Computergestüzte Methoden der exakten Naturwissenschaften

Prof. Dr. Roland Netz

29. Oktober 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 $M = max_{x \leq x_0 \leq \tilde{x}}(|f'(x_0)|)$. 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:

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