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

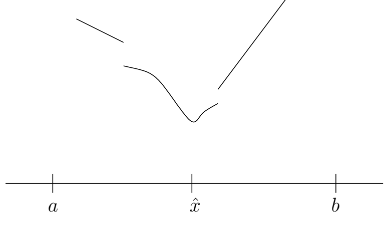
Die Lösung nichtlinearer Optimierungsaufgaben in einer Dimension tritt als Teilproblem in vielen Verfahren zur Lösung nichtlinearer Optimierungsprobleme in höheren Dimensionen auf.

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 (einwellig), 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 $\sigma = const$.

Diese Forderungen besitzen gewisse Ähnlichkeiten zu den Eigenschaften des Bisektions–Verfahrens zur Berechnung von Nullstellen. Auch da wird die Nullstelle in einem Intervall eingeschachtelt, man hat am Anfang zwei Funktionswerte und später pro Schritt einen zu berechnen und der Reduktionsfaktor ist konstant $\sigma = 0.5$.

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. Damit ist $\hat{x} = 0$. O.B.d.A. sei $0 < x < y < 1$. Wegen der verlangten Symmetrie 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. Damit ist gesichert, dass im nächsten Schritt der Punkt x einer der neuen Stützstellen ist, man den Funktionswert $f(x)$ noch einmal verwenden kann und nur den Funktionswert $f(z)$ neu berechnen muss. 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

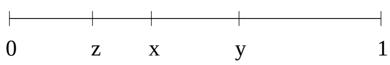
$$z = y - x = \sigma - (1 - \sigma) < 1/3 < x.$$

Bei der ersten Gleichung wurde die Symmetrie im Restintervall verwendet. 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.

Algorithmus 2.2 Goldener Schnitt.

1. *Initialisierung.*

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

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

Man nimmt $\tilde{x}$ als Approximation an $\hat{x}$. □

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 2ε eingeschachtelt ist. Da $\tilde{x}$ der Mittelpunkt des Intervalls ist, gilt für den Fehlern $|\hat{x} - \tilde{x}| \leq \varepsilon$. 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 des Gesamtprozesses (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 Differenzengleichung

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

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 $\hat{x}$ 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 $\hat{x}$, in dem das Minimum angenommen wird, gilt $\hat{x} \in [a_{n-2}, b_{n-2}]$, folgt somit $|\hat{x} - x_{n-2}| \leq h$. Damit ist durch (2.3) bei vorgegebenem h auch die Anzahl der Iterationen n gegeben.

Algorithmus 2.3 Fibonacci–Suche.

1. *Initialisierung.*

   gebe h vor, bestimme n, berechne die Fibonacci-Zahlen $F_0, \ldots, F_n$
   $\mathtt{i} := 0$; $\mathtt{a_0} := \mathtt{a}$; $\mathtt{b_0} := \mathtt{b}$; $\mathtt{h} := (\mathtt{b} - \mathtt{a})/\mathtt{F_n}$;
   $\mathtt{x_0} := \mathtt{a_0} + \mathtt{F_{n-2}h}$; $\mathtt{y_0} := \mathtt{a_0} + \mathtt{F_{n-1}h}$;
   $\mathtt{fx} := \mathtt{f(x_0)}$; $\mathtt{fy} := \mathtt{f(y_0)}$;

2. *Iteration.*

   Falls $\mathtt{fx} < \mathtt{fy}$, dann
      $\mathtt{a_{i+1}} := \mathtt{a_i}$; $\mathtt{b_{i+1}} := \mathtt{y_i}$;
      $\mathtt{x_{i+1}} := \mathtt{a_{i+1}} + \mathtt{F_{n-i-3}h}$; $\mathtt{y_{i+1}} := \mathtt{x_i}$;
      $\mathtt{fy} := \mathtt{fx}$; $\mathtt{fx} := \mathtt{f(x_{i+1})}$;
   sonst
      $\mathtt{a_{i+1}} := \mathtt{x_i}$; $\mathtt{b_{i+1}} := \mathtt{b_i}$;
      $\mathtt{x_{i+1}} := \mathtt{y_i}$; $\mathtt{y_{i+1}} := \mathtt{b_{i+1}} - \mathtt{F_{n-i-3}h}$;
      $\mathtt{fx} := \mathtt{fy}$; $\mathtt{fy} := \mathtt{f(y_{i+1})}$;
   $\mathtt{i} := \mathtt{i} + 1$;

3. *Abbruch.*

   Falls $\mathtt{i} < \mathtt{n} - 2$, dann
      gehe zu 2.
   sonst
      $\tilde{\mathtt{x}} := \mathtt{x_i}$;
      stop

Man nimmt $\tilde{x}$ als Approximation an $\hat{x}$. □

Analog wie beim Goldenen Schnitt kann man in der letzten Iteration 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 $\hat{x} = 2 - \pi/2 \approx 0.4292037$ 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

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

□

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 erhält man die gleiche Intervallreduktion und der Goldene Schnitt ist 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

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

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 $\hat{\mathbf{x}} \in \mathbb{R}^n$ ist

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

Zur Berechnung von möglichen Werten für $\hat{\mathbf{x}}$ 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.