Computerphysik

WS 2016/2017

Vorbemerkung

Dieses Skript ist inspiriert durch die Vorlesung Computerphysik gehalten von Prof. Roland Netz im WS13/14, WS 14/15 und WS 15/16 und daher in weiten Teilen nahezu identisch mit dem dazugehörigen Skript. Das vorliegende Skript dient lediglich dazu, die Inhalte darzustellen wie sie in der Vorlesung im WS16/17 präsentiert werden, d.h. mit evtl. geänderter Reihenfolge, anderen Beispielen und Randbemerkungen, und mit neuen Tipp- und anderen Fehlern. Es ist als Service zu verstehen, nicht als Plagiatsversuch. Da dieses Skript „mitwächst“ sind z.B. die Seitenzahlen noch nicht final.

Inhaltsverzeichnis

Einleitung

Was soll "Computerphysik"?

Zunächst wollen wir klarstellen, dass "Computerphysik" keine "neue" Art Physik ist und auch nicht die Physik beschreiben soll, welche im Computer passiert (Halbleiterchips etc.).

Computerphysik soll veranschaulichen wie physikalische Fragestellungen mit Hilfe des Computer, also durch numerische Verfahren, behandelt werden können. Zu diesem Zweck werden verschiedene wichtige Algorithmen für verschiedene typische numerische Probleme vorgestellt und miteinander verglichen. Die begleitenden Übungen sollen Beispiele für die praktische Anwendung aufzeigen und nebenher werden Grundzüge der Programmierung in Python gelernt. Ziel ist, die Nutzerin[1] in die Lage zu versetzen, physikalische Problemstellungen, wo angebracht, in eine numerische Aufgabe zu übersetzen und ein geeignetes Verfahren zu deren Lösung auszuwählen.

Computerphysik ist kein reiner Programmierkurs und auch keine Vorlesung in numerischer Mathematik, weswegen die Beweisführung (und einiges an mathematisch notwendiger sprachlicher Genauigkeit) in aller Regel vernachlässigt wird.

1 Fehler

Wie immer im Leben, können auch in der Computerphysik, d.h. beim Rechnen mit dem Computer Fehler auftreten. Hier sind einige Fehlertypen zu unterscheiden, und zwar solche, die dem Rechner "anzulasten" sind, und andere. Zu den anderen gehören

1.1 Eingabefehler

Eingabefehler entstehen z.B. durch Tippfehler, also die Eingabe eines falschen Wertes, den Aufruf des falschen (Unter-)Programmes, etc.

1.2 Modellfehler

Fehler werden auch immer dort gemacht, wo Näherungen an die Stelle einer exakten Beschreibung treten; Modelle beschreiben die Wirklichkeit immer nur so weit "wie nötig" (oder auch "wie möglich"). Beispiele sind die Wechselwirkung zweier Teilchen unter Vernachlässigung des Restes des Universums. Oder auch "sphärische Kühe". Ein weiteres Beispiel, das auch zu den Eingabefehlern gezählt werden kann, ist die (implizite) Verwendung falscher Einheiten, z.B. Zoll statt Zentimeter.

Fehler, die typisch für Computer sind, resultieren aus folgendem Fakt:

Im Computer sind die darstellbaren Zahlen endlich.

1.3 Diskretisierungsfehler

Die Maschinenzahlen (im Computer darstellbaren Zahlen) sind endlich und haben außerdem einen endlichen Abstand voneinander. Hieraus resultieren Diskretisierungsfehler dort, wo eine kontinuierliche Beschreibung nötig wäre. D.h. "unendlich kleine" (infinitesimale) Schrittweiten und Funktionswertveränderungen werden durch sehr kleine (finite) Schritte angenähert. Die Differenzzwischen theoretischer exakter und berechneter Lösung ist der Diskretisierungsfehler. diese werden wichtig bei Differentiation und Integration (Abschnitt 5), Differenzialgleichungen (Abschnitt 6) oder Finite-Elemente-Methoden und in den entsprechenden Abschnitten diskutiert.

[1] Zur besseren Lesbarkeit wird in diesem Skript nur die weibliche Form verwendet. Sebstverständlich sind immer alle Geschlechter gemeint.

1.4 Rundungsfehler

Rundungsfehler (auch) resultieren daher, dass eine Maschinenzahl nur mit endlicher Genauigleit dargestellt werden kann.

1.4.1 Binäre Zahlen: Integer und Gleitkommazahlen

Im Allgemeinen werden Zahlen im Comuter als Binärzahlen, zur Basis 2, durch die Ziffer 0 und 1 dargestellt. Im Dezimalsystem wird entsprechend die Basis 10 benutzt:

dezimal	0	1	2	3	4	5	6	7	8	9	10	11	...
binär	0	1	10	11	100	101	110	111	1000	1001	1010	1011	

.

Die Zahl 14 im Zehnersystem, setzt sich zusammen aus $1*10^1+4*10^0 = 10+4$. Im Binärsystem wird daraus $1*2^3+1*2^2+1*2^1+0*2^0 = 8+4+2$, also $(14)_{10} = (1110)_2$. Binärzahlen haben nicht nur den Vorteil, dass sie "an" und "aus" repräsentieren können, sondern sie sind auch gleich in "wahr (true)" oder "falsch (false)" zu übersetzen, so dass sich folgende logische Verknüpfungen leicht darstellen lassen, z.B.

Negation ("nicht"):

$\neg$	
0	1
1	0

Und (beide "true" bzw. "1"):

$\wedge$	0	1
0	0	0
1	0	1

Oder (mindestens eines von beiden "true" bzw. "1"):

$\vee$	0	1
0	0	1
1	1	1

Exklusives Oder (genau eines von beiden "true" bzw. "1"):

Xor	0	1
0	0	1
1	1	0

.

Das schriftliche Rechnen (auf dem Papier) in Addition und Multiplikation kann genauso erfolgen woe im Dezimalsystem, nur dass ein Übertrag eben schon bei 2 und nicht erst bei 10 auftritt.

Beispiel $(1110)_2 + (111)_2 = \begin{array}{r} 1110 \\ + \quad 111 \\ \hline 10101 \end{array} = (10101)_2$

Die Anzahl der darstellbaren Zahlen wird durch die Anzahl verfügbarer Stellen, Bit, begrenzt. Für 32 bit sind das ca. 4 Milliarden, und für 64 bit schon 18 Trillionen.

Integer-Zahlen dienen der Darstellung von Ganzzahlen z. Sie werden direkt als Binärzahlen dargestellt, wenn $z \geq 0$. Mit 8 bit sieht das also so aus:

$(z)_{10}$	$(z)_2$
0	00000000
1	00000001
2	00000010
3	00000011
...	...

Negative Zahlen, $z < 0$, werden im Zweierkomplement dargestellt. Dieses wird gebildet, indem man die Darstellung der positiven Zahl nimmt und umkehrt, also 0 und 1 vertauscht, und dann noch +1 addiert:

$(z)_{10}$	$(z)_2$	$(z)_2$
-1	11111110+1	11111111
-2	11111101+1	11111110
-3	11111100+1	11111101
...	...	...

Damit kann indirekt das erste, höchstwertige, Bit als Marker für das Vorzeichen verstanden werden. Es gibt keine negative Null. Der Vorteil liegt darin, dass keine beondere Unterscheidung beim Rechnen mit positiven oder negativen Zahlen gemacht werden muss. Rechnen wir beispielsweise $(-2)_{10} + (8)_{10} = (6)_{10}$ so ist das in Binärdarstellung mit Zweierkomplement

$$\begin{array}{rr} & 11111110 \\ + & 00001000 \\ \hline & 00000110 \end{array}$$

Gleitkommazahlen oder auch Gleitpunkt-, Fließkomma-, oder Fließpunktzahlen (floating point numbers) dienen der Darstellung von reellen Zahlen. Im Gegensatz zu Festkommazahlen, bei denen die Anzahl Stellen vor und nach dem Komma fest steht, ist die Position des Kommas in der Zahl mit codiert. Dies wird erreicht durch eine normaisierte Exponentialschreibweise der Form

$$m \cdot B^E$$

in der m die Mantisse, B die Basis und E den Exponenten angibt. Im Zehnersystem kann man z.B. schreiben $(12000)_{10} = 12.0 \cdot 10^3 = 1.2 \cdot 10^4 = 0.12 \cdot 10^5$ die Normalisierung wird durch Festlegung der Ziffer vor dem Komma erzielt, im Zehnersystem hier die 0, da sich so alle Dezimalzahlen schreiben lassen. Für Binärzahlen wird hingegen eine 1 als Ziffer vor dem Komma festgelegt. Nach IEEE Standard 754 ist eine Gleitkommazahl in *single precision* mit 32 bit so codiert, dass 1 bit auf das Vorzeichen entfällt, $r = 8$ bits auf den Exponenten und $p = 23$ auf die Mantisse. Dabei braucht die 1 vor dem Komma wegen der Normaisierung nicht geschrieben werden:

$$\underbrace{s}_{1}\ \underbrace{eeeeeeee}_{r=8}\underbrace{mmmmmmmmmmmmmmmmmmmmmmm}_{p=23}$$

Für *double precision*, also 64 bit, Gleitkommazahlen stehen $r = 11$ und $p = 52$ Zeichen für Exponent bzw. Mantisse zur Verfügung. Das Vorzeichen ergibt sich als $(-1)^s$, eine negative Zahl wird also durch $s = 1$ und eine positive durch $s = 0$ codiert. Um nicht auch noch ein Vorzeichen für den Exponenten mitzuführen wird dieser mit einem Bias b versehen, der sich aus der Anzahl Stellen für den Exponenten ergibt als $b = 2^{r-1} - 1$. In single precision ist das also $r = 8$ und $b = 2^7 - 1 = 127$. Ein Exponent von $E = (3)_{10}$ wird also zu $e = E + b = (3)_{10} + (127)_{10} = (130)_{10} = (10000010)_2$ und $E = (-3)_{10}$ entsprechend zu $e = E + b = (-3)_{10} + (127)_{10} = (124)_{10} = (01111100)_2$. Damit sind der niedrigste und höchste mögliche Exponent vorgegeben zu $-126 \leq E < 127$. Die Null wird durch $e = 0$ und $m = 0$ speziell codiert, durch

das Vorzeichenbit allerdings als positive oder negative Null. $\pm\infty$ wird durch $e = 2^r - 1$ coiert, wenn $m = 0$, Fälle mit $m > 0$ werden als NaN (Not a number) behandelt.

Die kleinste darstellbare Zahl ist also durch $1 \cdot 2^{E_{min}}$ festgelegt, die größte darstellbare Zahl entsprechend als $(1.111\ldots)_2 \cdot 2^{Emax}$ mit p vielen Einsen nach dem Komma, das entspricht der Dezimalzahl, die sich mit

$$\sum_{i=0}^{p} \frac{1}{2^i} = \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \ldots$$

errechnen lässt. Alternativ lässt sich die größte darstellbare Zahl als $(2 - 2^{-p}) \cdot 2^{Emax}$ angeben. Mit sagen wir z.B. $p = 4$ ist also die größtmögliche Mantisse (die Zahlen in der Rechnung sind nicht normalisiert dargestellt) $\begin{array}{rr} & (10.0000)_2 \\ + & (0.0001)_2 \\ \hline & (1.1111)_2 \end{array}$.

Beispiel zur Umrechnung von Dezimalzahlen in Binärzahlen soll sein die Zahl $(23.625)_{10}$. Wir wandeln Vorkomma und Nachkommateil der Zahl getrennt um. Für den Vorkommateil teilen wir solange durch 2 bis die Null erreicht ist. Den Rest notieren wir als das zu schreibende Bit:

23/2	=11	Rest 1
11/2	=5	Rest 1
5/2	=2	Rest 1
2/2	=1	Rest 0
1/2	= 0	Rest 1

Hierbei ist das letzte "errechnete" Bit das höchstwertige. Wir erhalten also $(23)_{10} = (10111)_2$. Den Nachkommteil erhalten wir durch multiplizieren mit 2. Ergibt sich ein Wert größer oder gleich 1 wird eine 1 gesetzt, ansonsten eine 0. Dann wird eine Kommastelle weitergerückt, also nach

0.625*2	=1.25 ≥ 1	1
0.25*2	=0.5 <1	0
0.5*2	=1.0≥ 1	1
0.0*2	=0.0 <1	0
1/2	= 0	0

ist der Nachkommaanteil $(0.625)_{10} = (10100)_{0.2}$ und zusammengesetzt erhalten wir $(23.625)_{10} = (10111.10100)_2$ in Festkommaschreibweise. Als normalisierte Gleitkommazahl müssen wir jetzt noch das Komma so weit verschieben, dass nur noch eine 1 vor dem Komma steht und denentsprechenden Exponenten in Binärschreibweise und mit Bias darstellen. Das bedeutet wir bekommen $(1.011110100) \cdot \left(2^4\right)_{10}$. Durch Addition des Bias ergibt sich für den Exponenten $E = e + b = 4 + 127 = (131)_{10} = (10000011)_2$. Das Vorzeichen ist positiv wird also durch eine 0 dargestellt, so dass schließlich in single precision Standarddarstellung:

$$0\ 10000011011110100000000000000000$$

Die endliche Menge an Ziffern (Bits) hat in der Gleitkommadarstellung nicht nur zur Folge, dass es eine kleinste und eine größte darstellbare Zahl gibt, sondern auch, dass Maschinenzahlen gerundete Zahlen sind. Als Maschinengenauigkeit *machine epsilon* bezeichnet man den Abstand der kleinsten normalisierte Gleitkommazahl mit p-stelliger Mantisse $x = \pm 1.\underbrace{d_1 d_2 \ldots}_{p}$, welche noch größer Eins zur Eins, also $eps = (1.000\ldots01)_B \cdot B^0 - 1$.

Dies ist aber nur der kleinste auftretende Rundungsfehler. Der Rundungsfehler der Gleitkommazahlen wird von $|MIN|$ nach $|MAX|$, also der betragsmäßig kleinsten zur betragsmäßig größten darstellbaren

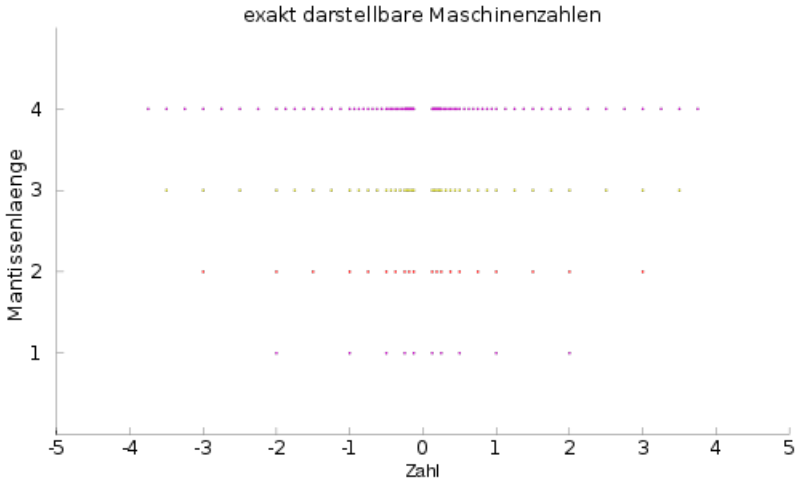

Abbildung 1: Exakt darstellbare Gleitkommazahlen für verschiedene Mantissenlängen, Basis: 2, Exponent −3 bis 1 (aus Wikipedia)

Zahl größer. Schauen wir uns ein Beispiel mit 5 Bit an, wobei 2 Bit auf den Exponenten und 2 auf die Mantisse entfallen.

$(x)_2$	$(x)_{10}$
$(0.00)_2 \cdot 2^0$	0
$(1.00)_2 \cdot 2^{-1}$	$\frac{4}{8}$
$(1.01)_2 \cdot 2^{-1}$	$\frac{5}{8}$
$(1.10)_2 \cdot 2^{-1}$	$\frac{6}{8}$
$(1.11)_2 \cdot 2^{-1}$	$\frac{7}{8}$
$(1.00)_2 \cdot 2^0$	$\frac{8}{8}$
$(1.01)_2 \cdot 2^0$	$\frac{10}{8}$
$(1.10)_2 \cdot 2^0$	$\frac{12}{8}$
$(1.11)_2 \cdot 2^0$	$\frac{14}{8}$
$(1.00)_2 \cdot 2^1$	$\frac{16}{8}$
$(1.01)_2 \cdot 2^1$	$\frac{20}{8}$
$(1.10)_2 \cdot 2^1$	$\frac{24}{8}$
$(1.11)_2 \cdot 2^1$	$\frac{28}{8}$

Die aus Wikipedia entnommene Abbildung 1 illustriert dies.

Rechnen mit Gleitkommazahlen ist entsprechend mit Fehlern behaftet. Die folgenden Beispiele sind der besseren Lesbarkeit wegen mit Dezimalzahlen dargestellt.

Auslöschung tritt auf, wenn zwei fast gleich große, auf den ersten k Stellen übereinstimmende, Zahlen voneinander subtrahiert werden. Im Ergebnis gehen dann k Stellen verloren. Bei Addition tritt dieses Problem nicht auf. Beispiel mit Rundung $rd()$ auf vier Nachkommastellen:

$x_1 = 0.11258762 \cdot 10^2$	$rd(x_1) = 0.1126 \cdot 10^2$
$x_2 = 0.11244891 \cdot 10^2$	$rd(x_2) = 0.1124 \cdot 10^2$
$x_1 - x_2 = 0.00013871 \cdot 10^2$	$rd(x_1) - rd(x_2) = 0.0002 \cdot 10^2$
$x_1 + x_2 = 0.22503653 \cdot 10^2$	$rd(x_1) + rd(x_2) = 0.2250 \cdot 10^2$

Wird eine kleine Zahl zu einer sehr grossen addiert, so wird die kleinere Zahl wegen der endlichen Darstellung in die grosse *absorbiert*, d.h. ie große Zahl ändert sich nicht. Für die Addition von $1.0 \cdot 10^5$ und $1.5 \cdot 10^1$ muss der Exponent angeglichen werden, also

$$\begin{array}{rr} & 1.00000 \cdot 10^5 \\ + & 0.00015 \cdot 10^5 \\ \hline & 1.00015 \cdot 10^5 \end{array} \text{ wird bei Rundung auf 3 Nachkommastellen zu } \begin{array}{rr} & 1.000 \cdot 10^5 \\ + & 0.000 \cdot 10^5 \\ \hline & 1.000 \cdot 10^5 \end{array}$$

Entsprechend kommt es bei der Addition von mehreren Zahlen auf die Reihenfolge an. Ein extremes Beispiel ist 2590+4+4 . Mit drei Ziffern bekommt man entweder $2590 + 4 \to 2590$ (gerundet von 2594) und noch einmal $2590 + 4 \to 2590$

oder

$4 + 4 \to 8$ und $2590 + 8 \to 2600$

anstelle der exakten 2598. Der Fehler im Ergebnis hängt also von der Reihenfolge der Addition ab und ist kleiner, wenn man zunächst die ähnlich großen Zahlen addiert.

Wir stellen also fest, dass im Computer nicht nur die darstellbaren Zahlen endlich sind , sondern auch

das Assoziativgesetz $(a + b) + c = a + (b + c)$ gilt im Computer nicht mehr

und auch

das Distributivgesetz $a\,(b + c) = (a \cdot b) + (a \cdot c)$ gilt im Computer nur noch eingeschränkt.

1.5 Fehlerfortpflanzung

Die durch die endliche Darstellung unvermeidbaren Rundungsfehler werden durch die Rechnung fortgetragen. Bei der Addition (Subtraktion entsprechend) ist der Gesamtfehler der gerundeten Zahlen $\tilde{x}$ und $\tilde{y}$ gegenüber den exakten Zahlen x und y

$$(\tilde{x} + \tilde{y})\,(x - y) = (\tilde{x} - x) + (\tilde{y} - y)$$

schlimmsten Falls additiv. Der Fehler in der Multiplikation (Division entsprechend) ist

$$(\tilde{x}\tilde{y}) - (xy) = \tilde{x}\,(\tilde{y} - y) + \tilde{y}\,(\tilde{x} - x) - (\tilde{x} - x)\,(\tilde{y} - y)$$

gegeben durch den jeweiligen einen Faktor multipliziert mit dem Fehler des anderen. Der relative Fehler des Produkts ist

$$\frac{(\tilde{x}\tilde{y}) - (xy)}{\tilde{x}\tilde{y}} = \frac{(\tilde{y} - y)}{\tilde{y}} + \frac{(\tilde{x} - x)}{\tilde{x}} - \frac{(\tilde{x} - x)\,(\tilde{y} - y)}{\tilde{x}\tilde{y}}$$

Funktionen angewendet auf eine fehlerbehaftete Zahl, also Auswertung von f an der Stelle $\tilde{x}$ anstatt x liefern auch fehlerbehaftete Funktionswerte $f\,(\tilde{x}) \neq f\,(x)$. Wie hoch diese Fehlerverstärkung durch die Funktionsauswertung ist, hängt von der Funktion ab. Wie lässt sich diese Fehlerverstärkung schätzen?

Hierzu bedienen wir uns des Mittelwertsatzes nachdem für ein x_0 im Intervall $[x, \tilde{x}]$, also einer Zahl die irgendwo zwischen der gerundeten und der exakten Zahl liegt, einen Funktionswert $g\,(x_0)$ bestimmen mit

$$\frac{\int_x^{\tilde{x}} g\,(x')\,dx'}{\tilde{x} - x} = g\,(x_0) \tag{1}$$

Wählen wir jetzt $g\,(x_0) = f\prime\,(x)$ als die Ableitung der zu betrachtenden Funktion $f\,(x)$ erhalten wir[2]

$$\begin{array}{rcl} \dfrac{f\,(\tilde{x}) - f\,(x)}{\tilde{x} - x} & = & f\prime\,(x_0) \\ |f\,(\tilde{x}) - f\,(x)| & = & |\tilde{x} - x| \cdot |f\prime\,(x_0)| \end{array}$$

[2]natürlich muss $f\,(x)$stetig, differenzierbar etc. sein.

Der Fehler wird also bei einer Ableitung >1 größer und bei einer Ableitung < 1 kleiner. Die schlimmstmögliche Fehlerverstärkung M ist gegeben durch

$$M = \max_{[x,\tilde{x}]} |f\prime(x_0)|$$

Da allerdings x_0 nicht bekannt ist (wir kennen x ja auch nicht), kann man die Fehlerverstärkung nur mit $|f\prime(\tilde{x})|$ schätzen[3].

Der relative Fehler bei Funktionsauswertung ist nach

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \leq \frac{\max_{[x,\tilde{x}]} |f\prime(x_0)|\,|x|}{|f(x)|} \cdot \frac{|\tilde{x} - x|}{|x|} \tag{2}$$

beschränkt und kann durch

$$\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \leq \frac{|f\prime(\tilde{x})|\,|x|}{|f(x)|} \cdot \frac{|\tilde{x} - x|}{|x|} \tag{3}$$

geschätzt werden. Der Faktor $\frac{|f\prime(\tilde{x})||x|}{|f(x)|}$ heißt *Konditionszahl* und gibt den Faktor an, um den ein Eingangsfehler im ungünstigsten Fall verstärkt werden kann. Den Betrag $|f(\tilde{x}) - f(x)|$nennt man *Stabilität*.

Beispiel 1: Für die Funktion $f(x) = \sin(x)$ mit Ableitung $f\prime(x) = \cos(x)$ wissen wir, dass $\max |f\prime(x)| = 1$. Der absolute Fehler bleibt also nach Funktionsauswertung gleich oder verkleinert sich.

Beispiel 2: Für die Funktion $f(x) = \sqrt{(x)}$ mit Ableitung $f\prime(x) = \frac{1}{2\sqrt{x}}$ ergibt sich bei einer Auswertung nahe Null eine starke Vergrößerung des absoluten Fehlers, z.B. $x = 0.010$ und $\tilde{x} = 0.011$. Dann ist $|\tilde{x} - x| = 0.001$ und $|f(\tilde{x}) - f(x)| = \left|\sqrt{0.011} - \sqrt{0.010}\right| = 0.00488 = 4.88 \cdot |\tilde{x} - x|$.

Nach dem Mittelwertsatz hätten wir geschätzt $\left|\sqrt{0.011} - \sqrt{0.010}\right| \leq \max_{x_0 \in [0.010, 0.011]} \frac{1}{2\sqrt{x_0}} = 5$.

Beispiel 3: Für Polynome der Form $f(x) = ax^b$ mit $f\prime(x) = abx^{b-1}$ ist die Konditionszahl

$$\frac{\left|abx^{b-1}\right|\,|x|}{|ax^b|} = \frac{\left|abx^b\right|}{|ax^b|} = b \tag{4}$$

gleich der Ordnung des Polynoms.

2 Nullstellensuche

Das Auffinden von Nullstellen x_0 einer stetigen Funktion $f(x)$ mit $f(x_0) = 0$ ist ein häufig auftretendes Problem im Rechenalltag, da sich viele Fragestellungen in Nullstellenprobleme umformen lassen. Beipielsweise kann der Schnittpunkt einer Parabel mit einer Geraden $ax^2 + b = cx + d$ als Nullstellenproblem $ax^2 - cx + b - d = 0$ umgeschrieben werden. Oder auch die Suche nach einem Minimum (Maximum) läuft über das Auffinden von Nullstellen der Ableitungsfunktion.

Diewichtigsten Fragen zur Nullstellensuche sind

- Gibt es Nullstellen?
- Wieviele Nullstellen gibt es?
- Wo (bei welchen Werten der unabhängigen Variablen) liegen die Nullstellen?

[3] Im Gegensatz zur Abschätzung liefert uns eine Schätzung nur eine grobe Idee, die bei sehr falschem $\tilde{x}$ auch noch komplett falsch sein kann, aber keine obere oder untere Schranke

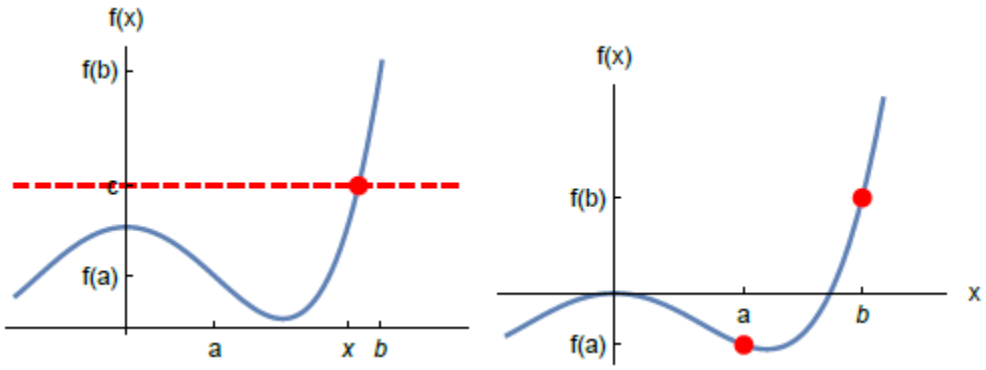

Abbildung 2: Zwischenwertsatz, links: beliebige Funktion mit $f(a) \leq c \leq f(b)$, rechts: Funktion mit $f(a) f(b) < 0$ und Nullstelle im Intervall $[a, b]$.

2.1 Zwischenwertsatz

Der Zwischenwertsatz gibt uns einen wichtigen Hinweis zum Nullstellenproblem, genauer zur ersten Frage:

Sei $f : [a, b] \to \mathbb{R}$eine stetige Funktion und gilt für $c \in \mathbb{R}$, dass $f(a) \leq c \leq f(b)$ oder $f(b) \leq c \leq f(a)$, dann gibt es ein $x \in [a, b]$, so dass $f(x) = c$.

Betrachten wir $c = 0$, also den Fall, dass $f(x) = 0$ bei x eine Nullstelle hat. Der Zwischenwertsatz sagt, dass dies im Intervall $[a, b]$ in der Tat der Fall ist, wenn entweder $f(a) \geq 0$ und $f(b) \leq 0$ oder $f(b) \geq 0$ und $f(a) \leq 0$. Mit anderen Worten, wenn von einen Intervallende zum Andern (mindestens) ein Vorzeichenwechsel stattfinden (oder der Funktionswert beide Male Null ist), gibt es (mindestens) eine Nullstelle in diesem Intervall.

Genau diese Eigenschaft machen wir uns jetzt zunutze im Intervallhalbierungs- oder Bisektionsverfahren.

2.2 Bisektionsverfahren

Beim Bisektionsverfahren beginnt man mit zwei Startpunkten a und b, die so gewählt werden, dass $f(a) f(b) < 0$, also im Intervall $[a, b]$ ein Vorzeichenwechsel erfolgt. Damit liegt in diesem Intervall nach dem Zwischenwertsatz mindestens eine Nullstelle. Um diese Nullstelle iterativ zu bestimmen geht, man folgendermaßen vor:

- Bestimme Funktionswerte $f(a)$, $f(b)$
- Bestimme Funktionswert in der Mitte des Intervalls (am Mittelwert von a und b): $f\left(a + \frac{b-a}{2}\right) = f\left(\frac{a+b}{2}\right)$

Wenn jetzt nicht gerade einer der Funktionswerte schon Null ist, gibt es zwei Möglichkeiten, wo die gesuchte Nullstelle x mit $f(x) = 0$ liegt, nämlich $x \in \left[a, \frac{a+b}{2}\right]$ oder $x \in \left[\frac{a+b}{2}, b\right]$, also in der einen oder der anderen Intervallhälfte. Um zu entscheiden in welcher, bedienen wir uns wieder des Zwischenwertsatzes. Wenn nämlich $x \in \left[a, \frac{a+b}{2}\right]$, so muss sich das Vorzeichen von $f(a)$ vom Vorzeichen von $f\left(\frac{a+b}{2}\right)$ unterscheiden, anderfalls findet der Vorzeichenwechsel zwischen $f\left(\frac{a+b}{2}\right)$ und $f(b)$ statt. Entsprechend setzen wir unsere Suche in der Intervallhälfte fort, in welcher der Vorzeichenwechsel, also die Nullstelle liegt. Mit angepassten Intervallgrenzen (die Indices $n+1$, $n+2$, etc. bezeichnen hier den jeweils nächsten Iterationsschritt)

- Wenn $f\left(\frac{a+b}{2}\right) \cdot f(a) \leq 0$, dann $b_{n+1} = \frac{a+b}{2}$, sonst $a_{n+1} = \frac{a+b}{2}$

wird die Suche fortgesetzt mit

- Bestimme $f\left(\frac{a_{n+1}+b_{n+1}}{2}\right)$

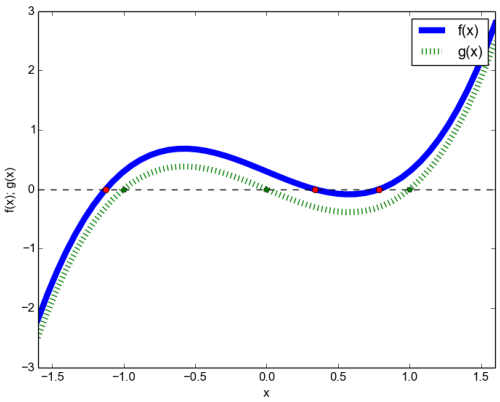

Abbildung 3: Funktion $f(x) = x^3 - x + 0.3$ (blau) und Hifsfunktion $g(x) = x^3 - x$ (grün gestrichelt). Die Nullstellen sind jeweils rot und grün markiert.

- Wenn $f\left(\frac{a_{n+1}+b_{n+1}}{2}\right) \cdot f(a_{n+1}) \leq 0$, dann $b_{n+2} = \frac{a_{n+1}+b_{n+1}}{2}$, sonst $a_{n+2} = \frac{a_{n+1}+b_{n+1}}{2}$

Solange bis entweder einer der evaluierten Funktionswerte gleich Null ist oder die Intervallbreite einen Schwellenwert erreicht $|a - b| < \epsilon_{ab}$ (oder ein Maximum an Iterationsschritten durchgeführt wurde.)

Vorteile: robust, nur Funktionsauswertung, keine Ableitung nötig

Nachteil: viele Iterationsschritte, braucht geeignete Startwerte (können auch durch Intervallhalbierung gesucht werden)

Beispiel soll die Funktion

$$f(x) = x^3 - x + 0.3$$

sein. Um Startwerte zu erraten schauen wir die Funktion $g(x) = x^3 - x$ an. Diese ist ein Polynom dritten Grades, hat also mindestens eine Nullstelle. Die analytischen Nullstellen liegen bei $x = 0$ und $x = \pm 1$. Das nutzen wir aus und betrachten diese als Startwerte, außerdem noch -2 und ± 0.5

x	-2	-1	-0.5	0.0	0.5	1
$f(x)$	-5.7	0.3	0.675	0.3	-0.075	0.3

Zwischen den jeweiligen x-Werten muss eine Nullstelle liegen, wenn sich das Vorzeichen von $f(x)$ jedes Mal ändert. Wir nehmen also an, dass es drei Nullstellen gibt (was ja auch stimmt), eine zwischen -2 und 1, eine weitere zwischen 0 und 0.5 und die dritte zwischen 0.5 und 1. Betrachten wir zunächst das Intervall zwischen $x = 0$ und $x = 0.5$, hierbei soll a jeweils die untere und b die obere Intervallgrenze bezeichnen, $\left(\frac{a+b}{2}\right)$ ist dann die Intervallmitte.

	$f(a)$	$f(b)$	$f\left(\frac{a+b}{2}\right)$
n=0	$f(0) = 0.3 > 0$	$f(0.5) = -0.075 < 0$	$f(0.25) = 0.065625 > 0$
n=1	$f(0.25) = 0.065625 > 0$	$f(0.5) = -0.075 < 0$	$f(0.375) = -0.022265625 < 0$
n=2	$f(0.25) = 0.065625 > 0$	$f(0.375) = -0.022265625 < 0$	$f(0.3125) = 0.018017578 > 0$
n=3	$f(0.3125) = 0.018017578 > 0$	$f(0.375) = -0.022265625 < 0$	$f(0.34375) = -0.003131104 < 0$

2.3 Newtonverfahren

Beim Newtonverfahren wird die Funktion durch eine Taylorreihe um einen Startwert x_0 angenähert

$$f(x) \approx f(x_0) + (x - x_0) f\prime(x_0) \stackrel{!}{=} 0 \tag{5}$$

wobei $f\prime(x_0) \neq 0$. Durch Umstellen erhalten wir

$$x = x_0 - \frac{f(x_0)}{f\prime(x_0)}$$

und daraus die Iterationsvorschrift

$$x_{n+1} = x_n - \frac{f(x_n)}{f\prime(x_n)} \tag{6}$$

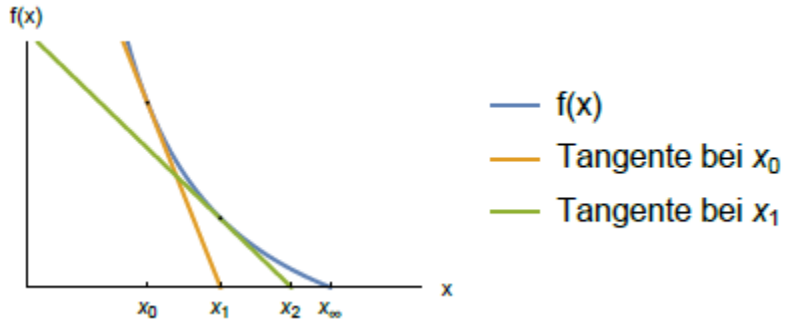

Abbildung 4: Die ersten beiden Schritte des Newtonverfahrens.

Für das obige Beispiel

$$f(x) = x^3 - x + 0.3$$

haben wir also

$$f\prime(x) = 3x^2 - 1$$

und mit der Iterationsvorschrift

$$x_{n+1} = x_n - \frac{x_n^3 - x_n + 0.3}{3x_n^2 - 1}$$

als erste Schritte

n	1. Nullstelle x_n	2. Nullstelle x_n	3. Nullstelle x_n
0	0.000	1.000	-1.0000
1	0.3000	0.8500	-1.1500
2	0.3370	0.7950	-1.1260
3	0.3389	0.7867	-1.1254
...	...	...	...
∞	0.338936241595	0.786482541162	-1.12541878276

Vorteil: Einfach und schnell,d.h. konvergiert in wenigen Schritten

Nachteil: Erfolg ist stark abhängig vom Startwert x_0. In jedem Schritt muss die Ableitung berechnet werden (analytisch oder numerisch), vereinfacht, aber langsamer kann mit konstantem Ableitungswert $f\prime(x_0)$ gearbeitet werden.

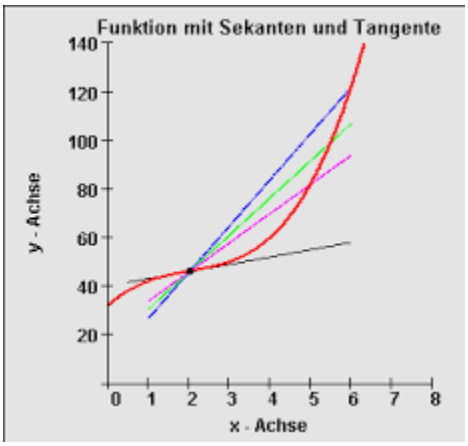

Abbildung 5: Funktion mit Sekanten und Tangente (mathenexus.zum.de)

2.3.1 Sekantenverfahren

Um die Berechnung der Ableitung $f\prime(x_n)$ in jedem Schritt zu vermeiden, benutzt man als Näherung die Sekante durch die Punkte $f(x_n)$ und $f(x_{n-1})$. Zur Erinnerung, die Steigung der Sekante $m_s = \frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$ nähert sich für $x_n \to x_{n-1}$ der Steigung der Tangente (also der Ableitung) an.

Die Iterationsvorschrift mit Näherung durch die Sekante ist dann

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}} \\ &= x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \end{aligned} \tag{7}$$

Die Konvergenz ist allerdings langsamer als im klassischen Newtonverfahren.

2.4 Fixpunktiteration

Eine Gleichung der Form $f(x) = x$ heißt *Fixpunktgleichung*. Ihre Lösungen, also die $x^* = f(x^*)$, heißen *Fixpunkte*.

Gegeben sei eine Funktion $f : [a, b] \to \mathbb{R}$ sowie ein $x_0 \in [a, b]$. Die rekursive Folge $x_{n+1} = f(x_n)$ mit $n = 0, 1, \ldots$ heißt *Fixpunktiteration* zum Startwert x_0.

Mit solch einer Iteration lassen sich also ggf. Fixpunkte finden. Was nützt uns das für die Nullstellensuche?

Anstelle der Suche nach der Nullstelle mit $f(x) = x^3 - x + 0.3 = 0$ kann man auch das Problem umformen als $\phi(x) = x^3 + 0.3 = x$. Gesucht ist also dann das x , für welches $\phi(x)$ die Ursprungsgerade $y = x$ schneidet.

Wir versuchen uns also an der Itration

$$x_{n+1} = \phi(x_n) = x^3 + 0.3$$

zu verschiedenen Startwerten

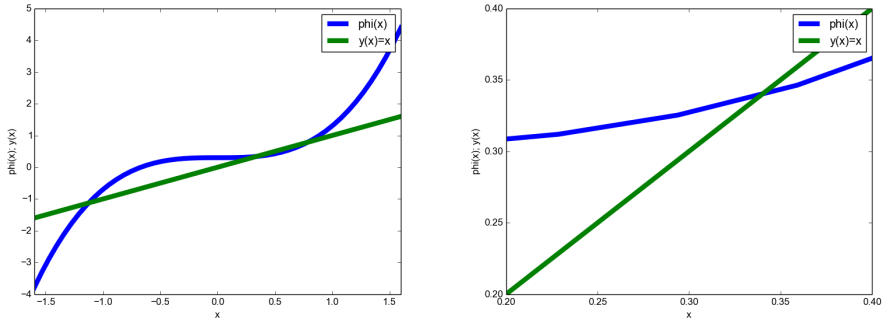

Abbildung 6: Fixpunktabbildung $\phi(x) = x^3 + 0.3 = x$. Rechts: Ausschnitt in der Nähe des anziehenden Fixpunkts.

n	x_n	x_n	x_n
0	-1.0	0.0	1.0
1	-0.7	0.3	1.3
2	-0.043	0.327	2.497
3	0.29992	0.33497	15.868
4	0.32698	0.33758	3996.37
5	0.33496	...	...
	...	...	...
∞	0.3389	0.3389	...

Offenbar konvergieren die Fixpunktiterationen mit den Startwerten $x_0 = 0 - 1$ und $x_0 = 0$, und zwar zu demselben Wert, die Iteration mit dem Startwert $x_0 = 1$ aber konvergiert nicht. Auch mit anderen Startwerten erhält man für diese Fixpunktiteration entweder 0.3389... oder die Folge divergiert. Dieser Fixpunkt ist also anziehend, während die anderen Fixpunkte abstoßend sind und daher mit dieser Iteration nicht angenähert werden können.

Betrachten wir Abbildung 6 und überlegen uns außerdem, dass der gesuchte Fixpunkt der Schnittpunkt mit der Ursprungsgeraden ist, erkennen wir, dass eine Konvergenz nur für den Fall möglich ist, dass $\phi(x)$ in der nähe des Fixpunkts weniger schnell ansteigt als die Ursprungsgerade und vermutlich auch umso schneller, je geringer die Steigung. Daraus lässt sich folgender Satz über Fixpunkte formulieren:

Sei $\phi : [a, b] \to \mathbb{R}$mit stetiger Ableitung $\phi\prime$ und $\bar{x} \in [a, b]$ ein Fixpunkt von ϕ. Dann gilt für die Fispunktiteration $x_{n+1} = \phi(x_n)$:

- Ist $|\phi\prime(\bar{x})| < 1$, so konvergiert x_n gegen $\bar{x}$, falls der Startwert x_0 nahe genug bei $\bar{x}$ liegt. Der Punkt $\bar{x}$ heißt dann *anziehender Fixpunkt*.
- Ist $|\phi\prime(\bar{x})| > 1$, so konvergiert x_n für keinen Startwert $x_0 \neq \bar{x}$. Der Punkt $\bar{x}$ heißt dann *abstoßender Fixpunkt*.

Konvergenzbedingungen und Fehlerabschätzung für Fixpunktiterationen werden präzisiert im

Banachscher Fixpunktsatz:
Sei $\phi : [a,b] \to [a,b]$ (d.h. ϕ bildet das Inervall $[a,b]$ in sich ab), und es existiere eine Konstante $\alpha < 1$ mit

$$|\phi(x) - \phi(y)| \leq \alpha |x - y|$$

für alle $x, y \in [a,b]$ (ϕ ist „kontraktiv“).
Dann gilt:

- ϕ hat genau einen Fixpunkt $\bar{x}$ in $[a,b]$.
- Die Fixpunktiteration $x_{n+1} = \phi(x_n)$ konvergiert gegen $\bar{x}$ für alle Startwerte $x_0 \in [a,b]$.
- Es gelten die Fehlerabschätzungen

$$|x_n - \bar{x}| \leq \frac{\alpha^n}{1-\alpha} |x_1 - x_0| \text{ a-priori Abschätzung}$$

$$|x_n - \bar{x}| \leq \frac{\alpha}{1-\alpha} |x_n - x_{n-1}| \text{ a-posteriori Abschätzung}$$

Die Schwierigkeit liegt darin, ein Intervall zu finden, das durch ϕ in sich abgebildet wird.

Ein weiteres Beipiel soll uns zeigen, dass auch die Wahl der Fixpunktiteration über den Erfolg entscheidet. Gesucht sind die Nullstellen der Funktion

$$f(x) = 2 - x^2 - \exp^x$$

Dazu können wir zwei Fixpunktgleichungen schreiben. Aus $0 = 2 - x^2 - \exp^x$ wird entweder $x^2 = 2 - \exp^x$ und damit

$$\phi_1(x) = \sqrt{2 - \exp(x)}$$

oder $-\exp^x = 2 - x^2$ und damit

$$\phi_2(x) = \ln\left(2 - x^2\right)$$

. Mit dem Startwert $x_0 = 0.5$ ergeben sich folgende Iterationen

n	$x_{n+1} = \phi_1$	$x_{n+1} = \phi_2$
0	0.5927	0.5596
1	0.4372	0.5229
2	0.6720	0.5463
3	0.2044	0.5316
4	0.8793	0.5408
	$2 - \exp(0.88) < 0$	0.5351
		...
∞		0.5373

Die Iteration ϕ_2 führt also zum Erfolg, ϕ_1 nicht. Im Nachhinein hätten wir es schon vermuten können, denn $\phi_1\prime = \frac{-\exp(x)}{2\sqrt{2-\exp(x)}}$ an der Stelle $\bar{x} = 0.537$ ist $|\phi\prime_1(\bar{x})| \approx 1.59 > 1$. Dahingegen ist $\phi\prime_2 = -\frac{2x}{2-x^2} = \frac{2x}{x^2-2}$ am Fixpunkt $|\phi\prime_2(0.537)| \approx 0.63 < 1$.

3 Gleichungssysteme und Matrizen

3.1 Lineare Gleichungssysteme - direkte Verfahren

Ziel ist es, eine relativ große Anzahl von Gleichungen mit einer Anzahl Unbekannten zu lösen. Für eine eindeutige Lösung muss die Anzahl der Gleichungen der der Unbekannten entsprechen.

Ein Gleichungssystem

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

in dem die a_{ij} und b_i bekannt und x_i gesucht sind lässt sich auch in Matrixform schreiben als

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ & & \ldots & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

wobei $\mathbf{A} = (a_{ij})$ eine $n \times n$ Matrix ist und $\mathbf{b}$ und $\mathbf{x}$ n-dimensionale Vektoren.

Solche Probleme lassen sich numerisch lösen durch *direkte Verfahren*. Diese liefern in endlich vielen Rechenschritten eine „exakte“ Lösung (soweit die fehlerbehafteten numerischen Rechnungen exakt sein können). *Iterative Verfahren* hingegen erzeugen eine Folge von Vektoren, die gegen die Lösung des Gleichungssystems konvergiert.

Wir betrachten beide Sorten und beginnen mit einem direkten Verfahren.

3.1.1 Gauß-Algorithmus

Beispiel Wir wollen das Gleichungssystem

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -14 & -16 \\ 0 & 0 & -3 \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

lösen. Das können wir schrittweise durch sog. Rückwärtseinsetzen, angefangen mit der letzten Gleichung die nur eine Unbekannt enthält. Deren Lösung setzten wir in die vorletzte Gleichung ein, etc.

1. $0 \cdot x_1 + 0 \cdot x_2 - 3 \cdot x_3 = 3$ woraus wird $x_3 = -1$
2. $0 \cdot x_1 + (-14) \cdot x_2 - 16 \cdot (-1) = 2$, also $-14 \cdot x_2 = -14$ liefert uns $x_2 = 1$
3. $1 \cdot x_1 + 2 \cdot 1 + 3 \cdot (-1) = 1$ ergibt $x_1 = 2$.

Allgemein kann man also formulieren, falls $\mathbf{A}$ eine Rechts-obere Dreiecksmatrix ist, d.h. $\mathbf{A}$ enthält unterhalb der Diagonalen nur Nullen

$$x_n = \frac{b_n}{a_{nn}} \tag{8}$$

sofern $a_{nn} \neq 0$. Andernfalls werden einfach Zeilen des gesamten Gleichungssystems vertauscht. Wenn alle $a_{in} = 0$ ist die Matrix nicht regulär und das Gleichungssystem nicht lösbar.

Für $i = n-1, n-2, \ldots, 1$ erhält man ferner

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \tag{9}$$

Für Links-untere Dreiecksmatritzen kann man eine entsprechende Vorschrift für „Vorwärtseinsetzen" formulieren.

Der Rechenaufwand sind $\frac{n(n+1)}{2}$ Punktoperationen, d.h. soviele Punktoperationen wie nicht-Nullelemente in der Matrix. Genauer sind das j viele Multiplikaitonen $a_{ij}x_j$ und eine Division $\frac{1}{a_{ii}}$.

Erzeugen der Dreiecksmatrix erfolgt mittels folgender Umformungen:

- Vertauschen zweier Zeilen $z_i \leftrightarrow z_j$ des Gleichungssystems, d.h. inklusive der rechten Seite
- Subtraktion eines vielfachen λ einer Zeile von einer darunter stehenden Zeile $z_j^{neu} = z_j^{alt} - \lambda z_i$ mit $i < j$ und $\lambda \in \mathbb{R}$.

In einem Algorithmus wird nun das Erzeugen der Nullen gemäß der erlaubten Umformungen von oben nach unten durchgeführt. Wir beginnen mit Nullen in der 1. Spalte und zweiten Zeile, also unterhalb von a_{11}:

$$z_2^{neu} = z_2^{alt} - \frac{a_{21}}{a_{11}} z_1$$

und so weiter für alle $j = 2, \ldots, n$

$$z_j^{neu} = z_j^{alt} - \frac{a_{j1}}{a_{11}} z_1$$

Für die Nullen in der zweiten Spalte (jetzt erst ab dritter Zeile) dann entsprechend

$$z_j^{neu} = z_j^{alt} - \frac{a_{j2}}{a_{22}} z_2$$

Schauen wir uns ein Beispiel an

$$\begin{array}{ccc} & \mathbf{Ax} & = \quad \mathbf{b} \\ \begin{pmatrix} 1 & 2 & 3 \\ 6 & -2 & 2 \\ -3 & 1 & -4 \end{pmatrix} & \cdot \mathbf{x} = & \begin{pmatrix} 1 \\ 8 \\ -1 \end{pmatrix} \end{array}$$

Um Nullen in Spalte 1 und Zeile 2 zu erzeugen, rechnen wir $z_2^{neu} = z_2^{alt} - \frac{a_{21}}{a_{11}} z_1 = z_2^{alt} - \frac{6}{1} z_1$. Damit bekommen wir

$$\begin{array}{ccc} & \mathbf{A}_1\mathbf{x} & = \quad \mathbf{b}_1 \\ \begin{pmatrix} 1 & 2 & 3 \\ 0 & -14 & -16 \\ -3 & 1 & -4 \end{pmatrix} & \cdot \mathbf{x} = & \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} \end{array}$$

Und in der dritten Zeile mit $z_3^{neu} = z_3^{alt} - \frac{a_{31}}{a_{11}} z_1 = z_3^{alt} + 3z_1$ dann

$$\begin{array}{ccc} & \mathbf{A}_2\mathbf{x} & = \quad \mathbf{b} \\ \begin{pmatrix} 1 & 2 & 3 \\ 0 & -14 & -16 \\ 0 & 7 & 5 \end{pmatrix} & \cdot \mathbf{x} = & \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \end{array}$$

Und schließlich noch für die 2. Spalte in der dritten Zeile mit $z_3^{neu} = z_3^{alt} - \frac{a_{32}}{a_{22}} z_2 = z_3^{alt} + \frac{-7}{-14} z_2 = z_3^{alt} + \frac{1}{2} z_2$ ergibt sich

$$\begin{array}{ccc} & \mathbf{A}_3\mathbf{x} & = & \mathbf{b}_3 \\ \begin{pmatrix} 1 & 2 & 3 \\ 0 & -14 & -16 \\ 0 & 0 & -3 \end{pmatrix} & \cdot\mathbf{x} = & \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \end{array}$$

und das haben wir schon oben gelöst :-)

Was tun wir aber nun, wenn wir ein ganz ähnliches Gleichungssystem lösen wollen, bei dem nur die rechte Seite[4] verschieden ist $\mathbf{Ay} = \mathbf{c}$, die Matrix $\mathbf{A}$ aber die Gleiche? Wir wissen ja schon, welche Umformungen gemacht werden müssen und sie nur noch auf den neuen Vektor $\mathbf{c}$ anwenden. Also brauchen wir eine geeignete Form, die Umformungsvorschriften „Subtrahiere von Zeile j das λ-fache von Zeile i... " abzuspeichern.

Bei genauer Betrachtung erkannt man, dass diese Umformungen sich durch eine Matrixmultiplikation ausdrücken lassen. Zu Erinnerung, die Matrixelemente m_{ij} der Ergebnismatrix $\mathbf{M} = \mathbf{AB}$ bestimmen sich aus $m_{ij} = \sum_k a_{ik} b_{kj}$.

Suchen wir also die Matrix $\mathbf{G_1}$,so dass $\mathbf{G}_1\mathbf{A} = \mathbf{A}_1$ also hier Nullen in der ersten Spalte der zweiten Zeile erzeugt. Da die erste und letzte Zeile der Matirx $\mathbf{A}$ zu diesem Zeitpunkt unverändert bleiben soll, wissen wir schon

$$\mathbf{G}_1 = \begin{pmatrix} 1 & 0 & 0 \\ g_{21} & g_{22} & g_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

Aus der Vorschrift für Matrixmultiplikationen wissen wir, dass

$$a_{21}^1 = g_{21}a_{11} + g_{22}a_{21} + g_{23}a_{31} = 0$$

Nun wollen wir die zweite und dritte Spalte noch unverändert lassen, wählen also $g_{22} = 1$ und $g_{23} = 0$, müssen also nur noch g_{21} bestimmen zu $g_{21} = -\frac{a_{21}}{a_{11}}$, ein Faktor, der uns schon recht bekannt vorkommt.

Unsere Umformungsmatritzen für unser Beispiel sind also

$$\mathbf{G_1} = \begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; \mathbf{G_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}; \text{ und } \mathbf{G_3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix}.$$

Da wir alle nacheinander auf die ursprüngliche Matrix anwenden, wissen wir

$$\mathbf{GA} = \mathbf{R}$$

wobei $\mathbf{R}$ die rechts-obere Dreiecksmatrix ist und $\mathbf{G} = \mathbf{G}_3\mathbf{G}_2\mathbf{G}_1$.

Für eine neues Problem wie

$$\mathbf{Ay} = \mathbf{c}$$

wenden wir also $\mathbf{G}$ an und bekommen außer

$$\mathbf{GA} = \mathbf{R}$$

auch

$$\mathbf{u} = \mathbf{Gc}$$

womit wir dann dann

$$\mathbf{Ry} = \mathbf{u}$$

lösen können.

[4] Der Lösungsvektor ist natürlich i.A. auch verschieden, deswegen nennen wir ihn hier $\mathbf{y}$.

3.1.2 Spaltenpivotisierung

Beim Eliminationsschritt im Gauß-Algorithmus

$$z_j^{neu} = z_j^{alt} - \frac{a_{ji}}{a_{ii}} z_i$$

kann es zu Fehlerverstärkung von z.B. Rundungsfehlern bei der Multiplikation mit $\frac{a_{ji}}{a_{ii}}$ kommen. Die Fehlerverstärkung ist umso größer je größer $\frac{a_{ji}}{a_{ii}}$ ist. Deswegen ist es günstig als Element a_{ii} im i-ten Schritt eine betragsmäßig möglichst froße Zahl zu haben. Dies kann durch Zeilenvertauschung erreicht werden, die so erfolgt, dass die i-te Zeile mit der darunter liegenden Zeile vertauscht wird, deren Element das maximale der i-ten Spalte ist. Die Vertauschung ändert das Gleichungssystem nicht, und die Vertauschung nur mit darunter liegenden Zeilen stellt sicher, dass einmal erzeugte Nullen erhalten bleiben. Bei betragsmäßig ähnlich großen Zahlen hat die Pivortisierung allerdings kaum Auswirkungen auf dem Fehler.

Anders sieht es aus bei dem folgenden konstruierten

Beipiel für Pivotisierung: Das Gleichungssystem

$$\begin{pmatrix} -10^{-4} & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

soll einmal ohne und einmal mit Zeilenvertauschung in vierstelligr Genauigkeit gelöst werden. Ohne Pivotiserung haben wir im Eliminationsschritt

$$\begin{aligned} z_2^{neu} &= z_2^{alt} - \frac{a_{21}}{a_{11}} z_1 \\ &= z_2^{alt} - \frac{2}{10^{-4}} z_1 \end{aligned}$$

Das umgeformte Gleichungssystem lautet also

$$\begin{pmatrix} -10^{-4} & 1 \\ 0 & 20000 + 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 20000 \end{pmatrix}$$

bzw. in vierstelliger Genauigkeit

$$\begin{pmatrix} -10^{-4} & 1 \\ 0 & 20000 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 20000 \end{pmatrix}$$

Durch Rückwärtseinsetzen ergibt sich

$$\begin{aligned} x_2 &= \frac{20000}{20000} = 1 \\ x_1 &= \frac{1}{-10^{-4}} (1 - 1) = 0 \end{aligned}$$

Machen wir die Probe

$$\begin{pmatrix} -10^{-4} & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

erkennen wir ein Problem.

Mit Zeilenvertauschung haben wir dagegen aus

$$\begin{pmatrix} 2 & 1 \\ -10^{-4} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{aligned} z_2^{neu} &= z_2^{alt} - \frac{a_{21}}{a_{11}} z_1 \\ &= z_2^{alt} - \frac{10^{-4}}{2} z_1 \end{aligned}$$

und damit

$$\begin{pmatrix} 2 & 1 \\ 0 & 5 \cdot 10^{-5} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \cdot 10^{-5} \end{pmatrix}$$

und dann durch Rückwärtseinsetzen

$$\begin{aligned} x_2 &= \frac{5 \cdot 10^{-5}}{5 \cdot 10^{-5}} = 1 \\ x_1 &= \frac{1}{2} - 1 = -\frac{1}{2} \end{aligned}$$

Die Probe

$$\begin{pmatrix} 2 & 1 \\ -10^{-4} & 1 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 - 5 \cdot 10^{-5} \end{pmatrix} \approx \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

zeigt uns, dass der komponentenweise Fehler nur noch $5 \cdot 10^{-5}$ ist.

3.1.3 Dreieckszerlegung/LR-Zerlegung/LU-Zerlegung

Die Umformungsvorschrift, welche eine Dreiecksmatrix nach

$$\mathbf{GA} = \mathbf{R}$$

erzeugt, hat noch eine weitere Bedeutung. Wir mutiplizieren von links mit ihrer Inversen

$$\begin{aligned} \mathbf{G^{-1}GA} &= \mathbf{G^{-1}R} \\ \mathbf{A} &= \mathbf{G^{-1}R} \end{aligned}$$

Dies ist eine Zerlegung der Matrix $\mathbf{A}$ in eine links-untere und eine rechts-obere Dreiecksmatrix. Benennen wir daher um $\mathbf{G}^{-1} = \mathbf{L}$ und schreiben

$$\mathbf{A} = \mathbf{LR}$$

In unserem Fall ist die Inverse $\mathbf{G}^{-1} = \mathbf{L}$ leicht zu erzeugen. Handelt es sich wie bei den Umformungsmatrizen $\mathbf{G}_n$ um eine eine sogennante Frobeniusmatrix, also eine Matrix, die sich nur in einer Spalte von der Einheitsmatrix unterscheidet, so erhält man die Inverse duch Vorzeichenwechsel der Unterdiagonalelemente.

Beispiel: $\mathbf{G_1} = \begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ und $\mathbf{G}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

also ist

$$\begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1+0+0 & 0 & 0 \\ -6+6+0 & 1 & 0 \\ 0+0+0 & 0+0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Aber Vorsicht beim Gegenbeispiel:$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}$ und $\mathbf{L}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 6 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix}$, die in zwei Spalten von der Einheitsmatrix verschieden sind. Dort ist $\begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 6 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} = \begin{pmatrix} 1+0+0 & 0 & 0 \\ -6+6+0 & 1 & 0 \\ 0+3+0 & \frac{1}{2}+\frac{1}{2} & 1 \end{pmatrix}$

Sinnvollerweise invertieren wir zuerst die einzelnen Matrizen durch Vorzeichenwechsel und multiplizieren dann, denn

$$\begin{aligned}\mathbf{A} &= \mathbf{LR}\\ \mathbf{A} &= \mathbf{L}_1\mathbf{L}_2\mathbf{L}_3\mathbf{R}\\ \mathbf{A} &= \mathbf{G}_1^{-1}\mathbf{G}_2^{-1}\mathbf{G}_3^{-1}\mathbf{R}\end{aligned}$$

wegen

$$\begin{aligned}\mathbf{GA} &= \mathbf{R}\\ \mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{A} &= \mathbf{R}\\ \mathbf{G}_2\mathbf{G}_1\mathbf{A} &= \mathbf{G}_3^{-1}\mathbf{R}\\ \mathbf{G}_1\mathbf{A} &= \mathbf{G}_2^{-1}\mathbf{G}_3^{-1}\mathbf{R}\\ \mathbf{A} &= \mathbf{G}_1^{-1}\mathbf{G}_2^{-1}\mathbf{G}_3^{-1}\mathbf{R}\end{aligned}$$

Hieraus erkennt man dann auch, dass $\mathbf{L}$ eine links-untere Dreiecksmatrix ist. Die Umformungsmatritzen $\mathbf{G}_n$ sind links-untere Dreiecksmatrizen (und Frobeniusmatrizen), ihre Inversen sind auch links-untere Dreiecksmatrizen und deren Produkt ist dann entsprechend auch eine links-untere Dreiecksmatrix.

3.1.4 Cholesky-Zerlegung

Für spezielle Matrizen gibt es besonders effiziente Dreieckszerlegungen. Eine davon ist die Cholesky-Zerlegung für symmetrische, positiv definite Matrizen. D.h. für die symmetrische $n \times n$ Matrix $\mathbf{A}$ gilt für alle $\mathbf{x} \in \mathbb{R}$, $\mathbf{x} \neq 0$ $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ (positiv definit bedeutet auch dass alle Eigenwerte positiv sind).

Es gibt für jede dieser positiv definiten $n \times n$ Matrizen $\mathbf{A}$ genau eine rechts-obere Dreiecksmatrix $\mathbf{R}$mit $r_{ii} > 0$ für $i = 1, \ldots n$ und der *Cholesky-Zerlegung* $\mathbf{A} = \mathbf{R}^T\mathbf{R}$.

Um eine Idee zu entwickeln, wie wir die Dreiecksmatrix erzeugen können, erinnern wir uns noch einmal an Matrixmultiplikation (Nullen sind einfach durch leere Stellen repräsentiert):

$$\mathbf{R}^T\mathbf{R} = \mathbf{A}$$

$$\begin{pmatrix} r_{11} & & & \\ r_{21} & r_{22} & & \\ \vdots & \cdots & \ddots & \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Das erste Element ist

$$a_{11} = \sum_k r_{1k}r_{k1} = r_{11}^2$$

wegen der Nullen in den Elementen r_{j1} und r_{1j} für $j \neq 1$. Also

$$r_{11} = \sqrt{a_{11}}$$

Sollte $a_{11} < 0$ kann hier schon abgebrochen werden, da die Matrix nicht positiv definit ist.

Das zweite Element

$$a_{12} = \sum_k r_{1k}r_{k2} = r_{11}r_{12}$$

liefert uns wir dann

$$r_{12} = \frac{a_{12}}{r_{11}} \ .$$

Entsprechend erhalten wir alle Elemente der ersten Zeile mit

$$r_{1j} = \frac{a_{1j}}{r_{11}}.$$

In der zweiten Zeile haben wir

$$a_{21} = \sum_k r_{2k} r_{k1} = r_{21} r_{11}$$

und damit natürlich aus Symmetriegründen $r_{21} = \frac{a_{21}}{r_{11}}$.

Das nächste Diagonalelement bestimmen wir aus

$$a_{22} = \sum_k r_{2k} r_{k2} = r_{21} r_{12} + r_{22} r_{22} = r_{12}^2 + r_{22}^2$$

zu

$$r_{22} = \sqrt{a_{22} - r_{12}^2}.$$

Und als letztes errechnen wir noch mit

$$a_{23} = \sum_k r_{2k} r_{k3} = r_{21} r_{13} + r_{22} r_{23}$$

also

$$r_{23} = \frac{a_{23} - r_{21} r_{13}}{r_{22}}.$$

Die Berechnung läuft also folgendermaßen, wobei wir eine Hilfsvariable S einführen:

Gegeben sei eine symmetrische $n \times n$ Matrix $\mathbf{A}$.
Für $i = 1, \ldots n$ berechne

- $S = a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2$ d.h. für $i = 1$ ist $S = a_{ii}$
- falls $S \leq 0$, Abbruch: $\mathbf{A}$ ist nicht positiv definit
- falls $S > 0$,
  - setze $r_{ii} = \sqrt{S}$
  - für $j = i+1, \ldots, n$ setze $r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ik} r_{kj} \right)$

Wir schauen uns ein Beispiel an: $\mathbf{A} = \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 26 \end{pmatrix}$.

Für $i = 1$ setzen wir $S = a_{ii} = 4 > 0$ also $r_{ii} = \sqrt{4}$=2.
Für $j = 2$ setzen wir $r_{12} = \frac{1}{r_{11}} \left(a_{12} - \sum_{k=1}^{0} r_{ik} r_{kj} \right) = \frac{a_{12}}{r_{11}} = \frac{4}{2} = 2$
Für $j = 3$ setzen wir $r_{13} = \frac{a_{13}}{r_{11}} = \frac{2}{2} = 1$
Für $i = 2$ setzen wir $S = a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 = a_{22} - \sum_{k=1}^{1} r_{12}^2 = 5 - 4 = 1$ also $r_{22} = \sqrt{1}$=1.
Für $j = 3$ setzen wir $r_{23} = \frac{1}{r_{22}} \left(a_{23} - \sum_{k=1}^{1} r_{21} r_{13} \right) = \frac{1}{1} (5 - 2 \cdot 1) = 3$
Für $i = 3$ setzen wir $S = a_{33} - \sum_{k=1}^{2} r_{k3}^2 = a_{33} - r_{13}^2 - r_{23}^2 = 26 - 1^2 - 3^2 = 26 - 1 - 9 = 16$ also $r_{33} = \sqrt{16}$=4.

Damit erhalten wir für die Dreiecksmatrix $\mathbf{R} = \begin{pmatrix} 2 & 2 & 1 \\ & 1 & 3 \\ & & 4 \end{pmatrix}$. Wir machen auch noch die Probe und

berechnen

$$\begin{pmatrix} 2 & & \\ 2 & 1 & \\ 1 & 3 & 4 \end{pmatrix} \begin{pmatrix} 2 & 2 & 1 \\ & 1 & 3 \\ & & 4 \end{pmatrix} = \begin{pmatrix} 4+0+0 & 4+0+0 & 2+0+0 \\ 4+0+0 & 4+1+0 & 2+3+0 \\ 2+0+0 & 2+3+0 & 1+9+16 \end{pmatrix}$$
$$= \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 2+3 \\ 2 & 2+3 & 1+9+16 \end{pmatrix}$$
$$\mathbf{A} = \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 26 \end{pmatrix} \tag{10}$$

Dreieckszerlegungen sind nützlich, denn z.B.

Determinanten von Dreiecksmatritzen sind besonders einfach zu errechnen.

$$\det \mathbf{R} = \det \begin{pmatrix} 1 & 2 & 3 \\ 0 & -14 & -16 \\ 0 & 0 & -3 \end{pmatrix}$$

Z.B. nach der „Jägerzaunmethode“, in der neben den Elementen der Matrix die erste und zweite Spalte noch einmal geschrieben werden

$$\begin{array}{ccccc} + & + & + & & \\ 1 & 2 & 3 & 1 & 2 \\ & & & & \\ 0 & -14 & -16 & 0 & -14 \\ 0 & 0 & -3 & 0 & 0 \\ & & & & \\ - & - & - & & \end{array}$$

Die Multiplikationen und Additionen/Subtraktionen sind dann in diesem Beipiel

$$+(1 \cdot (-14) \cdot (-3) + 2 \cdot (-16) \cdot 0 + 3 \cdot 0 \cdot 0) - 0 \cdot (-14) \cdot 3 - 0 \cdot (-16) \cdot 1 - (-3) \cdot 0 \cdot 2 = 42$$

woran zu erkennen ist, dass nur die Diagonalelemente beitragen, denn nur dort tauchen keine Nullen auf. Damit haben wir also

$$\det \mathbf{R} = \Pi_{i=1}^{n} r_{ii}$$

Wenn die Rechts-obere Dreiecksmatrix durch Gauss-Eliminierung erzeugt wurde und p-viele Zeilenvertauschungen durchgeführt wurden, um $\mathbf{R}$ zu erhalten, dann ist

$$\det \mathbf{A} = (-1)^p \det \mathbf{R} = (-1)^p \Pi_{i=1}^{n} r_{ii}$$

weil jede Vertauschung von Zeilen (oder Spalten) einer Determinante das Vorzeichen ändert. Die Addition des Vielfachen einer Zeile zu einer anderen aber ändert den Wert einer Determinante nicht.

Und für die Dreiecksmatritzen aus der Cholesky-Zerlegung $\mathbf{A} = \mathbf{R^T R}$ kann man schreiben

$$\mathbf{A} = \mathbf{L}^T \mathbf{D} \mathbf{L}$$

wobei sich $\mathbf{L}$ und $\mathbf{L}^T$ von $\mathbf{R}$ und $\mathbf{R}^T$ nur durch die Diagonalelemente (bei $\mathbf{L}$ „verschoben“ in die Diagonalmatrix $\mathbf{D}$) unterscheiden.

$$
\begin{aligned}
\mathbf{A} &= \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 26 \end{pmatrix} \\
\mathbf{R}^T\mathbf{R} &= \begin{pmatrix} 2 & & \\ 2 & 1 & \\ 1 & 3 & 4 \end{pmatrix} \begin{pmatrix} 2 & 2 & 1 \\ & 1 & 3 \\ & & 4 \end{pmatrix} \\
\mathbf{L}^T\mathbf{D}\mathbf{L} &= \begin{pmatrix} 1 & & \\ 2 & 1 & \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 4 & & \\ & 1 & \\ & & 16 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ & 1 & 3 \\ & & 1 \end{pmatrix}
\end{aligned}
$$

So dass die Determinante besonders leicht berechnet werden kann als

$$\det \mathbf{A} = \det\left(\mathbf{L}^T\mathbf{D}\mathbf{L}\right) = \det \mathbf{L}^T \det \mathbf{D} \det \mathbf{L} = \det \mathbf{D} = \Pi_i d_{ii}$$

wie die Probe zeigt

$$
\begin{aligned}
\det \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 26 \end{pmatrix} &= 4\,(5*26-5*5) - 4\,(4*26-5*2) + 2\,(4*5-5*2) \\
&= 64
\end{aligned}
$$

3.1.5 Fehlerbetrachtung bei Gleichungssystemen - Vektor-und Matrizennormen

Zur Erfassung und Bewertung der Fehler, die zwangsläufig bei numerischen Lösungen wegen der Rundungsfehler auftreten, brauchen wir ein Maß für die „Größe“ des Fehlers. Hierzu dienen Normen. Eine Norm ist eine Abbildung[5] $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_+$, wenn sie folgende Eingenschaften besitzt

- $\|\mathbf{x}\| > 0$, $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$
- $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$, $\alpha \in \mathbb{R}$
- $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

Die letzte Eigenschaft, „Dreiecksungleichung“, ergibt $\|\mathbf{x}-\mathbf{y}\| \geq |\|\mathbf{x}\| - \|\mathbf{y}\||$, woraus die Stetigkeit von Normen impliziert wird.

Gebräuchliche Vektor-und Matrizennormen sind

Vektor		Matrix	
$\|\mathbf{x}\|_2 := \left(\sum_{i=1}^n \|x_i\|^2\right)^{1/2}$	„Euklidische (l_2-)Norm“	$\|\mathbf{A}\|_2 := \left(\sum_{j,k=1}^n \|a_{jk}\|^2\right)^{1/2}$	„Quadratsummennorm“
$\|\mathbf{x}\|_\infty := \max_{i=1,\ldots,n} \|x_i\|$	„Maximum (l_∞)-Norm“	$\|\mathbf{A}\|_\infty := \max_{1\leq j<n} \sum_{k=1}^n \|a_{jk}\|$	„Max. Zeilensummennorm“
$\|\mathbf{x}\|_1 := \sum_{i=1}^n \|x_i\|$	l_1-Norm	$\|\mathbf{A}\|_1 := \max_{1\leq k<n} \sum_{j=1}^n \|a_{jk}\|$	„Max. Spaltensummennorm“

Obige Tabelle legt schon nahe, dass die jeweiligen Vektor-und Matrizennormen zueinander „passen“. Genauer spricht man davon, dass eine Norm $\|\cdot\|$ auf $\mathbb{R}^{n\times n}$ „verträglich“ ist mit einer Norm $\|\cdot\|$ auf $\mathbb{R}^n$, wenn gilt $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \, \|\mathbf{x}\|$.

Zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ sind äquivalent wenn $m \|\mathbf{x}\|_a \leq \|\mathbf{x}\|_b \leq M \|\mathbf{x}\|_a$.

Damit können wir jetzt den relativen Fehler der Lösung eines linearen Gleichungssystems „messen“. Sei $\tilde{\mathbf{A}} = \mathbf{A} + \delta\mathbf{A}$ eine reguläre mit dem Fehler $\delta\mathbf{A}$ gegenüber der regulären Matrix $\mathbf{A}$ behaftete Matrix und

[5] Der Raum der komplexen Zahlen $\mathbb{C}^n$, ist auch zugelassen, den schenken wir uns hier aber.

$\|\delta\mathbf{A}\| < \frac{1}{\|\mathbf{A}^{-1}\|}$, dann ist

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \underbrace{\|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\|}_{\text{cond}(\mathbf{A})} \cdot \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

der relative Fehler bezüglich der verwendeten Norm. Die Konditionszahl von $\mathbf{A}$, $\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\|$, liefert uns ein Maß wie die „Beschaffenheit der Matrix“ sich auf den Fehler der Lösung auswirkt, d.h. der Verstärkungsfaktor, mit dem sich relative Fehler in $\mathbf{A}$ und $\mathbf{b}$ auf den in $\mathbf{x}$ auswirken. Machen wir eine Abschätzung an einem

Beispiel Die Kondition einer Matrix sei $\text{cond}(\mathbf{A}) \sim 10^s$ und die Element von $\mathbf{A}$ und $\mathbf{b}$ haben Feher der Art $\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} \sim 10^{-k}$ und $\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \sim 10^{-k}$. Dann muss mit einem relativen Fehler der Größenordnung

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq 10^s \cdot 10^{-k}$$

gerechnet werden. Im Fall der Maximumnorm verliert man dann bis zu s Stellen Genauigkeit.

Eine Matrix mit Kondition $\text{cond}(\mathbf{A}) \sim 10^8$ist

$$\mathbf{A} = \begin{pmatrix} 1.2969 & 0.8646 \\ 0.2161 & 0.1441 \end{pmatrix}$$

$$\mathbf{A}^{-1} = 10^8 \begin{pmatrix} 0.1441 & -0.8646 \\ -0.2161 & 1.2969 \end{pmatrix}$$

und $\|\mathbf{A}\|_\infty = 2.1617$; $\left\|\mathbf{A}^{-1}\right\|_\infty = 1.513 \cdot 10^8$ also $\text{cond}(\mathbf{A}) \approx 3.3 \cdot 10^8$.

3.2 Lineare Gleichungssysteme - Iterative Verfahren

Das Gaußsche Eliminationsverfahren führt zwar im Prinzip zum Erfolg, ist aber für große Matrizen zu speicheraufwändig. Insbesondere bei dünn besetzten Matrizen (solche, die viele Nullen enthalten), können andere, iterative Verfahren günstiger sein.

Um eine Iterationsvorschrift zu konstruieren betrachten wir noch einmal das Gleichungssystem $\mathbf{Ax} = \mathbf{b}$ mit einer $n \times n$ Matrix $\mathbf{A}$und einem $n-$Vektor $\mathbf{b}$ komponentenweise geschrieben:

$$a_{jj}x_j + \sum_{k=1;k\neq j}^{n} a_{jk}x_k = b_j$$

für alle $j = 1, \ldots n$. Im Fall $a_{jj} \neq 0$ lässt sich das umformen in

$$x_j = \frac{1}{a_{jj}} \left(b_j - \sum_{k=1;k\neq j}^{n} a_{jk}x_k \right)$$

was uns schon aus dem Rückwärtseinsetzen (Gleichung 9) bekannt vorkommt. Wir können jetzt aber auch aus einem anderen Blickwinkel schauen und erkennen, dass man ein x_j aus einem x_k erzeugt.

Genau so etwas ist unsere Absicht, wir suchen eine Vorschrift, der Form

$$x^{n+1} = \mathbf{B}x^n + \mathbf{c} \tag{11}$$

also eine Fixpunktiteration im $n-$dimensionalen. Schließlich kann man das Gleichungssystem auch als ein Nullstellenproblem formulieren $\mathbf{Ax} - \mathbf{b} = \mathbf{0}$, allerdings mit einem Nullvektor auf der rechten Seite.

Wir erkennen auch schon, wie der Verschiebevektor $\mathbf{c}$ und die Iterationsmatrix $\mathbf{B}$ aufgebaut sind. Mit

$$c_j = \frac{b_j}{a_{jj}}$$

erhalten wir alle Komponenten des gesuchten Vektors. Und den Teil $\sum_{k=1;k\neq j}^{n} \frac{a_{jk}}{a_{jj}} x_k$ teilen wir noch einmal in $\sum_{k=1;k\neq j}^{n} a_{jk}x_k$ und $\frac{x_k}{a_{jj}}$. Das Erste ist nichts Anderes als „ $j-$Zeile $\times$ k-Spalte“, mit Ausnahme der j-ten Spalte für alle j Zeilen. Oder die Multiplikation einer Matrix mit dem Vektor $\mathbf{x}$, wobei die Matrix $\mathbf{A}$ ohne Diagonalelemente entspricht (bzw. alle diagonalelemente sind Null). Und $\frac{x_k}{a_{jj}}$ für alle j Zeilen ist der Vektor $\mathbf{x}$ „geteilt“ durch eine Matrix, also Multiplikation mit der Inversen einer Matrix, die nur die Diagonalelemente von $\mathbf{A}$ enthält.

Wir brauchen also

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{R}$$

$$\begin{pmatrix} a_{11} & a_{12} & & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} d_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{nn} \end{pmatrix} + \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \cdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} + \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}$$

eine *additive* Zerlegung in eine Diagonalmatrix und den Rest. Der Rest, die Matrix $\mathbf{A}$ ohne Diagonalelemente, ist aber eben die Summe der links-unteren und rechts-oberen Dreiecksmatrizen. Die additive Zerlegung sollte nicht mit der multiplikativen Cholesyk-Zerlegung verwechselt werden.

Die gesuchte Iterationsmatrix is also $\mathbf{B} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{R})$ und die Iterationsvorschrift des

Jacobi(gesamtschritt)verfahren

$$\mathbf{x}_{n+1} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{R})\,\mathbf{x} + \mathbf{D}^{-1}\mathbf{b}$$

Beispiel

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & 1 \\ 2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$$

Dann sind

$$\mathbf{D} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}; \mathbf{L} + \mathbf{R} = \begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & -2 & 0 \end{pmatrix}$$

und

$$\mathbf{D}^{-1} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}$$

also lautet die Iteration

$$\mathbf{x}_{n+1} = -\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}\left[\begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & -2 & 0 \end{pmatrix}\mathbf{x}_n - \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}\right]$$

mit

$$\mathbf{B} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}\begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 1 \\ 1 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{pmatrix}$$

und

$$\mathbf{D}^{-1}\mathbf{b} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}\begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix} = \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}$$

Für den Startvektor $\mathbf{x}^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ ergibt die Iteration die Folge

n	0	1	2	3	∞
$\mathbf{x}^n$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}$	$\begin{pmatrix} 1.2 \\ 2.22 \\ 3.03 \end{pmatrix}$	$\begin{pmatrix} 1.0475 \\ 2.0740 \\ 3.0480 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

Schauen wir uns die Iteration noch einmal komponentweise an

$$\begin{aligned} x_1^{n+1} &= 0x_1^n + 0.25x_2^n - 0.25x_3^n + 1.25 \\ x_2^{n+1} &= 0.4x_1^n + 0x_2^n - 0.25x_3^n + 2.2 \\ x_3^{n+1} &= -0.2x_1^n + 0.4x_2^n + 0.x_3^n + 2.4 \end{aligned}$$

Demnach können wir jede Komponente des neuen Vektors separat aus dem alten Vektor berechnen. Das kann günstig sein, um z.B. parallel zu rechnen.

Wir können aber auch für die zweite Komponente x_2 ausnutzen, dass wir schon eine Verbesserung von x_1 erzielt haben, und für x_3 entsprechend. Also setzen wir

$$\begin{aligned} x_1^{n+1} &= 0x_1^n + 0.25x_2^n - 0.25x_3^n + 1.25 \\ x_2^{n+1} &= 0.4x_1^{n+1} + 0x_2^n - 0.25x_3^n + 2.2 \\ x_3^{n+1} &= -0.2x_1^{n+1} + 0.4x_2^{n+1} + 0.x_3^n + 2.4 \end{aligned}$$

und erkennen, dass eine links-untere Dreiecksmatrix auf die neuen $n+1$ Elemente, und eine rechts-obere Dreiecksmatrix auf die alten Elemente angewandt wird, gemäß dem

Gauß-Seidel(Einzelschritt)verfahren

$$\mathbf{x}^{n+1} = -\mathbf{D}^{-1}\left(\mathbf{R}\mathbf{x}^n + \mathbf{L}\mathbf{x}^{n+1} - \mathbf{b}\right)$$

Das kann man ein wenig umformen zu

$$\begin{aligned} -\mathbf{D}\mathbf{x}^{n+1} &= \mathbf{R}\mathbf{x}^n + \mathbf{L}\mathbf{x}^{n+1} - \mathbf{b} \\ -\left(\mathbf{D}+\mathbf{L}\right)\mathbf{x}^{n+1} &= \mathbf{R}\mathbf{x}^n - \mathbf{b} \\ \mathbf{x}^{n+1} &= -\left(\mathbf{D}+\mathbf{L}\right)^{-1}\mathbf{R}\mathbf{x}^n + \left(\mathbf{D}+\mathbf{L}\right)^{-1}\mathbf{b} \end{aligned}$$

wobei die letzte Zeile die Fixpunktiteration darstellt.
Wir erhalten für das obige Beispiel

$$\mathbf{x}_{n+1} = \begin{pmatrix} 0 & 0.25 & -0.25 \\ 0 & 0 & -0.2 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{x}_n \begin{pmatrix} 0 & 0 & 0 \\ 0.4 & 0 & 0 \\ -0.2 & 0.4 & 0 \end{pmatrix}\mathbf{x}_{n+1} + \begin{pmatrix} 1.25 \\ 2.2 \\ 2.4 \end{pmatrix}$$

Die Iteration ergibt hier mit

n	0	1	2	3	∞
$\mathbf{x}_n$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1.25 \\ 2.7 \\ 3.23 \end{pmatrix}$	$\begin{pmatrix} 1.1175 \\ 2.0010 \\ 2.9769 \end{pmatrix}$	$\begin{pmatrix} 1.0060 \\ 2.0070 \\ 3.0016 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

schon im dritten Schritt eine sehr gute Näherungslösung.

Überrelaxatation /Successive Overrelaxation (SOR) nutzt die Information über die gewonnene Verbesserung im letzten Iterationsschritt über einen Zwischenwert $\tilde{\mathbf{x}}^{n+1}$, der nach Gauß-Seidel-Vefahren berechnet wird. Die Linearkombination

$$\mathbf{x}^{n+1} = \omega\tilde{\mathbf{x}}^{n+1} + (1-\omega)\,\tilde{\mathbf{x}}^n$$

mit dem Relaxationsparameter $\omega \geq 1$ gibt dann die nächsten Iterierte. Es ist also

$$\mathbf{x}^{n+1} = -\omega\mathbf{D}^{-1}\left(\mathbf{R}\mathbf{x}^n + \mathbf{L}\mathbf{x}^{n+1} - \mathbf{b}\right) + (1-\omega)\,\mathbf{x}^n$$

und der Iterationsschritt lautet

$$\mathbf{x}^{n+1} = \mathbf{B}_\omega\mathbf{x}^n + \omega\left(\mathbf{D}+\omega\mathbf{L}\right)^{-1}\mathbf{b} \tag{12}$$

mit der Iterationsmatrix

$$\mathbf{B}_\omega = \left(\mathbf{D}+\omega\mathbf{L}\right)^{-1}\left[(1-\omega)\,\mathbf{D} - \omega\mathbf{R}\right]$$

3.3 Konvergenz von Iterativen Verfahren

3.3.1 Konvergenzordnung

Eine Folge

$$\lim_{n\to\infty} x_n = \bar{x}$$

hat die *Konvergenzordnung* $p \geq 1$, wenn es eine Konstante k gibt, so dass für alle n

$$\|x_{n+1} - \bar{x}\| \leq k \cdot \|x_n - \bar{x}\|^p$$

Für lineare Konvergenz, d.h. $p = 1$ muss $k < 1$ sein, da die folge sonst nicht konvergiert. $p = 2$ nennt man quadratischeKkonvergenz, $p = 3$ kubisch etc.

Um zu illustrieren, dass quaratische konvergenz „schneller" ist als lineare betrachten wir ein

Beispiel Die linear konvergente Folge x_n und die quadratisch konvergente Folge y_n konvergieren zum selben Wert

$$\lim_{n\to\infty} x_n = \lim_{n\to\infty} y_n = \bar{x}$$

Nehmen wir weiter hin,dass $\|x_n - \bar{x}\| = \|y_n - \bar{x}\|$=0.1 und der Vorfaktor k sei der gleiche für beide Folgen. Dann haben wir im nächsten Schritt

$$\begin{aligned} \|x_{n+1} - \bar{x}\| &\leq k \cdot 0.1 \\ \|y_{n+1} - \bar{x}\| &\leq k \cdot 0.01 \end{aligned}$$

mit der quadratisch konvergenten Folge den Restfehler schon deutlich mehr reduziert.

Die Konvergenzordnung kann man mit Hilfe der Schrittfunktion ϕ bestimmen:

Sei $\phi(x)$ eine Schrittfunktion mit $\lim_{n\to\infty} x_n = \bar{x}$, so dass $\phi(\bar{x}) = \bar{x}$.

Ist weiter $\phi(x)$ für $x \in [a, b]$ p-mal stetig differenzierbar und

$$\phi\prime(\bar{x}) = \phi\prime\prime(\bar{x}) = \ldots = \phi^{p-1}(\bar{x}) = 0$$

sowie

$$\phi^p(\bar{x}) \neq 0$$

dann ist

$$x_{n+1} = \phi(x_n)$$

ein Iterationsverfahren der Konvergenzordnung p. Für $p = 1$ muss außerdem $|\phi\prime(\bar{x})| < 1$ erfüllt sein.

Für das Newtonverfahren ist die Iterationsvorschrift

$$x_{n+1} = x_n - \frac{f(x_n)}{f\prime(x_n)}$$

Sei $\bar{x}$ eine einfache Nullstelle. Dann ist $f(\bar{x}) = 0$ aber $f\prime(\bar{x}) \neq 0$. Hieraus sehen wir, dass $\phi(\bar{x}) = \bar{x} - 0$, sowie

$$\begin{aligned} \phi\prime(\bar{x}) &= 1 - \frac{(f\prime(\bar{x}))^2 - f(\bar{x})\, f\prime\prime(\bar{x})}{(f\prime(\bar{x}))^2} \\ &= 1 - \frac{(f\prime(\bar{x}))^2}{(f\prime(\bar{x}))^2} \\ &= 0 \end{aligned}$$

und für eine einfache Nullstelle muss

$$\begin{aligned}
\phi''(\bar{x}) &= \left[\frac{d\frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2}}{dx}\right]_{\bar{x}} \\
&= \left[\frac{d}{dx}\frac{(f'(x))^2}{(f'(x))^2} - \frac{\frac{d}{dx}f(x)\,f''(x)\cdot(f'(x))^2 - \frac{d}{dx}(f'(x))^2\cdot f(x)\,f''(x)}{(f'(x))^4}\right]_{\bar{x}} \\
&= \left[-\frac{[f'(x)\,f''(x) + f(x)\,f'''(x)]\cdot(f'(x))^2 - [2\,(f'(x))\cdot f''(x)]\cdot f(x)\,f''(x)}{(f'(x))^4}\right]_{\bar{x}} \\
&= \frac{[f'(\bar{x})\,f''(\bar{x})]\cdot(f'(\bar{x}))^2}{(f'(\bar{x}))^4}
\end{aligned}$$

Also ist $\phi''(\bar{x}) \neq 0$ wenn $f''(\bar{x}) \neq 0$. Wenn also nicht gerade $f(x)$an der Nullstelle einen Wendepunkt hat, dann konvergiert das Newtonverfahren quadratisch. Bei Nullstellen an Wendepunkten (z.B. $f(x) = \sin(x)$) ist dann aber $f'''(\bar{x}) \neq 0$ und $\phi''(\bar{x}) = 0$.

Verfahren	Konvergenzordnung p
Bisektion	1 ($k = 0.5$)
Newton, einfache Nullstelle	2
Newton, mehrfache Nullstelle	1
Newton ohne Update der Ableitung	1
Sekantenverfahren	$\frac{1+\sqrt{5}}{2} \approx 1.6$

3.3.2 Konvergenzabschätzung

Nach dem Bannachschen Fixpunktsatz kann man auch für mehrdimensionale Fixpunktiterationen, wie sie beim Lösen von linearen Gleichungssystemen z.B. mit dem Gauß-Seidel Verfahren auftauchen auch Abschätzungen über die Konvergenz vornehmen.

Sei$\|\mathbf{B}\|_\infty$ die (Maimum/Zeilensummen-)Norm der Iterationsmatrix $\mathbf{B}$ und $\bar{\mathbf{x}}$ ein (mehrdimensionaler) Fixpunkt, so ist$\bar{\mathbf{x}}$ anziehend, falls $\|\mathbf{B}\|_\infty < 1$. Diese Bedingung ist immer erfüllt, wenn die Matrix $\mathbf{B}$ strikt diagonaldominant ist, d.h.

$$|b_{ij}| > \sum_{i \neq j}^{n} |b_{ij}| \text{ für alle i.}$$

Man nennt das auch Zeilensummenkriterium.

Wiederum analog zur eindimensionalen Fixpunktiteration lassen sich folgende Abschätzungen machen

$$\|\mathbf{x}^n - \bar{\mathbf{x}}\|_\infty \leq \frac{\|\mathbf{B}\|_\infty{}^n}{1 - \|\mathbf{B}\|_\infty} \left\|\mathbf{x}^1 - \mathbf{x}^0\right\|_\infty \text{ a-priori Abschätzung}$$

$$\|\mathbf{x}^n - \bar{\mathbf{x}}\|_\infty \leq \frac{\|\mathbf{B}\|_\infty}{1 - \|\mathbf{B}\|_\infty} \left\|\mathbf{x}^n - \mathbf{x}^{n-1}\right\|_\infty \text{ a-posteriori Abschätzung}$$

Mit dem Spektralradius $spr(\mathbf{B}) = \max_{1\leq i\leq n} |\lambda_i(\mathbf{B})|$ der durch den größten Eigenwert (s. Gelichung14) der Matrix gegeben ist, hat man i.A. eine noch bessere Abschätzung wegen $\max_{1\leq i\leq n} |\lambda_i(\mathbf{B})| \leq \|\mathbf{B}\|_\infty$. Für hermitesche Matrizen ist die Spektralnorm $\|\mathbf{B}\|_2 = \max \frac{\|\mathbf{Bx}\|_2}{\|\mathbf{x}\|_2}$ gleich dem Spektralradius.

3.4 Nichtlineare Gleichungssysteme

Vektorielle Funktionen $\overrightarrow{f}(\mathbf{x}) = f(x_1, \ldots, x_n)$ lassen sich schreiben als

$$\overrightarrow{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ f_2(x_1, \ldots, x_n) \\ \ldots \\ f_n(x_1, \ldots, x_n) \end{pmatrix}$$

wobei in der Regel die eindimensionalen Funktionen $f_i(x_1, \ldots, x_n)$ nicht nur lineare x_i enthalten. Die Suche nach einem Lösungsvektor

$$\overrightarrow{f}(\mathbf{x}) = \mathbf{0}$$

erfordert dann die Lösung eines nichtlinearen Gleichungssystems. Dies geschieht mit einer allgemeineren Version des Newtonverfahrens.

Wir erinnern uns zunächst an die eindimensionale Form. Hier wird eine Taylorentwicklung 1.Ordnung der Funktion an der Stelle $\mathbf{x}_0$ gemacht

$$\begin{aligned} g(x) &\approx g(x_0) + g\prime(x_0)(x - x_0) = 0 \\ \vec{f}(x) &\approx \vec{f}(\mathbf{x}_0) + \mathbf{D}\left[\left(\vec{f}\mathbf{x}_0\right)\right](\mathbf{x} - \mathbf{x}_0) = 0 \end{aligned}$$

Im mehrdimensionale Fall geschieht genau das Gleiche. Nur anstelle der Ableitung wird eine Ableitungsmatrix, die Jacobimatrix, benutzt, in der die Matrixelemente alle partiellen Ableitungen aller Komponenten zu allen anderen Komponenten darstellen

$$\mathbf{D}\left[\left(\vec{f}\mathbf{x}\right)\right] = \begin{pmatrix} \frac{\partial f_1(x_1,\ldots,x_n)}{\partial x_1} & \frac{\partial f_1(x_1,\ldots,x_n)}{\partial x_2} & \cdots & \frac{\partial f_1(x_1,\ldots,x_n)}{\partial x_n} \\ \frac{\partial f_2(x_1,\ldots,x_n)}{\partial x_1} & \ddots & & \frac{\partial f_2(x_1,\ldots,x_n)}{\partial x_n} \\ \cdots & & \ddots & \\ \frac{\partial f_n(x_1,\ldots,x_n)}{\partial x_1} & \frac{\partial f_n(x_1,\ldots,x_n)}{\partial x_2} & \cdots & \frac{\partial f_n(x_1,\ldots,x_n)}{\partial x_n} \end{pmatrix} \tag{13}$$

Die Iterationsgleichung ist dann

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{D}\left[\left(\vec{f}\mathbf{x}^k\right)\right]^{-1} \cdot \left(\vec{f}\mathbf{x}^k\right)$$

Anstatt die Inverse der Jacobimatrix zu bilden, wird ein lineares Gleichungssystem

$$\mathbf{D}\left[\left(\vec{f}\mathbf{x}^k\right)\right] \cdot \mathbf{y}^k = -\left(\vec{f}\mathbf{x}^k\right)$$

gelöst und

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{y}$$

gerechnet.

Beispiel Das nichtlineare Gleichungssystem

$$\begin{aligned} x_1^2 + x_2^2 &= 1 \\ x_1^2 - x_2^2 &= -0.5 \end{aligned}$$

lösen wir mit dem verallgemeinerten Newtonverfahren und nehmen als Startvektor $\mathbf{x}^0 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$.

Wir schreiben

$$\overrightarrow{f}(\mathbf{x}) = f(x_1, \ldots, x_n) = \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ x_1^2 - x_2^2 + 0.5 \end{pmatrix}$$

und berechnen

$$\mathbf{D}\left[\overrightarrow{f}(\mathbf{x})\right] = \begin{pmatrix} 2x_1 & 2x_2 \\ 2x_1^2 & -2x_2 \end{pmatrix}$$

Dann ist für den ersten Iterationsschritt mit

$$\vec{f}(\mathbf{x}^0) = \vec{f}\begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 9 \\ -7.5 \end{pmatrix}$$

und

$$\mathbf{D}\left[\overrightarrow{f}(\mathbf{x}^0)\right] = \begin{pmatrix} 2 & 6 \\ 2 & -6 \end{pmatrix}$$

Dann lösen wir das lineare Gleichungssystem

$$\begin{aligned} \mathbf{D}\left[\overrightarrow{f}(\mathbf{x}^0)\right]\mathbf{y}^0 &= -\vec{f}(\mathbf{x}^0) \\ \begin{pmatrix} 2 & 6 \\ 2 & -6 \end{pmatrix}\mathbf{y}^0 &= \begin{pmatrix} 9 \\ -7.5 \end{pmatrix} \end{aligned}$$

dessen Lösung

$$\mathbf{y}^0 = -\begin{pmatrix} 0.375 \\ 1.375 \end{pmatrix}$$

ist. Das liefert dann

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{x}^0 + \mathbf{y}^0 \\ \mathbf{x}^1 &= \begin{pmatrix} 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 0.375 \\ 1.375 \end{pmatrix} \\ \mathbf{x}^1 &= \begin{pmatrix} 0.625 \\ 1.625 \end{pmatrix} \end{aligned}$$

Im nächsten Schritt müssen wir dann die Funktion $\overrightarrow{f}(\mathbf{x}^1) = \overrightarrow{f}\begin{pmatrix} 0.625 \\ 1.625 \end{pmatrix} = \begin{pmatrix} -2.0306 \\ 1.75 \end{pmatrix}$ und die Jacobimatrix $\mathbf{D}\left[\overrightarrow{f}(\mathbf{x}^1)\right] = \begin{pmatrix} 1.25 & 3.25 \\ 1.25 & -3.25 \end{pmatrix}$ für den neuen Vektor berechnen. Dann wiederum das lineare Gleichungssystem lösen, um $\mathbf{y}^1 = -\begin{pmatrix} 0.112 \\ 0.585 \end{pmatrix}$ und damit $\mathbf{x}^2 = \mathbf{x}^1 + \mathbf{y}^1$ zu erhalten. usw.

k	$\mathbf{x}^k$
0	$\begin{pmatrix} 1 \\ 3 \end{pmatrix}$
1	$\begin{pmatrix} 0.625 \\ 1.625 \end{pmatrix}$
2	$\begin{pmatrix} 0.512 \\ 1.04 \end{pmatrix}$
3	$\begin{pmatrix} 0.5001 \\ 0.88108 \end{pmatrix}$
4	$\begin{pmatrix} 0.50000 \\ 0.86615404 \end{pmatrix}$
5	$\begin{pmatrix} 0.50000 \\ 0.8660254 \end{pmatrix}$
∞	$\begin{pmatrix} 0.5 \\ \frac{\sqrt{3}}{2} = 0.8660254 \end{pmatrix}$

Auch das mehrdimensionale Newtonverfahren konvergiert quadratisch, und auch hier ist der Erfolg vom Startpunkt abhängig, d.h. $\mathbf{x}^0$ muss nahe genug an der Lösung $\bar{\mathbf{x}}$ liegen (bzw. die Taylorentwicklung 1. Ordnung muss eine gute Näherung sein). Wie auch beim eindimensionalen Newtonverfahren kann der Rechenaufwand (und leider auch die Konvergenzordnung) dadurch reduziert werden, dass z.B. die Jacobimatrix nur am Anfang (oder alle t Iterationsschritte) berechnet wird.

3.5 Eigenwerte und Eigenvektoren von Matrizen

Die Berechnung von Eigenwerten und zugehörigen Eigenwerten von Matrizen (und Operatoren) ist eine weitere häufig auftretende Aufgabe im wissenschaftlichen Rechnen. Beispiele sind die Bestimmung der Schwingungsfrequenzen eines Systems gekoppelter Oszillatoren, z.B. die Atome in einem Molekül oder Kristallschwingungen, deren zugehörige Eigenvektoren die Auslenkungen entlang der Schwingungsmoden beschreiben, oder die Berechnung der Energieeigenwerte des Hamiltonoperators (sowie die zugehörigen Eigenzustände (Eigenfunktionen)).

Die Definition

$$\mathbf{A}\mathbf{x}_i = \lambda \mathbf{x}_i \tag{14}$$

bedeutet, dass eine Matrix $\mathbf{A}$ angewandt auf einen Eigenvektor $\mathbf{x}_i$ erzeugt wieder desselben Vektor bis auf einen Vorfaktor. Der Vorfaktor λ_i ist der Eigenwert.

Wir befassen uns hier nur mit Matrizen, die selbstadjungiert, also hermitesch sind. Für solche Matrizen sind alle Eigenwerte reell und die zugehörigen Eigenvektoren bilden ein Orthogonalsystem. Die erste Eigenschaft ist bequem und die andere nützlich.

Zur Bestimmung der Eigenwerte kann man nun mit

$$\det\left(\mathbf{A} - \lambda \mathbf{I}\right) = 0 \tag{15}$$

das charakteristische Polynom

$$\lambda^n + c_1\lambda^{n-1} - c_2\lambda^{n-2} \ldots \pm c_n = 0 \tag{16}$$

aufstellen. Dessen Nullstellen $\lambda_1, \ldots, \lambda_n$ sind die gesuchten Eigenwerte.

Um die zugehörigen Eigenvektoren zu bestimmen löst man für jeden Eigenwert λ_i das lineare Gleichungssystem

$$\left(\mathbf{A} - \lambda_i \mathbf{I}\right)\mathbf{x}_i = 0 \tag{17}$$

z.B. mit Hilfe des Gauß-Algorithmus.

Damit können wir im Prinzip das Eigenwert/Eigenvektorproblem lösen. Allerdings ist die Aufstellung des charakteristischen Polynoms und die Bestimmung von dessen Nullstellen für große Matrizen nicht effizient und außerdem schlecht konditoniert. Deswegen wenden wir uns anderen Verfahren zu, die sich grob in zwei Klassen aufteilen lassen: 1.) Durch geeignete Umformung wir die ursprüngliche Matrix in eine Form gebracht, so dass sich das charakteristische Polynom und dessen Nullstellen besonders einfach bestimmen lassen. 2.) Iterative Näherungslösen liefern nicht alle, sondern nur ausgezeichnete Eigenwerte.

Das charakteristische Polynom von Dreiecksmatrizen ist besonders angenehm:

Für ine rechts-obere Dreiecksmatrix $\mathbf{R}$ berechnen wir

$$\det\left(\mathbf{R} - \lambda \mathbf{I}\right) = \begin{vmatrix} r_{11}-\lambda & r_{12} & \ldots & r_{1n} \\ 0 & r_{22}-\lambda & \ldots & r_{2n} \\ & & \ddots & \vdots \\ 0 & 0 & 0 & r_{4n}-\lambda \end{vmatrix} = 0$$

Wir hatten schon gesehen, dass die Determinante einer Dreiecksmatrix das Produkt der Diagonalelemente ist, also hier: $(r_{11}-\lambda)(r_{22}-\lambda)(r_{33}-\lambda)\ldots(r_{nn}-\lambda) = 0$ und erkennen, dass die Eigenwerte bereits auf der Diagonalen von $\mathbf{R}$ stehen.

Ähnlichkeitstransformationen

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \tag{18}$$

sind Transformationen, bei welchen die erzeugte Matrix dieselben Eigenwerte hat, wie die ursprüngliche Matrix und die zugehörigen Eigenvektoren von $\mathbf{B}$ sind $\mathbf{y}_i = \mathbf{T}^{-1}\mathbf{x}_i$. Transformationsmatrizen $\mathbf{T}$, die eine Ähnlichkeitstransformation erzeugen sind unitäre Matrizen, d.h. $\mathbf{T}^{-1} = \mathbf{T}^T$. Um zu zeigen, dass die

Eigenwerte unverändert bleiben zeigt man, dass das charakteristische Polynom gleich ist:

$$\begin{aligned}
\det(\mathbf{A}-\lambda\mathbf{I}) &= 0 \\
\det(\mathbf{B}-\lambda\mathbf{I}) &= \det\left(\mathbf{T}^{-1}\mathbf{A}\mathbf{T}-\lambda\mathbf{I}\right) \\
&= \det\left(\mathbf{T}^{-1}(\mathbf{A}-\lambda\mathbf{I})\mathbf{T}\right) \\
&= \underbrace{\det\left(\mathbf{T}^{-1}\right)}_{=1}\det((\mathbf{A}-\lambda\mathbf{I}))\underbrace{\det\left(\mathbf{T}^{1}\right)}_{=1}
\end{aligned}$$

Und für die Eigenwerte setzen wir die Behauptung in die Definition ein

$$\begin{aligned}
\mathbf{B}\left(\mathbf{T}^{-1}\mathbf{x}_i\right) &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{T}^{-1}\mathbf{x}_i \\
&= \mathbf{T}^{-1}\mathbf{A}\mathbf{x}_i \\
&= \mathbf{T}^{-1}\lambda_i\mathbf{x}_i \\
&= \lambda_i\left(\mathbf{T}^{-1}\mathbf{x}_i\right)
\end{aligned}$$

Ist die ursprüngliche Matrix $\mathbf{A}$, so ist $\mathbf{B}$ auch hermitesch. Für reelle Matrizen:

$$\mathbf{B}^T = \left(\mathbf{T}^{-1}\mathbf{A}\mathbf{T}\right)^T = \mathbf{T}^T\mathbf{A}\left(\mathbf{T}^{-1}\right)^T = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$$

Wir brauchen also eine unitäre Matrix, mit deren Hilfe wir eine Änlichkeitstransformation von $\mathbf{A}$ durchführen können, so dass eine Matrix von „einfacher Form“ entsteht. „Einfache Form“ und günstig in unserem Sinn sind Dreiecksmatrizen (s.o.) und Tridiagonalmatrizen (wird ergänzt.)

3.6 Reduktionsverfahren

3.6.1 QR-Zerlegung

Die QR-Zerlegung erzeugt eine orthogonale Matrix $\mathbf{Q}$ und eine rechts-obere Dreiecksmatrix $\mathbf{R}$

$$\mathbf{A} = \mathbf{Q}\mathbf{R}$$

Die orthogonale Matrix $\mathbf{Q}$ heisst so, weil ihre Spalten(vektoren) paarweise zueinander orthogonal sind. Sie ist unitär und es gilt $\mathbf{Q}\cdot\mathbf{Q}^T = \mathbf{I}$ und $\mathbf{Q}\cdot\mathbf{Q}^{-1} = \mathbf{I}$.

Die Transformation zur rechts-oberen Dreiecksmatrix ist dann

$$\mathbf{Q}^{-1}\mathbf{A} = \mathbf{R}$$

Um daraus die gesuchte Ähnlichkeitstransformation zu machen brauchen wir

$$\mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \mathbf{R}\mathbf{Q}$$

Wir erzeugen also auf der rechten Seite eine neue Matrix $\mathbf{A}_n$. Für diese können wir wieder eine QR-Zerlegung durchführen und bekommen so eine Iterationsvorschrift

$$\mathbf{A}_n = \mathbf{Q}_n\mathbf{R}_n$$

$$\mathbf{A}_{n+1} = \mathbf{Q}_n^{-1}\mathbf{A}_n\mathbf{Q}_n = \mathbf{R}_n\mathbf{Q}_n$$

die solange durchgeführt wird, bis $\mathbf{A}_{n+1}$ eine obere Dreiecksmatrix ist (oder $\mathbf{A}_n$ und $\mathbf{A}_{n+1}$ sich nicht mehr unterscheiden. Dann aber ist $\mathbf{Q}_n$ nahezu die Einheitsmatrix.). Man muss also abwechselnd QR-Zerlegung und Matrixmultiplikation durchführen.

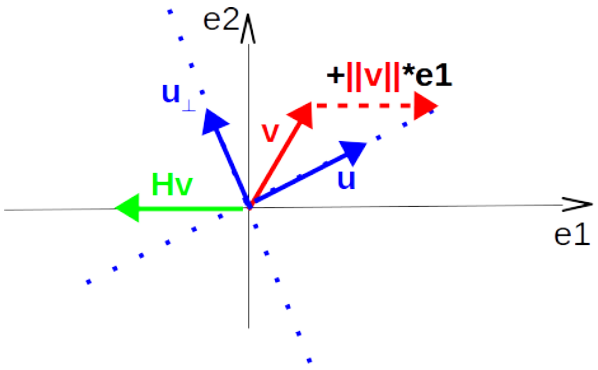

Abbildung 7: Die Householdertransformation $\mathbf{H}$ des Vektors $\mathbf{v}$ (rot) in ein Vielfaches des ersten Einheitsvektors $\mathbf{e}_1$ erzeugt $\mathbf{Hv}$ (grün). Die Transformation entspricht einer Spiegelung von $\mathbf{v}$ an einem Vektor der Richtung $\mathbf{u}_\perp$, dem Vektor orthogonal zu $\mathbf{u}$ (blau). $\mathbf{u}$ ist die Winkelhalbierende zwischen $\mathbf{v}$ und $\mathbf{e}_1$. Diese kann konstruiert werden, indem zum Vektor $\mathbf{v}$ ein Vektor mit Richtung von $\mathbf{e}_1$ und Länge von $\mathbf{v}$ hinzu addiert wird.

3.6.2 Householder-Spiegelung

Die gesuchte Transformationsvorschrift kann man sich (wieder) schrittweise aufgebaut so vorstellen, dass von der ersten bis zur $n-1$-ten Spalte die Unterdiagonalelemente zu null werden. Betrachten wir die erste Spalte, so ist die gesuchte Form eine Transformation, welche den Vektor, der ersten Spalte, in ein Vielfaches des ersten Einheitsvektors überführt. Dies kann durch eine geeignete Drehung geschehen (Givens-Rotation, hier nicht besprochen) oder durch Spiegelung an einer geeigneten Achse. Dies leistet die Householdertransformation. Abbildung 7 zeigt im zweidimensionalen wie der Vektor der Spielgelachse nach $\mathbf{u} = \mathbf{v} \pm \|\mathbf{v}\| \cdot \mathbf{e_1}$ erzeugt werden kann.

Die Householdermatrix ist

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}} \tag{19}$$

Dass es sich um eine Spiegelung an der orthogonalen Achse handelt, kann man auch rechnerisch zeigen. Wieder im $\mathbb{R}^2$ sei $\mathbf{u}$ irgendein normierter Vektor mit $\|\mathbf{u}\|_2 = 1$ und $\{\mathbf{u}, \mathbf{u}_\perp\}$ eine Basis wobei $\mathbf{u}^T\mathbf{u}_\perp = 0$. Dann kann man den Vektor $\mathbf{v} = \alpha\mathbf{u} + \beta\mathbf{u}_\perp$in dieser Basis ausdrücken. Die Householdertransformation angewendet ist dann (da $\mathbf{u}$ schon normiert ist, wird hier nicht mehr durch das Skalarprodukt $\mathbf{u}^T\mathbf{u} = 1$ geteilt)

$$\begin{aligned}
\mathbf{Hv} &= \mathbf{H}(\alpha\mathbf{u} + \beta\mathbf{u}_\perp) \\
&= \left(\mathbf{I} - 2\mathbf{u}\mathbf{u}^T\right)(\alpha\mathbf{u} + \beta\mathbf{u}_\perp) \\
&= \alpha\mathbf{u} + \beta\mathbf{u}_\perp - 2\alpha\underbrace{\mathbf{u}\mathbf{u}^T\mathbf{u}}_{1} - 2\beta\underbrace{\mathbf{u}\mathbf{u}^T\mathbf{u}_\perp}_{0} \\
&= -\alpha\mathbf{u} + \beta\mathbf{u}_\perp
\end{aligned}$$

Die Householdertransformation wird nun Schritt für Schritt auf die Matrix $\mathbf{A}$ angewandt, wobei die Spalten (bis auf die Elemente oberhalb der Diagonalen) die Vektoren $\mathbf{v}$ darstellen.

Beispiel Sei die Matrix

$$\mathbf{A} = \begin{pmatrix} 0 & -4 & 2 \\ 6 & -3 & -2 \\ 8 & 1 & -1 \end{pmatrix}$$

dann ist der erste Spaltenvektor $\mathbf{v_1} = \begin{pmatrix} 0 \\ 6 \\ 8 \end{pmatrix}$ mit $\|\mathbf{v_1}\| = \sqrt{0^2 + 6^2 + 8^2} = 10$, der auf den ersten Basisvektor $\mathbf{e_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ gespiegelt wird. Dann ist mit $\mathbf{u_1} = \mathbf{v_1} \pm \|\mathbf{v_1}\| \cdot \mathbf{e_1}$

$$\mathbf{u_1} = \begin{pmatrix} 0 \\ 6 \\ 8 \end{pmatrix} + 10 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 10 \\ 6 \\ 8 \end{pmatrix}$$

und die erste Housholdermatrix

$$\begin{aligned}
\mathbf{H}_1 &= \mathbf{I} - 2\frac{\mathbf{u_1}\mathbf{u_1}^T}{\mathbf{u_1}^T\mathbf{u_1}} \\
\mathbf{H}_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - 2\frac{\begin{pmatrix} 10 \\ 6 \\ 8 \end{pmatrix}\begin{pmatrix} 10 \\ 6 \\ 8 \end{pmatrix}^T}{\begin{pmatrix} 10 \\ 6 \\ 8 \end{pmatrix}^T\begin{pmatrix} 10 \\ 6 \\ 8 \end{pmatrix}} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - 2\frac{\begin{pmatrix} 100 & 60 & 80 \\ 60 & 36 & 48 \\ 80 & 48 & 64 \end{pmatrix}}{100 + 36 + 64} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0.60 & 0.80 \\ 0.60 & 0.36 & 0.48 \\ 0.80 & 0.48 & 0.64 \end{pmatrix} \\
\mathbf{H}_1 &= \begin{pmatrix} 0 & -0.60 & -0.80 \\ -0.60 & 0.64 & -0.48 \\ -0.80 & -0.48 & 0.36 \end{pmatrix}
\end{aligned}$$

Diese angewendet

$$\mathbf{H}_1\mathbf{A} = \mathbf{A_1}$$

$$\begin{pmatrix} 0 & -0.60 & -0.80 \\ -0.60 & 0.64 & -0.48 \\ -0.80 & -0.48 & 0.36 \end{pmatrix}\begin{pmatrix} 0 & -4 & 2 \\ 6 & -3 & -2 \\ 8 & 1 & -1 \end{pmatrix} = \begin{pmatrix} -10 & 1 & 2 \\ 0 & 0 & -2 \\ 0 & 5 & -1 \end{pmatrix}$$

ergibt wie gewünscht die Nullen in der ersten Spalte. Im zweiten Schritt betrachten wir nur die Submatrix $\begin{pmatrix} 0 & -2 \\ 5 & -1 \end{pmatrix}$, deren erste Spalte $\begin{pmatrix} 0 \\ 5 \end{pmatrix}$ auf $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ gespiegelt werden soll. Das entspricht einer Siegelung von $\mathbf{v_2} = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix}$ auf $\mathbf{e_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ mit $\|\mathbf{v_2}\| = 5$ via

$$\mathbf{u_2} = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix} + 5 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix}$$

Die zweite Housholdermatrix ist damit

$$
\begin{aligned}
\mathbf{H_2} &= \mathbf{I} - 2\frac{\mathbf{u_2}\mathbf{u_2}^T}{\mathbf{u_2}^T\mathbf{u_2}} \\
\mathbf{H_2} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - 2\frac{\begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix}\begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix}^T}{\begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix}^T\begin{pmatrix} 0 \\ 5 \\ 5 \end{pmatrix}} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - 2\frac{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 25 & 25 \\ 0 & 25 & 25 \end{pmatrix}}{50} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \\
\mathbf{H_2} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}
\end{aligned}
$$

Diese wenden wir an

$$\mathbf{H_2 A_1 = A_2}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}\begin{pmatrix} -10 & 1 & 2 \\ 0 & 0 & -2 \\ 0 & 5 & -1 \end{pmatrix} = \begin{pmatrix} -10 & 1 & 2 \\ 0 & -5 & 1 \\ 0 & 0 & 2 \end{pmatrix} = \mathbf{R}$$

und erhalten eine rechts-obere Dreiecksmatrix. Um die Zerlegung $\mathbf{A} = \mathbf{QR}$ zu komplettieren rechnen wir noch

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{H}_1\mathbf{H}_2 \\
\mathbf{Q} &= \begin{pmatrix} 0 & -0.60 & -0.80 \\ -0.60 & 0.64 & -0.48 \\ -0.80 & -0.48 & 0.36 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0.80 & 0.60 \\ -0.60 & 0.48 & -0.64 \\ -0.80 & -0.36 & 0.48 \end{pmatrix}
\end{aligned}
$$

Und zur Probe schließlich noch

$$
\begin{aligned}
\mathbf{A} &= \mathbf{QR} \\
&= \begin{pmatrix} 0 & 0.80 & 0.60 \\ -0.60 & 0.48 & -0.64 \\ -0.80 & -0.36 & 0.48 \end{pmatrix}\begin{pmatrix} -10 & 1 & 2 \\ 0 & -5 & 1 \\ 0 & 0 & 2 \end{pmatrix} \\
\mathbf{A} &= \begin{pmatrix} 0 & -4 & 2 \\ 6 & -3 & -2 \\ 8 & 1 & -1 \end{pmatrix}
\end{aligned}
$$

3.6.3 Tridiagonale Matrizen

Wendet man die Householdermatrix von beiden Seiten auf eine symmetrische Matrix an, erhält man eine tridiagonale Matrix[6] (Elemente nur auf der Hauptdiagonalen und den ersten beiden Nebendiagonalen)

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & \dots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \dots & \dots & a_{2n} \\ a_{13} & \dots & a_{33} & \dots & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & \dots & \dots & a_{nn} \end{pmatrix}$$

$$\mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ a_{12} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a_{23} & a_{33} & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & a_{n-1n-1} & a_{nn-1} \\ 0 & 0 & 0 & 0 & a_{nn-1} & a_{nn} \end{pmatrix}$$

Mit tridiagonalen Matrizen lässt sich die (anschließende) QR-Iteration effizient berechnen, die in der Iteration entstehenden Matrizen $\mathbf{A}_n$ sind auch wieder tridiagonal.

Wegen der vielen Nullen und der besonderen Struktur lässt sich das charakteristische Polynom einer tridiagonalen recht einfach aufstellen und berechnen.

3.7 Vektoriteration - Potenzmethode

Die Potenzmethode ist ein einfaches Verfahren, um den dominanten Eigenwert und den zugehörigen Eigenvektor einer Matrix zu bestimmen. Es gibt viele Anwednungen, in denen es genügt nur den größten (oder kleinsten) Eigenwert zu kennen, z.B. den Energieeigenwert des Grundzstands eines quantenmechanischen Systems.

Hierzu nutzen wir zum Einen die Definition von Eigenwert und Eigenvektor aus: $\mathbf{A}\mathbf{x}_i = \lambda\mathbf{x}_i$, d.h. die Matrix angewandt auf einen Eigenvektor erzeugt wieder diesen Eigenvektor bis auf einen Faktor, den Eigenwert. Wir können uns jetzt vorstellen, dass, uner Umständen, bei genügend häufiger Anwendung der Matrix auf einen beliebigen Vektor $\mathbf{y}_0$, am Ende ein Eigenvektor $\mathbf{x}$ dabei herauskommt (die Indices beziehen sich hier auf den Iterationsschritt, nicht die Nummer des Eigenvektors)

$$\begin{aligned} \mathbf{A}\mathbf{y}_0 &= \mathbf{y}_1 \\ \mathbf{A}\mathbf{y}_1 &= \mathbf{y}_2 \\ &\dots \\ \mathbf{A}\mathbf{y}_n &= \mathbf{x} \end{aligned}$$

Kompakt geschreiben bedeutet das

$$\lim_{n\to\infty} \mathbf{A}^n\mathbf{y}_0 = \mathbf{x}$$

Damit das Ganze funktioniert nutzen wir zum Anderen die Eigenschaft, dass die Eigenvektoren einer hermiteschen Matrix (und nur solche interessieren uns im Moment) ein Orthonormalsystem bilden, also eine Basis aufspannen, in der wir den beliebigen Vektor ausdrücken können

$$\mathbf{y}_0 = \sum_i c_i\mathbf{x}_i$$

[6] Bei nichtsymmetrischen Matrizen erhält man nach Householdertransformation sogenannte Hessenberg-Matrizen die oberhalb der tridiagonalen noch von Null verschiedene Elemente haben.

Kombinieren wir beides erhalten wir

$$\begin{aligned}\mathbf{A}^n\mathbf{y}_0 &= \mathbf{A^n}\sum_i^k c_i\mathbf{x}_i \\ &= \sum_i^k \lambda_i^n c_i\mathbf{x}_i\end{aligned}$$

wobei ausgenutzt wurde, dass bei n-maligem Anwenden der Matrix, der Eigenwert zur n-ten Potenz erhoben wird.

Schreibt man nun den ersten Term der Summe separat

$$\begin{aligned}\mathbf{A}^n\mathbf{y}_0 &= \sum_i^k \lambda_i^n c_i\mathbf{x}_i \\ &= \lambda_1^n c_i\mathbf{x}_1 + \sum_{i=2}^k \lambda_i^n c_i\mathbf{x}_i \\ &= \lambda_1^n\left(c_i\mathbf{x}_1 + \sum_{i=2}^k \left(\frac{\lambda_i}{\lambda_1}\right)^n c_i\mathbf{x}_i\right)\end{aligned}$$

Für separierte Eigenwerte, $|\lambda_1| > |\lambda_2| > |\lambda_3| \ldots |\lambda_k|$ und besonders gut, wenn $|\lambda_1| \gg |\lambda_2|$ konvergiert der Term $\left(\frac{\lambda_i}{\lambda_1}\right)^n$ gegen Null, also

$$\mathbf{A}^n\mathbf{y}_0 = \lambda_1^n\left(c_1\mathbf{x}_1 + \sum_{i=2}^k \underbrace{\left(\frac{\lambda_i}{\lambda_1}\right)^n c_i\mathbf{x}_i}_{\to 0 \text{ für } n\to\infty}\right) \tag{20}$$

und

$$\mathbf{A}^n\mathbf{y}_0 = \lambda_1^n c_1\mathbf{x}_1$$

Nach genügend vielen Iterationen dominiert also der betragsmäßig größte Eigenwert und die Iteration konvergiert zum (Vielfachen des) zugehörigen (ersten) Eigenvektors. Die entsprechende Iteratoinsvorschrift ist

$$\mathbf{y}_{n+1} = \frac{\mathbf{A}\mathbf{y}_n}{\|\mathbf{A}\mathbf{y}_n\|}$$

wobei die Normierung die Vorfaktoren „beseitigt“ und so auch den Vergleich zwischen den Vektoren aus zwei Iterationsschritten erleichtert. Um jetzt noch den Eigenwert zu erhalten nutzen wir wieder die Definition mit $\mathbf{A}\mathbf{y}_n = \lambda\mathbf{y}_n$, so dass nach Umformen der Eigenwert nach

$$\lambda = \frac{\mathbf{A}\mathbf{y}_n[j]}{\mathbf{y}_n[j]}$$

erhalten werden kann. $[j]$ ist hier ein beliebiges[7], z.B. das betragsmäßig größte, Element des Vektors $\mathbf{y}_n$. Um eine Zahl zu erhalten kann man auch anstelle der Betrachtung eines Elements des Vektors auch im Zähler und Nenner den Transponierten Zeilenvektor „dranmultiplizieren“, so dass man jeweils in Zähler und Nenner Skalarprodukte bekommt

$$\lambda = \frac{\mathbf{y}_n^T\mathbf{A}\mathbf{y}_n}{\mathbf{y}_n^T\mathbf{y}_n} \tag{21}$$

[7] Im Limit sollte für alle Elemente des Vektors der gleiche Vorfaktor /Eigenwert) erreicht sein.

Den Ausdruck in Gleichung 21 nennt man Rayleigh-Quotient.

Als Konvergenzkriterium dient z.B. ei sich nicht mehr ändernder Rayleigh-Quotient, oder allegemein

$$|\lambda_{n+1} - \lambda_n| \leq \mathsf{TOL}$$

wobei TOL irgendeine Toleranz bezeichnet und die Indices von λ den Iterationsschritt (und nicht den n-ten Eigenwert.)

Die Potenzmethode konvergiert umso schneller, je dominanter der erste Eigenwert ist, also je größer der Unterschied zum zweiten Eigenwert - und damit auch allen anderen Eigenwerten.

Den kleinsten Eigenwert kann man entsprechend

$$\frac{1}{\lambda_k}\mathbf{y}_n = \mathbf{A}^{-1}\mathbf{y}_n$$

mit der Iterationsmatrix $\mathbf{A}^{-1}$ erhalten, denn für den kleinsten Eigenwert λ_k is ja gerade $\frac{1}{\lambda_k}$ am größten.

In ähnlicher Weise kann man bei gut separierten Eigenwerten durch inverse Vektoriteration jeden Eigenwert λ_i bekommen, sofern man bereits eine gute Näherung μ zu diesem Eigenwert hat und

$$|\lambda_i - \mu| \ll |\lambda_j - \mu|$$

für alle $j \neq i$, also μ auch λ_i am nächsten liegt. Dann ist mit

$$(\mathbf{A} - \mu\mathbf{I})\,\mathbf{x}_i = (\lambda - \mu)\,\mathbf{x}_i$$

auch

$$(\mathbf{A} - \mu\mathbf{I})^{-1}\,\mathbf{x}_i = \frac{1}{(\lambda - \mu)}\mathbf{x}_i$$

und die Matrix

$$\mathbf{B} = (\mathbf{A} - \mu\mathbf{I})^{-1}$$

hat die Eigenwerte

$$\mu_i = \frac{1}{\lambda_i - \mu}$$

Mit der Vektoriteration wird das dominante (betragsmäßig größte) μ_i erhalten, und damit eben der Eigenwert λ_i, welcher der Näerung am nächsten liegt. $\mathbf{x}_i$ ist dann Eigenvektor von $\mathbf{B}$ zu μ_i , wie $\mathbf{x}_i$ auch Eigenvektor von $\mathbf{A}$ zu λ_i ist.

Anstelle der Berechnung einer Inversen um die Iterationsmatrix $\mathbf{B}$zu erhalten, implementiert man die Iteration wie über die Lösung eines linearen Gleichungssystems

$$(\mathbf{A} - \mu\mathbf{I})\,\mathbf{z}^{n+1} = \mathbf{z}^n$$

und setzte

$$\mathbf{y}^{n+1} = \frac{\mathbf{z}^{n+1}}{\|\mathbf{z}^{n+1}\|}$$

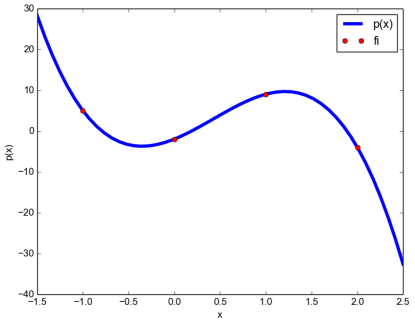

Abbildung 8: Wertepaare (Messpunkte, rot) und Interpolationspolynom (blau)

4 Interpolation und Ausgleichsrechnung

Ein häufiges Problem ist es, Messdaten geeignet durch einen funktionalen Zusammenhang zu beschreiben. Die Messdaten sind hierbei ein Satz von Wertepaaren $\{x_i, f_i\}$, *Stützpunkte* genannt. Die $\{x_i\}$ heißen *Stützpunkte* oder *Stützstellen*.

Grundsätzlich gibt es zwei Formen die Wertepaare durch Funktionen zu beschreiben, die Interpolation und die Ausgleichsrechnung („Fit“).

Bei der Interpolation wird eine einfache Funktion, in der Regel ein Polynom, gesucht, welches die Messdaten exakt reproduziert. Bei der Ausgleichrechnung hingegen wird eine Funktion gesucht, welche den Messwerten „am nächsten“ ist, obei die Messdaten nicht genau auf dem erhaltenen Funktionsgraph liegen müssen. Was „am nächsten“ bedeutet sehen wir im entsprechenden Abschnitt 4.2.

4.1 Interpolation

Gesucht ist eine stetige Funktion, die alle Messdaten/Wertepaare genau reproduziert und dabei eine einfache Form hat. Dies lässt sich durch Polynominterpolation erreichen.

Satz Für $n+1$-Wertepaare $\{x_i, f_i\}$ gibt es genau ein Polynom n-ten Grades

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots a_0 \tag{22}$$

so dass $p(x_i) = f_i$ für alle $i = 0, \dots n$.

Betrachten wir als Beispiel die 4 Wertepaare

x_i	-1	0	1	2
f_i	5	-2	9	-4

.

Diese können wir durch ein Polnom 3-ten Grades ausdrücken

$$p(x) = \sum_{j=0}^{3} a_j x^j$$

mit der Forderung dass auch

$$f_i = \sum_{j=0}^{3} a_j x_i^j$$

Das können wir schreiben als

$$a_3 x_i^3 + a_2 x_i^2 + a_1 x_i^1 + a_0 = f_i$$

für alle $i = 0, \dots, 3$. Das ist nichts Anderes als ein lineares Gleichungssystem

$$\begin{aligned} a_3(-1)^3 + a_2(-1)^2 + a_1(-1) + a_0 &= 5 \\ a_3(0)^3 + a_2(0)^2 + a_1(0) + a_0 &= -2 \\ a_3(1)^3 + a_2(1)^2 + a_1(1) + a_0 &= 9 \\ a_3(2)^3 + a_2(2)^2 + a_1(2) + a_0 &= -4 \end{aligned}$$

Dessen Lösungen a_j sind

$$\begin{aligned} a_0 &= -2 \\ a_1 &= 9 \\ a_2 &= 9 \\ a_3 &= -7 \end{aligned}$$

woraus sich das Polynom

$$p(x) = -7x^3 + 9x^2 + 9x - 2$$

ergibt (s. Abbildung 8).

Man kann also im Prinzip das Interpolationsproblem durch Lösen eines linearen Gleichungssystems erstzen. Das ist aber nicht effizient.

4.1.1 Interpolationspolynom nach Lagrange

Etwas effizienter (quadratisch statt kubisch skalierender Rechenaufwand) ist die Berechnung des Lagrange'schen Interpolationspolynoms nach

$$p(x) = \sum_{i=0}^{n} f_i l_i(x)$$

mit

$$l_i(x) = \Pi_{j=0;j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Für ein Polynom dritten Grades wie im obigen Beispiel ist also

$$l_0(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} \cdot \frac{x - x_3}{x_0 - x_3}$$

Setzen wir die Werte ein

$$\begin{aligned} l_0(x) &= \frac{x-0}{(-1)-0} \cdot \frac{x-1}{(-1)-1} \cdot \frac{x-2}{(-1)-2} \\ &= \frac{x-0}{(-1)} \cdot \frac{x-1}{-2} \cdot \frac{x-2}{-3} \\ &= -\frac{(x-0)(x-1)(x-2)}{6} \end{aligned}$$

Dieses Hilfspolynom ist nach Konstruktion gerade null für alle $x_j \neq x_i$, hier also hier für $x_1 = 0$, $x_2 = 1$, $x_3 = 2$, und gerade eins für $x_j = x_i$. Das gilt für alle

$$l_i(x) = \begin{cases} 0 & \text{für } x_j \neq x_i \\ 1 & \text{für } x_j = x_i \end{cases}$$

Damit haben wir die Forderung

$$p(x_i) = f_i$$

erfüllt. Die anderen Hilfspolynome sind

$$l_1(x) = \frac{x+1}{0-(-1)} \cdot \frac{x-1}{0-1} \cdot \frac{x-2}{0-2} = \frac{(x+1)(x-1)(x-2)}{2}$$
$$l_2(x) = \frac{x+1}{1-(-1)} \cdot \frac{x-0}{1-0} \cdot \frac{x-2}{1-2} = -\frac{(x+1)(x-0)(x-2)}{2}$$
$$l_3(x) = \frac{x+1}{2-(-1)} \cdot \frac{x-0}{2-0} \cdot \frac{x-1}{2-1} = \frac{(x+1)(x-0)(x-1)}{6}$$

Zusammengesetzt ergibt das

$$\begin{aligned}
p(x) &= -\frac{5}{6}(x-0)(x-1)(x-2) - \frac{2}{2}(x+1)(x-1)(x-2) - \frac{9}{2}(x+1)(x-0)(x-2) - \frac{4}{6}(x+1)(x-0)(x-1) \\
&= -\frac{5}{6}(x)\left(x^2-3x+2\right) - \frac{6}{6}(x+1)\left(x^2-3x+2\right) - \frac{27}{6}(x)\left(x^2-x-2\right) - \frac{4}{6}(x)\left(x^2-1\right) \\
&= -\frac{5}{6}\left(x^3-3x^2+2x\right) - \frac{6}{6}\left[\left(x^3-3x^2+2x\right)+\left(x^2-3x+2\right)\right] - \frac{27}{6}\left(x^3-x^2-2x\right) - \frac{4}{6}\left(x^3-x\right) \\
&= \left(-\frac{42}{6}\right)x^3 + \left(\frac{54}{6}\right)x^2 + \left(\frac{54}{6}\right)x - 2 \\
&= -7x^3 + 9x^2 + 9x - 2
\end{aligned}$$

was natürlich genau das gleiche Polynom ist, wie vorher bestimmt.

Bei Hinzunahme eines weiteren Stützpunktes ändern sich die Hilfspolynome in der Langrange-Methode.

4.1.2 Interpolation nach Newton

Eine bessere Methode ist deshalb die Verwenung von Hilfs(Basis-)polynomen nach Newton. Hierzu schreibt man das Interpolationspolynom ähnlich der Form von Lagrange als Produkt von Termen, allerdings verschachtelt:

$$\begin{aligned}
p(x) &= c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \dots c_n(x-x_0)(x-x_{n-1}) \\
&= \sum_{i=0}^{n} c_i \Pi_{k=0}^{i-1}(x-x_k) \qquad (23) \\
&\quad \sum_{i=0}^{n} f_{0,\dots,i} \Pi_{k=0}^{i-1}(x-x_k) \qquad (24)
\end{aligned}$$

Hieraus erkennen wir auch schon, dass $f_0 = p(x_0) = c_0$ da alle anderen Terme wegfallen. Entsprechend $f_1 = p(x_1) = c_0 + c_1(x-x_0)$, weil alle Terme mit $(x-x_1)$ wegfallen.etc.

Also:

$$\begin{aligned}
f_0 &= p(x_0) = c_0 \\
f_1 = p(x_1) &= c_0 + c_1(x-x_0) \\
f_2 = p(x_2) &= c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) \\
&\dots
\end{aligned}$$

Die Koeffizienten könnten wir dann z.B. rekursiv auflösen

$$\begin{aligned}
c_0 &= f_0 \\
c_1 &= \frac{f_1 - f_0}{(x-x_0)} \\
c_2 &= \frac{f_2 - c_0 - c_1(x-x_0)}{(x-x_0)(x-x_1)} \\
\dots &= \frac{(f_2-f_0)-(f_1-f_0)}{(x-x_0)(x-x_1)}
\end{aligned}$$

Und die Hinzunamhe weiterer Stützstellen erfordert nur die Berechnung des neuen, weiteren Terms. Allerdings muss für jeden Funktionswert die Rekursion erneut aufgerollt werden.

Alternativ bestimmt man die Koeffizienten c_i allerdings duch sogenannte „*dividierte Differenzen*“. Diese erhalten wir, auch wieder rekursiv, und am übersichtlichsten in einem Schema angeordnet. Es sind

$$f_{i,\dots i+k} = \frac{f_{i+1,\dots i+k} - f_{i,\dots i+k-1}}{x_{i+k} - x_i}$$

wobei $i = 0, \dots n$ und $k = 1, \dots n - i$

x_i	f_i	$k=1$	$k=2$	$k=3$
x_0=-1	f_0=5			
		$f_{0,1} = \frac{f_1 - f_0}{x_1 - x_0}$=-7		
x_1=0	f_1=-2		$f_{0,1,2} = \frac{f_{1,2} - f_{0,1}}{x_2 - x_0}$=9	
		$f_{1,2} = \frac{f_2 - f_1}{x_2 - x_1}$=11		$f_{0,1,2,3} = \frac{f_{1,2,3} - f_{0,1,2}}{x_3 - x_0}$=-7
x_2=1	f_2=9		$f_{1,2,3} = \frac{f_{2,3} - f_{1,2}}{x_3 - x_1}$=-12	
		$f_{2,3} = \frac{f_3 - f_2}{x_3 - x_2}$=-13		
x_3=2	f_3=-4			

Das Polynom also noch einmal umgeschrieben

$$p_{0,n}(x) = f_0 + f_{0,1}(x - x_0) + f_{0,1,2}(x - x_0)(x - x_1) + \dots f_{0,1,2,\dots n}(x - x_0)\dots(x - x_{n-1}) \tag{25}$$

Demnach sind also die Koeffizienten $c_0 = f_0$; $c_1 = f_{0,1}$; $c_2 = f_{0,1,2}$ etc. die dividierten Differenzen der „oberen Schräge“ (Terme mit $x_0; f_0$).

Wir erhalten dann

$$\begin{aligned} p(x) &= 5 - 7(x - x_0) + 9(x - x_0)(x - x_1) - 7(x - x_0)(x - x_1)(x - x_2) \\ &= 5 - 7(x + 1) + 9(x + 1)(x) - 7(x + 1)(x)(x - 1) \\ &= 5 - 7x - 7 + 9x^2 + 9x - 7x^3 + 7x \\ &= -7x^3 9x^2 + 9x - 2 \end{aligned}$$

wieder das obige Polynom.

Die Auswertung des Polynoms an einer Stelle ξ erfolgt nach dem Horner Schema:

$$b_n = c_n$$

Für $k = n - 1, \dots, 0$

$$b_k = c_k + \xi b_{k+1}$$

und $p(\xi) = b_0$

	c_n	c_{n-1}	$\dots$	c_1	c_0
ξ	0	ξb_n		ξb_2	ξb_1
	b_n	b_{n-1}		b_1	$b_0 = p_n(\xi)$
ξ	0	ξa_n		ξa_2	
	a_n	a_{n-1}		$a_1 = p_n\prime(\xi)$	
ξ	0	ξz_n	$\dots$		
	z_n	z_{n-1}	$z_1 = \frac{1}{2} p_n\prime\prime(\xi)$		
	0	$\dots$			

Mit den sich ergebenden Koeffizienzen b_k wird ein Polynom

$$p_{n-1}(x) = b_1 + b_2 x + b_3 x^2 + \dots + b_n x^{n-1}$$

definiert. Vergleich mit

$$p_n(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n$$

zeigt, dass

$$\begin{aligned} p_n(x) &= (b_0 - \xi b_1) + (b_1 - \xi b_2) x + (b_2 - \xi b_3) x^2 + \ldots + (b_{n-1} - \xi b_n) x^n \\ &= (x - \xi) p_{n-1} + r_0 \end{aligned}$$

mit $r_0 = b_0$. Also ist auch $p(\xi) = b_0$. Das Hornerschema spatet also den Linearfaktor ab und (für $x \neq \xi$)

$$\frac{p(x) - p(\xi)}{x - \xi} = p_{n-1}$$

wird mit $x \to \xi$ daraus $p'(\xi) = p_{n-1}$, was wiederum nach dem Horner-Schema berechnet werden kann.

Das lässt sich wiederholen, so dass

$$p_{n-1}(x) = (x - \xi) p_{n-2}(x) + r_1$$

und allgemein für $j = 0, \ldots, n-1$

$$p_{n-j}(x) = (x - \xi) p_{n-j-1}(x) + r_j$$

und $r_n = p_0$.

Wieder zusammengesetzt schreiben wir

$$\begin{aligned} p_n(x) &= r_0 + r_1 (x - \xi) + \ldots + r_n (x - \xi)^n \\ p_n(x) &= r_0 + (x - \xi) p_{n-1} \\ &= r_0 + (x - \xi)(r_1 + (x - \xi) p_{n-2}(x)) \\ &= r_0 + (x - \xi)(r_1 + (x - \xi)(r_2 + (x - \xi) p_{n-3}(x))) \\ &\ldots \end{aligned} \tag{26}$$

Der Vergleich mit der Taylor-Entwicklung von p an der Stelle ξ

$$p_n(x) = p(\xi) + (x - \xi) p'(\xi) + (x - \xi) \frac{p'(\xi)}{2} + \ldots + (x - \xi)^n \frac{p^{(j)}(\xi)}{j!}$$

zeigt

$$r_j = \frac{1}{j!} p^{(j)}(\xi)$$

4.1.3 Fehler bei Polynominterpolationen

Bei „ungünstigen“ Funktionen und außerdem ungünstiger Verteilung der Stützstellen, Beispiel $f(x) = \frac{1}{1+x^2}$, müssen Polynome hoher Ordnung bemüht werden, um die gegebenen Stützstellen abzubilden. Das führt dann außerhalb des Bereichs der Stützstellen zu Oszillationen[8] und zu Funktionswerten, die geben $\pm\infty$ streben (s. Abbildung 9).

Eine naive Extrapolation ist also i.A. nicht sinnvoll.

Das Interpolationspolynom gibt zwar alle Wertepaare der Stützstellen wieder, kann aber für die zwischenliegenden Werte beliebig falsch werden. Kennt man die Funktion, $f(x)$ die zu den Wertepaaren gehört (und erübrigt sich damit die Interpolation) gilt für den Fehler des Polynoms $p(x)$ vom Grad n zu den Wertepaaren (x_i, f_i) $i = 0, \ldots, n$:

Zu jedem $x \in [a, b]$ gibt es ein $\xi \in [a, b]$ mit

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi_{i=0}^{n} (x - x_i) \tag{27}$$

[8] Irgenwo müssen die Wendepunkte/Extrema ja hin.

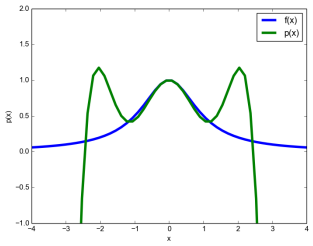

Abbildung 9: Die Funktion $f(x) = \frac{1}{1+x^2}$ und ein Interpolationspolynom 6ten Grades, welches die Funktion um $x \approx 0$ gut, für $x > 1$ aber nur schlecht beschreibt. An den Rändern strebt das Polynom gegen $-\infty$, die Funktion aber gegen Null.

4.1.4 Spline-Interpolation

Dem Fehler für Punkte zwischen den Stützstellen kann man besser beikommen, indem man nicht den gesamten Satz von Stützstellen auf einmal interpoliert, sondern stückweise. Bei zwei benachbarten Punkten kann dies nur eine lineare Interpolation sein (einfach beide Punkte mit dem Lineal verbinden). Das führt aber zu „Knicken“ also Unstetigkeitsstellen in der ersten Ableitungsfunktion. Um eine Interpolation mit stetiger erster und auch zweiter Ableitung zu bekommen führt man eine kubische Interpolation durch.

Ein kubischer Spline ist eine glatte Kurve, die durch alle Stützstellen läuft, überall zweimal stetig differenzierbar (glatt) ist, und stückweise durch eine kubische Funktion ausgedrückt wird.

Gegeben sind $n+1$ Stützstellen $\{x_i, f_i\}$ $i = 0, \ldots, n$, wobei $x_0 < x_1 < \ldots x_n$.

Man definiert Teilstücke des Splines mit

$$S_i(x) = a_i (x - x_i)^3 + b_i (x - x_i)^2 + c_i (x - x_i) + d_i$$

für das Kurvenstück zwischen (x_i, f_i) und (x_{i+1}, f_{i+1}). Enstprechend ist

$$S_{i-1}(x) = a_{i-1} (x - x_{i-1})^3 + b_{i-1} (x - x_{i-1})^2 + c_{i-1} (x - x_{i-1}) + d_{i-1}$$

das Kurvenstück „davor“, .d.h zwischen (x_{i-1}, f_{i-1}) und (x_i, f_i).

Die ersten beiden Ableitungen sind

$$S_i\prime(x) = 3a_i (x - x_i)^2 + 2b_i (x - x_i) + c_i$$

$$S_i\prime\prime(x) = 6a_i (x - x_i) + 2b_i$$

Dort, wo diese beiden Teilstücke zusammenstoßen gilt

$$S_{i-1}(x_i) = S_i(x_i) = f_i \text{ für } 1 < i \leq n$$

Außerdem müssen die erste und zweite Ableitung am Punkt x_i ebenfalls übereinstimmen (gleiche Steigung, gleiche Krümmung):

$$\begin{aligned} S\prime_{i-1}(x_i) &= S\prime_i(x_i) \text{ für } 1 < i \leq n \\ S\prime\prime_{i-1}(x_i) &= S\prime\prime_i(x_i) \text{ für } 1 < i \leq n \end{aligned}$$

Nutzen wir die Bedingungen aus, so bekommen wir mit

$$\begin{aligned} S_{i-1}(x_i) &= S_i(x_i) \\ a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} &= a_i \underbrace{(x_i - x_i)}_{=0}^3 + b_i \underbrace{(x_i - x_i)}_{=0}^2 + c_i \underbrace{(x_i - x_i)}_{=0} + d_i \\ a_{i-1}(x_i - x_{i-1})^3 + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} &= d_i = f_i \end{aligned} \quad (28)$$

und mit

$$\begin{aligned} S\prime_{i-1}(x_i) &= S\prime_i(x_i) \\ 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} &= 3a_i\underbrace{(x_i - x_i)}_{=0}^2 + 2b_i\underbrace{(x_i - x_i)}_{=0} + c_i \\ 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} &= c_i \end{aligned} \tag{29}$$

und auch noch

$$\begin{aligned} S\prime\prime_{i-1}(x_i) &= S\prime\prime_i(x_i) \\ 6a_{i-1}(x_i - x_{i-1}) + 2b_{i-1} &= 6a_i\underbrace{(x_i - x_i)}_{=0} + 2b_i \\ 6a_{i-1}(x_i - x_{i-1}) + 2b_{i-1} &= 2b_i \end{aligned} \tag{30}$$

Aus der letzen Gleichung 30 folgt

$$a_{i-1} = \frac{b_i - b_{i-1}}{3(x_i - x_{i-1})} \tag{31}$$

Das können wir in Gleichung 28 und Gleichung 29 einsetzen. In Gleichung 28 :

$$\begin{aligned} \frac{(b_i - b_{i-1})(x_i - x_{i-1})^2}{3} + b_{i-1}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} &= d_i \\ \frac{(b_i - b_{i-1})(x_i - x_{i-1})}{3} + b_{i-1}(x_i - x_{i-1}) + c_{i-1} &= \frac{d_i - d_{i-1}}{(x_i - x_{i-1})} \end{aligned}$$

$$c_{i-1} = \frac{d_i - d_{i-1}}{(x_i - x_{i-1})} - \frac{(b_i - b_{i-1})(x_i - x_{i-1})}{3} - b_{i-1}(x_i - x_{i-1})$$

und entsprechend

$$c_i = \frac{d_{i+1} - d_i}{(x_{i+1} - x_i)} - \frac{(b_{i+1} - b_i)(x_{i+1} - x_i)}{3} - b_i(x_{i+1} - x_i)$$

In Gleichung 29 eingesetzt:

$$\begin{aligned} (b_i - b_{i-1})(x_i - x_{i-1}) + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} &= c_i \\ (b_i + b_{i-1})(x_i - x_{i-1}) + c_{i-1} &= c_i \end{aligned} \tag{32}$$

Dahinein c_i und c_{i-1} weiter eingesetzt

$$(b_i + b_{i-1})(x_i - x_{i-1}) + \frac{d_i - d_{i-1}}{(x_i - x_{i-1})} - \frac{(b_i - b_{i-1})(x_i - x_{i-1})}{3} - b_{i-1}(x_i - x_{i-1})$$

$$= \frac{d_{i+1} - d_i}{(x_{i+1} - x_i)} - \frac{(b_{i+1} - b_i)(x_{i+1} - x_i)}{3} - b_i(x_{i+1} - x_i)$$

ist umgeformt

$$3(b_i + b_{i-1})(x_i - x_{i-1}) - (b_i - b_{i-1})(x_i - x_{i-1}) - 3b_{i-1}(x_i - x_{i-1}) + (b_{i+1} - b_i)(x_{i+1} - x_i) + 3b_i(x_{i+1} - x_i)$$

$$= 3\frac{d_{i+1} - d_i}{(x_{i+1} - x_i)} - 3\frac{d_i - d_{i-1}}{(x_i - x_{i-1})}$$

Jetzt erst einmal nur die linke Seite weiter umgeformt

$$= (2b_i + b_{i-1})(x_i - x_{i-1}) + (2b_i + b_{i+1})(x_{i+1} - x_i)$$

$$= (x_i - x_{i-1})b_{i-1} + 2b_i(x_{i+1} - x_i + x_i - x_{i-1})b_i + (x_{i+1} - x_i)b_{i+1}$$

So dass

$$\begin{aligned}(x_i - x_{i-1})\, b_{i-1} + 2\,(x_{i+1} - x_{i-1})\, b_i + (x_{i+1} - x_i)\, b_{i+1} &= 3\left(\tfrac{d_{i+1}-d_i}{(x_i - x_{i-1})} - \tfrac{d_i - d_{i-1}}{(x_i - x_{i-1})}\right)\\ \frac{(x_i - x_{i-1})}{(x_{i+1} - x_{i-1})} b_{i-1} + 2b_i + \frac{(x_{i+1} - x_i)}{(x_{i+1} - x_{i-1})} b_{i+1} &= \tfrac{3}{(x_{i+1} - x_{i-1})}\left(\tfrac{d_{i+1}-d_i}{(x_i - x_{i-1})} - \tfrac{d_i - d_{i-1}}{(x_i - x_{i-1})}\right)\end{aligned}$$

Da $d_i = f_i$ bekannt ist und die x_i auch , ist die rechte Seite bekannt. Die Koeffizienten b_i erhält man dann aus dem linearen Gleichungssystem

$$\begin{pmatrix} 2 & \alpha_0 & & & & \\ \beta_1 & 2 & \alpha_1 & & 0 & \\ & \beta_2 & \ddots & \cdots & & \\ & & & \ddots & \alpha_{n-2} & \\ & 0 & & \beta_{n-1} & 2 & \alpha_{n-1} \\ & & & & \beta_n & 2 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} = \begin{pmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \vdots \\ \delta_{n-1} \\ \delta_n \end{pmatrix} \tag{33}$$

wobei

$$\begin{aligned}\alpha_i &= \frac{(x_{i+1} - x_i)}{(x_{i+1} - x_{i-1})}\\ \beta_i &= \frac{(x_i - x_{i-1})}{(x_{i+1} - x_{i-1})}\\ \delta_i &= \frac{3}{(x_{i+1} - x_{i-1})}\left(\frac{d_{i+1} - d_i}{(x_i - x_{i-1})} - \frac{d_i - d_{i-1}}{(x_i - x_{i-1})}\right)\end{aligned} \tag{34}$$

für alle $i = 1, \ldots, n-1$

Die b_0 und b_n werden für natürliche Splines=0 gesetzt (natürliche Randbedingungen= linear außerhalb des Interpolationsbereichs. Für „erzwungene“ Randbedingungen setzt man eben diese Randbedingungen ein). Die Koeffizienten c_i und a_i ergeben sich durch zurück einsetzen in die Gleichungen 31 und 32.

Man kann die Splinefunktion auch aus einem anderen Blickwinkel herleiten:

Da die Spline-Funktion ein Polynom dritten Grades ist, sind die zweiten Ableitungen $S''(x)$ auf jedem Teilintervall mit $x \in [x_i, x_{i+1}]$ lineare Funktionen mit Steigung m_i bzw. m_{i+1} and den Teilintervallgrenzen (Stützstellen) x_i und x_{i+1}. Mit den Stetigkeitsbedingungen ist dann

$$S''(x) = m_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + m_{i+1} \frac{x - x_i}{x_{i+1} - x_i} \tag{35}$$

Das integriert ergibt

$$S'(x) = -m_i \frac{(x_{i+1} - x)^2}{2\,(x_{i+1} - x_i)} + m_{i+1} \frac{(x - x_i)^2}{2\,(x_{i+1} - x_i)} + A_i \tag{36}$$

und noch einmal integriert

$$S(x) = m_i \frac{(x_{i+1} - x)^3}{6\,(x_{i+1} - x_i)} + m_{i+1} \frac{(x - x_i)^3}{6\,(x_{i+1} - x_i)} + A_i\,(x - x_i) + B_i \tag{37}$$

Die Integrationskonstanten A_i und B_i lassen sich aus den Interpolationsbedingungen bestimmen und durch die m_i ausdrücken (also alternativ zu den Koeffizienten a_i,c_i) . Es sind (mit $m_i = 2b_i$)

$$A_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_{i+1} - f_i}{6}\,(m_{i+1} - m_i) \tag{38}$$

$$B_i = f_i - m_i \frac{(f_{i+1} - f_i)^2}{6} \tag{39}$$

Für das Spline-Problem gilt es n Intervalle für $n+1$ Stützstellen, also $4n$ Koeffizienten (je ein a, b, c, d pro Intervall) zu bestimmen.

Pro Intervall gibt es 2 Bedingungen aus den gleichen Funktionswerten an den Intervallrändern, den Stützstellen, also $2n$ Bedingungen. An den inneren Stützstellen müssen außerdem erste und zweite Ableitung identisch sein, das sind $2(n-1)$ Bedingungen. Mit den zusätzlichen 2 Randbedingungen sind das

$4n$ Bedingungen insgesamt.

Es müssen also $4n$ Koeffizienten aus $4n$ Bedingungen bestimmt werden.

4.2 Ausgleichsrechnung

Bei der Ausgleichsrechnung ist eine Funktion gesucht, welche einen Satz von Wertepaaren „am besten" wiedergibt. Diese Funktion muss kein Polynom sein, und die Wertepaare müssen auch nicht auf der Funktion liegen.

4.2.1 Lineare Ausgleichsrechung

Für n gegebene Wertepaare ist eine stetige Funktion$f(x)$ gesucht, so dass $f(x_i) \approx y_i$ für alle i. Um „möglichst gut" oder „möglichst nah" quantifizieren können wird ein Fehlerfunktional definiert:

$$E(f) = \sum_i (f(x_i) - y_i)^2$$

und gesucht ist dessen Minimum:$\min E = \sum_i (f(x_i) - y_i)^2$

Die gesuchte Ausgleichsfunktion (Fitfunktion) kann sich aus mehreren Funktionen$f_1, f_2 \ldots$zusammensetzen, derart dass

$$f(x) = \sum_j a_j f_j(x)$$

Bei der linearen Ausgleichsrechnung ist die Fitfunktion linear in den Koeffizienten a_i, die Funktionen f_i müssen dagegen nicht linear in x sein.

Betrachten wir ein Beispiel: Die Wertepaare

x	1	2	3	4
y	6	6.8	10	10.5

sollen durch eine Ausgleichsgerade $f(x) = ax + b = a_1 x^1 + a_2 x^0$ angenähert werden. Es gilt also $a = a_1$ und $b = a_2$ zu bestimmen.

Das bedeutet das Fehlerfunktional

$$E(a,b) = \sum_{i=1}^{4} (y_i - f(x_i))^2 = \sum_{i=1}^{4} (y_i - (ax_i + b))^2$$

soll in Bezug aus a und b minimiert werden. Dazu rechnen wir

$$\begin{aligned} \frac{\partial E}{\partial a} &= -2 \sum_{i=1}^{4} (y_i - (ax_i + b)) x_i = 0 \\ a \sum_{i=1}^{4} x_i^2 + b \sum_{i=1}^{4} x_i &= \sum_{i=1}^{4} y_i x_i \end{aligned}$$

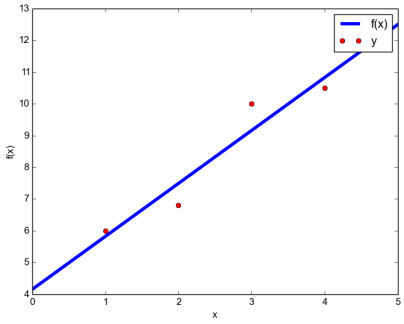

Abbildung 10: Ausgleichsgerade $f(x) = 1.67x + 4.15$ (blau) durch gegebene Wertepaare (rot).

$$\begin{aligned} \frac{\partial E}{\partial b} &= -2\sum_{i=1}^{4}(y_i - (ax_i + b)) = 0 \\ a\sum_{i=1}^{4}x_i + b\sum_{i=1}^{4}1 &= \sum_{i=1}^{4}y_i \end{aligned}$$

Das sind zwei Gleichungen mit zwei Unbekannten, es lässt sich also folgendes lineare Gleichungssystem aufstellen

$$\begin{pmatrix} \sum_{i=1}^{4}x_i^2 & \sum_{i=1}^{4}x_i \\ \sum_{i=1}^{4}x_i & \sum_{i=1}^{4}1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{4}y_i x_i \\ \sum_{i=1}^{4}y_i \end{pmatrix}$$

Und für das Beispiel

$$\begin{aligned} \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix} \\ \begin{pmatrix} a \\ b \end{pmatrix} &= \begin{pmatrix} 1.67 \\ 4.15 \end{pmatrix} \end{aligned}$$

Damit ist $f(x) = 1.67x + 4.15$ mit einem Restfehler von 1.323.

Allgemein formulieren wir das lineare Ausgleichsproblem folgendermaßen:

- Gegeben sind n Wertepaare (x_i, y_i)mit $i = 1, \ldots, n$.
- Die Funktionen
$$f(x) = \sum_{j}^{m} a_j f_j(x)$$
soll in ihren Koeffizienten a_j so optimiert werden, dass das
- Fehlerfunktional

$$\begin{aligned} E(\{a_j\}) &= \sum_{i=1}^{n}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{n}\left(y_i - \sum_{j}^{m}a_j f_j(x)\right)^2 \\ &= \|\mathbf{y} - \mathbf{A}\mathbf{a}\|_2^2 \end{aligned}$$

minimal wird. Hierbei sind $\mathbf{y} = \begin{pmatrix} y_1 \\ y_j \\ \vdots \\ y_m \end{pmatrix}$; $\mathbf{a} = \begin{pmatrix} a_1 \\ a_j \\ \vdots \\ a_m \end{pmatrix}$ und $\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_m(x_1) \\ f_1(x_2) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ f_1(x_n) & \cdots & \cdots & f_m(x_n) \end{pmatrix}$

Für eine Lösung gemäß $\mathbf{Aa} = \mathbf{y}$ ist $E = 0$. Allerdings gibt es typischerweise keine Lösungen, da die Anzahl der Wertepaare $n > m$ größer ist als die Anzahl der Fitparameter. Das Gleichungssystem ist überbestimmt. Bei $n < m$ gibt es auch keine eindeutige Lösung, da das Gleichungssystem unterbestimmt ist.

Stattdessen löst man ein Gleichungssystem, das sich aus den Normalengleichungen ergibt

$$\mathbf{A}^T\mathbf{Aa} = \mathbf{A}^T\mathbf{y} \tag{40}$$

Dieses Gleichungssystem liefert die geforderten Minima des Fehlerfunktionals bezüglich der Koeffizienten.

Der Begriff Normalengleichung erklärt sich aus der geometrischen Herleitung:

Es gilt wird $\|\mathbf{y} - \mathbf{Aa}\|_2$ durch $\mathbf{a}*$ minimiert, dann wird auch $\|\mathbf{y} - \mathbf{Aa}\|_2^2 = E(\mathbf{a})$ minimiert.

Setzen wir $\mathbf{Aa}* = \mathbf{v}*$, dann wird $\|\mathbf{y} - \mathbf{v}\|_2$ von $\mathbf{v}*$ minimiert. Die Vektoren $\mathbf{v} \in im\mathbf{A}$ lassen sich alle durch $\mathbf{Aa} = \mathbf{v}$ erzeugen. $\mathbf{v}*$ ist auch die Projektion von $\mathbf{y}$ auf die Ebene, in der alle $\mathbf{v} \in im\mathbf{A}$ liegen. Damit ist

$$(\mathbf{y} - \mathbf{v}*) \perp im\mathbf{A}$$

ein Normalenvektor zur Ebene. Jeder beliebige Vektor $\mathbf{w} = \mathbf{Ab} \in im\mathbf{A}$ ist dann auch orthogonal

$$\mathbf{w}^T(\mathbf{y} - \mathbf{v}*) = 0$$

bzw.

$$\begin{aligned} \mathbf{b}^T\mathbf{A}^T(\mathbf{y} - \mathbf{v}*) &= 0 \\ \mathbf{b}^T\mathbf{A}^T(\mathbf{y} - \mathbf{Aa}*) &= 0 \\ \mathbf{b}^T(\mathbf{A}^T\mathbf{y} - \mathbf{A}^T\mathbf{Aa}*) &= 0 \end{aligned}$$

Das ergibt umgeformt eben $\mathbf{A}^T\mathbf{Aa} = \mathbf{A}^T\mathbf{y}$.

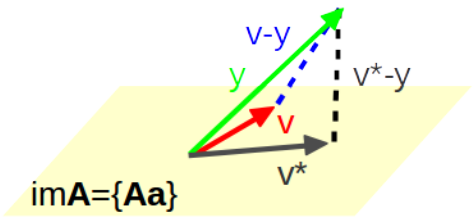

Abbildung 11: Der Vektor $\mathbf{v}* = \mathbf{Aa}*$ (grau) ist der Vektor in der Ebene welche das Bild der Matrix $\mathbf{A}$ aufspannt, welcher $\min \|\mathbf{v} - \mathbf{y}\|_2$ löst ($\|\mathbf{v}* - \mathbf{y}\|_2$(schwarz gestrichelt) $< \|\mathbf{v} - \mathbf{y}\|_2$ (blau gestrichelt)). Dies ist der Vektor $\mathbf{v} \in im\mathbf{A}$, der sich aus der Projektion von $\mathbf{y}$ (grün) auf die Ebene (blassgelb) ergibt.

Rechnerisch kann man entsprechend argumentieren:

Es gilt $E = \|\mathbf{y} - \mathbf{Aa}\|_2^2$ zu minimieren, also

$$\begin{aligned} E &= (\mathbf{y} - \mathbf{Aa})^T(\mathbf{y} - \mathbf{Aa}) \\ &= \mathbf{y^T y} - \mathbf{y^T Aa} - \underbrace{a^T\mathbf{A}^T\mathbf{y}}_{=\mathbf{y^T Aa}} + \mathbf{a^T A^T Aa} \\ &= \mathbf{y^T y} - 2\mathbf{y^T Aa} + \mathbf{a^T A^T Aa} \end{aligned}$$

Und um davon das Minimum zu bestimmen brauchen wir

$$\begin{aligned} \frac{\partial E}{\partial a_m} &= 0 \\ 0 &= \mathbf{2}\left(\mathbf{A^T A a - A^T y}\right) \\ 0 &= \left(\mathbf{A^T A a - A^T y}\right) \end{aligned}$$

Also $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$.

Die Ableitung erkennt man am besten in Komponenten (und Einsteinscher Summenkonvention $(A \cdot B)_{ij} = A_{ik} \cdot B_{kj} = \sum_{k=1} A_{ik} \cdot B_{kj}$)

$$\begin{aligned} E &= y_i^2 - 2A_{ij}a_j y_i + A_{ij}a_j A_{ik}a_k \\ \frac{\partial E}{\partial a_m} &= -2A_{im}y_i + A_{im}A_{ik}a_k + A_{ij}a_j A_{im} \\ &= -2A_{im}y_i + A_{im}A_{ik}a_k + A_{ik}a_k A_{im} \end{aligned}$$

wobei in der letzten Zeile einfach dich Indizierung von j nach k geändert wurde. Weiter erhält man

$$\begin{aligned} \frac{\partial E}{\partial a_m} &= 2A_{im}A_{ik}a_k - 2A_{im}y_i \\ 0 &= 2A^T_{mi}A_{ik}a_k - 2A^T_{mi}y_i \\ A^T_{mi}A_{ik}a_k &= A^T_{mi}y_i \end{aligned}$$

Zurück zum Beispiel. Für eine lineare Ansatzfunktion sind

$$\begin{aligned} f_1(x) &= x \\ f_2(x) &= 1 \end{aligned}$$

Damit ist

$$\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) \\ f_1(x_2) & f_2(x_2) \\ f_1(x_3) & f_2(x_3) \\ f_1(x_4) & f_2(x_4) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix}$$

und

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix} = \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix}$$

sowie

$$\mathbf{A}^T\mathbf{y} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 6.8 \\ 10 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix}$$

Damit ist das zu lösende Gleichungssystem $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$

$$\begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix}$$

und schließlich die Lösung

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1.67 \\ 4.15 \end{pmatrix}$$

wie schon vorher bestimmt.

Bemerkung: Das Gleichungssystem $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$ ist mit $\mathbf{M} = \mathbf{A}^\mathbf{T}\mathbf{A}$ und $\mathbf{A}^T\mathbf{y} = \mathbf{d}$ zwar ein lösbares lineares Gleichungssystem $\mathbf{M}\mathbf{a} = \mathbf{d}$. Robuster ist aber die Lösung über eine QR-Zerlegung. Wegen der nicht-quadratischen Matrix liefert $\mathbf{A} = \mathbf{Q}\mathbf{R}$ keine quadratische rechts-obere Dreieckmatrix. Man erhält stattdessen eine Matrix, die im oberen $m \times m$ Quadrat eine rechts-obere Dreieckmatrix ist und in den $n - m$ Zeilen darunter nur Nullen enthält (r steht hier von null verschiedene für Matrixelemente ohne weitere Indizierung)

$$\mathbf{R} = \begin{pmatrix} r & r & r & r \\ & r & r & r \\ & & r & r \\ & & & r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Streicht man die Zeilen mit Nullen bleibt eine $m \times m$ Matrix $\tilde{\mathbf{R}}$. Ähnlich erhält man mit $\mathbf{Q}^T\mathbf{y} = \mathbf{c}$ einen „zu großen“ Vektor, von dem nur die ersten m Einträge den Vektor $\tilde{\mathbf{c}}$ bilden.

Es gilt dann mit QR-Zerlegung

$$\begin{aligned} \mathbf{A}^T\mathbf{A}\mathbf{a} &= \mathbf{A}^T\mathbf{y} \\ \mathbf{a} &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y} \\ &= \left(\mathbf{R}^T\underbrace{\mathbf{Q}^TQ}_{=\mathbf{I}}\mathbf{R}\right)^{-1}\mathbf{R}^T\mathbf{Q}^T\mathbf{y} \\ &= \left(\mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{R}^T\mathbf{Q}^T\mathbf{y} \\ &= \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y} \end{aligned}$$

und man löst

$$\tilde{\mathbf{R}}\mathbf{a} = \tilde{\mathbf{c}}$$

4.3 Nichtlineare Ausgleichsrechnung

Bei nicht-lineare Ausgleichsproblemen treten die anzpassenden Parameter nicht nur linear, sondern auch in anderen funktionalen Zusammenhängen auf, so dass die Ausgleichsfunktion eben nicht.linear von den Parametern abhängt.

Ein „unechtes“ Beispiel ist

$$f(x) = a \cdot \exp(bx)$$

mit den Paramtern a und b. Dieses Problem lässt sich allerdings leicht linearisieren als

$$\ln f(x) = \ln a + bx$$

Echte Beispiele sind aber $f(x) = a\cos(bx)$ oder $f(x) = a\ln(x+b)$.

Der Ansatz zur Lösung von nicht-inearen Ausgleichsproblemen ist analog dem zu linearen Ausgleichsproblemen:

Zu einem Satz von n Wertepaaren (x_i, y_i) wird eine Ausgleichsfunktion

$$f(x) = f(a_1, a_2, \ldots, x)$$

mit Parametern $a_1, \ldots a_m$ gesucht, so dass das Fehlerfunktional

$$E(a_1, \ldots a_n) = \sum_{i=1}^{n} \left(y_i - f(a_1, \ldots a_m, x_i)\right)^2 = \|\mathbf{y} - f\|_2^2$$

minimiert wird.

Hierzu definieren wir einen Fehlervektor

$$g\left(a_1,\ldots a_m\right)=\left(\begin{array}{c} y_1-f\left(\mathbf{a},x_1\right) \\ \vdots \\ y_n-f\left(\mathbf{a},x_n\right) \end{array}\right)$$

mit dem das Fehlerfunktional zu

$$E\left(\mathbf{a}\right)=\left\|g\left(\mathbf{a}\right)\right\|_2^2$$

wird. Die Lösung dieses nichtlinearen Problems läuft über eine Linearisierung.

4.3.1 Gauss-Newton Verfahren

Wir erinnern uns noch einmal an das Newton-Verfahren zur Bestimmung von Nullstellen (oder MInima).

Für eine vorhandene Näherung des Lösungsvektors $\mathbf{a}_0$ wird eine Taylorentwicklung 1.Ordnung an der Stelle $\mathbf{a}_0$ gemacht, also eine Linearisierung

$$\vec{g}\left(\mathbf{a}\right) \quad \approx \quad \vec{g}\left(\mathbf{a}_0\right)+\mathbf{D}\left[\left(\vec{g}\mathbf{a}_0\right)\right]\left(\mathbf{a}-\mathbf{a}_0\right)$$

mit der Jacobimatrix

$$\mathbf{D}\left[\left(\vec{g}\mathbf{a}_0\right)\right]=\left(\begin{array}{cccc} \frac{\partial g_1(a_1,\ldots,a_n)}{\partial a_1} & \frac{\partial g_1(a_1,\ldots,a_n)}{\partial a_2} & \cdots & \frac{\partial g_1(a_1,\ldots,a_n)}{\partial a_n} \\ \frac{\partial g_2(a_1,\ldots,a_n)}{\partial a_1} & \ddots & & \frac{\partial g_2(a_1,\ldots,a_n)}{\partial a_n} \\ \cdots & & \ddots & \\ \frac{\partial g_n(a_1,\ldots,a_n)}{\partial a_1} & \frac{\partial g_n(a_1,\ldots,a_n)}{\partial a_2} & \cdots & \frac{\partial g_n(a_1,\ldots,a_n)}{\partial a_n} \end{array}\right) \tag{41}$$

bzw.

$$\mathbf{D}\left[\left(\vec{g}\mathbf{a}_0\right)\right]=\left(\begin{array}{cccc} \frac{\partial[y_1-f(a_1,\ldots,a_n,x_1)]}{\partial a_1} & \frac{\partial[y_1-f(a_1,\ldots,a_n,x_1)]}{\partial a_2} & \cdots & \frac{\partial[y_1-f(a_1,\ldots,a_n,x_1)]}{\partial a_n} \\ \frac{\partial[y_2-f(a_1,\ldots,a_n,x_2)]}{\partial a_1} & \ddots & & \frac{\partial[y_2-f(a_1,\ldots,a_n,x_2)]}{\partial a_n} \\ \cdots & & \ddots & \\ \frac{\partial[y_n-f(a_1,\ldots,a_n,x_n)]}{\partial a_1} & \frac{\partial[y_n-f(a_1,\ldots,a_n,x_n)]}{\partial a_2} & \cdots & \frac{\partial[y_n-f(a_1,\ldots,a_n,x_n)]}{\partial a_n} \end{array}\right) \tag{42}$$

Die Lösung des linearen Ausgleichproblems

$$\begin{array}{ll} \min\limits_{(\mathbf{a})} & \left\|g\left(\mathbf{a}\right)\right\|_2^2 \\ \approx & \min\left\|\vec{g}\left(\mathbf{a}_0\right)+\mathbf{D}\left[\left(\vec{g}\mathbf{a}_0\right)\right]\left(\mathbf{a}-\mathbf{a}_0\right)\right\|_2^2 \end{array}$$

liefert einen Vektor $\delta_0=\mathbf{a}-\mathbf{a}_0$ mit dem nach $\mathbf{a}_1=\delta+\mathbf{a}_0$ eine neue (verbesserte) Näherung erreicht wird, mit der wiederum das linearisierte Ausgleichsproblem gelöst wird, etc. Man löst also

$$\min_{(\mathbf{a}_k)}\left\|\vec{g}\left(\mathbf{a}_k\right)+\mathbf{D}\left[\left(\vec{g}\mathbf{a}_k\right)\right]\delta_k\right\|_2^2$$

bzw.

$$\mathbf{D}\left[\left(\vec{g}\mathbf{a}_k\right)\right]^T\mathbf{D}\left[\left(\vec{g}\mathbf{a}_k\right)\right]\delta_k=-\mathbf{D}\left[\left(\vec{g}\mathbf{a}_k\right)\right]^T\vec{g}\left(\mathbf{a}_k\right)$$

vgl.

$$\mathbf{A}^T\mathbf{A}\mathbf{a}=\mathbf{A}^T\mathbf{y}$$

und iteriert

$$\mathbf{a}_{k+1}=\delta_k+\mathbf{a}_k$$

solange bis z.B. $\left\|\delta_k\right\|_2^2<TOL$.

Vgl. auch die Iteration von nichtlinearen Gleichungssystemen.

$$\mathbf{x}^{k+1}=\mathbf{x}^k-\mathbf{D}\left[\left(\vec{f}\mathbf{x}^k\right)\right]^{-1}\cdot\left(\vec{f}\mathbf{x}^k\right)$$

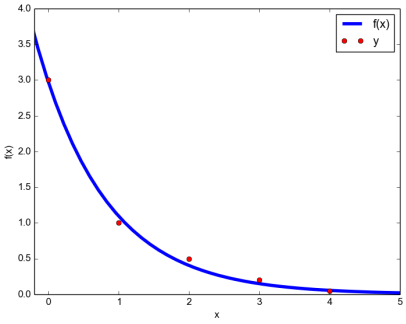

Abbildung 12: Wertepaare (rot) und nichtlineare Ausgleichsfunktion $f(x) = 2.98\exp(-1x)$ (blau).

4.3.2 Gedämpftes Gauss-Newton oder Levenberg-Marquardt Verfahren

Der Vektor δ_k liefert uns also in jedem Iterationsschritt die Richtung des mehrdimensionalen Optimierungsproblems, in welcher sich Verbesserung (minimierung) erzielen lässt. Wie weit aber in dieser Richtung ist die Verbesserung am größten? Eine Bestimmung (oder Einschränkung) der Schrittweite λ_k wird im gedämpften Gauss-Newton-Verfahren eingeführt, so dass

$$\mathbf{a}_{k+1} = \lambda\delta_k + \mathbf{a}_k$$

Der Faktor λ_k wird durch fortlaufende Halbierung $\lambda \in \left\{1, \frac{1}{2}, \frac{1}{4}, \ldots\right\}$ so bestimmt, dass

$$E(\mathbf{a}_k, \lambda) = \|\vec{g}(\mathbf{a}_k + \lambda_k\delta_k)\|_2^2 < E(\mathbf{a}_k, 0) = \|\vec{g}(\mathbf{a}_k)\|_2^2$$

womit ein Abstieg (Minimierung) erzwungen wird. (Dies ist eine einfache Version des Levenberg Marquardt-Verfahrens.) Zusätzlich wird so erreicht, dass das Gauss-Netwon-Verfahren, welches wie jedes Newton-Verfahren vom Startpunkt abhängt, für einen größeren Bereich von Startvektoren konvergiert.

Beispiel Gegeben sind die Wertepaare

x	0	1	2	3	4
y	3	1	0.5	0.2	0.05

. Diese sollen mit der Ansatzfunktion $f(x) = a_1\exp(a_2x)$ gefittet werden. Die gesuchte Vektorfuntkion $g(\mathbf{a})$ hat also die Komponenten

$$g_i(a_1, a_2) = y_i - a_1\exp(a_2x_i)$$

und die zugehörige Jacobi-Matrix ist

$$Dg(\mathbf{a}) = \frac{\partial g_i(\mathbf{a})}{\partial a_j} = (-\exp(a_2x_i), -a_1x_i\exp(a_2x_i))$$

Ein Blick auf die Lage der Stützpunkte lässt uns $a_1 > 0$ und $a_2 < 0$ erwarten. Für einen Startvektor $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix}$ erhalten wir ungedämpft

ungedämpft	0	1	2	3	4
a_1	1	2.99	1.26	2.91	2.98
a_2	-1.5	0.39	0.28	-0.86	-1.00

und gedämpft

gedämpft	0	1	2	3
a_1	1	1.99	2.91	2.98
a_2	-1.5	-0.554	-0.95	-0.999

Für einen Startvektor $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ konvergiert das ungedämpfte Verfahren nicht, für das gedämpfte Verfahren erhalten wir

gedämpft	0	1	2	3	∞
a_1	2	0.0038	0.00384	0.207	2.98
a_2	2	2	1.75	-0.709	-1.00

5 Differentiation und Integration

5.1 Differentiation

Numerische Differentiation ist immer dann nötig, wenn die Ableitungsfunktion nicht zur Verfügung steht. Oft ist auch die abzuleitende Funktion selber nicht völlig bekannt, sondern nur Abschnittsweise oder durch einzelne Wertepaare, so dass stückweise eine Näherung erfolgen kann. Differentiation an einzelnen Stellen haben wir auch schon mehrfach zur Berechnung der Jacobimatrix gebraucht, bisher aber immer durch bekannte Ableitungsfunktionen erreichen können. Ableitungen von Polynomen können natürlich über das Horner-Schema ermittelt werden.

Ableitung als Grenzwert des Differenzenquotienten

$$f\prime(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} \approx \frac{f(x_0 + h) - f(x_0)}{h} = D_1 f(x_0, h) \tag{43}$$

$D_1 f$ ist eine Differenzenformel oder finite Differenz 1.Ordnung.

Differenzenformel und Fehlerordnung Das betrachten wir genauer anhand einer Taylorentwicklung:

$$f(x_0 + h) = f(x_0) + hf\prime(x_0) + \frac{1}{2}h^2 f\prime\prime(x_0) + \frac{1}{6}h^3 f\prime\prime\prime(x_0) \ldots$$

Nach der ersten Ableitung umgeformt ist das

$$\begin{aligned} hf\prime(x_0) &= f(x_0 + h) - f(x_0) - \frac{1}{2}h^2 f\prime\prime(x_0) - \frac{1}{6}h^3 f\prime\prime\prime(x_0) \ldots \\ f\prime(x_0) &= \frac{f(x_0 + h) - f(x_0)}{h} - \frac{1}{2}hf\prime\prime(x_0) - \frac{1}{6}h^2 f\prime\prime\prime(x_0) \ldots \end{aligned}$$

Der Vergleich

$$f\prime(x_0) = D_1 f(x_0, h) - \frac{1}{2}hf\prime\prime(x_0) - \frac{1}{6}h^2 f\prime\prime\prime(x_0) \ldots$$

zeigt, dass der Fehler in der Vernachlässigung aller Terme ab zweiter Ordnung in der Taylorentwicklung liegt. Dies ist der *Abschneidefehler* oder *Diskretisierungsfehler* .

Die Fehlerordnung k ist durch die führende Potenz von h gegeben ($\mathcal{O}\left(h^k\right)$) : Falls es ein $C > 0$ gibt, so dass für genügend kleines h gilt

$$|Df(x_0, h - f\prime(x_0))| \leq Ch^k$$

In Der Differenzenformel 43 ist $K = 1$, die Fehlerordnung also 1. Gewünscht sind aber möglichst hohe Fehlerordnungen, also möglichst kleine Diskretisierungsfehler. Dazu bedarf es dann aber einer verbesserten Differenzenformel.

Diskretisierungsfehler und Rundungsfehler Man vermutet schon, dass der Fehler dieser Näherung irgendwie von h abhängt und die Näherung vermutlich mit kleinerem h immer besser wird.

Betrachten wir als Beispiel $f(x) = \sin x$ an der Stelle $x_0 = 1$ und wählen h immer kleiner. Die exakte Antwort ist natürlich $f\prime(x_0) = \cos(x_0) = \cos(1) = 0.5403023059$. Bei 10-stelliger dezimaler Rechnung bekommt man

h	$D_1 f(1, h) - \cos(1)$	h	$D_1 f(1, h) - \cos(1)$
10^{-1}	0.0429385529	10^{-7}	0.0003023059
10^{-2}	0.0042163259	10^{-8}	0.0003023059
10^{-3}	0.0004208059	10^{-9}	0.0403023059
10^{-4}	0.0000423059	10^{-10}	0.5403023059
10^{-5}	0.0000023059	10^{-11}	0.5403023059
10^{-6}	0.0000023059	10^{-12}	0.5403023059

Der Fehler wird also in der Tat zunächst immer kleiner. Dann aber wird er wieder größer und ab 10^{-10} bleibt er gleich groß. Was uns hier zu schaffen macht sind Rundungsfehler und Auslöschung (s. Abschnitt 1.4.1), d.h. bei kleinem h sind $\sin(1+h)$ und $\sin(1)$ fast gleich und die fehlerbehafteten Ziffern rutschen nach vorn. Durch ein sehr kleines h geteilt macht die Sache dann noch schlimmer. Rundungsfehler und Diskretisierungsfehler, die zusammen den Gesamtfehler ausmachen, laufen sich also entgegen. Daher muss es aber auch irgendwo ein Optimum geben, bei dem der Gesamtfehler möglichst klein ist. Im Beispiel ist das bei $h = 10^{-5}$ und $h = 10^{-6}$.

Um das Optimum zu bestimmen betrachten wir noch einmal den Gesamtfehler

$$\begin{aligned} |\mathsf{rd}\left(D_1 f\left(x_0,h\right)\right) - f\prime\left(x_0\right)| &= |\mathsf{rd}\left(D_1 f\left(x_0,h\right)\right) - D_1 f\left(x_0,h\right) + D_1 f\left(x_0,h\right) - f\prime\left(x_0\right)| \\ &\leq \underbrace{|\mathsf{rd}\left(D_1 f\left(x_0,h\right)\right) - D_1 f\left(x_0,h\right)|}_{\text{Rundungsfehler}} + \underbrace{|D_1 f\left(x_0,h\right) - f\prime\left(x_0\right)|}_{\text{Diskretisierungsfehler}} \\ &\approx \frac{2E_M}{h} + \frac{h}{2}\left|f\prime\prime\left(x_0\right)\right| \end{aligned}$$

wobei $E_M$den absoluten Maschinenfehler eines Funktionswerts mit $E_M = \mathsf{eps} \cdot |f(x_0)|$ angibt. Zur Bestimmung des Minimums bestimmen wir die erste Ableitung des Gesamtfehlers und setzten diese gleich Null:

$$\begin{aligned} \frac{d}{dh}\left(\frac{2E_M}{h} + \frac{h}{2}\left|f\prime\prime\left(x_0\right)\right|\right) &= 0 \\ &= -\frac{2E_M}{h^2} + \frac{1}{2}\left|f\prime\prime\left(x_0\right)\right| \end{aligned}$$

also

$$\begin{aligned} \frac{2E_M}{h^2} &= \frac{1}{2}\left|f\prime\prime\left(x_0\right)\right| \\ h^2 &= \frac{4E_M}{|f\prime\prime\left(x_0\right)|} \\ h &= \sqrt{\frac{4E_M}{|f\prime\prime\left(x_0\right)|}} \end{aligned}$$

Damit ist der Gesamtfehler minimal bei

$$h_{opt} = 2\sqrt{\frac{\mathsf{eps} \cdot |f\left(x_0\right)|}{|f\prime\prime\left(x_0\right)|}}$$

Verbesserte Differenzenformel Wir nähern uns von zwei Seiten, d.h. mit

$$\begin{aligned} f\left(x_0+h\right) &= f\left(x_0\right) + hf\prime\left(x_0\right) + \frac{1}{2}h^2 f\prime\prime\left(x_0\right) + \frac{1}{6}h^3 f\prime\prime\prime\left(x_0\right)\ldots \\ f\left(x_0-h\right) &= f\left(x_0\right) - hf\prime\left(x_0\right) + \frac{1}{2}h^2 f\prime\prime\left(x_0\right) - \frac{1}{6}h^3 f\prime\prime\prime\left(x_0\right)\ldots \end{aligned}$$

haben wir

$$f\left(x_0+h\right) - f\left(x_0-h\right) = 2hf\prime\left(x_0\right) + \frac{1}{3}h^3 f\prime\prime\prime\left(x_0\right)\ldots$$

und damit

$$D_2 f\left(x_0\right) = \frac{f\left(x_0+h\right) - f\left(x_0-h\right)}{2h} \tag{44}$$

Die Fehlerordnung ist dann K=2, denn

$$D_2 f\left(x_0\right) - f\prime\left(x_0\right) = \frac{1}{6}h^2 f\prime\prime\prime\left(x_0\right)\ldots$$

und diese zentrierten Differenzen (Gleichung 44) sind der einfachen Differenzenformel (Gleichung 43) vorzuziehen.

Höhere Ableitungen lassen sich auch wieder über die Taylorentwicklung erhalten. Wir addieren

$$\begin{aligned} f(x_0+h) &= f(x_0)+hf\prime(x_0)+\frac{1}{2}h^2f\prime\prime(x_0)+\frac{1}{6}h^3f\prime\prime\prime(x_0)\ldots \\ f(x_0-h) &= f(x_0)-hf\prime(x_0)+\frac{1}{2}h^2f\prime\prime(x_0)-\frac{1}{6}h^3f\prime\prime\prime(x_0)\ldots \end{aligned}$$

so dass

$$f(x_0+h)+f(x_0-h)=2f(x_0)+h^2f\prime\prime(x_0)+\ldots$$

und als Differenzenformel

$$f\prime\prime(x_0)=\frac{f(x_0+h)+f(x_0-h)-2f(x_0)}{h^2} \tag{45}$$

In mehreren (hier zwei) Dimensionen lässt sich entsprechend verfahren. So ist für $f(x,y)$

$$\frac{\partial f}{\partial x}=\frac{f(x+h,y)-f(x-h,y)}{2h}$$

und

$$\frac{\partial f}{\partial y}=\frac{f(x,y+h)-f(x,y-h)}{2h}$$

Für zweite nicht-gemischte Ableitungen erhält man

$$\frac{\partial^2 f}{\partial x^2}=\frac{f(x+h,y)+f(x-h,y)-2f(x,y)}{h^2}$$

und

$$\frac{\partial^2 f}{\partial y^2}=\frac{f(x,x+h)+f(x,y-h)-2f(x,y)}{h^2}$$

sowie für die gemischte Ableitung

$$\frac{\partial^2 f}{\partial x\partial y}=\frac{f(x+h,y+h)+f(x-h,y-h)-f(x+h,y-h)-f(x-h,y+h)}{4h^2}$$

aus

$$\begin{aligned} \frac{\partial^2 f}{\partial x\partial y} &= \frac{\frac{\partial f}{\partial y}f(x+h,y)-\frac{\partial f}{\partial y}f(x-h,y)}{2h} \\ &= \frac{\left[f(x+h,y+h)-f(x+h,y-h)\right]/2h-\left[f(x-h,y+h)-f(x-h,y-h)\right]/2h}{2h} \end{aligned}$$

oder entsprechend aus

$$\frac{\partial^2 f}{\partial y\partial x}=\frac{\frac{\partial f}{\partial x}f(x,y+h)-\frac{\partial f}{\partial x}f(x,y-h)}{2h}$$

5.2 Integration

Bei der numerischen Integration wird ein Näherungswert für das Integral

$$I = \int_a^b f(x)\,dx$$

mit $f : [a, b] \to \mathbb{R}$ gesucht.

Die einfachste Näherung ist eine Aufteilung des Integrationsintervalls in n Teile mit $h = \frac{b-a}{n}$ und $x_i = a + ih$ mit $i = 0, \ldots, n$, so dass $x_0 = a$ und $x_n = b$ ist.

Die Funktion $f(x)$ wird an den Stützstellen x_i durch ein Polynom approximiert, so dass anstelle von $f(x)$ ein Interpolationspolynom integriert wird.

5.2.1 Integrationspolynome

Die Rechteckregel, RI, ist ein Polynom 0ter Ordnung Sei zunächst das Intervall ein Stück, dann ist die Integralnäherung durch ein Rechteck

$$RI = f\left(\frac{a+b}{2}\right)(b-a)$$

wobei die Funktion in der Intervallhälfte ausgewertet wird (Höhe des Rechtecks). Für n Intervallstücke der Breite h mit stückweiser Integration von x_i bis $x_i + h$ wird daraus

$$\begin{aligned} RI(h) &= \sum_{i=0}^{n-1} \int_{x_i}^{x_i+h} f(x)\,dx \\ &\approx \sum_{i=0}^{n-1} f\left(\frac{x_i + x_i + h}{2}\right) \cdot h \\ &= h \sum_{i=0}^{n-1} f\left(x_i + \frac{h}{2}\right) \end{aligned}$$

Die Trapezregel, TI, ist ein Polynom 1ter Ordnung Die Funktion wird also durch eine Gerade von (bei einem Intervall stück) $f(a)$nach $f(b)$ angenähert und das Integral ist

$$TI = \frac{f(a) + f(b)}{2} \cdot (b-a)$$

bzw. bei n Intervallstücken der Breite h und stückweiser Integration von x_i bis $x_i + h$

$$\begin{aligned} TI(h) &= \sum_{i=0}^{n-1} \int_{x_i}^{x_i+h} f(x)\,dx \\ &\approx \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_i + h)}{2} \cdot h \\ &= h\left(\frac{f(x_0)}{2} + \sum_{i=1}^{n-1} f(x_i) + \frac{f(x_n)}{2}\right) \\ &= h\left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right) \end{aligned}$$

Die Simpsonregel, SI, ist ein Polynom 2.ter Ordnung Für das Interpolationspolynom 2. Ordnung erinnern wir uns an die newtonsche Form des Interpolationspolynoms 25:

$$\begin{aligned} p(x) &= f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) + \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}(x - x_0)(x - x_1) \\ &= f(a) + \frac{f(a+h) - f(a)}{h}(x - a) + \frac{\frac{f(b) - f(a+h)}{h} - \frac{f(a+h) - f(a)}{h}}{2h}(x - a)(x - a - h) \end{aligned}$$

wobei wir in der letzten Zeile $x_0 = a$; $x_1 = a + h$ und $x_2 = b$ gesetzt haben. Damit is die Intervallbreite $b - a = 2h$. Um das Polynom nun zu integrieren schauen wir uns jeden Term einzeln an und rechnen $SI = \int_a^b p(x) = I_0 + I_1 + I_2$. Der erste Term ist

$$\begin{aligned} I_0 &= \int_a^b dx f(a) \\ &= f(a) \int_a^b dx \\ &= f(a)(b - a) \\ &= 2h \cdot f(a) \end{aligned}$$

Der zweite Term ist

$$\begin{aligned} I_1 &= \int_a^b dx (x - a) \left(\frac{f(a+h) - f(a)}{h} \right) \\ &= \left(\frac{f(a+h) - f(a)}{h} \right) \frac{1}{2}(x - a)^2 |_a^b \\ &= \left(\frac{f(a+h) - f(a)}{h} \right) \frac{1}{2}(b - a)^2 \\ &= \left(\frac{f(a+h) - f(a)}{h} \right) \frac{1}{2}(2h)^2 \\ &= \left(\frac{f(a+h) - f(a)}{h} \right) 2h^2 \end{aligned}$$

und der dritte Term ist

$$\begin{aligned} I_2 &= \frac{\frac{f(b) - f(a+h)}{h} - \frac{f(a+h) - f(a)}{h}}{2h} \int_a^b dx (x - a)(x - a - h) \\ &= \frac{f(b) - 2f(a+h) + f(a)}{2h^2} \int_a^b dx \left(x^2 - 2ax - xh + a^2 + ah \right) \\ &= \frac{f(b) - 2f(a+h) + f(a)}{2h^2} \left(\frac{1}{3}x^3 - ax^2 - \frac{1}{2}x^2 h + a^2 x + ahx \right) |_a^b \\ &\dots \\ &= \frac{f(b) - 2f(a+h) + f(a)}{2h^2} \frac{2}{3}h^3 \end{aligned}$$

Setzen wir alles zusammen erhalten wir

$$\begin{aligned} SI &= f(a) \cdot 2h + \left(\frac{f(a+h) - f(a)}{h} \right) 2h^2 + \frac{f(b) - 2f(a+h) + f(a)}{2h^2} \frac{2}{3}h^3 \\ &= f(a) \cdot 2h + (f(a+h) - f(a)) 2h + \frac{f(b) - 2f(a+h) + f(a)}{3}h \\ &= \frac{h}{3}(6f(a) + 6f(a+h) - 6f(a) + f(b) - 2f(a+h) + f(a)) \\ &= \frac{h}{3}(f(a) + 4f(a+h) + f(b)) \end{aligned}$$

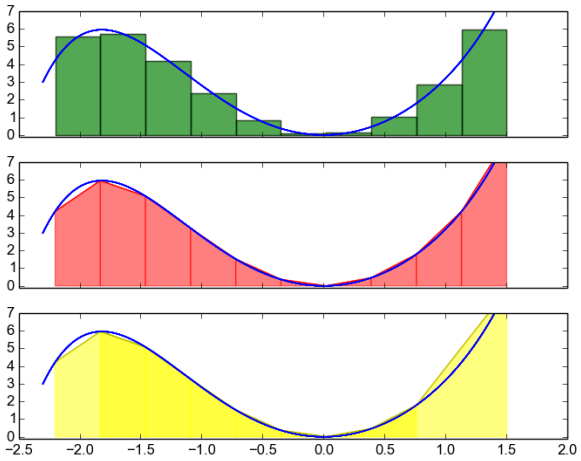

Abbildung 13: Integration der Funktion $f(x) = 0.2x^5 + 3x^2$ im Intervall$[-2.2, 1.5]$ durch Rechteckregel (grün), Trapezregel (rot) und Simpsonregel (gelb).

Im Prinzip lassen sich auch Polynome höheren Grades zur Integration verwenden, aber diese haben die Probleme, welche Interpolationspolynome hohen Grades eben haben (z.B. Überschwinger).

Integrationsformeln der Form

$$QI = \sum_{i=1}^{n} c_i f(x_i)$$

werden allgemein **Quadraturformeln** genannt.

5.2.2 Integrationsfehler

Eine Integrationsformel zur berechnung von $I = \int_a^b f(x)$ hat die Fehlerordnung k, wenn für alle Polynome vom Grad $k-1$ der Fehler verschwindet und für ein Polynom vom Grad k der Fehler ungleich null ist.

Wi schauen uns ein Beispiel an, inwelchem $a = 0$ und $b = a + 2h$, die Intervalllänge also $2h$ ist. Mit dem Interpolationspolynom $p(x) = \sum_i c_i x^i$ nähern wir

$$\begin{aligned} I &= \int_0^2 h \cdot p(x)\, dx \\ &= c_0 2h + \frac{1}{2} c_1 (2h)^2 + \frac{1}{3} c_3 (2h)^3 + \frac{1}{4} c_4 (2h)^4 + \frac{1}{5} c_5 (2h)^5 + \dots \\ &= c_0 2h + c_1 2h^2 + \frac{8}{3} c_3 h^3 + 4 c_4 h^4 + \frac{32}{5} c_5 h^5 + \dots \end{aligned}$$

Für die Rechteckregel bekommen wir so (mit $n = 2$ und $x_0 = 0$; $x_1 = x_0 + h = h$)

$$\begin{aligned}
RI &= h \sum_{i=0}^{n-1} f\left(x_i + \frac{h}{2}\right) \\
&= h\left(f\left(\frac{h}{2}\right) + f\left(h + \frac{h}{2}\right)\right) \\
RI &= h\left(p\left(\frac{h}{2}\right) + p\left(\frac{3h}{2}\right)\right) \\
&= h2c_0\underbrace{}_{x^0} + hc_1\underbrace{\left(\frac{h}{2} + \frac{3h}{2}\right)}_{x^1} + hc_2\underbrace{\left(\frac{h^2}{4} + \frac{9h^2}{4}\right)}_{x^2} + \ldots \\
&= h2c_0 + c_1 2h^2 + \underbrace{\frac{5}{2}c_3 h^3}_{\text{Abweichung}}
\end{aligned}$$

Also ist die Fehlerordnung für die Rechteckregel $k = 2$.

Die Trapezregel hat wegen

$$\begin{aligned}
TI &= h\left(\frac{f(a)}{2} + \sum_{i=1}^{n-1} f(x_i) + \frac{f(b)}{2}\right) \\
&= h\left(\frac{p(0)}{2} + p(h) + \frac{p(2h)}{2}\right) \\
&= h\left[c_0\left(\frac{x_0^0}{2} + x_1^0 + \frac{x_2^0}{2}\right) + c_1\left(\frac{x_0^1}{2} + x_1^1 + \frac{x_2^1}{2}\right) + c_2\left(\frac{x_0^2}{2} + x_1^2 + \frac{x_2^2}{2}\right) + \ldots\right] \\
&= h\left[c_0\left(\frac{1}{2} + 1 + \frac{1}{2}\right) + c_1\left(\frac{0}{2} + h + h\right) + c_2\left(\frac{0}{2} + h^2 + \frac{4h^2}{2}\right) + \ldots\right] \\
&= h\left[2c_0 + c_1(2h) + c_2 3h^2 + \ldots\right] \\
&= 2hc_0 + c_1 2h^2 + \underbrace{c_2 3h^3}_{\text{Abweichung}} + \ldots
\end{aligned}$$

ebenfalls die Fehlerordnung $k = 2$.

Für die Simpsonregel bekommen wir mit

$$\begin{aligned}
SI &= \frac{h}{3}\left(f\left(a\right)+4f\left(a+h\right)+f\left(b\right)\right)\\
&= \frac{h}{3}\left(p\left(0\right)+4p\left(h\right)+p\left(2h\right)\right)\\
&= \frac{h}{3}\left[c_0\left(x_0^0+4x_1^0+x_2^0\right)+c_1\left(x_0^1+4x_1^1+x_2^1\right)+c_2\left(x_0^2+4x_1^2+x_2^2\right)\right]\\
&\quad +\frac{h}{3}\left[c_3\left(x_0^3+4x_1^3+x_2^3\right)+c_4\left(x_0^4+4x_1^4+x_2^4\right)\ldots\right]\\
&= \frac{h}{3}\left[c_0\left(1+4+1\right)+c_1\left(0+4h+2h\right)+c_2\left(0+4h^2+4h^2\right)\right]\\
&\quad +\frac{h}{3}\left[c_3\left(0+4h^3+8h^3\right)+c_4\left(0+4h^4+16h^4\right)\right]\\
&= \frac{h}{3}\left[6c_0+c_16h+c_28h^2+c_312h^3+c_420h^4\ldots\right]\\
&= 2hc_0+c_12h^2+\frac{8}{3}c_2h^3+4c_3h^4+\underbrace{\frac{20}{3}c_4h^5}_{\text{Abweichung}}
\end{aligned}$$

eine Fehlerordnung $k=4$.

Schauen wir uns noch ein Zahlenbeispiel an bis zu welchem Polynomgrad die Quadraturformlen exakt sind. Das Integrationsintervall sein $[-1,1]$ und I bezeichne das exakte Integral

p	I	RI	TI	SI
x^0	$1+1=2$	2	2	2
x^1	$-1+1=0$	$2\cdot 0=0$	$-1+1=0$	$\frac{1}{3}\left(-1+0+1\right)=0$
x^2	$\frac{1}{3}x^3\vert_{-1}^1=\frac{1}{3}-\left(-\frac{1}{3}\right)=\frac{2}{3}$	$2\cdot 0=0$	$\left(-1\right)^2+1^2=2$	$\frac{1}{3}\left(\left(-1\right)^2+0^2+1^2\right)=\frac{2}{3}$
x^3	$\frac{1}{4}x^4\vert_{-1}^1=\frac{1}{4}-\frac{1}{4}=0$	2	2	$\frac{1}{3}\left(\left(-1\right)^3+0^3+1^3\right)=0$
x^4	$\frac{1}{5}x^5\vert_{-1}^1=\frac{1}{5}-\left(-\frac{1}{5}\right)=\frac{2}{5}$	2	2	$\frac{1}{3}\left(\left(-1\right)^4+0^4+1^4\right)=\frac{2}{3}$

Auch hier sehen wir wieder die Fehlerordnung $k=2$ für die Rechteck und Trapezregel und $k=4$ für die Simpsonregel.

Die Größe des Fehlers lässt sich mit Hilfe einer Taylorentwicklung abschätzen

$$f\left(x\right)=f\left(x_0\right)+\left(x-x_0\right)f\prime\left(x_0\right)+\frac{1}{2}\left(x-x_0\right)^2f\prime\prime\left(x_0\right)+\frac{1}{6}\left(x-x_0\right)^3f\prime\prime\prime\left(x_0\right)+\frac{1}{24}\left(x-x_0\right)^4f^{(4)}\left(x_0\right)+\ldots$$

Der Fehler der genäherten Integrale normiert auf die Intervallbreite, ist etwa

$$\begin{aligned}
\frac{I-RI}{2h} &\approx \frac{\left|\frac{8}{3}c_2h^3-\frac{5}{2}c_2h^3\right|}{2h}=\frac{1}{12}c_2h^2\approx\frac{f\prime\prime}{24}\\
\frac{I-TI}{2h} &\approx \frac{\left|\frac{8}{3}c_2h^3-3c_2h^3\right|}{2h}=\frac{1}{6}c_2h^2\approx\frac{f\prime\prime}{12}\\
\frac{I-SI}{2h} &\approx \frac{\left|\frac{32}{5}c_4h^5-\frac{20}{3}c_4h^5\right|}{2h}=\frac{2}{15}c_4h^4\approx\frac{f^{(4)}}{180}
\end{aligned}$$

5.2.3 Gauß-Quadratur

Bisher haben wir die Stützstellen für die Integrationspolynome mit festem Abstand h gewählt. Man kann allerdings höhere Fehlerordnungen bei gleicher Ordnung des Polynims erreichen, wenn die Stützstellen optimal platziert werden.

Die allgemeine Integrationsformel mit n Stützstellen lautet

$$GI = \sum_{i=1}^{n} b_i f(x_i)$$

wobei die x_i di Positionen der Stützstellen sind und die b_i als deren Gewichte bezeichnet werden. Beide sind so zu wählen, dass die Fehlerordnung optimal wird.

Wr betrachten nur wieder das Intervall $[-1, 1]$.

Für nur eine Stützstelle, also $n = 1$ haben wir zwei Freiheitsgrade, b_1 und x_1. Fordert man Exaktheit, Ordnung für Ordnung bekommen wir mit

$f(x)$	I	GI
$x^0 = 1$	2	b_1
x^1	0	$b_1 x_1$

$b_1 = 2$ und $x_1 = 0$.

Für den Fall $n = 2$ haben wir mit $GI = b_1 f(x_1) + b_2 f(x_2)$ vier Freiheitsgrade zu bestimmen. Aus der Forderung $GI = I$ im Intervall $[-1, 1]$erhalten wir ein lineares Gleichungssystem. Aus

$f(x)$	I	GI	
$x^0 = 1$	2	$b_1 + b_2$	Wir wählen $b_1 = b_2 = 1$
x^1	0	$b_1 x_1 + b_2 x_2$	Mit $b_1 x_1 = b_2 x_2$ ist dann $x_1 = -x_2$
x^2	$\frac{2}{3}$	$b_1 x_1^2 + b_2 x_2^2$	Aus $\frac{2}{3} = b_1 x_1^2 + b_2 x_2^2$ wird dann $\frac{2}{3} = 2 b_1 x_1^2$ und $x_{1,2} = \pm\frac{1}{\sqrt{3}}$
x^3	0	$b_1 x_1^3 + b_2 x_2^3$	Wird auch mit $x_1 = -x_2$ und $b_1 = b_2$ erfüllt.

Allgemein sind die optimalen n Stützstellen die Nullstellen der Legendre-Polynome

$$p_n(x) \quad = \quad \frac{1}{2^n n!} \frac{d^n}{dx^n} \left(x^2 - 1\right)^n$$

Die ersten drei lauten

$$G_1 I = 2 f(0)$$

$$G_2 I = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

$$G_3 I = \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right)$$

Der Vorteil liegt nun darin, dass für eine gewählte Anzahl Stützstellen deren Position und Gewichte einmal bestimmt werden und damit im Prinzip beliebige Funktionen interpoliert werden können. Die Transformation auf ein Intervall $[a, b]$ erfolgt mit

$$x = x(u) = \frac{b-a}{2} u + \frac{b+a}{2}$$

wobei $x - 1 = a$ und $x + 1 = b$ sind. Das Integral lautet dann

$$\int_a^b f(x)\, dx = \frac{b-a}{2} \int_{-1}^{1} \underbrace{\frac{b-a}{2} u + \frac{b+a}{2}}_{g(u)} du = \frac{b-a}{2} \int_{-1}^{1} g(u)\, du$$

Und die (n)-te Ableitung bekommt man mit

$$g^{(n)}(u) = \left(\frac{b-a}{2}\right)^n f^{(n)}\left(\frac{b-a}{2} u + \frac{b+a}{2}\right)$$

6 Differenzialgleichungen

Differentialgleichungen sind mathematische Gleichungen für eine Funktion einer oder mehrerer Variablen, in der auch Ableitungen der Funktion vorkommen. Viele Naturgesetze werden durch Differentialgleichungen beschrieben. Man unterscheidet zwischen gewöhnlichen Differentialgleichungen, bei denen die Lösungsfunktion nur von einer Variablen abhängt, und partiellen Differentialgleichungen, bei denen die Lösungsfunktion von mehreren Variablen abhängt.

Zur weitren Unterscheidung von Differentialgleichungen sollen hier zunächst einige Beispiele gelistet werden, von denen uns ein paar später wieder begegnen.

$$F\left(x, y\left(x\right), y\prime\left(x\right), \ldots, y^{(n)}\left(x\right)\right) = 0$$

ist eine gewöhnliche Differentialgleichung n-ter Ordnung. Die höchste vorkommende Ableitung ist die n-te. Oft werden solche Gleichungen geschrieben als

$$y^{(n)}\left(x\right) = F\left(x, y\left(x\right), y\prime\left(x\right), \ldots, y^{(n-1)}\left(x\right)\right)$$

Gewöhnliche lineare Differentialgleichungen 1. Ordnung beschreiben z.B. radioaktiven Zerfall (oder das Wachstum einer Bakterienpopulation) nach

$$f\prime\left(t\right) = \frac{1}{\tau} f\left(t\right)$$

Die Änderung der Menge an radioaktivem Material/der Größe der Bakterienpopulation hängt von der aktuellen Menge/Größe ab. Als Lösungsfunktion bietet sich eine e-Funktion an, da deren Ableitung die Funktion selbst nebst einem multiplikativem Faktor ist:

$$f\left(t\right) = f_0 \exp\left(t/\tau\right)$$

wobei f_0 eine zunächst noch unbestimmte Konstante ist. Um eine eindeutige Lösung zu erhalten muss der Funktionswert an einem bestimmten Punkt, z.b. $t = 0$ bekannt sein.

Gewöhnliche lineare Differentialgleichungen 2. Ordnung findet man in der Bewegungsgleichung des harmonischen Oszillators mit Masse m und Kraftkonstante k

$$mf\prime\prime\left(t\right) = -kf\left(t\right)$$

Der allgemeine Lösungsansatz ist eine Kombination aus Sinus und Kosinusfunktion, da deren zweite Ableitungen, bis auf einen Vorfaktor, wieder die Funktion selber ergeben

$$f\left(t\right))A\sin\left(\sqrt{\frac{k}{m}}t\right) + B\cos\left(\sqrt{\frac{k}{m}}t\right)$$

Zur eindeutigen Lösung werden hier zwei Anfangswerte, $f\left(0\right)$ und $f\prime\left(0\right)$ benötigt.

Gewöhnliche nicht-lineare Differentialgleichungen 2. Ordnung heisst eine Differentialgleichung, in welcher die Funktion selber nicht-linear auftritt und höchstens die zweite Ableitung auftaucht. Ein Beispiel ist die sich aus der Poisson-Gleichung

$$\Phi\prime\prime\left(z\right) = -\rho\left(z\right)/\epsilon$$

und der Boltzmann-Gleichung

$$\rho\left(z\right) = \rho_0 \exp\left(-q\Phi\left(z\right)/k_B T\right)$$

ergebende Poisson-Boltzmann-Gleichung

$$\Phi\prime\prime(z) = -\frac{\rho_0}{\epsilon}\exp\left(-q\Phi(z)/k_BT\right)$$

zur Beschreibung von elektrostatischen Wechselwirkungen worin $\Phi(z)$ das elektrostatische Potential, $\rho(z)$ die Ladungsdichte, ϵ die dielektrische Leitfähigkeit, q die Ladung, k_B die Boltzmannkonstante und T die Temperatur sind. $E = q(z)\Phi(z)$ ist die elektrostatische Energie. Die Lösungsfunktion $\Phi(z)$ steht hier im Exponenten, deswegen handelt es sich um eine nicht-lineare Differentialgleichung. In der Praxis wird oft zur Vereinfachung eine linearisierte Form davon benutzt, in welcher der Exponent in einer Potenzreihe entwickelt und diesem nach dem linearen Term abgebrochen wird.

Partielle lineare Differentialgleichungen findet man zur Beschreibung der Wärmeleitung in einer Dimension

$$\frac{\partial}{\partial t}u(z,t) = \lambda\frac{\partial^2}{\partial z^2}u(z,t)$$

Die Variablen sind hier Ort, z, und Zeit, t, λ bezeichnet die Wärmeleitfähigkeit.

Ein weiteres Beispiel ist die Schrödingergleichung

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left(-\frac{\hbar^2}{2m}\Delta + V(\mathbf{r},t)\right)\Psi(\mathbf{r},t)$$

mit dem Laplace-Operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.

Es gibt auch nicht-lineare partielle Differentialgleichungen, Stochastische Differentialgleichungen, Integro-Differentialgleichungen, Systeme von Differentialgleichungen, etc. etc.

6.1 Gewöhnliche Differentialgleichungen

6.1.1 Anfangswertprobleme

Gegeben ist eine Funktion $F : \mathbb{R} \to \mathbb{R}$ im Intervall $[a,b]$ und ein Anfangswert y_0. Gesucht ist die Funktion $y : [a,b] \to \mathbb{R}$, welche

$$y\prime(t) = F(t, y(t))$$

für alle $t \in [a,b]$ und $y(0) = y_0$ erfüllt.

Ein Lösungsansatz führt über das Richtungsfeld. Wir können in einem y,t-Diagramm an vorgegebenen Punkten $(t, y(t))$ die Steigung (Richtung) in diesem Punkt eintragen.

Im Beispiel $y\prime(t) = \underbrace{t^2 + y(t)}_{F}$ haben wir so für $(t=0, y=1)$die Steigung $y\prime(t) = 1$ und für $(t=0, y=-1)$die Steigung $y\prime(t) = -1$. Entsprechend ist bei $(t=1, y=0)$ und $(t=-1, y=0)$ $y\prime(t) = 1$, aber bei $(t=\pm 2, y=0)$ ist $y\prime(t) = 4$. Im Punkt $(t=-1, y=-1)$ ist $y\prime(t) = 0$, im Punkt $(t=-1, y=1)$ ist $y\prime(t) = 2$, usw. usf.

Die gesuchte Lösungskurve ist in jedem Punkt tangential zu, Richtungsfeld. ine Grundidee der numerischen Lösungen ist es also, dem Richtungsfeld (den Tangenten) so genau wie möglich zu folgen.

Das Euler-Verfahren verfolgt diese Idee, indem von einem Startwert $y(t=0)$ ausgehend der Tangente für ein kleines Stück gefolgt wird

$$y(t) + hy\prime(t) = y(t+h)$$

Das ist genau eine lineare Näherung der Funktion (oder eine Taylorentwicklung erster Ordnung).

Aus der Differentialgleichung wissen wir, dass $y\prime(t) = F(t, y(t))$ also ist

$$y(t) + hF(t, y(t)) = y(t+h)$$

Von der neuen Stelle $y(t+h)$ gehen wir wieder gemäß der aktuellen Steigung $y\prime(t+h)$ ein Stückchen weiter usw. Der Algorithums ist dann einfach

$$y_{n+1} = y_n + hF(t_n, y(t_n)) = y_n + hy\prime_n$$

und startet mit

$$y_1 = y_0 + hF(t_0, y(t_0)) = y_0 + hy\prime_0$$

Fehlerabschätzung Sei $Z(t)$ die exakte Lösung des Anfangswertproblems $Z\prime(t) = F(t, Z)$ mit $Z(t_n) = y_n$. Sei weiter y_{n+1} der numerisch ermittelte Wert $y(t_n + h)$. Der *lokale Fehler* ist dann definiert als

$$\phi(t_n, h) = Z(t_n + h) - y(t_n + h)$$

Ein Verfahren hat die *Konsistenzordnung* p falls gilt

$$\left|\phi(t_n, h) \leq ch^{p+1}\right|$$

für genügend kleine h und eine Konstante $c > 0$.

Das Euler-Verfahren hat mit $\phi(t_n, h) \approx \frac{h^2}{2} y\prime\prime(t_n)$, d.h. der Fehler lässt sich durch das nächste Glied in der Taylorentwicklung abschätzen, die Fehlerordnung $p = 1$.

Sei weiter $Z(t_0) = y_0$ der Anfangswert und y_n die mit n Schritten berechnete numerische Näherung an der Stelle $t_n = t_0 + nh$. Dann ist der Gesamtfehler oder *globale Fehler* $Z(t_n) - y_n$.

Das Verfahren hat die *Konvergenzordnung* p falls gilt

$$|Z(t_n) - y_n| \leq ch^p$$

Für das Eulerverfahren ist $|Z(t_n) - y_n| \leq \frac{h}{2} \max|y\prime\prime(t)| \frac{\exp(c(t_n - t_0) - 1)}{c}$.

Mehrstufige Verfahren dienen dazu bessere Fehlerordnungen zu erzielen. Eine Möglichkeit ist die Funktion an der halben Schrittweite auszuwerten. Dort ist mit

$$F\left(t_n + \frac{h}{2}, y\left(t_n + \frac{h}{2}\right)\right) = \frac{y(t_{n+1}) - y(t_n)}{h}$$

die lineare Näherung natürlich mindestens so gut wie für $F(t_n + h, y(t_n + h))$.

Wir brauchen aber zunächst

$$y\left(t_n + \frac{h}{2}\right) = y(t_n) + \frac{h}{2} F(t_n, y(t_n))$$

machen also einen halben Euler-Schritt. Zusammen ist das

$$F\left(t_n + \frac{h}{2}, y(t_n) + \frac{h}{2} F(t_n, y(t_n))\right) = \frac{y(t_{n+1}) - y(t_n)}{h}$$

Der Integrationsschritt lautet also

$$y(t_{n+1}) = y(t_n) + hF\left(t_n + \frac{h}{2}, y(t_n) + \frac{h}{2} F(t_n, y(t_n))\right)$$

Wir können dieses Zweischrittverfahren formal anders aufschreiben als

$$\begin{aligned} k_1 &= F(t_n, y(t_n)) \\ k_2 &= F\left(t_n + \frac{h}{2}, y(t_n) + \frac{h}{2} k_1\right) \\ y_{n+1} &= y_n + hk_2 \end{aligned}$$

Dieses Verfahren hat Konsistenz- und Konvergenzordnung $p = 2$.

Mehrstufige Runge-Kutta-Verfahren lassen sich allgemein schreiben als

$$k_i = F\left(t_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{ij}, k_j\right)$$

und Integrationsvorschrift

$$y_{n+1} = y_n + h\sum_{j=1}^{s} b_j k_j$$

für $i = 1, \ldots, s$ mit a_{ij}, b_i, c_j Koeffizienten die so gewählt sind, dass die Konvergenzordnung mögichst hoch ist.

Das klassische Runge-Kutta-Verfahren ist vierstufig mit

$$\begin{aligned} k_1 &= F(t_n, y_n) \\ k_2 &= F\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= F\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= F(t_n + h, y_n + hk_3) \end{aligned}$$

sowie

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Es hat die Konsistenz- und Konvergenzordnung $p = 4$.

Differentialgleichungen höherer Ordnung können als System von Differentialgleichungen erster Ordnung geschrieben werden. Nehmen wir die Bewegungsgleichung des harmonischen Oszillators als Beispiel

$$my"(t) = -ky(t)$$

Führen wir die Geschwindigkeit ein ist

$$\begin{aligned} y'(t) &= v(t) \\ v'(t) &= -\frac{k}{m}y(t) \end{aligned}$$

Mit zwei Anfangswerten, $y(0)$ und $v(0)$ lässt sich die Integration dann mit Euler- oder Runge-Kutta-Verfahren durchführen.

Man rechnet mit Zeitschritt Δt

$$\begin{aligned} v_{n+1} &= v_n + \Delta t \cdot v'_n = v_n - \Delta t \cdot \frac{k}{m}y_n \\ y_{n+1} &= y_n + \Delta t \cdot y'_n = y_n + \Delta t \cdot v_n \end{aligned}$$

6.1.2 Randwertprobleme

Eine Klasse von physikalischen Problemen sind sogenannte Randwertprobleme. In diesen sind nicht nur Anfangswerte vorgegeben sondern auch weitere Werte am „anderen Rand“, also z.B. zu einem Zeitpunkt $t_1 = a$. Für Differentialgleichungen 2. Ordnung, $y"(t) = F(t, y(t), y'(t))$, gibt es folgende mögliche Kombinationen von Randbedingungen

- $y(0) = y_0$ und $y(t_1) = y_1$

Abbildung 14: Typische Struktur von $\psi(x)$ zum ersten Eigenwert.

- $y'(0) = v_0$ und $y(t_1) = y_1$
- $y(0) = y_0$ und $y'(t_1) = v_1$
- $y'(0) = v_0$ und $y'(t_1) = v_1$

Die Lösung erfolg mittels des Schießverfahrens. Hiebrei wandelt man zunächst die Differentialgleichung 2. Ordnung in ein System aus Differentialgleichungen 1. Ordnung um. Für dieses System löst man ein Anfangswertproblem mit gewähltem Startwert für den unbekannten Anfangswert, z.B. $\tilde{v}_0$, wenn y_0 und y_1 gegeben sind. Dann integriert man bis $t = t_1$ und erhält ein $\tilde{y}_1$, welches im Allgemeinen von y_1 verschieden ist.

Nun wird $\tilde{v}_0$ so variiert bis $\tilde{y}_1(\tilde{v}_0) - y_1 = 0$ ist.

Wir haben es also nun mit einem Nullstellenproblem zu tun, das z.B. mit dem Bisektionsverfahren oder dem Sekanten(Newton-)verfahren zu lösen ist. Zur Integration des Anfangswertproblems wird am besten das Runge-Kutta-Verfahren verwendet.

6.1.3 Eigenwertprobleme bei Differentialgleichungen

Die Gleichung für stehende Transversalwellen einer elastischen Federkette (s. auch Abschnitt 6.2.1 für die Herleitung) ist

$$u"(x) + \kappa^2 u(x) = 0$$

mit typischen Randbedingungen $u(0) = u(L) = 0$, d.h. die Kette ist an den Rändern eingespannt. Die möglichen Lösungen der Differentialgleichung unter den gegebenen Randbedingungen bestimmen die Eigenwerte κ^2. In diesem Beispiel lässt sich die Lösung anaytisch finden als

$$u(x) = A \sin(\kappa x)$$

mit

$$\kappa^2 = \left(\frac{n\pi}{L}\right)^2$$

und $n = 1, 2, 3 \ldots$

Generell ließe sich das Problem mittels Schießverfahren lösen. Man startet bei $u(0) = 0$ und $u'(0) = v_0$ mit einem Schätzwert $\tilde{\kappa}$. Durch Integration wird dann $u\left(L, \tilde{\kappa}^2\right)$ bestimmt und $\tilde{\kappa}$ solange variiert bis wie gefordert $u\left(L, \tilde{\kappa}^2\right) = 0$.

Der gewählte Startwert v_0 legt nur die Amplitude A von $u(x)$ fest und ist zur Bestimmung der Eigenwerte $\tilde{\kappa}$ irrelevant.

Die zeitunabhängige Schrödinger-Gleichung in 1D lautet

$$-\frac{\hbar}{2m}\frac{d^2}{dx^2}\psi(x) + V(x)\psi(x) = E\psi(x)$$

wobei m die Teilchenmasse und $\hbar$ die Planck-konstante sind. $V(x)$ ist das Potential und E der Energieeigenwert. Gegeben seien $V(x)$, sowie die Randbedingungen $\psi(x) = 0$ für $|x| \to \infty$ (gebundener Zustand).

Formen wir leicht um so ist

$$\psi"(x) + \frac{2m}{\hbar}[E - V(x)]\psi(x) = 0$$

an den Stellen mit $V(x) = E$ ist also $\psi"(x) = 0$, wir haben hier Wendepunkte (s. 14).

Zur Lösung des Eigenwertproblems startet man nun von $x_0 = \pm\infty$ (in der Praxis von sehr großen Startwerten), d.h. jeweils von links und von rechts um die Integrationsfehler klein zu halten[9]. Mit $\psi(x_0) = \psi(-x_0) = 0$ und einem Schätzwert, $\tilde{E}$, für E wird nun von beiden Seiten bis zum Punkt $V(x_R) = \tilde{E}$ integriert, wobei x_R den rechten Schnittpunkt bezeichnet[10]. Um die Kontinuität der Wellenfunktion $\psi(x)$ zu gewährleisten muss $\psi_L(x_R) = \psi_R(x_R)$, d.h. der Funktionswert von links oder rechts kommend muss gleich sein (entsprechend bei x_L). Das lässt sich leicht durch Skalierung erreichen mit

$$\psi_L(x) = \psi_L(x) \frac{\psi_R(x_R)}{\psi_L(x_R)}$$

Außerdem müssen aber auch die ersten Ableitungen von rechts oder links kommend gleich sein, also

$$f\left(\tilde{E}\right) = \psi'_L(x_R) - \psi'_R(x_R) = \frac{\psi_L(x_R+h) - \psi_L(x_R-h) - \psi_R(x_R+h) + \psi_R(x_R-h)}{2h} = 0$$

Das wird im Allgemeinen zunächst nicht gegeben sein. Man variiert dann also $\tilde{E}$ solange bis $f\left(\tilde{E}\right) = 0$, löst also wieder ein Nullstellenproblem. Durch verschiedenen Startwerte von $\tilde{E}$ lässt sich das vollständige Spektrum bestimmen.

Der Startwert ψ'_0 für die Lösung des Anfangswertproblems bestimmt wieder die Amplitude und ist irrelevant, wenn die Wellenfunktion normiert wird.

Das matching $\psi'_L(x_R) - \psi'_R(x_R)$ wird bei x_R oder x_L , also $\psi"(x) = 0$ durchgeführt. Dadurch tretenKorrekturen zur Differenzenformel erst in h^3 auf, die numerische Stabilität ist also höher.

[9] Zur Erinnerung: die Additionsreihenfolge sollte im Computer immer so sein, das ähnlich große Zahlen zunächst addiert werden. Hier heißt es also erst die kleinen Zahlen (ψ ist nahe Null) aufzuaddieren.

[10] Man kann natürlich auch den linken Schnittpunkt x_L nehmen. Bei symmetrischen Potentialen kann man sogar von rechts zum rechten und von links zum linken Schnittpunkt integrieren.

6.2 Partielle Differentialgleichungen

6.2.1 Homogene Differentialgleichungen

Als Beispiel für eine partielle Differentialgleichung leiten wir die Gleichung für Transveralwellen in einer Federkette her.

Gegeben seien $N+1$ Kugeln mit Masse m, die über N Federn der Stärke k miteinander verbunden sind. Das Gesamtpotential der Federkette setzt sich zusammen aus den Einzelpotentialen nach

$$V_{Kette} = \frac{k}{2} \sum_{i=0}^{N-1} (u_i - u_{i+1})^2$$

wobei U_i die Transversalauslenkung der i-ten Kugel ist. Die Bewegungsgleichung einer Kugel i ist wie beim harmonischen Oszillator

$$m \frac{\partial^2}{\partial t^2} u_i(t) = F_i = -\frac{\partial V_{Kette}}{\partial u_i}$$

Setzen wir V_{Kette} ein ist das

$$\begin{aligned} m \frac{\partial^2}{\partial t^2} u_i(t) &= -\frac{k}{2} \frac{\partial}{\partial u_i} \sum_{i=0}^{N-1} (u_i - u_{i+1})^2 \\ &= -\frac{k}{2} \frac{\partial}{\partial u_i} \left[(u_{i-1} - u_i)^2 + (u_i - u_{i+1})^2 \right] \\ &= -\frac{k}{2} \left[-2(u_{i-1} - u_i) + 2(u_i - u_{i+1}) \right] \\ &= k \left[u_{i-1} - 2u_i + u_{i+1} \right] \end{aligned}$$

Der laterale Abstand zwischen den Kugeln sei a. Dann ergibt sich mit

$$\frac{m}{a} \frac{\partial^2}{\partial t^2} u_i = ka \frac{u_{i-1} - 2u_i + u_{i+1}}{a^2}$$

auf der rechten Seite gerade die zweite Ableitung nach dem lateralen Ort x als Differenzenformel (vgl. Gl. 45) und wir schreiben (jetzt wieder für alle Kugeln)

$$\frac{m}{a} \frac{\partial^2}{\partial t^2} u(x,t) = ka \frac{\partial^2}{\partial x^2} u(x,t)$$

Ein wenig umsortiert wird daraus

$$\frac{\partial^2}{\partial t^2} u(x,t) - c^2 \frac{\partial^2}{\partial x^2} u(x,t) = 0 \tag{46}$$

mit der Wellengeschwindigkeit $c = \sqrt{\frac{ka^2}{m}}$. [11] Ein Ansatz zur Lösung von Gleichung 46 ist

$$u(x,t) = f(x \pm ct)$$

wobei $f(\circ)$ eine beliebige, propagierende Funktion ist. Ein Punkt der Welle (z.B der Berg), der zum Zeipunkt $t = 0$ ist zum Zeitpunkt $t = x'/c$ nach x' „weitergewandert“.

Formal lässt sich die Gleichung über einen Separationsansatz

$$u(x,t) = \psi(x)\, \phi(t)$$

[11] kurze Dimensionsbetrachtung: $\sqrt{\frac{N/m \cdot m^2}{kg}} = \sqrt{\frac{kg \cdot m/ms^2 \cdot m^2}{kg}} = \sqrt{\frac{m^2}{s^2}} = m/s$

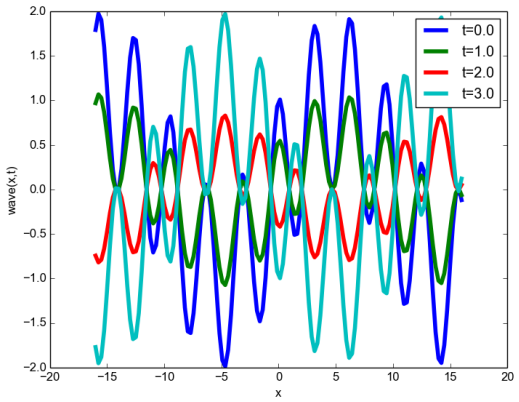

Abbildung 15: Welle $\psi(x) = \cos(2x) + \sin(\sqrt{\pi/L}x)$ zu verschiedenen Zeitpunkten, hier propagiert durch $\phi(t) = \cos(t)$.

lösen. Dann ist

$$\begin{aligned} \frac{\partial^2}{\partial t^2}\psi(x)\,\phi(t) - c^2\frac{\partial^2}{\partial x^2}\psi(x)\,\phi(t) &= 0 \\ \frac{\ddot{\phi}(t)}{\phi(t)} &= c^2\frac{\psi''(x)}{\psi(x)} \end{aligned}$$

Da die rechte und linke Seite jeweils bezüglich der anderen Variablen konstant sind, muss ferner

$$\begin{aligned} \frac{\ddot{\phi}(t)}{\phi(t)} &= -c^2\kappa^2 \\ \frac{\psi''(x)}{\psi(x)} &= -\kappa^2 \end{aligned}$$

und damit erhalten wir zwei entkoppelte Eigenwertgleichungen

$$\ddot{\phi}(t) + c^2\kappa^2\phi(t) = 0 \tag{47}$$

$$\psi''(x) + \kappa^2\psi(x) = 0 \tag{48}$$

Die zeitunabhängige Gleichung 48 für stehende Wellen mit der allgemeinen Lösung

$$\psi(x) = A\sin(\kappa x) + B\cos(\kappa x)$$

haben wir schon mit den Randbedingungen $\psi(0) = \psi(L) = 0$ gelöst zu $\psi(x) = \sin\kappa x$ mit $\kappa^2 = \left(\frac{n\pi}{L}\right)^2; n = 1, 2, \ldots$.

Die Lösung für den zeitabhängigen Teil ist allgemein

$$\phi(t) = C\sin(\omega t) + D\cos(\omega t) \tag{49}$$

mit $\omega^2 = c^2\kappa^2 = \left(\frac{cn\pi}{L}\right)^2; n = 1, 2, \ldots$

Zusammengesetzt ergibt das die allgemeinste Lösung

$$u(x,t) = \sum_{n=1}^{\infty}\left(C_n\sin(\omega t) + D_n\cos(\omega t)\right)\sin(\kappa_n x)$$

wobei die C_n und D_n aus den initialen ($t = 0$) Orten und Geschwindigkeiten bestimmt werden:

$$u(x,0) = \sum_{n=1}^{\infty} D_n \sin(\kappa_n x)$$

$$v(x,0) = \dot{u}(x,0) = \sum_{n=1}^{\infty} \omega C_n \sin(\kappa_n x)$$

Die Eigenfunktionen sind alle orthogonal zueinander, d.h.

$$\int_0^L dx \sin(\kappa_m x) \sin(\kappa_n x) = \delta_{mn} \frac{L}{2}$$

Dann ist auch für die Welle zum Zeipunkt $t = 0$ mit allen anderen Wellen zu diesem Zeitpunkt

$$\begin{aligned} \int_0^L dx u(x,0) \sin(\kappa_m x) &= \int_0^L dx \sum_{n=1}^{\infty} D_n \sin(\kappa_n x) \sin(\kappa_m x) \\ &= \sum_{n=1}^{\infty} \delta_{mn} D_m \frac{L}{2} \\ &= D_m \frac{L}{2} \end{aligned}$$

und damit

$$D_n = \frac{2}{L} \int_0^L dx u(x,0) \sin(\kappa_n x)$$

entsprechend erhält man

$$C_n = \frac{2}{L\omega_n} \int_0^L dx v(x,0) \sin(\kappa_n x)$$

Damit ist das Anfangswertproblem in t und das Randwertproblem in x gelöst.

6.2.2 Inhomogene Differentialgleichung

Der Separationsansatz ist auch für viele nicht mehr exakt lösbare Probleme nützlich. Als Beispiel betrachten wir eine Federkette, die in den Massen inhomogen ist mit einer stationären Dichteverteilung $\rho(x)$. Die Differentialgleichung dazu lautet

$$\rho(x) \frac{\partial^2}{\partial t^2} u(x,t) = T \frac{\partial^2}{\partial x^2} u(x,t) - \rho(x) g$$

worin $g = 9.8\text{m/s}$ die Erdbeschleunigung ist und $T = ka$ (Kraftkonstante mal Abstand der Kugeln). Die inhomogene Gleichung

$$\frac{\partial^2}{\partial t^2} u(x,t) = \frac{T}{\rho(x)} \frac{\partial^2}{\partial x^2} u(x,t) - g$$

löst man durch Superposition der allgemeinen homogenen Lösung ($g = 0$) und einer speziellen Lösung der inhomogenen Gleichung.

Wir machen einen Separationsansatz für die homogene Gleichung

$$\begin{aligned} u_h(x,t) &= \psi(x) \phi(t) \\ \psi(x) \ddot{\phi}(t) &= \frac{T}{\rho(x)} \psi''(x) \phi(t) \\ \frac{\ddot{\phi}(t)}{\phi(t)} &= \frac{T}{\rho(x)} \frac{\psi''(x)}{\psi(x)} = -\omega^2 \end{aligned}$$

Der ortsabhängige Teil

$$\psi''(x) + \frac{\omega^2 T}{\rho(x)}\psi(x) = 0$$

kann als Eigenwertproblem für beliebige Randwerte $\psi(0), \psi(L)$ mit Schießverfahren gelöst werden. Der zeitabhängige Teil

$$\ddot{\phi}(t) + \omega^2 \phi(t) = 0$$

kann exakt über $\phi(t) = A\sin(\kappa t) + B\cos(\kappa t)$ gelöst werden.

Als spezielle (partikuläre) Lösung des inhomogenen Problems kann die statische Lösung ($\frac{\partial u}{\partial t} = 0$) benutzt werden

$$T\frac{\partial^2}{\partial x^2}u_p(x) - \rho(x)\, g = 0$$

das ist ein normales Randwertproblem mit $u_p(0)$ und $u_p(L)$.

Die allgemeine Lösung

$$u(x,t) = u_h(x,t) + u_p(x)$$

kann dann wieder aus den Anfangsbedingungen $u(x,t) = 0$ konstruiert werden.

6.2.3 Partielle Differentialgleichung mit zeitlich veränderlichen Inhomogenitäten

Solche Differentialgleichungen beschreiben z.B. Partikeldiffusion, wobei die Partikel lokal erzeugt oder vernichtet werden. Auf die Gleiche Weise wird Wärmediffusion, Migration von Zellen, chemische Reaktions-Diffusionssysteme und vieles mehr beschrieben.

Die Lösung erfolgt über eine Diskretisierung der Zeit $(t, \delta t)$ und des Orts als ein Gitter , zunächst eindimensional. Dann bezeichnet $n_i(t)$ die Anzahl Partikel auf dem Gitterplatz i zum Zeitpunkt t. Die Diffusion wird dann so beschreiben, dass in einem Zeitschritt δt ein Partikel mit einer Wahrscheinlichkeit p einen Schritt auf einen benachbarten Gitterplatz $i-1$ oder $i+1$ macht oder mit Wahrscheinlichkeit $(1-p)$ verbleibt.

Damit ergibt sich für die Anzahl Partikel im Gitterplatz i zum neuen Zeitpunkt

$$n_i(t+\delta t) = \underbrace{(1-p)\, n_i(t)}_{\text{bleibt in } i} + \underbrace{\frac{p}{2} n_{i+1}(t)}_{\text{kommt von } i+1} + \underbrace{\frac{p}{2} n_{i-1}(t)}_{\text{kommt von } i-1} + \underbrace{S_i(t)}_{\text{erzeugt/vernichtet}} \tag{50}$$

Das formen wir so um, dass wir die Änderung pro Zeitschritt erhalten

$$\begin{aligned} n_i(t+\delta t) - n_i(t) &= \frac{p}{2}\left(-2n_i + n_{i+1}(t) + n_{i-1}(t)\right) + S_i(t) \\ \frac{n_i(t+\delta t) - n_i(t)}{\delta t} &= \frac{pa^2}{2\delta t}\left(\frac{-2n_i + n_{i+1}(t) + n_{i-1}(t)}{a^2}\right) + \frac{S_i(t)}{\delta t} \end{aligned}$$

wobei wir auf der rechten Seite mit dem Gitterabstand a erweitert haben. Auf der rechten Seite haben wir also in diskreter Form die erste Ableitung der Teilchenzahl nach der Zeit und auf der rechten Seite die zweite Ableitung nach dem Ort ($x = ia$)

$$\frac{\partial n(x,t)}{\partial t} = D\frac{\partial^2}{\partial x^2} n(x,t) + S(x,t) \tag{51}$$

Hierin sind $D = \frac{pa^2}{2\delta t}$ die Diffusionskonstante (die Wärmeleitfähigkeit ...), $S(x,t) = \frac{S_i(t)}{\delta t}$ die räumlich und zeitlich veränderliche Produktions- oder Vernichtungsrate, und $n(x,t)$ ist die Partikeldichte (das Temperaturfeld) zur Teit t. In drei Dimensionen sieht die Diffusionsgleichung so aus

$$\frac{\partial n(r,t)}{\partial t} = D\Delta n(r,t) + S(r,t) \tag{52}$$

mit $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.

Die numerische Lösung kann, mit Anfangsbedingungen $n_i(0)$ und $S_i(t)$ durch Integration mit Zeitschritten δt geschehen. Mittels Euler-Verfahren nach

$$n_i(t+\delta t) = n_i(t) + \delta t \frac{D}{a^2}\left(n_{i+1}(t) + n_{i-1}(t) - 2n_i(t)\right) + S_i(t)$$

erzielt man aber nur schlechte Konvergenz. Geschickter ist hier auch wieder ein Zweischrittberfahren und die rechte Seite für $t + \delta t/2$ zu berechnen

$$\begin{aligned}\frac{n_i(t+\delta t) - n_i(t)}{\delta t} &= \frac{D}{2a^2}\left(n_{i+1}(t) + n_{i-1}(t) - 2n_i(t) + n_{i+1}(t+\delta t) + n_{i-1}(t+\delta t) - 2n_i(t+\delta t)\right) \\ &\quad + \frac{1}{2}\left(S_i(t) + S_i(t+\delta t)\right)\end{aligned}$$

Sortieren führt zu

$$\begin{aligned}n_i(t+\delta t) - \frac{\delta t D}{2a^2}\left(n_{i+1}(t+\delta t) + n_{i-1}(t+\delta t) - 2n_i(t+\delta t)\right) &= \\ n_i(t) + \frac{\delta t D}{2a^2}\left(n_{i+1}(t) + n_{i-1}(t) - 2n_i(t)\right) + \frac{1}{2}\left(S_i(t) - S_i(t+\delta t)\right)\end{aligned}$$

Das führt zu $N-1$ Gleichungen für alle Gitterplätze $i = 1, \ldots, N-1$, wenn n_0 und n_N feste Randwerte haben.

$$\begin{pmatrix} 1 & & & & \\ -\frac{\delta t D}{2a^2} & 1 - \frac{\delta t D}{a^2} & -\frac{\delta t D}{2a^2} & & \\ & -\frac{\delta t D}{2a^2} & 1 - \frac{\delta t D}{a^2} & -\frac{\delta t D}{2a^2} & \\ & & -\frac{\delta t D}{2a^2} & 1 - \frac{\delta t D}{a^2} & -\frac{\delta t D}{2a^2} \\ & & & & 1 \end{pmatrix} \begin{pmatrix} n_0 \\ n_{i-1}(t+\delta t) \\ n_i(t+\delta t) \\ n_{i+1}(t+\delta t) \\ n_N \end{pmatrix} = \begin{pmatrix} b_0 \\ b_{i-1} \\ b_i \\ b_{i+1} \\ b_N \end{pmatrix}$$

wobei

$$b_i = n_i(t) + \frac{\delta t D}{2a^2}\left(n_{i+1}(t) + n_{i-1}(t) - 2n_i(t)\right) + \frac{1}{2}\left(S_i(t) - S_i(t+\delta t)\right)$$

Das Gleichungssystem kann z.B. mit dem Gaußalgorithmus gelöst werden.

Die formale Lösung führt über denn Green's-Funktions-Formalismus. Wir betrachten den Initialwert $n(r, t=0) = 0$. Nach der Green's Methode wird der Quellterm $S(r,t)$ in kurze Impulse zerhackt

$$S(r,t)\,\delta(t-\tau)$$

und die Lösung als Konvolution (Faltung) über die separaten Lösungen (Green's Funktionen) geschrieben

$$n(r,t) = \int_0^\infty d\tau \eta(r,t,\tau) \tag{53}$$

wobei wir η aus der Differentialgleichung

$$\frac{\partial \eta}{\partial t} = D\Delta\eta(r,t) + S(r,t)\,\delta(t-\tau) \tag{54}$$

erhalten haben. Die Integration über τ (s. Gleichung 53) liefert die alte Differentialgleichung 52 zurück:

$$\begin{aligned}\int_0^\infty d\tau \frac{\partial \eta(r,t,\tau)}{\partial t} &= \int_0^\infty d\tau D\Delta\eta(r,t) + \int_0^\infty d\tau S(r,t)\,\delta(t-\tau) \\ \frac{\partial}{\partial t}\int_0^\infty d\tau \eta(r,t,\tau) &= D\Delta n(r,t) + S(r,t) \\ \frac{\partial}{\partial t} n(r,t) &= D\Delta n(r,t) + S(r,t)\end{aligned}$$

Wir machen den Ansatz

$$\eta(r,t,\tau)\,\Theta(t-\tau)$$

um das inhomogene Problem in ein homogenes umzuwandeln. Hierbei ist

$$\Theta(x)=\begin{cases}1 & \text{für } x\geq 0\\ 0 & \text{für } x<0\end{cases}$$

die Heaviside (Stufen-)funktion. Für diese gilt

$$\Theta'(x)=\delta(x)$$

wegen

$$\int_{-a}^{b} dy\delta(y)=1 \quad \text{für} \quad -a<0 \text{ und } b>0$$

sowie

$$\int_{-a}^{b} dy\delta(y)=0 \quad \text{für} \quad -a<b<0 \text{ und } b>-a>0$$

ist

$$\int_{-\infty}^{x} dy\delta(y)=\Theta(x)$$

Also

$$\Theta(t-\tau)=\begin{cases}1 & \text{für } t-\tau\geq 0\Longrightarrow t>\tau\\ 0 & \text{für } t-\tau<0\Longrightarrow t<\tau\end{cases}$$

und nur der Bereicht für $t>\tau$, also nach dem Impuls ist überhaupt von Interesse.

Den Ansatz setzen wir in die partielle DGL ein

$$\begin{aligned}\frac{\partial}{\partial t}\eta(r,t,\tau)\,\Theta(t-\tau) &= \Theta(t-\tau)\,\frac{\partial}{\partial t}\eta(r,t,\tau)+\eta(r,t,\tau)\,\delta(t-\tau)\\ &= D\Delta\eta(r,t,\tau)\,\Theta(t-\tau)+S(r,\tau)\,\delta(t-\tau)\end{aligned}$$

wobei die erste Zeile der linken und die zweite Zeile der rechten Seite von 54 entspricht.

Von den delta-Termen bleibt jeweils $\eta(r,\tau,\tau)$ und $S(r,\tau)$. Damit erhalten wir eine homogene DGL

$$\frac{\partial\eta(r,t,\tau)}{\partial t}=D\Delta\eta(r,t,\tau)$$

mit dem Anfangswert

$$\eta(r,\tau,\tau)=S(r,\tau)$$

die lösbar ist.Wir machen dazu einen Separationsansatz

$$\eta(\mathbf{r},t)=\psi(\mathbf{r})\,\phi(t)$$

Damit wird die Differentialgleichung zu

$$\psi(\mathbf{r})\,\phi'(t)=D\phi(t)\,\Delta\psi(\mathbf{r})$$

also wieder einmal

$$\frac{\Delta\psi(\mathbf{r})}{\psi(\mathbf{r})}=\frac{\phi'(t)}{D\phi(t)}=-k^2=-|\mathbf{k}|^2$$

Daraus ergeben sich die entkoppelten Gleichungen

$$\phi'(t) + \omega\phi(t) = 0$$

mit $\omega = Dk^2$ und

$$\Delta\psi(\mathbf{r}) + k^2\psi(\mathbf{r}) = 0$$

Im isotropen unendlichen Raum sind die allgemeinen Lösungen dazu

$$\psi(\mathbf{r}) = A(\mathbf{k})\exp(i\mathbf{kr})$$

und

$$\phi(t) = B\exp(-\omega(t-\tau))$$

Zusammen ergibt das

$$\eta(\mathbf{r},t) = \int \frac{d^3k}{(2\pi)^3} A(\mathbf{k})\exp(i\mathbf{kr} - \omega(t-\tau))$$

$A(\mathbf{k})$ bestimmen wir aus der Anfangsbedingung

$$\eta(\mathbf{r},\tau) = S(\mathbf{r},\tau)$$

Ohne Abhängigkeit von t ist demnach

$$\begin{aligned}
S(\mathbf{r},\tau) &= \int \frac{d^3k}{(2\pi)^3} A(\mathbf{k})\exp(i\mathbf{kr}) \\
\int d^3r S(\mathbf{r},\tau)\exp(-i\mathbf{qr}) &= \int d^3r \int \frac{d^3k}{(2\pi)^3} A(\mathbf{k})\exp(i\mathbf{r}(\mathbf{k}-\mathbf{q})) \\
&= \int \frac{d^3k}{(2\pi)^3} A(\mathbf{k}) \int d^3r \exp(i\mathbf{r}(\mathbf{k}-\mathbf{q})) \\
&= \int \frac{d^3k}{(2\pi)^3} A(\mathbf{k}) (2\pi)^3 \delta(\mathbf{k}-\mathbf{q}) \\
\int d^3r S(\mathbf{r},\tau)\exp(-i\mathbf{qr}) &= A(\mathbf{q})
\end{aligned}$$

wobei wir $A(\mathbf{q})$ mittels Fouriertransformation „befreit“ haben.

Also

$$\eta(\mathbf{r},t) = \int \frac{d^3k}{(2\pi)^3} \int d^3r' S(\mathbf{r}',\tau)\exp(-i\mathbf{kr}')\exp(i\mathbf{kr} - \omega(t-\tau))$$

Dem Exponenten rücken wir jetzt noch mit einer quadratischen Ergänzung zu Leibe, da ja $\omega = Dk^2$. Wir haben also im Exponenten

$$-Dk^2(t-\tau) + i\mathbf{k}(\mathbf{r}-\mathbf{r}') = -D(t-\tau)\left[\mathbf{k} - \frac{1}{2}\frac{i(\mathbf{r}-\mathbf{r}')}{D(t-\tau)}\right]^2 - \frac{1}{4}\frac{(\mathbf{r}-\mathbf{r}')^2}{D(t-\tau)}$$

Setzen wir weiter $\mathbf{k}' = \mathbf{k} - \frac{1}{2}\frac{i(\mathbf{r}-\mathbf{r}')}{D(t-\tau)}$ bleibt also[12]

$$\begin{aligned}
\eta(\mathbf{r},t) &= \int \frac{d^3k'}{(2\pi)^3} \int d^3r' S(\mathbf{r}',\tau)\exp\left(-D(t-\tau)[\mathbf{k}']^2 - \frac{1}{4}\frac{(\mathbf{r}-\mathbf{r}')^2}{D(t-\tau)}\right) \\
&= \int d^3r' S(\mathbf{r}',\tau)\exp\left(-\frac{1}{4}\frac{(\mathbf{r}-\mathbf{r}')^2}{D(t-\tau)}\right) \int \frac{d^3k'}{(2\pi)^3}\exp\left(-D(t-\tau)[\mathbf{k}']^2\right)
\end{aligned}$$

[12]die Integrationsgrenzen müssen streng genommen auch verschoben werden, aber bei $\pm\infty$spielt das kaum eine Rolle

Das letzte Integral hat die Form

$$\int_{-\infty}^{\infty} dx \exp\left(-ax^2\right) = \sqrt{\frac{\pi}{a}}$$

heißt Gauß-Integral und hat eine bekannte Lösung.
Damit ergibt sich[13]

$$\eta\left(\mathbf{r},t\right) = \left(\frac{1}{4\pi D\left(t-\tau\right)}\right)^{\frac{3}{2}} \int d^3r' S\left(\mathbf{r}',\tau\right) \exp\left(-\frac{1}{4}\frac{\left(\mathbf{r}-\mathbf{r}'\right)^2}{D\left(t-\tau\right)}\right)$$

Das setzen wir ein in (hier nutzen wir wieder das Kausalitätsprinzip $\tau < t$, d.h. die Wirkung (Diffusion oder was immer) kann erst NACH der Ursache, d.h. dem Impuls S erfolgen)

$$\begin{aligned} n\left(r,t\right) &= \int_0^{\infty} d\tau \eta\left(r,t,\tau\right) \Theta\left(t-\tau\right) \\ &= \int_0^{t} d\tau \eta\left(r,t,\tau\right) \\ &= \int_0^{t} d\tau \int d^3r' S\left(\mathbf{r}',\tau\right) \left(\frac{1}{4\pi D\left(t-\tau\right)}\right)^{\frac{3}{2}} \exp\left(-\frac{1}{4}\frac{\left(\mathbf{r}-\mathbf{r}'\right)^2}{D\left(t-\tau\right)}\right) \end{aligned}$$

Und das letzte Integral muss für allgemeine $S\left(\mathbf{r},t\right)$ numerisch gelöst werden.

[13] Gauß-Integral hoch drei für alle drei Raumrichtungen

7 Fouriertransformation

7.1 Fourier-Reihe

Nach dem Fourier-Satz können periodische, stetige Funktionen $f(t) = f(t+T)$ mit Periode τ als Fourier-Reihe

$$f(t) = \sum_{n=-\infty}^{\infty} g_n \exp(-i\omega_o t) \tag{55}$$

darfestellt werden. Hierbei ist $\omega_0 = \frac{2\pi}{\tau}$ die fundamentale Winkelfrequenz. Mit der Eulerschen Funktion

$$\exp(ix) = \cos(x) + i\sin(x)$$

erkennen wir mit

$$f(t) = \sum_{n=-\infty}^{\infty} A_n \cos(nt) + B_n \sin(nx) \tag{56}$$

die allgemeinste Lösung, Gleichung 49, z.B. der Wellengleichung wieder. Die Fourierkoeffizienten sind

$$g_n = \frac{1}{T}\int_0^T dt f(t) \exp(-in\omega_0 t) \tag{57}$$

Das lässt sich herleiten, wenn man einen orthonormalen Basissatz mit Basisfunktionen

$$\phi_n(t) = \frac{1}{\sqrt{T}} \exp(-in\omega_0 t)$$

betrachtet. Wegen der Orthonormalität ist

$$\begin{aligned}\int_{t_0}^{t_0+T} \phi_m^*(t)\,\phi_n(t) &= \frac{1}{T}\int_{t_0}^{t_0+T} \exp(-i(m-n)\omega_0 t) \\ &= \delta_{mn} = \begin{cases} 1 & \text{für } m = n \\ 0 & \text{für } m \neq n \end{cases}\end{aligned}$$

Multiplizieren wir nun die Fourier-Reihe mit $\phi_m^*(t)$ und integrieren über t so ist

$$\begin{aligned}\int_{t_0}^{t_0+T} \phi_m^*(t) f(t) &= \int_{t_0}^{t_0+T} \phi_m^*(t) \sum_n^{\infty} g_n \sqrt{\tau}\phi_n(t) \\ &= \sqrt{T}\sum_n^{\infty} g_n \int_{t_0}^{t_0+T} \phi_m^*(t)\,\phi_n(t) \\ &= \sqrt{T}\sum_n^{\infty} g_n \delta_{mn} \\ &= \sqrt{T} g_m\end{aligned}$$

Und das nach dem Fourier-Koeffizienten g_m aufgelöst

$$\begin{aligned} g_m &= \frac{1}{\sqrt{T}}\int_{t_0}^{t_0+\tau} \phi_m^*(t) f(t) \\ &= \frac{1}{T}\int_{t_0}^{t_0+\tau} \exp(im\omega_0 t) f(t)\end{aligned}$$

wie in Gleichung 57 behauptet.

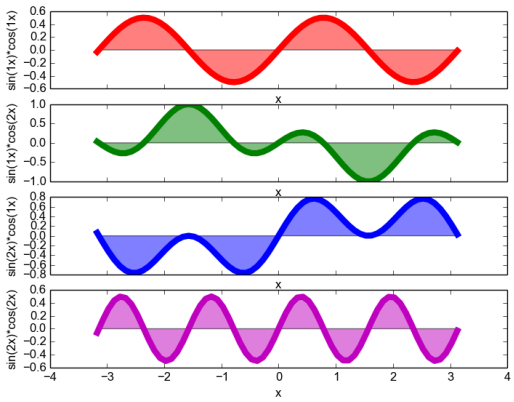

Abbildung 16: $\int_{-\pi}^{\pi} \cos(mx) \cdot \sin(nx)\, dx = 0$ für $m = 1, 2$ und $n = 1, 2$

7.2 Spektralanalyse

Wir erkennen also (oder erinnern uns): Von einer aus orthonormalen Basisfunktionen zusammengesetzten Funktion (z.B. nach dem Fourier-Satz), bleibt nach Multiplikation mit einer (komplex, konjugierten) Basisfunktion und Integration über eine volle Periode τ nur der Anteil erhalten, der durch die „anmultiplizierte" Basisfunktion beschrieben wird. Das Ausmaß, wieviel erhalten bleibt ist der Fourierkoeffizient. Wir machen uns das noch einmal mit $\sin(nx)$ und $\cos(nx)$ im Intervall $[-\pi, \pi]$ klar:

$\sin(1x)$ und $\cos(0x)$ sind orthonormal zueinander, denn $\int_{-\pi}^{\pi} \sin(x) \cdot 1 dx = 0$, ebenso[14] ist $\int_{-\pi}^{\pi} \cos(x) \cdot 1dx = 0$, also $\cos(1x)$ orthonormal zu $\cos(0x)$. Es ist aber auch $\int_{-\pi}^{\pi} \cos(x) \cdot \sin(x)\, dx = 0$, sowie $\int_{-\pi}^{\pi} \cos(nx) \cdot \sin(x)\, dx = 0$ und $\int_{-\pi}^{\pi} \cos(x) \cdot \sin(nx)\, dx = 0$. Sinus und Kosinus sind also hervorragend geeignete Basisfunktionen.

Um nun zu ermitteln, aus welchen Vielfachen n von x ein periodisches Signal zusammengesetzt ist, kann man nun nacheinander alle $n = 0, 1, 2 \ldots$ durchprobieren und an die zusammengesetzte Signalfunktion $f(x)$ entweder $\sin(nx)$ oder $\cos(nx)$ anmultiplizieren und integrieren[15]. Wenn das Integral nicht verschwindet, ist „etwas" $\sin(nx)$ im Signal enthalten, wobei „etwas" durch den Wert des Integrals gegeben ist.

7.3 Kontinuierliche Fouriertransformation

Schreiben wir nun

$$g_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \exp(-in\omega_0 t)\, dt$$

und lassen die Periodendauer $T \to \infty$ gehen, so dass $\omega_0 = \frac{2\pi}{T} \to 0$. damit wird aus der Fourier-Reihe ein Integral

$$f(t) \approx \int_{-\infty}^{\infty} g(n) \exp(-in\omega_0 t)\, dn$$

Definieren wir noch $\omega = n\omega_0$ und $\frac{d\omega}{dn} = \omega_0$, so dass $dn = \frac{d\omega}{\omega_0}$, dann ist

$$f(t) = \int_{-\infty}^{\infty} \frac{g(\omega)}{\omega_0} \exp(-i\omega t)\, d\omega$$

[14] Die Multiplikation mit $\sin(0) = 0$ und anschließendes integrieren zu 0 schenken wir uns.
[15] Da $\cos(x) = \sin\left(x + \frac{\pi}{2}\right)$ reicht z.B. der Sinus

Formal ist $\frac{g(\omega)}{\omega_0} = \frac{g(\omega)\tau}{2\pi}$ und mit $g(\omega)\tau = \tilde{f}(\omega)$ wird so

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \tilde{f}(\omega)\exp(-i\omega t)\,d\omega \tag{58}$$

und

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} f(t)\exp(i\omega t)\,dt \tag{59}$$

Fourier-Transformation und -Rücktransformation laufen also auf gleiche Weise ab, es sind nur der Vorfaktor $1/2\pi$ und das Vorzeichen im Exponenten zu beachten.

Machen wir nun einmal eine Rückwärtstransformation einer (gleichzeitigen) Vorwärtstransformation

$$\begin{aligned} f(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \tilde{f}(\omega)\exp(-i\omega t)\,d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp(-i\omega t)\int_{-\infty}^{\infty} f(t')\exp(i\omega t')\,dt'd\omega \\ &= \int_{-\infty}^{\infty} f(t')\,dt'\frac{1}{2\pi}\int_{-\infty}^{\infty}\exp(i\omega(t'-t))\,d\omega \end{aligned}$$

An dieser Stelle führen wir die Fourierdarstellung der Delta-Funktion ein

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\exp(i\omega t)\,d\omega \tag{60}$$

und

$$\tilde{\delta}(\omega) = \int_{-\infty}^{\infty}\delta(t)\exp(i\omega t)\,dt = 1$$

Also ist

$$\begin{aligned} f(t) &= \int_{-\infty}^{\infty} f(t')\,\delta(t'-t)\,dt' \\ &= f(t) \end{aligned}$$

7.4 Diskrete Fouriertransformation

Im Computer haben wir es mit diskreten Funktionen zu tun und häufig sind Signale (durch digitale Messtechnik) auch „nur" als diskrete Messwerte vorhanden.

Betrachten wir eine FUnktion $F(t)$ im Intervall $t \in [0,T]$. Dann kann man sich diskrete Daten erzeugen (oder vorstellen) durch Rasterung der Zeiten mit $t = k\frac{T}{N}$ mit $k = 0,1,\ldots N-1$. Insgesamt sind also N Datenpunkte zu transformieren.

Mit $\tau = \frac{T}{N}$ schreiben wir $t = \tau k$. Und aus die Fourierkomponenten werden dann zu

$$g_n = \frac{1}{N}\sum_{k=0}^{N-1} F(k\tau)\exp\left(\frac{ikn2\pi}{N}\right) \tag{61}$$

Die Rücktransformation erfolgt nach

$$f_k = f(\tau k) = \frac{1}{N}\sum_{j=0}^{N-1} g_j\exp\left(\frac{-ijk2\pi}{N}\right) \tag{62}$$

Jetzt überprüfen wir noch, ob Vorwärts-und Rückwärtstransformation sich wieder „aufheben“:

$$\begin{aligned} g_n &= \frac{1}{N}\sum_{k=0}^{N-1}\exp\left(\frac{ikn2\pi}{N}\right)\sum_{j=0}^{N-1} g_j \exp\left(\frac{-ijk2\pi}{N}\right) \\ &= \frac{1}{N}\sum_{j=0}^{N-1} g_j \sum_{k=0}^{N-1}\exp\left(\frac{ik2\pi}{N}\right)(n-j) \end{aligned}$$

Für $n = j$ ergibt die Exponentialfunktion $\exp\left(\frac{ik2\pi}{N}\right)(n-j) = 1$, für alle anderen $n \neq j$ hingegen 0. Also ist

$$\begin{aligned} g_n &= \frac{1}{N}\sum_{j=0}^{N-1} g_j \delta_{nj} \\ &= \frac{1}{N} N g_n \end{aligned}$$

7.5 Real-und Imaginärteil

Da die Eulersche Formel $\exp(ix) = \cos(x) + i\sin(x)$ uns die Aufteilung in Real-und Imaginärteil zeigt, liegt es nahe, auch die Fourier-transformierte in Real-und Imaginärteil zu zerlegen

$$f_k = \Re f_k + i\Im f_k \tag{63}$$

Für die Koeffizienten bedeutet dies

$$\begin{aligned} g_n &= \frac{1}{N}\sum_{k=0}^{N-1} \Re f_k + i\Im f_k \exp\left(\frac{ikn2\pi}{N}\right) \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\Re f_k \exp\left(\frac{ikn2\pi}{N}\right) + i\Im f_k \exp\left(\frac{ikn2\pi}{N}\right)\right] \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\Re f_k\left(\cos\frac{kn2\pi}{N} + i\sin\frac{kn2\pi}{N}\right) + i\Im f_k\left(\cos\frac{kn2\pi}{N} + i\sin\frac{kn2\pi}{N}\right)\right] \end{aligned}$$

Real-und Imaginärteil der Koeffizienten $g_n = \Re g_n + i\Im g_n$ sind dann:

$$\begin{aligned} \Re g_n &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\Re f_k \cos\left(\frac{kn2\pi}{N}\right) - \Im f_k \sin\left(\frac{kn2\pi}{N}\right)\right] \\ \Im g_n &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\Im f_k \cos\left(\frac{kn2\pi}{N}\right) + \Re f_k \sin\left(\frac{kn2\pi}{N}\right)\right] \end{aligned}$$

Entsprechend sind

$$\begin{aligned} \Re f_k &= \frac{1}{N}\sum_{j=0}^{N-1}\left[\Re g_j \cos\left(\frac{kn2\pi}{N}\right) + \Im g_j \sin\left(\frac{kn2\pi}{N}\right)\right] \\ \Im f_k &= \frac{1}{N}\sum_{j=0}^{N-1}\left[\Im g_j \cos\left(\frac{kn2\pi}{N}\right) - \Re g_j \sin\left(\frac{kn2\pi}{N}\right)\right] \end{aligned}$$

7.6 Fast-Fouriertransformation (FFT)

Sind $N = 2^M$ Summationsterme gegeben, so lassen sich diese mehrfach (M-mal) in zwei Hälften aufteilen. Eine solche Aufteilung kann z.B. durch Summation von 0 bis $\frac{N-1}{2}$ in der ersten Hälfte und dann von $\frac{N-1}{2}+1$ bis $N-1$ in der zweiten Hälfte erfolgen. Oder aber, man summiert einmal nur alle geraden und einmal alle ungeraden Indizes:

$$g_n = \frac{1}{N}\sum_{k=0}^{N-1} f_k \exp\left(\frac{i2\pi kn}{N}\right)$$

wird zu

$$\begin{aligned} g_n &= \frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(\frac{i2\pi\,(2m)\,n}{N}\right) + \frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(\frac{i2\pi\,(2m+1)\,n}{N}\right) \\ &= \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(\frac{i2\pi\,(2m)\,n}{N}\right)}_{X_n} + \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(\frac{i2\pi\,(2m)\,n}{N}\right)}_{Y_n} \exp\left(\frac{i2\pi n}{N}\right) \end{aligned}$$

wobei die erste Summe X_n die geraden $k = 2m = 2\cdot 0, 2\cdot 1, 2\cdot 2, \ldots = 0, 2, 4, \ldots$und die zweite Summe $Y_n \exp\left(\frac{i2\pi n}{N}\right)$ die ungeraden Indizes $k = 2m+1 = 2\cdot 0+1, 2\cdot 1+1, 2\cdot 2+1, \ldots = 1, 3, 5, \ldots$ berücksichtigt. Wir schreiben also

$$g_n = X_n + Y_n \exp\left(\frac{i2\pi}{N}\right)^n$$

Jetzt nutzen wir die Symmetrie und Periodizität der Basisfunktionen aus. Für $n < \frac{N}{2}$ ist nämlich $X_n = X_{n+\frac{N}{2}}$ und entsprechend $Y_n = Y_{n+\frac{N}{2}}$, wegen

$$\begin{aligned} \exp\left(\frac{i2\pi 2m\left(n+\frac{N}{2}\right)}{N}\right) &= \exp\left(\frac{i2\pi 2mn}{N}\right)\exp\left(\frac{i2\pi 2m\frac{N}{2}}{N}\right) \\ &= \exp\left(\frac{i2\pi 2mn}{N}\right)\exp\left(i2\pi m\right) \\ &= \exp\left(\frac{i2\pi 2mn}{N}\right)\left[\cos\left(2\pi m\right) + i\sin\left(2\pi m\right)\right] \\ &= \exp\left(\frac{i2\pi 2mn}{N}\right)\left[1+0\right] \end{aligned}$$

Außerdem ist

$$\begin{aligned} \exp\left(\frac{i2\pi\left(\frac{N}{2}\right)}{N}\right) &= \exp\left(i\pi\right) \\ &= \cos\left(\pi\right) + i\sin\left(\pi\right) \\ &= -1 \end{aligned}$$

Damit wird also

$$\begin{aligned} g_n &= X_n + Y_n \exp\left(\frac{i2\pi}{N}\right)^n \\ g_{n+\frac{N}{2}} &= X_n - Y_n \exp\left(\frac{i2\pi}{N}\right)^n \end{aligned}$$

für $n < \frac{N}{2}$. Das bedeutet, dass die Summen X_n und Y_n nur für die Hälfte der Koeffizienten berechnet werden muss, die andere Hälfte ergibt sich einfach dur Kombination mit Minuszeichen statt Pluszeichen. Das betreiben wir jetz weiter, indem wiederum jede der Summen X_n und Y_n aufgespalten werden zu

$$X_n = \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{4}-1} f_{2\cdot 2m=0,4,8\ldots} \exp\left(\frac{i2\pi\,(2m)\,n}{N/2}\right)}_{(XX)_n} + \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{4}-1} f_{2\cdot(2m+1)=2,6,10\ldots} \exp\left(\frac{i2\pi\,(2m)\,n}{N/2}\right)}_{(XY_n)} \left(\frac{i2\pi n}{N/2}\right)$$

sowie

$$Y_n = \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{4}-1} f_{2\cdot 2m+1=1,5,9\ldots} \exp\left(\frac{i2\pi\,(2m)\,n}{N/2}\right)}_{(YX)_n} + \underbrace{\frac{1}{N}\sum_{m=0}^{\frac{N}{4}-1} f_{2\cdot(2m+1)+1=3,7,\ldots} \exp\left(\frac{i2\pi\,(2m)\,n}{N/2}\right)}_{(YY_n)} \left(\frac{i2\pi n}{N/2}\right)$$

also mit $w = \exp(i2\pi/N)$

$$X_n = (XX)_n + (XY)_n\, w^{2n}$$

und

$$Y_n = (YX)_n + (YY)_n\, w^{2n}$$

Wieder gilt die Symmetrie, so dass

$$X_{n+\frac{N}{4}} = (XX)_n - (XY)_n\, w^{2n}$$

für $n < \frac{N}{4}$, entsprechend für Y_n. Nun kann die Summation solange aufgespalten werden, bis nur noch zwei Datenpunkte addiert werden müssen. Um hinterher die Summen wieder richtig zusammenzusetzen hilft eine Veranschaulichung in Bit-Sortierung. Für $N = 2^3$ ist $k = 0, 1, \ldots 7$ (der sogenannten „Twiddle-Faktor“ w^{jn} ist hier nicht geschrieben):

$$g_n$$

$$\begin{array}{cccc} & X_n & & Y_n \\ (XX)_n & (XY)_n & (YX)_n & (YY)_n \\ \underbrace{000}_{k=0},\underbrace{100}_{4} & \underbrace{010}_{k=2},\underbrace{110}_{6} & \underbrace{001}_{k=1},\underbrace{101}_{5} & \underbrace{011}_{k=3},\underbrace{111}_{7} \end{array}$$

vertausche Bit-Reihenfolge

$$\begin{array}{cccc} \underbrace{000}_{0},\underbrace{001}_{1} & \underbrace{010}_{2},\underbrace{011}_{3} & \underbrace{100}_{4},\underbrace{101}_{5} & \underbrace{110}_{6},\underbrace{111}_{7} \end{array}$$

Das heisst die Zusammensetzung läuft nach dem Bit-vertauschten von $k = 0, 1, 2, \ldots$ in Binärschreibweise.

8 Zufall

Für viele numerische Anwendungen, z.B. Simulationen, braucht man Zufallszahlen. Im Computer ist aber nichts wirklich zufällig, und nur mittels Software, kann man nur Pseudo-Zufallszahlen generieren.

Für Anwendungen, sie echte Zufallszahlen erfordern, (Passwörter, Zufallscodes oder auch stochastische Simulationen), werden noch Hardwareinformationen mitverwendet. Dies kann der Inhalt eines „zufällig" gewählten Stücks Arbeitsspeicher sein, die Mausbewegungen (als es noch Mäuse gab ;-)) der letzten paar Sekunden, oder der nicht-gepufferte Tastatureingaberhythmus. Atomare Zerfälle, Lava-Lampen und atmosphärisches Rauschen sind besonders beliebt (s. `random.org`) um echte Zufallszahlen zu erzeugen.

8.1 Zufallszahlgeneratoren

Wir befassen uns hier nur mit dem allereinfachsten Generator für Pseudo-Zufallszahlen, den linearen kongruenten Generatoren.

Da Pseudo-zufallszahlen sich irgendwann wiederholen, verlangt man eine möglichts lange Periode, für Integer-Zahlen mit 32bit also möglichst nahe an $2^{31}-1=2.147483647$

Die Sequenz der erzeugten Zufallszahlen soll möglichst zufällig sein. Was heißt das? Z.B. soll die Zweipunktkorrelation

$$C_l^{(2)}=\frac{1}{M-l}\sum_{i=1}^{M-l}\left(x_i-\bar{x}\right)\left(x_{i+l}-\bar{x}\right)$$

also die Korrelation einer Zufallszahl x_i mit der $l-$nächsten Zufallszahl x_{i+l} für $l\geq 1$ verschwinden. Mit $\bar{x}$ ist der Mittelwert aller x_i gemeint. Für $l=0$ ist die Korrelation immer von Null verschieden. Das sieht man an Zufallszaheln im Intervall $x\in[0,1]$. Der Mittelwert ist $\bar{x}=\frac{1}{M}\sum_{i=1}^{M}x_i\approx\int_0^1 dxp\left(x\right)$ Sind alle Zahlen gleich Wahrscheinlich, also $p\left(x\right)=1$ ist $\int_0^1 dxp\left(x\right)=0.5$. Und für die Korrelation mit $l=0$ erhalten wir

$$\begin{aligned}
C_0^{(2)} &= \frac{1}{M}\sum_{i=1}^{M}\left(x_i-\bar{x}\right)^2\\
&\approx \int_0^1 dxp\left(x\right)\left(x-\bar{x}\right)^2\\
&= \int_0^1 dxx^2-2x\bar{x}+\bar{x}^2\\
&= \frac{1}{3}-2\frac{1}{2}\bar{x}+\bar{x}^2\\
&= \frac{1}{3}-\frac{1}{2}+\frac{1}{4}\\
&= \frac{1}{12}
\end{aligned}$$

Außer der Korrelation ist die sogenannten „Mutual Information", MI, auch noch ein Maß für zwei (statistisch) unabhängige Zufallsvariablen (Zufallszahlen) x,y

$$MI=\sum_x\sum_y p\left(x,y\right)\log\frac{p\left(x,y\right)}{p\left(x\right)p\left(y\right)}$$

worin $p\left(x\right),p\left(y\right)$ die Einzel-Wahrscheinlichkeitsverteilungen für x bzw, y bedeuten und $p\left(x,y\right)$ ist die gemeinsame Wahrscheinlichkeitsverteilung.

Zuletzt sollten Zufallszahlengeneratoren noch schnell sein, weil häufig eine große Anzahl Zufallszahlen gebraucht wird.

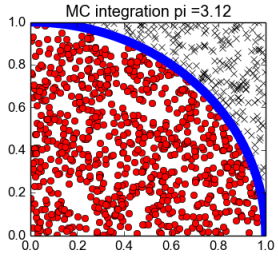

Abbildung 17: Hit (rot)-or-miss (schwarz)-Verfahren mit 1000 Zufallspunkten zur Bestimmung von π.

Lineare Kongruente Generatoren sind ein einfaches Schema, um PseudoZufallszahlen zu erzeugen. Die Sequenz wird mit der Iterations-Gleichung

$$x_{i+1} = (ax_i + b) \,\%c$$

erzeugt. Für 32bit-Zahlen wählt man $c = 2^{31} - 1$, um eine lange Periode zu erhalten. Aus ähnliche Überlegnungen ist die „magicnumber" $b = 0$. Und $a = 7^5 = 16807$ hat sich wohl auch als günstig erwiesen. Nun muss nur noch ein Startwert x_0, der „random seed" (der deutlich kkleiner als c sein muss) gewählt werden[16]. Eine beliebte Initialisierung ist die Systemzeit mit t_6 :Jahr, t_5 :Monat, t_4 :Tag, t_3 :Stunde, t_2 : Minute, t_1 :Sekunde so dass

$$x_0 = t_6 + 70\,(t_5 + 12\,(t_4 + 3)\,(t_3 + 23\,(t_2 + 59t_1)))$$

8.2 Monte-Carlo-Integration

Monte-Carlo-Verfahren basieren auf einer sehr großen Zahl von Zufallsexperimenten und kommen vor allem dann zu Einsatz, wenn keine analytische oder auch keine numerische deterministische Lösung möglich oder ineffizient ist. Letzteres ist insbesondere bei Problemen in hohen Dimensionane der Fall.

Zur veranschaulichung der Monte-Carlo-Integration betrachten wir die Bestimmung des Wertes von π durch ein „Hit-or-miss" - Verfahren. Genauer gesagt, bestimmen wir die Fläche eines Vierteleinheitskreises, also $\frac{\pi}{4}$. Die Kreislinie kann als $f(x) = \sqrt{1 - x^2}$oder als $x^2 + y^2 = 1$ beschrieben werden.

Man generiert nun N Paare von Zufallszahlen x_i, y_i. Ist $x_i^2 + y_i^2 \leq 1$, dann liegt der zugehörige Punkt innerhalb des Viertelkreises und man zählt einen Treffer. Das Verhältnis von Treffern (rote Punkte in Abbildung 17) zu allen „Würfen" entspricht dann dem Verhältnis von Viertelkreisflache zur Fläche des umgebenden Quadrat

$$\begin{aligned} \frac{N_{hit}}{N} &= \frac{A_{\bigcirc}}{A_{\square}} \\ &= \frac{\frac{\pi}{4}}{1} \end{aligned}$$

also ist demnach

$$\pi = \frac{4N_{hit}}{N}$$

.

Entsprechend lässt sich für jede beliebige Funktion das Integral $I = \int_a^b f(x)\,dx$ berechnen. Man generiert dazu Zufallspunkte $x_i \in [a, b]$, $y_i \in [f_{min}, f_{max}]$. Die Anzahl Treffer, also der (roten) Punkte unterhalb der Funktionskurve, I, im Verhältnis zur Gesamtzahl (Pnukte + Kreuze) entspricht dem Verhältnis der Fläche

[16] Sequenzen Zufallszahlen die nur mittels linearer kongruenter Generatoren mit dem gleichen Seed erzeugt werden, sind identisch.

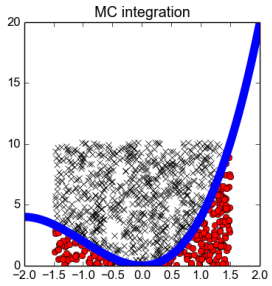

Abbildung 18: Monte-Carlo-Integration der Funktion $f(x) = x^3 + 3x^2$ mittels hit-or-miss.

unter der Kurve (abzüglich eines Rechtecks der Fläche $f_{min}(b-a)$) zur Fläche des betrachteten Rechtecks :

$$\frac{N_{hit}}{N} = \frac{I - f_{min}(b-a)}{(f_{max} - f_{min})(b-a)}$$

Alternativ können natürlich auch nicht die Punkte in einem Gebiet, sondern die Stützstellen x_i zufällig gewält werden und man berechnet

$$\begin{aligned} I &= \int_a^b f(x)\,dx \\ &\approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \end{aligned}$$

In vielen, d, Dimensionen kann man entsprechend mit dem Zufallsvektor $\mathbf{x}_i$ annähern.

$$\begin{aligned} I &= \int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \ldots \int_{a_d}^{b_d} dx_d f(\mathbf{x}) \\ &\approx \frac{(b_1-a_1)(b_2-a_2)\cdots(b_d-a_d)}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \end{aligned}$$

Vergleichen wir das mit der Rechteckregel

$$\begin{aligned} I &= \int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \ldots \int_{a_d}^{b_d} dx_d f(\mathbf{x}) \\ RI &\approx \frac{(b_1-a_1)}{N_1}\frac{(b_2-a_2)}{N_2}\cdots\frac{(b_d-a_d)}{N_3}\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\cdots\sum_{i_d=1}^{N_d} f\begin{pmatrix} \frac{h_1}{2}+a_1+i_1h_1 \\ \frac{h_2}{2}+a_2+i_2h_2 \\ \vdots \\ \frac{h_d}{2}+a_d+i_dh_d \end{pmatrix} \end{aligned}$$

Mit $\Pi_i N_i = N$ werden gleich viele Stützstellen verwendet Der wesentliche Unterschied der beiden Methoden ist die zufällige Wahl der Stützstellen bei der Monte-Carlo-Integration.

Welche ist dann besser? Hierzu betrachten wir den Fehler beider Methoden. Für die Rechteckmethode ist der Fehler in jedem Intervall und jeder[17] Dimension $\Delta_i \approx h^3 f''$ dat $f''(x_i) = \frac{f(x_i+h)+f(x_i-h)-2f(x_i)}{h^2}$. und der Gesamtfehler als Summe der Einzehlfehler $\Delta = \sum_i \Delta_i \approx Nh^3 \approx \frac{(b-a)^3}{N^2}$. In d Dimensionen sind

[17] sei die Länge der Einzelintervalle in allen Dimensionen gleich $h = h_i = \frac{(b_i-a_i)}{N_i}$

die N Stützstellen gleichmäßig über die Dimensionen verteilt, so dass $N = \left(N^{\frac{1}{d}}\right)^d$. Mit $N^{\frac{1}{d}}$ Stützstellen pro Raumrichtung ist der Fehler pro Richtung dann $\sigma_i = (b_i - a_i)^3 \left(\frac{1}{N^{\frac{1}{d}}}\right)^2 \propto \frac{1}{N^{\frac{2}{d}}}$.

Zur Betrachtung des Fehlers bei der Monte-Carlo-Integration müssen wir uns zunächst über Fehler bei Zufallsvariablen Gedanken machen.

Die Wahrscheinlichkeit $w_N(m)$ nach N „Würfen" m Treffer gelandet zu haben, wenn die Wahrscheinlichkeit, zu treffen p ist (und die Wahrscheinlichkeit nicht zu treffen $1-p = q$) folgt einer Binomialverteilung

$$\begin{aligned} w_N(m) &= \frac{N!}{m!\,(N-m)!} p^m (1-p)^{N-m} \\ &= \binom{N}{m} p^m q^{N-m} \end{aligned} \tag{64}$$

Die Berechnung des Mittelwerts und der Varianz ist im Anhang zu finden (A.1). Die relative Fluktuation ist der von uns gesuchte Fehler

$$\begin{aligned} \frac{\Delta m}{\langle m \rangle} &= \frac{\sqrt{Npq}}{Np} \\ &= \sqrt{\frac{q}{p}} \cdot \frac{1}{\sqrt{N}} \\ &\xrightarrow{N \to \infty} 0 \end{aligned} \tag{65}$$

Für sehr große N wird die relative Fluktuation also sehr klein und für unendlich große N strebt sie gegen Null. Das nennt man das Gesetz der großen Zahlen.

Für die Monte-Carlo-Integration bedeutet dies, dass mit wachsender Anzahl von „Würfen" N, der Integrationsfehler immer kleiner wird.

Bei sehr kleiner Wahrscheinlichkeit $p \leqslant 1$zu treffen aber großer Anzahl „Würfen", so dass pN immer noch endlich ist, ergibt sich aber nach

$$\begin{aligned} \frac{\Delta m}{\langle m \rangle} &= \sqrt{\frac{q}{p}} \cdot \frac{1}{\sqrt{N}} \\ &= \sqrt{\frac{1-p}{p}} \cdot \frac{1}{\sqrt{N}} \\ &= \sqrt{\frac{1 - \frac{\langle m \rangle}{N}}{\frac{\langle m \rangle}{N} N}} \\ &= \sqrt{\frac{1 - \frac{\langle m \rangle}{N}}{\langle m \rangle}} \\ &\approx \frac{1}{\sqrt{\langle m \rangle}} \end{aligned}$$

immer noch ein endlicher Fehler.

Für Funktionen, die an manchen Stellen kleine und an anderen große Funktionswerte haben, ist die Wahrscheinlichkeit des Treffers bei kleinen Werten viel geringer. Wir machen an diesen Stellen also einen größeren Fehler[18]. Selbst, wenn man eine genügend große Trefferwahrscheinlichkeit überall hat, so werden doch bei gleichmäßig zufälliger Verteilung der Stützpunkte (oder der „Würfe") sehr viel merh Versuche gebraucht, um an den „niedrigen" Stellen einmal zu treffen. Geschickter wäre es doch dort, wo das Treffen besonders schwierig ist, mehr Stützstellen evaluieren, bzw. häufiger „werfen" und an Stellen, bei denen wir ohnehin fast immer treffen entsprechend weniger.

[18] Man könnte jetzt argumentieren, dass der Fehler dann bei den großen Funktionswerten umso kleiner wird, aber es geht noch weiter...

8.3 Importance Sampling

Importance Sampling folgt genau der Idee, dort mehr Zufallspunkte zu setzen, wo sie wichtig sind. Wie aber lenkt man Zufallspunkte so dass diese bevorzugt in einem bestimmten Bereich landen? Man wählt diese Punkte eben nicht aus einer gleichmäßigen Verteilung, bei welcher jeder Zufallspunkt die gleiche Wahrscheinlichkeit hat, sondern eine Wahrscheinlichkeitsverteilung, die in gewünschter Weise wichtet. Sind die Zufallspunkte also z.B. gemäß einer Exponentialfunktion $p(x) = \exp(x)$ wahrscheinlichkeitsverteilt, so sind besonders große x besonders wahrscheinlich. Diese Wahrscheinlichkeitsverteilung dient uns als Wichtungsfunktion. In der Monte-Carlo.Integration funktioniert wird nun die zu integreirende Funktion mit der Wichtungsfunktion$g(x)$ multipliziert (und wieder dividiert), so dass

$$I = \int_a^b f(x)\,dx = \int_a^b g(x)\,\frac{f(x)}{g(x)}dx$$

Noch haben wir nichts gewonnen. Wir definieren nun ein , so dass $g(x) = \frac{dy}{dx}$, also $y(x) = \int dx' g(x')$. Und damit substituieren wir unser Integral

$$\begin{aligned} I &= \int_{y(a)}^{y(b)} \frac{f(x(y))}{g(x(y))}dy \\ &\approx \frac{(y(b) - y(a))}{N}\sum_{i=1}^{N}\frac{f(x(y_i))}{g(x(y_i))} \end{aligned}$$

worin y_i uniform verteilte Zufallszahlen im Intervall $[y(a), y(b)]$ sind, äquivalent zu $I \approx \frac{b-a}{N}\sum_{i=1}^{N}\frac{f(x_i)}{g(x_i)}$ mit x_i Zufallszahlen der Wahrscheinlichkeitsverteilung $g(x)$in $[a, b]$. Um diese Idee zu realisieren, benötigen wir auch noch die Umkehrfunktion von $y(x)$,also $x(y)$.

Betrachten wir ein Beispiel:

$$I = \int_0^1 \exp\left(-x^2\right)dx$$

Wir wählen als Wichtungsfunktion $g(x) = \exp(-x)$. Dann ist

$$\begin{aligned} y(x) &= \int_0^x g(x')\,dx' \\ &= \int_0^x \exp(-x')\,dx' \\ &= 1 - \exp(-x) \end{aligned}$$

Und mit

$$x(y) = -\ln(1-y)$$

wird das Integral zu

$$\begin{aligned} I &= \int_0^1 \frac{\exp\left(-x^2\right)}{\exp(-x)}\exp(-x)\,dx \\ &= \int_0^{1-\frac{1}{\exp}} \frac{\exp\left(-\left(\ln(1-y)\right)^2\right)}{\exp(-\ln(1-y))}dy \\ &\approx \frac{1-\frac{1}{\exp}}{N}\sum_{i=1}^{N}\frac{\exp\left(-\left(\ln(1-y_i)\right)^2\right)}{1-y_i} \end{aligned}$$

evaluiert durch N uniform verteilten Zufallszahlen y_i im Intervall $\left[0, \left(1-\frac{1}{\exp}\right)\right]$.

8.4 Metropolis-Monte-Carlo

Wie wir gesehen haben ist Monte-Carlo-Integration besonders günstig bei Problemen in hohen Dimensionen. Typische Probleme in der statistischen Physik ist die Berechnung von Erwartungswerten einer Observablen O nach

$$\langle O \rangle = \frac{\int dq^{3N} dp^{3N} \rho(\mathbf{q},\mathbf{p}) O(\mathbf{q},\mathbf{p})}{\int dq^{3N} dp^{3N} \rho(\mathbf{q},\mathbf{p})}$$

mit der Boltzmannverteilung $\rho(\mathbf{q},\mathbf{p}) = \exp(-H(\mathbf{q},\mathbf{p})/k_B T)$. Darin sind $H(\mathbf{q},\mathbf{p})$ eine Hamiltonfunktion, die von dem Positionen $\mathbf{q}$ und Impulsen $\mathbf{p}$ aller N Partikel in allen drei Raumrichtungen abhängt und einen Energiewert liefert[19]. T ist die Temperatur und k_B die Boltzmannkonstante. Kürzen wir nun das Integral im Nenner mit Z und $\mathbf{q},\mathbf{p}$ mit ζ ab, so ist

$$\langle O \rangle = \frac{1}{Z} \int d\zeta \rho(\zeta) O(\zeta)$$

Als Wichtungsfunktion bietet sich $\rho(\zeta)$ an, so dass

$$\begin{aligned} \langle O \rangle &= \frac{1}{Z} \int d\zeta \rho(\zeta) \frac{f(\zeta)}{\rho(\zeta)} \\ &\approx \frac{1}{Z} \frac{1}{N} \sum_{i=1}^{N} \frac{f(\zeta_i)}{\rho(\zeta_i)} \\ &\approx \frac{1}{Z} \left\langle \frac{f(\zeta)}{\rho(\zeta)} \right\rangle \end{aligned}$$

Um nun das Integral zu berechnen. Die Stützstellen würden also gemäß einer Verteilung $\rho(\zeta)$ gezogen. Dummerweise kennen wir zwar den formalen Ausdruck für die Boltzmann-verteilung, müssten aber zu deren Berechnung über die Hamiltonfunktion den Energiewert an allen $(\mathbf{q},\mathbf{p})$ bestimmen. Für einfache Systeme und einfache Funktionen (z.B. ein ideales Gas) ist das analytisch möglich. Für die meisten anderen Systeme, macht schon das „alle“ ($-\infty$ bis $+\infty$) dieses Unterfangen unmöglich (Die Berechnung der Zustandssumme Z ist genauso hoffnungslos).

Der Ausweg führt darüber, sich die Verteilung „unterwegs“ zu erzeugen. Dieser basiert darauf, dass Verhältnisse von Wahrscheinlichkeiten zweier Punkte im $(\mathbf{q},\mathbf{p})$-Raum, dem *Phasenraum*, vergleichsweise einfach zu bestimmen sind, auch weil die *Zustandssumme* Z sich heraus kürzt.

Wir betrachten nun die Stützstellen, oder Punkte im Phasenraum nacheinander. Dabei muss es egal sein, ob wir von Punkt ζ_i nach ζ_k wandern, oder anders herum. Mit anderen Worten, die *Detailed-balance-Bedingung* muss erfüllt sein

$$\rho(\zeta_i) T(\zeta_i \to \zeta_k) = \rho(\zeta_k) T(\zeta_k \to \zeta_i) \tag{66}$$

Die Wahrscheinlichkeit im Punkt ζ_i zu sein und von Punkt ζ_i nach Punkt ζ_k zu gehen, ist gleich der Wahrscheinlichkeit im Punkt ζ_k zu sein und von Punkt ζ_k nach Punkt ζ_i zu gehen. Und umgeformt

$$\frac{\rho(\zeta_i)}{\rho(\zeta_k)} = \frac{T(\zeta_k \to \zeta_i)}{T(\zeta_i \to \zeta_k)}$$

Bei nur zwei Zuständen sieht man leicht, dass so die gewünschte Verteilung, also die gewünschten relativen Wahrscheinlihckeiten erzeugt werden können. Wenn ζ_i z.B. neun mal so wahrscheinlich sein soll wie ζ_k, dann muss die Wahrscheinlichkeit ζ_j zu verlassen, um nach ζ_i zu gehen , also $T(\zeta_k \to \zeta_i)$ neun mal höher sein als $T(\zeta_i \to \zeta_k)$. Oder umgekehrt, wenn aus ζ_i immer nur jeder neunte geht, dann sind (nach einer genügend langen Weile) neun mal soviele geblieben wie gegangen.

[19] und damit das physikalische Modell beinhaltet. Die Hamiltonfunktion kann eine klassische Mechanische FUnktion sein, oder ein quantenmechanischer Hamiltonoperator, der auf einen Energieeigenwert führt.

Im Monte-Carlo-Verfahren teilen wir zunächst die Wahrscheinlichkeit des Übergangs auf in eine Wahrscheinlichkeit, den Schritt[20] vorzuschlagen, T_{prop}, und eine Wahrscheinlichkeit, ihn zu akzeptieren, T_{acc}.

$$\frac{\rho(\zeta_i)}{\rho(\zeta_k)} = \frac{T_{prop}(\zeta_k \to \zeta_i) \cdot T_{acc}(\zeta_k \to \zeta_i)}{T_{prop}(\zeta_i \to \zeta_k) \cdot T_{acc}(\zeta_i \to \zeta_k)}$$

Ist die Wahrscheinlichkeit für den Vorschlag symmetrisch, also „vorwärts"- und „rückwärts"gehen gleich wahrscheinlich $T_{prop}(\zeta_i \to \zeta_k) = T_{prop}(\zeta_k \to \zeta_i)$ verbleibt noch

$$\frac{\rho(\zeta_i)}{\rho(\zeta_k)} = \frac{T_{acc}(\zeta_k \to \zeta_i)}{T_{acc}(\zeta_i \to \zeta_k)}$$

Die Akzetanzwahrscheinlichkeit folgt dann nach dem sogenannten Metropolis-Kriterium

$$T_{acc}(\zeta_k \to \zeta_i) = \min\left\{1, \frac{\rho(\zeta_i)}{\rho(\zeta_k)}\right\}$$

Wenn der neue Zustand/der neue Punkt also eine höhere Wahrscheinlichkeit hat als der alte, wird mit Sicherheit akzeptiert. Wenn nicht, wird der Schritt mit der Wahrscheinlichkeit $\frac{\rho(\zeta_i)}{\rho(\zeta_k)} < 1$ akzeptiert. Realisert werden kann dies durch ziehen einer gleichverteilten Zufallszahl $r \in [0,1]$. Ist $r < \frac{\rho(\zeta_i)}{\rho(\zeta_k)}$, wird der Schritt akzeptiert, sonst wird der Schritt verworfen und am aktuellen Punkt verblieben. Aus der Häufigkeit, wie oft welcher Punkt besucht wurde, lässt sich dann direkt die Wahrscheinlichkeitsverteilung abschätzen. Und für die Obervable ist dann

$$\langle O \rangle = \frac{1}{N} \sum_{i=1}^{N} f(\zeta_i)$$

Algorithmus

1. Wähle Startpunkt $\zeta_n = \zeta_0$
2. Schlage neuen Punkt vor durch Verschiebung um $\Delta\zeta$, wobei pus und minus gleich wahrscheinlich sind: $\zeta_{trial} = \zeta_0 \pm \Delta\zeta$
3. Berechne $\frac{\rho(\zeta_i)}{\rho(\zeta_k)}$
   (a) falls $\frac{\rho(\zeta_i)}{\rho(\zeta_k)} > 1$, akzeptiere und $\zeta_{n+1} = \zeta_{trial}$
   (b) falls $\frac{\rho(\zeta_i)}{\rho(\zeta_k)} < 1$, ziehe Zufallszahl $r \in [0,1]$
      i. falls $r < \frac{\rho(\zeta_i)}{\rho(\zeta_k)}$, akzeptiere und $\zeta_{n+1} = \zeta_{trial}$
      ii. falls $r < \frac{\rho(\zeta_i)}{\rho(\zeta_k)}$, verwerfe und $\zeta_{n+1} = \zeta_n$
4. Gehe zu 2.

[20] bei vielen Zuständen können wir ja $\zeta_k, \zeta_l, \zeta_m, \zeta_n, \ldots$ als neue Zustände vorschlagen

9 Graphen und Netzwerke

10 Simulationen

A Anhang

A.1 Zufallsvariablen, Binomialverteilung und Schwankungen

Dazu überlegen wir uns, dass hit-or-miss einer Binomial-Verteilung gehorcht, wobei $w_N(m)$ die Wahrscheinlichkeit ist, nach N „Würfen" m Treffer gelandet zu haben, wenn die Wahrscheinlichkeit, zu treffen p ist (und die Wahrscheinlichkeit nicht zu treffen $1-p=q$). $N!$ ist die Anzahl allermöglichen Reihenfolgen von Würfen, egal, wo diese landen. Innerhalb der Treffer sit deren Reihenfolge auch egal, dewswegen dividieren wir durch $m!$, und entsprechend für die nicht-Treffer durch $(N-m)!$. Damit kennzeichnet der Bruch die Anzahl der verschiedenen Reihenfolgen (bei zwei Treffern (1) aus drei Würfen z.B. 110; 101; oder 011 sind das $\frac{3!}{2!1!}$=3). Diese Anzahl multipliert mit der Wahrscheinlichkeit zu treffen für jeden Treffer, p^m und ebenso für die Nicht-Treffer q^{N-m}.

$$\begin{aligned} w_N(m) &= \frac{N!}{m!\,(N-m)!}p^m\,(1-p)^{N-m} \\ &= \binom{N}{m}p^m q^{N-m} \end{aligned}$$

Der Erwartungswert (oder Mittelwert) einer Zufallszahl x ist gegeben durch

$$\langle x\rangle = \sum_i x_i w(x_i)$$

wobei $w(x)$die zugehörige Wahrscheinlichkeitsverteilung ist.

Die Varianz einer Zufallszahl ist die mittlere Abweichung vom Mittelwert und daher

$$(\Delta x)^2 = \left\langle (x-\langle x\rangle)^2\right\rangle = \langle x^2\rangle - \langle x\rangle^2$$

mit

$$\langle x^2\rangle = \sum_i x_i^2 w(x_i)$$

Die Wurzel aus der Varianz ist dann die Standardabweichung $\sigma(x)=\Delta x$. Der Fehler für den wir uns jetzt interessieren ist die relative Fluktuation

$$\frac{\Delta x}{\langle x\rangle}$$

Für die binomische Verteilung $w_N(m)$ betrachten wir also zuerst die mittlere Anzahl Treffer

$$\langle m\rangle = \sum_{m=0}^{N} m\cdot w_N(m) \tag{67}$$

Um das zu berechnen nutzen wir einen „Trick"

$$\begin{aligned} p\frac{\partial}{\partial p}\sum_{m=0}^{N} w_N(m) &= p\frac{\partial}{\partial p}\sum_{m=0}^{N}\frac{N!}{m!\,(N-m)!}p^m\,(q)^{N-m} \\ &= p\cdot\sum_{m=0}^{N} m\cdot\frac{N!}{m!\,(N-m)!}p^{m-1}\,(q)^{N-m} \\ &= \sum_{m=0}^{N} m\cdot\frac{N!}{m!\,(N-m)!}p^m\,(q)^{N-m} \\ &= \sum_{m=0}^{N} m\cdot w_N(m) \\ &= \langle m\rangle \end{aligned} \tag{68}$$

Unter Ausnutzung des binomischen Lehrsatzes

$$\sum_{m=0}^{N} w_N(m) = 1 \tag{69}$$

$$\begin{aligned} 1 &= \sum_{m=0}^{N} \frac{N!}{m!\,(N-m)!} p^m (1-p)^{N-m} \\ 1 &= \sum_{m=0}^{N} \binom{N}{m} p^m q^{N-m} \\ 1 &= (p+q)^N \end{aligned} \tag{70}$$

können wir dann schreiben

$$\begin{aligned} \langle m \rangle &= p\frac{\partial}{\partial p} \sum_{m=0}^{N} w_N(m) \\ &= p\frac{\partial}{\partial p} (p+q)^N \\ &= pN\underbrace{(p+q)^{N-1}}_{=1} \\ \langle m \rangle &= Np \end{aligned} \tag{71}$$

Zur Bestimmung der Varianz nutzen wir den gleichen „Trick“ von eben zweimal

$$\begin{aligned} p\frac{\partial}{\partial p} p\frac{\partial}{\partial p} \sum_{m=0}^{N} w_N(m) &= p\frac{\partial}{\partial p} \sum_{m=0}^{N} m \cdot w_N(m) \\ &= \sum_{m=0}^{N} m^2 \cdot w_N(m) \\ &= \left\langle m^2 \right\rangle \end{aligned} \tag{72}$$

Somit ist

$$\begin{aligned} \left\langle m^2 \right\rangle &= p\frac{\partial}{\partial p} p\frac{\partial}{\partial p} \sum_{m=0}^{N} w_N(m) \\ &= p\frac{\partial}{\partial p} p\frac{\partial}{\partial p} (p+q)^N \\ &= p\frac{\partial}{\partial p} pN(p+q)^{N-1} \\ &= p\left[N(p+q)^{N-1} + pN(N-1)(p+q)^{N-2}\right] \\ &= \left[pN\underbrace{(p+q)^{N-1}}_{=1} + p^2 N(N-1)\underbrace{(p+q)^{N-2}}_{=1}\right] \end{aligned} \tag{73}$$

$$\left\langle m^2 \right\rangle = pN + p^2 N(N-1) \tag{74}$$

Die Fluktuation ist dann

$$
\begin{aligned}
\Delta m &= \sqrt{\langle m^2 \rangle - \langle m \rangle^2} \\
&= \sqrt{N\left(N-1\right)p^2 + Np - \left(Np\right)^2} \\
&= \sqrt{N^2p^2 - Np^2 + Np - N^2p^2} \qquad (75) \\
&= \sqrt{N\left(p - p^2\right)} \\
&= \sqrt{Np\left(1-p\right)} \\
&= \sqrt{Npq} \qquad (76)
\end{aligned}
$$

und die relative Fluktuation

$$
\begin{aligned}
\frac{\Delta m}{\langle m \rangle} &= \frac{\sqrt{Npq}}{Np} \\
&= \sqrt{\frac{q}{p}} \cdot \frac{1}{\sqrt{N}} \\
&\xrightarrow{N \to \infty} 0 \qquad (77)
\end{aligned}
$$

Für sehr große N wird die relative Fluktuation also sehr klein und für unendlich große N strebt sie gegen Null. Das nennt man das Gesetz der großen Zahlen.