

1. Übung

Ein bekanntes Beispiel in der Optimierung ist das Rosenbrock-Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) \quad \text{mit} \quad f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2. \quad (1)$$

Abbildung 1 zeigt das Profil und die Höhenlinien der Funktion, die auch als Bananenfunktion bezeichnet wird. Das Rosenbrock-Problem soll als Beispiel verwendet werden, um die Konvergenzeigenschaften der in Vorlesung behandelten Verfahren numerisch mit Hilfe von MATLAB zu untersuchen.

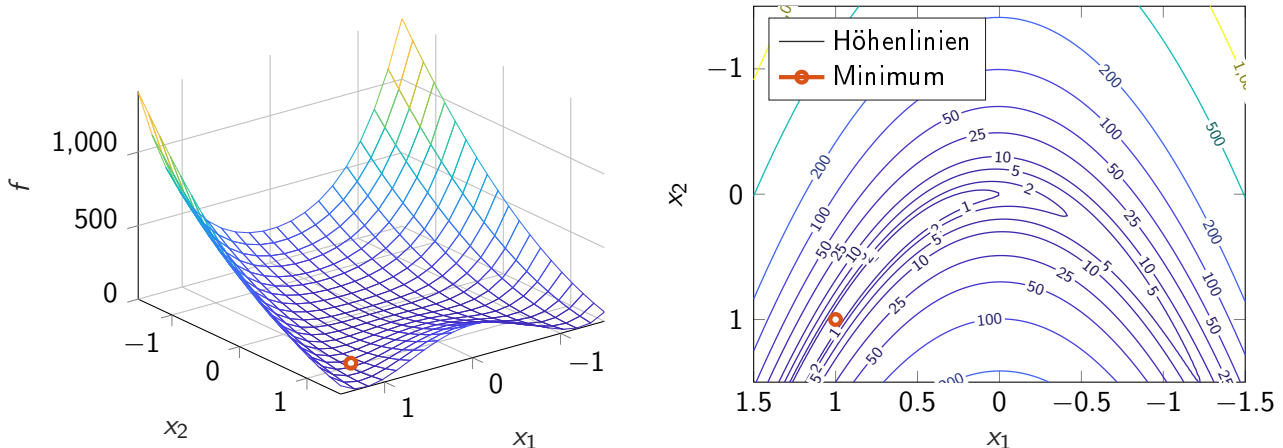


Abbildung 1: Profil und Höhenlinien von Rosenbrock's Bananenfunktion.

Aufgabe 1.1

Verifizieren Sie, dass der Punkt $\mathbf{x}^* = [1, 1]^T$ ein eindeutiges Minimum darstellt. Sind die Funktion $f(\mathbf{x})$ und das Optimierungsproblem (1) konvex?

Zur Lösung von unbeschränkten Optimierungsproblemen stellt die Optimization Toolbox von MATLAB die folgenden Funktionen zur Verfügung

- `fminunc`: Liniensuche: Gradientenverfahren, Quasi-Newton-Verfahren
Methode der Vertrauensbereiche: Newton-Verfahren
- `fminsearch`: Simplex-Verfahren nach Nelder und Mead.

Eine empfehlenswerte Alternative ist die frei zugängliche MATLAB-Funktion `minFunc`¹, die eine große Auswahl an Liniensuchverfahren bietet. Tabelle 1 zeigt einige Vergleichsdaten für die numerische Lösung des Rosenbrock-Problems (ausgehend vom Startwert $\mathbf{x}^0 = [-1, -1]^T$), die mit Hilfe von `fminunc`, `fminsearch` und `minFunc` berechnet wurden. Abbildung 2 stellt zusätzlich die Iterationsverläufe für die Verfahren dar, die unter `fminunc` und `fminsearch` implementiert sind.

Beim Gradientenverfahren fällt die langsame Konvergenz auf, da auch nach dem Erreichen der maximalen Anzahl an Funktionsauswertungen (200) das Minimum noch nicht erreicht ist. In Abbildung 3 ist der

¹<http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>

| Verfahren | Iter. | $f(\mathbf{x}^*)$ | $\nabla f(\mathbf{x}^*)$ | Funktionsaufrufe: | | |
|-----------------------------|-------|----------------------|--------------------------|-------------------|------------------------|--------------------------|
| | | | | $f(\mathbf{x})$ | $\nabla f(\mathbf{x})$ | $\nabla^2 f(\mathbf{x})$ |
| LS: Gradientenverfahren | 57 | 0.1232 | 1.1978 | 200 | 200 | - |
| LS: Konj.-Grad.-Verf. | 31 | $4.1 \cdot 10^{-17}$ | $6.9 \cdot 10^{-9}$ | 91 | 90 | - |
| LS: Newton-Verfahren | 21 | $3.5 \cdot 10^{-17}$ | $3.5 \cdot 10^{-8}$ | 29 | 28 | 24 |
| LS: Quasi-Newton (BFGS) | 23 | $5.4 \cdot 10^{-12}$ | $9.2 \cdot 10^{-6}$ | 29 | 29 | - |
| VB: Newton-Verfahren | 25 | $2.2 \cdot 10^{-18}$ | $2.1 \cdot 10^{-8}$ | 26 | 26 | 26 |
| Simplex-Verf. (Nelder-Mead) | 67 | $5.3 \cdot 10^{-10}$ | - | 125 | - | - |

Tabelle 1: Vergleich der numerischen Verfahren für das Rosenbrock-Problem (LS=Linienuche, VB=Methode der Vertrauensbereiche).

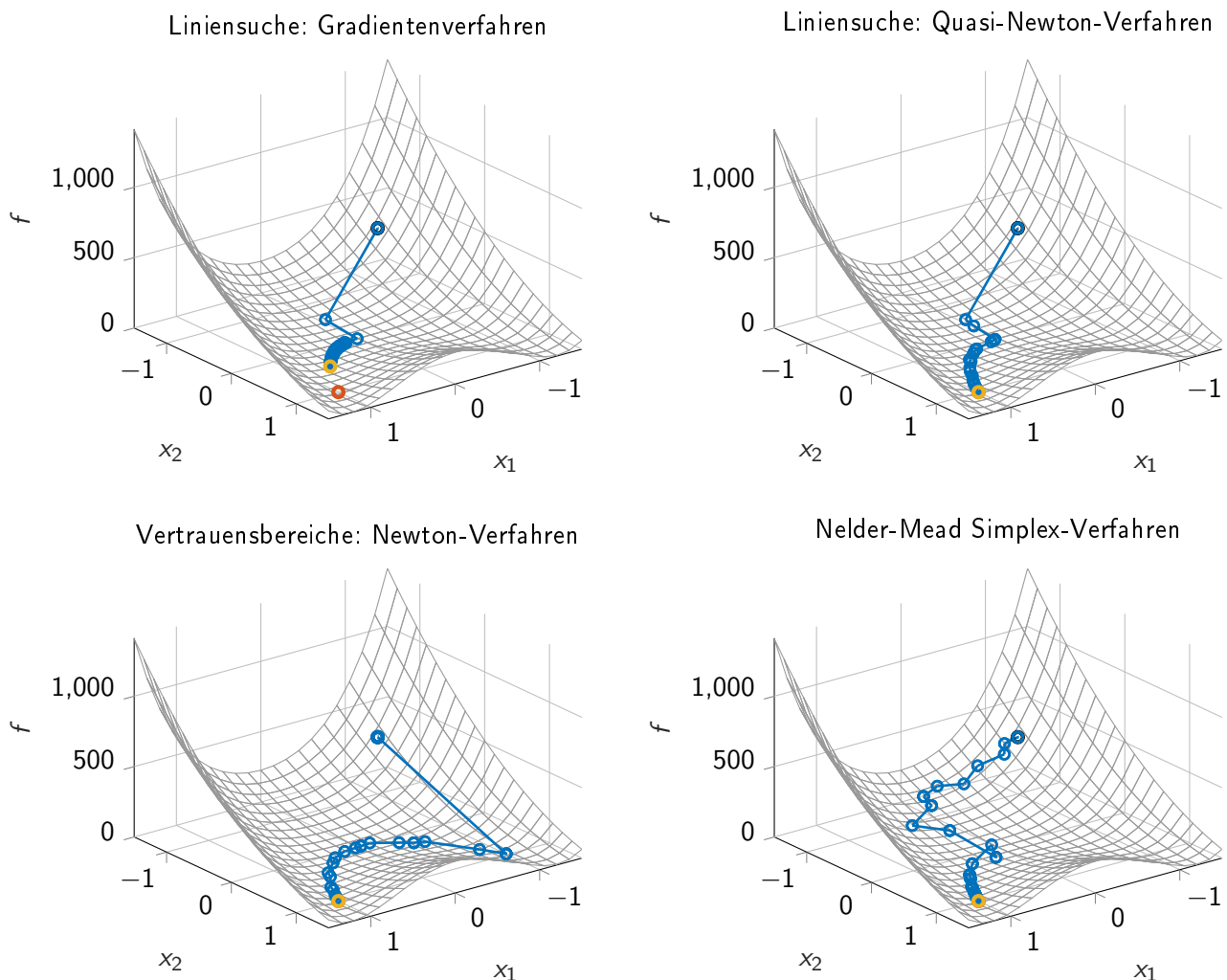


Abbildung 2: Rosenbrock's Bananenfunktion: Vergleich der numerischen Verfahren mit fminunc und fminsearch.

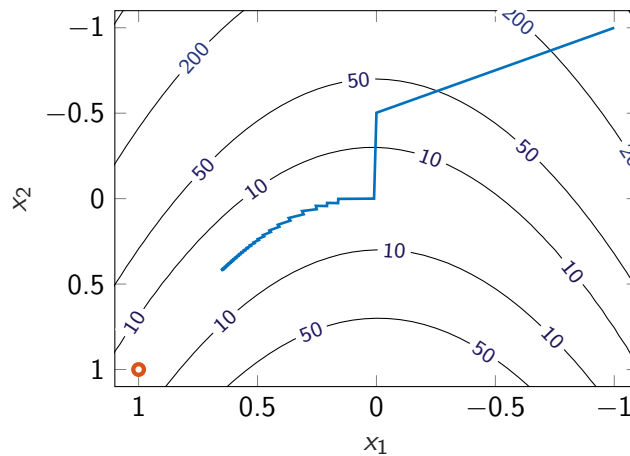


Abbildung 3: Darstellung der Iterationen des Gradientenverfahrens.

Iterationsverlauf des Gradientenverfahrens noch einmal detailliert über die Höhenlinien der Rosenbrock-Funktion (1) dargestellt. Erkennbar ist, dass sich das Gradientenverfahren an der maximalen Abstiegsrichtung orientiert, die senkrecht zu der jeweiligen Höhenlinie verläuft. In Richtung des Minimums werden die Iterationsschritte immer kleiner. Die niedrige Konvergenzgeschwindigkeit des Gradientenverfahrens wurde bereits in der Vorlesung veranschaulicht und soll in der folgenden Aufgabe näher untersucht werden:

Aufgabe 1.2

Berechnen Sie für das Minimum $\mathbf{x}^* = [1, 1]^T$ des Rosenbrock-Problems (1) die Konvergenzrate des Gradientenverfahrens gemäß Satz 3.4.

Das Konvergenzverhalten des Quasi-Newton-Verfahrens in Abbildung 2 ist wesentlich besser als beim Gradientenverfahren. Das Newton-Verfahren (Methode der Vertrauensbereiche) in Abbildung 2 startet zunächst in die "falsche" Richtung, was durch die quadratische Approximation (3.32) am Startpunkt $\mathbf{x}^0 = [-1, 1]^T$ zu erklären ist, dessen Minimum in der Nähe von $\mathbf{x} \approx [-1, 1]^T$ liegt. Allerdings sind die einzelnen Schritte entlang des Tales der Rosenbrock-Funktion deutlich größer, da das Newton-Verfahren die Hesse-Matrix explizit verwendet und nicht auf eine Approximation angewiesen ist wie im Fall des Quasi-Newton-Verfahrens.

Zusätzlich sind in Tabelle 1 und Abbildung 2 die Ergebnisse für das Simplex-Verfahren von Nelder-Mead angegeben, die mit der MATLAB-Funktion `fminsearch` erzielt wurden. Allerdings bietet die Grafikausgabe unter `fminsearch` nicht die Möglichkeit, die einzelnen Simplexe darzustellen. In der nächsten Aufgabe soll das Simplex-Verfahren deshalb näher untersucht werden, um einen Eindruck von den Simplex-Operationen und der Robustheit des Verfahrens zu erhalten.

Aufgabe 1.3

Schreiben Sie eine MATLAB-Funktion, die das Rosenbrock-Problem (1) mit Hilfe des Simplex-Verfahrens nach Nelder-Mead numerisch löst (siehe Abschnitt 3.4.2 im Skript). Konstruieren Sie den ersten Simplex mit den Eckpunkten

$$\mathbf{x}^{0,1} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \mathbf{x}^{0,2} = \mathbf{x}^{0,1} + s \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}^{0,3} = \mathbf{x}^{0,1} + s \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1)$$

in Abhängigkeit der Seitenlänge $s = 1$. Verwenden Sie für die Abbruchbedingung in Tabelle 3.5 die Schranke $\epsilon = 10^{-9}$ und vergleichen Sie Ihre Ergebnisse mit den `fminsearch`-Ergebnissen in Tabelle 1. Stellen Sie die Simplexe aus den einzelnen Iterationen grafisch dar. Untersuchen Sie die Robustheit und das Konvergenzverhalten des Simplex-Verfahrens für verschiedene Seitenlängen s des Startsimplex und unterschiedliche Startpunkte $\mathbf{x}^{0,1}$.

Aufgabe 1.4

Schreiben Sie eine MATLAB-Funktion, die das Rosenbrock-Problem (1) mit Hilfe des Newton-Verfahrens (Linien-
suche) löst, siehe Abschnitt 3.3.4. Verwenden Sie zur Bestimmung der Schrittweite α^k das Backtracking-
Verfahren mit dem Startwert $\alpha = 1$ und der Armijo-Bedingung (3.21)

$$f(\mathbf{x}^k + \alpha^k \mathbf{s}^k) \leq f(\mathbf{x}^k) + c_1 \alpha^k \nabla f(\mathbf{x}^k)^T \mathbf{s}^k \quad (1)$$

als Abbruchkriterium für die Schrittweitenbestimmung. Vergleichen Sie die Konvergenzergebnisse mit den
Werten in Tabelle 1. Stellen Sie die einzelnen Iterationen grafisch dar und variieren Sie die Startpunkte \mathbf{x}^0 .

Aufgabe 1.5

Das ungedämpfte Newton-Verfahren mit der natürlichen Schrittweite $\alpha = 1$ kann divergieren, wenn der
Startwert \mathbf{x}^0 nicht nah genug am optimalen Punkt \mathbf{x}^* liegt. Zeigen Sie diese Problematik für das Beispiel-
problem

$$\min_{x \in \mathbb{R}} f(x) = \log(e^x + e^{-x}) \quad (1)$$

mit den Startwerten $x^0 = 1.1$ und $x^0 = 1.0$. Stellen Sie $f(x)$ und die ersten Iterationen grafisch dar. Verwen-
den Sie zum Vergleich das gedämpfte Newton-Verfahren mit der Backtracking-Liniensuche und vergleichen
Sie die Ergebnisse.

Aufgabe 1.6 Zusatzaufgabe

Das Simplex-Verfahren ist ein heuristisches Verfahren, für das die Konvergenz zu einem stationären Punkt
in der Regel nicht garantiert werden kann. Ein bekanntes Beispiel, das dieses Problem aufzeigt, ist das
McKinnon-Problem

$$\min_{\mathbf{x} \in \mathbb{R}^2} f(\mathbf{x}) \quad f(\mathbf{x}) = \begin{cases} \theta \phi |x_1|^\tau + x_2 + x_2^2 & \text{falls } x_1 \leq 0 \\ \theta x_1^\tau + x_2 + x_2^2 & \text{falls } x_1 > 0 \end{cases} \quad (1)$$

Die Funktion $f(\mathbf{x})$ hat ein globales Minimum an der Stelle $\mathbf{x}^* = [0, -0.5]^T$ mit $f(\mathbf{x}^*) = -0.25$. Im
Folgenden werden die Parameter $\theta = 6$, $\phi = 60$ und $\tau = 2$ betrachtet. Stellen Sie die McKinnon-Funktion
und ihre Höhenlinien grafisch dar (MATLAB-Befehle `contour` und `mesh` bzw. `surf`). Verwenden Sie das
implementierte Simplex-Verfahren, um das Problem zu lösen. Verwenden Sie dafür den Startsimplex

$$\mathbf{x}^{0,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{x}^{0,2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{x}^{0,3} = \frac{1}{8} \begin{bmatrix} 1 + \sqrt{33} \\ 1 - \sqrt{33} \end{bmatrix} \quad (2)$$

Zeigen Sie grafisch anhand der einzelnen Simplexiterationen, dass das Verfahren nicht zum globalen Mini-
mum, sondern zu einem nicht-stationären Punkt konvergiert.