Computergestüzte Methoden der exakten Naturwissenschaften

24. November 2013

Inhaltsverzeichnis

1 Fehler

Ein Ziel der Naturwissenschaften ist die Beschreibung der Natur mit Hilfe von mathematischen Gleichungen und deren Lösungen, daraus ergibt sich allerdings ein Problem.

Problem: Die Gleichungen der naturwissenschaftlichen Beschreibungen können nicht immer mit Bleistift und Papier zu gelöst werden.

Lösung 1: Vereinfachung der Gleichungen $\hat{=}$ Näherung/Approximation

Lösung 2: Numerische Lösung der Gleichungen.

Diese Vorlesung möchte sich mit der zweiten Lösungsmethode befassen, hierbei ist es allerdings wichtig die Genauigkeit der numerisch ermittelten Ergebnisse (die Fehler) mit zu berücksichtigen.

Allgemein gibt es für es verschiedene Quellen für Fehler:

Eingabefehler: Diese entstehen durch Ungenauigkeiten innerhalb der Eingabedaten.

Näherungsfehler: Solche entstehen aus der Verwendung vereinfachter mathematischer Ausdrücke anstelle der exakten.

Modellfehler: Diese entstehen aus der Nutzung vereinfachter physikalischer Modelle.

Rundungsfehler: Solche entstehen aus der numerischen Darstellung von Zahlen und der damit verbundenen endlichen Genauigkeit.

1.1 Beispiele für Näherungsfehler

Viele mathematische Gleichungen der Physik sind in ihren exakten Formulierungen nicht oder nur sehr aufwendig lösbar. Ein Ausweg stellen Approximationen dar aus welchen allerdings zusätzliche Näherungsfehler resultieren. Beispiele hierfür sind über unendliche Reihen definierte Funktionen aber auch Differentialgleichungen im Kontinuum.

Exponentialfunktion: Die Exponentialfunktion ist definiert durch:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Eine solche Funktion kann durch eine endliche Reihe genähert werden:

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!}$$

Differentialgleichung im Kontinuum: Eine Differentialgleichung im Kontinuum kann durch die Lösung der zugehörigen diskretisierten Gleichung genähert werden. Sei die Differentialgleichung gegeben durch:

$$\frac{d}{dx} f(x) = a\ f(x),$$

so ergibt sich die dikretisierte Gleichung aus der Diskretisierung auf bestimmte Gitterpunkte x_i mit dem Abstand $\Delta x = x_{i+1} - x_i$:

$$\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = a\ \frac{f(x_{i+1}) + f(x_i)}{2}.$$

Zur Verbesserung der Diskretisierung kann dann Δx immer weiter gegen 0 gesetzt werden. Ein *Nachteil* ist hierbei die Erhöhung der Rechenoperationen und der damit verbundenen Rechenzeit. Außerdem vergrößern sich hiermit die Rundungsfehler.

Das Ziel der Numerik besteht nun im optimalen Kompromiss zwischen Fehler und Rechenzeit.

1.2 Beispiel für Modellfehler

Als Beispiel für einen aus einem Modell resultierenden Fehler wird die Planetenbewegung betrachtet. Nach dem ersten newtonschen Gesetz gilt:

$$\mathbf{F} = m\mathbf{a} = m\ddot{\boldsymbol{r}} = -\frac{M\ \mathbf{r}}{|\mathbf{r}|^3}$$

Hierbei gehen allerdings eine reihe von Näherungen ein:

- Die Sonnenmasse M wird relativ zur Planetenmasse als sehr groß angenommen
- Eine geschwindigkeitsabhängige Reibungskraft $\boldsymbol{F_R} = \gamma\ \dot{\boldsymbol{r}}$ wird vernachlässigt, diese ist allerdings für kleinere Objekte wichtig.
- Auch relativistische Effekte werden vernachlässigt, solche erklären allerdings Phänomene wie die Periheldrehung des Merkurs.
- Eigentlich handelt es sich um ein Mehrkörperproblem der Form:

$$m_1\ddot{\boldsymbol{r}}_{\boldsymbol{i}} = \sum \boldsymbol{F}_i = -\sum_{j\neq i} G m_i m_j \frac{\boldsymbol{r_i} - \boldsymbol{r_j}}{|\boldsymbol{r_i} - \boldsymbol{r_j}|^3}.$$

Eine Gleichung für mehr als 3 Objekte kann so also leicht geschrieben werden. Allerdings ist das Problem bereits ab einer Beteiligung von 3 Objekten nur noch unter Annahme bestimmter Bedingungen und ab 4 Objekten überhaupt nicht mehr exakt lösbar.

1.3 Rundungsfehler

Beim durchführen von Rechenoperationen mit reellen Zahlen am Computer muss gerundet werden, die daraus entstehenden Fehler heißen Rundungsfehler.

1.3.1 Gleitpunktarithmetik

Reelle Zahlen werden am Computer in das Gleitpunktformat umgewandelt. Der Vorteil gegenüber dem Festpunktformat liegt im geringeren Speicherbedarf. Hierzu werden die eingegebenen Zahlen in der Form:

$$x = \pm \sum_{i=1}^{n} z_i \; B^{E-i} := \pm \underbrace{(0, z_1 z_2 ... z_n)_B}_{\text{Mantisse}} \; B^E$$

dargestellt. Dabei gilt des weiteren für den Exponenten $E \in \mathbb{Z}$: $m \leq E \leq M$. Außerdem gilt $z_i \in \{0, 1...B-1\}$.
Beispiel:

$$1234,567 = (0,1234567)_{10} \cdot 10^4$$

Die Werte für n, B, m und M sind hierbei maschinenabhängig, werden also durch den Rechner und den Compiler bestimmt.
Übliche Basen sind:

$B = 2$: Dualzahlen

$B = 8$: Oktalzahlen

$B = 10$: Dezimalzahlen

$B = 16$: Hexadezimalzahlen

Standartformate für B=2:
Single: Dieses Format besteht aus 32 Bits bzw. 4 Bytes. Diese ergeben sich aus:

Vorzeichen: 1 Bit

Exponent: 8 Bits

Mantisse: 23 Bits

Genauigkeit: 6 Ziffern unterscheidbar

Double: Dieses Format besteht hingegen aus 64 Bits:

Vorzeichen: 1 Bit

Exponent: 11 Bits

Mantisse: 52 Bits

Genauigkeit: 15 Ziffern unterscheidbar

Beispiel Binäre Darstellung von $(5,0625)_{10}$:

$$(5,0625)_{10} = (0,50625)_{10} \cdot 10^1 = 2^2 + 2^0 + 2^{-3} + 2^{-4}$$
$$\Rightarrow (5,0625)_{10} = (101,0001)_2 = (0,1010001)_2 \cdot 2^{(11)_2}$$

Manche Zahlen wie $(0,3)_{10}$ lassen sich allerdings nur schwer als duale Zahlen darstellen.

Die **größte darstellbare Zahl** ergibt sich zu:

$$x_{max} = (0, \underbrace{[B-1][B-1]...[B-1]}_{\text{n Ziffern}})_B \; B^M = B^M[B-1]\frac{B^{-n}(B^n-1)}{B-1} = B^M(1-B^{-n}).$$

Dagegen ergibt sich die **kleinstmögliche Zahl** zu:

$$x_{min} = B^{m-1}$$

Folglich ist die Menge der darstellbaren Maschinenzahlen endlich. Ergibt sich während der Rechnung eine Zahl $x > x_{max}$ folgt ein overflow und die Zahl wird auf ∞ gesetzt. In gleicher weise ergibt sich für $x < x_{min}$ der underflow und die Zahl wird auf 0 gesetzt.
Beispiele:

$$x_{max} + x_{max} = \infty \tag{1.1}$$
$$x_{min} \; B^{-1} = 0 \tag{1.2}$$

Jede reelle Zahl die keine Maschinenzahl ist muss in eine solche umgewandelt werden. Idealerweise wählt man Maschinenzahl dabei möglichst nahe der reellen Zahl $\hat{=}$ Rundung.

1.3.2 Rundung

Beim Runden wird für eine Zahl x eine Näherung $rd(x)$ unter den Maschinenzahlen geliefert, so dass der absolute Fehler $|x - rd(x)|$ minimal ist. Der dabei unvermeidbare Fehler heißt Rundungsfehler. Eine n-stellige Dezimalzahl im Gleitpunktformat $\tilde{x} = \pm(0, z_1...z_n)_{10} = rd(x)$ hat einen maximalen Fehler von:

$$|x - rd(x)| \leq 0, \underbrace{00...00}_{\text{n Ziffern}} 5 \cdot 10^E = 0,5 \cdot 10^{E-n}.$$

Für eine allgemeine Basis B ergibt sich:

$$|x - rd(x)| \leq \frac{B}{2} \; \frac{1}{B} \; B^{E-n} = \frac{1}{2} B^{E-n}.$$

Rundungsfehler werden durch die gesamte Rechnung getragen.

Bei einer **n-stellige Gleitpunktarithmetik** wird jede einzelne Rechenoperation auf $n+1$ Stellen genau berechnet und dann auf n Stellen gerundet. Es wird also nicht nur das Endergebnis gerundet.

Beispiel: $2590 + 4 + 4$ in *3-stelliger dezimaler Gleitpunktarithmetik*
Von links nach rechts:

$$2590 + 4 = 2594 \underbrace{\Rightarrow}_{\text{Rundung}} 2590$$
$$2590 + 4 = 2594 \underbrace{\Rightarrow}_{\text{Rundung}} 2590$$

Von rechts nach links:

$$4 + 4 = 8 \underbrace{\Rightarrow}_{\text{Rundung}} 8$$
$$2590 + 8 -> 2598 \underbrace{\Rightarrow}_{\text{Rundung}} 2600$$

Das exakte Ergebnis wäre 2598. Die Reihenfolge der Ausführungen der Rechenoperationen verändert also das Ergebnis. Daraus folgt die **Regel**, dass beim **Addieren** die Summanden in der Reihenfolge ihrer aufsteigender Beträge addiert werden. So erhält man bei gleicher Rechenzeit bessere Ergebnisse.

Einschub:Maß für die Rechenzeit eines Computers:
flops$\hat{=}$floating point operations per second, dabei sind Multiplikation und Division typische Operationen. Eine Rangliste schnellsten Computer wird auf www.top500.org geführt.

Der **relative Fehler** ist meist relevanter als der absolute Fehler. Die Näherung $\tilde{x}$ zu dem exaktem Wert x ergibt einen relativer Fehler: $\epsilon = |\frac{\tilde{x}-x}{x}| \approx |\frac{\tilde{x}-x}{\tilde{x}}|$.
Daraus ergibt sich der maximaler Rundungsfehler zu:

$$\epsilon_{max} = \frac{\frac{1}{2}\, B^{E-n}}{B^{E-1}} = \frac{1}{2}\, B^{1-n}$$

Für duale Rechnungen im Computer gilt also $B = 2\epsilon_{max} \cdot 2^{-n}$.
ϵ_{max} wird auch **Maschinengenauigkeit** genannt und gibt die kleinste positive Zahl an für die gilt $1 \cdot \epsilon_{max} \neq 1$.
ϵ_{max} kann aus Rechenoperationen rekonstruiert werden.

Rundungsfehler bei Rechenoperationen
Beispiele: (mit 4er Mantissen und 1er Exponentenziffer, Dez.)

Addition und Subtraktion von Zahlen mit stark unterschiedlichen Exponenten:

Rundungsfehler kann verloren gehen: $1234 + 0,5 = 0{,}1234 \cdot 10^4 + 0{,}5 = 1235$.
Fehler: $0,5$, rel. Fehler: $0,00040$, $\epsilon_{max} = 0{,}5 \cdot 10^{-3} + 0{,}5 = 1235$, der Rundungsfehler ist also kleiner als der Maximale.

Multiplikation und Division:

(underflow/overflow möglich): $0,2 \cdot 10^2 \cdot 0{,}3 \cdot 10^{-6} = 0{,}6 \cdot 10^{-12} = 0$
$0,2 \cdot 10^{-2} \cdot 0{,}3 \cdot 10^{-6} = 0{,}6 \cdot 10^{12} = \infty$ (Hier wäre der rel. Fehler ∞)

Fehler beim Assoziativgesetz:

a) $0{,}1111 \cdot 10^{-3} + (-0,1234 + 0,1234) = 9{,}111 \cdot 10^{-3} + 0,0009 = 0{,}101\,11 \cdot 10^{-2} = 0{,}1011 \cdot 10^{-2}$)

b) $(-0,1234 + 0,1234 = 9{,}111 \cdot 10^{-3} + 0,0009) + 0,1243 = -0,1233 + 0,1243 = 0,0010 = 0,100 \cdot 10^{-2}$

Der exakte Wert wäre aber: $0,10111 \cdot 10^{-2}$ Daraus folgt:

a) Fehler: $0,00001 \cdot 10^{-2}$, rel. Fehler: $0,01\%$

b) Fehler: $0,00111 \cdot 10^{-2}$, rel. Fehler: 1%

Im Fall b) ist also $\epsilon > \epsilon_{max}$.

1.3.3 Fehlerfortpflanzung bei Rechenoperationen

Fehler werden beim Rechnen weitergetragen, selten werden dabei die Fehler kleiner (meistens werden sie größer!). Durch das Umstellen von Formeln können Fehler minimiert werden, trotzdem müssen Fehler abgeschätzt werden.

Additionsfehler:
Gegeben: Fehlerbehaftete Größen $\tilde{x}$ und $\tilde{y}$ zu den Werten x und y.
Fehler der Summe: $\tilde{x} + \tilde{y} - (x + y) = (\tilde{x} - x) + (\tilde{y} - y)$
Im ungünstigsten Fall addieren sich die Fehler: bei Additionen und Subtraktionen addieren sich die Absolutbeträge der Fehler der einzelnen Terme.

Multiplikation: Fehler: $\tilde{x}\tilde{y} - xy = \tilde{x}(\tilde{y} - y) + \tilde{y}(\tilde{x} - x) - (\tilde{x} - x)(\tilde{y} - y)$, also hat das Produkt von $\tilde{y}$ mit einer Maschinenzahl ohne Fehler den $\tilde{x}$-fachen Fehler; Produkt der Fehler typischerweise vernachlässigbar.
Der absolute Fehler eines Produkts ist gegeben durch das Produkt des Faktors mit dem Fehler des anderen Faktors. (=2 Terme, oft ist einer der Fehler dominant.)

Relativer Fehler Multiplikation:

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

beim Multiplizieren addieren sich die relativen Fehler, Division analog.

1.3.4 Fehlerfortpflanzung bei Funktionen

Die Funktion wird an der Stelle $\tilde{x}$ anstatt x ausgewertet, daraus folgt ein fehlerbehafteter Funktionswert. Je nach Funktion resultiert ein kleiner oder großer Fehler. Bei weiteren Funktionsauswertungen wird der Fehler typischerweise größer.
Aus dem Mittelwertsatz folgt:

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

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

an einer unbekannten Stelle x_0 im Intervall $(x, \tilde{x})$.
Wähle $g(x) = f'(x)$:

$$|f(\tilde{x}) - f(x))| = |\tilde{x} - x||f'(x_0)|.$$

Der absolute Fehler vergrößert sich also für $|f'(x_0)| > 1$ und wird für $|f'(x_0)| < 1$ kleiner. Die Ableitung kann also als Verstärkungsfaktor des Fehlers interpretiert werden.

Abschätzung des absoluten Fehlers:

$$|f(x) - f(\tilde{x})| \leq M \ |x - \tilde{x}|,$$

mit $M = \max\limits_{x \leq x_0 \leq \tilde{x}}(|f'(x_0)|)$. Schätzung des Fehlers: $|f(x) - f(\tilde{x})| \approx f'(\tilde{x})|x - \tilde{x}|$.

Beispiel 1:
Fortpflanzung des absoluten Fehlers von $f(x) = sin(x)$:

$$f'(x) = cos(x) \rightarrow M = 1,$$

das heißt für die meisten Argumente verringert sich der absolute Fehler.
Beispiel 2:

$$f(x) = \sqrt{x}; f'(x) = \frac{0,5}{\sqrt{x}},$$

divergiert also für $x \rightarrow 0$

Der relative Fehler bei Funktionsauswertung:

$$\frac{|f(x) - f(\tilde{x}|)}{|f(x)|} \leq \frac{M|x|}{|f(x)|}\frac{|x - \tilde{x}|}{|x|}$$

$$\approx \underbrace{\frac{|f'(\tilde{x})||\tilde{x}|}{|f(\tilde{x})|}}_{\textbf{Konditionszahl}} \cdot \frac{|x - \tilde{x}|}{|\tilde{x}|}.$$

Die Konditionszahl ist also ein Verstärkungsfaktor für relative Fehler; qualitativ: Probleme wenn Konditionszahl>>1 ;schlecht konditioniertes Problem.

2 Nullstellenproblem

Gegeben: Funktion $\mathbb{R} \to \mathbb{R}$
Gesucht: Nullstellen also x_0 aus R mit $f(x_0) = 0$ Grundsätzlich:

- Gibt es überhaupt Nullstellen? Wenn ja, in welchem Bereich?
- Gibt es mehrere Lösungen?

Zwischenwertsatz
$f : [a, b] \to R$, stetig für $C \in R$ mit $f(a) \leq c \leq f(b)$ gibt es ein $x_0 \in [a, b]$ so dass $f(x_0) = c$.
Für $c = 0$ ist dieser Satz bei der Nullstellensuche hilfreich.
Suche Funktionswerte mit unterschiedlichem Vorzeichen: $f(a) \cdot f(b) < 0$. Dann gibt es zwischen a und b mindestens eine Nullstelle.

2.1 Bisektionsverfahren

$$f(a) \cdot f(b) < 0$$
$$\hat{=}\text{Nullstellen in } (a, b)$$

Berechne Vorzeichen von $f(\frac{a+b}{2})$
$\to f(x) = 0$ in $(a, \frac{a+b}{2})$ oder $(\frac{a+b}{2}, b) \to$ Berechne Vorzeichen von $f(\frac{a+b}{4})$ oder $\frac{3}{4}(a+b)$...

Beispiel:

$f(x) = x^3 - x + 0,3 = 0$
Wie viele Nullstellen?

x	-2	-1	0,5	1
f(x)	-5,7	0,3	-0,075	0,3

Wo sind die Nullstellen?
Bestimme die Nullstelle zwischen $x = 0$ und $x = 0,5$ genau auf eine Stelle nach dem Komma.

$$f(0,25) > 0 \to \text{Nullstelle in } [0,25; 0,5]$$
$$f(0,375) < 0 \to \text{Nullstelle in } [0,25; 0,375]$$
$$f(0,3125) > 0 \to \text{Nullstelle in } [0,3125; 0,375]$$

Also ist die Nullstelle bei $0,3$....

2.2 Fixpunkt-Iteration

Eine Gleichung der Form $x^{n+1} = f(x^{(n)})$ wird als Fixpunktgleichung bezeichnet. Die Lösung(en) $\bar{x}$ mit $\bar{x} = f(\bar{x})$ heißen Fixpunkte. (da unter der Abbildung der Punkt $\bar{x}$ frei (=unveränderlich) bleibt.) Jedes Nullstellenproblem kann als eine solche Fixpunktgleichung definiert werden.

Beispiel:
Finde Nullstelle von $g(x) = x^3 - x + 0,3$. Umformen der Fixpunktgleichung: $x^3 - x + 0,3 = 0$:

$$x^3 - x + 0,3 = 0 \rightarrow x = x^3 + 0,3$$
$$\rightarrow x^{(n+1)} = (x^{(n)})^3 + 0,3$$

Wir wählen zunächst Startwerte nahe der vermuteten Nullstellen ($\hat{=}$ Fixpunkten) und berechnen Werte für folgende n.

n	$x^{(n)}$	$x^{(n)}$	$x^{(n)}$
0	-1	0	1
1	-0,7	0,3	1,3
2	-0,043	0,327	2,497
3	0,2999	0,3349	15,87
4	0,327	0,337	
8	0,33877	0,33891	

Die ersten zwei Startwerte konvergieren zum selben Fixpunkt. In der Tat ist nur der Fixpunkt $\bar{x} = 0,3389...$ anziehend, die anderen werden als abstoßend bezeichnet. Der Kreuzungspunkt zwischen der linken und der rechten Seite der Gleichung liefert den Fixpunkt.
Vergleich der Steigungen von $y = x$ und $y = f(x)$ am Fixpunkt:
Steigung von $f(x)$ ist kleiner als x, also $f'(x) < 1 \rightarrow$ Grund für Konvergenz. Die Konvergenz ist also umso schneller je kleiner $f'(x)$ am Fixpunkt.

Fixpunktsatz:
Sei $f : [a, b] \rightarrow \mathbb{R}$ mit stetiger Ableitung $f'(x)$ und $\bar{x}$ ein Fixpunkt von f, dann gilt für die Iteration:

$$x^{(n+1)} = (x^{(n)})^3 + 0,3 :$$

ist $|f'(\bar{x})| < 1$ so konvergiert $x^{(n)}$ gegen $\bar{x}$ falls $x^{(0)}$ nage genug an $\bar{x}$ liegt: $\bar{x}$ ist ein anziehender Fixpunkt.
Ist $|f'(\bar{x})| > 1$ so konvergiert $x^{(n)}$ für keinen Startwert $x^{(n)} nicht \bar{x}$. $\bar{x}$ ist dann ein abstoßender Fixpunkt.

zurück zum Beispiel:

Plot der Funktionen und Kontrolle der Startpunkte.
Die Fixpunkte sind $\bar{x_1} = -1,125$, $\bar{x_2} = 0,3389$ und $\bar{x_3} = 0,7864$.
Plot der Abbildung $\rightarrow$ stabiler Punkt ablesbar.

2.3 Newton Verfahren

Gegeben: differenzierbare Funktion $f(x)$
Gesucht: Nullstelle $\bar{x}$ mit $f(\bar{x}) = 0$
Ausgangspunkt x_0 (in der Nähe von $\bar{x}$)
Lösung:
Linearisierung von $f(x)$ um x_0:
$f(x) \approx f(x_0) + (x - x_0)\ f'(x_0) = 0$
$f'(x_0) \neq 0 \rightarrow x_0 - \frac{f(x_0)}{f'(x_0)}$
Das heißt die Funktion $f(x)$ wird durch die Tangente am Pukt x_0 genähert. Verbesserungen sind im Prinzip möglich.
Wird dieses Prinzip iterativ angewandt redet man vom Newton-Verfahren.

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

An unserem Beispiel ergibt sich:

n	x_n		
0	-1	0	1
1	-1,15	0,3	0,85
2	-1,12615	0,33699	0,7951
3	-1,1254	0,3389	0,78668
4	-1,12542	0,33894	0,78649

Nach nur 4 Iterationen hat man eine Genauigkeit von 10^{-4} erreicht. Das Newton-Verfahren ist sehr schnell und beliebt; ein Nachteil liegt allerdings darin, dass in jedem Schritt eine Ableitung berechnet werden muss.

Lösung 1: *Das Vereinfachte Newton-Verfahren*
Statt in jedem Schritt $f'(x_n)$ zu berechnen wird immer wieder $f'(x_0)$ verwandt:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

Allerdings konvergiert dieser Ansatz nicht so schnell.

Lösung 2: *Sekantenverfahren*
Statt der Steigung im Punkt x_n wird die Aleitung durch Differenzenbildung berechnet:

$$x_2 = x_1 - \frac{f(x_1)\ (x_1 - x_0)}{f(x_1) - f(x_0)}$$

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

Hier wird also ebenfalls keine Ableitung benötigt; allerdings werden zwei Startwerte benötigt. Die Konvergenzgeschwindigkeit ist nicht ganz so gut wie beim Newton-Verfahren, allerdings ist der Rechenaufwand gering. (In jedem Schritt muss nur eine Funktionsauswertung vorgenommen werden.) Im Allgemeinen ist das Sekantenverfahren besser als das Newton-Verfahren.

2.4 Konvergenzkriterien

Die Effizienz von Nullstellensuche hängt von der Konvergenzgeschwindigkeit ab.
Definition: Sei x_n eine Folge mit $lim_{n\to\infty} = \bar{x}$, dann hat die Folge eine Konvergenzordnung $q \geq 1$ wenn es die Konstante $c > 0$ gibt mit:

$$|x_{n+1}| \leq c\ |x_n - \bar{x}|^q \text{für alle n}$$

Für $q = 1$ muss auch $c < 1$ gelten. Aus $q = 1$ folgt die lineare Konvergenz aus $q = 2$ die quadratische.

Beispiel:
Sei x_n eine lineare, konvergente Folge und y_n eien quadratisch, konvergente Folge mit den Grenzwerten x_n und $\bar{x}$. Wir starten mit dem gleichen Abstand zur NS: $|x_n - \bar{x}| \leq 0,1$, $|y_n - \bar{x}| \leq 0,1$. Im nächsten Schritt: $|x_{n+1} - \bar{x}| \leq c \cdot 0,1$ und $|y_{n+1} - \bar{x}| \leq c \cdot 0,01$. Das quadratische Verfahren konvergiert also deutlich schneller.
Bemerkung: Für einfache Nullstellen konvergiert das Newtonverfahren quadratisch, das Sekantenverfahren mit $q = \frac{\sqrt{5}+1}{2}$ und das vereinfachte Newton-Verfahren linear.

Außerdem lässt sich für mehrfache Nullstellen das Newton-Verfahren verbessert werden und eine quadratische Konvergenz erreicht werden:

$$x_{n+1} = x_n + n\ \frac{f(x_n)}{f'(x_n)},$$

wobei m die Vielfachheit der Nullstelle beschreibt.

Cave: numerische Auslöschung bei $f'(x_n) \approx 0$.

2.5 Zusammenfassung

Bisektion: Einfaches Verfahren, schlechte Konvergenz, Fehler halbiert sich mit jedem Iterationsschritt. Allerdings wird dieses Verfahren häufig verwendet um die Nullstelle einzugrenzen und dann ein Verfahren mit besserer Konvergenz zu nutzen.

Fixpunktiteration: Linearkonvergenz bei anziehenden Nullstellen, aber Vorsicht bei abstoßenden Nullstellen. Diese Methode ist allerdings einfach anzuwenden und numerisch günstig, da immer nur eine Rechenoperation ausgeführt werden muss.

Newton-Verfahren: Quadratische Konvergenz bei einfachen Nullstellen, allerdings die die Berechnung aufwendig, da ABleitungen gebildet werden müssen. Ein Vorteil ist die Möglichkeit der Anwendung auf mehrdimensionale Probleme, allerdings benötigt dann jede Iteration die Lösung eines linearen Gleichungssystems.

Vereinfachtes Newton-Verfahren: Weniger Rechenaufwand, da die Ableitung nicht neu berechnet werden muss allerdings nur lineare Konvergenz.

Sekantenverfahren: Eine relativ gute Konvergenz mit $q \approx 1,618$ und auch gute numerische Effizienz, da nicht die Ableittung sondern die Differenzenquotient genutzt wird. Allerdings muss die Auslöschung bei Nullstellen beachtet werden: $\frac{f(x_n)}{f'(x_n)} \approx \frac{0}{0}$.

3 Lineare Gleichungssysteme

WIr betrachten ein Gleichungssystem $a_{11}x_1+a_{12}x_2+...+a_{1n}x_n = b_1$, $a_{21}x_1+a_{22}x_2+...+a_{2n}x_n = b_2$ und $a_{31}x_1 + a_{32}x_2 + ... + a_{3n}x_n = b_3$. Ein solches Gleichungssytem ist exakt lösbar wenn die Anzahl der unabhängigen Gleichungen gleich der Anzahl der Unbekannten ist. Die Lösung von größeren Gleichungssystem erfordert numerische Verfahren, dabei unterscheidet man zwischen zwei verschiedenen Klassen.

Direkte Verfahren: In einer endlichen Anzahl von Schritten erhält man die exakte Lösung im Rahmen der numerischen Genauigkeit.

Iterative Verfahren: Hier wird eine Folge b_k von Lösungsvektoren erzeugt die gegen b konvergiert.

3.1 Gauß-Verfahren

Ein Gleichungssystem der Form $\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ 0 & a_{22} & ... & a_{2n} \\ . & . & & \\ . & . & & \\ . & . & & \\ 0 & 0 & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ x_n \end{pmatrix}$

Um eine beliebige Matrix auf diese dieser Form zu bringen können die Zeilen einzeln durch elementare Zeilenoperationen umgeformt werden:

$$z_j = z_j - \frac{a_{ji}}{a_{ii}},$$

für Spalte i und Zeile j mit $i = 1, .., n$ und $j = i + 1, ..., n$.

Das Ergebnis lässt sich dann durch Rückeinsetzung lösen.

Bemerkung: DIe Anzahl der benötigten Schritte wächst beim Gauß-Verfahren kubisch mit der Größe des LGS n: $\sigma(n^3)$.

Beispiel: Fehlerminimierung mit Pivotisierung $A = \begin{pmatrix} -16^{-4} & 1 \\ 2 & 1 \end{pmatrix}$ und $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, wir rechnen in 4-stelliger Dezimalgenauigkeit. Aus dem Gauß-Algorithmus folgt dann:

$$z_2 = [0, 20000 + 1, 20000]$$

Beim Rückeinsetzen ergibt sich dann:

$$x_2 = \frac{20000}{20000+1} \approx 1$$
$$x_1 = \frac{1}{-10^{-4}} \cdot (1-1) = 0.$$

Die exakten Lösungen wären allerdings $x_1 = -0,499975$ und $x_2 = 0,99995$.

Um einen kleineren Fehler zu erhalten werden zunächst die erste und die zweite Zeile getauscht. Daraus ergibt sich dann:

$$z_2 = [-10^{-4+10^{-5}} \approx 0,1 + 0,5\ 10^{-4} \approx 1,1 + 0]$$
$$x_2 = 1$$
$$x_1 = -\frac{1}{2}.$$

Satz: Der Fehler im Gauß-Verfahren wird durch die sogenannte *Spaltenpivotisierung* minimiert. Dabei werden vor jedem Eliminationsschritt für die i-te Spalte die Zeilen des LGS so umsortiert, dass gilt:

$$|a_{ii}| = \max\{|a_{ij}|, j = 1, ...n\}.$$

Dann gilt im Eliminationsschritt für den Fehler. $|\frac{a_{ji}}{a_{ii}}| \leq 1$. Dann wird der Gauß-Algorithmus für die Spalte i angewandt und für $i+1$ erneut auf das betragsmäßig größte Element in den Zeilen $j = i+1, ...n$ geprüft.

3.2 LR(LU)-Zerlegung

Möchte man mehrere LGS mit der selben Koeffizientenmatrix aber anderer rechten Seite lösen so empfiehlt es sich die Elementaren Umformungen zu merken.

Für $A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & -2 & 2 \\ -3 & 1 & 4 \end{pmatrix}$ führten wir durch:

$$z_2 = z_2 - 6z_1 \to L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$z_3 = z_3 + 3z_1 \to L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$
$$z_3 = z_3 + \frac{1}{2}z_2 \to L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}.$$

Daraus folgt dann:

$$L = L_3\ (L_2 \cdot L_1) = \begin{pmatrix} 1 & 0 & 0 \\ -6 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}.$$

Damit ergibt sich dann die rechts-obere Dreiecksmatrix R zu:

$$R = L(A)$$

So lässt sich dann für ein beliebiges LGS $A\ x = c$ der Lösungsvektor aus $L(c)$ berechne denn da die linke Seite von

$$R = L(c)$$

bereits eine rechts-obere Dreiecksmatrix ist kann die Gleichung direkt durch Rückeinsetzen gelöst werden.

Satz Für jede nxn-Matrix, für die der Gauß-Algorithmus ohne Zeilenvertauschungen durchführbar ist, gibt es nxn-Matrizen L und R mit den Eigenschaften:

- L ist eine links-untere Dreiecksmatrix mit l_{22} für $i = 1, ..., n$
- R ist eine rechts-obere Dreiecksmatrix mit $r_{22} \neq 0$ für $i = 1, ..., n$
- $A = L^{-1}\ R$ bezeichnet man als LR-Zerlegung von A.

Es gilt:

$$A\ x = b \Leftrightarrow Lb = y \quad \text{und} \quad Rx = y$$

3.3 Cholesky-Zerlegung

Definition: Eine symmetrische nxn-MatrixA heißt positiv-definit wenn gilt:

$$x^T Ax > 0$$

SatzFür jede positiv-definite Matrix A gibt es genau eine rechts-obere Dreiecksform mit $r_{ii} > 0 \quad$ für $\quad 1, ..., n$ und $A = R^T R$, diese Zerlegung nennt man Cholesky-Zerlegung.
Die Berechnung der Zerlegung erfolgt wie folgt:[1]

```
For i = 1 To n
    For j = 1 To i
        Summe = a(i, j)
        For k = 1 To j-1
            Summe = Summe - a(i, k) * a(j, k)
        If i > j Then
            a(i, j) = Summe / a(j, j)
        Else If Summe > 0 Then
            a(i, i) = Sqrt(Summe)
        Else
            ERROR
```

[1] http://de.wikipedia.org/wiki/Cholesky-Zerlegung#Pseudocode

Damit ergibt sich für $A = \begin{pmatrix} 4 & 4 & 2 \\ 4 & 5 & 5 \\ 4 & 5 & 26 \end{pmatrix}$, $R = \begin{pmatrix} 2 & 2 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 4 \end{pmatrix}$.

Bermerkung Der numerische Aufwand des Verfahrens beträgt $(\frac{1}{6}n^3 + \frac{1}{2}n^2 - \frac{2}{3}n)$ und eine Wurzelberechnungen. Das Gauß-Verfahren benötigt$(\frac{n^3}{3} + \frac{n}{3})$, ist also für $n \geq 2$ langsamer.

3.3.1 Fehlerrechnung bei LGSW

WIr wollen untersuchen wie sich Fehler in einem LGS auf dessen Lösung auswirken, dazu benötigen wir ein Maß. **Definition:** Eine Abbildung $||\ ||: \mathbb{R}^n \to \mathbb{R}$ heißt Vektornorm, wenn für alle $x, y \in \mathbb{R}^n$ und alle $\lambda \in \mathbb{R}$ gilt:

- $||x|| \geq 0$ und $||x|| = 0$ wenn $x = 0$
- $||\lambda x|| = |\lambda| ||x||$
- $||x + y|| \leq ||x|| + ||y||$

Beispiele hierfür sind Summennorm, Maximunmnorm und die euklidische Norm. Diese sind äquivalent wenn gilt:...

Entsprechend lassen sich auch Matrixnormen definieren: Spaltensummennorm Spektralnorm, Zeilensummennorm.

Bemerkung: Matrixnormen erfüllen die Eigenschaften der zugrundliegenden Vektornormen, insbesondere auch die Äquivalenz:

$$||Ax||_v \leq ||A||_v ||x||_v,$$

man sagt dann: die Norm ist kompatibel.

Satz Sei $||\ ||$ eine Norm und A eine reguläre nxn Matrix und:$A\ x = b$ und $A\ x' = b'$

Dann gilt:

$$\begin{aligned} A(x - x') &= b - b' \\ x - x' &= A^{-1}(b - b') \\ ||x - x'|| &\leq ||A^{-1}||\ |b - b'|. \end{aligned}$$

Und ensprechend mit der MUltiplikation von $||A||||x|| \geq ||b|| \to ||\frac{1}{x}|| \leq \frac{||A||}{||b||}$:

$$\frac{||x - x'||}{||x||} \leq ||A||||A^{-1}|| \cdot \frac{||b - b'||}{||b||}.$$

Man nennt $||A||\ ||A^{-1}|| = cond(A)$ die Konditionszahl der Matrix A bzgl. der verwendeten Norm.

Bermerkung: Die Koordinationszahl gibt also die max. Verstärkung des relativen Fehleran, währen $||A^{-1}||$ die maximale Verstärkung des absoluten Faktors ist.

Satz: Ist nicht nur die rechte Seite eines LGS fehlerbehaftet, sondern auch die Koeffizientenmatrix, so gilt für die Lösungen der beiden LGS $A\ x = b$ und $A'\ x' = b'$ mit $\Delta A = A - A'$ und $\Delta x = x - x'$

$$\frac{||\Delta x||}{||x||} \leq \frac{cond(A)}{1 - cond(A)\frac{||\Delta|A|]}{A}}(norm(deltA(A + norm(deltab/b)$$

3.4 Iterative Verfahren

Idee: Das Linerares Gleichungssystem $Ax = b$ ist äquivalent zu einem vektoriellen Nullstellenproblem:

$$Ax - b = 0.$$

Anstelle des Nullstellenproblems betrachtet man dann das Fixpunktproblem. Um auf der rechten Seite der Matrix ein x zu erzeugen muss die Matrix zerlegt werden: $A = I + A - I$, mit der Einheitsmatrix I. Daraus folgt:

$$\begin{aligned} 0 = (I + A - I)x - b \Rightarrow Ix &= (I - A)x + b \\ \Rightarrow x &= (I - A)x + b. \end{aligned}$$

Dies entspricht dann einer vektoriellen Fixpunktgleichung, welche als Iterationsgleichung aufgefasst werden kann.

Mit einem Startwer $x^{(0)}$ folgt damit:

$$\begin{aligned} x^{(1)} &= (I - A)x^{(0)} + b, \\ x^{(n+1)} &= (I - A)x^{(n)} + b. \end{aligned}$$

Ein Fixpunkt der Itertaionsgleichung einspricht dann also der Lösung des LGS.

Beispiel:

Aus $A = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix}$, $b = \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$ folgt: $x^{(n+1)} = \begin{pmatrix} -3 & 1 & -1 \\ 2 & -4 & -1 \\ -1 & 2 & -4 \end{pmatrix} x^{(n)} + \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}$, was nicht konvergiert.

3.5 Jacobi Verfahren(Gesamtschnitt)

Betrachte Zerlegung $A = \underbrace{\begin{pmatrix} 0 & 0 & 0 & . \\ a_{21} & 0 & 0 & . \\ a_{31} & a_{32} & 0 & . \\ . & . & . & . \end{pmatrix}}_{L} + \underbrace{\begin{pmatrix} a_{11} & 0 & 0 & . \\ 0 & a_{22} & 0 & . \\ 0 & 0 & a_{33} & . \\ . & . & . & . \end{pmatrix}}_{D} + \underbrace{\begin{pmatrix} 0 & a_{12} & a_{13} & . \\ 0 & 0 & a_{23} & . \\ 0 & 0 & 0 & . \\ . & . & . & . \end{pmatrix}}_{R}$.

Daraus ergibt sich dann:

$$\begin{aligned} Ax - b = 0 = (L + D + R)x - b \\ \Rightarrow Dx = -(L + R)x + b \\ \Rightarrow x = -D^{-1}(L + R)x + D^{-1}b \\ \Rightarrow x^{(n+1)} = -D^{-1}(L + R)x^{(n)} + D^{(-1)}b \end{aligned}$$

Dies liefert für unser Beispiel von vorher:

$$x^{(n+1)} = \begin{pmatrix} 0 & 0,25 & -0,25 \\ 0,4 & 0 & -0,2 \\ -0,2 & 0,4 & 0 \end{pmatrix} x^{(n)} + \begin{pmatrix} 1,25 \\ 2,2 \\ 2,4 \end{pmatrix}.$$

Dies liefert dann:

i	0	1	2	3	4	5
$x^{(i)}$	0	1,25	1,2	1,0475	1,0065	0,9973
	0	2,2	2,22	2,074	2,0094	1,9986
	0	2,4	3,03	3,048	3,0201	3,0024

dies konvergiert gegen: $\bar{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

3.5.1 Einzelschrittverfahren (Gauß-Seidel Verfahren)

Jacobi-Verfahren in Komponenten:

$$\begin{aligned} x_1^{(n+1)} &= 0,25x_2^{(n)} - 0,25x_3^{(n)} + 1,25 \\ x_2^{(n+1)} &= 0,4x_1^{(n)} - 0,2x_3^{(n)} + 2,2 \\ x_3^{(n+1)} &= -0,2x_1^{(n)} + 0,4x_2^{(n)} + 2,4. \end{aligned}$$

Wenn man annimmt, dass $x^{(n+1)}$ komponentenweise näher am Lösungsvekotr liegt, sollte man in der zweiten Gleichung x_1^{n+1} u.s.w. benutzen, damit ergibt sich:

$$\begin{aligned} x_1^{(n+1)} &= 0,25x_2^{(n)} - 0,25x_3^{(n)} + 1,25 \\ x_2^{(n+1)} &= 0,4x_1^{(n+1)} - 0,2x_3^{(n)} + 2,2 \\ x_3^{(n+1)} &= -0,2x_1^{(n+1)} + 0,4x_2^{(n+1)} + 2,4. \end{aligned}$$

In Matrix Schreibweise:

$$x^{(n+1)} = \begin{pmatrix} 0 & 0,25 & -0,25 \\ 0 & 0 & -0,2 \\ 0 & 0 & 0 \end{pmatrix} x^{(n)} + \begin{pmatrix} 0 & 0 & 0 \\ 0,4 & 0 & -0 \\ -0,2 & 0,4 & 0 \end{pmatrix} x^{n+1} + \begin{pmatrix} 1,25 \\ 2,2 \\ 2,4 \end{pmatrix},$$

oder einfacher:

$$\begin{aligned} x^{(n+1)} &= -D^{-1}(Rx^{(n)} + Lx^{(n+1)} - b) \\ \Leftrightarrow (D+L)x^{(n+1)} &= -Rx^{(n)} + b \\ \Leftrightarrow x^{(n+1)} &= -(D+L)^{-1}Rx^{(n)} + (D+L)^{-1}b. \end{aligned}$$

In der Praxis wird nicht das Inverse der Matrix $(D+L)$ berechnet, sondern das Gleichungssystem $(D+L)x^{(n+1)} = -Rx^{(n)} + b$ wird gelöst(durch Vorwärtseinsetzen).

Zurück zum Beispiel:

0	1	2	3	4
0	1,25	1,1175	1,006	1,001
0	2,7	2,001	2,007	2,0002
0	3,24	2,977	3,002	2,9998

, konvergiert also schneller.

Konvergenz von linearen Iterationsverfahren

Gegeben sei eine lineare Matrix-Fixpunkt-Iterationsgleichung $x^{n+1} = Bx^{(n)} + b$, wobei B eine mxm Matrix, $b \in \mathbb{R}^m$ und $\bar{x}$ ein Fixpunkt mit: $\bar{x} = B\bar{x} + b$.
Sei $||.||$ eine Matrixnorm, dann gilt:
$\bar{x}$ anziehender Fixpunkt falls $||B|| < 1$
$\bar{x}$ abstoßender Fixpunkt falls $||B|| > 1$

Matrix-Fixpunktsatz für anziehenden Fixpunkt: Fixpunktiteration konvergiert für alle Startwerte.
Bemerkungen: F+r Gesamtschrittverfahren (Jacobi) gilt $B = -D^{-1}(L + R)$. Für die ∞-Norm (Zeilensummennorm)gilt:

$$||B||_\infty = max_{i,m} \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} = max_{i,m} \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1$$
$$\Leftrightarrow |a_{ii}| > \sum_{j \neq i} |a_{ij}|,$$

für alle i, dies wird Zeilensummenkriterium genannt. Eine solche Matrix heißt diagonaldominant.

Im Einzelschrittverfahren ist $B = -(D + L)^{-1}R$ und man kann zeigen, dass $||(D + L)^{-1}R||_\infty \leq ||D^{-1}(L + R)||_\infty$. Konvergiert also das Gesamtschrittverfahren so konvergiert auch dasd Einzelschrittverfahren, die Umkehrung gilt nicht.

3.6 Nichtlineare Gleichungssyteme-Newton Verfahren

häufig: n nicht-lineare Gleichungen mit n Unbekannten
gegeben: vektorielle Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$
gesucht: Vektor $\bar{x} \in \mathbb{R}^n$ mit $f(\bar{x}) = 0$.
$f(x) = f(x_1, ..., x_n) = \begin{pmatrix} f_1(x_1, ..., x_n) \\ f_2(x_1, ..., x_n) \\ ... \\ f_3(x_1, ..., x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ ... \\ 0 \end{pmatrix}$. Es gibt kein einfaches Verfahren um zu prüfen ob ein Gleichungs-System lösbar ist und wieviele Lösungen es gibt. $\to$ verallgemeinere Newton-Verfahren auf n-Dimensionen.

$$g : \mathbb{R} \to \mathbb{R} \Rightarrow g(x) \approx g(x_0) + (x - x_0)g'(x_0) = 0$$
$$f : \mathbb{R}^n \to \mathbb{R}^n \Rightarrow f(x) \approx g(x^{(0)}) + (x - x^{(0)})D(f(x^{(0)})) = 0$$

Zu lösen ist also das Gleichungssystem:

$$x = x^{(0)} - (D(f(x^{(0)})))^{-1} f(x^{(0)})$$
$$x^{(n+1)} = x^{(n)} - (D(f(x^{(n)})))^{-1} f(x^{(n)}).$$

Dies wird als Newton-Verfahren bezeichnet.

Jacobi Matrix

Die Jacobi Matrix der partiellen Ableitungen ist definiert als:

$$D(f(x)) = \frac{\partial f_i(x)}{\partial x_j} = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ . & & & \\ . & & & \\ . & & & \\ \frac{\partial f_n(x)}{\partial x_1} & \dots & & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix}.$$

So ergibt sich mit $f(x_1, x_2) = \begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 + +8x_2^3 \end{pmatrix}$ und $x^{(0)} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$ $\bar{x} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$

Satz: Das Newton Verfahren konvergiert quadratisch wenn:

- $x^{(0)}$ nahe genug an $\bar{x}$ liegt
- $D(f(\bar{x}))$ regulär ist
- f dreimal stetig differenzierbar ist

Achtung: Nicht immer konvergiert das Newton Verfahren gegen eine Nullstelle. Ein Beispiel hierfür ist die Funktion: $f = \begin{pmatrix} x_1^3 - x_2 - 1 \\ x_1^2 - x_2 \end{pmatrix}$. Die liegt daran, dass die zugehörige Jacobi-Matrix nicht regulär (nicht invertierbar) ist.

Bemerkung: in der Praxis werden partielle Ableitungen durch Differenzieren angenähert: $D(f(x))_{ij} = \frac{f_i(x+h\ e_j) - f_i(x)}{h}$, mit dem Einheitsvektor in j-Richtung e_j.

Vereinfachtes Newton-Verfahren

Der Rechenaufwand pro Schritt wird dadurch minimiert, dass stets $D(f(x^{(0)}))$ benutzt wird, dadurch ist die Konvergenz nur noch linear.

4 Interpolation und Ausgleichsrechnung

In vielen Anwendungen: Messdaten sollen durch eine Formel beschrieben werden, zum Beispiel um Integrale, Differentiale, etc. zu berechnen.
Definition: gegeben seien $n+1$ Wertepaare (x_1, f_i) mit $i = (0, ..., n)$; gesuchten stetigen Funktionen $f(x)$ mit $f(x_i) = f_i$ für alle $i = 0, .., n$.
$\{x_n\}$: Stützstellen, $\{f_n\}$: Stützwerte; $\{x_n, f_n\}$ Stützpunkte und $f(x)$ Interpolierende der Stützpunkte.
Bemerkung: Das Interpolationsproblem ist nicht eindeutig.

4.0.1 Polynom-Interpolation

Ein Polynom n-ten Grades besitzt $n + 1$ Freiheitsgrade, es ist also möglich die $n + 1$ Koeffizienten des Polynoms so zu wählen, dass $f(x_i) = f_i$ ist!

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

Satz: Gegeben seien $n+1$ Wertepaare (x_i, f_i), dann gibt es genau ein Polynom vom Grad höchstens n so dass das Polynom $p(x_i) = f_i$ für alle i. Hierbei spricht man bei $n = 1$ von einer linearen, bei $n = 2$ von einer quadratischen Interpolation und so weiter.

Ineffiziente Methode:

Explizite Lösung eines linearen Gleichungssystems.
Beispiel:

$$p(x) = \sum_{i=0}^{3} a_i x^i$$
$$f_j = \sum_{i=0}^{3} a_i x_j^i,$$

also 4 Gleichungen für 4 Unbekannte:

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

Als Lösung ergibt sich:

$$\begin{aligned} a_0 = 2, \quad a_1 = 9, \quad a_2 = 9 \\ \Rightarrow p(x) = -2 + 9x + 9x^2 - 7x^3. \end{aligned}$$

Lagrange Form

$$p(x) = \sum_{i=0}^{3} f_i \; l_i(x)$$

mit $l_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}$.
In unserem Beispiel: $l_0(x) = (\frac{x-x_1}{x_0-x_1})(\frac{x-x_2}{x_0-x_2})(\frac{x-x_3}{x_0-x_3})$. Hier ist zu sehen, dass $l_0(x)$ an allen Stützstellen außer an x_0, und $l_0(x_0) = 1$. (analoges gilt für alle anderen Polynome l_1, l_2, l_3) $\rightarrow p(x_i) = f_i$.
Jedes l_1 ist ein Polynom dritten Grades. Die Berechnung von $p(x)$ durch Summation von $l_i(x)$ ist allerdings immer noch aufwendig, es gibt eine schnellere Methode.

Newtonsche Interpolationsformel

Gegeben sind Wertepaare (x_i, f_i) für $i = 0, ...n$. Zunächst werden "dividierte Differenzen"berechnet:
(Pseudocode hier)
Bemerkung: x_i müssen nicht nach Größe sortiert sein außerdem können neue Wertepaare angefügt werden.

In unserem Beispiel ergeben sich die dividierten Differenzen für $k = 1$ zu:

$$f(x_0, x_1) = -7$$
$$f(x_1, x_2) = 11$$
$$f(x_2, x_3) = -13$$

und für für $k = 2$:

$$f(x_0, x_1, x_2) = 9$$
$$f(x_1, x_2, x_3) = -12$$

Und daraus ergibt sich:

$$f(x_0, x_1, x_2, x_3) = -7.$$

Interpolationsformel

$$p(x) = \sum_{i=0}^{n} f(x_0, ..., x_i) \prod_{j=0}^{i-1} (x - x_j)$$
$$= f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_i)$$
$$+ f(x_0, x_1, x_2, x_3)(x - x_0)(x - x_1)(x - x_2)$$

Für unser Beispiel ergibt sich also:

$$p(x) = 5 + (-7)(x - (-1)) + 9(x - (-1))(x - 0) + (-7)(x + 1)x(x - 1)$$
$$= -7x^3 + 9x^2 + 9x - 2$$

Außerdem kann p(x) durch Ausklammern umgeformt werden:

$$p(x) = f(x_0) + (x - x_0)(f(x_0, x_1) + (x - x_1)(f(x_0, x_1, x_2) + (x - x_2)f(x_0, x_1, x_2, x_3))))$$

Durch das einsparen von Punktoperationen kann so Rechenzeit gespart werden.

Iterative Auswertung des Newton Polynoms, Horner Schema

```
r_n=f(x_0,x_1,...,x_n)
For k = n-1,0
        r_k=r_(k+1) (x-x_k)+f(x_0,...,x_k)
End for p(x)=r_0
```

Man benötigt also zur Auswertung des Newton Polynoms nur n Punkt-Operatoren. Daraus ergibt sich für unser Beispiel für:

$$\begin{aligned} r_3 &= f(x_0, x_1, x_2, x_3) = -7 \\ r_2 &= r_3(x - x_2) + f(x_0, x_1, x_2) = -7(x-1) + 9 \\ r_1 &= r_2(x - x_1) + f(x_0, x_1) = (-7(x-1) + 9)x - 7 \\ r_0 &= r_1(x - x_0) + f(x_0) = ((-7(x-1) + 9)x - 7)(x+1) + 5 \end{aligned}$$

Problem bei Polynom-Interpolation

Gegeben sei die Funktion $f(x) = \frac{1}{1+x^2}$ mit:

x_i	-3	-2	-1	0	1	2	3
f_i	0,1	0,2	0,5	1			

$\Rightarrow p(x) = 1 - 0,64x^2 + 0,15x^4 - 0,06x^6$ Das Polynom weißt an den Rändern des Stützstellenintervalls Überschwinger auf. Erhöhung von n verbessert die Situation nicht. Außerdem divergiert das Polynom außerhalb des Intervalls.
→ Interpolationspolynome sind nicht für Extrapolation geeignet.

4.1 Spline Interpolation

Ausgangsüberlegung: eine gute Näherung einer Funktion hat viele Stützstellen, allerdings besitzt ein Polynom hoher Ordnung auch viele Überschwingungen und divergiert stark.
Idee: Teile Intervall in Teilintervalle auf, und benutze jeweils Polynome niedriger Ordnung.

- **Einfachste Stufe:** Stückweise Interpolation mit Geraden, hierbei entstehen allerdings Knicke.

- **Kubische Splines:** Hier wird zur ABhilfe eine stetige erste und zweite Ableitung gefordert.

Definition: eine Funktion $s(x) : [a, b] \to \mathbb{R}$ heißt kubischer Spline zu den Stützstellen $a = x_0 < x_1, ... < x_n = b$ falls gilt:

- $s''(x)$ existiert und ist stetig

- $s(x)$ ist auf den Intervallen $[x_i, x_{i+1}]$ jeweils ein Polynom 3. Grades
- interpolieren der Spline Werten $f_0 ... f_n : s(x_i) = f_i \quad \text{für} \quad i = 0, ..., n$
- periodischer Spline: zusätzlich $s(a) = s(b)$ und $s'(a) = s'(b)$ und $s''(a) = s''(b)$
- natürlicher Spline: zusätzlich $s''(a) = s''(b) = 0$

Bestimmung der Koeffizienten

Ein Polynom 3. Grades hat 4 Koeffizienten $\hat{=}$ Freiheitsgrade; für n Intervalle haben wir also $4n$ Koeffizienten!

- Pro Intervall 2 Interpolationsbedingungen (rechts und links) $\rightarrow 2n$
- An den Stützstellen $x_1, ..., x_{n-1}$ sollen 1. und 2. Ableitungen gleich sein $\rightarrow 2(n-1)$ Bedingungen, zusammen gibt es also $4n - 2$ Bedingungen.

Es sind also 2 zusätzliche Randbedingungen nötig.

Praktische Berechnung von kubischen Splines

Wir definieren die "Momenteëines interpolierenden Splines als $M_i = s''(x_i)$. Mit diesen Momenten erhalten wir im Intervall $[x_{i-1}, x_i], i = 1, ..., n$.

$$S_i(x) = M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + C_i(x - \frac{x_{i-1} + x_i}{2}) + D_i$$

$$\rightarrow s_i'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i + C_i}$$

$$\rightarrow s_i''(x) = M_{i-1}\frac{x_i - x}{h_i} + M_i\frac{(x - x_{i-1})}{h_i}$$

$$h_i = x_i - x_{i-1}$$

Die Ableitungen stimmen dabei an den Stützstellen überein:
z.B.: $s_i''(x = x_i) = s_{i+1}''(x = x_i) \Leftrightarrow M_i = M_i$
$s_i'(x_i) = s_{i+1}'(x_i)$ und $s_i(x_i) = s_{i+1}(x_i)$ führen zu:

$$M_i M_{i-1} + 2M_i + (1 - \mu_i)M_{i+1} = 6f(x_{i-1}, x_i, x_{i+1})$$

$$C_i = \frac{f_i - f_{i-1}}{hi} - \frac{h_i}{6}(M_i - M_{i-1}$$

$$D_i = \frac{f_i + f_{i-1}}{2} - \frac{h_i^2}{12}(M_i + M_{i-1})$$

$$\mu_i = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}$$

Zusätzlich muss gelten:

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

- natürliche splines: $M_0 = M_n = 0$
- preiodische splines: $M_0 = M_n$, $\mu_n M_{n-1} + 2M_n + (1-\mu_n)M_n = 6f(x_{n-1}, x_n, x_{n+1})$

wobei $x_{n+1} := x_n + x_1 - x_0$ und $f_{n+1} := f_n$. Mit dem Beispiel ergibt sich daraus:

$$\begin{aligned} f(x_0, x_1, x_2) &= 9 \\ f(x_1, x_2, x_3) &= -12 \\ \mu_i &= \frac{1}{2} \\ \text{natürlicher spline:} M_0 = M_3 = 0 \rightarrow \frac{1}{2}M_0 + 2M_i + \frac{1}{2}M_2 &= 54 \\ \frac{1}{2}M_i + 2M_2 + \frac{1}{2}M_3 &= -72 \end{aligned}$$

Dieses Gleichungssystem ist dann zu lösen nachdem sich C_i und D_i durch einsetzen ergeben. Für $n > 3$: lineares Gleichungssystem für M_n! Daraus resultieren 3 Splines für die 3 Intervalle:

$$\begin{aligned} [-1,0] : s(x) &= 38,4\frac{(x+1)^3}{6} + C_i(x+0,5) + D_1 \\ [0,1] : s(x) &= 38,4\frac{(x-1)^3}{6} - 45,6\frac{x^3}{6} + C_2(x-0,5) + D_2 \\ [1,2] : s(x) &= -45,6\frac{(2-x)^3}{6} + C_3(x-1,5) + D_3 \end{aligned}$$

Zurück zum Beispiel $f(x) = \frac{1}{1+x^2}$. (Problem der Überschwinger). Der kubische Spline erzeugt hierbei keine Überschwinger.

4.2 Lineare Ausgleichsrechnung

gegeben: n Wertepaare $(x_i, y_i), i = 1, ..., n$ mit $x_i \neq x_j$ für $i \neq j$
gesucht: stetige Funktion so dass möglichste genau $f(x_i) \approx y_i$ für alle i. Wenn man die zulässigen Funktionen nicht einschränkt dann gelangt man zum Interpolationsproblem mit $f(x_i) = y_i$. Wir schränken also den absoluten Funktionsraum ein.
Def.: gegebene Menge F von stetigen Funktionen sowie n Wertepaare (x_i, y_i); Funktionen $f \in F$ heißt Ausgleichsfunktion falls das Fehlerfunktional:

$$E(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

minimiert wird.
Bemerkung:

x_i	1	2	3	4
y_i	6	6,8	10	10,5

- häufig mehr Funktionswerte als Anpassparameter
- Funktion f ist optimal bzgl.der 2er-Norm
- es ist möglich die Wertepaare zu wichten: $E(f) = \omega_i \sum_{i=1}^{n}(y_i - f(x_i))^2$ mit $\omega_i \in [0,1]$
- für F =Menge aller Geraden $\rightarrow f$ = Ausgleichs- oder Regressionsgerade.

Beispiel: Wertepaare:
Wie lautet die Ausgleichsgerade?
Funktion $y = ax + b$ oder allgemein: $F := \{a_1 f_1 + a2 f_2 \| a_1, a_2 \in \mathbb{R}$ mit Ansatzfunktionen $f_1(x) = x$ und $f_2(x) = 1$.
Aufgrund der Linearität in den Koeffizienten a_1 und a_2 wird dies lineares Problem genannt. Daraus ergibt sich das Fehlerintervall: $E(a,b) = \sum_i^4 (y_i - (ax_i + b))^2$:

$$\frac{\partial E}{\partial a} = -2\sum_{i=1}^{n}(y_i(ax_i + b))x_i = 0$$
$$\rightarrow a\sum_i x_i^2 + b\sum_i x_i = \sum_i x_i y_i$$
$$\frac{\partial E}{\partial b} = -2\sum_{i=1}^{n}(y_i - (ax_i + b_i)) = 0$$
$$\rightarrow a\sum_i x_i + b\sum_i 1 = \sum_i y_i$$

Also ein lineares Gleichungssystem für a und b:

$$\begin{pmatrix} \sum x_i^2 & \sum_i x_i \\ \sum x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{pmatrix}$$
$$\rightarrow \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 91,6 \\ 33,3 \end{pmatrix}$$

Die allgemeine Definition des Ausgleichproblems ergibt sich wie folgt:
gegeben: Ansatzfunktion: $f = \sum_{i=1}^{m} a_i f_i$ mit $a_i \in \mathbb{R}$
und Wertepaare (x_i, y_i) mit $i = 1...n$
Fehlerintervall: $E(a_n) = \sum_{i=1}^{n}(y_i - f(x_i))^2 = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{m} a_j f_j(x_i))^2 = ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{a}||_2$,
mit:

$$A = \begin{pmatrix} f_1(x_1) & f_2(x_1) & ... & f_m(x_1) \\ . & & & . \\ . & & & . \\ . & & & . \\ f_1(x_n) & ... & & f_m(x_n) \end{pmatrix} a = \begin{pmatrix} a_1 \\ a_2 \\ . \\ . \\ a_m \end{pmatrix}, y = \begin{pmatrix} y_1 \\ . \\ . \\ . \\ a_m \end{pmatrix}$$

$\boldsymbol{y} = \boldsymbol{Aa}$ nennt man das Fehlergleichungssystem und typischerweise gibt es keine Lösung, da sonst der Fehler 0 wäre → Interpolation.
Die Gleichungen $0 = \frac{\partial E(\{a_m\})}{\partial a_i}$ mit $i = 1...m$ heißen Normalgleichungen. Das Normalgleichungssystem ergibt sich als: $A^T Aa = A^T y$
Herleitung:

$$\begin{aligned} E(\{a_m\}) = (\boldsymbol{y} - \boldsymbol{Aa}))^2 &= (y_i - A_{ij}a_j)(y_i - A_{ik}a_k) \\ &= y_i^2 - A_{ij}a_j y_i - y_i A_{ik}a_k + A_{ij}a_j A_{ik}a_k \\ &= y_i^2 - 2A_{ij}a_j y_i + A_{ij}a_j A_{ik}a_k \\ \frac{\partial E}{\partial a_l} &= -2A_{il}y_i + A_{il}A_{ik}a_k + A_{ij}a_j A_{il} \\ = -2A_{il}y_i + 2A_{il}A_{ik}a_k) = 0 &\rightarrow A_{il}A_{ik}a_k = A_{il}y_i \end{aligned}$$

$$\leftrightarrow A_{li}^T A_{ik}a_k = A_{li}^T y_i \leftrightarrow \underbrace{\boldsymbol{A}^T\boldsymbol{A}}_{\text{mxm}} \quad \underbrace{\boldsymbol{a}}_{\text{m Komponenten}} \quad = \quad \underbrace{\boldsymbol{A}^T}_{\text{m Komponenten}} \quad \underbrace{\boldsymbol{y}}_{\text{n Komponenten}}$$

Bemerkungen: Das Fehlergleichungssystem besteht aus n-Gleichungen(=Wertepaare), allerdings gibt es nur m Unbekannte(Anpassungsparamter). $n > m \rightarrow$ das Fehlergleichungssystem ist überbestimmt, es gibt also im allgemeinen keine Lösung.
Man bezeichnet $r = Aa - y$, als Residuumsvektor welcher im Allgemeinen nicht verschwindet.
Im Fall $r = 0$ ist a Lösung des Fehlergleichungssystems und es liegt eine perfekt Modellanpassung vor (Interpolation). Die ist typischerweise der Fall für $m = n$ (Außer wenn Anpassungsfunktionen oder Wertepaare redundant sind...).

Also bedeutet die Lösung des Ausgleichproblems, dass das Fehlerfunktional bzgl. $\boldsymbol{a}$ minimal ist. Die entspricht dann der Lösung des Normalengleichungssystems. Dies entspricht einem Gleichungssystem aus $A^T A$ und $A^T y \rightarrow$ Cholesky Dekomposition möglich, der Lösungsvektor ist dann $\boldsymbol{a}$. Die Symmetrie für die CHolesky Dekomposition ist gegeben weil $B_{ik} = A_{ij}^T A_{jk} = A_{ki}^T A_{ji} = A_{jk}A_{ij}^T = A_{ij}^T A_{jk}$ In unserem Beispiel ergab sich:

x_i	1	2	3	4
y_i	6	6,8	10	10,5

Als Anfangsfunktionen nutzen wir: $f_1(x) = x$ und $f_2(x) = 1$.

$$A = \begin{pmatrix} f_1(x_1) & f_2(x_1) \\ f_1(x_2) & f_2(x_2) \\ f_1(x_3) & f_2(x_3) \\ f_1(x_4) & f_2(x_4) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix}, \quad A^T A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix} = \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix}$$

$$A^T y = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 6,8 \\ 10 \\ 10,5 \end{pmatrix} = \begin{pmatrix} 91,6 \\ 33,3 \end{pmatrix}$$

Also:

$$A^T AA = A^T y \leftrightarrow \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 91,6 \\ 33,3 \end{pmatrix}$$

Also das gleiche Gleichungssystem wie vorher. Auch bei einem nicht-linearen Funktionsansatz $f_1(x) = x^2$ und $f_2(x) = 1$ ergibt sich eine lineare Ausgleichsrechnung:

$$A = \begin{pmatrix} 1 & 1 \\ 4 & 1 \\ 9 & 1 \\ 16 & 1 \end{pmatrix}, \quad A^T A = \begin{pmatrix} 1 & 4 & 9 & 16 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 4 & 1 \\ 9 & 1 \\ 16 & 1 \end{pmatrix} = \begin{pmatrix} 354 & 30 \\ 30 & 4 \end{pmatrix}$$

$$A^T y = \begin{pmatrix} 1 & 4 & 9 & 16 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 6,8 \\ 10 \\ 10,5 \end{pmatrix} = \begin{pmatrix} 291,2,6 \\ 33,3 \end{pmatrix}$$

Wie versprochen immer noch linear:

$$A^T Aa = A^T y \rightarrow a_1 = 1,296 \quad a_2 = -5,59$$
$$\text{mit: } f(x) = a_1 x^2 + a_2$$

Für unser Beispiel ist der lineare Fit besser, also $|r|$ ist kleiner.

4.3 Nicht-lineare Ausgleichsprobleme, Gauß-Newton Verfahren

Beispiel: Wir betrachten wieder die Wertepaare x_i und y_i und die Funktion $f(x) = ae^{bx}$ mit den Anpassungsparametern a und b. $f(x)$ hängt also nichtlinear von b ab!
Trick: betrachte $ln(f(x)) = ln(a) + bx$ und die logarithmischen Funktionswerte in der Form:

x_i	1	2	3	4
$ln(y_i)$	1,72	1,52	2,30	2,35

Damit ergibt sich: $f(x) = ax^b \rightarrow ln(f)c + bx$

Änhlich kann verwendet werden: $f(x) = ax^b \rightarrow ln(f) = ln(a) + bln(x) = c + b\tilde{x}$

Echtes Beispiel: $f(x) = aln(x + b)$

Definition eines allgemeinen Ausgleichssystems: Gegeben ist eine Ansatzfunktion $f_a(a_1, ...a_m, x)$ und Wertepaare (x_i, y_i) mit $i = 1...n$ mit dem Fehlerfunktional $E(a_I, ...a_m) = \sum_{i_1}^{n} (y_i - f_a(a_{ij}, ...a_{mj}, x_i))^2$
Dasraus ergibt sich das Ausgleichsproblem: Minimiere E im erlaubten Parameterraum.

Bermerkung: im allgemeinen Fall ist ein lineares Problem enthalten. Im Prinzip kann man das nicht-lineare Gleichungssystem $\frac{\partial E}{\partial a_i} = 0$ mit dem Newton-Verfahren lösen; Nachteil: instabil und zweite partielle Ableitungen werden benötigt.
$\rightarrow$ Gauß-Newton-Verfahren

Definiere Vektor $\boldsymbol{g}(a_i, ...a_n) = \begin{pmatrix} y_1 - f_a(a_1, ...a_m, x_1) \\ . \\ . \\ . \\ y_n - f_a(a_1, ...a_m, x_n) \end{pmatrix}$ mit $f : \mathbb{R}^m \to \mathbb{R}^n$. Das Fehlerfunktion ist dann: $E(\boldsymbol{a}) = g^2,\ E : \mathbb{R}^m \to \mathbb{R}$

Problem: Finde $\boldsymbol{a}$ so dass $E(\boldsymbol{a})$ minimal wird. Dies beschreibt ein Quadramittelproblem.

Idee: $g(\boldsymbol{a})$ in E wird durch eine Linearisierung ersetzt: hierzu wird der lineare Ausdruck minimiert. Dieser Prozess wird dann iteriert wie beim Newton-Verfahren.

Starte mit Schätzwert: $\boldsymbol{a}^{(0)}$, das lineare Ausgleichproblem ist dann definiert durch:

$$E(\boldsymbol{a^0}) = (g(\boldsymbol{a}) + D(g(a^0))(\boldsymbol{a} - \boldsymbol{a^0}))^2$$

mit der Jacobi-Matrix D. Berechne dann $\delta^{(n)} = a(n+1) - a^{(n)}$ als Lösung des linearen Ausgleichsproblems, also:

$$\min_{\delta^{(n)}}(g(a^{(n)}) + D(g(a^{(n)}))\delta^{(n)})^2$$

Setze dann $a^{(n+1)} = a^{(n)} + \delta^{(n)}$ und löse:

$$D^T(g(a^{(n)}))D(g(a^{(n)}))\delta^{(n)} = -D^T(g(a^{(n)}))g(a^{(n)})$$

und iteriere.

In der Praxis benutzt man ein gedämpftes Gauß-Newton-Verfahren:

- Berechne wieder $\delta^{(n)}$
- Bestimmung des Dämpfungsfaktor:
  Wähle die größte Zahl $t \in \{1, \frac{1}{2}, \frac{1}{4}, ...\}$ für die mit $\phi(t) = (g(a^{(n)} + t\delta^{(n)}))^2$ gilt: $\phi(t) < \phi(0)$.
- Setze $a^{(n+1)} = a^{(n)} t\delta^{(n)}$ und iteriere.

Bemerkung: Der Dämpfungsfaktor t erhöht den EInzugsbereich des Verfahrens, d.h. das gedämpfte Verfahren konvergiert für einen Bereich von Startvektoren $\boldsymbol{a^{(0)}}$ (vergleiche mit $t = 1$, ungedämpft).

- Fehler wird geringer in jedem Schritt: $(\boldsymbol{g}(\boldsymbol{a}^{(n+1)}))^2 = \phi(t) < \phi(0) = (\boldsymbol{g}(\boldsymbol{a}^{(n)}))^2)^2$
  (aber: Konvergenz zum echten Minimum nicht garantiert.)
- Abbruchbedingung: $\sqrt{(t\boldsymbol{\delta}^{(n)})^2} < \text{TOL}$
  (aber: bedeutet nicht dass man sich im Abstand TOL vom gesuchten Minimum befinde!)

x_i	0	1	2	3	4
$ln(y_i)$	3	1	0,5	0,2	0,05

i	0	1	2	3	4
$a^{(i)})$	1	2,99	1,26	2,91	2,98
	-1,5	0,32	0,279	-0,86	-1,00

Beispiel: mit der Ansatzfunktion: $f(x) = a_1 e^{a_2 x}$.
Die Verteilungsfunktion lautet also: $g(a_1, a_2)_i = y_i - a_1 e^{(a_2 x_i)}$ und die Jacobi Matrix: $D(g) = \frac{\partial g_i(a_m)}{\partial a_j} = (-e^{(a_2 x_i)}, -a_i x_i e^{(a_2 x_i)})$. Wir erwarten: $a_1 > 0, a_2 < 0$.
Ungedämpft $t = 1$:

Für den Startwert $a^{(0)} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ keine Konvergenz.
Gedämpftes Verfahren:

i	0	1	2	3
$a^{(i)})$	1	1,99	2,92	2,98
	-1,5	-0,56	-0,951	-0,999

Das gedämpfte Verfahren ist schneller und konvergenter.

5 Numerische Differentiation/Integration

5.1 Numerische Differentiation

In vielen Anwendungen $\rightarrow f'(x_0), f''(x_0)$ eienr Funktion $f(x)$ werden benötigt, aber diese Funktionen stehen nicht zur Verfügung $\rightarrow$ Näherungen!
Definition: Ableitung als Grenzwert eines Differenzenquotienten: $f'(x_0) = lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} \approx \frac{f(x) - f(x_0)}{h} := D_1 f(x_0, h)$
$D_1 f(x_0, h)$ ist dabei die Differenzenformel oder finite Differenz erster Ordnung.
Herleitung genauer Differenzenformel und Fehlerschätzung mit Taylorentwicklung:

$$f(x_0 + h) = f(x_0) + h f'(x_0) + \frac{1}{2} h^2 f''(x_0) + ... \tag{5.1}$$

$$f(x_0 - h) = f(x_0) + -h f'(x_0) + \frac{1}{2} h^2 f''(x_0) + ... \tag{5.2}$$

Fehlerabschätzung:
Aus der ersten Gleichung ist ersichtlich: $h f'(x_0) = f(x_0 + h) - f(x_0) - \frac{1}{2} h^2 f''(x_0) + ...$

i	0	1	2	3	...	10
$a^{(i)}$	2	0,003	0,004	0,207	...	10
	2	2,00	1,75	-0,79	...	-1,00

$$\rightarrow f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{1}{2}hf''(x_0) + ...$$
$$f'(x_0) = D_1(f(x_0,h)) - \frac{1}{2}hf''(x_0)$$

Definition Diskretisierungsfehler: Bei Differenzenformeln für $f'(x_0)$ bezeichnet man den Fehler $|D_f(x_0,h) - f'(x_0)|$ als Diskretisierungs- oder Abschneidefehler. Die Formel D_f hat die Fehlerordnung k falls es ein $C > 0$ gibt so dass für kleines h gilt:

$$|D_f(x_0,h) - f'(x_0)| \leq Ch^k$$

Als Kurzschreibweise nutzt man für den Fehler: $\sigma(h^k)$. In unserem Beispiel erhalten wir also: $D_1 \rightarrow |D(f(x_0,h)) - f'(x_0)| \approx \frac{1}{2}hf''(x_0) \leq Ch^k$ mit $k = 1$. In unserem Beispiel ist die Fehlerordnung also $k = 1$! Formeln mit höherer Ordnung sind besser, da bei Halbierung von h der Fehler um den Faktor $(\frac{1}{2})^k$ kleiner wird(Vorfaktoren C sind also typischerweise nicht so wichtig).

Herleitung einer besseren Formel für Differentiale:
Berechne:

$$f(x_0+h) - f(x_0-h) = 2hf'(x_0) + \frac{1}{3}h^3f'''(x_0) + ...$$
$$\rightarrow D_2f(x_0,h) := \frac{f(x_0+h)-f(x_0-h)}{2h}$$
$$\text{und} D_2f(x_0,h) - f'(x_0) = \frac{1}{6}h^2f'''(x_0)$$

D_2f ist also eine Differenzenformel der Fehlerordnung $k = 2$ für $f'(x)$ (außerdem eine symmetrische zentrale Differenzenformel).

höhere Ableitungen: Addition der Taylor-Ausdrücke:

$$f(x_0+h) + f(x_0-h) = 2f(x_0) + h^2f''(x_0) + \frac{1}{12}h^4f^{\mathrm{IV}}(x_0) + ...$$
$$\rightarrow f''(x_0) = \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2} - \frac{1}{12}h^2f^{\mathrm{IV}}(x_0)$$

Balance von Diskretisierungs- und Rundungsfehlern: Beispiel: Formel $D, f = \frac{f(x_0+h)-f(x_0)}{h}$ für $f(x) = sin(x)$ und $x_0 = 1$; $f'(x) = cos(x), f'(1) = 0,54...$:

h	$\lvert D_1 f(x_0 = 1, h) - cos(x_0 = 1)\rvert$
10^{-1}	$0,043$
10^{-2}	$0,004$
10^{-3}	$0,0004$
10^{-4}	$0,00004$
10^{-5}	$0,000002$
10^{-6}	$0,000002$
10^{-7}	$0,0003$
10^{-9}	$0,04$
10^{-10}	$0,54$

Daraus folgt: $h = 10^{-12} \to 0,54$
Für abnehmende h wird der Fehler zunächst kleiner, steigt dann aber wieder an und bleibt für $h < 10^{-10} = \epsilon$ konstant, da $x_0 = 1 = 1 + h$ und damit $D_1(f(x_0 = 1, h < 10^{-10})) = 0$ Dilemma: Für große h ergibt sich ein Diskretisierungsfehler und für kleine ein Rundungsfehler.