Kapitel III

Nichtlineare Optimierung ohne Restriktionen

Einleitung

Die im 17. Jahrhundert durch Newton und Leibniz begründete Infinitesimalrechnung liefert das theoretische Fundament der Nichtlinearen Optimierung. Mit entsprechenden analytischen Hilfsmitteln der Analysis lassen sich z.B. Minima von reellen Funktionen $f : \boldsymbol{R}^n \to \boldsymbol{R}$ charakterisieren. Die konkrete Bestimmung der Minima, also die Berechnung von reellen Vektoren, die die charakterisierenden Bedingungen erfüllen, ist in den seltensten Fällen explizit mit theoretischen Mitteln möglich.

In der Einführung zur Mathematischen Optimierung werden nur nichtlineare Optimierungsprobleme ohne explizite Restriktionen behandelt. Bei diesen Aufgaben muß das Minimum einer gegebenen Funktion $f : \boldsymbol{R}^n \to \boldsymbol{R}$ bestimmt werden, d.h. es soll eine Lösung der Aufgabe

$$\min_{x \in \boldsymbol{R}^n} f(x) =: f(\hat{x})$$

gefunden werden. So benutzt man zum Beispiel in der Regressionsrechnung Optimierungsmodelle zur Fixierung unbekannter Parameter.

Regressionsrechnung

Zur mathematischen Formulierung von Gesetzmäßigkeiten, die ein technisches oder physikalisches Phänomen beschreiben, wird häufig eine Hypothese über den möglichen funktionalen Zusammenhang bekannter und beobachtbarer Variablen formuliert, die noch etliche freie Parameter enthält, etwa

$$z = h(x, \lambda).$$

Hierbei ist $x \in \boldsymbol{R}^n$ ein Vektor, der die unabhängigen Variablen eines Experiments enthält, $\lambda \in \boldsymbol{R}^p$ ein Vektor unbekannter Parameter, und z eine Variable, deren Abhängigkeit modelliert werden soll.

Nach Auswertung von m Experimenten kennt man m Paare von Variablenwerten z_i, x_i, die unter Berücksichtigung möglicher Beobachtungsfehler ϵ_i in etwa die funktionale

Abhängigkeit erfüllen sollten, d.h. bei passenden Parametern muß

$$z_i = h(x_i, \lambda) + \epsilon_i$$

für alle $i \in \{1, 2, \ldots, n\}$ gelten. Um möglichst kleine Abweichungen ϵ_i zu erhalten, die man als Beobachtungsfehler deuten kann, versucht man die unbekannten Parameter λ optimal anzupassen, indem man entsprechende Optimierungsaufgaben löst, z.B.

$$\min \left\{ \sum_{i=1}^{m} (z_i - h(x_i, \lambda))^2 \mid \lambda \in \boldsymbol{R}^p \right\}$$

oder

$$\min \left\{ \sum_{i=1}^{m} w_i (z_i - h(x_i, \lambda))^r \mid \lambda \in \boldsymbol{R}^p \right\},$$

wobei $r \geq 1$ gerade, $w > 0$. $w > 0$ ist komponentenweise zu verstehen, wobei 0 den 0-Vektor entsprechender Dimension bezeichnet. Analog benutzen wir diese Konvention auch für andere Vektoren mit identischen Komponenten, insbesondere für den häufig auftretenden 1-Vektor.

1 Minimierung in einer Variablen

Die Lösung nichtlinearer Optimierungsaufgaben im eindimensionalen Fall tritt als Teilproblem in den meisten Verfahren zur Lösung nichtlinearer Optimierungsaufgaben in höheren Dimensionen auf. Zu gegebenem abgeschlossenen Intervall $I \subseteq \boldsymbol{R}$ und gegebener Funktion $f : I \to \boldsymbol{R}$ wird ein $\hat{x} \in I$ mit

$$f(\hat{x}) = \min_{x \in I} f(x) \tag{1.1}$$

gesucht. Um ein solches Minimum mit Hilfe eines Verfahrens effektiv bestimmen zu können, müssen einschränkende Annahmen über die Funktion gemacht werden. Je nach Eigenschaften der Funktion ist dann ein passender Algorithmus zu wählen.

Unimodale Funktionen

Definition (Unimodalität): f heißt *unimodal*, falls für ein $\hat{x} \in I := [a, b]$ gilt:
f streng monoton fallend auf $[a, \hat{x}]$, f streng monoton wachsend auf $[\hat{x}, b]$.

Intervallreduktion

Unimodale Funktionen erlauben nach Auswertung der Funktion an zwei Stellen $x < y$ im Intervall die Reduktion des Intervalls. Unter gewissen Annahmen liefert die rekursive Anwendung dieser Beobachtung einen festgelegten Algorithmus:

1. Im ersten Schritt zwei Auswertungen, in den weiteren Schritten eine weitere Auswertung von f im Restintervall zur Reduktion verwenden,

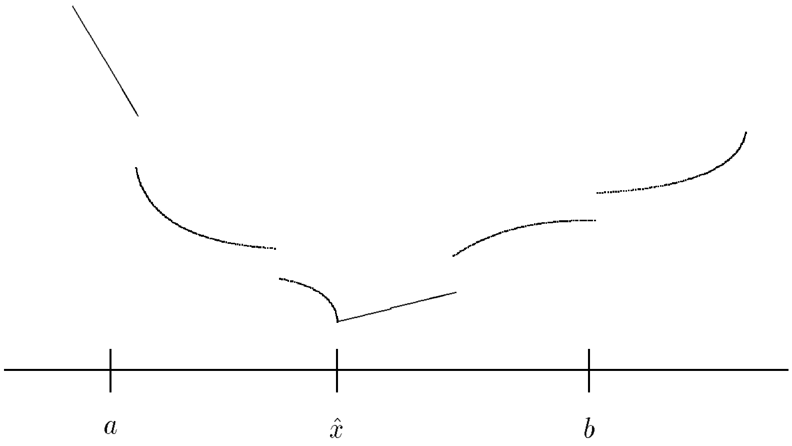

Abbildung III.1: Unimodale Funktionen

2. x und y stets symmetrisch im Restintervall wählen,

3. Reduktionsfaktor σ sei konstant, $\frac{1}{2} < \sigma < 1$.

Durch die symmetrische Wahl von x, y im Restintervall wird der Reduktionsfaktor unabhängig von der Funktion f. Verlangt man einen konstanten Reduktionsfaktor in allen Iterationen, so ist dieser eindeutig bestimmt. Um diesen von f unabhängigen Faktor zu berechnen, genügt es oBdA eine streng monoton wachsende Funktion $f : [0,1] \to \boldsymbol{R}$ zu betrachten. Im ersten Intervall sei oBdA $0 < x < y < 1$, und daher $y = \sigma$, $x = 1 - \sigma$.

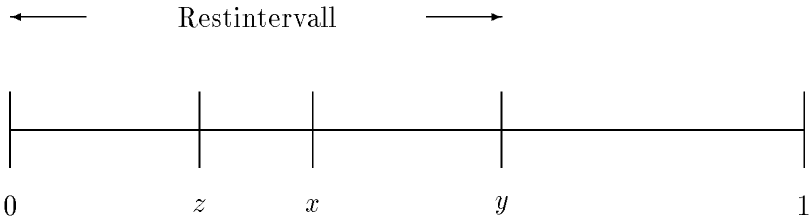

Abbildung III.2: Intervallreduktion bei unimodalen Funktionen

Im ersten Schritt wird das Intervall auf das Restintervall $[0, y]$ reduziert. In diesem Restintervall wird dann z symmetrisch zu x gewählt. Je nach Wahl von y wird $z < x$ oder $x \leq z$.

1. Fall: $\sigma < \frac{2}{3}$. Dann gilt $x = 1 - \sigma > \frac{1}{3}$ und daher $z = y - x = \sigma - (1 - \sigma) < \frac{1}{3} < x$. Konstanz der Reduktion bedeutet nun $\frac{x}{y} = \frac{y}{1}$, d.h. $0 = y^2 - x = \sigma^2 + \sigma - 1$. Die Lösung σ dieser quadratischen Gleichung mit $\frac{1}{2} < \sigma < 1$ ist $\sigma = \frac{1}{2}(\sqrt{5} - 1) = 0,618\ldots$.

2. Fall: $\sigma \geq \frac{2}{3}$. Analog zum ersten Fall folgt: $z = \sigma - (1 - \sigma) \geq \frac{1}{3} \geq x$. Also muß zur Konstanz der Reduktion $\frac{z}{y} = \frac{y}{1}$, d.h. $0 = y^2 - z = \sigma^2 - 2\sigma + 1$ gelten. Diese quadratische Gleichung besitzt offenbar keine Lösung σ mit $\frac{1}{2} < \sigma < 1$.

Der im ersten Fall gefundene Reduktionsfaktor $\sigma = 0,618\ldots$ ist also unter den getroffenen Annahmen der einzig mögliche und legt den Algorithmus fest.

Algorithmus 1.1 (Goldener Schnitt)

1. **Initialisierung:**
   $i := 0$; $a_0 := a$; $b_0 := b$;
   $x_i := a + (1-\sigma)(b_i - a_i)$; $y_i := a + \sigma(b_i - a_i)$;
   $fx := f(x_i)$; $fy := f(y_i)$.

2. **Iteration:**
   Wenn $fx < fy$, dann

   $a_{i+1} := a_i$; $b_{i+1} := y_i$;
   $x_{i+1} := a_{i+1} + (1-\sigma)(b_{i+1} - a_{i+1})$; $y_{i+1} := x_i$;
   $fy := fx$; $fx := f(x_{i+1})$;

   sonst

   $a_{i+1} := x_i$; $b_{i+1} := b_i$;
   $x_{i+1} := y_i$; $y_{i+1} := b_{i+1} - (1-\sigma)(b_{i+1} - a_{i+1})$;
   $fx := fy$; $fy := f(y_{i+1})$;

   $i := i + 1$;

3. **Abbruch:**
   Wenn $\frac{1}{2}(b_i - a_i) > \epsilon$, dann gehe nach 2.
   $\tilde{x} := \frac{1}{2}(a_i + b_i)$; Stop.

Im obigem Algorithmus wird ein Funktionswert zuviel berechnet; in der Praxis kann man die tatsächliche Berechnung des neuen fx bzw. fy zurückstellen, bis der Wert für den Vergleich benötigt wird.

Mögliche Abbruchkriterien

Wie oben beschrieben bricht der Algorithmus ab, sobald das Minimum bis auf eine vorgegebene Genauigkeit ϵ lokalisiert ist. Da nach $n+1$ Funktionsauswertungen die Länge des Restintervalls auf $\sigma^n(b-a)$ reduziert ist, kann man n in Abhängigkeit von ϵ a priori abschätzen.

Ist f auf dem Restintervall eine quadratische Funktion, so kann man den Wert des Minimums auf dem Restintervall nach unten durch

$$f(\hat{x}) \geq f_{\min} - \sigma(f_{\max} - f_{\min})$$

abschätzen. Liegt die Abweichung innerhalb einer vorgegebenen Genauigkeit, etwa wenn $f_{\max} - f_{\min} \leq \delta$ erfüllt ist, so kann man das Verfahren abbrechen.

Wenn man bereit ist, auf die Konstanz der Reduktion zu verzichten, kann man sich fragen, ob es eine Modifikation des Verfahrens gibt, das bei gleicher Anzahl von Funktionsauswertungen ein kleineres Restintervall liefert. Wir wollen mit L_n die maximale Länge eines mittels n Auswertungen in L_n auf $L_1 = 1$ reduzierbaren Intervalls bezeichnen.

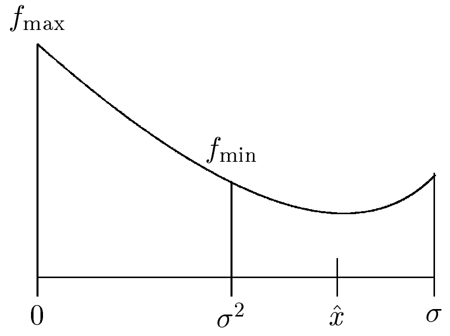

Abbildung III.3: Abbruchkriterien

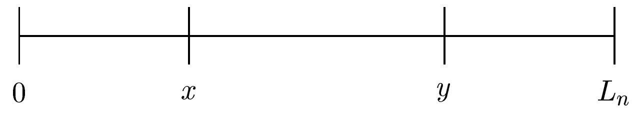

Abbildung III.4: Maximale Intervallreduktion

Durch jede der ersten beiden Stützstellen $x < y$ in L_n, etwa durch x, wird dieses Intervall in zwei Teilintervalle zerlegt. Das linke enthält höchstens $n-2$, das rechte höchstens $n-1$ der n Stützstellen. Entsprechend sind die Längen dieser Teilintervalle höchstens L_{n-2} bzw. L_{n-1}. Also gilt

$$L_n \leq L_{n-1} + L_{n-2}$$

und $L_0 = L_1 = 1$, da mindestens 2 Stützstellen zur Reduktion notwendig sind. Nach dieser Abschätzung erhalten wir die bestmögliche Reduktion, falls eine Lösung die Ungleichungen als Gleichungen erfüllt.

Fibonacci-Zahlen

Die entsprechende Rekursionsformel $F_0 := F_1 := 1, F_n := F_{n-1} + F_{n-2}$ definiert bekanntlich gerade die Fibonacci-Zahlen, deren Eigenschaften man etwa im Buch von Knuth [11] nachlesen kann. Eine Darstellung der Lösung in geschlossener Form ist

$$F_i = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{i+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{i+1} \right)$$

In Algorithmen erweist sich die Fortsetzung $F_{-2} := 1, F_{-1} := 0$ der Lösung als hilfreich.

Umnormierung auf Intervall $[a, b]$

Die Iterationspunkte bei beliebigem Startintervall $[a, b]$ ergeben sich durch lineare Transformation mit dem Faktor $h := \frac{b-a}{F_n}$. Die Länge des Restintervalls in Iteration $i := 0, 1, \ldots, n-1$ ist $F_{n-i}h$. Im vorletzten Restintervall $[a_{n-2}, b_{n-2}]$ der Länge $2h$ fallen die Stützpunkte zusammen: $x_{n-2} = y_{n-2} = a_{n-2} + h$. Wegen $\hat{x} \in [a_{n-2}, b_{n-2}]$ gilt dann $|\hat{x} - x_{n-2}| \leq h$.

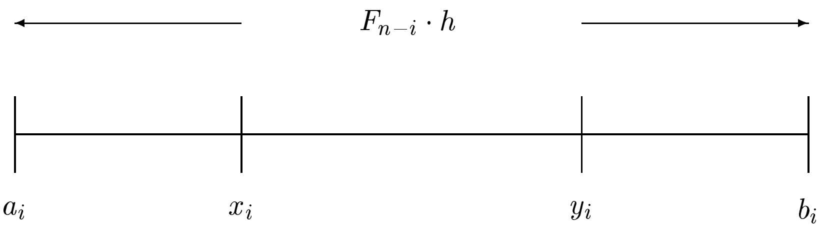

Abbildung III.5: Fibonacci-Suche

Algorithmus 1.2 (Fibonacci-Suche)

1. **Initialisierung:**
   Berechne die Fibonacci-Zahlen F_0 bis F_n.
   $i := 0$; $a_0 := a$; $b_0 := b$; $h := \frac{b-a}{F_n}$;
   $x_0 := a_0 + F_{n-2}h$; $y_0 := a_0 + F_{n-1}h$;
   $fx := f(x_0)$; $fy := f(y_0)$;

2. **Iteration:**
   Wenn $fx < fy$, dann

   $a_{i+1} := a_i$; $b_{i+1} := y_i$;
   $x_{i+1} := a_{i+1} + F_{n-i-3}h$; $y_{i+1} := x_i$;
   $fy := fx$; $fx := f(x_{i+1})$;

   sonst

   $a_{i+1} := x_i$; $b_{i+1} := b_i$;
   $x_{i+1} := y_i$; $y_{i+1} := b_{i+1} - F_{n-i-3}h$;
   $fx := fy$; $fy := f(y_{i+1})$;

   $i := i + 1$.

3. **Abbruch:**
   Wenn $i < n - 2$, dann gehe nach 2.
   $\hat{x} := x_i$; Stop.

Auch im obigen Algorithmus wird ein Funktionswert zuviel berechnet, da der im letzten Iterationschritt berechnete Funktionswert schon bekannt ist.

Ein Beispiel zur Fibonacci-Suche

Die Funktion f, definiert durch $f(x) = \sin(x - 2)$, ist unimodal auf $[a, b] := [0, 2]$. Gesucht wird der Minimalpunkt $\hat{x} = 2 - \frac{\pi}{2} \approx 0,429$. Die Fibonacci-Zahlen F_0 bis F_6 sind: $1, 1, 2, 3, 5, 8, 13$, d.h. bei Fibonacci-Suche mit $n = 6$ findet man ein Restintervall der Länge

$$h := \frac{b-a}{F_n} = \frac{2}{13} \approx 0,154.$$

dessen Genauigkeit hier ausreichen soll.

Die im Verfahren auftretenden Stützstellen (und Intervallgrenzen) sind alle von der Form $a + kh$ mit $k \in \{0, 1, \ldots, F_n\}$. Für eine Stützstelle $t \in [a, b]$ nennt man die entsprechende ganze Zahl $k(t) := \frac{t-a}{h}$ den Fibonacci-Index von t. Beim Start gilt $k(a_0) = 0$, $k(x_0) = F_{n-2}$, $k(y_0) = F_{n-1}$, $k(b_0) = F_n$.

In der Iteration werden die Funktionswerte $f(x_i)$ und $f(y_i)$ verglichen. Die Variable mit kleinerem Funktionswert bleibt Stützstelle, die mit größerem Funktionswert wird neue Intervallgrenze. Der Fibonacci-Index der neuen Stützstelle ergibt sich wegen der symmetrischen Lage der Stützstellen im Restintervall aus $k(x_{i+1}) - k(a_{i+1}) = k(b_{i+1}) - k(y_{i+1})$. Dies erspart bei Handrechnung das Nachschlagen in der Tabelle der Fibonaccizahlen. Ordnet man alle Werte einer Iteration zeilenweise in einer Tabelle an, so verschieben sich diese Werte bei Übergang zur nächsten Iteration nach rechts bzw. nach links ab der Position der neuen Stützstelle:

i	$k(a_i)$	$k(x_i)$	$k(y_i)$	$k(b_i)$	$f(x_i)$	$f(y_i)$
0	0	5	8	13	$-0,943$	$-0,696$
1	0	3	5	8	$-0,999$	$-0,943$
2	0	2	3	5	$-0,993$	$-0,999$
3	2	3	4	5	$-0,999$	$-0,983$
4	2	3	3	4	$-0,999$	$-0,999$

Mit $x_4 = y_4 = 0 + 3h$ bricht das Verfahren ab und es gilt: das Minimum von f ist näherungsweise gegeben durch $\hat{x} = 0,462 \pm 0,154$ ($\hat{x}$ liegt mit Sicherheit im Intervall $[a_4, b_4] = [0 + 2h, 0 + 4h]$).

Vergleich von goldenem Schnitt und Fibonacci-Suche

Aus $1 - \sigma = \sigma^2$ folgt induktiv $\sigma^n = (-1)^n(F_{n-2} - F_{n-1}\sigma)$. Aus $F_0 = 1 = F_1$ und $F_1 = 1 < \sigma^{-1} < 2 = F_2$ erhalten wir $F_2 < \sigma^{-2} \cdot (\sigma^2 + \sigma) < F_3$, also mit Hilfe der ersten Formel $F_2 < \sigma^{-2} < F_3$. Induktiv folgen für $n > 1$ die Abschätzungen:

$$\frac{1}{F_n} > \sigma^n > \frac{1}{F_{n+1}}, \qquad \lim_{n\to\infty} \frac{F_{n-1}}{F_n} = \sigma.$$

Demnach ist der Goldene Schnitt bei gleichem Aufwand von vergleichbarer kaum geringerer Genauigkeit. Auf der anderen Seite muß bei Fibonacci-Suche die gewünschte Genauigkeit a priori bekannt sein, um n zu bestimmen.

Beide Verfahren reduzieren die Intervallänge linear, d.h. es gilt $|b_{i+1} - a_{i+1}| \leq \lambda |b_i - a_i|$ für ein $\lambda \in (0, 1)$. Dabei ist der Reduktionsfaktor λ durch σ bzw. durch eine Schranke für $\frac{F_{n-1}}{F_n}$ gegeben. Nach

$$n = \left\lceil \frac{1}{\log \lambda} \left(\log(b - a) + \log \frac{1}{\epsilon} \right) \right\rceil$$

Iterationsschritten ist die Länge des Restintervalls auf weniger als ϵ reduziert. Als log wird in solchen Abschätzungen üblicherweise der Logarithmus zur Basis 2 gewählt, da dann die Logarithmen der Eingabedaten als Größe des Problems (Speicherplatz in einem Computer) interpretiert werden können.

Differenzierbare und konvexe Funktionen

Da das Minimum einer auf einem offenen Intervall $I \subseteq \boldsymbol{R}$ stetig differenzierbaren Funktion f notwendigerweise auch eine Nullstelle der Ableitung $g \equiv f'$ ist, versucht man zunächst, ein $\hat{x} \in I$ mit $g(\hat{x}) = 0$ zu finden. Kennt man ein Intervall $[a, b] \subset I$ mit

$$g(a) < 0 < g(b),$$

so folgt aus der Stetigkeit von g die Existenz einer Nullstelle in (a, b). Ist g monoton, so gibt es genau eine Nullstelle und diese ist das globale Minimum von f. Das einfachste Verfahren zur Bestimmung einer Nullstelle in I ist das Bisektionsverfahren.

Algorithmus 1.3 (Das Bisektionsverfahren)

1. **Iteration:**
   $x := \frac{a+b}{2}$; $g_x := g(x)$;
   Wenn $g_x \approx 0$, dann $\hat{x} := x$; Stop.
   Wenn $g_x < 0$, dann $a := x$;
   Wenn $g_x > 0$, dann $b := x$;

2. **Abbruch:**
   Wenn $b - a < \epsilon$, dann $\hat{x} := x$; Stop.
   Gehe nach 1.

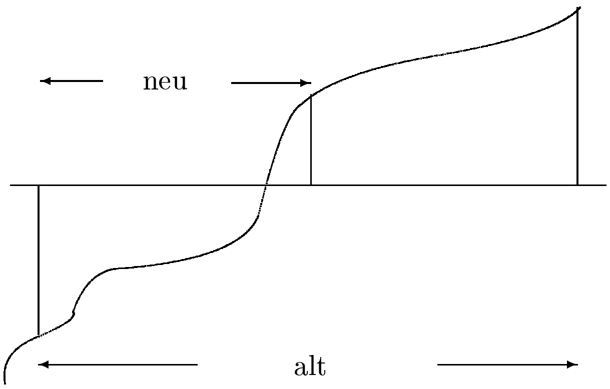

Abbildung III.6: Bisektion

Da das Bisektionsverfahren das Restintervall in jeder Iteration um den Faktor $\lambda = \frac{1}{2}$ reduziert, handelt es sich wieder um ein lineares Verfahren, das unter den angegebenen

Voraussetzungen stets eine Nullstelle von g lokalisiert. Schnellere Verfahren wie das im folgenden beschriebene Newton-Verfahren konvergieren im allgemeinen nur unter strengeren Voraussetzungen.

Sei wieder Intervall $[a,b] \subset I$ mit $g(a) < 0 < g(b)$ bekannt. Außerdem sei nunmehr $f \in C_2(I)$. Das Newton-Verfahren wird mit Hilfe der Iterationsfunktion

$$\Phi(x) := x - \frac{g(x)}{g'(x)}$$

beschrieben.

Algorithmus 1.4 (Newton-Verfahren)

1. $x_0 := b$.

2. Für $k = 0, 1, \ldots$*:*

$$x_{k+1} := \Phi(x_k);$$

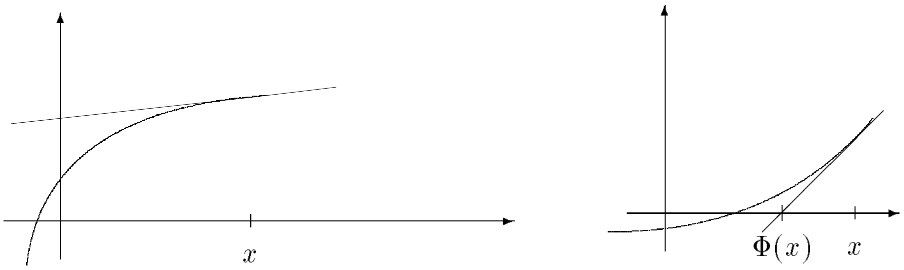

Abbildung III.7: Newton-Verfahren

Selbst wenn g monoton ist und daher eine eindeutige Nullstelle in (a,b) existiert, kann das Newtonverfahren versagen. Im folgenden betrachten wir eine wichtige Klasse von Funktionen, für die das Newton-Verfahren zum Erfolg führt.

Definition (Konvexe Funktionen): $g : I \to \boldsymbol{R}$ heißt *konvex*, falls

$$g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda) g(y)$$

für alle $\lambda \in (0,1)$, $x, y \in I$.

Im Abschnitt 2 werden konvexe Funktionen im $\boldsymbol{R}^n$ untersucht und auf verschiedene Weise charakterisiert. Für unsere Analyse des Newton-Verfahrens genügt uns die folgende grundlegende Eigenschaft konvexer, differenzierbarer Funktionen:

Lemma 1.1 (Gradientenungleichung) *$g : I \to \boldsymbol{R}$ sei eine konvexe, auf $(a,b) \subseteq I$ stetig differenzierbare Funktion. Dann gilt*

$$g(y) \geq g(x) + g'(x)(y-x)$$

für alle $x, y \in (a,b)$.

Beweis. Die Konvexitätsbedingung $g(\lambda y + (1-\lambda)x) \leq \lambda g(y) + (1-\lambda)g(x)$ impliziert $\frac{g(x+\lambda(y-x))-g(x)}{\lambda(y-x)}(y-x) \leq g(y)-g(x)$. Der Grenzübergang $\lambda \to 0$ führt daher auf $g'(x)(y-x) \leq g(y)-g(x)$. □

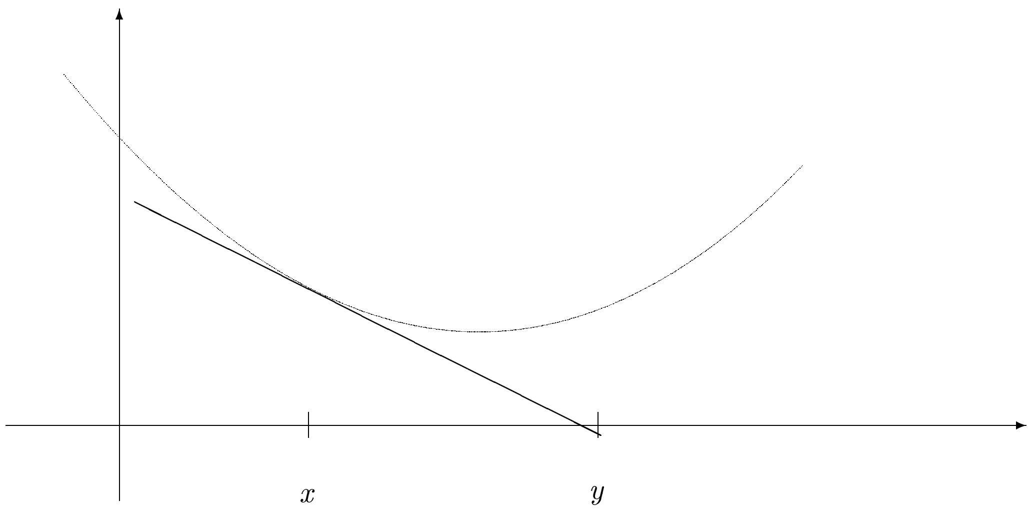

Abbildung III.8: Gradientenungleichung

Folgerung: Differenzierbare konvexe Funktionen besitzen monoton wachsende Ableitungen:

$$x < y \Rightarrow g'(x) \leq g'(y),$$

denn setzt man die Gradientenungleichung mit vertauschten Argumenten, d.h. $g(x) \geq g(y) + g'(y)(x-y)$ in die Gradientenungleichung ein, so folgt $g(y) \geq g(y) + g'(y)(x-y) + g'(x)(y-x)$. Also gilt $g'(y)(y-x) \geq g'(x)(y-x)$, woraus wegen $y - x > 0$ die Behauptung $g'(y) \geq g'(x)$ folgt.

Satz 1.2 (Globale Konvergenz des Newton-Verfahrens) *$g : I \to \boldsymbol{R}$ sei eine konvexe, auf $(a,b) \subseteq I$ stetig differenzierbare Funktion mit $g(a) < 0 < g(b)$. Dann besitzt g eine eindeutige Nullstelle $x_* \in (a,b)$, wobei $g'(x_*) > \frac{-g(a)}{b-a} > 0$, und für die im Newtonverfahren erzeugte Folge aus Algorithmus 1.4 gilt:*

1. *$(x_k) \searrow x_*$ (streng monoton fallend)*

Sei $\epsilon > 0$. Dann gilt zusätzlich:

2. *$g(x_k) \leq \epsilon \qquad$ für $2k \geq \left\lceil \log_2 \left(\frac{g'(b)g(b)}{g'(x_*)} \right) + \log_2 \frac{1}{\epsilon} \right\rceil$*

3. *$x_k - x_* \leq \epsilon \qquad$ für $2k \geq \left\lceil \log_2 \left(\frac{g'(b)g(b)}{(g'(x_*))^2} \right) + \log_2 \frac{1}{\epsilon} \right\rceil$*

Beweis. Da g stetig, gibt es nach Zwischenwertsatz eine Nullstelle $x_* \in (a,b)$. Angenommen, x'_* ist eine zweite Nullstelle. OBdA $x_* < x'_*$, d.h. $x_* = \lambda a + (1-\lambda)x'_*$ für ein $\lambda \in (0,1)$.

Da g konvex, folgt mit $0 = g(x_*) \leq \lambda g(a) + (1-\lambda) g(x_*') = \lambda g(a) < 0$ ein Widerspruch. Im Rest des Beweises sei stets $a \leq x < x_* < y \leq b$. Aus der Eindeutigkeit der Nullstelle folgt für solche Argumente:

$$g(x) < 0 < g(y) \tag{1.2}$$

Die Gradientenungleichung liefert $g(a) \geq g(y) + g'(y)(a-y)$. Also folgt

$$g'(y) > \frac{g(y)-g(a)}{y-a} \geq \frac{-g(a)}{b-a} > 0. \tag{1.3}$$

Wegen (1.2) und (1.3) gilt $\Phi(y) < y$. Die Gradientenungleichung $g(y) + g'(y)(x_* - y) \leq g(x_*) = 0$ kann man nun durch $g'(y)$ teilen und erhält $x_* \leq y - \frac{g(y)}{g'(y)} = \Phi(y)$. Also ist die durch Algorithmus (1.4) erzeugte Newtonfolge x_k streng monoton fallend mit $x_k \geq x_*$.

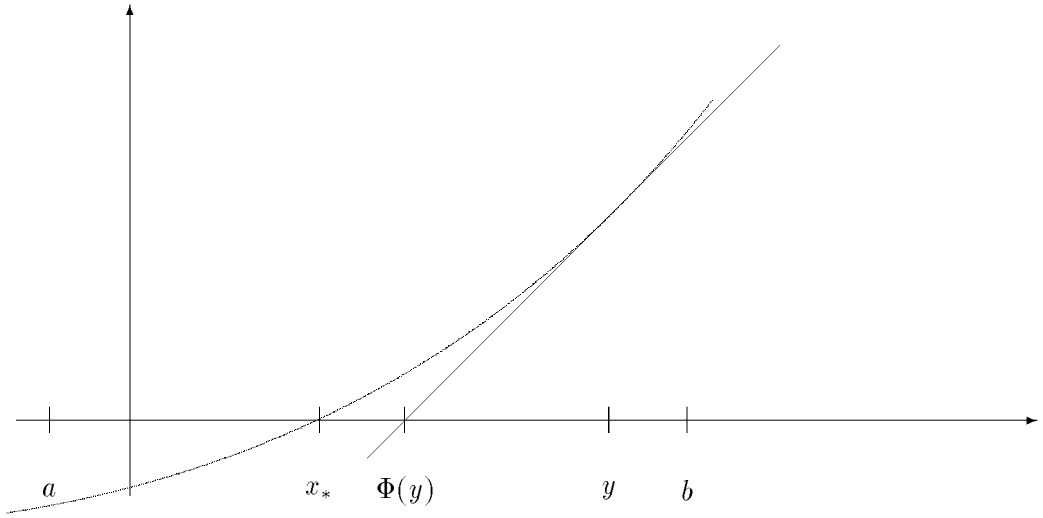

Abbildung III.9: Newton-Verfahren für konvexe Funktionen

Die Gradientenungleichung impliziert weiterhin

$$g(y) \geq g(\Phi(y)) + g'(\Phi(y))(y - \Phi(y)).$$

Mit Division durch $g(y)$ erhält man

$$1 \geq \frac{g(\Phi(y))}{g(y)} + \frac{g'(\Phi(y))}{g'(y)} =: \alpha + \beta.$$

Die Ungleichung zwischen dem geometrischen und dem arithmetischen Mittel ergibt nun $\sqrt{\alpha \cdot \beta} \leq \frac{1}{2}(\alpha+\beta) \leq \frac{1}{2}$, also $\alpha \cdot \beta \leq \frac{1}{4}$. Offensichtlich gilt

$$g\left(\Phi(y)\right) g'\left(\Phi(y)\right) \leq \frac{1}{4} g(y) g'(y),$$

woraus wir nach k Iterationen

$$g(x_k) g'(x_k) \leq (\frac{1}{4})^k g(b) g'(b)$$

erhalten. Zusammen mit $g'(x_k) \geq g'(x_*)$ liefert diese Ungleichung:

$$g(x_k) \leq (\frac{1}{4})^k \frac{g(b)g'(b)}{g'(x_*)}, \tag{1.4}$$

woraus (2.) folgt. Eine weitere Anwendung der Gradientenungleichung zeigt $g(x_k) \geq g(x_*) + g'(x_*)(x_k - x_*)$, also $\frac{g(x_k)}{g'(x_*)} \geq x_k - x_*$. Damit folgt aus (1.4) auch (3.). □

Definition (Konvergenzgeschwindigkeit): Sei (r_n) eine positiveNullfolge, d.h. $r_n \to 0, r_n > 0$. Dann heißt (r_n) *(mindestens) konvergent* von der Ordnung p, falls es ein $b > 0$ und ein $N \in \boldsymbol{N}$ gibt, so daß $r_{n+1} \leq b \cdot r_n^p$ für alle $n > N$.

(r_n) hat die *Konvergenzordnung* $p := \sup\{q > 0 \mid 0 \leq \beta_q = \limsup_{n\to\infty} \frac{r_{n+1}}{r_n^q} < \infty\}$. Dabei heißt β_q *Konvergenzfaktor* zur Ordnung q.

Beispiel:

1. Für $r_{n+1} := \frac{r_n}{2}$ mit $r_0 := 1$ gilt: $p = 1$, $\beta_1 = \frac{1}{2}$.
2. Für $r_{n+1} := r_n^2$ mit $r_0 := \frac{1}{2}$ gilt: $p = 2$, $\beta_2 = 1$.
3. Für $r_n := \frac{1}{n}$ gilt: $\beta_1 = 1$, $p = 1$.
4. Für $r_n := (\frac{1}{2})^{n^2}$ gilt: $\beta_1 = 0$,$p = 1$.

Insbesondere bezeichnet man $p = 1$, $0 < \beta_1 < 1$ als *lineare Konvergenz*, $p \geq 1, \beta_1 = 0$ als *superlineare Konvergenz* und $p = 2$ als *quadratische Konvergenz*. Die bei den bisherigen Verfahren (Goldener Schnitt, Fibonacci-Suche, Bisektionsverfahren) betrachtete Länge des Restintervalls ist jeweils eine linear konvergente Nullfolge, für die das Konvergenzverhalten unmittelbar ab $N = 1$ einsetzt. Im allgemeinen beschreibt die Konvergenzgeschwindigkeit nur das asymptotische Verhalten der Folge. Dies wird auch bei der folgenden Untersuchung des Newton-Verfahrens deutlich, die wir für lipschitzstetig differenzierbare Funktionen durchführen.

Lemma 1.3 (L-Lipschitzstetig differenzierbare Funktionen) *Sei $L > 0$ und g eine auf (a, b) stetig differenzierbare Funktion mit $|g'(y) - g'(x)| \leq L \cdot |y - x|$. Dann gilt:*

$$|g(y) - g(x) - g'(x) \cdot (y - x)| \leq \frac{1}{2} L (y - x)^2$$

für alle $x, y \in (a, b)$.

Beweis.

$$\begin{aligned} |g(y) - g(x) - g'(x)(y - x)| &\leq \int_x^y |g'(z) - g'(x)| dz \\ &\leq L \int_x^y (z - x) dz \\ &= \frac{L}{2}(y - x)^2 \end{aligned}$$

□

Mit Hilfe dieser Abschätzung kann man die Konvergenzgeschwindigkeit des Newtonverfahrens bestimmen.

Satz 1.4 (Lokale Konvergenz des Newton-Verfahrens) *Sei f eine auf (a, b) zweimal stetig differenzierbare Funktion mit L-lipschitzstetig differenzierbarer Ableitung $g \equiv f'$ und mit $|g'| \geq \rho > 0$. Besitzt $g(x) = 0$ eine Lösung $x_* \in (a, b)$, dann existiert ein $\eta > 0$ mit*

1. $|x_0 - x_*| < \eta \quad \Rightarrow \quad (x_k) \to x_*$
2. $|x_{k+1} - x_*| \leq \frac{L}{2\rho}(x_k - x_*)^2$

für die Newtonfolge $x_{k+1} := x_k - \frac{g(x_k)}{g'(x_k)}$.

Beweis. (Übungsaufgabe) □

Der Satz formuliert lokal, hinreichend nahe beim gesuchten Minimum x_*, eine Konvergenzaussage. Dort liefert das Newtonverfahren in jedem Schritt eine Verdopplung der Genauigkeit, d.h. eine typische Entwicklung des Abstandes $|x_{k+1} - x_*|$ ist etwa 10^{-2}, 10^{-4}, $10^{-8}, \ldots$.

Newton-Verfahren mit finiten Differenzen

Approximiert man die zur Durchführung des Newtonverfahrens explizit benötigte Ableitung von f, so erhält man eine Reihe von Varianten. Die geschickte Approximation $g'(x_k) \approx \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$ führt auf das Sekantenverfahren:

$$x_{k+1} = x_k - g(x_k) \frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})}, \quad k \in \boldsymbol{N}.$$

wobei zur Initialisierung zwei Startpunkte x_0, x_1 gewählt werden müssen. Bei allgemeiner Approximation der Ableitung mit Hilfe des Differenzenquotienten zur Schrittweite h_c spricht man vom Newton-Verfahren mit finiten Differenzen. Der jeweils neue Iterationspunkt x_+ ergibt sich aus dem alten x_c durch

$$a_c := \frac{g(x_c + h_c) - g(x_c)}{h_c}; \quad x_+ := x_c - \frac{g(x_c)}{a_c}$$

Konvergenzabschätzung

Für eine auf (a, b) L-lipschitzstetig differenzierbare Funktion g mit $x_c, x_c + h_c \in (a, b)$ ist die Qualität der Approximation abschätzbar durch:

$$|a_c - g'(x_c)| \leq \frac{L}{2}|h_c|,$$

denn nach Lemma 1.3 gilt $|g(x_c + h_c) - g(x_c) - h_c g'(x_c)| \leq \frac{L}{2}|h_c|^2$. Ist $g(x_*) = 0$, so ergibt sich

$$\begin{aligned} x_+ - x_* &= -\frac{g(x_c)}{a_c} + x_c - x_* \\ &= \frac{1}{a_c}\left[g(x_*) - g(x_c) - a_c(x_* - x_c)\right] \\ &= \frac{1}{a_c}\left[g(x_*) - g(x_c) - g'(x_c)(x_* - x_c) + (g'(x_c) - a_c)(x_* - x_c)\right] \end{aligned}$$

Mit $e_+ := |x_+ - x_*|, e_c := |x_* - x_c|$ folgt $e_+ \leq \frac{1}{a_c}\left[\frac{L}{2}(x_* - x_c)^2 + \frac{L}{2}|h_c|e_c\right]$, und daher

$$e_+ \leq \frac{L}{2|a_c|}(e_c + |h_c|)e_c \tag{1.5}$$

Diese Abschätzung impliziert unter einer Reihe unterschiedlicher Annahmen Konvergenzaussagen:

Satz 1.5 (Lokale Konvergenz mit finiten Differenzen) *Sei f eine auf (a,b) zweimal stetig differenzierbare Funktion mit L-lipschitzstetig differenzierbarer Ableitung $g \equiv f'$ und mit $|g'| \geq \rho > 0$. Besitzt $g(x) = 0$ eine Lösung $x_* \in (a,b)$, dann existieren $\eta, \eta' > 0$, so daß für $0 < |h_k| \leq \eta', |x_0 - x_*| < \eta$ gilt:*

1. *x_k konvergiert mindestens linear gegen x_*,*
2. *falls $h_k \to 0$, so ist die Konvergenz mindestens superlinear,*
3. *falls $|h_k| \leq \alpha|x_k - x_*|$ oder $|h_k| \leq \alpha|g(x_k)|$ für ein $\alpha > 0$, so ist die Konvergenz mindestens quadratisch,*
4. *falls $|h_k| \leq \alpha|x_k - x_{k-1}|$ für ein $\alpha > 0$, so ist die Konvergenz mindestens quadratisch über je zwei Schritte .*

Beweis. (Übungsaufgabe), mittels (1.5) □

Bemerkung (Numerischer Vergleich und numerische Durchführung):

1. Das Newtonverfahren benötigt zwei Funktionsauswertungen (von g und g') pro Iteration und ist mindestens quadratisch konvergent. Unter den gleichen Voraussetzungen benötigt das Sekantenverfahren zwei Auswertungen (von g) in zwei Iterationen und ist wegen (4.) des vorstehenden Satzes, wenn man je zwei Iterationen zusammenfaßt, ebenfalls quadratisch konvergent. Eine genauere Konvergenzanalyse (s. Stoer [29]) zeigt sogar $p = 2.618\ldots = 1 + \frac{\sqrt{5}+1}{2}$ für Doppelschritte beim Sekantenverfahren.

2. Finite Differenzen sind numerisch nicht unproblematisch. Aus $h_k := \alpha|g(x_k)| \to 0$ folgt $x_k + h_k \to x_k$. Eine praktikable Wahl ist etwa $h_k := \sqrt{eps} \cdot \max\{\text{typ } x, |x_k|\}$. Hierbei gibt typ x die zu erwartende Größenordnung von x und eps die Maschinengenauigkeit an.

3. Im Gegensatz zur Nullstellensuche ist bei der Minimierung von Funktionen eine natürliche Qualitätskontrolle zur Hand. Falls $f(x_{k+1}) < f(x_k)$, ist die Iteration sinnvoll. Anderenfalls, falls wenigstens eine Abstiegsrichtung vorliegt, d.h. $f'(x_k)(x_{k+1} - x_k) < 0$ gilt, so stimmt die Richtung des Iterationsschritts und man kann versuchen durch, falls notwendig mehrfache, Halbierung der Schrittweite einen besseren Iterationspunkt zu finden. Ist die Richtung keine Abstiegsrichtung, so kann man in der Gegenrichtung suchen.

4. Als Abbruchkriterien kann man viele verschiedene Bedingungen verwenden, die alle mehr oder minder gut motiviert werden können. Beim Newton-Verfahren benutzt man die auftretenden relativen bzw. absoluten Änderungen:
$$\frac{|x_+ - x_c|}{\max\{\text{typ } x, |x_+|\}} < \tau_1; \quad \left|\frac{f'(x_+)}{f(x_+)}x_+\right| < \tau_2; \quad \left|\frac{f'(x_+)}{f(x_+)}\right| < \tau_3.$$

Interpolation

Sei f zweimal stetig differenzierbar auf dem offenen Intervall I. Gesucht wird wieder $f(x_*) = \min\{f(x) \mid x \in I\}$. Man kann nun f durch ein Interpolationspolynom P zu f approximieren und versuchen, dessen Minimum x_+ mit $P'(x_+) = 0, P''(x_+) > 0$ als Näherung für $\hat{x}$ zu verwenden. Das quadratische Interpolationspolynom P (zu f an der Stelle x_c) und seine Ableitung sind durch

$$P(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{1}{2}f''(x_c)(x - x_c)^2$$
$$P'(x) = f'(x_c) + f''(x_c)(x - x_c)$$

gegeben (Taylorentwicklung). Löst man die zweite Gleichung nach x auf, so erkennt man, daß eine Newton-Iteration eine Nullstelle von P' liefert:

$$x_+ = x_c - \frac{f'(x_c)}{f''(x_c)}.$$

Diese Idee verwendet man in der quadratischen Interpolation iterativ. Allerdings wird statt des Taylorpolynoms das Lagrange'sche Interpolationspolynom zu drei meist äquidistanten Stützstellen benutzt:

$$f(x) \approx \sum_{i=1}^{3} f_i \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)} =: P(x)$$

Äquidistante quadratische Interpolation

Zu äquidistanten Stützstellen $x := y-h$, y, $z := y+h$ mit Schrittweite h seien die Stützwerte $f_\chi := f(\chi)$ für $\chi \in \{x, y, z\}$ bekannt. Dann ist die Nullstelle x_+ von P' gegeben durch:

$$x_+ = y + \frac{h}{2}\frac{f_x - f_z}{f_x - 2f_y + f_z}.$$

Der Nenner ist wohldefiniert, wenn die 2. Ableitung nicht verschwindet, denn

$$P''(x_+) = \frac{1}{h^2}(f_x - 2f_y + f_z),$$

woran man auch erkennt, ob ein Minimum oder Maximum bestimmt wurde. Die für ein Minimum hinreichende Bedingung $f_y < \min(f_x, f_z)$ impliziert auch $x_+ \in [x, z]$. Daher wird zunächst eine Näherungslösung y zur Schrittweite h gesucht, für die die Bedingung

$$f_y \leq f_x, f_z$$

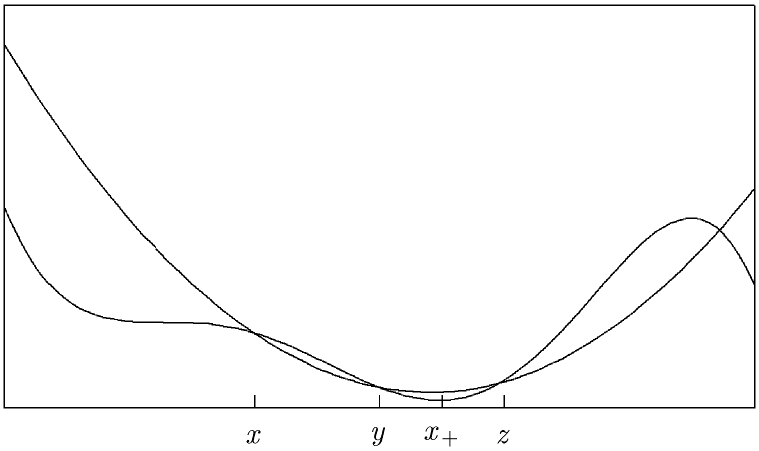

Abbildung III.10: Quadratische Interpolationsfunktionen

erfüllt ist. Dann wird x_+ berechnet und anschließend die Schrittweite h verkleinert. Kennt man näherungsweise den minimalen Funktionswert $f_* \approx f(x_*)$, und die Ableitung $f'(y)$, so kann die Schrittweite mit

(1.6) $$h := 2\frac{f_* - f_y}{f'(y)}$$

automatisch gesteuert werden. Diese Formel erhält man, wenn man die Taylorentwicklung (1. Ordnung) der Ableitung

$$0 = f'(x_*) = f'(y) + f''(y)h$$

mit $(-\frac{h}{2})$ multipliziert, von der Taylorentwicklung (2. Ordnung) der Funktion

$$f_* = f(x_*) = f(y) + f'(y)h + \frac{1}{2}f''(y)h^2$$

abzieht, und die resultierende Gleichung

$$f_* = f(y) + \frac{h}{2}f'(y)$$

nach h auflöst. Im folgenden Verfahren ist die systematische Suche nach einem passenden Näherungswert y mit einer teilweisen Schrittweitensteuerung gekoppelt. Die in Gleichung (1.6) angegebene Schrittweite kann gegebenenfalls in den Schritten (1.) und (6.) verwendet werden.

Algorithmus 1.5 (Das Verfahren nach Davies, Swann und Campey)

1. Wähle Näherung y und Schrittweite h.

2. Falls $f(y) < f(y-h)$ und $f(y) < f(y+h)$, dann gehe nach Schritt 5.

   Anderenfalls wähle das Vorzeichen von h, so daß $f(y) \geq f(y+h)$.

3. $h_0 := h$;

   Solange $f(y+h) \geq f(y+2h)$: $h := 2h$;

   Falls $h = h_0$, dann $y := y + h$ und gehe nach 5.

   $h := \frac{1}{2}h$.

4. Falls $f(y+2h) \leq f(y+3h)$, dann $y := y + 2h$ und gehe nach 5.

   $y := y + 3h$.

5. $x_+ := y + \frac{h}{2} \cdot \frac{f(y-h)-f(y+h)}{f(y-h)-2f(y)+f(y+h)}$

6. Falls $f(y) > f(x_+)$, dann $y := x_+$.

   Falls Abbruchbedingung erfüllt, stop mit Ergebnis y.

   Wähle kürzere Schrittlänge h und gehe nach 2.

2 Lokale und Globale Minima

Stetig differenzierbare Funktionen bezeichnen wir wie üblich kurz als C_1-Funktionen, m-mal stetig differenzierbare als C_m-Funktionen. Ist f eine C_m-Funktion auf $X \subseteq \boldsymbol{R}^n$, so schreiben wir kurz $f \in C_m(X)$. Die Menge der inneren Punkte von X bezeichnen wir mit $\mathrm{int}X$. Ist $f \in C_1$, so bezeichnet $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$ den Gradienten, ist $f \in C_2$, so bezeichnet $F = \nabla^2 f = (\frac{\partial^2 f}{\partial x_i \partial x_j})$ die Matrix der zweiten partiellen Ableitungen von f.

In diesem Abschnitt stellen wir Begriffe, Optimalitäts- und Konvergenzbedingungen für Verfahren zulässiger Richtungen zusammen, die als theoretische Grundlage zur Lösung der Aufgabe

$$\min\{f(x) \mid x \in X\}$$

nützlich sind.

Optimalitätsbedingungen

Definition (Minima): $U_\epsilon(x_*) := \{x \mid \| x - x_* \| < \epsilon\}$ heißt *ϵ-Kugel* um x_*.

1. $x_* \in X$ heißt *lokales Minimum* (von f auf X) genau dann, wenn

$$f(x_*) \leq f(x) \quad \exists\, \epsilon > 0 \quad \forall\, x \in X \cap U_\epsilon(x_*), x \neq x_*.$$

2. $x_* \in X$ heißt *globales Minimum* (von f auf X) genau dann, wenn

$$f(x_*) \leq f(x) \quad \forall\, x \in X, x \neq x_*.$$