

## Kapitel 3

# Nullstellenberechnung von nichtlinearen Funktionen

In dieser Vorlesung wird nur die Nullstellenberechnung reeller Funktionen einer reellen Variablen  $f : \mathbb{R} \rightarrow \mathbb{R}$  betrachtet. Man nennt die Nullstellen von nichtlinearen Funktionen auch Wurzeln.

In der Praxis sind Nullstellenberechnungen mehrdimensionaler Funktionen von großer Bedeutung. Die Behandlung von Verfahren für solche Funktionen ist jedoch aus Zeitgründen nicht möglich.

### 3.1 Theorie

Man unterscheidet zwei Normalformen von nichtlinearen Gleichungen, die Nullstellengleichung

$$f_1(x) = 0 \tag{3.1}$$

und die Fixpunktgleichung

$$f_2(x) = x. \tag{3.2}$$

Eine Zahl  $\alpha \in \mathbb{R}$  wird Fixpunkt von  $f_2$  genannt, falls  $f_2(\alpha) = \alpha$ .

Jede Nullstellengleichung lässt äquivalent sich in eine Fixpunktgleichung überführen. Sei  $g(x)$  eine stetige Funktionen mit  $g(x) \neq 0$ , dann wird (3.1) mit  $g(x)$  multipliziert und auf beiden Seiten wird  $x$  addiert:

$$F_1(x) := g(x)f_1(x) + x = x.$$

Eine Nullstelle von  $f_1$  ist ein Fixpunkt von  $F_1$  und umgekehrt. Subtrahiert man in (3.2)  $x$ , so erhält man die Nullstellengleichung

$$F_2(x) := f_2(x) - x = 0.$$

Das heißt, ein Fixpunkt von  $f_2$  ist eine Nullstelle von  $F_2$  und umgekehrt.

**Bemerkung 3.1 Kondition des Wurzelfindens.** Sei  $f_1 \in C^1(a, b)$ , betrachte

$$f_1(x) - d = 0, \quad d \in \mathbb{R}.$$

Man kann zeigen, dass das Problem des Findens der Wurzel nur dann wohldefiniert ist, wenn  $f_1$  in einer Umgebung von  $d$  invertierbar ist. Sei  $\alpha = f_1^{-1}(d)$ ,  $f_1^{-1}$  – inverse Funktion (Umkehrfunktion). Mit der Differentiationsregel für die inverse Funktion erhält man

$$(f_1^{-1})'(d) = \frac{1}{f_1'(\alpha)}.$$

Dann ist das Wurzelfinden einer nichtlinearen Gleichung gut konditioniert, falls die Wurzel  $\alpha$  einfach ist, das heißt  $f_1'(\alpha) \neq 0$ , und  $f_1'(\alpha)$  hinreichend weit von Null entfernt ist. Ansonsten ist das Problem schlecht konditioniert.  $\square$

Numerische Verfahren zur Lösung nichtlinearer Gleichungen sind im allgemeinen iterativ. Beginnend mit einem Startwert  $x^{(0)}$  wird vom Verfahren eine Folge  $\{x^{(k)}\}_{k \geq 0}$  generiert, mit dem Ziel, dass

$$\lim_{k \rightarrow \infty} x^{(k)} = \alpha.$$

Man möchte nun ein Maß für die Geschwindigkeit der Konvergenz haben, womit man dann die Effizienz von Verfahren beschreiben kann.

**Definition 3.2** Eine Folge  $\{x^{(k)}\}$  konvergiert gegen  $\alpha$  mit Ordnung  $p \geq 1$ , falls eine Konstante  $C > 0$  existiert, so dass gilt

$$\frac{|x^{(k+1)} - \alpha|}{|x^{(k)} - \alpha|^p} \leq C \quad \forall k \geq k_0,$$

wobei  $k_0 \in \mathbb{N}$  ein geeigneter Index ist. Man sagt dann, dass die Methode von  $p$ -ter Ordnung ist.

Ist  $p = 1$ , so muss notwendigerweise  $C < 1$  sein, damit  $\{x^{(k)}\}$  gegen  $\alpha$  konvergiert. In diesem Fall wird  $C$  Konvergenzfaktor der Methode genannt.  $\square$

Ob ein Iterationsverfahren überhaupt konvergiert, kann auch vom Startwert  $x^{(0)}$  abhängen.

**Definition 3.3** Sei  $I$  die Menge aller zulässigen Startwerte. Gibt es eine Umgebung  $U(\alpha)$  der Wurzel  $\alpha$  (offene Menge, die  $\alpha$  enthält), so dass ein Iterationsverfahren für alle  $x^{(0)} \in U(\alpha)$  konvergiert, so nennt man das Verfahren lokal konvergent. Konvergiert das Verfahren für alle  $x^{(0)} \in I$ , dann wird es global konvergent genannt.  $\square$

Zur Lösung der Fixpunktgleichung (3.2) kann man das Iterationsverfahren

$$x^{(k+1)} = f_2(x^{(k)}), \quad k = 0, 1, \dots \quad (3.3)$$

verwenden. *MATLAB-Demo*

Zur Existenz eines Fixpunktes und zur Konvergenz der Iteration (3.3) gibt es folgenden berühmten Satz.

**Satz 3.4 Fixpunktsatz von Banach (1922).** Die Funktion  $f_2$  sei im abgeschlossenen Intervall  $I = [a, b] \subset \mathbb{R}$  kontraktiv, das heißt:

- $I$  wird durch  $f_2$  in sich abgebildet,  $f_2(I) \subseteq I$ ,
- $f_2$  ist Lipschitz-stetig mit einer Lipschitz-Konstanten  $L \in [0, 1)$ :

$$|f_2(x) - f_2(y)| \leq L|x - y| \quad \forall x, y \in I.$$

Dann besitzt  $f_2$  in  $I$  genau einen Fixpunkt  $\alpha$  und die nach der Iterationsvorschrift (3.3) berechnete Folge konvergiert gegen  $\alpha$  für jeden Startwert  $x^{(0)} \in I$ . Es gelten die Fehlerabschätzungen

$$\begin{aligned} |\alpha - x^{(k)}| &\leq \frac{L^k}{1-L} |x^{(1)} - x^{(0)}|, \\ |\alpha - x^{(k)}| &\leq \frac{L}{1-L} |x^{(k)} - x^{(k-1)}|. \end{aligned}$$

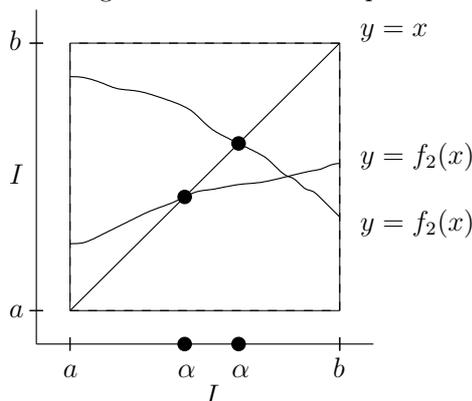
**Beweis:** HMI I, Literatur. ■

**Bemerkung 3.5**

1. Die Aussagen des Banachschen Fixpunktsatzes gelten allgemein in Banach-Räumen, also insbesondere auch für Funktionen  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $n \in \mathbb{N}$ .
2. Ist  $f_2$  in  $I$  stetig differenzierbar bis zum Rand,  $f_2 \in C^1([a, b])$ , so ist  $f_2$  Lipschitz-stetig und es gilt

$$L = \max_{x \in [a, b]} |f_2'(x)|.$$

3. Geometrische Deutung des Banachschen Fixpunktsatzes.



Sei  $f_2$  differenzierbar in  $[a, b]$  und sei  $L < 1$ . Der Betrag des Anstiegs von  $f_2$  ist kleiner als 1, siehe Bemerkung 2. Falls  $f_2$  die Gerade  $y = x$  einmal geschnitten hat, kann es keinen zweiten Schnittpunkt geben, weil  $f_2$  dann schneller ansteigen müsste als  $y = x$ . Das geht aber nicht, da  $y' = (x)' = 1$ . Dass überhaupt ein Schnittpunkt von  $f_2$  und  $y = x$  existiert, folgt aus

- $f_2$  bildet  $I$  in  $I$  ab,
- $f_2$  ist stetig,
- $I$  ist abgeschlossen.

Man kann leicht Beispiele dafür konstruieren, dass der Banachsche Fixpunktsatz nicht mehr gilt, wenn auch nur eine dieser drei Voraussetzungen nicht erfüllt ist.

4. Sei  $L > 0$ . Aus (3.3) und der Taylor-Entwicklung von  $f_2$  in  $\alpha$  folgt

$$\begin{aligned} x^{(k+1)} - \alpha &= f_2(x^{(k)}) - f_2(\alpha) = f_2(\alpha) + f_2'(\xi)(x^{(k)} - \alpha) - f_2(\alpha) \\ &= f_2'(\xi)(x^{(k)} - \alpha) \end{aligned}$$

mit  $\xi \in [a, b]$ . Mit Bemerkung 2 hat man

$$\frac{|x^{(k+1)} - \alpha|}{|x^{(k)} - \alpha|} = |f_2'(\xi)| \leq L.$$

Die Konvergenzordnung ist also linear und der Konvergenzfaktor ist  $L$ . Die Konvergenzgeschwindigkeit ist desto besser, je kleiner  $L$  ist. □

Numerische Verfahren, die auf dem Prinzip der Fixpunktiteration beruhen, werden in den Abschnitten 3.3 und 3.4 behandelt.

## 3.2 Das Bisektions-Verfahren

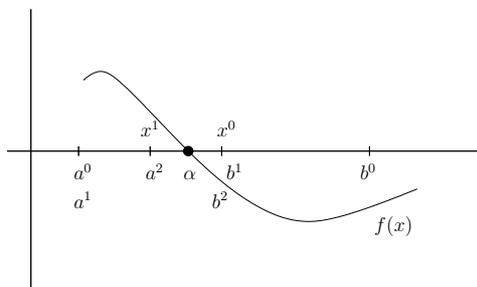
Bei diesem Verfahren handelt es sich um das einfachste und sicherste Verfahren zur Berechnung einer Nullstelle einer stetigen Funktion im Intervall  $I = [a, b]$ . Dieses Verfahren lässt sich jedoch nicht auf höhere Dimensionen erweitern. Es beruht auf dem Zwischenwertsatz: Jede stetige Funktion  $f : [a, b] \rightarrow \mathbb{R}$  nimmt jeden Funktionswert zwischen  $f(a)$  und  $f(b)$  an (falls  $f(a) \leq f(b)$ , ansonsten umgekehrt). Damit folgt aus  $f(a)f(b) < 0$ , dass  $f$  mindestens eine Nullstelle in  $[a, b]$  besitzt.

Beim Bisektions-Verfahren versucht man, ein genügend kleines Intervall zu finden, in dem eine Nullstelle von  $f$  liegt. Man muss zunächst Werte  $a$  und  $b$  finden, so dass  $f(a)$  und  $f(b)$  unterschiedliches Vorzeichen besitzen. Setze dann  $I_0 := [a, b]$ . Dann werden Teilintervalle  $I_k := [a^{(k)}, b^{(k)}]$ ,  $k \geq 0$ ,  $I_k \subset I_{k-1}$ ,  $k \geq 1$ , gesucht, so dass  $f(a^{(k)})f(b^{(k)}) < 0$ . Das geschieht mit dem folgenden Algorithmus:

### Algorithm 3.6 Bisektions-Verfahren.

- Setze  $a^{(0)} := a$ ,  $b^{(0)} := b$ ,  $x^{(0)} := \frac{a^{(0)} + b^{(0)}}{2}$ .
- Iteration: falls  $k \geq 0$ :
  - falls  $f(x^{(k)})f(a^{(k)}) < 0$ , setze  $a^{(k+1)} := a^{(k)}$ ,  $b^{(k+1)} := x^{(k)}$ ,
  - falls  $f(x^{(k)})f(b^{(k)}) < 0$ , setze  $a^{(k+1)} := x^{(k)}$ ,  $b^{(k+1)} := b^{(k)}$ ,
 anschließend setze  $x^{(k+1)} := \frac{a^{(k+1)} + b^{(k+1)}}{2}$ .
- Abbruch: Man bricht das Bisektions-Verfahren ab, falls
  - $f(x^{(k+1)}) = 0$ , das heißt, man hat eine Nullstelle gefunden,
  - oder  $|b^{(k+1)} - a^{(k+1)}| = |I_{k+1}| < \varepsilon$ , wobei  $\varepsilon > 0$  eine vorgegebene Toleranz ist, das heißt das Restintervall ist hinreichend klein. Nehme als Approximation an die Nullstelle den Mittelpunkt  $x^{(k+1)}$  von  $I_{k+1}$ .

*MATLAB-Demo* □



Nun wird die Konvergenzgeschwindigkeit des Bisektions-Verfahrens betrachtet. Es sind

$$|I_0| = b - a, \quad |I_k| = \frac{|I_0|}{2^k} = \frac{b - a}{2^k}, \quad k \geq 0.$$

Sowohl die  $k$ -te Iterierte  $x^{(k)}$  als auch die Lösung  $\alpha$  liegen im Intervall  $I_k$ . Da  $x^{(k)}$  der Mittelpunkt von  $I_k$  ist, gilt für den absoluten Fehler

$$\left| e^{(k)} \right| = \left| x^{(k)} - \alpha \right| \leq \frac{|I_k|}{2} = \frac{b - a}{2^{k+1}} \implies \lim_{k \rightarrow \infty} \left| e^{(k)} \right| = 0.$$

Das gilt für beliebige Startwerte  $a^{(0)}, b^{(0)}$  mit  $f(a^{(0)})f(b^{(0)}) < 0$ . Das Bisektions-Verfahren ist also global konvergent.

Sei  $\varepsilon > 0$  gegeben. Gesucht ist die Anzahl von Bisektions-Schritten, damit  $|x^{(m)} - \alpha| < \varepsilon$ . a  $x^{(k)}$  der Mittelpunkt von  $I_k$  ist, ist das sicher erfüllt falls

$$|I_m| < 2\varepsilon \iff \frac{b - a}{2^{m+1}} < \varepsilon \iff m > \frac{\log((b - a)/\varepsilon)}{\log(2)} - 1.$$

Man kann die Formel so interpretieren, als dass jede Bisektion die Genauigkeit um eine Dualstelle verbessert. Daraus folgt, dass man zur Verbesserung um eine Dezimalstelle  $\log_2(10) \approx 3.32$  Bisektionen benötigt. Insgesamt ist die Konvergenz der Bisektion zwar sicher, aber langsam.

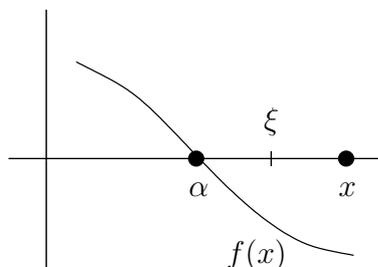
Die Konvergenz braucht nicht monoton zu sein, siehe Beispiele in der Literatur. Aus diesem Grunde passt das Bisektions-Verfahren nicht in den Rahmen der Definition 3.2 über die Konvergenzordnung. Man kann also nicht von einer Methode erster Ordnung im Sinne dieser Definition sprechen.

Die Bisektion eignet sich vor allem als Annäherungsverfahren an die Nullstelle, um einen Startwert für schnellere Iterationsverfahren zu finden, die nur lokal konvergent sind.

### 3.3 Das Sekanten-Verfahren und Varianten

Sei  $\alpha \in I = [a, b]$  eine Nullstelle von  $f \in C^1(I)$ . Sei  $x$  nahe genug an  $\alpha$ , o.B.d.A. mit  $x > \alpha$ . Dann erhält man mit Taylor-Entwicklung

$$f(x) = f(\alpha) + (x - \alpha) f'(\xi), \quad \xi \in (\alpha, x).$$



Diese Beziehung kann man in die Fixpunktgleichung

$$\alpha = x - \frac{f(x) - f(\alpha)}{f'(\xi)} = x - \frac{f(x)}{f'(\xi)} \quad (3.4)$$

umformen, falls  $f'(\xi) \neq 0$ . Diese Eigenschaft ist zum Beispiel in einer Umgebung von  $\alpha$  gesichert, falls  $\alpha$  eine einfache Nullstelle von  $f$  ist. Die Verfahren, die in diesem Abschnitt vorgestellt werden, approximieren  $f'(\xi)$  in (3.4) unter Benutzung von Werten von  $f$ .

Die Grundidee der Approximation einer Ableitung besteht in der Verwendung einer finiten Differenz:

$$f'(x) \approx \frac{f(x_1) - f(x_2)}{x_1 - x_2},$$

wobei  $x_1, x_2$  sich in einer Umgebung von  $x$  befinden. Sind  $x_1$  oder  $x_2$  relativ weit voneinander oder von  $x$  entfernt, wird die Approximation relativ ungenau sein. Liegen andererseits  $x_1, x_2$  sehr dicht beieinander, kann es sowohl im Zähler als auch im Nenner der finiten Differenz Auslöschung geben, was wiederum zu ungenauen Ergebnissen führen kann.

Beim Sekanten-Verfahren verwendet man

$$f'(\xi) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}, \quad k \geq 0,$$

wobei  $x^{(k-1)}, x^{(k)}$  die beiden letzten Iterierten sind, siehe Abbildung 3.1.

Man benötigt also zunächst zwei Startwerte  $x^{(-1)}, x^{(0)}$ . Sind diese gegeben, so lautet das Sekanten-Verfahren

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)}), \quad k \geq 0. \quad (3.5)$$

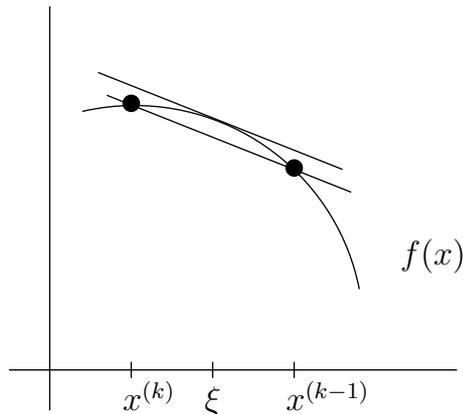


Abbildung 3.1: Approximation einer Tangente durch eine Sekante.

*MATLAB-Demo*

Zur die Konvergenz dieses Verfahrens gibt es folgende Aussage:

**Satz 3.7** Seien  $U(\alpha)$  eine geeignete Umgebung von  $\alpha$ ,  $f \in C^2(U(\alpha))$ ,  $f'(\alpha) \neq 0$  und  $f''(\alpha) \neq 0$ . Falls die Startwerte  $x^{(-1)}, x^{(0)} \in U(\alpha)$  hinreichend nahe an  $\alpha$  gewählt werden, konvergiert die mit (3.5) berechnete Folge  $\{x^{(k)}\}$  gegen  $\alpha$  und die Konvergenzordnung ist  $p = (1 + \sqrt{5})/2 \approx 1.62$ .

**Beweis:** Literatur, zum Beispiel [SK06]. ■

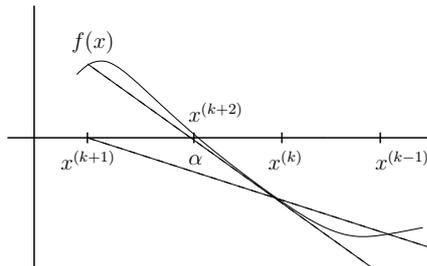


Abbildung 3.2: Graphische Illustration zum Sekantenverfahren.

**Bemerkung 3.8**

1. In der Praxis bricht man das Verfahren im allgemeinen ab, falls  $|f(x^{(m)})| < \varepsilon$  oder  $|x^{(m)} - x^{(m-1)}| < \varepsilon$  für eine vorgegebene Toleranz  $\varepsilon > 0$ .
2. Das Sekanten-Verfahren ist lokal konvergent, aber im allgemeinen nicht global.
3. Man benötigt neben der Berechnung von  $f(x^{(k)})$  im  $k$ -ten Iterationsschritt keine zusätzliche Funktionswertberechnung, da man  $f(x^{(k-1)})$  noch aus dem vorangegangenen Iterationsschritt kennt.
4.  $x^{(k+1)}$  liegt nicht unbedingt im Intervall  $[x^{(k)}, x^{(k-1)}]$ , siehe Abbildung 3.2. Das geschieht, wenn  $f(x^{(k)})$  und  $f(x^{(k-1)})$  gleiches Vorzeichen besitzen.
5. Verallgemeinerungen auf komplexe Nullstellen und Funktionen in Banach-Räumen sind möglich.
6. Eine weitere Verallgemeinerung besteht darin, statt einer Sekante eine Parabel durch drei Punkte zu nehmen – Verfahren von Muller, Konvergenzordnung 1.84.

□

Falls  $f(x^{(k)})$  und  $f(x^{(k-1)})$  gleiches Vorzeichen besitzen und fast gleichgroß sind, kann  $x^{(k+1)}$  sehr weit entfernt von  $\alpha$  liegen. Es kann sein, dass  $f$  für  $x^{(k+1)}$  gar nicht definiert ist oder sehr große Werte annimmt, die zu einem Überlauf führen.

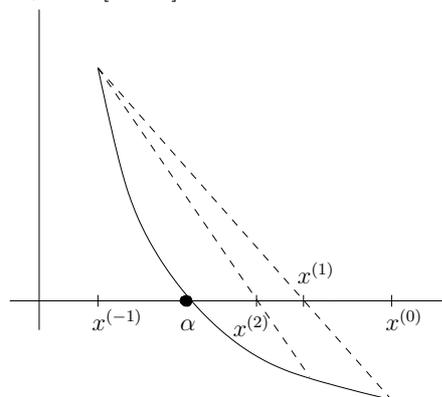
Ein Verfahren, bei denen alle Iterierten  $x^{(k)}$  im Intervall  $[x^{(-1)}, x^{(0)}]$  liegen, und welches ebenfalls Sekanten benutzt, ist die Regula falsi (Regel des falschen Ansatzes). Anstatt die Sekante immer durch die letzten beiden Iterierten zu nehmen, verwendet man die Sekante durch  $(x^{(k)}, f(x^{(k)}))$  und  $(x^{(k')}, f(x^{(k')}))$ , wobei  $k'$  der größte Index kleiner als  $k$  ist, für welchen  $f(x^{(k)}) f(x^{(k')}) < 0$  gilt. Für die Regula falsi benötigt man zwei Startwerte  $x^{(-1)}, x^{(0)}$  mit  $f(x^{(0)}) f(x^{(-1)}) < 0$ . Die Iterationsvorschrift lautet

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k')}}{f(x^{(k)}) - f(x^{(k')})} f(x^{(k)}), \quad k \geq 0. \quad (3.6)$$

*MATLAB-Demo*

**Satz 3.9** Sei  $f \in C([x^{(-1)}, x^{(0)}])$ . Dann gilt für die mit (3.6) erzeugte Folge  $\{x^{(k)}\}$ , dass  $x^{(k)} \in [x^{(-1)}, x^{(0)}]$ ,  $k \geq 1$  und  $\lim_{k \rightarrow \infty} x^{(k)} = \alpha$ . Sind  $f \in C^2([x^{(-1)}, x^{(0)}])$ ,  $\alpha$  eine einfache Nullstelle ( $f'(\alpha) \neq 0$ ) und  $f''(\alpha) \neq 0$ , dann ist die Konvergenzordnung der Regula falsi gleich Eins.

**Beweis:** Literatur, z.B. [SK06]. ■



In diesem Fall ist  $k' = 1$  für alle  $k$ . Das ist der schlechteste Fall, der dafür sorgt, dass im allgemeinen die Konvergenzordnung nicht größer als Eins ist.

**Bemerkung 3.10**

1. Die Regula falsi ist global konvergent. Da sie im allgemeinen jedoch langsam ist, eignet sie sich für allem zur Beschaffung von Startwerten für lokal konvergente, schnellere Verfahren.
2. Gemessen an Funktionswertberechnungen, ist der Aufwand von Regula falsi und Sekanten-Verfahren gleich.
3. Es gibt Varianten der Regula falsi, die eine bessere Konvergenzordnung besitzen, indem sie vermeiden, dass  $k'$  fixiert wird: Illinois-Algorithmus  $p = \sqrt[3]{3} \approx 1.442$ ; Pegasus-Verfahren,  $p \approx 1.642$ .
4. Des weiteren gibt es Verfahren, die Bisektion (Sicherheit) und Sekanten-Verfahren (Schnelligkeit) kombinieren, zum Beispiel das Verfahren von Dekker und Brent (1973), siehe [QSS04].

□

### 3.4 Das Newton–Verfahren

- Newton 1669,
- veröffentlicht von Wallis 1685,
- in der heutigen Form von Raphson 1697, deshalb auch manchmal Newton–Raphson–Verfahren.

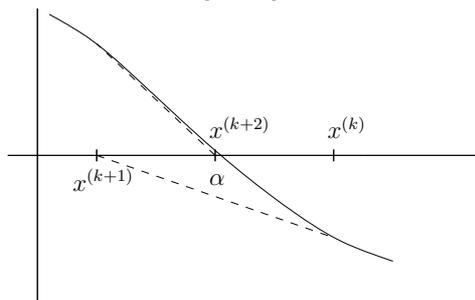
Sei  $f : [a, b] \rightarrow \mathbb{R}$  stetig differenzierbar. Das Newton–Verfahren verwendet zur Approximation von  $f'(\xi)$  in (3.4) den Wert der Ableitung von  $f$  an der Stelle  $x^{(k)}$ . Sei also  $x^{(0)}$  gegeben, dann lautet das Newton–Verfahren

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k \geq 0. \quad (3.7)$$

*MATLAB–Demo*

**Satz 3.11** Seien  $f \in C^3([a, b])$ ,  $\alpha \in [a, b]$  und  $f'(\alpha) \neq 0$  (einfache Nullstelle). Dann existiert eine Umgebung  $U(\alpha)$ , so dass für jeden Startwert  $x^{(0)} \in U(\alpha)$  die mit (3.7) berechnete Folge  $\{x^{(k)}\}$  gegen  $\alpha$  konvergiert. Die Konvergenzordnung ist mindestens 2.

**Beweis:** Literatur, zum Beispiel [SK06]. ■



In  $(x^{(k)}, f(x^{(k)}))$  wird die Tangente angelegt. Der Schnittpunkt der Tangente mit der  $x$ -Achse ist  $x^{(k+1)}$ .

**Bemerkung 3.12**

1. Man bricht die Iteration ab, falls  $|x^{(m)} - x^{(m-1)}| < \varepsilon$  oder  $|f(x^{(m)})| < \varepsilon$  für eine vorgegebene Toleranz  $\varepsilon > 0$ .
2. Das Newton–Verfahren ist lokal konvergent, im allgemeinen jedoch nicht global.
3. Bei einer  $l$ -fachen Nullstelle,  $l \geq 2$ , konvergiert das Newton–Verfahren (3.7) linear mit dem Konvergenzfaktor  $C = (l - 1)/l$ . Ist die Vielfachheit der Nullstelle bekannt, so konvergiert das modifizierte Newton–Verfahren

$$x^{(k+1)} = x^{(k)} - l \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k \geq 0,$$

quadratisch.

4. Das Newton–Verfahren benötigt zwei Funktionsaufrufe pro Iteration,  $f(x^{(k)})$  und  $f'(x^{(k)})$ . Diesbezüglich ist es doppelt so teuer wie das Sekanten–Verfahren. Zwei Schritte des Sekanten–Verfahrens haben die Konvergenzordnung  $(1 + \sqrt{5})^2 / 4 \approx 2.62$ . Damit konvergiert, gemessen am Aufwand, das Sekanten–Verfahren schneller.

5. *Optimale Fixpunktiteration.* Die Nullstellengleichung  $f(x) = 0$  lässt sich äquivalent in die Fixpunktgleichung

$$x = x + g(x) f(x), \quad g(x) \neq 0,$$

umformen. Der Banachsche Fixpunktsatz 3.4 besagt, dass die Fixpunktiteration

$$x^{(k+1)} = x^{(k)} + g(x^{(k)}) f(x^{(k)})$$

desto schneller konvergiert, je kleiner die Lipschitz-Konstante der rechten Seite der Fixpunktgleichung ist. Seien  $f(x), g(x)$  in  $U(\alpha)$  stetig differenzierbar, dann folgt

$$L = \max_{x \in U(\alpha)} |(x + g(x) f(x))'| = \max_{x \in U(\alpha)} |1 + g'(x) f(x) + g(x) f'(x)|.$$

Man kann hoffen, dass  $L$  in  $\overline{U(\alpha)}$  klein ist, wenn für  $x = \alpha$  der Ausdruck im Betrag verschwindet. Da  $f(\alpha) = 0$ , folgt

$$1 + g(\alpha) f'(\alpha) = 0 \quad \implies \quad g(\alpha) = -\frac{1}{f'(\alpha)}.$$

Eine Funktion, die diese Bedingung erfüllt, ist  $g(x) = -1/f'(x)$ , womit man das Newton-Verfahren erhält.

Man erhält also das Newton-Verfahren, indem man die Funktion  $g(x)$  in der Fixpunktgleichung so wählt, dass man auf eine möglichst kleine Lipschitz-Konstante in  $\overline{U(\alpha)}$  hoffen kann.

6. Ist  $f \in C^2(\overline{U(\alpha)})$ , dann erhält man aus dem Banachschen Fixpunktsatz folgende hinreichende Bedingung für die Konvergenz des Newton-Verfahrens:

$$\begin{aligned} 1 > L &= \max_{x \in \overline{U(\alpha)}} \left| \left( x - \frac{f(x)}{f'(x)} \right)' \right| = \max_{x \in \overline{U(\alpha)}} \left| 1 - \frac{(f'(x))^2 - f(x) f''(x)}{(f'(x))^2} \right| \\ &= \max_{x \in \overline{U(\alpha)}} \left| \frac{f(x) f''(x)}{(f'(x))^2} \right|. \end{aligned}$$

7. Die Übertragung des Newton-Verfahrens auf komplexe Nullstellen und auf Banach-Räume ist möglich. □

## 3.5 Nullstellen von Polynomen

Dieser Abschnitt stellt ein Verfahren vor, mit dem man alle Nullstellen eines Polynoms

$$p_n(x) = \sum_{i=0}^n a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0, \quad (3.8)$$

$a_i \in \mathbb{R}, i = 0, \dots, n$ , berechnen kann.

Der Fundamentalsatz der Algebra besagt, dass  $p_n(x)$  genau  $n$  Nullstellen in  $\mathbb{C}$  besitzt, wobei mehrfache Nullstellen ihrer Vielfachheit gemäß gezählt werden. Des weiteren ist bekannt:

- $n \in \{1, 2\}$  – es gibt einfache Berechnungsvorschriften für die Nullstellen,
- $n \in \{3, 4\}$  – es gibt relativ komplizierte Berechnungsvorschriften für die Nullstellen,

- $n \geq 5$  – es gibt im allgemeinen keine expliziten Berechnungsvorschriften für die Nullstellen.

Das heißt, für Polynome mit  $n \geq 3$  ist die Nutzung numerischer Verfahren zu empfehlen, für  $n \geq 5$  ist sie im allgemeinen notwendig.

Bevor wir zur Nullstellenberechnung kommen, wollen wir erst einmal ein Schema kennenlernen, mit dessen Hilfe man Funktionswerte eines Polynoms schnell berechnen kann. Nutzt man den naivsten Weg, jeden Summanden von (3.8) einzeln zu berechnen, so hat man für den  $i$ -ten Summanden  $i$  Multiplikationen und zum Schluss  $n$  Additionen. Damit ist die Anzahl der Flops

$$\sum_{i=0}^n i + n = \frac{n}{2}(n+1) + n = \frac{n^2}{2} + \frac{3}{2}n.$$

Günstiger ist bereits, wenn man die Potenzen von  $x$  zwischendurch speichert:

```

p := a0 + a1x
t := x
for i = 2 : n
    t := tx
    p := p + ait
end

```

Dieser Weg benötigt  $n$  Additionen und  $(2n - 1)$  Multiplikationen, also  $3n - 1$  Flops.

Die Darstellung (3.8) ist nicht die einzig mögliche für ein Polynom. Äquivalent dazu ist

$$p_n(x) = a_0 + x(a_1 + x(a_2 + x(\dots + x(a_{n-1} + a_n x) \dots))). \quad (3.9)$$

Nutzt man (3.9), so kann man  $p_n(x)$  mit  $n$  Additionen und  $n$  Multiplikationen, also  $2n$  Flops berechnen. Man nennt (3.9) auch eingebettete Multiplikation und (3.9) ist die Basis des sogenannten Horner-Schemas (synthetische Division) zur Berechnung von  $p_n(z)$ :

```

bn := an
for i = n - 1 : -1 : 0
    bi := ai + bi+1z
end

```

**Beispiel 3.13**  $n = 3$ :

$$p_3(z) = a_0 + z \left( a_1 + z \left( a_2 + \underbrace{a_3 z}_{b_3} \right) \right) \underbrace{\hspace{1.5cm}}_{b_2} \underbrace{\hspace{2.5cm}}_{b_1} \underbrace{\hspace{3.5cm}}_{b_0}.$$

□

Dieses Verfahren, veröffentlicht von Horner (1818), findet man schon bei Newton über 100 Jahre früher. Damals hat man das Verfahren handschriftlich durchgeführt und dabei folgendes Schema entwickelt:

$$\begin{array}{ccccccc} p_n & a_n & a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ z & - & b_n z & b_{n-1} z & \dots & b_2 z & b_1 z \\ \hline & & b_n & b_{n-1} & b_{n-2} & \dots & b_1 & b_0 \end{array}$$

Das ist das sogenannte Horner-Schema.

**Beispiel 3.14** Berechne  $p_3(z) = -2z^3 + 20z^2 - 2z - 13$  an der Stelle  $z = 3$ :

$$\begin{array}{r} p_3 \quad -2 \quad 20 \quad -2 \quad -13 \\ 3 \\ \hline \quad -2 \quad 14 \quad 40 \quad 107 \end{array}$$

Es ist  $p_3(3) = 107$ . □

Man kann das Polynom  $p_n(x)$  eindeutig zerlegen in

$$p_n(x) = p_0 + (x - z)p_{n-1}(x),$$

wobei  $p_0$  eine Konstante ist und  $p_{n-1}(x)$  ein Polynom vom Grad  $n - 1$  (Polynomdivision). Setzt man  $x = z$ , so folgt  $p_n(z) = b_0 = p_0$ , also folgt

$$p_{n-1}(x) = \frac{p_n(x) - b_0}{x - z}.$$

Man findet durch Nachrechnen *Übungsaufgabe*

$$p_{n-1}(x) = b_1 + b_2x + \dots + b_nx^{n-1} = \sum_{i=1}^n b_i x^{i-1} =: q_{n-1}(x).$$

Einsetzen führt zu der Darstellung

$$p_n(x) = b_0 + (x - z)q_{n-1}(x). \tag{3.10}$$

Ist  $z$  eine Nullstelle von  $p_n(x)$ , dann ist  $b_0 = 0$  und man hat

$$p_n(x) = (x - z)q_{n-1}(x).$$

Die restlichen  $(n - 1)$  Nullstellen von  $p_n(x)$  sind nun gerade die Nullstellen von  $q_{n-1}(x)$ . Man hat die Nullstelle  $z$  abgespalten. Diesen Vorgang nennt man Deflation. Auf dieser Basis kann man folgende allgemeine Strategie zur Berechnung aller Nullstellen von  $p_n(x)$  angeben:

1. finde eine Nullstelle  $z$  von  $p_n(x)$  mit einem geeigneten Verfahren,
2. berechne die Koeffizienten von  $q_{n-1}(x)$  mit dem Horner-Schema,
3. setze  $p_{n-1}(x) := q_{n-1}(x)$ ,  $n := n - 1$ , gehe zu 1.

Beim ersten Punkt kann man das Newton-Verfahren verwenden. Dazu benötigt man  $p_n(z)$  und  $p'_n(z)$  für die gegenwärtige Iterierte  $z$ . Aus (3.10) folgt

$$p'_n(x) = q_{n-1}(x) + xq'_{n-1}(x) - zq'_{n-1}(x),$$

also

$$p'_n(z) = q_{n-1}(z).$$

Diesen Wert kann man gemeinsam mit  $p_n(z)$  mit dem erweiterten Horner-Schema berechnen:

$$\begin{array}{r} p_n \quad a_n \quad a_{n-1} \quad a_{n-2} \quad \dots \quad a_1 \quad a_0 \\ z \quad - \quad b_n z \quad b_{n-1} z \quad \dots \quad b_2 z \quad b_1 z \\ \hline p_{n-1} \quad b_n \quad b_{n-1} \quad b_{n-2} \quad \dots \quad b_1 \quad b_0 = p_n(z) \\ z \quad - \quad c_n z \quad c_{n-1} z \quad \dots \quad c_2 z \\ \hline c_n \quad c_{n-1} \quad c_{n-2} \quad \dots \quad c_1 = p'_n(z) \end{array}$$

**Beispiel 3.15** Berechne  $p_4(z)$ ,  $p_4'(z)$  für

$$p_4(x) = 3x^4 - 5x^2 + 26x - 17, \quad z = 2.$$

$p_4$	3	0	-5	26	-17
	2	-	6	12	14
					80
$p_3$	3	6	7	40	63
	2	-	6	24	62
	3	12	31	102	

Damit sind  $63 = p_4(2)$  und  $p_4'(2) = 102$ . □

Dieses Vorgehen kann man erweitern, um alle Ableitungen von  $p_n(x)$  an der Stelle  $z$  zu berechnen, vollständiges Horner-Schema, siehe Literatur.

**Bemerkung 3.16**

1. Die Verbindung von Newton-Verfahren und Horner-Schema zur Berechnung einer Nullstelle von  $p_n(x)$  nennt man Newton-Horner-Verfahren. Ein Iterationsschritt benötigt  $4n$  Flops ( $p_n(z) : 2n, q_{n-1}(z) : 2(n-1)$ , eine Division, eine Subtraktion).
2. Beginnt man mit einer reellen Startnäherung, so erhält man beim Newton-Horner-Verfahren nur reelle Iterierte. Komplexe Nullstellen kann man so nicht finden. Sowohl das Newton-Verfahren als auch das Horner-Schema besitzen für komplexe Zahlen die gleiche Gestalt wie für reelle Zahlen. Verwendet man eine komplexe Startnäherung, so erhält man im allgemeinen komplexe Iterierte. Dazu benötigt man Computerarithmetik für komplexe Zahlen. Diese wird in MATLAB automatisch erledigt, falls komplexe Eingangsdaten vorliegen. *MATLAB-Demo von Newton-Verfahren im Komplexen*
3. Eine fortgesetzte Deflation erhöht den Einfluss von Rundungsfehlern. Um diesen gering zu halten, sollte man zwei Dinge beachten:
  - Berechne zuerst die betragsmäßig kleinen Nullstellen, weil diese am anfälligsten gegen Rundungsfehler sind.
  - Hat man eine Approximation für eine Nullstelle mit dem Newton-Horner-Verfahren gefunden, so nehme man diese Approximation als Startnäherung für das Newton-Verfahren angewandt auf das Originalpolynom  $p_n(x)$ : Newton-Horner-Verfahren mit Verfeinerung.
4. In MATLAB erhält man alle Nullstellen eines Polynoms mit dem Befehl `roots`.

□

### 3.6 Abbruchkriterien für iterative Verfahren

Ein wesentliches Problem in der Praxis sind Kriterien, mit denen man die Iteration abbricht. Eine erste Möglichkeit ist die Kontrolle des Residuums:

Sei  $\varepsilon > 0$  vorgegeben, dann breche die Iteration ab, falls  $|f(x^{(k)})| < \varepsilon$ .

Es gibt jedoch Situationen, in denen dieses Kriterium entweder zu stark oder zu optimistisch ist, siehe Abbildung 3.3

Aus diesem Grunde ist es oft besser, die Änderung der Korrektur zu betrachten:

Sei  $\varepsilon > 0$  vorgegeben, dann breche die Iteration ab, falls  $|x^{(k+1)} - x^{(k)}| < \varepsilon$ .

Darüber hinaus sollte immer eine maximale Anzahl von Iterationen als Abbruchkriterium vorgegeben sein.

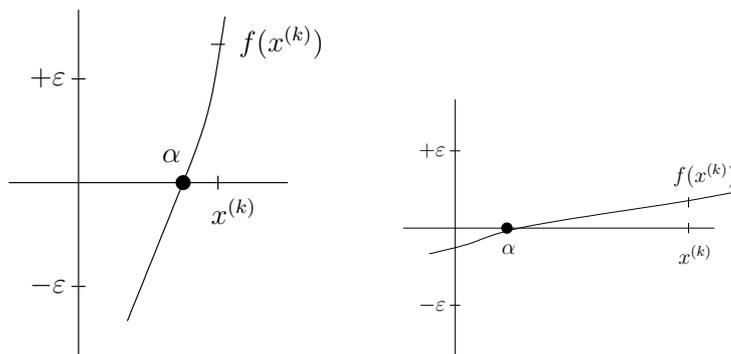


Abbildung 3.3: Links: Residuumkriterium zu stark, da  $f'(\alpha)$  sehr groß ist; rechts: Residuumkriterium zu optimistisch, da  $f'(\alpha) \approx 0$ .