Teil II

Nichtlineare Optimierung

Kapitel 1

Einleitung

In diesem Abschnitt wird die Optimierung von Funktionen

$$\min_{\mathbf{x}\in\Omega} \{f(\mathbf{x})\}$$

betrachtet, wobei $\Omega \subset \mathbb{R}^n$ eine abgeschlossene Menge und $f : \Omega \to \mathbb{R}$ eine gegebene Funktion ist. Die Funktion f heißt Zielfunktion und Ω sei durch endlich viele Nebenbedingungen beschrieben. Bei den betrachteten Problemen können sowohl die Zielfunktion als auch die Nebenbedingungen nichtlinear sein.

Beispiel 1.1 *Ausgleichsrechnung.* Ein konkretes Beispiel für einen Aufgabe der Nichtlinearen Optimierung ist die Ausgleichsrechnung. Zur mathematischen Formulierung von Gesetzmäßigkeiten, die ein technisches oder physikalisches Phänomen beschreiben, wird häufig eine Hypothese über den möglichen funktionalen Zusammenhang bekannter und beobachteter Variablen formuliert. Diese enthält im allgemeinen noch freie Parameter und hat etwa die Gestalt

$$z = g(\mathbf{x}, \boldsymbol{\lambda}), \quad \mathbf{x} \in \mathbb{R}^n, \ \boldsymbol{\lambda} \in \mathbb{R}^p.$$

In diesem Modell ist $\boldsymbol{\lambda}$ ein unbekannter Parametervektor mit p Komponenten und z ist eine Variable, deren Abhängigkeit modelliert werden soll.

Nach Auswertung von m Experimenten kennt man m Paare von Variablenwerten $(z_i, \mathbf{x}_i)$, die unter Berücksichtigung möglicher Beobachtungsfehler ε_i die funktionale Abhängigkeit annähernd erfüllen sollen. Das heißt, bei passenden Parametern muss gelten

$$z_i = g(\mathbf{x}_i, \boldsymbol{\lambda}) + \varepsilon_i, \quad i = 1, \ldots, m.$$

Um möglichst kleine Abweichungen ε_i zu erhalten, versucht man, die unbekannten Parameter $\boldsymbol{\lambda}$ optimal anzupassen, indem man entsprechende Optimierungsaufgaben löst, zum Beispiel

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^p} \sum_{i=1}^{m} \left(z_i - g(\mathbf{x}_i, \boldsymbol{\lambda})\right)^2$$

(Methode der kleinsten Quadrate) oder

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^p} \sum_{i=1}^{m} \omega_i \left|z_i - g(\mathbf{x}_i, \boldsymbol{\lambda})\right|^r,$$

wobei $r \geq 1$ eine ganze Zahl ist und $\boldsymbol{\omega} \geq \mathbf{0}$. □

Beispiel 1.2 *Standortplanung.* Eine Firma möchte n neue Lager der Kapazität a_i, $i = 1, \ldots, n$, errichten. Gegeben sind die Standorte der Abnehmer (α_j, β_j) sowie der Bedarf b_j, $j = 1, \ldots, m$. In einigen Gebieten darf zudem nicht gebaut werden (ε_k–Umgebungen von (γ_k, δ_k), $k = 1, \ldots, p$.)

Gesucht sind die Standorte der Lager (x_i, y_i), die den Lieferplan z optimieren. Mit den Bezeichnungen

- z_{ij} – Transportmenge vom Lager i zum Abnehmer j,
- d_{ij} – Entfernung Lager i und Abnehmer j,
- Δ_{ik} – Entfernung Lager i zum Mittelpunkt (γ_k, δ_k) der Verbotszone k,

lautet die Optimierungsaufgabe wie folgt:

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij} z_{ij} &\rightarrow \min\ ! \\
\sum_{j=1}^{m} z_{ij} &\leq a_i, \quad i = 1, \ldots, n \\
\sum_{i=1}^{n} z_{ij} &= b_j, \quad j = 1, \ldots, m \\
\Delta_{ik} &\geq \varepsilon_k, \quad i = 1, \ldots, n,\ k = 1, \ldots, p, \\
z_{ij} &\geq 0, \quad i = 1, \ldots, n,\ j = 1, \ldots, m.
\end{aligned}$$

Diese Aufgabe kann für verschiedene Entfernungsmaße gestellt werden, etwa für

$$d_{ij} = |x_i - \alpha_j| + |y_i - \beta_j|$$

oder die Euklidische Entfernung

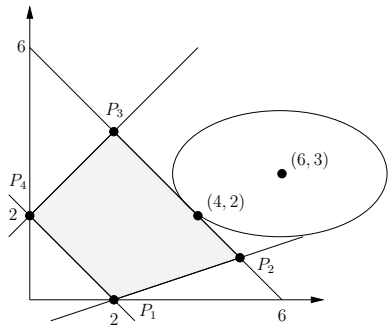
$$d_{ij} = \sqrt{(x_i - \alpha_j)^2 + (y_i - \beta_j)^2}.$$

Wird sowohl für d_{ij} als auch für Δ_{ik} die Euklidische Entfernung gewählt, so erhält man eine Optimierungsaufgabe mit quadratischer Zielfunktion und quadratischen Nebenbedingungen. □

Beispiel 1.3 *Quadratische Zielfunktion über konvexem Polyeder.* Wir betrachten ein nichtlineares Programm mit quadratischer Zielfunktion und linearen Nebenbedingungen:

$$\begin{aligned}
z = (x_1 - 6)^2 + 2(x_2 - 3)^2 &\rightarrow \min\ ! \\
x_1 + x_2 &\geq 2 \\
x_1 - x_2 &\geq -2 \\
x_1 + x_2 &\leq 6 \\
x_1 - 3x_2 &\leq 2 \\
\mathbf{x} &\geq \mathbf{0}.
\end{aligned}$$

Durch die Zielfunktion werden konzentrische Ellipsen mit Mittelpunkt $(6, 3)^T$ beschrieben. Die optimale Lösung ist ein Punkt auf dem Rand $\partial\Omega$ von Ω, aber kein Eckpunkt: $\mathbf{x}_{\text{opt}} = (4, 2)^T$, $z_{\text{opt}} = 6$.



Betrachtet man eine andere Zielfunktion

$$z = (x_1 - 2)^2 + 2\,(x_2 - 2)^2 \to \min\ !,$$

dann ist das Optimum $\mathbf{x}_{\text{opt}} = (2,2)^T$, $z_{\text{opt}} = 0$. Damit liegt das Optimum im Inneren von Ω.

Eine andere quadratische Zielfunktion ist

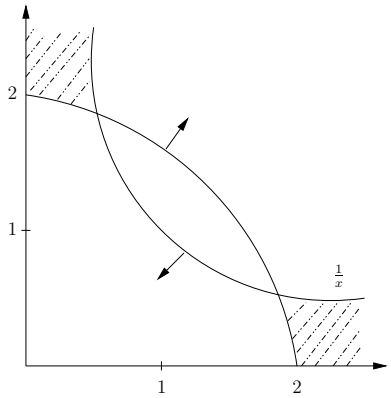
$$z = 2\,(x_1 - 2)^2 + (x_2 - 2)^2 \to \max\ !$$

Bei dieser Zielfunktion gibt es eine Ellipse, die gleichzeitig durch P_1 und P_3 geht. Beide Eckpunkte von Ω sind lokales Maximum mit $z = 4$. Ein weiteres lokales Maximum erhält man in P_4 mit $z = 8$. Das globale Maximum wird jedoch in P_2 mit $z = 19$ angenommen. Ein simplexartiges Vorgehen, wobei man von Eckpunkt zu Eckpunkt geht und in jedem Schritt den Zielfunktionswert verbessert (vergrößert), d.h. $P_1 \to P_4$ oder $P_3 \to P_4$ führt hier nicht zum Ziel. □

Beispiel 1.4 *Unzusammenhängender zulässiger Bereich.* Der zulässige Bereich Ω sei definiert durch

$$\begin{aligned} x_1^2 + x_2^2 &\geq 4 \\ x_1 x_2 &\leq 1 \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned}$$

Der zulässige Bereich besteht aus zwei getrennt liegenden Teilen. Beide sind nicht konvex. Selbst bei einer linearen Zielfunktion können lokale Minima auftreten, die keine globalen sind.



□

In der Vorlesung werden u.a. folgende Problemklassen innerhalb der nichtlinearen Programme nicht betrachtet:

- mehrere Zielfunktionen,
- unendlich viele Nebenbedingungen (semi–infinite Programme),
- stochastische Daten (stochastische Optimierung).

Kapitel 2

Nichtlineare Optimierung ohne Nebenbedingungen

In diesem Abschnitt sollen im wesentlichen Verfahren zur Bestimmung des Minimums von nichtglatten Funktionen in einer Variablen im Detail vorgestellt werden, wobei der zulässige Bereich nicht durch Nebenbedingungen eingeschränkt ist. Die Minimierung glatter Funktionen ohne Nebenbedingungen kann auf die Bestimmung von Nullstellen zurückgeführt werden. Verfahren zur Lösung dieser Aufgabe sind bereits aus der Grundvorlesung *Praktische Mathematik* bekannt.

2.1 Minimierung nichtglatter Funktionen in einer Variablen

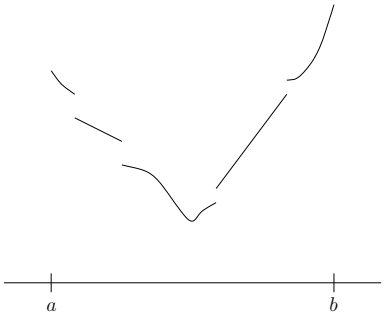
Die Lösung nichtlinearer Optimierungsaufgaben in einer Dimension tritt als Teilproblem in den meisten Verfahren zur Lösung nichtlinearer Optimierungsprobleme in höheren Dimensionen auf.

Seien $I \subset \mathbb{R}$ ein gegebenes abgeschlossenes Intervall und $f : I \to \mathbb{R}$ eine gegebene Funktion. Dann sucht man ein $\hat{x} \in I$, welches

$$f(\hat{x}) = \min_{x \in I} f(x) \tag{2.1}$$

erfüllt. Um ein solches Minimum mit Hilfe eines Verfahrens effizient bestimmen zu können, müssen einige einschränkende Bedingungen an f gestellt werden. Je nach Eigenschaften der Funktion, ist dann ein entsprechendes Verfahren zu wählen.

Definition 2.1 Unimodale Funktion. *Eine Funktion $f : I \to \mathbb{R}$ heißt unimodal, falls für ein $\hat{x} \in I = [a, b]$ gilt, dass f streng monoton fallend auf $[a, \hat{x}]$ und streng monoton steigend auf $[\hat{x}, b]$ ist.*



Unimodale Funktionen erlauben nach Auswertung der Funktion an zwei Stellen $x < y$ mit $a < x < y < b$ die Reduktion des Intervalls, in der man das Minimum zu suchen hat:

- 1. Fall: $f(a) \leq f(x)$, dann ist das Minimum in $[a, x]$.
- 2. Fall: $f(a) > f(x) > f(y)$. Dann kann das Minimum nicht in $[a, x]$ sein.
- 3. Fall: $f(a) > f(x)$ und $f(x) < f(y)$ und $f(y) < f(b)$. Dann kann das Minimum nicht in $[y, b]$ sein.
- 4. Fall: $f(y) \geq f(b)$. Dann ist das Minimum in $[y, b]$.

Alle anderen Fälle widersprechen der Definition einer unimodalen Funktion.

Der goldene Schnitt

Wir wollen jetzt durch die rekursive Anwendung dieser Beobachtung einen Algorithmus zur Approximation des Punktes konstruieren, an dem das Minimum angenommen wird. Dafür stellen wir zunächst einige Forderungen an den Algorithmus:

1.) Im ersten Schritt sollen zwei Funktionswertauswertungen, in allen weiteren Schritten nur eine weitere Funktionswertauswertung im Restintervall verwendet werden.
2.) Die Punkte x und y sind stets symmetrisch im Restintervall zu wählen.
3.) Der Reduktionsfaktor σ für die Intervalllänge sei konstant, $\sigma \in (0.5, 1)$. Das heißt, der Quotient der Länge des neuen Intervalls und der Länge des vorherigen Intervalls ist $\sigma = const.$

Durch die symmetrische Wahl von x, y im Restintervall wird der Reduktionsfaktor unabhängig von der konkreten Funktion f. Verlangt man einen konstanten Reduktionsfaktor in allen Iterationen, so ist dieser eindeutig bestimmt.

Um diesen von f unabhängigen Reduktionsfaktor zu bestimmen, genügt es eine streng monoton wachsende Funktion $f : [0, 1] \to \mathbb{R}$ zu betrachten. O.B.d.A. sei $0 < x < y < 1$. Daraus folgt $y = \sigma, x = 1 - \sigma$.

Im ersten Schritt wird das Intervall $[0, 1]$ auf das Restintervall $[0, y]$ reduziert. In diesem Restintervall wird dann ein Punkt z symmetrisch zu x gewählt. Je nach Wahl von y gilt $z < x$ oder $z \geq x$. Diese Fälle entsprechen der Wahl von σ aus unterschiedlichen Intervallen.

1.Fall: $\sigma < 2/3$. Dann gilt $x = 1 - \sigma > 1/3$ und damit

$$z = y - x = \sigma - (1 - \sigma) < 1/3 < x.$$

Konstante Reduktion bedeutet nun

$$\frac{x}{y} = \frac{y}{1}, \quad \Longrightarrow \quad 0 = y^2 - x = \sigma^2 + \sigma - 1.$$

Die Lösung dieser quadratischen Gleichung ist

$$\sigma = \frac{1}{2}\left(\sqrt{5} - 1\right) \approx 0.618 < \frac{2}{3}.$$

2. Fall: $\sigma \geq 2/3$. Wird analog zum ersten Fall behandelt. Man erhält keine Lösung. *Übungsaufgabe*

Der gefundene Reduktionsfaktor ist also unter den obigen Annahmen der einzig mögliche und er legt den Algorithmus fest.

Algorithm 2.2 Goldener Schnitt.

1. *Initialisierung.*

$$\texttt{i} := 0;\ \texttt{a}_0 := \texttt{a};\ \texttt{b}_0 := \texttt{b};$$
$$\texttt{x}_\texttt{i} := \texttt{a} + (1 - \sigma)(\texttt{b}_\texttt{i} - \texttt{a}_\texttt{i});\ \texttt{y}_\texttt{i} := \texttt{a} + \sigma(\texttt{b}_\texttt{i} - \texttt{a}_\texttt{i});$$
$$\texttt{fx} := \texttt{f}(\texttt{x}_\texttt{i});\ \texttt{fy} := \texttt{f}(\texttt{y}_\texttt{i});$$

2. *Iteration.*

   Falls $\mathtt{fx} < \mathtt{fy}$, dann

   $\mathtt{a_{i+1}} := \mathtt{a_i};\ \mathtt{b_{i+1}} := \mathtt{y_i};$

   $\mathtt{x_{i+1}} := \mathtt{a_{i+1}} + (1-\sigma)(\mathtt{b_{i+1}} - \mathtt{a_{i+1}});\ \mathtt{y_{i+1}} := \mathtt{x_i};$

   $\mathtt{fy} := \mathtt{fx};\ \mathtt{fx} := \mathtt{f}(\mathtt{x_{i+1}});$

   sonst

   $\mathtt{a_{i+1}} := \mathtt{x_i};\ \mathtt{b_{i+1}} := \mathtt{b_i};$

   $\mathtt{x_{i+1}} := \mathtt{y_i};\ \mathtt{y_{i+1}} := \mathtt{b_{i+1}} - (1-\sigma)(\mathtt{b_{i+1}} - \mathtt{a_{i+1}});$

   $\mathtt{fx} := \mathtt{fy};\ \mathtt{fy} := \mathtt{f}(\mathtt{y_{i+1}});$

   $\mathtt{i} := \mathtt{i} + 1;$

3. *Abbruch.*

   Falls $(\mathtt{b_i} - \mathtt{a_i})/2 > \varepsilon$, dann

   gehe zu 2.

   sonst

   $\tilde{\mathtt{x}} = (\mathtt{a_i} + \mathtt{b_i})/2;$

   stop

In der Praxis führt man erst den Vergleich für den Abbruch durch und berechnet dann den neuen Funktionswert. Das spart im letzten Schritt eine Funktionswertberechnung. Diese Einsparung kann sich akkumulieren, falls man das Verfahren im Rahmen eines komplexen Problems oft aufruft.

Algorithmus 2.2 bricht ab, sobald der Funktionswert, für den das Minimum angenommen wird, in einem Intervall der Länge ε eingeschachtelt ist. Da nach $n+1$ Funktionswertauswertungen die Länge des Restintervalls $\sigma^n(b-a)$ ist, kann man n in Abhängigkeit von ε a priori abschätzen.

Die Fibonacci–Suche

Wir wollen jetzt auf die Konstanz der Reduktion der Intervalllänge verzichten und untersuchen, ob es Modifikationen von Algorithmus 2.2 gibt, die bei gleicher Anzahl von Funktionswertauswertungen ein kleineres Restintervall liefern. Sei L_n die maximale Länge eines Intervalls, welches man mittels n Funktionswertauswertungen in L_n auf $L_1 = 1$ reduzieren kann. Seien $x < y$ die beiden Stützstellen in L_n. Durch jede dieser Stützstellen wird L_n in zwei Teilintervalle zerlegt. Betrachtet man die Stützstelle x, so enthält das linke Teilintervall höchstens $n-2$ Stützstellen (alle außer x und y) und das rechte Teilintervall höchstens $n-1$ der Stützstellen (alle außer x). Entsprechend sind die Längen dieser Teilintervalle höchstens L_{n-2} beziehungsweise L_{n-1}. Damit gilt

$$L_n \leq L_{n-1} + L_{n-2}, \quad n = 2, 3, \ldots. \tag{2.2}$$

Da mindestens zwei Stützstellen für eine Intervallreduktion nötig sind, gilt $L_0 = L_1 = 1$. Mit der obigen Abschätzung hat man das größtmögliche Intervall L_n, falls eine Lösung der Ungleichung (2.2) diese sogar als Gleichung erfüllt.

Die Lösung der Gleichungen

$$F_0 = F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

ist bekannt. Es sind die sogenannten Fibonacci–Zahlen. Die Darstellung dieser Zahlen in geschlossener Form ist

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

Für den Algorithmus der Fibonacci–Suche setzen wir noch $F_{-2} := 1, F_{-1} := 0$.

Bei der Fibonacci–Suche wird die geforderte Länge des Intervalls, in dem $\tilde{x}$ eingeschachtelt werden soll, vorgegeben. Die Iterationspunkte bei einem beliebigen Startintervall $[a, b]$ ergeben sich durch lineare Transformation mit dem Faktor

$$h = \frac{b-a}{F_n}. \tag{2.3}$$

Die Länge des Restintervalls in der 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: $x_{n-2} = y_{n-2} = a_{n-2} + h$. Da für den Punkt $\tilde{x}$, in dem das Minimum angenommen wird, gilt $\tilde{x} \in [a_{n-2}, b_{n-2}]$, folgt somit $|\tilde{x} - x_{n-2}| \leq h$. Damit ist durch (2.3) bei vorgegebenem h auch die Anzahl der Iterationen n gegeben.

Algorithm 2.3 Fibonacci–Suche.

1. *Initialisierung.*

   gebe h vor, bestimme n, berechne die Fibonacci--Zahlen $F_0, \ldots, F_n$

   $\mathtt{i} := 0;\ \mathtt{a}_0 := \mathtt{a};\ \mathtt{b}_0 := \mathtt{b};\ \mathtt{h} := (\mathtt{b} - \mathtt{a})/\mathtt{F}_\mathtt{n};$

   $\mathtt{x}_0 := \mathtt{a}_0 + \mathtt{F}_{\mathtt{n}-2}\mathtt{h};\ \mathtt{y}_0 := \mathtt{a}_0 + \mathtt{F}_{\mathtt{n}-1}\mathtt{h};$

   $\mathtt{fx} := \mathtt{f}(\mathtt{x}_0);\ \mathtt{fy} := \mathtt{f}(\mathtt{y}_0);$

2. *Iteration.*

   Falls $\mathtt{fx} < \mathtt{fy}$, dann

   $\mathtt{a}_{i+1} := \mathtt{a}_i;\ \mathtt{b}_{i+1} := \mathtt{y}_i;$

   $\mathtt{x}_{i+1} := \mathtt{a}_{i+1} + \mathtt{F}_{\mathtt{n}-\mathtt{i}-3}\mathtt{h};\ \mathtt{y}_{i+1} := \mathtt{x}_i;$

   $\mathtt{fy} := \mathtt{fx};\ \mathtt{fx} := \mathtt{f}(\mathtt{x}_{i+1});$

   sonst

   $\mathtt{a}_{i+1} := \mathtt{x}_i;\ \mathtt{b}_{i+1} := \mathtt{b}_i;$

   $\mathtt{x}_{i+1} := \mathtt{y}_i;\ \mathtt{y}_{i+1} := \mathtt{b}_{i+1} - \mathtt{F}_{\mathtt{n}-\mathtt{i}-3}\mathtt{h};$

   $\mathtt{fx} := \mathtt{fy};\ \mathtt{fy} := \mathtt{f}(\mathtt{y}_{i+1});$

   $\mathtt{i} := \mathtt{i} + 1;$

3. *Abbruch.*

   Falls $\mathtt{i} < \mathtt{n} - 2$, dann

   gehe zu 2.

   sonst

   $\tilde{\mathtt{x}} = \mathtt{x}_\mathtt{i};$

   stop

Analog wie beim Goldenen Schnitt kann man im letzten Schritt eine Funktionswertberechnung sparen.

Beispiel 2.4 *Fibonacci–Suche.* Wir betrachten die Funktion $f(x) = \sin(x-2)$ auf $[a, b] = [0, 2]$. Auf diesem Intervall ist die Funktion f unimodal und sie nimmt ihr Minimum in $\tilde{x} = 2 - \pi/2 \approx 0.4292037$ an. Wir wollen die Fibonacci–Suche mit $n = 6$ durchführen. Die Fibonacci–Zahlen $F_0, \ldots, F_6$ sind $1, 1, 2, 3, 5, 8, 13$. Daraus folgt, dass man ein Restintervall der Länge

$$h = \frac{2}{13} \approx 0.1538462$$

findet.

Die im Verfahren auftretenden Stützstellen und Intervallgrenzen sind alle von der Form $a + kh$ mit $k \in \{0, 1, 2, 3, 5, 8, 13\}$. Für eine Stützstelle $t \in [a, b]$ nennt man die entsprechende ganze Zahl $k(t) := (t - a)/h$ den Fibonacci–Index von t. Beim Start gilt $k(a_0) = 0$, $k(x_0) = F_{n-2} = F_4 = 5$, $k(y_0) = F_{n-1}$, $k(b_0) = F_n$.

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

i	$k(a_i)$	x_i	$k(x_i)$	y_i	$k(y_i)$	$k(b_i)$	$f(x_i)$	$f(y_i)$
0	0	0.7692308	5	1.2307692	8	13	- 0.9427456	- 0.6955828
1	0	0.4615385	3	0.7692308	5	8	- 0.9994773	- 0.9427456
2	0	0.3076923	2	0.4615385	3	5	- 0.9926266	- 0.9994773
3	2	0.4615385	3	0.6153846	4	5	- 0.9994773	- 0.9827183
4	2	0.4615385	3	0.4615385	3	4	- 0.9994773	- 0.9994773

Mit $x_4 = 0 + 3h$ bricht das Verfahren ab und für das Minimum von f gilt

$$\tilde{x} \in [0.4615385 - h, 0.4615385 + h].$$

□

Vergleich von Goldenem Schnitt und Fibonacci–Suche

Aus

$$1 - \sigma = \sigma^2 \tag{2.4}$$

folgt induktiv (mit $F_{-2} = 1, F_{-1} = 0$)

$$\sigma^n = (-1)^n \left(F_{n-2} - F_{n-1}\sigma\right).$$

Übungsaufgabe Aus $F_0 = F_1 = 1$ und $F_1 = 1 < 1/\sigma < 2 = F_2$ erhält man

$$F_2 < \frac{\sigma^2 + \sigma}{\sigma^2} = 1 + \frac{1}{\sigma} < F_3.$$

Mit Hilfe von (2.4) folgt $F_2 < 1/\sigma^2 < F_3$. Induktiv erhält man für $n > 1$ die Abschätzung

$$\frac{1}{F_n} > \sigma^n > \frac{1}{F_{n+1}} \quad \text{und} \quad \lim_{n\to\infty} \frac{F_n}{F_{n+1}} = \sigma.$$

Unter der Annahme, dass der rechte Grenzwert existiert, kann man diesen mit Hilfe von (2.4) berechnen. *Übungsaufgabe* Asymptotisch ist der Goldene Schnitt bei gleichem Aufwand demnach von kaum geringerer Genauigkeit. Auf der anderen Seite muss bei Fibonacci–Suche die gewünschte Genauigkeit a priori festgelegt werden, um n zu bestimmen.

Beide Verfahren reduzieren die Intervalllänge linear, dass heisst es gilt

$$\frac{|b_{i+1} - a_{i+1}|}{|b_i - a_i|} \leq \lambda \quad \text{mit } \lambda \in (0, 1).$$

Dabei ist der Reduktionsfaktor λ beim Goldenen Schnitt durch σ gegeben und bei der Fibonacci–Suche durch eine Schranke für F_{n-1}/F_n. Soll das n–te Intervall kleiner als ein gegebenes ε sein, so erhält man aus

$$\varepsilon \geq \lambda^n(b - a) = \lambda^n(b_0 - a_0)$$

die Abschätzung

$$n \geq \log\left(\frac{\varepsilon}{b - a}\right) / \log(\lambda).$$

2.2 Differenzierbare Funktionen

Sei $f : \mathbb{R}^n \to \mathbb{R}$ mit $f \in C^1(\mathbb{R}^n)$ gegeben. Die notwendige Bedingung für ein lokales Minimum im Punkt $\tilde{\mathbf{x}} \in \mathbb{R}^n$ ist

$$\nabla f(\tilde{\mathbf{x}}) = \mathbf{0}.$$

Zur Berechnung von möglichen Werten für $\tilde{\mathbf{x}}$ hat man damit ein nichtlineares Gleichungssystem zu lösen. Verfahren zur Lösung nichtlinearer Gleichungssysteme werden in der Vorlesung *Praktische Mathematik* behandelt und deshalb soll hier nur die Namen einiger Verfahren genannt werden:

- Bisektion falls $n = 1$,
- Gradientenverfahren (Verfahren des steilsten Abstiegs),
- Fixpunktiteration, wobei die nachfolgenden Verfahren Spezialfälle sind,
- Newton–Verfahren,
- vereinfachtes und Quasi–Newton–Verfahren.