$\forall (i=0,\ldots,n-1):$$
$$b_i=\frac{f_{i+1}-f_i}{h_i}-\frac{1}{2}\cdot c_i\cdot h_i-\frac{1}{6}\cdot\frac{c_{i+1}-c_i}{h_i}\cdot h_i^2 \tag{3.29}$$
$$=\frac{f_{i+1}-f_i}{h_i}-\frac{2\cdot c_i+c_{i+1}}{6}\cdot h_i$$

und alle Größen sind in den Unbekannten c_i ausgedrückt.
Aus (3.24.b) ergibt sich nun

$$\forall (i=1,\ldots,n-1):\left(b_i=b_{i-1}+c_{i-1}\cdot h_{i-1}+\frac{1}{2}\cdot d_{i-1}\cdot h_{i-1}^2\right) \tag{3.30}$$

beziehungsweise

$$\frac{f_{i+1}-f_i}{h_1}-\frac{2\cdot c_i+c_{i+1}}{6}\cdot h_i=\frac{f_i-f_{i-1}}{h_{i-1}}-\frac{2\cdot c_{i-1}-c_i}{6}\cdot h_{i-1}+c_{i-1}\cdot h_{i-1}+\frac{1}{6}\cdot\frac{c_i-c_{i-1}}{h_{i-1}}\cdot h_{i-1}^2 \tag{3.31}$$

oder

$$\forall (i=1,\ldots,n-1):\left(\frac{h_{i-1}}{6}\cdot c_{i-1}+\frac{h_{i-1}+h_i}{3}\cdot c_i+\frac{h_i}{6}\cdot c_{i+1}=\frac{f_{i+1}-f_i}{h_i}-\frac{f_i-f_{i-1}}{h_{i-1}}\right) \tag{3.32}$$

Zusammen mit $c_0=c_n=0$ erhält man also für die Berechnung der $\forall (i=1,\ldots,n-1):(c_i)$ ein lineares Gleichungssystem mit symmetrischer tridiagonaler Koeffizientenmatrix. Aus der Größe der Einträge kann man schließen, dass diese positiv-definit und damit nichtsingulär ist.
Damit erhält man aus (3.32) eine eindeutige Lösung $\forall (i=1,\ldots,n-1):(c_i)$. Die Beziehungen (3.26), (3.28) und (3.29) liefern dann eine vollständige Darstellung (3.23) des gesuchten Splines s.

4 Differentiation

Zu gegebenem $f\in C^1([a,b],\mathbb{R})$ und $\bar{t}\in[a,b]$ ist der Wert $\dot{f}(\bar{t})$ zu bestimmen. Darf man als Daten nur Funktionswerte von f verwenden, so ist dieses Problem schlecht gestellt.

Beispiel 4.1

Sei $f:\mathbb{R}\to\mathbb{R}$ gegeben durch
$f(t)=\tanh(c\cdot t)$ [Grafik: Skizze des Graghen von tanh(t)]
mit $c>0$. Wegen
$$\dot{f}(t)=\frac{c}{(\cosh(c\cdot t))^2}$$
gilt für $\bar{t}=0$:
$$\dot{f}(\bar{t})=c$$
Das heißt: obwohl
$$\forall (t\in\mathbb{R}):(|f(t)|<1)$$
gilt, kann $|\dot{f}(\bar{t})|$ beliebig groß werden.
Trotz der Schlechtgestelltheit des Problems werden in manchen Situationen numerische Verfahren zur Differentiation verwendet. Man muss dann besonders auf die Auswirkung von Rundungsfehlern achten.

4.1 Differenzenverfahren

Die Idee bei dem Differenzenverfahren ist, die gegebene Funktion durch ein Interpolationspolynom p zu ersetzen und $\dot{p}(\bar{t})$ als Approxmation an $\dot{f}(\bar{t})$ zu verwenden.

(Dieser Vorgang heißt Diskretisierung, da eine Funktion, die womöglich nur durch unendlich viele Koeffizienten beschrieben werden kann, umgewandelt wird in eine Funktion (nämlich ein Polynom), die mit e ndlich vielen Koeffizienten auskommt. Der Fehler, der durch die Annäherung durch ein Polynom an eine beliebige Funktion entsteht, heißt Diskretisierungsfehler.).

Da man dabe ein Element eines unendlichdimensnalen Vektorraums durch ein Element eines endlichdimensionalen Vektorraums ersetzt, spricht man von Diskretisierung.

Interpoliert man f in den Stellen $\forall(0,\ldots,n):\left(t_i\right)$, so erhält man das Polynom

$$p(t)=\sum_{j=0}^{n}\left(f_j\cdot l_j(t)\right),\quad l_j(t)=\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right) \tag{4.1}$$

mit $f_j=f(t_j)$. Es gilt dann

$$\dot{p}(t)=\sum_{j=0}^{n}\left(f_j\cdot \dot{l}_j(t)\right),\quad \dot{l}_j(t)=\sum_{\substack{l=0\\l\neq j}}^{n}\left(\frac{1}{t_j-t_l}\cdot\prod_{\substack{k=0\\k\neq j\\k\neq l}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right)\right) \tag{4.2}$$

Typischerweise wählt man das Gitter äquidistant, das heißt: $\forall(i=0,\ldots,n):\left(t_i=t_0+i\cdot h\right)$ mit einer Schrittweite $h>0$ und so, dass $\bar{t}$ ein Gitterpunkt ist.

Beispiel 4.2

Für $n=1$ und $t_0=\bar{t}$, $t_1=\bar{t}+h$ erhält man aus

$$\dot{p}(t_0)=f_0\cdot\frac{1}{t_0-t_1}+f_1\cdot\frac{1}{t_1-t_0}=\frac{f_1-f_0}{h}$$

die Approximation

$$\dot{p}(\bar{t})\approx\frac{f(\bar{t}+h)-f(\bar{t})}{h} \tag{4.3}$$

das heißt, man hat gerade die Grenzwertbildung bei der Definition der Ableitung rückgängig gemacht.

[...zu spät gekommen, von Dominic abgeschrieben...]
Für $n=2$ und $t_0=\bar{t}-h$, $t_1=\bar{t}$, $t_2=\bar{t}+h$ erhält man aus

$$\dot{p}(t_1)=f_0\cdot\left(\frac{1}{t_0-t_1}\cdot\frac{t_1-t_2}{t_0-t_2}+\frac{1}{t_0-t_2}\cdot\frac{t_1-t_1}{t_0-t_1}\right)+f_1\cdot\left(\frac{1}{t_1-t_0}\cdot\frac{t_1-t_2}{t_1-t_2}+\frac{1}{t_1-t_2}\cdot\frac{t_1-t_0}{t_1-t_0}\right)$$
$$+f_2\cdot\left(\frac{1}{t_2-t_0}\cdot\frac{t_1-t_1}{t_2-t_1}+\frac{1}{t_2-t_1}\cdot\frac{t_1-t_0}{t_2-t_0}\right)$$
$$=f_0\cdot\left(\frac{-1}{2\cdot h}\right)+f_1\cdot\left(\frac{1}{h}-\frac{1}{h}\right)+f_2\cdot\left(\frac{1}{2\cdot h}\right)$$

Beweis

Wegen $\forall(i=0,\ldots,n):\left(\left(p(t_i)=f(t_i)\right)\wedge\left(w(t_i)=0\right)\right)$ ist die Behauptung richtig für die Wahl $\forall(i=0,\ldots,n):\left(\bar{t}=t_i\right)$. Sei also im folgenden $\forall(i=0,\ldots,n):\left(\bar{t}\neq t_i\right)$. Dann ist $w(\bar{t})\neq 0$ und es gibt ein $C\in\mathbb{R}$ mit

$$f(\bar{t})-p(\bar{t})-C\cdot w(\bar{t})=0$$

Damit besitzt die Funktion $g=f-p-C\cdot w$ in I die $n+2$ Nullstellen $\bar{t},t_0,\ldots,t_n$.
Nach dem Satz von Rolle besitzt $\dot{g}$ dann mindestens $n+1$ Nullstellen in I, entsprechend $\ddot{g}$ mindestens n Nullstellen usw. Also besitzt $g^{(n+1)\cdot}$ [n+1 te Ableitung von g] mindestens eine Nullstelle
[...]

Für $h\to 0$ nehmen also die Rundungsfehler z. Man darf demnach h weder zu groß noch zu klein wählen. Eine mögliche Wahl für (4.3) ist

(4.6) $$h=\sqrt{eps}\cdot|f(\bar{t})|$$

Gilt $\dot{f}(\bar{t})\approx 1$, so unterscheiden sich $f(\bar{t})$ und $f(\bar{t}+h)$ in etwa auf der zweiten Hälfte der Mantisse. Die erste Hälfte wird durch Auslöschung zugunsten des Diskretisierungsfehlers geopfert.

4.2 Symbolisches|automatisches Differenzieren

Jede auf einem Rechner implementierte Funktion f besteht aus einer endlichen Komposition von Elementaroperationen. Die Ableitung einer Elementaroperation setzt sich aber wieder aus einer endlichen Zahl von Elementaroperation zusammen. Man kann also mit einem Programm zur Berechnung von $f(x)$ jede auftretende Variable als Funktion von x auffassen. Führt man für jede Variable eine neue Variable für den Ableitungswert ein, so erhält man ein Programm zur Berechnung von $f'(x)$, indem man vor jede Zuweisung eine entsprechende abgeleitete Zuweisung setzt, sogenanntes symbolisches Differenzieren.

In einer Programmiersprache, in der man wie etwa in C++ Operatoren überladen kann, kann man auch einen Variablentyp definieren, der aus dem ursprünglichen Wert und dem Ableitungswert besteht, und dazu alle Elementaroperationen derart überladen, dass neben den üblichen Berechnungen mit Hilfe der Ableitungsregeln der Wert der Ableitung berechnet wird, sogenanntes automatisches Differenzieren.

5. Integration

Sei $a<b$ und $f\in X=C([a,b],\mathbb{R})$. Zu bestimmen ist

(5.1) $$I(f)=\int_a^b (f(t)\,dt)$$

das heißt: wir betrachten die Ableitung

(5.2) $$I: X\to X,\ f\mapsto \int_a^b (f(t)\,dt)$$

Für diese gilt

(5.2) $$\begin{aligned}|I(f_2)-I(f_1)| &= \left|\int_a^b ((f_2(t)-f_1(t))\,dt)\right| \leq \int_a^b (|f_2(t)-f_1(t)|\,dt) \\ &\leq (b-a)\cdot \max_{t\in[a,b]} (|f_2(t)-f_1(t)|) = (b-a)\cdot \|f_2-f_1\|_X\end{aligned}$$

das heißt: die Kondition (bezüglich des absoluten Fehlers) wird beschrieben durch $\kappa=b-a$.

5.1 Newton-Cotes-Formeln

Wie bei der Differentiation kann das obige Problem mittels Polynom-Interpolation diskretisiert werden.
Ausgehend von

(5.4) $$a\leq t_0<t_1<\ldots<t_n\leq b$$

und Daten $\forall(i=0,\ldots,n):(t_i,f_i)$ mit $f_i=f(t_i)$ erhält man ein Interpolationspolynom $p\in\Pi_n$.
Integration liefert

(5.5) $$\int_a^b (p(t)\,dt)=\int_a^b \left(\sum_{j=0}^n (f_j\cdot l_j(t))\,dt\right)=\sum_{j=0}^n \left(f_j\cdot\int_a^b (l_j(t)\,dt)\right)=\sum_{j=0}^n (b_j\cdot f_j)$$

mit sogenannten Gewichten

(5.6) $$\forall(j=0,\ldots,n):\left(b_j=\sum_a^b (l_j(t)\,dt)\right)$$

Damit hat man eine Näherungsformel der Form

(5.7) $$I(f)\approx\sum_{j=0}^n (b_j\cdot f(t_j))$$

für das Integral (5.1) erhalten. Man nennt Näherungsformeln zur Integration auch Quadraturformeln.

Bemerkung 5.1

Per Konstruktion sind die Quadraturformeln (5.7) mit (5.6) exakt für alle Polynome $p\in\Pi_n$, das heißt:

(5.8) $$\forall(p\in\Pi_n):\left(I(p)=\sum_{j=0}^n (b_j\cdot p(t_j))\right)$$

Wählt man als Basis in Π_n die Monom-Basis, so erfüllen die b_j die Beziehung

(5.9) $$\forall(i=0,\ldots,n):\left(\sum_{j=0}^n (b_j\cdot t_j^{\,i})=\int_a^b (t^i\,dt)=\frac{1}{i+1}\cdot(b^{i+1}-a^{i+1})\right)$$

Die b_j genügen also einem linearen Gleichungssystem mit einer nichtsingulären Vandermonde-Matrix als Koeffizientenmatrix.

Newton|Cotes-Formeln

(5.4) $a=t_0<t_1<\ldots<t_n\leqslant b\,,\ f_i=f(t_i)$

(5.5|5.7) $I(f)=\int_a^b (f(t)\,dt)\approx\int_a^b (p(t)\,dt)=\sum_{j=0}^{n}(b_i\cdot f_j)$

(5.6) $b_j=\int_a^b (l_j(t)\,dt),\ l_j(t)=\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right)$

(5.9) $\forall(i=0,\ldots,n):\left(\sum_{j=0}^{n}(b_j\cdot t_j^{\,i})=\frac{1}{i+1}\cdot(b^{i+1}-a^{i+1})\right)$

Definition 5.2

Ist eine Quadraturformel exakt für alle $p\in\Pi_m$, aber nicht exakt für ein $p\in\Pi_{m+1}$, so heißt $n+1$ die Ordnung der Quadraturformel.

Nach Bemerkung 5.1 haben also die Quadraturformeln (5.7) mit (5.6) mindestens die Ordnung $n+1$.

Satz 5.3

Sei $f\in C^{n+1}([a,b],\mathbb{R})$. (Sei f eine mindestens n+1 mal differenzierbare Funktion.) Dann gilt für (5.7) mit (5.6)

(5.10) $\text{Diskretisierungsfehler}=|I(f)-I(p)|\leqslant C\cdot H^{n+2}$

wobei $H=b-1$ und $C\geqslant 0$ unabhängig von H

Beweis

Mit (3.18) erhält man

$$|I(f)-I(p)|=\left|\int_a^b (f(t)-p(t))\right|=\left|\int_a^b\left(\frac{f^{(n+1)}(\tau(t))}{(n+1)!}\cdot\omega(t)\,dt\right)\right|\leqslant(b-1)\cdot\frac{\max_{t\in[a,b]}\left(\left|f^{(n+1)}(t)\right|\right)}{(n+1)!}=C\cdot H^{n+2}$$

Sei das Gitter im folgenden äquidistant gemäß

(5.11) $\forall(i=0,\ldots,n):\left(t_i=a+i\cdot h\ \text{mit}\ h=\frac{b-a}{n}\right)$

Dann gilt mit $t=a+s\cdot h$:

(5.12) $b_j=\int_a^b\left(\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{t-t_k}{t_j-t_k}\right)dt\right)=h\cdot\int_0^n\left(\prod_{\substack{k=0\\k\neq j}}^{n}\left(\frac{s-k}{j-k}\right)ds\right)=h\cdot\omega_j$

Die so erhaltenen Quadraturformeln heißen Newton|Cotes-Formeln. Diese sind symmetrisch im folgenden Sinn:
vergleiche Übungen

Definition 5.4

Eine Quadraturformel (5.7) heißt symmetrisch, wenn gilt

(5.13) $\forall (j=0,\ldots,n): \left(t_{n-j}=(a+b)-t_j,\ b_{n-j}=b_j\right)$

Lemma 5.5

Gegeben sei eine symmetrische Quadraturformel (5.6) mit mindestens ungerader Ordnung $m+1$. Dann hat diese mindestens die Ordnung $m+2$

Beweis

Nach Voraussetzung ist die Quadraturformel exakt für alle Polynome $p\in\Pi_m$.
Setzt man speziell

$$p(t)=\left(t-\frac{a+b}{2}\right)^{m+1}$$

so gilt

$$\int_a^b \left(p(t)\,dt\right)\approx 0$$

da p bezüglich der Intervallmitte $\frac{a+b}{2}$ punktsymmetrisch ist.

Auf der anderen Seite gilt

$$\sum_{j=0}^{n}\left(b_j\cdot\left(t_j-\frac{a+b}{2}\right)^{m+1}\right)=\frac{1}{2}\cdot\sum_{j=0}^{n}\left(b_j\cdot\left(t_j-\frac{a+b}{2}\right)^{m+1}+b_{n-j}\left(t_{n-j}-\frac{a+b}{2}\right)^{m+1}\right)=0$$

Also wird dieses spezielle $p\in\Pi_{m+1}$ vom exakten Grad $m+1$ ebenfalls von der Quadraturformel exakt integriert. Wegen der Linearität von Integral und Quadraturformel ist diese dann für alle $p\in\Pi_{m+1}$ exakt.

Damit ist gezeigt, dass symmetrische Quadraturformeln immere eine gerade Ordnung besitzen.

Beispiel 5.6: Newton|Cotes-Formeln

Für $n=1$ gilt

$$\sigma_0=-\int_0^1\left((s-1)\,ds\right)=-\left(\frac{s^2}{2}-s\right)\Big|_0^1=\frac{1}{2},\ \sigma_1=\int_0^1\left(s\,ds\right)=\frac{1}{2}$$

und man erhält die sogenannte **Trapezregel**

(5.14) $$I(f)\approx\frac{b-a}{2}\cdot\left(f(a)+f(b)\right)$$

mit der Ordnung 2.

Für $n=2$ gilt:

$$\sigma_0=\int_0^2\left(\frac{s-1}{0-1}\cdot\frac{s-2}{0-2}ds\right)=\frac{1}{2}\cdot\int_0^2\left(\left(s^2-3\cdot s+2\right)ds\right)=\frac{1}{2}\cdot\left(\frac{8}{3}-\frac{12}{2}+4\right)=\frac{1}{3}$$

$$\sigma_1=\int_0^2\left(\frac{s-0}{1-0}\cdot\frac{s-2}{1-2}ds\right)=-\int_0^2\left(\left(s^2-2\cdot s\right)ds\right)=-\left(\frac{8}{3}-\frac{8}{2}\right)=\frac{4}{3}$$

$$\sigma_2=\int_0^2\left(\frac{s-0}{2-0}\cdot\frac{s-1}{2-1}ds\right)=\frac{1}{2}\int_0^2\left(\left(s^2-s\right)ds\right)=\frac{1}{2}\cdot\left(\frac{8}{3}-\frac{4}{2}\right)=\frac{1}{3}$$

und man erhält die sogenannte **Simpson-Regel**

$$I(f)\approx\frac{b-a}{6}\cdot\left(f(a)+4\cdot f\left(\frac{a+b}{2}\right)+f(b)\right) \tag{5.15}$$

mit der Ordnung 4, vergleiche Lemma 5.5

Zur Verkleinerung des Diskretisierungsfehlers kann man vorab das Intervall $[a,b]$ unterteilen und die Quadraturformel auf Teilintervalle anwenden. Man spricht von summierten Quadraturformeln.
Wählt man die Unterteilungspunkte

$$\forall(k=0,\ldots,N):\left(t_k=a+k\cdot H\right),\ H=\frac{b-a}{N} \tag{5.16}$$

so interpoliert man im Fall der Newton|Cotes-Formeln den Integranden f auf jedem Teilintervall $[t_k,t_{k+1}]$ in den Punkten

$$\forall(i=0,\ldots,n):\left(t_{ik}=t_k+i\cdot h\right),\ h=\frac{H}{n}=\frac{b-a}{N\cdot n} \tag{5.17}$$

Bezeichnet p_K das Interpolationspolynom auf $[t_k,t_{k+1}]$ und I_k die Integration über $[t_k,t_{k+1}]$, so hat die summierte Formel die Form

$$I(f)\approx\sum_{k=0}^{N-1}\left(I_k(p_k)\right) \tag{5.18}$$

Satz 5.7

Sei $f\in C^{n+1}\left([a,b],\mathbb{R}\right)$. Dann gilt für die summierten Newton|Cotes-Formeln

$$|I(f)-\text{Diskretisierungsfehler}|=\left|I(f)-\sum_{k=0}^{N-1}\left(I_k(p_k)\right)\right|\leqslant C\cdot h^{n+1} \tag{5.19}$$

mit $C\geqslant 0$ unabhängig von h.

5.2 Gitteranpassung

Um die Genauigkeit einer Quadraturformel zu erhöhen, kann man die Intervalle verkleinern, auf die man sie anwendet. Für die Effizienz (gemessen etwa durch die Anzahl der Funktionsauswertungen) und Genauigkeit (abhängig etwa von der Anzahl der zu summierenden Teilintervalle) sollte die Länge der Teilintervalle an den jeweiligen Funktionsverlauf angepasst werden. Man spricht von Gitteranpassung oder auch Schrittweitensteuerung.

Datum: 17.01.2002

Ausgehend davon, dass man Integration von a bis zu einem Punkt $t_k \in (a,b)$ bereits durchgeführt hat, möchte man ein in irgendeiner Weise passendes t_{k+1} für das nächste Teilintervall wählen. Sei dazu

(5.20) $a = t_0 < t_1 < \ldots < t_k < b$

mit

(5.21) $h_k = t_k - t_{k-1}$

und

(5.22) $I_k(f) = \int_{t_{k-1}}^{t_k} (f(t)\,dt)$

Weiter bezeichen $\tilde{I}_k$ das Ergebnis einer Quadraturformel der Ordnung q und $\hat{I}_k$ das Ergebnis einer Quadraturformel mit einer Ordnung höher als q, jeweils angewandt auf $[t_{k-1}, t_k]$. In Anlehnung an (5.10) macht man die Annahme

(5.23) $|I_k(f) - \tilde{I}_k(F)| \approx C \cdot h_k^{q+1}$

mit einer Konstanten C. Außerdem sei $\hat{I}_k$ wesentlich genauer als $\tilde{I}_k$, sodass die Annahme

(5.24) $\tilde{I}_k(f) \approx I_k(f)$

gerechtfertigt ist.

Wegen

(5.25) $error = |\hat{I}_k(f) - \hat{I}_k(f)| \approx |I_k(f) - \tilde{I}_k(f)|$

macht man die Modellannahme

(5.26) $error = C \cdot h_k^{q+1}$

Für eine vorgegebene Toleranz *toleranz* (vorgegebener maximaler erlaubter Fehler) soll die Bedingung

(5.27) $error \leqslant toleranz$

für die Genauigkeit der Integration eingehalten werden.

„Für die Effizienz wäre es natürlich wichtig, die Schrittweite möglichst groß zu wählen. Deswegen sollte die Toleranz so groß wie gerade noch annehmbar gewählt werden."

Zu *toleranz* gehört eine optimale Schritweite $\bar{a}$ gemäß

(5.28) $toleranz = C \cdot \bar{h}^{q+1}$

Elimination von C liefert

(5.29) $\frac{error}{toleranz} = \left(\frac{h_k}{\bar{h}}\right)^{q+1}$

beziehungsweise

(5.39) $\bar{h} = h_k \cdot \sqrt[q+1]{\frac{toleranz}{error}}$

Dabei ist (5.27) äquivalent zu

(5.31) $\bar{h} \geqslant h_k$

Diese Überlegungen führen zu folgender Strategie: Ist $error > toleranz$, so wird der Integrationsschritt für $[t_{k-1}, t_k]$ verworfen und mit $h_k = \bar{h}$ wiederholt. Ist $error \leqslant toleranz$, so

wird der Schritt akzeptiert.
$\tilde{I}_k(f)$ oder $\hat{I}(f)$ wird zum aktuellen Wert des Integrals addiert und der nächste Schritt wird mit $h_{k+1}=h$ durchgeführt.
Um zu viele Ablehnungen von Schritten zu vermeiden, wird (5.30) typischerweise in etwas konservativerer Form

(5.32) $$\overline{h}=reduction\cdot h_k\cdot\sqrt[q+1]{\frac{toleranz}{error+safety}}$$

verwendet. Außerdem wird die Änderung der Schrittweite eingeschränkt durch die Forderung

(5.33) $$r_{min}\leqslant\frac{\overline{h}}{h_k}\leqslant r_{max}$$

Die Wahl der Konstanten $reduction$, $safety$ sowie r_{min}, r_{max} ist von philosophischer Natur. Eine mögliche Wahl wäre

$$reduction=0.9,\ safety=eps,\ r_{min}=0.2,\ r_{max}=2.0$$

Strittig ist auch die Frage, ob man $\tilde{I}_k(f)$ oder $\hat{I}_k(f)$ zur gewünschten Approximation von I (f) aufaddieren sollte. (Für $\tilde{I}_k(f)$ hat man eine Fehlerschätzung, wobei allerdings angenommen wurde, dass $\hat{I}_k(f)$ der genauere wert ist).

Algorithmus 5.16

Gegeben sei $h_1\leqslant b-a$, $t_0=a$, sowie $toleranz>0$

(5.34)

Setze $K=1$ sowie $I=0$
Schleife über die erlaubte Anzahl von Versuchen:
Setze $t_k=t_{k-1}+h_k$
Bestimme $\overline{h}$ entsprechend (5.32) und (5.33)
Ist $error>toleranz$, so setze $h_k=\overline{h}$ und mache die Schleife weiter
Ist $error\leqslant toleranz$, so setze $h_{k+1}=\min\left(\overline{h}, b-t_k\right)$, $I=I+\hat{I}_k(f)$. • Ist $t_k=b$, dann höre auf. • Ansonsten mache die Schleife weiter mit $k=k+1$

Man beachte, dass man bei dieser Methode die Richtung (von a nach b) auszeichnet.

6. Nichtlineare Gleichungssysteme

Gesucht ist die Lösung der Gleichung
(6.1) $$f(x^*)=0$$
wobei $f:D\to R^n$ mit $D\subseteq R^n$ sei.

Ist f stetig differenzierbar und $f'(x^*)$ nichtsingulär, so besagt der Satz über die Umkehrfunktion, dass f in einer Umgebung V von $y^*=0$ eine Inverse f^{-1} besitzt. Insbesondere ist $x^*=f^{-1}(y^*)$ in $u=f^{-1}(v)$ die einzige Nullstelle von f (Wohlgestelltheit des Problems) und es gilt nach Abschnitt 1.3 für die differentielle Kondition

(6.2) $\kappa = \left\| (f^{-1})'(y^*) \right\| = \left\| \left(f'(x^*) \right)^{-1} \right\|$

6.1 Iterationsverfahren

Zur Lösung von (6.1) gibt es im allgemeine keine Lösungsformel, das heißt: keine geschlossene Darstellung der Umkehrfunktion. Stattdessen überführt man (6.1) in eine sogenannte Fixpunktform

(6.3) $x = \varphi(x)$

die zu (6.1) äquivalent ist, das heißt

(6.4) $\left(f(x^*) = 0 \right) \Leftrightarrow \left(x^* = \varphi(x^*) \right)$

Man nennt ein x^* mit $x^* = \varphi(x^*)$ auch einen Fixpunkt von φ.

Die Fixpunktform (6.3) legt dann ein iteratives Vorgehen gemäß

(6.5) $x_{\nu+1} = e(x_\nu),\ \nu \in \mathbb{N}_0,\ x_0$ gegeben

nahe mit der Hoffnung, dass dadurch eine Folge $\left(x_\nu \right)_{\nu \in \mathbb{N}_0}$ definiert wird, für die $x_\nu \to x^*$ gilt.
Ein Problem (6.1) kann auf vielfältige Weise in eine Fixpunktform (6.3) gebracht werden, von denen nicht jede zum Ziel führt.

Beispiel 6.1

Die Gleichung

$$e^x + x = 0$$

besitzt eine eindeutige Lösung x^*, die bei zehnstelliger Rechnung gegeben ist durch

$$x^* = -0.5671432904$$

(Das x^* ist nicht in geschlossener Form (durch Elementaroperationen) darstellbar.)

Ausgehend von $x_0 = -0.5$ erhält man für

$$x_{y+1} = -e^{x_y}$$

z.B.

$$x_9 = -0.5675596343$$

während für

$$x_{\nu+1} = \log(-x_\nu)$$

die Iterierte x_5 schon nicht mehr definiert ist.

> „Es ist also grundsätzlich möglich, dass die Iteration an einer bestimmten Stelle nicht weiter möglich ist. Wir sollten herausfinden, in welchen Fällen dies möglich ist."

> „Wenn aber eine Iteration möglich ist, dann konvergiert das x_ν zur gesuchten Lösung. Es entsteht also eine Folge, dessen Grenzwert die Lösung ist. Jedoch können wir auf dem Rechner nicht eine unendliche Folge ausrechnen sondern müssen uns mit einem „endlichen" Folgenglied begnügen. Dieses weicht aber möglicherweise von der Lösung ab. Diese Abweichung wird Abbrechfehler gennannt."

Man beachte, dass selbst im Fall $x_\nu \to x^*$ auf dem Rechner die Iteration irgendwann abgebrochen werden muss und das letzte berechnete x_ν als x^* akzeptiert werden muss. Der

zugehörige Fehler x^*-x_ν heißt **Abbrechfehler**.

Satz 6.2: Banachscher Fixpunktsatz

Gegeben sei $\varphi: D\to D$ mit $D\subseteq\mathbb{R}^n$ abgeschlossen. (Dies ist eine ziemlich starke Voraussetzung, dass die Funktion nicht etwa in $\mathbb{R}^n$ abbildet, sondern nur in die Teilmenge von $\mathbb{R}^n$, die auch zum Definitionsbereich gehört.)
Ist φ kontraktiv, also gilt:

(6.6) $$\forall(x_1,x_2\in D):\left(\left\|\varphi(x_2)-\varphi(x_1)\right\|\leq L\cdot\left\|x_2-x_1\right\|\right)$$

mit $L<1$ (also ist bei gegebenen Urbildern der Abstand der Bilder kleiner als der Abstand der Urbilder, konvergieren somit zwei x_1, x_2 immer mehr zueinander), so ist durch (6.5) für jedes $x_0\in D$ eine Folge $\left(x_\nu\right)_{\nu\in\mathbb{N}_0}$ definiert, die gegen den einzigen Fixpunkt x^* von φ konvergiert.

Insbesondere gilt:

(6.7)
a. $$\forall(\nu\in\mathbb{N}_0):\left(\left\|x^*-x_{\nu+1}\right\|\leq L\cdot\left\|x^*-x_\nu\right\|\right)$$
b. $$\forall(\nu\in\mathbb{N}_0):\left(\left\|x^*-x_\nu\right\|\leq\frac{L^\nu}{1-L}\cdot\left\|x_1-x_0\right\|\right)$$

und man spricht von **lineare**r **Konvergenz**.

Beweis

Wegen $\varphi(D)\subseteq D$ können die Iterierten D nicht verlassen.
Weiterhin gilt:

$\left\|x_{\nu+1}-x_\nu\right\|=\left\|\varphi(x_\nu)-\varphi(x_\nu-9)\right\|\leq L\cdot\left\|x_\nu-x_{\nu-1}\right\|$ (Der Abstand zwischen den Iterierten verkürzt sich immer weiter)

und damit gilt induktiv:

$$\left\|x_{\nu+1}-x_\nu\right\|\leq L^\nu\cdot\left\|x_1-x_0\right\|$$

Wegen

$$\begin{aligned}\left\|x_{\nu+\mu}-x_\nu\right\|&=\left\|x_{\nu+\mu}-x_{\nu+\mu-1}+x_{\nu+\mu-1}-x_{\nu+\mu-2}+x_{\nu+\mu-2}-\ldots-x_{\nu+1}+x_{\nu+1}-x_\nu\right\|\\&\leq\left\|x_{\nu+\mu}-x_{\nu+\mu-1}\right\|+\ldots+\left\|x_{\nu+2}-x_{\nu+1}\right\|+\left\|x_{\nu+1}-x_\nu\right\|\\&\leq\left(L^{\nu+\mu-1}+\ldots+L^{\nu+1}+L^\nu\right)\cdot\left\|x_1-x_0\right\|\\&=L^\nu\cdot\left(L^{\nu-1}+\ldots+L+1\right)\cdot\left\|x_1+x_0\right\|\\&\leq\frac{L^\nu}{1-L}\cdot\left\|x_1-x_0\right\|\end{aligned}$$

ist $\left(x_\nu\right)_{\nu\in\mathbb{N}_0}$ eine Cauchy-Folge.

„Cauchy-Folge heißt, dass wenn ich ein ϵ vorgebe, einen Index finden kann, ab der die Differenz zwischen den aufeinanderfolgenden Folgengliedern ab diesem Index garantiert kleiner als ϵ ist."

Da $\mathbb{R}^n$ vollständig ist, gibt es ein $x^*\in\mathbb{R}^n$ mit $x_\nu\to x^*$. Da $\forall(\nu\in\mathbb{N}_0):\left(x_\nu\in D\right)$ und D abgeschlossen ist, folgt $x^*\in D$.

Wegen

$$\left\|x^* - \varphi(x^*)\right\| = \left\|x^* - x_{\nu+1} + \varphi(x_\nu) - \varphi(x^*)\right\|$$
$$\leqslant \left\|x^* - x_{\nu+1}\right\| + L \cdot \left\|x_\nu - x^*\right\| \to 0$$

muss $\left\|x^* - \varphi(x^\nu)\right\| = 0$ gelten, das heißt: $x^* = \varphi(x^*)$.

Sind x^* und x^{**} Fixpunkte von φ, so gilt

$$\left\|x^{**} - x^*\right\| = \left\|\varphi(x^{**}) - \varphi(x^*)\right\| \leqslant L \cdot \left\|x^{**} - x^*\right\|$$

beziehungsweise

$$(1-L) \cdot \left\|x^{**} - x^*\right\| \leqslant 0$$

oder

$$\left\|x^{**} - x^*\right\| \leqslant 0$$

Also gilt $\left\|x^{**} - x^*\right\| = 0$ beziehungsweise $x^{**} = x^*$.
Schließlich ergibt sich aus (6.7.a) direkt aus (6.6) für $x_1 = x_\nu$, $x_2 = x^*$ und (6.7.b) aus der Abschätzung für $\left\|x_{\nu+\mu} - x_\nu\right\|$ für $\nu \to \infty$.

Die Voraussetzung des Banachschen Fixpunktsatzes sind oft schwierig nachzuweisen. Bei dem folgenden Satz ist es oft einfacher. Dafür setzt er die Existenz eines Fixpunktes voraus.

Satz 6.3

Sei $x^* \in D$ ein Fixpunkt von $\varphi : D \to \mathbb{R}^n$, $D \subseteq \mathbb{R}^n$ offen. (Man beachte, dass hier nicht mehr nur auf den Definitionsbereich abgebildet wird.)
Weiterhin sei $\varphi \in C^1(D, \mathbb{R}^n)$ (φ ist in der Menge der mindestens einmal stetig ableitbaren Funktionen) und

(6.8) $\left\|\varphi'(x^*)\right\| < 1$

in einer Operatornorm.
Dann existiert eine Umgebung $U \subseteq D$ von x^*, sodass $\varphi|_U$ (Die Funktion φ eingeschränkt auf den Definitionsbereich U) die Vorraussetzung des Banachschen Fixpunktsatzes erfüllt.

Beweis

Sei $\delta = 1 - \left\|\varphi*(x^*)\right\|$. Wegen $\varphi \in C^1(D, \mathbb{R}^n)$ und $\left\|\varphi'(x^*)\right\| = 1 - \delta < 1$ existiert ein $\epsilon > 0$, sodass

$$\forall \left(x \in U = \overline{K_\epsilon(x^*)} = \left\{ x \in \mathbb{R}^n \middle| \left\|x - x^*\right\| \leqslant \epsilon \right\} \right) : \left(\left\|\varphi'(x)\right\| \leqslant 1 - \frac{\delta}{2} \right)$$

Damit folgt für $x_1, x_2 \in U$ in der zugehörigen Vektornorm (jetzt benutze ich den Hauptsatz der Integral- und Differenzialrechnung rückwärts, ich fasse φ als Stammfunktion auf)

$$\begin{aligned}\left\|\varphi(x_2)-\varphi(x_1)\right\| &= \left\|\varphi(x_1+s\cdot(x_2-x_1))\Big|_0^1\right\| \\ &= \left\|\int_0^1 \left(\varphi'(x_1+s\cdot(x_2-x_1))\cdot(x_2-x_1)\,ds\right)\right\| \\ &\leqslant \int_0^1 \left(\left\|\varphi'(x_1+s\cdot(x_2-x_1))\right\|\cdot\left\|x_1-x_1\right\|ds\right) \\ &\leqslant \left(1-\frac{\delta}{2}\right)\cdot\left\|x_2-x_1\right\|\end{aligned}$$

das heißt $\varphi|_U$ ist kontraktiv mit $L=1-\frac{\delta}{2}$.

Wählt man speziell $x_1=x^*$ und $x_2=x\in U$, so liefert obige Ungleichung

$$\left\|\varphi(x)-x^*\right\|=\left\|\varphi(x)-\varphi(x^*)\right\|\leqslant\left(1-\frac{\delta}{2}\right)\cdot\left\|x-x^*\right\|\leqslant\left\|x-x^*\right\|\leqslant\epsilon$$

das heißt

$$\forall(x\in U):\left(\varphi(x)\in U\right)$$

beziehungsweise

$$\varphi|_U: U\to U$$

Ist also φ stetig differenzierbar und gilt (6.8), so muss man nur x_0 hinreichend nahe bei x^* wählen, um Konvergenz zu erzielen. Man spricht von lokaler Konvergenz. Dabei kann die Bestimmung eines hinreichend guten Startwertes selbst ein schwieriges Problem sein.

Datum: 29.01.2003

Beispiel 6.4

Für Beispiel 6.1 mit $\varphi(x)=-e^x$ gilt bei der Wahl $D=[-1.0,-0.1]$ wegen

$$\varphi(-1.0)\approx-0.368\ ,\ \varphi(-0.1)\approx-0.905$$

un der Monotonie von φ, dass $\varphi(D)\subseteq D$.

Mit dem Mittelwertsatz (vergleiche Bemerkung 1.19) findet man als mögliches L in (6.6)

$$L=\sup_{x\in D}\left(\left|\varphi'(x)\right|\right)=\sup_{x\in[-1.0,-0.1]}\left(e^x\right)=e^{-0.1}\approx0.905<1$$

Nach dem Banachschen Fixpunktsatz existiert also ein eindeutiger Fixpunkt x^* von φ in $[-1.0,-0.1]$.

Satz 6.3 ist ebenfalls anwendbar mit

$$\tilde{L}=\left|\varphi'(x^*)\right|\approx0.567$$

Will mann x^* auf 12 Dezimalen bestimmen, so liefert der Ansatz

$$\tilde{L}^{\nu}=10-12\ ,$$

wenn man in (6.7b) einfach $\frac{1}{1-L}\cdot\left\|x_1-x_0\right\|=1$ setzt,

$$\nu=\frac{\log\left(10^{-12}\right)}{\log\left(\tilde{L}\right)}\approx49$$

als Schätzung für die nötige Anzahl von Iterationsschritten. Tatsächlich ist x_{44} die erste Iterierte, die auf 12 Dezimalstellen genau ist.

6.2 Intervallmethoden

Sei $f:[a,b]\to\mathbb{R}$ stetig und $f(a)\cdot f(b)\leqslant0$. Nach dem Zwischenwertsatz existiert dann ein $x^*\in[a,b[$ mit $f(x^*)=0$, Wählt man $c\in]a,b[$, so kann man allein durch Prüfen des Vorzeichens von $f(c)$ entscheiden, ob in $[a,c]$ oder in $[c,b]$ eine Nullstelle von f liegt. Durch iteratives Vorgehen erhält man sfort ein mögliches Verfahren zur Bestimmung einer Nullstelle von f. Wählt man jeweils die Intervallmitte, das heißt $c=\frac{1}{2}\cdot(a+b)$, so spricht man von Bisektion.

Algorithmus 6.5: Bisektion

Gegeben sei $a_0=a$, $b_0=b$ mit $f(a)\cdot f(b)\leqslant0$ und eine Toleranz $toleranz>0$.

Setze $k=0$
Schleife:
Setze $c_k=\frac{1}{2}\cdot(a_k+b_k)$
Ist $b_k-a_k\leqslant toleranz$, so akzeptiere c_k als Approximation an eine Nullstelle von f und höre auf (stop).
Ist $f(a_k)\cdot f(c_k)\leqslant 0$, so setze $a_{k+1}=a_k$ und $b_{k+1}=c_k$, andernfalls setze $a_{k+1}=c_k$ und $b_{k+1}=b_k$.
Setze $k=k+1$ und beginne Schleife

Natürlich ist im Fall $f(a_k)\cdot f(b_k)=0$ entweder a_k oder b_k Nullstelle und man könnte sofort abbrechen.

Satz 6.6

Ist $f:[a,b]\to\mathbb{R}$ stetig und $f(a)\cdot f(b)\leqslant 0$, so
[...abgewischt...]

und es gilt $f(x^*)=0$ mit der Abschätzung

(6.11) $$\left|x^*-c_k\right|\leqslant\left(\frac{1}{2}\right)^{k+1}\cdot(b-a)$$ (lineare Konvergenz)

Beispiel 6.7

Algorithmus 6.5 liefert angewandt auf das Problem aus Beispiel 6.1 mit $a=-1$, $b=0$ sowie $toleranz=10^{-12}$ als Approximation an x^* den Wert c_{40}. Man benötigt also weniger Iterationen als in Beispiel 6.4. Aus der Forderung

$$\left(\frac{1}{2}\right)^{k+1}\cdot(b-a)=10^{-12}$$

erhält man als Schätzung für die Anzahl der benötigten Iterationsschritte

$$k=\frac{\log\left(10^{-12}\right)}{\log(0.5)}-1\approx 39$$

Bemerkung 6.8: Regula falsi

Bei der Bisektion wird c unabhängig von den Werten $f(a)$ und $f(b)$ gewählt. Eine mögliche Wahl von c, die die Werte benutzt, ist die (eindeutige) Nullstelle der zugehörigen Interpolierenden.
Aus

(6.12) $$p(x)=\frac{f(b)-f(a)}{b-a}\cdot(x-a)+f(a)$$

folgt mit $p(c)=0$ die Darstellung

(6.13) $$c=a-f(a)\cdot\frac{b-a}{f(b)-f(a)}=\frac{a\cdot f(b)-b\cdot f(a)}{f(a)-f(b)}$$

Die sich durch iteratives Vorgehen entsprechend Algorithmus 6.5 ergebende Methode heißt

regula falsi. Außerdem muss das Abbrechkriterium geändert werden in

$$(c_k - a_k \leq toleranz) \vee (b_k - c_k \leq toleranz)$$

da im allgemeinen $b_k - a_k \to 0$.

Beispiel 6.9

Die regula falsi liefert mit leicher Ausgangssituation wie in Beispiel 6.7 als Approximation an x^* den Wert c_{13} .

6.3 Sekantenverfahren

Sei $f: D \to \mathbb{R}$ stetig mit $D \subseteq \mathbb{R}$ offen. Zu gegebenen Werten $a, b \in D$ kann man auch ohne die Bedingung $f(a) \cdot f(b) \leq 0$ die Nullstelle der linearen Interpolierenden berechnen, so lange $f(a) \neq f(b)$ ist. Verwendetman in (6.13) stets die zwei aktuellesten Approximationen, so erhält man die iterative Methode

(6.14) $$\forall (\nu \in \mathbb{N}): \left(x_{\nu+1} = \frac{x_{\nu-1} \cdot f(x_\nu) - x_\nu \cdot f(x_{\nu-1})}{f(x_\nu) - f(x_{\nu-1})} \right)$$ (Wobei x_0, x_1 gegeben sind)

das sogenannte Sekantenverfahren.

Während man im letzten Abschnitt immer mit Intervallen arbeitete, die eine Nullstelle x^* enthielten, berechnet man hier wie bei den Iterationsverfahren aus Abschnitt 6.1 eine Folge $\left(x_\nu\right)_{\nu \in \mathbb{N}_0}$, falls man in D verbleibt und $f(x_\nu) \neq f(x_{\nu-1})$ für alle $\nu \in \mathbb{N}$ gilt.

Satz 6.10

Sei x^* eine Nullstelle von $f \in C^2(D, \mathbb{R})$, $D \subseteq \mathbb{R}$ offen, mit $f'(x^*) \neq 0$. Dann definiert (6.14) für hinreichend gute Startdaten $x_0, x_1 \in D$ mit $x_0 \neq x_1$ eine Folge $\left(x_\nu\right)_{\nu \in \mathbb{N}_0}$ mit $x_\nu \to x^*$. Insbesondere gilt

(6.15) $$\left| x^* - x_\nu \right| \leq C \cdot K^{q^\nu}$$

mit $C \geq 0$, $k \in]0,1[$ und $q = \frac{1}{2} \cdot (1 + \sqrt{5}) \approx 1.618$.

Im Vergleich zu (6.15) haben (6.7b) und (6.11) eine Schranke der Form $C \cdot K^{\nu}$. Da q^{ν} wegen $q>1$ schneller wächst als ν, konvertgiert die Schranke in (6.15) schneller gegen null für $\nu \to \infty$. Man spricht von **superlinearer Konvergenz**.

Da die Startwerte hier auch hinreichend nahe an der gesuchten Lösung liegen müssen, liegt wieder lokale Konvergenz vor.

Beispiel 6.11

Für das Problem aus Beispiel 6.1 erhält man mit $x_0=-1$, $x_1=0$ die folgenden Werte:

$x_2=-0.\underline{57}2\ 181\ 412\ 091$
$x_3=-0.\underline{567\ 1}02\ 080\ 172$
$x_4=-0.\underline{567\ 143\ 3}27\ 950$
$x_5=-0.\underline{567\ 143\ 290\ 410}$

„Die lineare Konvergenz wäre, wenn nach jedem Schritt eine konstante Anzahl von Stellen signifikant werden. Superlineare Konvergenz liegt dann vor, wenn die Anzahl der der signifikanten Stellen schneller als linear wächst."

Man erkennt, dass man aufgrund der superlinearen Konvergenz desto mehr richtige Stellen pro Iterationsschritt gewinnt, je genauer die aktuelle Iterierte ist.

6.4 Newton-Verfahren

Statt die zu $x_{\nu-1}$, x_{ν} gehörige Sekante zu verwendent, kann man bei differenzierbarem f auch die zu x_{ν} gehörige Tangente verwenden. Diese ist gegeben durch

(6.16) $p(x)=f'(x_{\nu})\cdot(x-x_{\nu})+f(x_{\nu})$

mit der Nullstelle

(6.17) $x_{\nu+1}=x_{\nu}-\left(f'(x_{\nu})\right)^{-1}\cdot f(x_{\nu})$

Dies ist auch für Abbildungen $f: D \to \mathbb{R}^n$, $D \subseteq \mathbb{R}^n$ durchführbar.
Die Vorschrift (6.17) entspricht einem Iterationsverfahren (6.5) mit

(6.18) $\varphi(x)=x-\left(f'(x)\right)^{-1}\cdot f(x)$

Wegen

$$\varphi(x)=I-\left(\frac{d}{dx}\cdot\left(\left(f'(x)\right)^{-1}\right)\right)\cdot f(x)-\left(f'(x)\right)^{-1}\cdot f'(x)=-\left(\frac{d}{dx}\cdot\left(\left(f'(x)\right)^{-1}\right)\right)\cdot f(x)$$

gilt hier:

(6.19) $\varphi'(x^*)=0$ (in diesem Fall $f(x^*)=0$)

[...durch Windows-Aufhängen verpasst...]

Man nennt das durch (6.17) definierte Verfahren **Newton-Verfahren**.

Satz 6.12

Sei $f \in C^1(D, \mathbb{R}^n)$, $D \subseteq \mathbb{R}^n$ offen, $f'(x)$ nichtsingulär $\forall (x \in D)$ und

(6.20) $\forall (x, y \in D): \left(\left\| (f'(x))^{-1} (f'(y) - f'(x)) \right\| \leq \omega \cdot \|y - x\| \right)$

Weiter sei $x^* \in D$ eine Lösung von [...]

[...]

Beispiel 6.13

Für das Problem aus Beispiel 6.1 ist

$f(x) = e^x + x$, $f'(x) = e^x + 1$.

Das zugehörige Newton-Verfahren hat also die Form

$$x_{\nu+1} = x_\nu - \frac{e^{x_\nu} + x_\nu}{e^{x_\nu} + 1}$$

Ausgehend von $x_0 = -0.5$ erhält man die folgenden Werte:

$x_1 = -0.\underline{566}\,311003197$

$x_2 = -0.\underline{567143}\,165035$

$x_3 = -0.\underline{567143290410}$

Man erkennt die quadratische Konvergenz daran, dass sich die Anzahl der richtigen Stellen in jedem Interationsschritt verdoppelt.

Bemerkung 6.14 (Table-Lookup-Verfahren zur Berechnung von $\sqrt{a}$)

Im folgenden soll eine Möglichkeit, wie man das Newton-Verfahren als Basis für die Auswertung der Wurzelfunktion verwenden kann, behandelt werden. Offensichtlich besitzt

(6.23) $f(x) = x^2 - a$ mit $a > 0$

die eindeutige positive Nullstelle $x^* = \sqrt{a}$.

Das zugehörige Newton-Verfahren kann auf die Form

(6.24) $x_{\nu+1} = \frac{1}{2} \cdot \left(x_\nu + \frac{a}{x_\nu} \right)$ (Verfahren von Heron)

gebracht werden.

Schreib man a in der Form

(6.25) $a = m \cdot 2^r$

so sind m und r eindeutig festgelegt durch die Forderung

(6.26) $m \in \left[\frac{1}{4}, 1 \right)$ wobei r gerade ist

Es ist dann

(6.27) $\sqrt{a} = \sqrt{m} \cdot 2^{\frac{r}{2}}$

das heißt: man kann sich auf $a \in \left[\frac{1}{4}, 1 \right)$ einschränken.

Für (6.24) gilt

$$
\begin{aligned}
x_{\nu+1}-\sqrt{a} &= \frac{1}{2}\cdot\left(x_\nu+\frac{a}{x_\nu}\right)-\sqrt{a} \\
&= \frac{1}{2\cdot x_\nu}\cdot\left(x_\nu{}^2-2\cdot\sqrt{a}\cdot x_\nu+a\right) \\
&= \frac{1}{2\cdot x_\nu}\cdot\left(x_\nu-\sqrt{a}\right)^2
\end{aligned}
\tag{6.28}
$$

das heißt für $x_\nu\in\left[\frac{1}{2},1\right]$

$$\left|x_\nu-\sqrt{a}\right|\leqslant\left(\left|x_0-\sqrt{a}\right|\right)^{2^\nu} \tag{6.29}$$

(vergleiche: quadratische Konvergenz)

Betrachtet man nur die ersten k Bits von a, so erhält man ein $\tilde{a}\in\left[\frac{1}{4},1\right]$ mit

$$|a-\tilde{a}|\leqslant 2^{-k} \tag{6.30}$$

Als Startwert verwendet man dann $x_0=\sqrt{\tilde{a}}$, das man (etwa bis auf Maschinengenauigkeit) fest abgespeichert hat (sogenannter **Table-Lookup**).

Beachtet man, dass

$$\left|\sqrt{a}-\sqrt{\tilde{a}}\right|\leqslant|a-\tilde{a}| \tag{6.31}$$

nach dem Mittelwertsatz, und wählt man k und ν so, dass

$$2^{-k}\leqslant\sqrt[2^\nu]{eps}\,, \tag{6.32}$$

sogilt

$$\left|x_\nu-\sqrt{a}\right|\leqslant\left(\left|x_0-\sqrt{a}\right|\right)^{2^\nu}=\left(\left|\sqrt{\tilde{a}}-\sqrt{a}\right|\right)^{2^\nu}\leqslant\left(|\tilde{a}-a|\right)^{2^\nu}\leqslant\left(2^{-k}\right)^{2^\nu}\leqslant eps \tag{6.33}$$

Mit $eps=2^{-l}$ ergibt sich aus (6.32)

$$k\geqslant\frac{l}{2^\nu} \tag{6.34}$$

Damit hat man für $l=53$ (typische IEEE-Arithmetik) die Möglichkeiten

(6.35)

ν	0	1	2	3	4	5
k	53	27	14	7	4	2

Datum: 05.02.2003

Dabei ist ein Speicherbedarf von etwa 2^k Fließkommazahlen zu beachten. Eine günstige Wahl scheint deshalb $\nu=3$, $k=7$ zu sein. Man beachte, dass (6.24) den Nachteil besitzt, eine Division zu enthalten.

„Es gibt noch bessere Verfahren, die keine Division enthalten, aber $\frac{1}{\sqrt{a}}$ zurückliefern. Dies ist aber nicht schlimm, weil man dies mit a multiplizieren kann, um $\sqrt{a}$ zu erhalten. Weiterhin existieren Verfahren, die sogar kubisch statt quadratisch konvergieren. Diese beinhalten allerdings schon Parallelisierungstechniken."

7. Eigenwertprobleme

Gegeben sei eine Matrix $A \in M(n \times n, \mathbb{C})$. Gesucht sind einzelne oder alle Eigenwerte von A, das heißt Werte $\lambda \in \mathbb{C}$, für die

(7.1) $\exists \left(x \in \left(\mathbb{C}^n \setminus \{0\} \right) \right) : \left(A \cdot x = \lambda \cdot x \right)$

gilt.

7.1 Kondition

Satz 7.1: Bauer|Fike

Seien $A, E \in M(n \times n, \mathbb{C})$. Ist μ Eigenwert von $A+E$ und $X^{-1} \cdot A \cdot X = D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, so gilt

(7.2) $\min_{i=1,\ldots,n} \left(\left| \lambda_i - \mu \right| \right) \leq \operatorname{cond}_2(X) \cdot \left\| E \right\|_2$

Die Kondition des Eigenwertproblems wird also für diagonalisierbare Matrizen beschrieben durch

(7.3) $\kappa = \operatorname{cond}_2(X)$

das heißt durch die Kondition der Translationsmatrix X, die A diagonalisiert (und nicht durch die Kondition von A selbst). „Dies ist ein großer Unterschied, ob die Kondition von X oder von A abhängt."

Bemerkung 7.2

Ist $A \in M(n \times n; \mathbb{C})$ hermitesch, das heißt: $A^H = A$, wobei

(7.4) $\left(a_{ik} \right)^H = \left(\overline{a}_{ij} \right), \quad A = \left(a_{ij} \right)$

so ist

(7.5) $\kappa = 1$

das heißt, das Eigenwertproblem ist in diesem Fall gut konditioniert.
Man beachte, dass (7.4) für $A \in M(n \times n; \mathbb{R})$ gleichbedeutend mit $A^T = A$, das heißt mit A symmetrisch, ist. „(hermitesch ist lediglich die Verallgemeinerung der der Symmetrie reeller Zahlen auf die komplexen Zahlen)"

Bemerkung 7.3

Die Eigenwerte von A sind gerade die Nullstellen des zugehörigen charakteristischen Polynoms

(7.6) $p(\lambda) = \det \left(A - \lambda \cdot E_n \right)$

Der Algorithmus, zuerst p und dann dessen Nullstellen zu bestimmenm, ist allerdings instabil.
„Dies sollten Sie also gar nicht erst versuchen, erst das charakteristische Polynom und dann dessen Nullstellen zu bestimmen. In der Realität wird oft gerade der Weg andersherum gegangen, wenn man die Nullstellen eines Polynoms bilden möchte: Man generiert eine Matrix, dessen charakteristisches Polynom dem gegebenem Polynom entspricht und berechnet die Eigenwerte dieser Matrix."

„Man kann sich folgende Überlegung zu nutze machen: Man nehme einen beliebigen Vektor x und wende sukzessive immer wieder A darauf an. Man hofft, dass sich dadurch der Vektor x in die Richtung des betragsgrößten Eigenvektors dreht. Ist dies der Fall, so lässt sich dann

die jeweilige Verlängerung des Vektors x als ein Eigenwert interpretieren."

7.2 Vektoriteration und inverse Iteration

Hat A einen eindeutigen betragsgrößten Eigenwert, so kann man das folgende Verfahren verwenden.

Algorithmus 7.4 (Vektoriteration bzw. Potenzmethode)

Gegeben sei $A \in M(n \times n, \mathbb{C})$ und $x^{(0)} \in \left(\mathbb{C}^n \setminus \{0\}\right)$, sowie $toleranz > 0$

(7.7)

Setze $k = 1$	
Schleife über erlaubte Anzahl von Schritten	
	Setze $z^{(k)} = A \cdot x^{(k-1)}$
	Setze $\lambda^{(k)} = \frac{z^{(k)}{}_i}{x^{(k-1)}{}_i}$ wobei $\left\|z^{(k)}{}_i\right\| = \left\|z^{(k)}\right\|_\infty$: Man wählt also ein $i = 1, \ldots, n$, wobei das $\left\|z_i^{(k)}\right\|$ das betragsgrößte ist.
	Ist $\left\|\lambda^{(k)} - \lambda^{(k-1)}\right\| \leqslant toleranz$, so akzeptiere $\lambda^{(k)}$ als Eigenwert von A und stoppe.
	Setze $x^{(k)} = \frac{z^{(k)}}{z^{(k)}{}_i}$ („Renormierung, damit sich der Vektor nicht zu stark in Richtung 0 oder unendlich bewegt, weil sonst sehr schnell ein Über- oder Unterlauf passieren kann")
	Setze $k = k + 1$ und beginne Schleife.

Lemma 7.5

Sei $A \in M(n \times n, \mathbb{C})$ diagonalisierbar mit $X^{-1} \cdot A \cdot X = diag\left(\lambda_1, \ldots, \lambda_n\right)$ und sei

(7.8) $\left|\lambda_1\right| > \left|\lambda_2\right| \geqslant \left|\lambda_3\right| \geqslant \ldots \geqslant \left|\lambda_n\right|$

Dann gilt für Algorithmus 7.3 für hinreichend allgemein („spricht: per Zufallsgenerator") gewähltes $x^{(0)} \in \left(\mathbb{C}^N \setminus \{0\}\right)$, dass

(7.9) $\left|\lambda^{(k)} - \lambda - \lambda_1\right| \leqslant C \cdot \left(\left|\frac{\lambda_2}{\lambda_1}\right|\right)^k$ (lineare Konvergenz)

mit $C \geqslant 0$ unabhängig von k.

Um nicht nur den betragsgrößten Eigenwert von A erhalten zu können, kann man folgendermaßen vorgehen. Hat A den Eigenwert λ, so hat

- $A - \mu \cdot E_n$ den Eigenwert $\lambda - \mu$ (Addieren von Einheitsmatrizen auf die Matrix)
- $\left(A - \mu \cdot I\right)^{-1}$ den Eigenwert $\left(\lambda - \mu\right)^{-1}$ (Invertieren von Matrix und Eigenwert)

und liegt μ näher an λ als an allen anderen Eigenwerte von A, so ist $\left(\lambda - \mu\right)^{-1}$ der betragsgrößte Eigenwert von $\left(A - \mu \cdot E_n\right)^{-1}$.

Zu dessen Berechnung bietet sich damit Vektoriteration mit A ersetzt durch $\left(A-\mu\cdot E_n\right)^{-1}$ an. Dies ist die sogenannte **inverse Iteration** nach Wielandt.

Algorithmus 7.6 (Inverse Iteration nach Wielandt)

Gegeben sei $A\in M(n\times n,\mathbb{C})$, $x^{(0)}\in\left(\mathbb{C}^N\setminus\{0\}\right)$ und $\mu\in\mathbb{C}$ sowie $toleranz>0$

(7.10)

Setze $k=1$	
Berechne die LR-Zerlegung von $A-\mu\cdot E_n$. Ist die Zerlegung nicht möglich (weil die Matrix $A-\mu\cdot I$) singulär ist, dann sollten wir nicht traurig sein. In diesem Fall ist nämlich μ ein Eigenwert von A und wir können aufhören.	
Schleife über die erlaubte Anzahl von Schritten	
	Bestimme $z^{(k)}=\left(A-\mu\cdot E_n\right)^{-1}\cdot x^{(k-1)}$ durch Vorwärts-|Rückswärts-substitution.
	Setze $\tilde{\lambda}^{(k)}=\frac{z^{(k)}{}_i}{x^{(k-1)}{}_i}$ wobei $\left\|z^{(k)}{}_i\right\|=\left\|z^{(k)}\right\|_\infty$: : Man wählt also ein $i=1,\ldots,n$, wobei das $\left\|z_i^{(k)}\right\|$ das betragsgrößte ist.
	Ist $\left\|\tilde{\lambda}^{(k)}-\tilde{\lambda}^{(k-1)}\right\|\leqslant toleranz$, so akzeptiere $\lambda^{(k)}=\frac{1}{\tilde{\lambda}^{(k)}}+\mu$ als Eigenwert von A und stoppe.
	Setze $x^{(k)}=\frac{z^{(k)}}{z^{(k)}{}_i}$ („Renormierung, damit sich der Vektor nicht zu stark in Richtung 0 oder unendlich bewegt, weil sonst sehr schnell ein Über- oder Unterlauf passieren kann“)
	Setze $k=k+1$ und beginne Schleife.