

Mathematikaufgabe 143

[Home](#) | [Startseite](#) | [Impressum](#) | [Kontakt](#) | [Gästebuch](#)

Aufgabe: Lösen Sie das Trefferproblem des schiefen Wurfs durch ein rekurrentes neuronales Netzwerk.

Lösung: Wenn wir einen Gegenstand werfen, hat unser Gehirn kaum eine Sekunde Zeit, um die drei entscheidenden Größen, i.e. Winkel in Azimut und Elevation und Abgangsgeschwindigkeit, auf die richtigen Werte zu bringen, damit das Geschöß ins Schwarze trifft. Der Wurf würde wohl komplett mißlingen, hätten wir nicht in unserem Gedächtnis die Erinnerung an einen erfolgreichen früheren Wurf abgespeichert. Folglich wird das Gehirn bei jedem weiteren Versuch immer wieder probieren, den erfolgreichen Wurf zu reproduzieren, indem es die verfügbaren Wurfparameter approximiert. Aufgabe des rekurrenten neuronalen Netzes ist es, in der kurzen zur Verfügung stehenden Zeit sämtliche Anfangsparameter mit denen der optimalen Flugbahn abzugleichen. Wir benötigen also zur Lösung der Aufgabenstellung ein mathematisches Modell, welches den schiefen Wurf beschreibt, einen experimentellen Aufbau, der die Realität wiedergibt, und die Angabe der anzustrebenden Idealbedingungen, damit das neuronale Netzwerk weiß, wo das Trefferoptimum liegen soll.¹ Den experimentellen Aufbau haben wir in Abb. 1 dargestellt.

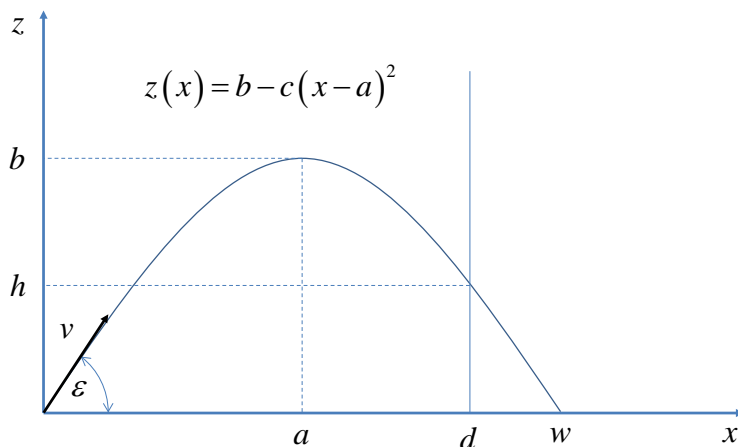


Abbildung 1. Bahnkurve eines idealen schiefen Wurfs längs der x-Achse

Im Abstand d vom Werfer bzw. Schützen befinde sich der anzuvisierende Zielpunkt in der Höhe h . Die Bahnkurve einer solchen Wurfparabel lautet

$$z(x) = b - c(x - a)^2,$$

wobei wir als Randbedingungen die Spezialfälle

$$z(0) = 0 \Leftrightarrow b = ca^2 \Leftrightarrow z(x) = 2acx - cx^2$$

¹ Manchmal wird gefordert, daß das neuronale Netzwerk das Optimum selbst finden soll. Das ist allerdings erst nach einer Vielzahl von Probewürfen möglich, indem das Netz die erfolgreichen Würfe im Gedächtnis behält, d.h. gewichtet, die nicht erfolgreichen hingegen ausblendet. Die Zeit für das Auffinden des Optimums kann in den meisten praktischen Anwendungen jedoch kaum aufgebracht werden, so daß man dem neuronalen Netzwerk die optimalen Trainingswerte (durch überwachtes Lernen) vorgeben muß.

und

$$z(w) = 0 \Leftrightarrow w = 2a$$

haben. Mit der Festlegung

$$c = \frac{g}{2v^2 \cos^2 \varepsilon} \quad \text{und} \quad a = \frac{v^2 \sin \varepsilon \cos \varepsilon}{g}$$

bzw. $2ac = \tan \varepsilon$ folgt daraus die parametrisierte Bahngleichung des schiefen Wurfs,

$$z(x) = x \tan \varepsilon - \frac{g}{2v^2 \cos^2 \varepsilon} x^2,$$

mit einer Reichweite von

$$w = \frac{v^2 \sin 2\varepsilon}{g}.$$

Als Kriterium für einen Treffer gilt nach Abb. 1

$$z(d) = 2acd - cd^2 = h.$$

In der Aufgabenstellung angemesseneren Zylinderkoordinaten können wir die Bahngleichung auch wie folgt schreiben:

$$z(\rho) = \rho \tan \varepsilon - \frac{g}{2v^2 \cos^2 \varepsilon} \rho^2.$$

Die Variable z hängt hierbei nicht vom Azimutwinkel α ab, wohl aber vom Elevationswinkel ε und der Abgangsgeschwindigkeit v . Im Abstand

$$\rho(\alpha) = \frac{d_0}{\cos \alpha}$$

treffe das Geschöß die Zielscheibe dezentral in der Höhe h . Nehmen wir nun an, der Wurf erfolge unter idealen Bedingungen und würde das Geschöß im Abstand d_0 , in einer Höhe h_0 , unter einem Elevationswinkel ε_0 und einem Azimutwinkel $\alpha_0 = 0$ mit der passenden Geschwindigkeit v_0 ins Ziel bringen. Dann gilt

$$x_0 = d_0, \quad y_0 = 0, \quad z_0 = d_0 \tan \varepsilon_0 - \frac{g}{2v_0^2 \cos^2 \varepsilon_0} d_0^2 = h_0.$$

Unter realen Bedingungen wäre der Trefferabstand $d \geq d_0$, und auch der Abgangswinkel ε und die Abgangsgeschwindigkeit v wären bei einem Ungeübten nicht optimal, was sich im Schnittpunkt des Geschosses mit der Zielscheibe in folgender Bedingung ausdrückt:

Mathematikaufgabe 143

$$z = \frac{d_0 \tan \varepsilon}{\cos \alpha} - \frac{gd_0^2}{2v^2 \cos^2 \varepsilon \cos^2 \alpha} = h.$$

Um also das Ziel in beliebiger Höhe h zu treffen, muß eine Geschwindigkeit

$$v = \frac{\sqrt{gd_0^2}}{\sqrt{2 \cos^2 \varepsilon (d_0 \cos \alpha \tan \varepsilon - h \cos^2 \alpha)}}$$

aufgebracht werden, womit nicht nur die z -Koordinate sub-optimal ist, sondern auch die y -Koordinate und der Azimutwinkel des Einschlagspunktes sub-optimale Werte annehmen.

Hingegen gilt für optimale Zielkoordinaten

$$\alpha = \alpha_0, \quad \rho(\alpha_0) = \frac{d_0}{\cos \alpha_0}, \quad z(\alpha_0, \varepsilon_0, v_0) = \frac{d_0 \tan \varepsilon_0}{\cos \alpha_0} - \frac{gd_0^2}{2v_0^2 \cos^2 \alpha_0 \cos^2 \varepsilon_0} = h_0.$$

Dabei ist v_0 definitionsgemäß diejenige Wurfgeschwindigkeit, welche die Zielscheibe unter den Winkeln α_0 und ε_0 in der vorgegebenen Höhe h_0 trifft:

$$v_0 = \frac{\sqrt{gd_0^2}}{\sqrt{2 \cos^2 \varepsilon_0 (d_0 \cos \alpha_0 \tan \varepsilon_0 - h_0 \cos^2 \alpha_0)}}$$

Legen wir nun unser Koordinatensystem aus Gründen der Einfachheit so, daß die x -Achse durch die Mitte des Ziels verläuft, dann erhalten wir für $\alpha_0 = 0$ den einfacheren Ausdruck

$$v_0 = \frac{\sqrt{gd_0^2}}{\sqrt{2 \cos^2 \varepsilon_0 (d_0 \tan \varepsilon_0 - h_0)}}$$

Fordern wir zusätzlich noch, daß der ideale Wurf unter einem Winkel $\varepsilon_0 = \pi/4$ erfolgen soll, ergibt sich daraus die elementare Beziehung

$$v_0 = \sqrt{\frac{gd_0^2}{d_0 - h_0}} \approx \sqrt{gd_0} \left(1 + \frac{1}{2} \frac{h_0}{d_0} \right),$$

d.h. die Geschwindigkeitseinstellung läßt sich auf eine Relativgeschwindigkeits- und Winkelschätzung reduzieren. Liegt schließlich der Abwurfort in Höhe der Zielmitte $h_0 = 0$, dann ist

$v_0 = \sqrt{gd_0}$ eine absolute bzw. reine Geschwindigkeitsabschätzung.²

² Genauer gesagt schätzen wir die Wurzel aus einem Geschwindigkeitsquadrat.

Mathematikaufgabe 143

Allgemein gilt für den Trefferabstand zur Zielscheibenmitte $x - x_0 = 0$, $y - y_0 = d_0 \tan \alpha$ und

$$z - z_0 = d_0 \left(\frac{\tan \varepsilon}{\cos \alpha} - \tan \varepsilon_0 \right) - \left(\frac{1}{v^2 \cos^2 \varepsilon \cos^2 \alpha} - \frac{1}{v_0^2 \cos^2 \varepsilon_0} \right) \frac{g d_0^2}{2}.$$

Die euklidische Abstandsfunktion zwischen realem und idealem Treffer lautet somit

$$\|\mathbf{r} - \mathbf{r}_0\| = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}.$$

Aus den beiden Ortsvektoren für idealen und realen Ort

$$\begin{aligned} \mathbf{r}_0 &= d_0 \mathbf{e}_x + h_0 \mathbf{e}_z, \\ \mathbf{r} &= d_0 \mathbf{e}_x + d_0 \tan \alpha \mathbf{e}_y + z \mathbf{e}_z \end{aligned}$$

erhalten wir die Differenz der Ortsvektoren und damit den Richtungsvektor

$$\mathbf{r} - \mathbf{r}_0 = \rho \sin \alpha \mathbf{e}_y + (z - h_0) \mathbf{e}_z.$$

Mit den Transformationsgleichungen von kartesischen auf Zylinderkoordinaten

$$\begin{aligned} \mathbf{e}_x &= \cos \alpha \mathbf{e}_\rho - \sin \alpha \mathbf{e}_\alpha, \\ \mathbf{e}_y &= \sin \alpha \mathbf{e}_\rho + \cos \alpha \mathbf{e}_\alpha \end{aligned}$$

ergibt sich der Richtungsvektor in Zylinderkoordinaten zu

$$\mathbf{r} - \mathbf{r}_0 = \rho \sin^2 \alpha \mathbf{e}_\rho + \rho \sin \alpha \cos \alpha \mathbf{e}_\alpha + (z - h_0) \mathbf{e}_z.$$

Wir definieren nun den Abstand $\|\mathbf{r} - \mathbf{r}_0\|$ als Hyperfläche in Zylinderkoordinaten:

$$f(\rho, \alpha, z) \equiv \sqrt{\rho^2 \sin^2 \alpha + (z - h_0)^2}.$$

Mit Hilfe des Gradienten

$$\nabla f = \mathbf{e}_\rho \frac{\partial f}{\partial \rho} + \mathbf{e}_\alpha \frac{1}{\rho} \frac{\partial f}{\partial \alpha} + \mathbf{e}_z \frac{\partial f}{\partial z}$$

sowie den partiellen Ableitungen

$$\frac{\partial f}{\partial \rho} = \frac{\rho \sin \alpha \sin \alpha}{\sqrt{\rho^2 \sin^2 \alpha + (z - h_0)^2}}, \quad \frac{\partial f}{\partial \alpha} = \frac{\rho^2 \sin \alpha \cos \alpha}{\sqrt{\rho^2 \sin^2 \alpha + (z - h_0)^2}}, \quad \frac{\partial f}{\partial z} = \frac{z - h_0}{\sqrt{\rho^2 \sin^2 \alpha + (z - h_0)^2}}$$

folgt die Richtungsableitung³

³ Eigentlich hat der Richtungsvektor genau das umgekehrte Vorzeichen, folglich müssen wir diesen mit den negativen partiellen Ableitungen multiplizieren.

Mathematikaufgabe 143

$$\nabla f \cdot \frac{\mathbf{r} - \mathbf{r}_0}{\|\mathbf{r} - \mathbf{r}_0\|} = \sqrt{\rho^2 \sin^2 \alpha + (z - h_0)^2} = \sqrt{d_0^2 \tan^2 \alpha + (z - h_0)^2} \equiv f.$$

Sie wird minimal für $\alpha = 0$ und $z = h_0$. Damit ist die Richtungsableitung der Funktion f die Funktion f selbst. Substituieren wir noch die Koordinaten (α, ρ, z) durch (α, ε, v) und setzen alle Differenzen ein, so ergibt sich mittels der Definition $f(\alpha, \rho, z) \equiv f(\alpha, \varepsilon, v)$ eine Hyperfläche, die von drei für das neuronale Netzwerk besser geeigneten Variablen abhängt:

$$f(\alpha, \varepsilon, v) = d_0 \sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2},$$

wobei ein kleines

$$\frac{h_0}{d_0} = \tan \varepsilon_0 - \frac{gd_0}{2v_0^2 \cos^2 \varepsilon_0}$$

die ungefähre Zielpeilung angibt. Erneut erhalten wir für $\varepsilon_0 = \pi/4$ die einfachere Relation

$$\frac{h_0}{d_0} = 1 - \frac{gd_0}{v_0^2},$$

so daß nach Einsetzen aller Größen gilt:

$$f(\alpha, \varepsilon, v) = d_0 \sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - 1 + \frac{gd_0}{v_0^2} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} \right)^2}.$$

Man sieht sofort, daß diese Hyperfläche ihr Minimum bei

$$\alpha = 0, \quad \varepsilon = \pi/4, \quad v = v_0$$

aufweist. Man kann nun für jedes Wertepaar $(d_0, h_0) - v_0$ ist eine davon abhängige Größe – eine Monte-Carlo-Simulation durchführen, um herauszufinden, für welche Werte von α , ε und v man noch ins Schwarze trifft, d.h. welche Parameterkombinationen mit entsprechend klein gewähltem δ die Ungleichung

$$\|\mathbf{r} - \mathbf{r}_0\| < \delta$$

erfüllen. Sämtliche Ergebnisse, die positiv ausfallen, spannen eine Hyperfläche auf und können als Trainingsmuster für ein neuronales Netzwerk dienen, um daraus letztendlich die Netzgewichte zu bestimmen. Die Positionsdaten des Ziels können vom Gehirn nur geschätzt werden. Daß eine Schätzung möglich ist, beweist die Tatsache, daß ein Autofahrer sehr gut erkennen kann, wie schnell ein anderes Fahrzeug sich nähert und ob es noch weit genug entfernt ist. Mittels der so ermittelten Schätzgrößen sucht unser Gedächtnis nach geeigneten Winkel-

Mathematikaufgabe 143

und Geschwindigkeitskombinationen, welche sich bereits in der Lernphase als erfolgreich erwiesen haben. Mathematisch gesehen liegen diese Größen auf einer zeichnerisch nicht darstellbaren dreidimensionalen Hyperfläche. Einem ausreichend gut trainierten Schützen wird es ohne weiteres gelingen, einen Treffer zu erzielen, auch wenn die approximierte Kombination aus Entfernung und Höhendifferenz nicht explizit trainiert wurde, denn unser Algorithmus hat die Fähigkeit, Punkte auf der Hyperfläche zu interpolieren. Darin liegt der große Vorteil neuronaler Netzwerke: Man muß nicht jede erdenkliche Erfahrung explizit gemacht haben, um neue Herausforderungen meistern zu können, man sollte allerdings ein ausgiebiges Trainingspensum absolviert haben, um weiße Flecken auf der Hyperfläche überbrücken zu können. Wer beispielsweise ein Leben lang nur durch ein Weltraum-Teleskop geblickt hat, wird die beiden disjunkten Mengen Makrokosmos und Mikrokosmos a priori kaum miteinander in Verbindung bringen können und daher auf dem Gebiet der Mikroskopie nur selten ebenfalls eine Koryphäe werden.

In Abb. 2 ist das rekurrente neuronale Netzwerk unserer Aufgabenstellung schematisch dargestellt. Aus Gründen der Übersichtlichkeit verzichten wir zeichnerisch darauf hinzuweisen, daß jeder Ausgabewert eines Neurons iterativ wieder als Eingabewert ins Netz eingespeist werden muß. Wie man sieht, besteht unser Netzwerk aus 4 Schichten, einer Eingabeschicht, in der die *i*ten Eingabeparameter $(\alpha_i, \varepsilon_i, v_i)$ eines speziellen Wurfs eingegeben werden, um daraus die zweiten partiellen Ableitungen der ersten verdeckten Schicht ermitteln zu können, eine weitere verdeckte Schicht und die Ausgabeschicht. In der zweiten verdeckten Schicht werden die ersten partiellen Ableitungen ermittelt, in der Ausgabeschicht die Funktion selbst, nicht zu vergessen die sogenannten Bias-Werte, die zu Beginn der Iteration als konstante Schrittweiten $(\Delta\alpha, \Delta\varepsilon, \Delta v)$ eingegeben werden müssen, selbst aber keine iterativen Größen sind. Nach *n* Iterationen ist die Fehlerrückführung⁴ im lokalen Minimum angelangt.

Im Anschluß wird nun das Backpropagation-Verfahren beschrieben. Zunächst benötigen wir das totale Differential der Hyperfläche, das gegeben ist durch

$$df = \frac{\partial f(\alpha, \varepsilon, v)}{\partial \alpha} d\alpha + \frac{\partial f(\alpha, \varepsilon, v)}{\partial \varepsilon} d\varepsilon + \frac{\partial f(\alpha, \varepsilon, v)}{\partial v} dv,$$

wobei sich die partiellen Ableitungen wie folgt berechnen:

$$\begin{aligned} \frac{\partial f(\alpha, \varepsilon, v)}{\partial \alpha} \equiv f_\alpha(\alpha, \varepsilon, v) &= \frac{d_0}{\sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2}} \tan \alpha \\ &\times \left[\frac{1}{\cos^2 \alpha} + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{v^2 \cos^2 \varepsilon \cos^2 \alpha} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) \right], \end{aligned}$$

⁴ Backpropagation

Mathematikaufgabe 143

$$\frac{\partial f(\alpha, \varepsilon, v)}{\partial \varepsilon} \equiv f_\varepsilon(\alpha, \varepsilon, v) = \frac{d_0}{\sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2}} \frac{1}{\cos^2 \varepsilon}$$

$$\times \left(\frac{1}{\cos \alpha} - \frac{gd_0 \tan \varepsilon}{v^2 \cos^2 \alpha} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right),$$

$$\frac{\partial f(\alpha, \varepsilon, v)}{\partial v} \equiv f_v(\alpha, \varepsilon, v) = \frac{d_0}{\sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2}}$$

$$\times \frac{gd_0}{v^3 \cos^2 \alpha \cos^2 \varepsilon} \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right).$$

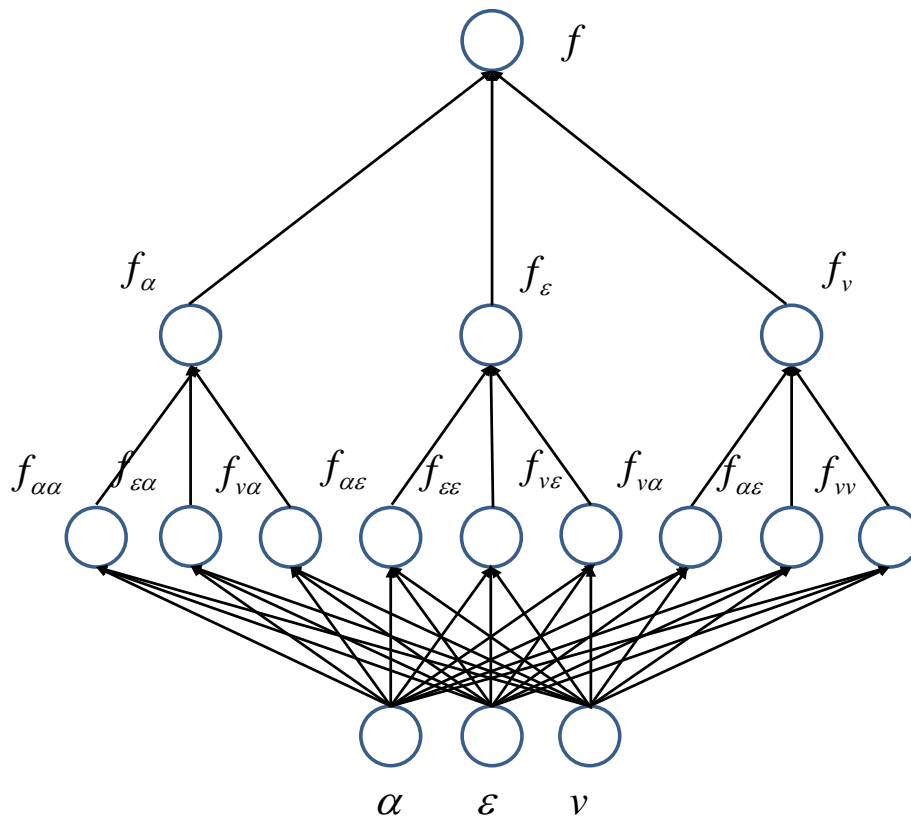


Abbildung 2. Schematische Darstellung eines rekurrenten neuronalen Netzwerks mit vier Lagen

Ersetzen wir die Differentiale durch Differenzen, sehen wir sofort, daß sich jedes weitere Folglied iterativ aus dem vorhergehenden bestimmen läßt:

$$f(\alpha + \Delta\alpha, \varepsilon + \Delta\varepsilon, v + \Delta v) \approx f(\alpha, \varepsilon, v) + f_\alpha(\alpha, \varepsilon, v)\Delta\alpha$$

$$+ f_\varepsilon(\alpha, \varepsilon, v)\Delta\varepsilon + f_v(\alpha, \varepsilon, v)\Delta v.$$

In Vektornotation heißt das

Mathematikaufgabe 143

$$f \begin{pmatrix} \alpha + \Delta\alpha \\ \varepsilon + \Delta\varepsilon \\ v + \Delta v \end{pmatrix} - f \begin{pmatrix} \alpha \\ \varepsilon \\ v \end{pmatrix} = \nabla f \cdot \begin{pmatrix} \Delta\alpha \\ \Delta\varepsilon \\ \Delta v \end{pmatrix}.$$

Die Extremwerte einer Funktion mehrerer Veränderlichen können wir algebraisch durch Nullsetzen der partiellen Ableitungen bestimmen. Entscheidend dafür, ob wir es mit einem relativen Maximum oder Minimum zu tun haben, ist das Vorzeichen der sogenannten Funktionaldeterminante an der Stelle des Extremums

$$\begin{vmatrix} \frac{\partial^2 f}{\partial \alpha^2} & \frac{\partial^2 f}{\partial \alpha \partial \varepsilon} & \frac{\partial^2 f}{\partial \alpha \partial v} \\ \frac{\partial^2 f}{\partial \varepsilon \partial \alpha} & \frac{\partial^2 f}{\partial \varepsilon^2} & \frac{\partial^2 f}{\partial \varepsilon \partial v} \\ \frac{\partial^2 f}{\partial v \partial \alpha} & \frac{\partial^2 f}{\partial v \partial \varepsilon} & \frac{\partial^2 f}{\partial v^2} \end{vmatrix} = \begin{vmatrix} \frac{\partial f}{\partial \alpha} & \frac{\partial f}{\partial \varepsilon} & \frac{\partial f}{\partial v} \\ \frac{\partial f}{\partial \alpha} & \frac{\partial f}{\partial \varepsilon} & \frac{\partial f}{\partial v} \\ \frac{\partial f}{\partial \alpha} & \frac{\partial f}{\partial \varepsilon} & \frac{\partial f}{\partial v} \end{vmatrix} = \begin{vmatrix} f_{\alpha\alpha} & f_{\alpha\varepsilon} & f_{\alpha v} \\ f_{\varepsilon\alpha} & f_{\varepsilon\varepsilon} & f_{\varepsilon v} \\ f_{v\alpha} & f_{v\varepsilon} & f_{vv} \end{vmatrix}.$$

Leider bleibt uns eine Auswertung nicht erspart, auch wenn wir bereits wissen, daß sich das gesuchte Minimum auf der Hyperfläche in der Mitte des Ziels befinden muß.

Zum Auffinden dieses Minimums setzen wir die partiellen Ableitungen allesamt null, so daß wir drei algebraische Gleichungen mit drei Unbekannten erhalten:

$$\begin{aligned} \tan \alpha \left[\frac{1}{\cos^2 \alpha} + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{v^2 \cos^2 \varepsilon \cos^2 \alpha} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) \right] &= 0, \\ \left(\frac{1}{\cos \alpha} - \frac{gd_0 \tan \varepsilon}{v^2 \cos^2 \alpha} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) &= 0, \\ \frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} &= 0. \end{aligned}$$

Setzen wir die dritte Gleichung in die erste ein, folgt sofort $\alpha = 0$. Die beiden verbliebenen Gleichungen lauten somit

$$\begin{aligned} \left(1 - \frac{gd_0 \tan \varepsilon}{v^2} \right) \left(\tan \varepsilon - \frac{gd_0}{2v^2 \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) &= 0, \\ \tan \varepsilon - \frac{gd_0}{2v^2 \cos^2 \varepsilon} - \frac{h_0}{d_0} &= 0. \end{aligned}$$

Die zweite Gleichung hat eine Nullstelle bei

$$v(\varepsilon) = \frac{\cos \varepsilon_0}{\cos \varepsilon} \frac{\sqrt{gd_0}}{\sqrt{gd_0 + 2 \cos^2 \varepsilon_0 (\tan \varepsilon - \tan \varepsilon_0) v_0^2}} v_0.$$

Da $v(\varepsilon_0) = v_0$ gelten muß, folgt

Mathematikaufgabe 143

$$\cos \varepsilon \sqrt{gd_0 + 2 \cos^2 \varepsilon_0 (\tan \varepsilon - \tan \varepsilon_0) v_0^2} = \cos \varepsilon_0 \sqrt{gd_0}.$$

Diese Gleichung ist nur erfüllt für $\varepsilon = \varepsilon_0$. Der Richtungsvektor hat somit folgende Gestalt:

$$\begin{pmatrix} \alpha \\ \varepsilon \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ \varepsilon_0 \\ v_0 \end{pmatrix}.$$

Für eine rekursive Taylorentwicklung bis zu Gliedern erster Ordnung folgen wir dem obigen Ansatz:

$$\begin{aligned} f(\alpha_{i+1}, \varepsilon_{j+1}, v_{k+1}) &\approx f(\alpha_i, \varepsilon_j, v_k) + f_\alpha(\alpha_i, \varepsilon_j, v_k) \Delta \alpha \\ &\quad + f_\varepsilon(\alpha_i, \varepsilon_j, v_k) \Delta \varepsilon + f_v(\alpha_i, \varepsilon_j, v_k) \Delta v, \end{aligned}$$

wobei die partiellen Ableitungen im Punkt $(\alpha_i, \varepsilon_j, v_k)$ gegeben sind durch

$$\begin{aligned} f_\alpha(\alpha_i, \varepsilon_j, v_k) &= \frac{d_0}{\sqrt{\tan^2 \alpha_i + \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right)^2}} \tan \alpha_i \left[\frac{1}{\cos^2 \alpha_i} \right. \\ &\quad \left. + \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{v_k^2 \cos^2 \varepsilon_j \cos^2 \alpha_i} \right) \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right) \right], \end{aligned}$$

$$\begin{aligned} f_\varepsilon(\alpha_i, \varepsilon_j, v_k) &= \frac{d_0}{\sqrt{\tan^2 \alpha_i + \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right)^2}} \frac{1}{\cos^2 \varepsilon_j} \\ &\quad \times \left(\frac{1}{\cos \alpha_i} - \frac{gd_0 \tan \varepsilon_j}{v_k^2 \cos^2 \alpha_i} \right) \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right), \end{aligned}$$

$$\begin{aligned} f_v(\alpha_i, \varepsilon_j, v_k) &= \frac{d_0}{\sqrt{\tan^2 \alpha_i + \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right)^2}} \frac{gd_0}{v_k^3 \cos^2 \alpha_i \cos^2 \varepsilon_j} \\ &\quad \times \left(\frac{\tan \varepsilon_j}{\cos \alpha_i} - \frac{gd_0}{2v_k^2 \cos^2 \alpha_i \cos^2 \varepsilon_j} - \frac{h_0}{d_0} \right). \end{aligned}$$

Mit den partiellen Ableitungen selbst verfahren wir genauso wie mit der Funktion. Wir entwickeln alle partiellen Ableitungen erster Ordnung in eine Taylorreihe bis zur zweiten Ordnung. Das liefert uns folgende Näherungen:

Mathematikaufgabe 143

$$f_\alpha(\alpha + \Delta\alpha, \varepsilon + \Delta\varepsilon, v + \Delta v) \approx f_\alpha(\alpha, \varepsilon, v) + \frac{\partial f_\alpha(\alpha, \varepsilon, v)}{\partial \alpha} \Delta\alpha \\ + \frac{\partial f_\alpha(\alpha, \varepsilon, v)}{\partial \varepsilon} \Delta\varepsilon + \frac{\partial f_\alpha(\alpha, \varepsilon, v)}{\partial v} \Delta v,$$

$$f_\varepsilon(\alpha + \Delta\alpha, \varepsilon + \Delta\varepsilon, v + \Delta v) \approx f_\varepsilon(\alpha, \varepsilon, v) + \frac{\partial f_\varepsilon(\alpha, \varepsilon, v)}{\partial \alpha} \Delta\alpha \\ + \frac{\partial f_\varepsilon(\alpha, \varepsilon, v)}{\partial \varepsilon} \Delta\varepsilon + \frac{\partial f_\varepsilon(\alpha, \varepsilon, v)}{\partial v} \Delta v,$$

$$f_v(\alpha + \Delta\alpha, \varepsilon + \Delta\varepsilon, v + \Delta v) \approx f_v(\alpha, \varepsilon, v) + \frac{\partial f_v(\alpha, \varepsilon, v)}{\partial \alpha} \Delta\alpha \\ + \frac{\partial f_v(\alpha, \varepsilon, v)}{\partial \varepsilon} \Delta\varepsilon + \frac{\partial f_v(\alpha, \varepsilon, v)}{\partial v} \Delta v.$$

Insgesamt haben wir also neun partielle Ableitungen zu bilden, wovon wir exemplarisch nur die erste angeben:

$$\frac{\partial f_\alpha(\alpha, \varepsilon, v)}{\partial \alpha} = \frac{d_0}{\sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2}^3} \\ \times \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{1}{\cos^2 \alpha} \frac{h_0}{d_0} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) \\ \times \left[\frac{1}{\cos^2 \alpha} + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{v^2 \cos^2 \varepsilon \cos^2 \alpha} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right) \right] \\ + \frac{2d_0 \tan^2 \alpha}{\sqrt{\tan^2 \alpha + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{d_0} \right)^2}} \\ \times \left[\frac{\tan \alpha}{\cos^2 \alpha} + \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{2v^2 \cos^2 \alpha \cos^2 \varepsilon} - \frac{h_0}{2d_0} \right) \left(\frac{\tan \varepsilon}{\cos \alpha} - \frac{gd_0}{v^2 \cos^2 \alpha \cos^2 \varepsilon} \right) \right],$$

womit zugleich gezeigt ist, daß eine aufwendige analytische Berechnung prinzipiell möglich ist. Die weiteren Ableitungen wollen wir uns jedoch aus Zeitgründen ersparen,⁵ zumal wir ja bereits wissen, daß das Verfahren an sich funktionieren muß. Im beigefügten Programm können wir die Differenzen daher durch die exakten Differentiale ersetzen. Damit ist unser Netzwerkalgorithmus vollständig beschrieben.

Es sei noch darauf hingewiesen, daß bei gleicher Schrittzahl alle Indizes durch einen einzigen Index ausgedrückt werden können. Mit den Notationen

⁵ Das ist mit einer der Gründe, warum die Entwicklung neuronaler Netzwerke ein zeitaufwendiges und kostspieliges Vorhaben ist.

$$f(\alpha_i, \varepsilon_i, v_i) = d_0 \sqrt{\tan^2 \alpha_i + \left(\frac{\tan \varepsilon_i}{\cos \alpha_i} - \frac{gd_0}{2v_i^2 \cos^2 \alpha_i \cos^2 \varepsilon_i} - \frac{h_0}{d_0} \right)^2}$$

und

$$\sqrt{\frac{f^2(\alpha_i, \varepsilon_i, v_i)}{d_0^2} - \tan^2 \alpha_i} = \frac{\tan \varepsilon_i}{\cos \alpha_i} - \frac{gd_0}{2v_i^2 \cos^2 \alpha_i \cos^2 \varepsilon_i} - \frac{h_0}{d_0}$$

und aus Gründen der Programmierfreundlichkeit führen wir folgende verkürzte Schreibweise ein:

$$f_\alpha(\alpha_i, \varepsilon_i, v_i) = \frac{d_0^2}{f(\alpha_i, \varepsilon_i, v_i)} \tan \alpha_i \left[\frac{1}{\cos^2 \alpha_i} + \left(\sqrt{\frac{f^2(\alpha_i, \varepsilon_i, v_i)}{d_0^2} - \tan^2 \alpha_i} + \frac{h_0}{d_0} \right) \times \sqrt{\frac{f^2(\alpha_i, \varepsilon_i, v_i)}{d_0^2} - \tan^2 \alpha_i} \right].$$

In Abb. 3 ist zunächst das schrittweise Ergebnis für f_α dargestellt. Die Konvergenz in der Nähe der Singularität ist aufgrund der Nichtdifferenzierbarkeit an dieser Stelle eingeschränkt.

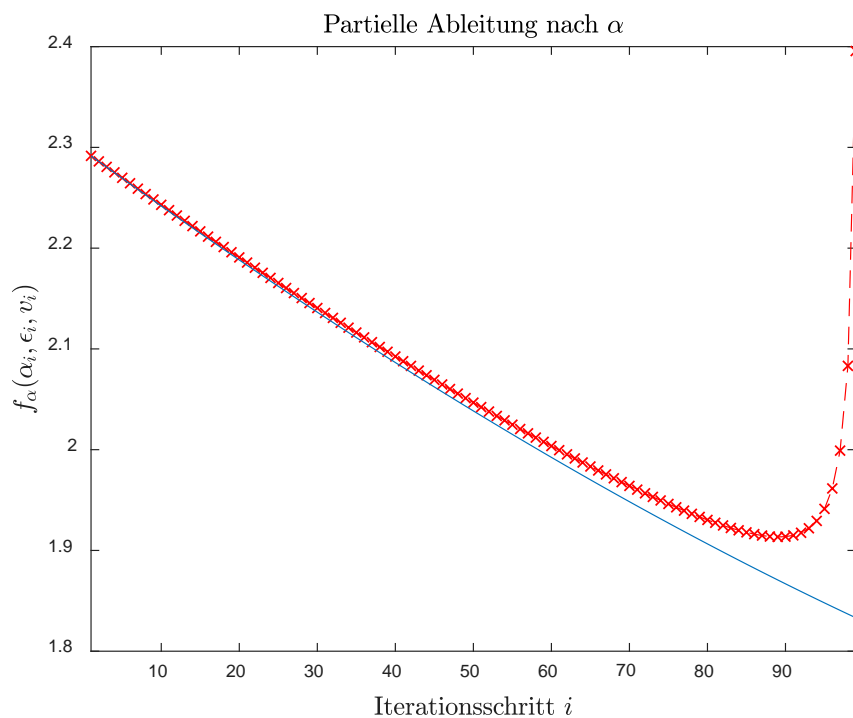


Abbildung 3. Die partielle Ableitung im Azimut wird bei Annäherung an die Singularität zunehmend ungenau

Abb. 4 zeigt das Ergebnis für f_ε , wobei die Konvergenz in Elevation wesentlich besser ist:

Mathematikaufgabe 143

$$f_\varepsilon(\alpha_i, \varepsilon_i, v_i) = \frac{d_0^2}{f(\alpha_i, \varepsilon_i, v_i)} \frac{1}{\cos^2 \varepsilon_i} \left(\frac{1}{\cos \alpha_i} - \frac{gd_0 \tan \varepsilon_i}{v_i^2 \cos^2 \alpha_i} \right) \sqrt{\frac{f^2(\alpha_i, \varepsilon_i, v_i)}{d_0^2} - \tan^2 \alpha_i}.$$

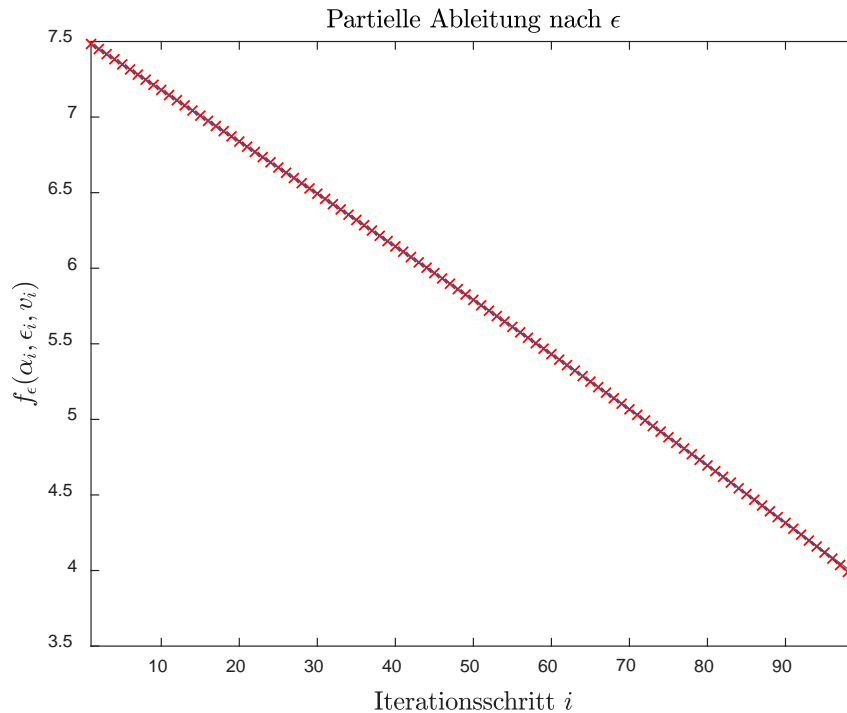


Abbildung 4. Die partielle Ableitung in Elevation weist in der Umgebung der Singularität einen geringen Fehler auf

Es verbleibt die partielle Ableitung nach v ,

$$f_v(\alpha_i, \varepsilon_i, v_i) = \frac{d_0^2}{f(\alpha_i, \varepsilon_i, v_i)} \frac{gd_0}{v_i^3 \cos^2 \alpha_i \cos^2 \varepsilon_i} \sqrt{\frac{f^2(\alpha_i, \varepsilon_i, v_i)}{d_0^2} - \tan^2 \alpha_i},$$

die in Abb. 5 dargestellt ist. Auch hier ist die Annäherung an die Singularität relativ unproblematisch.

Um das Grenzwertverhalten auszuloten, führen wir eine Differenzierbarkeitsbetrachtung in der Umgebung der Singularität durch und berechnen dazu die partiellen Ableitungen

$$f_\alpha(\Delta\alpha, \varepsilon_0 + \Delta\varepsilon, v_0 + \Delta v) = \frac{d_0 \tan \Delta\alpha}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right)^2}} \times \left[\frac{1}{\cos^2 \Delta\alpha} + \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \right) \right] \times \left[\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right],$$

Mathematikaufgabe 143

$$f_\varepsilon(\Delta\alpha, \varepsilon_0 + \Delta\varepsilon, v_0 + \Delta v) = \frac{d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right)^2}} \times (1 + \tan^2(\varepsilon_0 + \Delta\varepsilon)) \left(\frac{1}{\cos \Delta\alpha} - \frac{gd_0 \tan(\varepsilon_0 + \Delta\varepsilon)}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \right) \times \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right),$$

$$f_v(\Delta\alpha, \varepsilon_0 + \Delta\varepsilon, v_0 + \Delta v) = \frac{d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right)^2}} \times \frac{gd_0}{(v_0 + \Delta v)^3 \cos^2 \Delta\alpha \cos^2(\varepsilon_0 + \Delta\varepsilon)} \times \left(\frac{\tan(\varepsilon_0 + \Delta\varepsilon)}{\cos \Delta\alpha} - \frac{gd_0(1 + \tan^2(\varepsilon_0 + \Delta\varepsilon))}{2(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} - \frac{h_0}{d_0} \right).$$

Mit den Substitutionen

$$\tan(\varepsilon_0 + \Delta\varepsilon) = \frac{\tan \varepsilon_0 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon \tan \varepsilon_0} = \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon}$$

und

$$1 + \tan^2(\varepsilon_0 + \Delta\varepsilon) = \frac{(1 - \tan \Delta\varepsilon \tan \varepsilon_0)^2 + (\tan \varepsilon_0 + \tan \Delta\varepsilon)^2}{(1 - \tan \Delta\varepsilon \tan \varepsilon_0)^2} = 2 \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2}$$

vereinfachen sich die Ableitungen im Minimum zu

$$f_\alpha(\Delta\alpha, \Delta\varepsilon, \Delta v) = \frac{d_0 \tan \Delta\alpha}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right)^2}} \times \left[\frac{1}{\cos^2 \Delta\alpha} + \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} \right) \times \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right) \right],$$

Mathematikaufgabe 143

$$f_{\varepsilon}(\Delta\alpha, \Delta\varepsilon, \Delta v) = \frac{2d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right)^2}} \times \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} \left(\frac{1}{\cos \Delta\alpha} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} \right) \times \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right),$$

$$f_v(\Delta\alpha, \Delta\varepsilon, \Delta v) = \frac{d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right)^2}} \times \frac{2gd_0}{(v_0 + \Delta v)^3 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} \left(\frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} - \frac{gd_0}{(v_0 + \Delta v)^2 \cos^2 \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} - \frac{h_0}{d_0} \right).$$

Des weiteren lassen sich alle Ableitungen mittels der Funktion

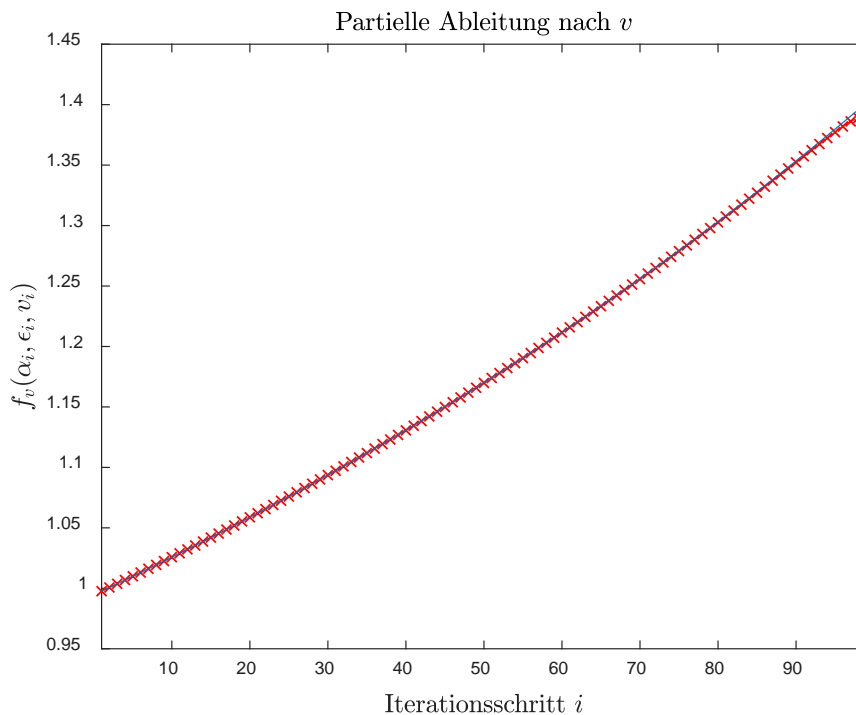


Abbildung 5. Die partielle Ableitung nach v hat Differenzierbarkeitsprobleme in der Umgebung der Singularität

Mathematikaufgabe 143

$$\begin{aligned}\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) &= \frac{1}{\cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} \left(1 - \frac{gd_0}{v_0^2 (1 + \Delta v/v_0)^2 \cos \Delta\alpha (1 - \tan \Delta\varepsilon)} \right) \\ &\approx (1 + \Delta\varepsilon)^2 \left(1 - (1 + \Delta\varepsilon) \left(1 - \frac{h_0}{d_0} \right) \left(1 - 2 \frac{\Delta v}{v_0} \right) \right)\end{aligned}$$

noch einmal vereinfachen:

$$\begin{aligned}f_\alpha(\Delta\alpha, \Delta\varepsilon, \Delta v) &= \frac{d_0 \tan \Delta\alpha}{\sqrt{\tan^2 \Delta\alpha + \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right)^2}} \\ &\quad \times \left[\frac{1}{\cos^2 \Delta\alpha} + \phi(\Delta\alpha, \Delta\varepsilon, \Delta v) \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right) \right],\end{aligned}$$

$$\begin{aligned}f_\varepsilon(\Delta\alpha, \Delta\varepsilon, \Delta v) &= \frac{2d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right)^2}} \frac{1}{\cos \Delta\alpha} \\ &\quad \times \left(1 - \frac{gd_0}{(v_0 + \Delta v)^2 \cos \Delta\alpha} \frac{1 + \tan \Delta\varepsilon}{1 - \tan \Delta\varepsilon} \right) \\ &\quad \times \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right),\end{aligned}$$

$$\begin{aligned}f_v(\Delta\alpha, \Delta\varepsilon, \Delta v) &= \frac{2d_0}{\sqrt{\tan^2 \Delta\alpha + \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right)^2}} \frac{gd_0}{(v_0 + \Delta v)^3 \cos^2 \Delta\alpha} \\ &\quad \times \frac{1 + \tan \Delta\varepsilon}{(1 - \tan \Delta\varepsilon)^2} \left(\phi(\Delta\alpha, \Delta\varepsilon, \Delta v) - \frac{h_0}{d_0} \right).\end{aligned}$$

Die klassische Berechnung ergibt im Zielmittelpunkt folgende Ergebnisse:

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\alpha(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 1,8332;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\varepsilon(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 3,9715;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_v(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 1,4014,$$

wohingegen sich mit der Näherungsrechnung zum Teil größere Abweichungen ergeben,

Mathematikaufgabe 143

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\alpha(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 2,7003;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\varepsilon(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 3,8879;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_v(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 1,3719.$$

Das neuronale Netzwerk hingegen weist noch einmal leicht unterschiedliche Werte auf:

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\alpha(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 2,3956;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_\varepsilon(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 3,9223;$$

$$\lim_{\Delta\alpha, \Delta\varepsilon, \Delta v \rightarrow 0} f_v(\Delta\alpha, \pi/4 + \Delta\varepsilon, v_0 + \Delta v) = 1,3841.$$

Im vorletzten Iterationsschritt erreicht die Richtungsableitung einen Wert von 0,0289, während das neuronale Netzwerk den Abstand $f(\alpha_{n-1}, \varepsilon_{n-1}, v_{n-1}) = 0,0221$ annimmt und uns damit ziemlich gut in den Zielmittelpunkt führt.

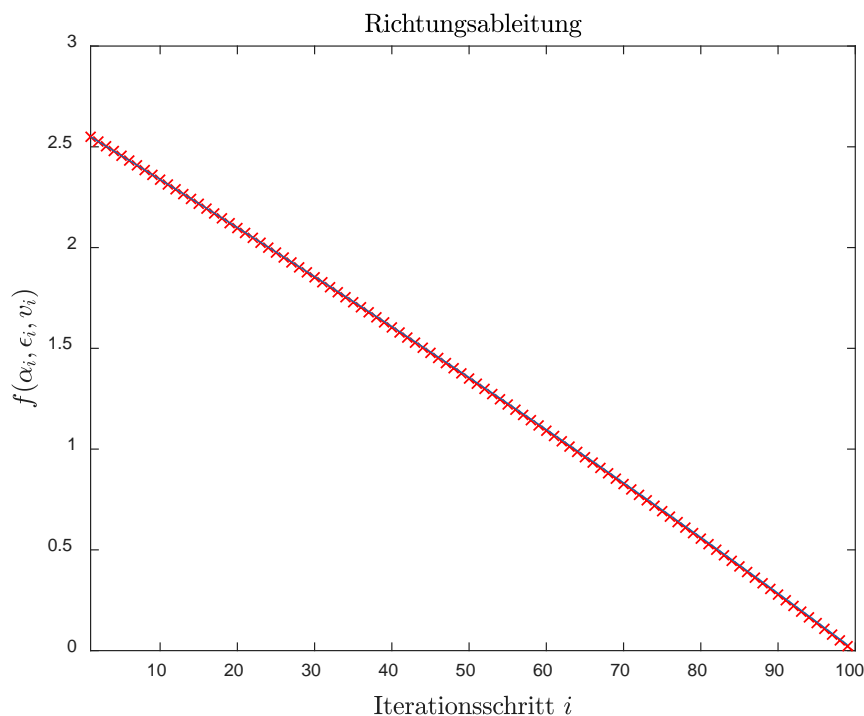


Abbildung 6. Die Richtungsableitung führt nach 100 Iterationsschritten ins Ziel

Mit diesen Ergebnissen können wir, wie in Abb. 6 gezeigt, die Richtungsableitung bestimmen. Das neuronale Netzwerk führt dabei die Hand des Schützen – falls dieser vorher ausgiebig trainiert hat –, noch ehe er das Projektil losläßt, sicher ins Ziel.

In Abb. 7 ist eine dreidimensionale Untermenge der vierdimensionalen Hyperfläche für konstantes v_0 graphisch dargestellt. Abb. 8 zeigt ein asymmetrisches Höhenlinienprofil.

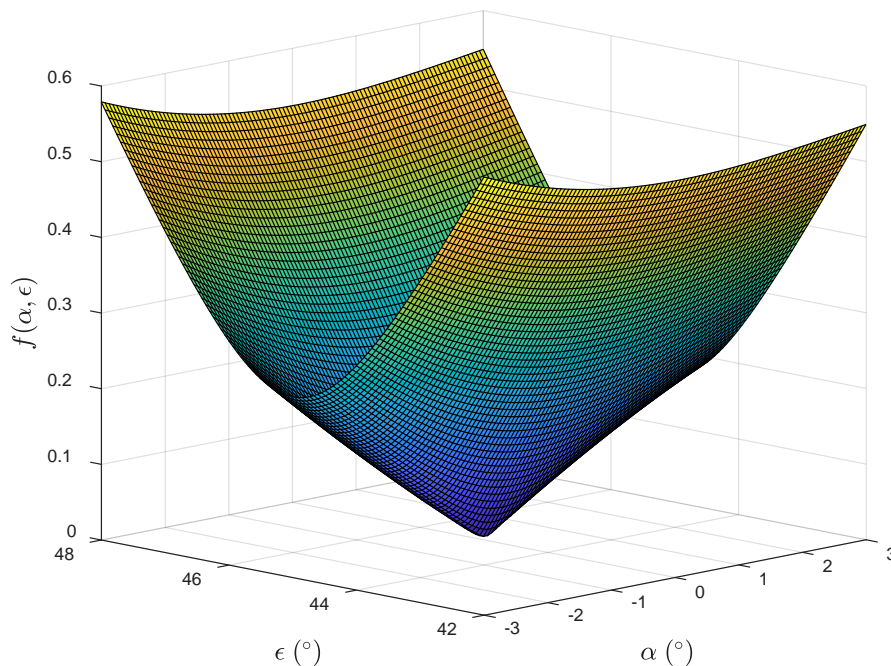


Abbildung 7. Oberflächendiagramm des schiefen Wurfs bei idealer Wurfgeschwindigkeit

Es empfiehlt sich also, bei der Auslegung eines rekurrenten neuronalen Netzwerks folgende vier Schritte zu berücksichtigen:

1. Man benötigt ein mathematisches Modell, mit dem man für einen späteren Anwendungslauf die erfolgreichen Trainingsgewichte ermitteln kann.
2. Falls die n Gleichungen mit n Unbekannten, welche die Minimalbedingung erfüllen, analytisch nicht lösbar sind, muß man die optimalen Parameterkombinationen mit Hilfe einer Näherungsrechnung herauszufinden. Die ideale Fehlerquadratsumme muß im Ergebnis null liefern.
3. In der Regel müssen alle partiellen Ableitungen bis zur zweiten Ordnung analytisch berechnet oder numerisch bestimmt werden, um mit den richtigen Anfangswerten auf der Hyperfläche starten zu können.
4. Mittels der erfolgreich bewerteten Parameterkombinationen aus der Trainingsphase können durch paralleles Prozessieren die Netzwerkgewichte bestimmt werden, womit das neuronale Netzwerk nun für jede beliebige Parameterkombination dem Gradienten der Trajektorie auf der Hyperfläche bis ins Minimum folgen kann. Es kann durchaus sein, daß Anfangswerte, die nicht auf der Hyperfläche liegen, gewisse seltsame Effekte hervorrufen.

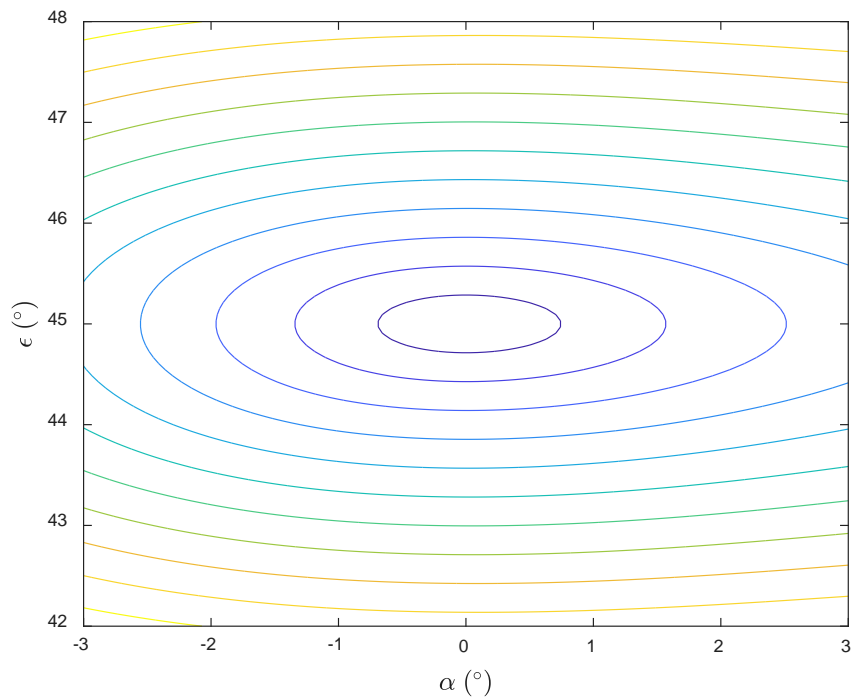


Abbildung 8. Asymmetrie im Höhenlinienprofil des schiefen Wurfs

Beim Speerwerfen oder Bogenschießen müssen die endgültigen Wurfgrößen vom Gehirn bereits ermittelt sein, bevor das Geschöß losgelassen wird, weil jedes nicht gelenkte Geschöß unmittelbar danach ballistisch weiterfliegt, d.h. nicht mehr beeinflusst werden kann. Unser Wurfarm hat also je nach Armlänge weniger als einen Meter zur Verfügung, um durch neuronalen Abgleich die endgültigen Wurfparameter zu bestimmen. Außerdem steht unserem gedanklichen Ablauf kein Taschenrechner zur Verfügung, damit wir uns diese Parameter ausrechnen könnten, wir sind rein auf Augenpeilungen und Schätzungen angewiesen. Jeder Treffer, den wir durch ausgiebiges Training erzielen, sollte bei einem ambitionierten Schützen von einem Freudentaumel⁶ begleitet sein, der in der Großhirnrinde dafür sorgt, daß dieses Erlebnis immer in angenehmer Erinnerung bleibt.⁷ Beim nächsten Schuß oder Wurf wird das Gehirn dann versuchen, den bereits erfolgreichen Wurf noch einmal zu reproduzieren, indem es sich an die alten Wurfparameter erinnert, es arbeitet also nach demselben Soll-Ist-Vergleich wie die Methode der kleinsten Quadrate, wenn auch auf etwas andere Art.

⁶ Wegen der Endorphinausschüttung

⁷ Genaueres kann die Wissenschaft dazu bislang nicht sagen.

Anhang

```
% Rekurrentes neuronales Netz Schiefer Wurf
clear all

n = 100;
% Konstante Bahnparameter des idealen Wurfs
g = 10; % [m/s^2]
epsilon_0 = 45/180*pi; % [rad]
h0 = 2; % [m]
d0 = 10; % [m]
v0 = sqrt(g*d0^2/(d0*sin(2*epsilon_0)-2*h0*cos(epsilon_0)^2));

% Anfangsbedingungen
alpha(1) = 3/180*pi;
epsilon(1) = 48/180*pi;
v(1) = 13; % [m/s]

% Schrittweiten
delta_alpha = alpha(1)/(n-1);
delta_epsilon = (epsilon(1)-epsilon_0)/(n-1);
delta_v = (v(1)-v0)/(n-1);

% Partielle Ableitungen im Minimum
phi = 1/cos(delta_alpha)*(1+tan(delta_epsilon)/(1-tan(delta_epsilon))-
g*d0*(1+tan(delta_alpha)^2)/(v0+delta_v)^2*(1+tan(delta_epsilon)/(1-
tan(delta_epsilon))^2));
zaehler = d0*tan(delta_alpha)*(1/cos(delta_alpha)^2+phi*(phi-h0/d0));
nenner = sqrt(tan(delta_alpha)^2+(phi-h0/d0)^2);
falpha0 = zaehler/nenner
mean_term_epsilon = 1/cos(delta_alpha)*(1+tan(delta_epsilon)/(1-
tan(delta_epsilon))^2)*(1-
g*d0/(v0+delta_v)^2/cos(delta_alpha)*(1+tan(delta_epsilon)/(1-
tan(delta_epsilon))));
fepsilon0 = 2*d0/nenner*(phi-h0/d0)*mean_term_epsilon
mean_term_v = (1+tan(delta_epsilon)/(1-
tan(delta_epsilon))^2)*g*d0/(v0+delta_v)^3/cos(delta_alpha);
fv = 2*d0/nenner*(phi-h0/d0)*mean_term_v

% Parameterintervalle
for i=1:n
    alpha(i) = alpha(1) - (i-1)*delta_alpha;
    alpha_grad(i) = alpha(i)*180/pi;
    epsilon(i) = epsilon(1) - (i-1)*delta_epsilon;
    epsilon_grad(i) = epsilon(i)*180/pi;
    v(i) = v(1) - (i-1)*delta_v;
end

figure(1)
plot(alpha_grad, 'b')
title('Aziumutwinkel in Grad', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$\alpha_i$', 'interpreter', 'latex')

figure(2)
plot(epsilon_grad, 'g')
title('Elevationswinkel in Grad', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$\epsilon_i$', 'interpreter', 'latex')
```

Mathematikaufgabe 143

```
figure(3)
plot(v, 'r')
title('Geschwindigkeit [m/s]', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$v_i$', 'interpreter', 'latex')

% Neuronales Netzwerk
f(1) = d0*sqrt( tan(alpha(1))^2+( tan(epsilon(1))/cos(alpha(1)) -
g*d0/2/v(1)^2/cos(alpha(1))^2/cos(epsilon(1))^2 - h0/d0 )^2 );
for i=1:n-1
    w(i)=sqrt(f(i)^2/d0^2-tan(alpha(i))^2);
    f_alpha(i) =
d0^2*tan(alpha(i))/f(i)*(1/cos(alpha(i))^2+(w(i)+h0/d0)*w(i));
    f_epsilon(i) = d0^2/f(i)*1/cos(epsilon(i))^2*(1/cos(alpha(i)) -
g*d0*tan(epsilon(i))/v(i)^2/cos(alpha(i))^2)*w(i);
    f_v(i) = g*d0^3*w(i)/f(i)/cos(epsilon(i))^2/cos(alpha(i))^2/v(i)^3;
    f(i+1) = f(i) - f_alpha(i)*delta_alpha - f_epsilon(i)*delta_epsilon -
f_v(i)*delta_v;
end

% Klassische Berechnung
for i=1:n-1
    f_c(i) = d0*sqrt( tan(alpha(i))^2+( tan(epsilon(i))/cos(alpha(i)) -
g*d0/2/v(i)^2/cos(alpha(i))^2/cos(epsilon(i))^2 - h0/d0 )^2 );
    w(i)=sqrt(f_c(i)^2/d0^2-tan(alpha(i))^2);
    f_alpha_c(i) =
d0^2*tan(alpha(i))/f_c(i)*(1/cos(alpha(i))^2+(w(i)+h0/d0)*w(i));
    f_epsilon_c(i) = d0^2/f_c(i)*1/cos(epsilon(i))^2*(1/cos(alpha(i)) -
g*d0*tan(epsilon(i))/v(i)^2/cos(alpha(i))^2)*w(i);
    f_v_c(i) =
g*d0^3*w(i)/f_c(i)/(cos(epsilon(i))^2*cos(alpha(i))^2*v(i)^3);
end

figure(4)
plot(f, '--xr')
title('Richtungsableitung', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$f(\alpha_i, \epsilon_i, v_i)$', 'interpreter', 'latex')
xlim([1 100])
ylim([0 3])
hold on
plot(f_c)

figure(5)
plot(f_alpha, '--xr')
hold on
plot(f_alpha_c)
title('Partielle Ableitung nach $\alpha$', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$f_{\alpha}(\alpha_i, \epsilon_i, v_i)$', 'interpreter', 'latex')
xlim([1 99])

figure(6)
plot(f_epsilon, '--xr')
hold on
plot(f_epsilon_c)
title('Partielle Ableitung nach $\epsilon$', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$f_{\epsilon}(\alpha_i, \epsilon_i, v_i)$', 'interpreter', 'latex')
xlim([1 99])
```

```
figure(7)
plot(f_v, '--xr')
hold on
plot(f_v_c)
title('Partielle Ableitung nach $v$', 'interpreter', 'latex')
xlabel('Iterationsschritt $i$', 'interpreter', 'latex')
ylabel('$f_v(\alpha_i, \epsilon_i, v_i)$', 'interpreter', 'latex')
xlim([1 99])

% Surface Plot der Richtungsableitung mit v=const
k=100;
n=100;
alpha(1) = -3/180*pi;
alpha(100) = 3/180*pi;
epsilon(1) = 42/180*pi;
epsilon(100) = 48/180*pi;
delta_alpha = (alpha(100)-alpha(1))/(n-1);
delta_epsilon = (epsilon(100)-epsilon(1))/(n-1);
v(k) = v0;
for i=1:n
    alpha(i) = alpha(1) + (i-1)*delta_alpha;
    alpha_grad(i) = alpha(i)*180/pi;
    for j=1:n
        epsilon(j) = epsilon(1) + (j-1)*delta_epsilon;
        epsilon_grad(j) = epsilon(j)*180/pi;
        flaeche(i,j) = d0*sqrt( tan(alpha(i))^2+(
tan(epsilon(j))/cos(alpha(i)) -
g*d0/2/v(k)^2/cos(alpha(i))^2/cos(epsilon(j))^2 - h0/d0 )^2 );
    end
end

figure(8)
surf(alpha_grad, epsilon_grad, flaeche)
xlim([-3 3])
xlabel('$\alpha\;(^{\circ})$', 'interpreter', 'latex')
ylabel('$\epsilon\;(^{\circ})$', 'interpreter', 'latex')
zlabel('$f(\alpha, \epsilon)$', 'interpreter', 'latex')

figure(9)
contour(alpha_grad, epsilon_grad, flaeche)
xlim([-3 3])
xlabel('$\alpha\;(^{\circ})$', 'interpreter', 'latex')
ylabel('$\epsilon\;(^{\circ})$', 'interpreter', 'latex')
```