Mehrdimensionale Optimierung ohne Nebenbedingungen

Sonja Biedermann, Felix Freynschlag, Bernhard Hayden,
Stepan Kharin, Christoph Pressler

23. November 2015

Inhaltsverzeichnis

1 Das Nelder-Mead-Verfahren

1.1 Einführung

Das Nelder-Mead-Verfahren (auch Downhill-Simplex genannt) verfolgt einen geometrischen Ansatz zur iterativen Optimierung: ein Simplex bestehend aus $N+1$ Punkten (im Falle $N=2$, dem zweidimensionalen Raum, also ein Dreieck) wird fortwährend so transformiert, sodass sich der Simplex dem Minimum nähert, um welches er sich dann zusammenzieht.

1.2 Funktionsweise

Der Algorithmus besteht aus drei Phasen, die in jeder Iteration durchlaufen werden:

1. Ordering
2. Centroid
3. Transformation

Ordering Zu Beginn besitzen wir $N+1$ Punkte, die anhand irgendwelche Kriterien (durchaus auch zufällig) gewählt wurden. Diese müssen wir nun anhand ihrer Güte ordnen, also sortieren. Im Falle des zweidimensionalen Raumes erhalten wir also die drei mnemonisch benannten Punkte B (*best*), G (*good*) und W (*worst*). B ist im Falle eine Minimierung der Punkt, dessen Funktionswert am geringsten ist, G der mit dem zweit geringsten Wert, und so weiter.

Centroid Als nächstes berechnen wir den Mittelpunkt der beiden besseren Punkte

$$M = \frac{B+G}{2}$$

und nehmen diesen als Basis für unsere weiteren Berechnungen.

Transformation Nun reflektieren wir den schlechtesten Punkt W am Mittelpunkt

$$R = M + \alpha(M - W)$$

und prüfen, ob dieser Punkt r die Relation $f(B) \leq f(R) \leq f(W)$ erfüllt.

Wenn ja, ersetzen wir den schlechtesten Punkt mit R und brechen die jetzige Iteration ab. Diesen Vorgang nennt man auch das „Akzeptieren“ eines Punktes.

Wenn nein, prüfen wir, ob der reflektierte Punkt R besser als unser bisher bester Punkt ist, also ob $f(R) < f(B)$ gilt. Wenn dem so ist, expandieren wir den Punkt R testweise weiter:

$$E = M + \gamma(M - W)$$

und prüfen wiederum, ob dieser Punkt E besser als unser reflektierte Punkt R ist. Wenn ja, akzeptieren wir E. Wenn nein, akzeptieren wir R.

Gilt $f(R) \geq f(G)$, so ist der reflektierte Punkt kein guter Kandidat. Wir müssen nun die zwei möglichen kontrahierten Punkte C_1, C_2 berechnen. Gilt $f(R) < f(W)$, berechnen wir den äußeren kontrahierten Punkt C_2, ansonsten den inneren kontrahierten Punkt C_2.

$$\begin{aligned} C_1 &= M + \rho(R - M) \\ C_2 &= M + \rho(W - M) \end{aligned}$$

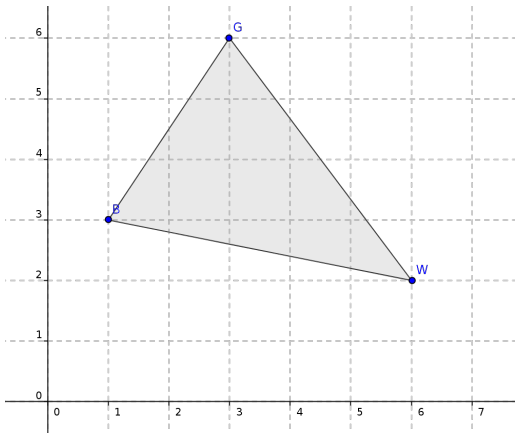
Sind die berechneten kontrahierten Punkte besser als die zur Berechnung verwendeten Punkte (R und W respektive), so werden diese akzeptiert. Ist dem nicht so, müssen wir zum nächsten Schritt, der Komprimierung, fortfahren. Bei der Komprimierung werden jediglich die zwei schlechteren Punkte G und W in Richtung des besten Punktes B gezogen.

$$\begin{aligned} G &= B + \sigma(B - G) \\ W &= B + \sigma(B - W) \end{aligned}$$

Diese Schritte werden fortgeführt, bis eine Abbruchbedingung eintritt, wie etwa dass der Unterschied zwischen dem besten Funktionswert $f(B)$ und dem schlechtesten Funktionswert $f(W)$ ein gegebenes ε unterschreitet, oder der Umfang des Simplex einen gegebenen Wert unterschreitet.

1.3 Erläuterung der einzelnen Transformationsschritte

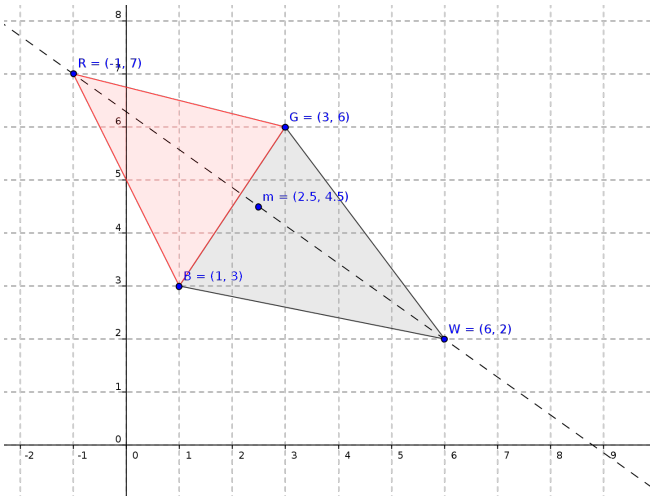
Folgend werden die einzelnen Transformationsschritte anhand eines Dreiecks als Simplex vorgeführt und erläutert. Für unsere Koeffizienten wählen wir die Werte $\alpha = 1$, $\gamma = 2$, $\rho = \frac{1}{2}$, $\sigma = \frac{1}{2}$. Unser Dreieck $\triangle BGW$ ist definiert mit $B = (1, 3)$, $G = (4, 6)$, $W = (6, 2)$.



1.3.1 Reflektion

Wir reflektieren den schlechtesten Punkt an dem Mittelpunkt in die vermeintlich bessere Richtung.

$$R = M + M - W = 2M - W = \begin{pmatrix} 5 \\ 9 \end{pmatrix} - \begin{pmatrix} 6 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 7 \end{pmatrix}$$

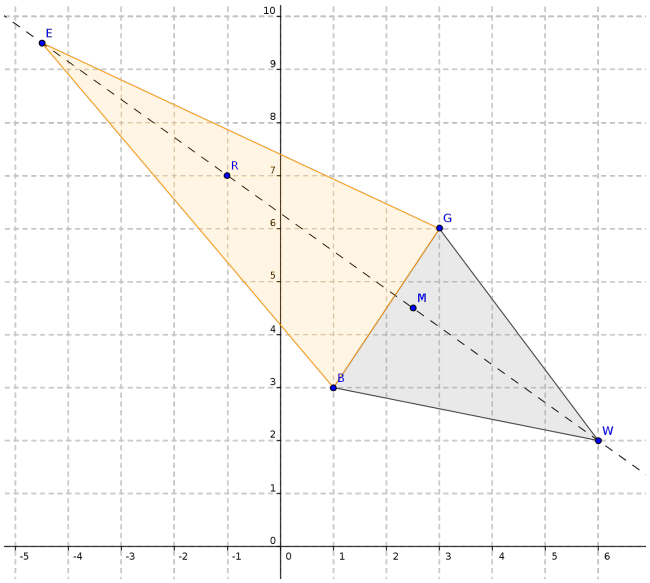


Dies entspricht dem „Heruntertaumeln“ des Simplex, welches die Optimierung in den ersten Iterationsschritten antreibt.

1.3.2 Expansion

Nehmen wir an, dass $f(R)$ besser als unser bis jetzt bester Wert ist. Wir expandieren also weiter:

$$E = M + 2(M - W) = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} + \begin{pmatrix} -7 \\ 5 \end{pmatrix} = \begin{pmatrix} -4.5 \\ 9.5 \end{pmatrix}$$

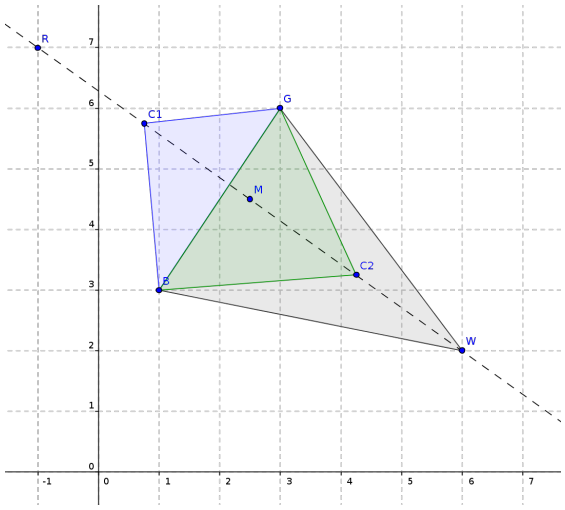


Dies entspricht einem *greedy approach* – dieser Schritt ist nicht notwendig, minimiert aber zusätzlich die Anzahl der nötigen Iterationen.

1.3.3 Kontraktion

Gehen wir jetzt vom Gegenteil aus – Punkt R war also kein geeigneter Punkt und Punkt E wurde gar nicht berechnet – so müssen wir die beiden Kontraktionspunkte C_1, C_2 berechnen.

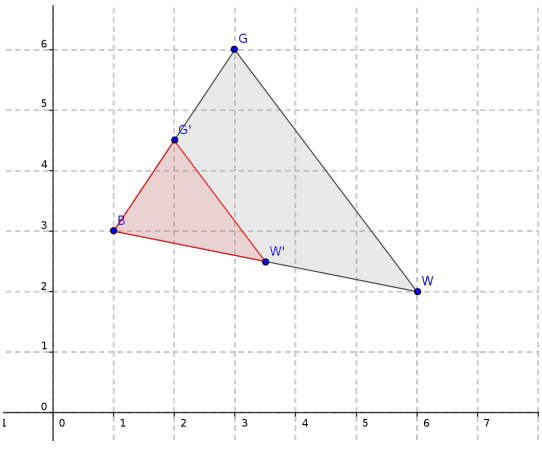
$$\begin{aligned} C_1 &= M + \frac{R-M}{2} = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} + \frac{\begin{pmatrix} -1 \\ 7 \end{pmatrix} - \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix}}{2} = \begin{pmatrix} -1.75 \\ 1.25 \end{pmatrix} \\ C_2 &= M + \frac{W-M}{2} = \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix} + \frac{\begin{pmatrix} 6 \\ 2 \end{pmatrix} - \begin{pmatrix} 2.5 \\ 4.5 \end{pmatrix}}{2} = \begin{pmatrix} 4.25 \\ 1.25 \end{pmatrix} \end{aligned}$$



1.3.4 Komprimierung

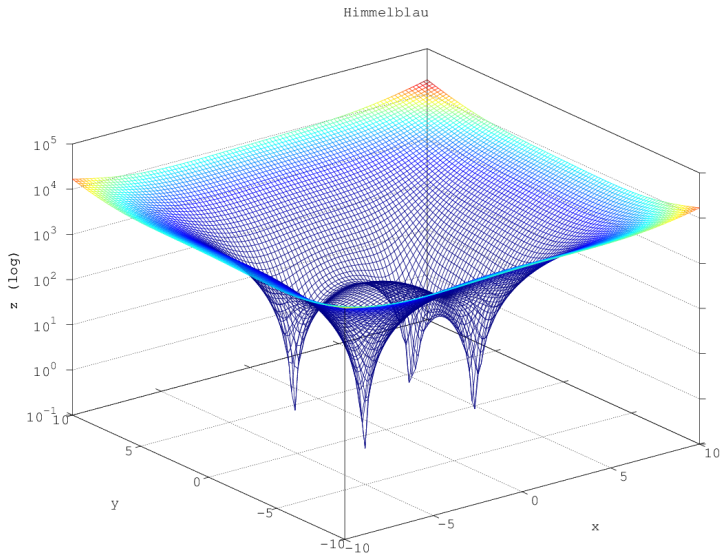
Wir komprimieren die Punkte G und W in Richtung B.

$$\begin{aligned} G' &= B + \frac{G-B}{2} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \begin{pmatrix} 1 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 2 \\ 4.5 \end{pmatrix} \\ W' &= B + \frac{W-B}{2} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \begin{pmatrix} 2.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 2.5 \end{pmatrix} \end{aligned}$$



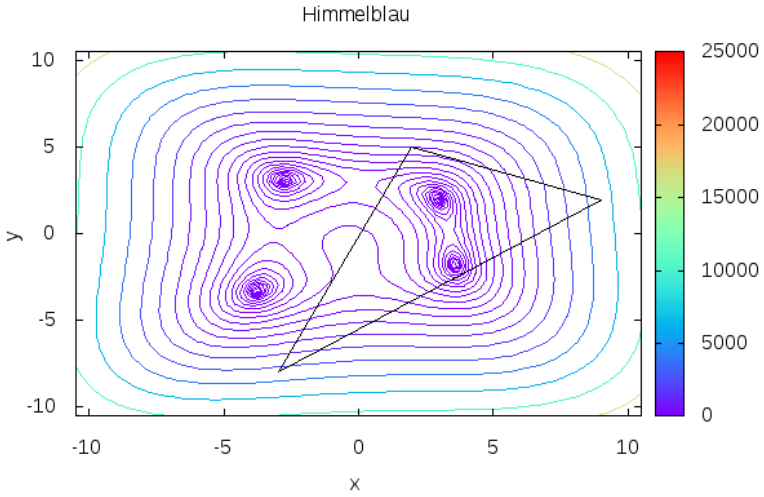
Dies entspricht dem Zusammenziehen des Simplex' in den letzten Iterationsschritten des Algorithmus'.

1.4 Ein Beispiel: die Himmelblau-Funktion



Die Himmelblau-Funktion hat 4 gleiche lokale Minima. Wir wollen unseren Algorithmus auf sie loslassen, um eines davon zu finden. Als zufällige Startpunkte wählen wir

$$\begin{aligned} P_1 &= (2, 5) \\ P_2 &= (9, 2) \\ P_3 &= (-3, -8) \end{aligned}$$



fehlend

2 Das Gradientenverfahren

2.1 Die Idee und der Zweck des Gradientenverfahren

Grundsätzlich ist das Gradientenverfahren ein Optimierungsverfahren, welches das Minimum einer mehrdimensionalen Funktion errechnen. Der Gradient ($\nabla f(x,y)$) einer Funktion $f(x)$ zeigt, aus später gezeigten (HIER VERWEIS EINFÜGEN) Gründen, immer in die Richtung des steilsten Anstiegs, folglich zeigt -$\nabla f(x,y)$ in die Richtung des stärksten Abstiegs. Das Gradientenverfahren macht sich dies zunutze und man "wandert" eine gewisse Dauer in diese Richtung, wie "lange" man in die errechnete Richtung "wandert" hängt unter anderem auch von der gewählten Methode ab.

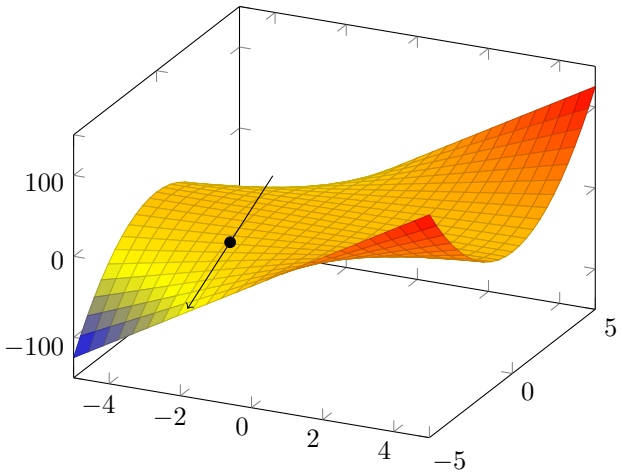


Abbildung 1: Hier zu sehen ist -$\vec{\nabla} f(-2,-2)$ der Funktion $f(x,y) = x * y^2$

2.2 Der Ablauf des Gradientenverfahrens

1. Es wird ein beliebiger Startpunkt $P = (x,y)$ ausgewählt.
2. Es wird die Ableitung der Funktion $f(x,y)$, also $\nabla f(x)$ gebildet. Sollte dieser Gradient bereits den Wert null haben sind wir bereits fertig und haben ein lokales Minimum.
3. Der Punkt P wird in den Gradienten eingesetzt und der negative Wert berechnet, also $-\nabla f(x_P, y_P)$.
4. Um eine neuen Punkt $P^{[n+1]}$ zu finden stellen wir nun folgende Gleichung auf: $P^{[n+1]} = P^{[n]} + \alpha_{[n]} * s^{[n]}$. Nun haben wir die Richtung in die optimiert werden muss gefunden ($s^{[n]}$ ($= -\nabla f(x_P, y_P)$)), jetzt muss noch entschieden werden wie weit wir in diese Richtung "wandern" ($\alpha^{[n]}$, liegt zwischen 0 und 1) (Hinweis: n ist die aktuelle Iteration).

   Das kann man einerseits mit Hilfe eines fixen Werts machen, andererseits auch mit einer Folge (wie z.B $\frac{1}{\sqrt{n}}$) oder einer jedes mal neu errechneten Schrittweite. Zu beachten ist, dass die ausgewählte Folge auf jeden Fall divergent sein muss, da man sonst bis zu einer maximalen Entfernung "wandern" kann und die Wahrscheinlichkeit groß ist das Optimum nie zu erreichen. Sollte man jedes mal einen fixen Wert verliert man zwar den Aufwand für die Berechnung der Schrittweite, jedoch ist die Wahrscheinlichkeit hoch, dass man viele Iterationen benötigt oder in einer endlosen Schleife festhängt.

   Es gibt verschieden Arten wie man die Schrittweite berechnen kann, im Folgenden werde ich die Schrittweitenbestimmung nach Armijo erklären. Dieses Verfahren ist eines der sogenannten

"line-search" Verfahren, das von mir gezeigte ist jedoch kein exaktes line-search Verfahren sondern lediglich eine einfache Heuristik.

Die Armijo-Bedingung: $\varphi(\alpha) \leq \varphi(0) + c * \alpha^{[n]} * \varphi(0)'$.

Wobei c eine Konstante zwischen null und eins ist und dazu dient das die Bedingung nicht zu restriktiv ist, $\varphi(\alpha) = f(P^{[n]} + \alpha^{[n]} s^{[n]})$ und $\varphi(0)' = \nabla f(P^n) * s^{[n]}$

5. Für den Anfang wird $\alpha = 1$ gewählt und errechnet ob die Armijo-Bedingung erfüllt ist - sollte sie erfüllt sein haben wir ein passende α gefunden, wenn nicht wird α mit einem β, welches ebenfalls zwischen null und eins liegt, multipliziert und wieder geprüft ob die Ungleichung erfüllt ist. (Hinweis: In der Praxis liefern die Werte $c = 0.01$ und $\beta = 0.9$ häufig gute Ergebnisse. (QUELLE: https://www.unibw.de/lrt1/gerdts/lehre/lpnlp/lp-nlp.pdf)

6. Wenn man einen neuen Punkt gefunden hat wird bei Punkt 3 fortgesetzt. Da es oft einen großen Rechenaufwand erfordert das genaue Minimum zu finden wird das Verfahren meist beendet wenn der Gradient einen akzeptablen Wert erreicht hat (So kann man als Abbruchbedingung z.B. die stärke des Anstiegs in einem Punkt (also den Betrag des Gradienten) nehmen).

2.3 Beispiele

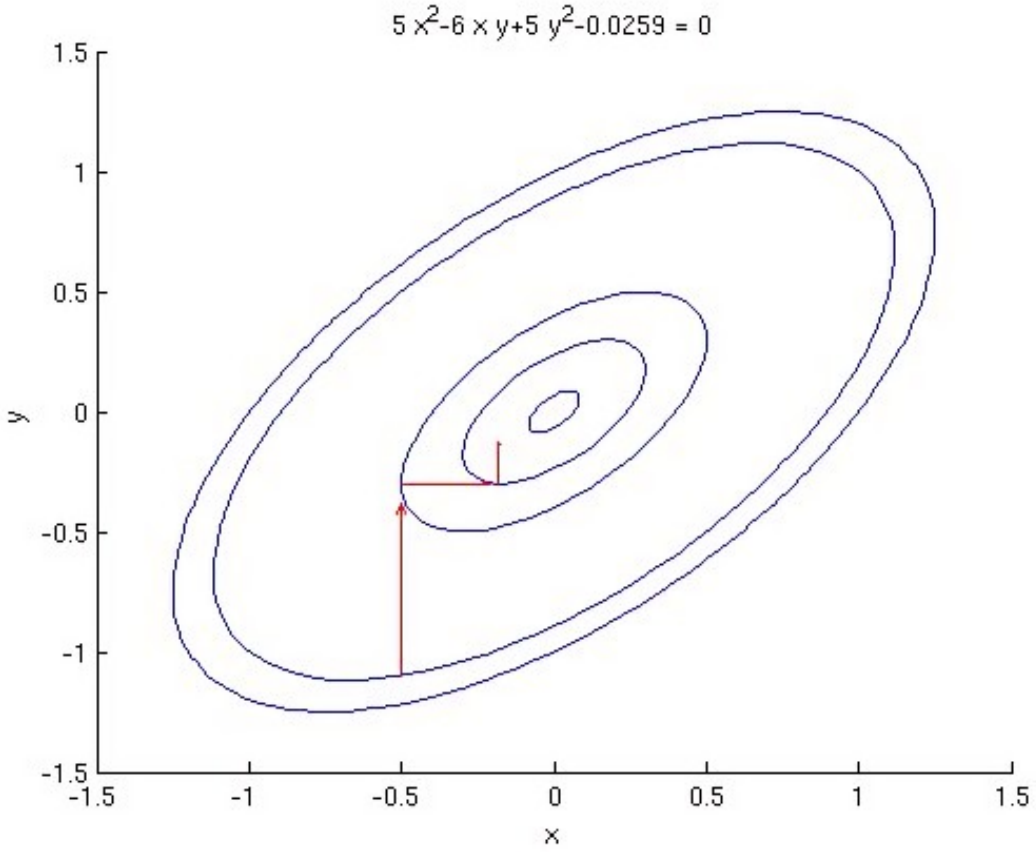
2.3.1 Beispiel 1

3 Die Koordinatenabstiegsmethode

3.1 Einführung

Bei der Koordinatenabstiegsmethode wird bei jedem Schritt eine Koordinate i gewählt, nach der man optimiert. Es wird bei einem beliebigen Startpunkt begonnen und in jedem Iterationsschritt setzt man alle Koordinaten des aktuellen Punktes in die Funktion ein , nur die Koordinate , nach der man minimieren will , behält man als Variable bei.

Es wird immer in Richtung einer Achse minimiert und die Stelle , an der das Minimum ist , als neue Koordinate beibehalten und so gelangt man mit einer Anzahl an Iterationen zum Minimum der Funktion.



Wenn zum Beispiel die Funktion $f(x, y) = 2x^2 + 2xy + 1.5y^2$ und der Startpunkt $P = (1, 1)$ gegeben ist, so muss man, wenn entlang der x-Achse minimiert werden soll folgendes eingesetzt werden:

$$\begin{aligned} x &= 1 + 1d \\ y &= 1 + 0d \end{aligned}$$

Soll entlang der y-Achse optimiert werden, so muss folgendes eingesetzt werden:

$$\begin{aligned} x &= 1 + 0d \\ y &= 1 + 1d \end{aligned}$$

3.2 Beispiel

Wir möchten die Funktion $f(x, y) = 2x^2 + 2xy + 1.5y^2$ optimieren und wählen den Startpunkt $P_0 = (1, 1)$. Wir minimieren zuerst entlang der x-Achse und setzen ein

$$\begin{aligned} f(d) &= 2(1 + d)^2 + 2(1 + d) + 1.5 \\ &= 2d^2 + 6d + 5.5 \end{aligned}$$

Diese soeben erhaltene Funktion müssen wir nun noch minimieren.

$$\begin{aligned} f'(d) &= 4d + 6 \\ 4d + 6 &= 0 \\ d &= \frac{3}{2} \end{aligned}$$

Setzen wir dies in die Gleichung für die x-Koordinate ein und verwenden die alte y-Koordinate erhalten wir den neuen Punkt $P_1 = (-\frac{1}{2}, 1)$. Nun minimieren wir nach y, wir setzen also ein:

$$\begin{aligned} x &= -\frac{1}{2} + 0d \\ y &= 1 + 1d \end{aligned}$$

und erhalten

$$f(d) = 1.5d^2 - d + 6$$

Dies minimieren wir wiederum und erhalten die neue y-Koordinate $\frac{1}{3}$. Damit haben wir den neuen Punkt $(-\frac{1}{2}, \frac{1}{3})$. Dieser Vorgang wird wiederholt, bis sich die errechneten Werte nurnoch marginal unterscheiden. Danach wird abgebrochen.

3.2.1 Problemfälle

Zeigen Sie, dass die Koordinatenabstiegsmethode für die stetige Funktion

$$f(x, y) = \begin{cases} (x + y - 5)^2 + (x - y - 2)^2 & falls\, x \leq y \\ (x + y - 5)^2 + (x - y + 2)^2 & sonst \end{cases}$$

beginnend beim Startpunkt $(0, 0)$ nicht funktioniert.

Wir setzen ein:

$$\begin{aligned} x &= d \\ y &= 0 \end{aligned}$$

und erhalten die Funktion

$$\begin{aligned} f(d) &= (d - 5)^2 + (d - 2)^2 \\ &= 2d^2 - 14d + 29 \end{aligned}$$

welche ein Minimum an der Stelle $d = \frac{7}{2}$. Dies fortgesetzt für y erhalten wir den neuen Punkt $P = (\frac{7}{2}, \frac{7}{2})$. Bei weiteren Iterationen merken wir, dass d immer 0 beträgt – was bedeutet, dass sich der Punkt nicht weiterbewegt. Wir sitzen auf einem lokalen Minimum fest. Führen wir das mit dem anderen Zweig der Funktion ebenfalls durch, so sehen wir das gleiche Problem.