Universität Passau
Lehrstuhl für Mathematik
mit Schwerpunkt Digitale Bildverarbeitung

Bachelorarbeit

Implizite Funktionen zur Berechnung von Höhenlinien

Autorin Christina Brandmeier
Matrikelnummer 105080
Prüfer Prof. Dr. Thomas Sauer

20. Dezember 2024

Inhaltsverzeichnis

1 Ein Blick hinter die Karten: Die Mathematik der Höhenlinien 1

2 Theorie impliziter Funktionen 3
2.1 Definition und grundlegende Eigenschaften 3
2.2 Beispiele für implizite Funktionen 4
2.3 Der Hauptsatz über implizite Funktionen 7
2.4 Partielle Ableitungen und Gradienten 11

3 Theoretische Grundlagen der Höhenlinien 14
3.1 Höhenlinien und deren Anwendung 14
3.2 Darstellung von Höhenlinien in der Topografie 15
3.3 Verfahren zur Generierung von Höhenlinien 16

4 Mathematische Modellierung von Höhenlinien 18
4.1 Die implizite Darstellungen 18
4.2 Verfahren zur Berechnung von Höhenlinien 20
4.3 Beispielhafte Berechnung 21

5 Implementierung in Octave 23
5.1 Die Software Octave 23
5.2 Implementierung von impliziten Funktionen in Octave 23
5.3 Visualisierung von Höhenlinien 27
5.4 Herausforderungen 30

6 Fazit und Ausblick 31

Anhang A Quellcode in Octave 32
A.1 Gesamter Quellcode 32
A.2 Weiteres Beispiel 34

Literatur **38**

Eidesstattliche Erklärung **41**

1 Ein Blick hinter die Karten: Die Mathematik der Höhenlinien

Rund jeder Zehnte in Deutschland verbringt seine Freizeit am liebsten mit Wandern.[1] Bei der Planung einer Wanderroute im Internet erhält man sofort eine Vielzahl an Informationen: Dauer, Distanz, durchschnittliche Geschwindigkeit und die zu bewältigenden Höhenmeter. Während die ersten Angaben noch leicht nachvollziehbar erscheinen, stellt sich die Frage: Wie werden die Höhenmeter berechnet, und was bedeuten sie für uns als Wanderer?

Zur Berechnung und Darstellung von Höhenlinien auf Karten existieren verschiedene mathematische Verfahren. Viele dieser Verfahren leiden jedoch unter numerischen Instabilitäten oder sind nicht in der Lage, komplexe Geländestrukturen präzise zu erfassen. In dieser Arbeit soll der Fokus auf die Anwendung impliziter Funktionen zur Berechnung von Höhenlinien gelegt werden, da diese ein stabiles und genaues Verfahren zur Modellierung von Geländeprofilen darstellen können. Das Hauptziel dieser Arbeit besteht darin, die theoretischen Grundlagen impliziter Funktionen auf die Berechnung von Höhenlinien anzuwenden und deren Vorteile gegenüber klassischen Verfahren herauszuarbeiten. Hierzu wird eine umfassende Analyse der mathematischen Modellierung durchgeführt und eine Implementierung in der Software Octave entwickelt. Ziel ist es, ein Verfahren zu entwerfen, das sowohl numerische Stabilität bietet als auch komplexe Geländeformen präzise darstellt. Darüber hinaus wird die praktische Anwendbarkeit der Methoden durch Fallstudien und Beispielprojekte demonstriert. Die Arbeit ist in sechs Kapitel gegliedert. Zunächst werden die theoretischen Grundlagen impliziter Funktionen und Höhenlinien erläutert. Darauf aufbauend wird die mathematische Modellierung von Höhenlinien untersucht und eine eigenen Implementierung in Octave beschrieben. Anschließend folgen Fallstudien, in denen die entwickelten Methoden angewendet und

[1] Dies zeigte eine Statistik der Allensbacher Markt- und Werbeträgeranalyse 2023.[Dep24]

evaluiert werden. Abschließend werden die Ergebnisse diskutiert und mögliche Ansätze für zukünftige Weiterentwicklungen aufgezeigt.

2 Theorie impliziter Funktionen

Um einen Algorithmus zur Berechnung und Modellierung von Höhenlinien mit Hilfe von impliziten Funktionen implementieren zu können, werden wir zuerst die mathematischen Grundlagen impliziter Funktionen betrachtet. Damit soll ein grundlegendes Verständnis für implizite Funktionen geschaffen werden.

2.1 Definition und grundlegende Eigenschaften

Eine implizite Funktion ist eine Funktion, die über eine Gleichung, wie

$$f(x) - x^2 - 1 = 0$$

gegeben ist. Dabei schreibt man y statt $f(x)$ und erhält somit

$$y - x^2 - 1 = 0.$$

Der Unterschied zu gewöhnlichen Funktionen, wie $f(x) = x^2 + 1$, ist, dass man die Funktionswerte zunächst nicht *explizit* bestimmen kann. Das heißt, bei der *expliziten* Funktion $f(x) = x^2 + 1$ kann man sofort für den Wert $x = 3$ den Funktionswert $f(3) = 3^2 + 1 = 10$ berechnen. Bei *impliziten* Funktionen hingegen muss man dazu zuerst die Gleichung umstellen:

$$y - 3^2 - 1 = 0 \Rightarrow y = 3^2 + 1 = 10.$$

Implizit bedeutet also, dass der Funktionswert nicht direkt dasteht, sondern die Beziehung zwischen den Variablen nur durch eine Gleichung gegeben ist. Die linke Seite bezeichnet man dann mit $F(x, y)$.

Definition 2.1.1 (implizite Funktion)**.** Eine IMPLIZITE FUNKTION gibt Beziehungen zwischen Variablen x und y nur auf indirekte Weise an, nämlich als

$$F(x,y)=0, \qquad x\in U\subseteq\mathbb{R}^m, \quad y\in V\subseteq\mathbb{R}^n, \quad F:\mathbb{R}^{n+m}\to\mathbb{R}^d.$$

[Sau24, S.95]

In einer solchen Gleichung kann es natürlich beliebig viele Variablen geben. In der Regel steht dabei die Variable, welche den Funktionswert beschreibt, am Ende.

Beispiel 2.1.2 (Implizite Funktion mit drei Variablen)**.** Man betrachte die Gleichung

$$0=F(x,y,z)=x^2y+xy^2-zxy \quad (\text{mit } z=f(x,y)).$$

Wenn wir den Punkt $(x,y)=(1,2)$ gegeben haben, folgt $z=f(1,2)=3$, weil nur dann die Gleichung erfüllt ist.

Es können aber nicht nur mehr Variablen, sondern auch mehrere Funktionen in solch einer Gleichung beschrieben werden. Gegeben sei dafür ein beliebiges Gleichungssystem der Form

$$\begin{aligned} g_j(x)=h_j(x), \quad j=1,\dots,d \quad &\Leftrightarrow \quad G(x)=H(x) \\ &\Leftrightarrow \quad (G-H)(x)=0 \end{aligned}$$

Ziel ist es nun, einige Variablen in Abhängigkeit der anderen Variablen auszudrücken. Dafür wird eine Funktion $f:\mathbb{R}^n\to\mathbb{R}^m$ gesucht, so dass $F(x,f(x))=0, \quad x\in U$, erfüllt ist. [Sau24, S.95]

2.2 Beispiele für implizite Funktionen

In dem vorhergehenden Abschnitt 2.1 wurden bereits zwei *implizite Funktionen* vorgestellt. Doch bei der impliziten Darstellung von Funktionen und beim Lösen dieser können zwei Probleme auftreten. Diese werden nun anhand der folgenden zwei Beispiele näher untersucht.

Beispiel 2.2.1 (Einheitskreis)**.** Man betrachte den Einheitskreis $\mathbb{S}^2$ mit der Gleichung $x^2 + y^2 = 1$. Dieser wird durch den impliziten Ausdruck

$$F(x, y) = x^2 + y^2 - 1 = 0$$

beschrieben. Wir wollen nun versuchen, diese Gleichung nach y aufzulösen, um die Funktion $y = f(x)$ ableiten zu können, die uns für jedes x den zugehörigen y-Wert auf dem Einheitskreis liefert. Durch Umstellen erhalten wir wegen der Fallunterscheidung eine positive und eine negative Lösung für y. Diese Lösungen können wir durch die beiden reellen Funktionen

$$f_1 : [-1, 1] \to [0, 1], \quad f_1(x) = \sqrt{1 - x^2}$$
$$f_2 : [-1, 1] \to [-1, 0], \quad f_2(x) = -\sqrt{1 - x^2}$$

beschreiben, wobei die Funktion f_1 für die obere Hälfte des Einheitskreises $\mathbb{S}^2$ darstellt und f_2 die untere Hälfte des Kreises.
Es gibt also nicht eine einzelne Funktion $y = f(x)$, die uns an jedem Punkt des Kreises den richtigen Wert für y gibt, sondern zwei mögliche Funktionen. Diese **Uneindeutigkeit** macht es unmöglich, jedem x-Wert eindeutig einen y-Wert auf $\mathbb{S}^2$ zuzuordnen.
Dies können wir umgehen, wenn wir nicht den ganzen Kreis auf einmal betrachten, sondern uns auf die Umgebung eines gegebenen Punktes (x_0, y_0) auf $\mathbb{S}^2$ beschränken. Für $y_0 \neq 0$ können wir den Punkt eindeutig der Funktion f_1 oder f_2 zuordnen. In diesem Fall kann man in der Umgebung dieses Punktes die Gleichung nach y stets eindeutig lösen - außer bei den beiden Punkten $(-1, 0)$ und $(1, 0)$.
Was macht diese beiden Punkte besonders?
Geometrisch betrachtet, macht der implizite Ausdruck $F(x, y)$ bei $y_0 = 0$ eine "Kehrtwende". Das bedeutet, dass die beschriebene Kurve an diesen beiden Punkten ihre Richtung entlang der x-Achse ändert. Dies führt dazu, dass es zwei gültige Funktionsgraphen gibt. Somit markieren diese Punkte den Übergang zwischen den beiden Kreishälften, an denen es keine eindeutige Lösung gibt.
Die beiden kritischen Punkte können jedoch aus der Sichtweise der Variable x auch als Extrema aufgefasst werden. Ein Extremum wird durch das Nullwerden der partiellen Ableitung nach der Zielvariablen berechnet. Wenn wir uns die partielle Ableitung unseres impliziten Ausdrucks nach y ansehen, verschwindet diese an den beiden kritischen Punkten:

$$\frac{\partial F(x, y)}{\partial y}(\pm 1, 0) = [2y]_{x=\pm 1, y=0} = 0.$$

Für $y_0 \neq 0$ gilt dies jedoch nicht, wodurch eine eindeutige Auflösung nach y möglich bleibt.
Das Beispiel lässt vermuten, dass in dem Fall $\partial F/\partial y = 0$ ein impliziter Ausdruck der Form $F(x, y) = 0$ ein Extremum haben kann und eine eindeutige Auflösung nach der Zielvariablen y nicht möglich ist.

[Are+22, S. 899]

In dem Abschnitt 2.1 wurden bereits lineare Gleichungssysteme angeschnitten und das Ziel, Variablen in Abhängigkeit der anderen Variablen auszudrücken. Doch wann kann ein impliziten Ausdruck überhaupt nach einer bestimmten Variablen aufgelöst werden?

Beispiel 2.2.2 (Lineares Gleichungssystem)**.** Wir betrachten ein lineares Gleichungssystem mit $n + m$ Unbekannten und n gegebenen Gleichungen

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n + a_{1n+1}x_{n+1} + \cdots + a_{1n+m}x_{n+m} &= b_1 \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n + a_{nn+1}x_{n+1} + \cdots + a_{nn+m}x_{n+m} &= b_n, \end{aligned}$$

welches wir auf die Form

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 - a_{1n+1}x_{n+1} - \cdots - a_{1n+m}x_{n+m} \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n - a_{nn+1}x_{n+1} - \cdots - a_{nn+m}x_{n+m} \end{aligned}$$

umschreiben, um diese nach x_1 bis x_n aufzulösen. Dies ist genau dann möglich, wenn die entsprechende Koeffizientendeterminante

$$\det A_{n\times n} = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix}$$

nicht verschwindet. [1] Da die Lösung des Gleichungssystems alle restlichen Variablen x_{n+1} bis x_{n+m} beinhaltet, besteht die Lösung aus durch $x_k = \chi_k(x_{n+1}, \ldots, x_{n+m}), \quad k = 1, \ldots, n$, definierte Funktionen.

[1] Dies ist die Voraussetzung dafür, dass die Matrix $A_{n\times n}$ invertierbar und damit das Gleichungssystem $A_{n\times n} \cdot x = b - B \cdot y$ eindeutig lösbar ist. [Are+22, S. 606]

Diese Erkenntnisse aus linearen Systemen lassen sich direkt auf nichtlineare Gleichungssysteme übertragen. In einer kleinen Umgebung eines Punktes kann jede nichtlineare, differenzierbare Abbildung $\mathbb{R}^n \to \mathbb{R}^n$ durch eine lineare Abbildung approximiert werden. Dabei spielt die Jacobi-Matrix in nichtlinearen Systemen eine ähnliche Rolle wie die Koeffizientenmatrix in linearen, weshalb die Koeffizienten die Elemente der Jacobi-Matrix sind, $a_{ij} = \frac{\partial f_i}{\partial x_j}$.
Analog zu linearen Gleichungssystemen erwarten wir, dass eine Auflösung von nichtlinearen, differenzierbaren Systemen genau dann möglich ist, wenn die entsprechende Jacobi-Determinante nicht verschwindet. [Are+22, S. 899 f.]

2.3 Der Hauptsatz über implizite Funktionen

Da nun geklärt ist, was *implizite Funktionen* sind und wie sie durch implizite Gleichungen dargestellt werden, stehen zwei zentrale Fragen im Raum:
Wann kann man eine implizite Funktion nach geeigneten Variablen auflösen und wann ist der implizit gegebene Funktionswert eindeutig?
Diese beiden Fragen bilden den Kern der Theorie impliziter Funktionen. Mit den in Abschnitt 2.2 herausgearbeiteten Ansätzen können wir nun den folgenden Satz über implizite Funktionen einführen, der diese Fragen beantwortet.

Satz 2.3.1 (Satz über implizite Funktionen)**.** *Seien $U \subset \mathbb{R}^k$ und $V \subset \mathbb{R}^m$ offene Teilmengen und*

$$F : U \times V \to \mathbb{R}^m, (x, y) \mapsto F(x, y),$$

eine stetig differenzierbare Abbildung. Sei $(a, b) \in U \times V$ ein Punkt mit $F(a, b) = 0$.
Die $m \times m$-Matrix

$$\frac{\partial F}{\partial y} := \frac{\partial (F_1, \ldots, F_m)}{\partial (y_1, \ldots, y_m)} := \begin{pmatrix} \frac{\partial F_1}{\partial y_1} & \cdots & \frac{\partial F_1}{\partial y_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial y_1} & \cdots & \frac{\partial F_m}{\partial y_m} \end{pmatrix}$$

sei im Punkt (a, b) invertierbar. Dann gibt es eine offene Umgebung $U' \subset U$ von a, eine Umgebung $V' \subset V$ von b sowie eine stetig differenzierbare Abbildung $g : U' \to V' \subset \mathbb{R}^m$ mit $g(a) = b$, so dass

$$F(x, g(x)) = 0 \quad \textit{für alle } x \in U'.$$

Ist $(x, y) \in U' \times V'$ ein Punkt mit $F(x, y) = 0$, so folgt $y = g(x)$. [For17, S. 104]

Der Satz über implizite Funktionen besagt also, dass unter der Bedingung, dass die partielle Ableitung von F nach y an einem Punkt (a, b) nicht verschwindet, eine eindeutige, lokal definierte Funktion $y = g(x)$ existiert, welche die implizite Gleichung löst.

Bemerkung 2.3.2 (Implizite Funktionen)**.**

1. Eine grundlegende Voraussetzung von Satz 2.3.1 ist, dass die Anzahl der Komponenten gleich der Dimension von y ist, da nur so die Jacobi-Matrix $\frac{\partial f}{\partial y}$ quadratisch und damit invertierbar wäre. [Sau24, S. 99]

2. Zerlegt man die Funktion F in die Komponenten, erhält man ein System von m Gleichungen. Dieses Gleichungssystem kann nach den Variablen $y_1, \ldots, y_m$ aufgelöst werden. Die Anzahl der Komponenten von F bestimmt somit, wie viele Variablen man eliminieren und durch die verbleibenden Variablen ausdrücken kann. Dieses Vorgehen ist vergleichbar mit der schrittweisen Lösung von linearen Gleichungssystemen, wobei hier auch nichtlineare Gleichungen behandelt werden können. [Sau24, S. 99]

3. Für die Gültigkeit des Satzes ist wesentlich, dass U und V gegebenenfalls verkleinert werden, da der Satz über implizite Funktionen nur **lokale** Lösungen liefern kann. Global, d.h. in ganz $U \times V$, könnte es zu einem gegebenen x mehrere y-Werte (oder auch gar keinen) geben, die der Gleichung $F(x, y) = 0$ genügen. Dieser Satz stellt somit sicher, dass die Funktion $y = g(x)$ in einer kleinen Umgebung existieren, aber nicht notwendigerweise in ganz $\mathbb{R}^k \times \mathbb{R}^m$. Dies wird in Abb. 2.1 verdeutlicht, wo eine kleinere Umgebung gewählt werden muss, um eine eindeutige Lösung für y bestimmen zu können. [For17, S. 104 f.]

Lemma 2.3.3. *Ist F stetig differenzierbar und $\frac{\partial F}{\partial y}$ an einer Stelle (a, b) invertierbar, so ist sie das auch in einer Umgebung von (a, b). [For17, S. 105]*

Beweis. F ist in $U \times V$ stetig differenzierbar, daher sind auch alle partiellen Ableitungen von F in $U \times V$ stetig. Daraus folgt, dass die Determinante

$$\delta : U \times V \to \mathbb{R}, \quad \delta(x, y) := \det\left(\frac{\partial F}{\partial y}(x, y)\right)$$

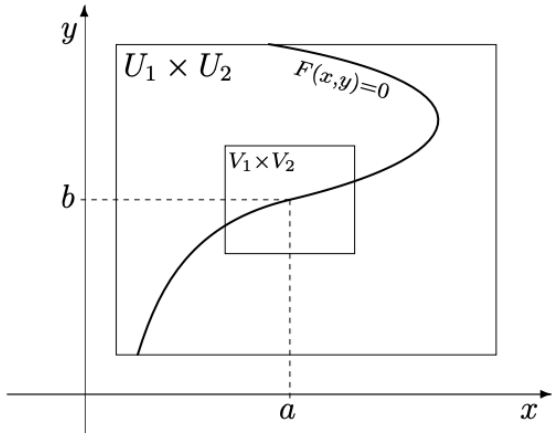


Abbildung 2.1: Verkleinerung der Umgebungen U und V

in $U \times V$ stetig ist, da sie ein Polynom in den stetigen Funktionen $\frac{\partial F_i}{\partial y_i}$ ist. Wegen der Invertierbarkeit von $\frac{\partial F}{\partial y}(a,b)$ gilt $\delta(a,b) \neq 0$. Aufgrund der Stetigkeit von $\delta(x,y)$ existiert eine Umgebung $U' \times V' \subset U \times V$ um (a,b), sodass $\delta(x,y) \neq 0$ für alle $(x,y) \in U' \times V'$. Damit ist die Jacobimatrix $\frac{\partial F}{\partial y}(x,y)$ für alle $(x,y) \in U' \times V'$ invertierbar. □

Der Beweis für den Satz 2.3.1 ist recht umfangreich und geht über den Rahmen dieser Arbeit hinaus. Daher wird im Folgenden nur eine Beweisskizze präsentiert, die einen ausreichenden Überblick über die wesentlichen Schritte und Ideen vermittelt, ohne dabei ins Detail zu gehen. Der vollständige Beweis des Satzes über die implizite Funktion ist in [For17, S. 106 ff.] zu finden.

Beweis. (von Satz 2.3.1)
1. Vorbereitungen:
O.B.d.A. sei $(a,b) = (0,0) \in \mathbb{R}^k \times \mathbb{R}^m$, so dass $F(0,0) = 0$. Nach Voraussetzung ist $\frac{\partial F}{\partial y}(0,0)$ invertierbar. Mit Lemma 2.3.3. folgt die Invertierbarkeit von $\frac{\partial F}{\partial y}(x,y)$ für (x,y) nahe (0,0). Wir definieren nun die Iterationsabbildung $G : U \times V \to \mathbb{R}^m$ durch

$$G_x(y) := y - B^{-1}F(x,y) \in \mathbb{R}^m \text{ mit fixiertem } x \text{ und } B := \frac{\partial F}{\partial y}(0,0).$$

2. Existenz und Eindeutigkeit:
Sei nun y ein Fixpunkt von G_x, dann folgt für alle $y \in V'$:

$$G_x(y) = y \Leftrightarrow 0 = B^{-1}F(x,y) \Leftrightarrow F(x,y) = 0.$$

Um diesen Fixpunkt von G_x zu finden, zeigt man zuerst durch Umformungen, dass $G_x(y)$ eine wohldefinierte Kontraktion auf der abgeschlossenen Kugel V' ist. Anschließend kann der Banachscher Fixpunktsatz angewendet werden und es gilt:

$$\forall\, x \in U' \quad \exists!\, y = g(x) \in V' : \qquad G_x(y) = y \Leftrightarrow F(x, g(x)) = 0.$$

3. Stetige Differenzierbarkeit der Lösung:
Um zu zeigen, dass die Abbildung $g : U' \to V'$ stetig ist, betrachten wir die stetigen und beschränkten Abbildungen $\varphi : U' \to \mathbb{R}^m$ auf dem Banachraum $\mathcal{C}_b(U', \mathbb{R}^m)$ und wenden wieder den Banachschen Fixpunktsatz an. Damit zeigen wir, dass die Abbildung $G_x(y)$ auch in Bezug auf Stetigkeit eine Kontraktion ist.
Es bleibt zu zeigen, dass die Abbildung g stetig differenzierbar ist. Dafür genügt der Beweis der Differenzierbarkeit von g im Nullpunkt $0 \in U'$ durchzuführen. Für die anderen Punkte geht der Beweis analog. Wir setzen

$$A := \frac{\partial F}{\partial x}(0,0), \qquad B := \frac{\partial F}{\partial y}(0,0)$$

Da F in $(0,0)$ differenzierbar ist, kann die Taylorentwicklung

$$F(x,y) = Ax + By + \varphi(x,y) \text{ mit } \varphi(x,y) = o(\|(x,y)\|)$$

betrachtet werden. Für alle $x \in U'$ gilt nach Teil 2 $F(x, g(x)) = 0$, d.h.

$$g(x) = -B^{-1}Ax - B^{-1}\varphi(x, g(x)).$$

Für die Differenzierbarkeit von g in $(0,0)$ ist somit nur zu beweisen, dass

$$-B^{-1}\varphi(x, g(x)) = o(\|x\|).$$

Wenn dies gezeigt ist, folgt analog, dass g in jedem Punkt differenzierbar ist. Zusätzlich ist die Ableitung von g nach x

$$\frac{\partial g}{\partial x}(x) = -\frac{\partial F}{\partial y}(x, g(x)) \cdot \frac{\partial F}{\partial x}(x)$$

stetig.
Somit ist die Abbildung g stetig differenzierbar. □

Eine zentrale Voraussetzung des Satzes über implizite Funktionen ist die Invertierbarkeit der $m \times m-$ Matrix $\frac{\partial F}{\partial y}$ im Punkt (a, b). Das folgende Lemma liefert hierfür eine notwendige und hinreichende Bedingung.

Lemma 2.3.4 (Invertierbarkeitskriterium)**.** *Eine quadratische Matrix ist genau dann invertierbar, wenn ihre Determinante nicht verschwindet. [Heu04, S. 306]*

Um die Überprüfung dieser Voraussetzung für mögliche Abbildungen F systematisch zu ermöglichen, werden die folgenden Begriffe eingeführt.

Definition 2.3.5 (Reguläre und singuläre Punkte)**.** Eine Stelle $x \in U \subseteq \mathbb{R}^n$ heißt SINGULÄRER PUNKT einer an x differenzierbaren Funktion $f : U \to \mathbb{R}^n$, wenn $\det f'(x) = 0$ ist. Sonst heißt x ein REGULÄRER PUNKT. [Sau24, S. 103]

2.4 Partielle Ableitungen und Gradienten

Für die theoretische Grundlage für den weiteren Verlauf dieser Arbeit wollen wir uns noch genauer mit der Ableitung beschäftigen, die im Hauptsatz über implizite Funktionen eine wichtige Voraussetzung ist.

Definition 2.4.1 (Partielle Ableitung)**.** Sei $U \subset \mathbb{R}^k$ eine offene Teilmenge und $f : U \to \mathbb{R}$ eine reelle Funktion. f heißt im Punkt $x \in U$ *partiell differenzierbar* in der i-ten Koordinatenrichtung, falls der Limes

$$\frac{\partial f}{\partial x_i}(x) = \lim_{h \to 0} \frac{f(x + he_i) - f(x)}{h}$$

existiert. Dabei ist $e_i \in \mathbb{R}^k$ der i-te Einheitsvektor,

$$e_i = (0, \ldots, 0, 1, 0, \ldots, 0) \qquad \text{mit 1 an der } i\text{-ten Stelle}$$

und für den Limes $h \to 0$ hat man sich auf solchen $h \in \mathbb{R}$ zu beschränken, für die $h \neq 0$ und $x + he_i \in U$.

$\frac{\partial f}{\partial x_i}(x)$ heißt i-te partielle Ableitung von f in x. [For17, S. 62]

Definition 2.4.2 (Partielle Differenzierbarkeit)**.** Sei $U \subset \mathbb{R}^k$ offen. Eine Funktion $f : U \to \mathbb{R}$ heißt *partiell differenzierbar*, falls $\frac{\partial f}{\partial x_i}(x)$ für alle $x \in U$ und $i = 1, \ldots, n$ existiert. f heißt *stetig partiell differenzierbar*, falls diese zusätzlich stetig sind. [For17, S. 63]

Bemerkung 2.4.3. Wenn die Definition 2.4.2 für alle Variablen der Funktion f gilt, dann ist f *(stetig) differenzierbar.*

Bei impliziten Funktionen kennen wir jedoch die Funktion nicht direkt. Am Ende des Beweises von Satz 2.3.1 wird bereits eine Form der Ableitung einer impliziten Funktion herausgearbeitet, welche wir hier nochmal genauer aufführen.

Lemma 2.4.4 (Ableitung implizit definierter Funktion)**.** *Ist $F : U \times V \to \mathbb{R}^m$ differenzierbar, $U \subseteq \mathbb{R}^n, V \subseteq \mathbb{R}^m$ und ist $(x, y) \in U \times V$ mit $F(x, y) = 0$ und $detF'_y(x, y) \neq 0$, dann gilt für die implizit definierte Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$*

$$f'(x) = -F'_y(x, f(x))^{-1} F'_x(x, f(x)).$$

[Sau24, S. 102]

Definition 2.4.5 (Gradient)**.** Sei $U \subset \mathbb{R}^k$ offen und $f : U \to \mathbb{R}$ eine partiell differenzierbare Funktion. Dann heißt der Vektor

$$\nabla f = \text{grad } f(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_k}(x) \right)$$

[For17, S. 66]

Bemerkung 2.4.6. Der Gradient gibt die Richtung des stärksten Anstiegs von f an und sein Betrag $|\nabla f(x)|$ ist gerade dieser stärkste Anstieg, wobei man immer von einem festen Punkt x ausgeht. [Heu04, S. 274] Außerdem ist der Gradient einer Funktion $f(x, y)$ orthogonal zu allen Kurven der Form $f(x, y) = c$. Diese Eigenschaft wird in späteren Abschnitten bei der Analyse von Höhenlinien verwendet.

Beispiel 2.4.7. Betrachten wir nun die in Beispiel 2.2.1 definierte Funktion $F(x, y) = x^2 + y^2 - 1$, so ergibt sich

$$\nabla F(x, y) = (2x, 2y).$$

D.h. für $x, y \neq 0$ verschwindet die Determinante nicht und mit dem Invertierbarkeitskriterium sind die Voraussetzungen für den Satz über implizite Funktionen erfüllt. Die Funktion ist also nach x oder nach y auflösbar. Für $x = 0$ ist $x(y)$ jedoch, wie in Beispiel 2.2.1 ausführlich dargestellt, nicht eindeutig, weshalb an dieser Stelle nach y aufzulösen ist. Analog für $y = 0$. [Sau24, S. 104]

3 Theoretische Grundlagen der Höhenlinien

Nachdem bereits geklärt wurde, was implizite Funktionen sind, kommt die Frage auf, wo man überhaupt auf implizit gegebene Funktionen stößt. Ein markantes Beispiel der Anwendung des Satzes über implizite Funktionen sind *Höhenlinien*, welche in diesem Kapitel genauer betrachtet werden.

3.1 Höhenlinien und deren Anwendung

Zu Beginn dieser Arbeit wurde bereits eine Lieblingsfreizeitbeschäftigung der Deutschen thematisiert - das Wandern. Bei Wanderungen ist es jedoch essenziell zu wissen, wie der Weg verläuft. Besonders für gesundheitlich Eingeschränkte stellen sich Fragen wie: Geht es ständig bergauf? Wie steil ist das Gelände? Wie groß sind die Höhenunterschiede zwischen dem Start- und dem Zielpunkt? Die Antwort darauf liefern Höhenlinien - ein unverzichtbares Hilfsmittel in topografischen Karten.

Eine *Höhenlinie* (oder auch *Niveaulinie* genannt) verbindet alle Punkte einer Landschaft, die auf gleicher Höhe über dem Meeresspiegel (NN = Normal-Null) liegen. Eine solche imaginäre Linie läuft wie ein horizontaler Rundweg um einen Hügel oder Berg.

Die typische Verwendung von Höhenlinien ist in topografischen Karten zu finden. Dabei dienen die Höhenlinien als Planungsgrundlage für vielfältige Anwendungen wie Straßenplanung, Umweltschutz, Energiewirtschaft und Stadtplanung [Lana]. Aber auch als Hilfestellung für Wanderer in Wanderkarten sind Höhenlinien essenziell. In Abb. 3.1 ist ein beispielhafter Ausschnitt einer topografischen Karte zu sehen, welcher ein beliebtes Wandergebiet östlich des Tegernsees in Bayern zeigt. Neben den klassischen Zeichen wie

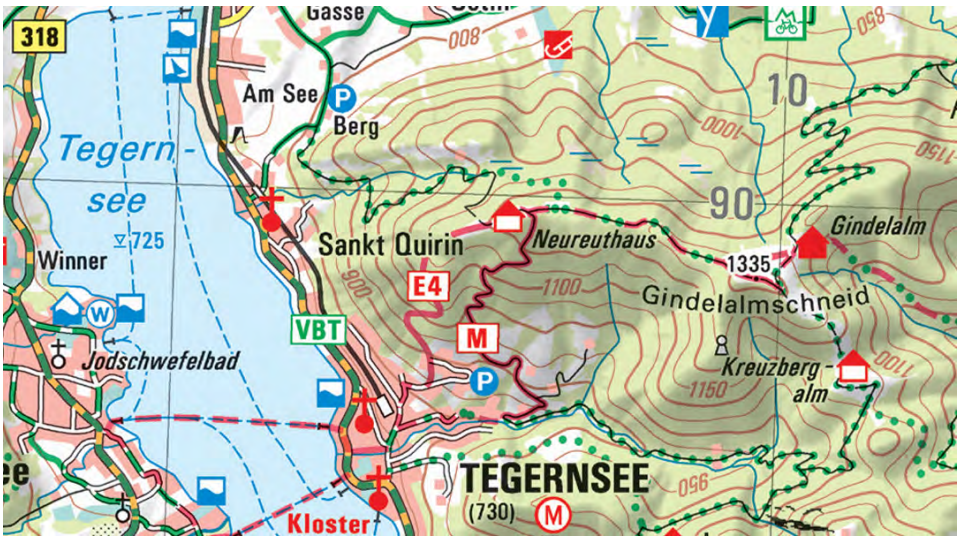


Abbildung 3.1: Ausschnitt der topografischen Karte östlich des Tegernsees, Bayern

Siedlungen, Verkehr, Vegetation und Gewässer ist auch das Relief, darunter Höhenlinien, abgebildet.

Neben der Anwendung in topografischen Karten lassen sich mit Hilfe der Höhenlinien eine Höhenschichtkarte und damit ein Höhenprofil für einen bestimmten Ausschnitt erstellen.

3.2 Darstellung von Höhenlinien in der Topografie

Um ein Gelände in einer topografischen Karte möglichst anschaulich darstellen zu können, wird bildlich gesehen das Gelände in einer bestimmten Höhe horizontal "geschnitten". Jeweils zwei benachbarte Schnittlinien haben immer einen gleichen Höhenabstand zueinander, die sogenannte "Äquidistanz". Diese ist abhängig vom Landschaftstyp. Für Hügelland und Mittelgebirge sind zehn Meter Äquidistanz üblich. Für höhere Erderhebungen werden auch 50 m Äquidistanz verwendet, wie in Abbildung 3.1. Um die Höhe der Schnittlinien später wieder ablesen zu können, werden diese so beschriftet, dass man sie in Richtung des steigenden Geländes lesen kann. Dabei wird die Höhe in Metern angegeben und bezieht sich auf den mittleren Meeresspiegel in Amsterdam ("Höhen über NN"). Alle Schnittlinien des Geländes ergeben zusammen das *Höhenlinienbild*. Somit können Form und Steigung eines Geländes durch die Anordnung der Höhenlinien sichtbar gemacht werden. Je enger die Linien zusammenliegen, desto steiler ist das Gelände, je

weiter sie auseinanderliegen, desto flacher ist es. Bilden die Höhenlinien eine eng gewundene, geschlossene Linie, so zeigt dies eine Kuppe oder einen Berggipfel an. Die höchste Stelle ist zudem mit einem Punkt und einer zugehörigen Höhenangabe versehen. Um im flachen Gelände die Reliefform besser darstellen zu können, wird das Höhenlinienbild um gestrichelte *Hilfshöhenlinien* ergänzt. Höhenlinien können zusätzlich unterschiedliche Farbgebungen haben. Im Normalfall sind diese braun. In felsiger Landschaft hingegen sind die Höhenlinien schwarz, im Gewässer und in Gletschergebieten blau eingezeichnet.

Mit Hilfe dieser Eigenschaften beschreiben Höhenlinien die Geländebeschaffenheit und bieten damit wichtige Anhaltspunkte unter anderem fürs Wandern.

[Ver]

3.3 Verfahren zur Generierung von Höhenlinien

Die Erstellung von Höhenlinien basiert auf der präzisen Messung von Höhenunterschieden und der Erfassung von Geländestrukturen. Hierfür stehen sowohl klassische als auch moderne Verfahren zur Verfügung. Im Folgenden wird eine Auswahl an Verfahren näher erläutert.
Eine bewährte Methode ist das *Nivellieren*, ein präzises, jedoch zeit- und arbeitsintensives Verfahren. Dabei wird der Höhenunterschied zwischen zwei Geländepunkten ermittelt, indem mit zwei Nivellierlatten ein Lot auf den Punkten errichtet und mit einem Nivelliergerät, beispielsweise einer Kanalwaage, eine horizontale Ziellinie zwischen den beiden Latten hergestellt wird. An den Skalen der Nivellierlatten können die Höhenwerte der Ziellinie abgelesen werden. Der Höhenunterschied ergibt sich schließlich als Differenz der Ablesewerte. [Kah93, S. 373] Aufgrund der hohen Messgenauigkeit eignet sich das Nivellieren insbesondere für kleinere Flächen oder Anwendungen mit besonders hohen Genauigkeitsanforderungen.
Das wichtigste Verfahren zur Erstellung von Höhenlinien ist die *Interpolation*. Dabei werden zunächst charakteristische Geländepunkte hinsichtlich ihrer Lage und Höhe erfasst und in ein digitales Geländemodell übertragen. Die Höhenlinien selbst werden anschließend mithilfe von Algorithmen interpoliert. Das bedeutet, dass zwischen zwei Punkten mit bekannten Höhen die Linie konstanter Höhe berechnet wird. Dieser Prozess erfolgt in der Regel rechnergestützt. [Kah93, S. 536] Um ein konsistentes und präzises Höhenbild

zu erhalten, müssen jedoch alle verfügbaren Daten berücksichtigt und kleinere, genauigkeitsbedingte Widersprüche ausgeglichen werden [Kah93, S. 736 f.].
Ein weiteres Verfahren stellt das Flugzeug-getragene Laserscanning, auch bekannt als *LiDAR-Technologie* (engl. Light Detection and Ranging) dar. Dabei wird ein Laserstrahl ausgesendet, dessen Laufzeit nach der Reflexion am Boden gemessen wird. Aus der gemessenen Laufzeit lassen sich Entfernungen und Höhen präzise berechnen. LiDAR ermöglicht die Erfassung einer sehr dichten Punktwolke und eignet sich daher besonders für die Erstellung digitaler Höhenmodelle. [GB11, S. 54 ff.] Diese Methode zeichnet sich durch ihre hohe Genauigkeit aus, ist im Vergleich zu traditionellen Verfahren wie dem Nivellieren jedoch deutlich weniger arbeitsintensiv. Aufgrund dieser Vorteile wird LiDAR häufig zur Erstellung topografischer Karten eingesetzt. Allerdings sind die Kosten für die Lasertechnologie sowie den Flugzeugbetrieb vergleichsweise hoch.
Eine ähnliche Methode ist die *Satellitenphotogrammetrie*, die sich durch ihre weltweite Verfügbarkeit und wirtschaftliche Effizienz auszeichnet. Dabei werden Höheninformationen aus Luft- oder Satellitenbildern gewonnen [Hei16, S. 5]. Landschaften werden aus verschiedenen Blickwinkeln aufgenommen, und mithilfe einer Stereoauswertung, bei der sich überlappende Bildpaare analysiert werden, lässt sich die Höhe jedes Punktes berechnen. Die gewonnenen Daten können anschließend in ein digitales Geländemodell überführt werden, aus dem sich Höhenlinien ableiten lassen. [Hei16, S. 14] Dieses Verfahren ermöglicht eine schnelle und großflächige Geländekartierung.

Zusammenfassend basieren Höhenlinien auf präzisen Höhenmessungen und der anschließenden Interpolation der Daten. Klassische Verfahren wie das Nivellieren zeichnen sich durch ihre hohe Genauigkeit aus, sind jedoch arbeitsintensiv. Moderne Technologien wie LiDAR und Satellitenphotogrammetrie ermöglichen hingegen eine schnelle und großflächige Erstellung von digitalen Höhenmodellen und topografischen Karten, erfordern jedoch höhere finanzielle und technische Ressourcen.

Viele dieser Verfahren, wie die Interpolation, basieren direkt oder indirekt auf den in Kapitel 2 kennengelernten impliziten Funktionen. Deshalb wollen wir uns im weiteren Verlauf damit auseinandersetzen, wie Höhenlinien mit impliziten Funktionen in Verbindung gebracht werden können.

4 Mathematische Modellierung von Höhenlinien

In den beiden vorherigen Kapiteln wurden die theoretischen Grundlagen für implizite Funktionen und für Höhenlagen gelegt. Nun wollen wir diese vereinen und zum Kern dieser Arbeit gelangen - der Berechnung von Höhenlinien durch implizite Funktionen.

Höhenlinien und implizite Funktionen stehen in einem engen Zusammenhang, da beide Konzepte Flächen oder Objekte im Raum durch bestimmte Bedingungen beschreiben.

4.1 Die implizite Darstellungen

Wir wissen bereits, dass Höhenlinien Punkte gleicher Höhe z auf einer Oberfläche $f(x, y)$ in einer zweidimensionalen Darstellung repräsentieren. Dabei kann ein Gelände visuell in ein dreidimensionales Koordinatensystem übertragen werden. Die Höhenlinien bilden dann eine Projektion des Schnitts der Oberfläche auf die x-y-Ebene, wobei alle Punkte einer Linie den gleichen z-Wert haben. Geometrisch betrachtet bilden die Höhenlinien in der x-y-Ebene Kurven, die durch eine konstante Höhe definiert sind.

Diese Eigenschaften lassen sich durch implizite Funktionen formal und präzise beschreiben. Eine implizite Funktion beschreibt eine Menge von Punkten (x, y), die einen konstanten Wert c annehmen:

$$F(x, y) := f(x, y) - c = 0 \qquad \Leftrightarrow \qquad f(x, y) = c.$$

Die Menge dieser Punkte bildet zusammen die Höhenlinie der Höhe c. Damit erhalten wir eine mathematische Definition für Höhenlinien.

Definition 4.1.1 (Höhenlinien)**.** Sei $U \subset \mathbb{R}^2$ eine offene Menge und

$$f : U \to \mathbb{R}, \qquad (x, y) \mapsto f(x, y),$$

eine stetig differenzierbare Funktion. Dann können *Höhenlinien* als Schar $N_f(c), c \in \mathbb{R}$ ihrer Niveaukurven dargestellt werden. Dabei ist

$$N_f(c) := \{(x, y) \in U : f(x, y) = c\} \in \mathbb{R}^2$$

die Menge aller Punkte, in denen f die Höhe c annimmt. [For17, S. 62, 109]

Da wir Höhenlinien nun als implizite Funktion angeben können, wissen wir aus Abschnitt 2.4, dass implizite Funktionen nach Satz 2.3.1 lokal aufgelöst werden können, wenn der grad f nicht verschwindet. Nach [For17, S. 86] steht der Gradient von f auf $N_f(c)$ orthogonal. Diesen direkten Zusammenhang zwischen dem Gradienten, also der Richtung des stärksten Anstiegs, und der Höhenlinie können wir uns zu Nutzen machen, um die Struktur und das Verhalten von Höhenlinien zu untersuchen.

Lemma 4.1.2. *Wir betrachten die Höhenlinien in der Umgebung eines Punktes* $(a, b) \in U, f(a, b) =: c$ *und* $(gradf)(a, b) \neq (0, 0)$*. Nun wenden wir Satz 2.3.1 an, wobei im Fall* $\frac{\partial f}{\partial x} \neq 0$ *die Rollen von* x *und* y *vertauscht werden. Daraus ergeben sich zwei Fälle:*

1. *Falls* $\frac{\partial f}{\partial y} \neq 0$*, existieren offene Intervalle* $I_1, I_2 \subset \mathbb{R}$ *mit* $(a, b) \in I_1 \times I_2 \subset U$ *und eine stetige differenzierbare Funktion* $\varphi : I_1 \to I_2$ *so dass*

$$N_f(c) \cap (I_1 \times I_2) = (x, y) \in I_1 \times I_2 : y = \varphi(x).$$

2. *Falls* $\frac{\partial f}{\partial x} \neq 0$*, existieren offene Intervalle* $J_1, J_2 \subset \mathbb{R}$ *mit* $(a, b) \in J_1 \times J_2 \subset U$ *und eine stetige differenzierbare Funktion* $\varphi : J_1 \to J_2$ *so dass*

$$N_f(c) \cap (J_1 \times J_2) = (x, y) \in J_1 \times J_2 : y = \varphi(x).$$

[For17, S. 109 f.]

Dieses Lemma zeigt uns, dass Höhenlinien in der Umgebung eines Punktes, in dem der Gradient nicht verschwindet, stets als Graph einer Funktion dargestellt werden kann.

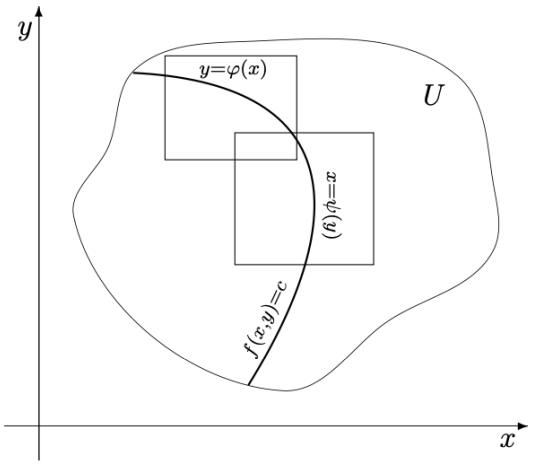


Abbildung 4.1: Höhenlinien in einer Umgebung U

Wie in Abbildung 4.1 zu sehen ist, kann dabei die Funktion entweder in Abhängigkeit von x oder von y dargestellt werden.

4.2 Verfahren zur Berechnung von Höhenlinien

In Abschnitt 3.3 wurden bereits einige Verfahren aufgezeigt, wie Höhenlinien in der Praxis erstellt werden. Da wir nun wissen, dass Höhenlinien unter gewissen Voraussetzungen als Graph einer impliziten Funktion darstellbar sind, wollen wir uns als Nächstes damit beschäftigen, wie wir bei gegebener Funktion die Niveaumenge $N_F(c)$ erhalten. Auch hierfür gibt es verschiedene Verfahren.

Die *analytische Methode* wird bei einfachen Funktionen wie Kreisen ($F(x, y) = x^2 + y^2 = c$), Parabeln oder Hyperbeln verwendet. Bei solchen impliziten Funktionen kann die Gleichung analytisch umgestellt und explizit gelöst werden. Diese Methode ist jedoch nur bei Funktionen möglich, die nach x oder y algebraisch oder mit Substitution auflösbar sind. Diese Vorgehensweise wurde bereits intuitiv in Beispiel 2.2.1 verwendet.

Bei komplexeren Funktionen, die sich nicht analytisch lösen lassen, kommen *numerische Verfahren* zum Einsatz. Dabei wird zuerst ein Definitionsbereich der Form x_{min}, x_{max} und y_{min}, y_{max} in ein regelmäßiges Gitter von diskreten Punkten $(x_i), (y_i)$ unterteilt. Anschließend werden die Funktionswerte $f(x, y)$ an jedem Punkt des Gitters berechnet und die Bereiche identifiziert, in denen $f(x, y) = c$ gilt oder der Wert c wechselt. Um

letztlich die exakten Punkte auf den Gitterkanten zu bestimmen, an denen die Gleichung erfüllt ist, wird lineare Interpolation auf zwei benachbarte Gitterpunkte (x_i, y_i) und (x_{i+1}, y_i), an denen ein Wechsel des Wertes c auftritt, angewendet. Dieser Schritt wird für alle benachbarte Punkte in y-Richtung wiederholt. Anschließend können die Punkte, welche die Gleichung für den gleichen Wert c erfüllen, miteinander verbunden werden. Eine andere Möglichkeit wäre an dieser Stelle die Verwendung des Newton-Verfahrens, um die Nullstellen von $F(x, y) = f(x, y) - c$ genauer berechnen zu können. Durch die Skalierbarkeit und Flexibilität numerischer Verfahren kommen dies Methoden auch in der Praxis häufiger vor als die klassischen. [Sau23, S. 125 ff.]

Das letzte Verfahren basiert auf der Berechnung mit Hilfe des *Gradientens*. Wir wissen bereits, dass der Gradient grad $f(x, y)$ die Richtung des steilsten Anstiegs angibt und orthogonal zur Höhenlinie ist. Bei dieser Methode wählt man zuerst einen Startpunkt (x_0, y_0), für welchen $f(x_0, y_0) = c$ bereits gilt. Für diesen Punkt wird der Gradient bestimmt. Die Bewegungsrichtung entlang der Höhenlinie h ist orthogonal zum Gradienten, d.h.:

$$h = \begin{bmatrix} -\nabla f_y \\ \nabla f_x \end{bmatrix}$$

Anschließend wird der Punkt um einen kleinen Schritt s in Richtung h entlang der Höhenlinie so verschoben, dass

$$(x_1, y_1) = (x_0, y_0) + s \cdot h.$$

Falls nun $f(x_1, y_1) \neq c$ gilt, korrigiert man den neuen Punkt durch Projektion auf die Niveaulinie. Durch Iteration können weitere Punkte berechnet werden, bis die Niveaumenge vollständig ist. [Heu04]

4.3 Beispielhafte Berechnung

Zum Abschluss dieses theoretischen Teils über Höhenlinien als implizite Funktion, soll nun ein konkretes Beispiel dargestellt werden. Dazu betrachten wir die Funktion

$$f(x, y) = \frac{1}{x^2 + y^2 + 5}.$$

Um nun die zugehörigen Höhenlinien als Graph einer impliziten Funktion darstellen zu können, wird zuerst die Gleichung $f(x, y) = c$ mit der Konstante c aufgestellt und anschließend umgeformt. Somit erhalten wir

$$x^2 + y^2 = \frac{1}{c} - 5.$$

Die neue Gleichung lässt bereits erkennen, dass es sich bei diesen Höhenlinien um Kreislinien mit Mittelpunkt $(0, 0)$ und Radius $r = \sqrt{\frac{1}{c} - 5}$ handeln, sofern $\frac{1}{c} > 5$ gilt.

Aus Beispiel 2.4.7 wissen wir bereits für Kreislinien, unabhängig vom Radius, dass nach dem Hauptsatz über implizite Funktionen lokal nach x oder nach y aufgelöst werden kann. Daraus ergibt sich dann nach Umstellung nach y die Gleichung

$$y(x) = \sqrt{\frac{1}{c} - x^2 - 5},$$

analog für die Auflösung nach x. Mit dieser neuen expliziten Funktion können nun für verschiedene Werte c die Funktionen für die Höhenlinien in Abhängigkeit einer Variablen dargestellt werden.

Für $c = \frac{1}{10}$ ergibt sich bspw. ein Radius $r = \sqrt{5}$ und $y(x) = \sqrt{10 - x^2 - 5} = \sqrt{x^2 + 5}$.
Für $c = \frac{1}{20}$ ergibt sich $r = \sqrt{15}$ und $y(x) = \sqrt{20 - x^2 - 5} = \sqrt{x^2 + 15}$.

Wir können zwar nun für verschiedene Höhen c den Radius und eine explizite Höhenfunktion berechnen, doch darunter vorstellen, wie das Höhenlinienbild nun aussehen könnte, können wir uns bei den meisten Funktionen jedoch nicht. Deshalb wollen wir im nächsten Kapitel die berechneten Höhenlinien visualisieren.

5 Implementierung in Octave

Nachdem in Kapitel 4 die mathematische Grundlage zur Berechnung von Höhenlinien gelegt wurde, wird in diesem Kapitel die praktische Umsetzung dieser Theorie beschrieben. Dazu wird die Software *GNU Octave* verwendet, um die Höhenlinien zu visualisieren und die Anwendbarkeit der Theorie zu demonstrieren. Der Quellcode in Octave soll anhand der Funktion aus Abschnitt 4.3 dargestellt werden.

5.1 Die Software Octave

GNU Octave ist eine interaktive Skriptsprache, die in erster Linie für numerische Berechnungen gedacht ist und größtenteils mit der MATLAB kompatibel ist. Octave hat genauso einen integrierten 2D/3D-Plot und Visualisierungswerkzeuge. Der größte Vorteil von Octave gegenüber der kommerziellen Software MATLAB ist die freie Zugänglichkeit für alle gängigen Systeme. Der Octave-Interpreter kann im GUI-Modus, als Konsole oder als Teil eines Shell-Skripts aufgerufen werden. [KJ12] Die Wahl von GNU Octave als Software ermöglicht eine freie und reproduzierbare Implementierung der beschriebenen Methoden. Im Folgenden wird detailliert auf die Implementierung eingegangen.

5.2 Implementierung von impliziten Funktionen in Octave

Zu Beginn der Visualisierung wird die implizite Funktion in der Form $f(x, y) = 0$ in Octav definiert. Anschließend wird ein Intervall für x und y definiert und ein Gitter mit dem Befehl *meshgrid* erzeugt, auf dem die Funktion ausgewertet wird. Bei der Definition der Intervalle ist der Befehl $x = linspace(x_{start}, x_{end}, N)$ nützlich, da hierbei

nicht nur der Start- und Endwert, sondern auch die N Anzahl der Punkte, die zwischen den beiden Werten berechnet werden soll, angegeben werden kann. Die Berechnung der Funktionswerte erfolgt, indem man diese Werte einer neuen Variablen z zuweist.

Um nun die Funktion zu plotten, verwendet man den Befehl *surf*. Damit erhält man nicht nur die einzelne Punkte (x, y), sondern ganze Flächen. Für die Übersichtlichkeit werden noch die Achsen und die Figur beschriftet. Optional kann die Farbe der Oberfläche in Abhängigkeit der Variable z anders eingefärbt werden, um den Höhenverlauf wie in einer topografischen Karte darzustellen. Dies gelingt durch die Erstellung einer benutzerdefinierten Colormap.

Im folgenden Listing 5.2 ist die Implementierung der beispielhaften impliziten Funktion in Octave zu sehen. Dabei wurde Schritt für Schritt wie oben beschrieben implementiert.

```
%Definition der impliziten Funktion
f = @(X,Y) 1 ./ (X.^2 + Y.^2 + 5);

%Erstellung des Gitters für x und y
x = linspace(-5, 5, 100);
y = linspace(-5, 5, 100);
[X, Y] = meshgrid(x, y);

% Berechnung von f(x, y)
Z = f(X,Y);

%3D-Plot der Oberfläche von f
figure;
surf(X, Y, Z);
shading interp; %Glättung der Farbverläufe

%Beschriftung als Koordinatensystems
xlabel('x');
ylabel('y');
zlabel('f(x,␣y)');
title('3D-Plot␣von␣f(x,␣y)␣=␣1/(x^2+y^2+5)',
    'FontSize', 13, 'FontWeight', 'bold');

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_1.pdf','-bestfit');

% Definieren der Höhenfarben im topografischen Stil
```

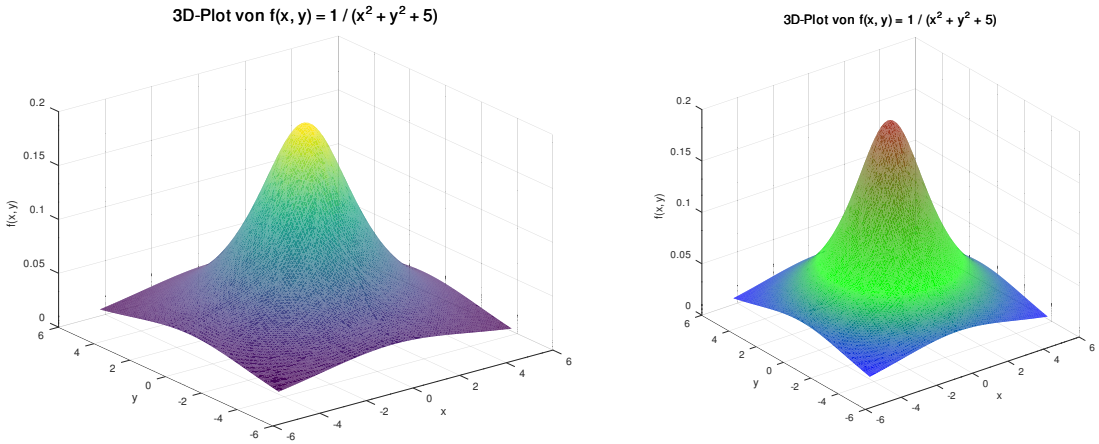
```
blau = [0,0,1];
gruen =  [0,1,0];
braun = [0.65,0.2,0.15];

%Erstellung der Farbskala
n = 256; % Anzahl der Farben
blau_gruen_braun = [linspace(blau(1), gruen(1), round(n * 0.25))',
                    linspace(blau(2), gruen(2), round(n * 0.25))',
                    linspace(blau(3), gruen(3), round(n * 0.25))'];
gruen_braun = [linspace(gruen(1), braun(1), round(n * 0.75))',
                linspace(gruen(2), braun(2), round(n * 0.75))',
                linspace(gruen(3), braun(3), round(n * 0.75))'];
custom_cmap = [blau_gruen_braun; gruen_braun];

% Setze die Colormap
colormap(custom_cmap);
caxis([min(Z(:)), max(Z(:))]); % Farbskala anpassen

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_2.pdf','-bestfit');
```

Listing 5.1: Octave-Code zur Implementierung impliziter Funktionen



(1) 3D-Plot ohne Farbanpassung

(2) 3D-Plot mit Farbanpassung

Abbildung 5.1: Visualisierung der Funktion $f(x, y) = \frac{1}{x^2+y^2+5}$.

In Abb. 5.1 ist links die visualisierte Figur ohne und rechts die Darstellung mit Anpassung der Farben an eine topografische Karte. Dabei ist im Code 5.2 zu Erkennen, dass Figur 1 vor der Erstellung der benutzerdefinierten Colormap gespeichert wurde und

Figur 2 danach.

5.3 Visualisierung von Höhenlinien

Nun wollen wir zusätzlich zu der dreidimensionalen Oberfläche der Funktion Höhenlinien einzeichnen und diese beschriften. Als erstes wird wieder die Definition der Funktion und da Gitter von vorhin übernommen, um die Funktionswerte Z berechnen zu können.

Zur Implementierung der Höhenlinien wird die Funktion *contour* verwendet. Dabei werden in die Funktion die Variablen, in welchem Wertebereich und mit welcher Äquidistanz die Höhenlinien eingezeichnet werden sollen, eingetragen. Da die Größe dieser Werte abhängig von der Funktion sind, um ein passendes Höhenbild zu erhalten, wurden hierfür in die Funktionen die Variablen $A, B, DundV$ eingesetzt. Dabei steht A für den minimalen und B für den maximalen x- bzw. y-Wert. D ist die gewünschte Äquidistanz. Die zusätzliche Variable V ist Vielfaches von der Äquidistanz. Diese Variable soll später zur Linienbeschriftung beitragen.

```
%Definition der impliziten Funktion
f = @(X,Y) 1 ./ (X.^2 + Y.^2 + 5);

%Erstellung des Gitters für x und y
x = linspace(-5, 5, 100);
y = linspace(-5, 5, 100);
[X, Y] = meshgrid(x, y);

% Berechnung von f(x, y)
Z = f(X,Y);

%Definiere Start-, Endpunkt und Aequidistanz
A = -0,1; %Startpunkt
B = 0.3; %Endpunkt
D = 0.0125 %Aequidistanz
V = 2*D %Vielfache

%Zeichnen der Höhenlinien
figure;
contour(X, Y, Z, A:D:B);

% Manuelles Farbschema für die Topografie
% Blau für niedrige Werte, grün für mittlere, braun für hohe
topo_colors = [
0.2 0.6 0.8; % Blau (z. B. Wasser)
0.4 0.8 0.4; % Grün (tieferes Land)
```

```
0.8 0.8 0.4; % Gelblich (mittleres Land)
0.6 0.4 0.2; % Braun (Hügel)
0.9 0.9 0.9; % Weiß (Berggipfel)
];

% Festlegen des Farbschemas
colormap(topo_colors);

% Bestimme die Vielfachen von D im Bereich von A bis B
levels = A:D:B;
labeled_levels = levels(mod(levels, V) == 0); % Nur Vielfache von 2*D

% Beschriftung der Vielfachen der Höhenlinien
[C, h] = contour(X, Y, Z, A:D:B);
clabel(C, h, labeled_levels, 'FontSize', 10);
 %Beschriftung mit Schriftgröße

%Beschriftung der Achsen
xlabel('x');
ylabel('y');
title('Höhenlinien␣der␣impliziten␣Funktion␣f');

grid on;  %Hinzufügen von Hilfslinien
axis equal; %Gleichheit der Maßstäbe der x- und der y-Achse

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_3.pdf');
```

Listing 5.2: Octave-Code zur Implementierung von Höhenlinien

Nun können die Höhenlinien mit den Befehlen *figure* und *contour*gezeichnet werden. Dabei werden die Linien in ein zweidimensionales Raster übertragen. Um die Höhenlinien wie vorhin in den typischen Farben einer topografischen Karte zu gestalten, wurde wieder ein benutzerdefiniertes *Colormap* erstellt.

Mit der Funktion *clabel* können schließlich noch die Höhenlinienbeschriftungen hinzugefügt werden. Dabei wird der Funktion die Konturen zugeordnet und mit *Fontsize* die Schriftgröße. Für zusätzliche Lesbarkeit wurde die Vielfachheit V als zweifache Äquidistanz definiert. Mit der erstellten Funktion *labeled_levels* werden alle Höhenlinien bestimmt, die eine gewünschte Vielfachheit von D aufweisen, in diesem Beispiel $2 \cdot D$. Dies bewirkt, dass nicht alle Höhenlinien, sondern nur jede zweite beschriftet werden.

Am Ende der Implementierung werden die Achsen des zweidimensionalen Plots beschriftet und für die bessere Ablesbarkeit der Punkt mit *grid on* Hilfslinien eingezeichnet. Der letzte Befehl *axis equal* ist nicht notwendig, sondern nur in diesem Fall für die optische Darstellung.

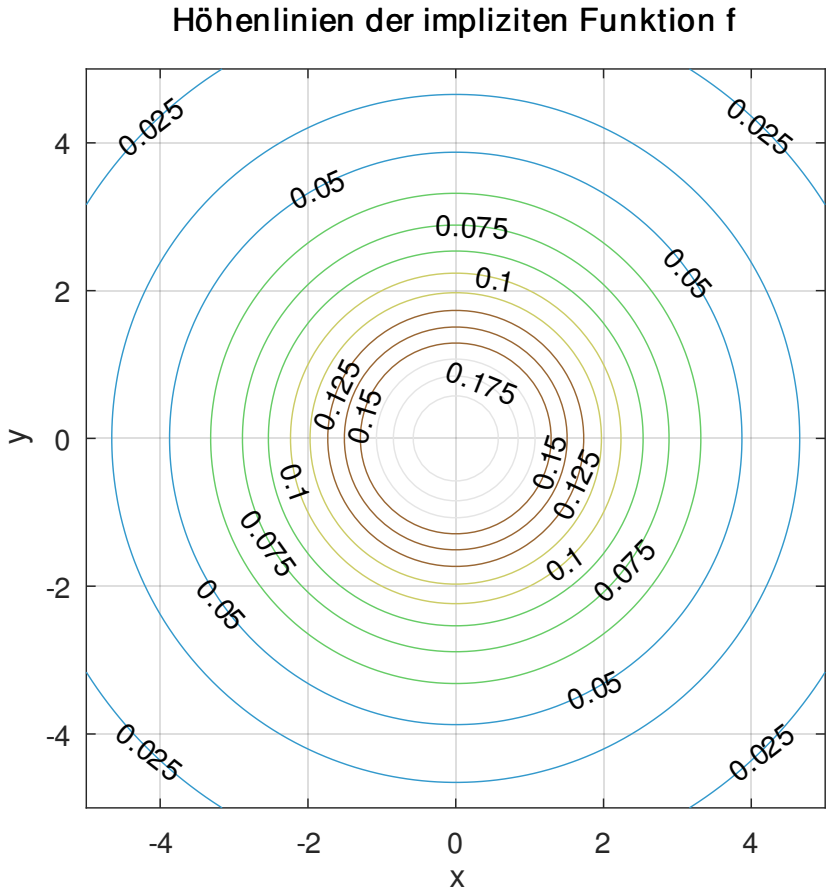


Abbildung 5.2: 2D-Plot mit Farbanpassung

Besonders in der Abbildung 5.2 erkennt man, dass je enger zwei benachbarte Höhenlinien aneinanderliegen, desto steiler wird das Gelände.

Durch Zusammenfügen der beiden Listings 5.2 und 5.3 konnte somit ein Code in Octave herausgearbeitet werden, mit welchem das Höhenlinienbild einer gegebenen Funktion gezeichnet werden kann.

Der gesamte Code ist in Abschnitt A.1 zu finden, sowie eine weitere Visualisierung einer impliziten Funktion und deren Höhenlinien Abschnitt A.2.

5.4 Herausforderungen

Die Herangehensweise an die Erstellung von Höhenlinien durch implizite Funktionen wurde nun ausführlich dargestellt. Doch nur weil wir uns jetzt einen Code in Octave implementiert haben, heißt das leider nicht, dass wir alle beliebigen impliziten Funktionen einsetzten können und sofort eine fehlerfreie Darstellung angezeigt wird. Vor allem bei komplexeren oder trigonometrischen Funktionen, wie in Listing A.2, müssen Einstellungen passend gewählt werden.

Eine Schwierigkeit stellen numerische Instabilitäten dar. Bei Singularitäten zum Beispiel bricht die Methode zusammen. Eine mögliche Lösung könnte die Anwendung lokaler Approximationen oder spezifischer numerischer Verfahren sein, wie das Newton-Verfahren.

Eine weitere Herausforderung bei der topografischen Darstellung ist die Überprüfung, ob die Matrizen zur Erstellung der Colormap auch die richtige Größe haben, da sonst die Implementierung an diesem Schritt abbrechen kann.

Wenn wir den Code in 5.2 nochmal genauer anschauen, ist der zweite und wichtiges Schritt nach der Eingabe der Funktion die Wahl der Gittergröße und der Schrittweite. Die im Abschnitt 4.2 beschriebenen numerischen Verfahren werden hier durch die Implementierung in Octave konkretisiert. Das verwendete Raster und die Schrittweitenwahl zeigen die praktische Anwendung der theoretischen Grundlagen. Dabei sollte beachtet werden, dass die Funktion auf diesem Bereich definiert ist. Außerdem gilt zu Beachten, dass Berechnungen auf einem feineren oder auf einem unproportional großem Raster rechenintensiver sind. Wird die Schrittweite Δx, Δy zu groß gewählt, könnte die Genauigkeit der Darstellung drunter leiden. Ist die Krümmung der Funktion hoch, wie z.B. bei einem Sattelpunkt), variiert die Funktion zwischen den Gitterpunkten stark, was zu signifikanter Abweichungen der Höhenlinie führt. Dadurch können wichtige Details der Höhenlinie verloren gehen.

6 Fazit und Ausblick

Diese Arbeit hat auf theoretisch fundierter Grundlage eindrucksvoll demonstriert, dass implizite Funktionen ein präzises und stabiles Werkzeug zur Berechnung von Höhenlinien darstellen. Insbesondere die Implementierung in Octave hat die praktische Anwendbarkeit dieser Methode verdeutlicht und ihre Effizienz sowie Genauigkeit im Vergleich zu klassischen Ansätzen herausgestellt. Trotz einiger Herausforderungen, wie numerischer Instabilitäten oder hoher Rechenintensität bei komplexeren Modellen, zeigt sich die Überlegenheit dieser Methode in ihrer Stabilität und Vielseitigkeit.

Die zentrale topografische Relevanz des Themas wurde im Verlauf der Arbeit mehrfach betont. Ein Ausblick auf mögliche Weiterentwicklungen verdeutlicht das breite Anwendungspotenzial. So können auf der Grundlage der Implementierung in Octave digitale Höhenlinienkarten und detaillierte Geländemodelle erstellt werden. Diese können nicht nur zur anschaulichen Visualisierung der Erdoberfläche und zur Unterstützung von Bauvorhaben dienen, sondern auch als Grundlage für Simulationen wie Hochwasser- und Lawinenprognosen eingesetzt werden. Damit würde diese Technologie nicht nur technische Fortschritte ermöglichen, sondern könnte in Extremsituationen sogar lebensrettend sein.

A Quellcode in Octave

A.1 Gesamter Quellcode

```
%Definition der impliziten Funktion
f = @(X,Y) 1 ./ (X.^2 + Y.^2 + 5);

%Erstellung des Gitters für x und y
x = linspace(-5, 5, 100);
y = linspace(-5, 5, 100);
[X, Y] = meshgrid(x, y);

% Berechnung von f(x, y)
Z = f(X,Y);

%3D-Plot der Oberfläche von f
figure;
surf(X, Y, Z);
shading interp; %Glättung der Farbverläufe

%Beschriftung als Koordinatensystems
xlabel('x');
ylabel('y');
zlabel('f(x, y)');
title('3D-Plot von f(x, y) = 1/(x^2+y^2+5)',
            'FontSize', 16, 'FontWeight', 'bold');

% Definieren der Höhenfarben im topografischen Stil

blau = [0,0,1];
gruen =  [0,1,0];
braun = [0.6,0.3,0.1];

%Erstellung der Farbskala
```

```
n = 256; % Anzahl der Farben
blau_gruen_braun = [linspace(blau(1), gruen(1), round(n * 0.25))',
                    linspace(blau(2), gruen(2), round(n * 0.25))',
                    linspace(blau(3), gruen(3), round(n * 0.25))'];
gruen_braun = [linspace(gruen(1), braun(1), round(n * 0.75))',
               linspace(gruen(2), braun(2), round(n * 0.75))',
               linspace(gruen(3), braun(3), round(n * 0.75))'];
custom_cmap = [blau_gruen_braun; gruen_braun];

% Setze die Colormap
colormap(custom_cmap);
caxis([min(Z(:)), max(Z(:))]); % Farbskala anpassen
colorbar;

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_2.pdf');

%Definiere Start-, Endpunkt und Aequidistanz
A = -0,1; %Startpunkt
B = 0.3; %Endpunkt
D = 0.0125 %Aequidistanz
V = 2*D %Vielfache

%Zeichnen der Höhenlinien
figure;
contour(X, Y, Z, A:D:B);

% Manuelles Farbschema für die Topografie
% Blau für niedrige Werte, grün für mittlere, braun für hohe
topo_colors = [
0.2 0.6 0.8; % Blau (z. B. Wasser)
0.4 0.8 0.4; % Grün (tieferes Land)
0.8 0.8 0.4; % Gelblich (mittleres Land)
0.6 0.4 0.2; % Braun (Hügel)
0.9 0.9 0.9; % Weiß (Berggipfel)
];

% Festlegen des Farbschemas
colormap(topo_colors);

% Bestimme die Vielfachen von D im Bereich von A bis B
levels = A:D:B;
```

```
labeled_levels = levels(mod(levels, V) == 0); % Nur Vielfache von 2*D

% Beschriftung der Vielfachen der Höhenlinien
[C, h] = contour(X, Y, Z, A:D:B);
clabel(C, h, labeled_levels, 'FontSize', 10);
 %Beschriftung mit Schriftgröße

%Beschriftung der Achsen
xlabel('x');
ylabel('y');
title('Höhenlinien␣der␣impliziten␣Funktion␣f');

grid on;  %Hinzufügen von Hilfslinien
axis equal; %Gleichheit der Maßstäbe der x- und der y-Achse

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_3.pdf');
```

Listing A.1: Octave-Code zur Implementierung von impliziten Funktionen und deren Höhenlinien

A.2 Weiteres Beispiel

```
%Definition der impliziten Funktion
f = @(X, Y) 1 ./ (X.^2 + Y.^2 +1) .* cos(pi * X) .* cos(pi * Y);

%Erstellung des Gitters für x und y
x = linspace(-1.5, 1.5, 100);
y = linspace(-1.5, 1.5, 100);
[X, Y] = meshgrid(x, y);

% Berechnung von f(x, y)
Z = f(X,Y);

%3D-Plot der Oberfläche von f
figure;
surf(X, Y, Z);
shading interp; %Glättung der Farbverläufe

%Beschriftung als Koordinatensystems
```

```
xlabel('x');
ylabel('y');
zlabel('f(x, y)');
title('3D-Plot von f(x, y) = 1/(x^2+y^2+1).*cos(pi*X).*cos(pi*Y)',
                            'FontSize', 12, 'FontWeight', 'bold');

% Definieren der Höhenfarben im topografischen Stil

blau = [0,0,1];
gruen =  [0,1,0];
braun = [0.6,0.3,0.1];

%Erstellung der Farbskala
n = 256; % Anzahl der Farben
n1 = round(n * 0.25);
%horzcat: horizontale Verkettung von Matrizen
blau_gruen_braun = horzcat(
    linspace(blau(1), gruen(1), n1)',
    linspace(blau(2), gruen(2), n1)',
    linspace(blau(3), gruen(3), n1)');
n2 = round(n * 0.75);
gruen_braun = horzcat(
    linspace(gruen(1), braun(1), n2)',
    linspace(gruen(2), braun(2), n2)',
    linspace(gruen(3), braun(3), n2)'
);
custom_cmap = [blau_gruen_braun; gruen_braun];

% Setze die Colormap
colormap(custom_cmap);
caxis([min(Z(:)), max(Z(:))]); % Farbskala anpassen
colorbar;

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_4.pdf');

%Definiere Start-, Endpunkt und Aequidistanz
A = -0.3; %Startpunkt
B = 0.3; %Endpunkt
D = 0.025 %Aequidistanz
V = 2*D %Vielfache
```

```
%Zeichnen der Höhenlinien
figure;
contour(X, Y, Z, A:D:B);

% Manuelles Farbschema für die Topografie
% Blau für niedrige Werte, grün für mittlere, braun für hohe
topo_colors = [
0.2 0.6 0.8; % Blau (z. B. Wasser)
0.4 0.8 0.4; % Grün (tieferes Land)
0.8 0.8 0.4; % Gelblich (mittleres Land)
0.6 0.4 0.2; % Braun (Hügel)
0.9 0.9 0.9; % Weiß (Berggipfel)
];

% Festlegen des Farbschemas
colormap(topo_colors);

% Bestimme die Vielfachen von D im Bereich von A bis B
levels = A:D:B;
labeled_levels = levels(mod(levels, V) == 0); % Nur Vielfache von 2*D

% Beschriftung der Vielfachen
[C, h] = contour(X, Y, Z, A:D:B);
clabel(C, h, labeled_levels, 'FontSize', 10);
 %Beschriftung mit Schriftgröße

 %Beschriftung der Achsen und der Höhenlinien
xlabel('x');
ylabel('y');
title('Höhenlinien␣der␣impliziten␣Funktion␣f');

grid on;  %Hinzufügen von Hilfslinien
axis equal; %Gleichheit der Maßstäbe der x- und der y-Achse

%Speicherung der Figur als PDF
print('-dpdf', 'Figur_5.pdf');
```

Listing A.2: Octave-Code zur Implementierung von $f(x,y) = \frac{1}{x^2+y^2-5} cos(\pi x) cos(\pi y)$

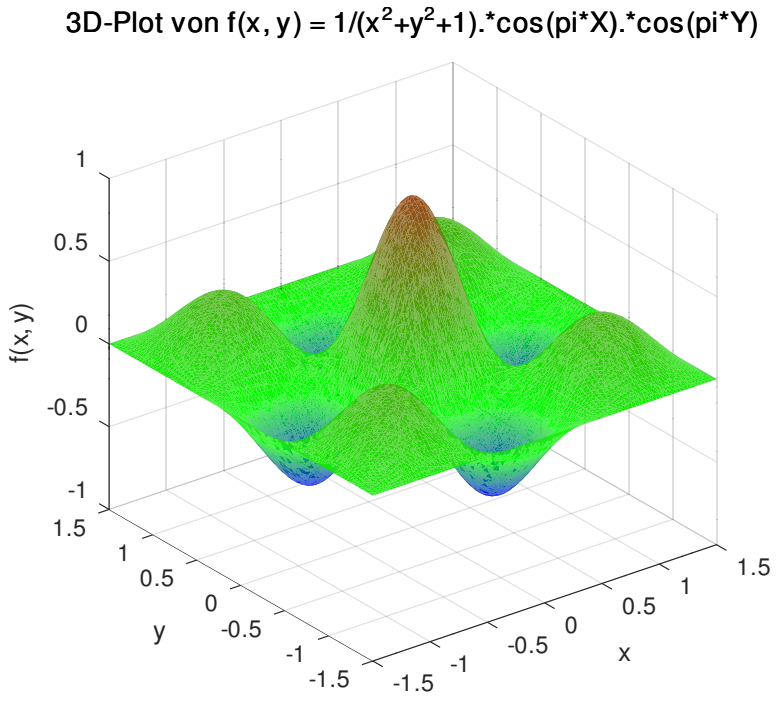


Abbildung A.1: Visualisierung der Funktion $f(x,y) = \frac{1}{x^2+y^2-5} cos(\pi x) cos(\pi y)$

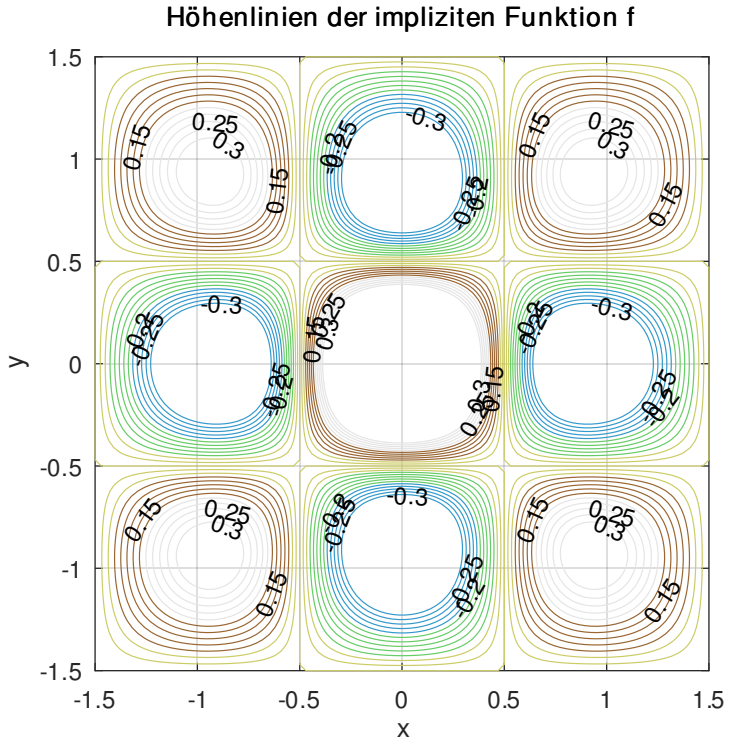


Abbildung A.2: Höhenlinienbild der Funktion $f(x,y) = \frac{1}{x^2+y^2-5} cos(\pi x) cos(\pi y)$

Literatur

[Are+22] Tilo Arens u. a. *Mathematik*. 5. Aufl. Springer Berlin Heidelberg, 2022. ISBN: 9783662643884 (siehe S. 6, 7).

[Dep24] Statista Research Department. *Beliebteste Hobbys, Freizeitaktivitäten und Sportarten in Deutschland nach häufiger Ausübung in den Jahren 2021 bis 2023*. 2024. URL: https://de.statista.com/statistik/daten/studie/171168/umfrage/haeufig-betriebene-freizeitaktivitaeten/ (siehe S. 1).

[For17] Otto Forster. *Analysis 2 - Differentialrechnung im $\mathbb{R}^n$, gewöhnliche Differentialgleichungen*. Springer Berlin Heidelberg, 2017. ISBN: 9783658194109 (siehe S. 8, 9, 11, 12, 19, 40).

[GB11] Willibald Günther und André Borrmann. *Digitale Baustelle - innovativer Planen, effizienter Ausführen*. Springer Berlin Heidelberg, 2011. ISBN: 9783642164859 (siehe S. 17).

[Hei16] Christian Heipke. *Photogrammetrie und Fernerkundung*. Springer Berlin Heidelberg, 2016. ISBN: 9783662470930 (siehe S. 17).

[Heu04] Harro Heuser. *Lehrbuch der Analysis Teil 2*. 13. Aufl. Teubner Verlag Wiesbaden, 2004. ISBN: 9783519622321 (siehe S. 11, 12, 21).

[Kah93] Heribert Kahmen. *Vermessungskunde*. 18. Aufl. De Gruyter, 1993. ISBN: 9783110874068 (siehe S. 16, 17).

[KJ12] Torsten Krüger und Frohne Jörg. „Kompakte Einführung in GNU Octave“. 2012 (siehe S. 23).

[Lana] Breitband und Vermessung Bayern Landesamt für Digitalisierung. *Digitale Höhenlinienkarte*. URL: https://www.ldbv.bayern.de/produkte/3dprodukte/digi_hoehe (siehe S. 14).

[Lanb] Breitband und Vermessung Bayern Landesamt für Digitalisierung. *Digitale Höhenlinienkarte*. URL: https://www.ldbv.bayern.de/produkte/topo/atk.html (siehe S. 40).

[Sau23] Thomas Sauer. „Einführung in die numerische Mathematik“. Skript zur Vorlesung an der Universität Passau im Sommersemester 2023. 2023 (siehe S. 21).

[Sau24] Thomas Sauer. „Analysis II“. Skript zur Vorlesung an der Universität Passau im Wintersemester 2023/24. 2024 (siehe S. 4, 8, 11–13).

[Ver] Landesamt für Vermessung und Geoinformationen Bayern. *Tipps zum Kartenlesen*. URL: https://elearning.uni-regensburg.de/pluginfile.php/1247724/coursecat/description/Faltblatt%20Kartenkunde.pdf (siehe S. 16).

Abbildungsverzeichnis

2.1 Verkleinerung der Umgebungen U und V 9

3.1 Ausschnitt der topografischen Karte östlich des Tegernsees, Bayern . . . 15

4.1 Höhenlinien in einer Umgebung U 20

5.1 Visualisierung der Funktion $f(x,y) = \frac{1}{x^2+y^2+5}$. 25
5.2 2D-Plot mit Farbanpassung 29

A.1 Visualisierung der Funktion $f(x,y) = \frac{1}{x^2+y^2-5} cos(\pi x) cos(\pi y)$ 37
A.2 Höhenlinienbild der Funktion $f(x,y) = \frac{1}{x^2+y^2-5} cos(\pi x) cos(\pi y)$ 37

Quellenangaben

2.1 Verkleinerung der Umgebungen U und V [For17, S. 105]
3.1 Ausschnitt der topografischen Karte... [Lanb]
4.1 Höhenlinien in einer Umgebung U [For17, S. 110]

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich und sinngemäß übernommenen Passagen aus anderen Werken kenntlich gemacht habe.
Die Arbeit ist weder von mir noch von einer anderen Person an der Universität Passau oder an einer anderen Hochschule zur Erlangung eines akademischen Grades bereits eingereicht worden.

Passau, 20. Dezember 2024

Christina Brandmeier