Verallgemeinerte Newton-Trajektorien in der globalen Optimierung

Dissertation
zur Erlangung des Doktorgrades
des Fachbereichs Mathematik
der Universität Hamburg

vorgelegt von
Leszek Bajorski
aus Jaroslaw

Hamburg
2000

Als Dissertation angenommen vom Fachbereich
Mathematik der Universität Hamburg

auf Grund der Gutachten von Prof. Dr. Klaus Glashoff
und Prof. Dr. Carl Geiger

Hamburg, den 26.04.2000

Prof. Dr. Ulrich Eckhardt
Dekan des Fachbereiches Mathematik

Inhaltsverzeichnis

Einführung **ix**

1 Trajektorien in der Optimierung **1**
1.1 Problemstellung 1
1.2 Trajektorien 4
1.2.1 Motivation und theoretischer Ansatz 4
1.2.2 Eigenschaften und Charakterisierung 5
1.2.3 Numerische Ansätze 8
1.3 Klassische Newton-Trajektorie 13
1.3.1 Zugang über eine Differentialgleichung 13
1.3.2 Zugang über ein nichtlineares Gleichungssystem 15
1.3.3 Zugang über eine vorgegebene Richtung 23
1.4 Richtungsfeld-Trajektorie 25
1.4.1 Zugang über ein Richtungsfeld 25
1.4.2 Zugang über ein nichtlineares Gleichungssystem 27
1.4.3 Richtungsfeld-Trajektorie mit mehreren vorgegebenen Startpunkten 30
1.4.4 Richtungsfeld-Trajektorie für restringierte Probleme 44

2 Suche nach den Trajektorienkomponenten **55**
2.1 Rekursive Konstruktion des Verfahrens 55
2.1.1 Newton-Blätter 55
2.1.2 Rekursive Konstruktion für klassische Newton-Trajektorien 58
2.1.3 Rekursive Konstruktion für Richtungsfeld-Trajektorien 64
2.2 Strategien zur Bestimmung der Gitterpunkte 70
2.2.1 Strategie der Berührungspunkte 70
2.2.2 Strategie des äquidistanten Netzes 75
2.2.3 Strategie des inäquidistanten Netzes 77

2.2.4 Vergleich der Gitterpunktstrategien 78
2.3 Graphentheoretische Betrachtung 79
2.4 Rekursionsalgorithmus 85
2.4.1 BFS Algorithmus (breadth first search) 85
2.4.2 DFS Algorithmus (depth first search) 89
2.4.3 Vergleich der vorgestellten Algorithmen 91

3 Ausgewählte Verfahren 93
3.1 Prinzip des Prädiktor-Korrektor-Verfahrens 93
3.1.1 Prädiktormethode 94
3.1.2 Korrektormethode 95
3.2 Kontinuerliches Newton-Verfahren (KNV) 96
3.2.1 Korrektor 96
3.2.2 Schrittlängensteuerung 99
3.3 Kontinuierliches Quasi-Newton-Verfahren (KQNV) 101
3.3.1 Broyden Aufdatierungsformel 102
3.3.2 Broyden Aufdatierungsformel bei der Trajektorienrekonstruktion 103
3.3.3 Fehlerkontrolle der aufdatierten Matrizen 106
3.3.4 QR-Aufdatierung 106
3.4 Ableitungsfreies Surrogate-Verfahren 109
3.4.1 Modellbildung 110
3.4.2 Modellanpassung 123

4 Numerische Ergebnisse 129
4.1 Testprobleme ohne Restriktionen 130
4.1.1 Zweidimensionale Testfunktionen 130
4.1.2 Mehrdimensionale Testfunktionen 135
4.2 Trajektorien 140
4.3 Gitterpunkt-Strategien 142
4.4 Verfahren 144
4.5 Testprobleme mit Restriktionen 145
4.5.1 Zweidimensionale Testfunktionen 145
4.5.2 Mehrdimensionale Testfunktionen 150
4.6 Restringierte Optimierung 152

Zusammenfassung und Ausblick 155

Literaturverzeichnis 157

Abbildungsverzeichnis

1.1 Beispieltrajektorien T_1,T_2 und T_3 für die Funktion f_a 10
1.2 Beispieltrajektorie T_4 für die Zielfunktion f_a 12
1.3 Beispieltrajektorie T_5 für die Zielfunktion f_a 13
1.4 Beispieltrajektorien T_1 und T_2 im Gradientenfeld der Testfunktion f_a 24
1.5 Richtungsfeld für die Richtungsfeld-Trajektorie mit zwei Startpunkten 33
1.6 Richtungsfeld für die Richtungsfeld-Trajektorie mit drei Startpunkten 34
1.7 Richtungsfeld-Trajektorie mit zwei vorgegebenen Punkten z_0 und z_1 im Konturplot der Zielfunktion f_a. 42
1.8 Richtungsfeld-Trajektorie mit drei vorgegebenen Punkten z_0, z_1 und z_2 im Konturplot der Zielfunktion f_a. 43
1.9 Richtungsfeld variablen Spiegelung mit drei vorgegebenen Punkten z_0, z_1 und z_2 und die Zielfunktion f_a. 45
1.10 Richtungsfeld-Trajektorie variablen Spiegelung mit drei vorgegebenen Punkten z_0, z_1 und z_2 im Konturplot der Zielfunktion f_a. 45
1.11 Richtungsfeld für ein restringiertes Optimierungsproblem mit kreisförmigem Zulässigkeitsbereich 49
1.12 Richtungsfeld-Trajektorie für das restringierte Optimierungsproblem mit kreisformigem Zulässigkeitsbereich 50
1.13 Richtungsfeld für ein Optimierungsproblem mit dreieckigem Zulässigkeitsbereich 52
1.14 Richtungsfeld-Trajektorie für das restringierte Optimierungsproblem mit dreieckigem Zulässigkeitsbereich 54

2.1 Berührungspunkte (t^1, t^3, t^4, t^5) einer Trajektorie bzgl. der Richtung q. t^2 ist kein Berührungspunkt der Trajektorie $T(f)$. 71
2.2 Suche nach einem Berührungspunkt 71
2.3 Trajektoriennetz zur Strategie der Berührungspunkte 73
2.4 Trajektoriennetz zur Strategie des äquidistanten Netzes 76

2.5 Trajektoriennetz zur Strategie des inäquidistanten Netzes . . . 78
2.6 Digraph des rekursiv konstruierten Trajektoriennetzes 80
2.7 Digraph G_{BP} für das nach Beispiel der Strategie der Berührungspunkte erzeugte Trajektoriennetz 81
2.8 Erzeugender Baum $B_{BP}(K^0)$ der von der Komponente K^0 mittels der Verbindungstrajektorien erreichbaren Komponenten 82
2.9 Die Komponente K^1 wird durch mehrere Verbindungstrajektorien geschnitten, besitzt aber keine Berührpunkte. 83
2.10 Baum $B_{BP}(K^2)$ der von der Komponente K^2 mittels der Verbindungstrajektorien erreichbaren Komponenten. 83
2.11 Baum $B_{BP}(K^3)$ der von der Komponente K_3 mittels der Verbindungstrajektorien erreichbaren Komponenten. 83
2.12 Ungerichteter Graph für das nach Beispiel der Strategie des äquidistanten Netzes erzeugtes Trajektoriennetz 84
2.13 Klassendiagramm mit der UML-Notation 87

3.1 Prädiktor-Korrektor-Schritt 96
3.2 Winkeltest bei der Rekonstruktion einer Trajektorie 100
3.3 Sprünge bei der Rekonstruktion einer Trajektorie 101
3.4 Regulärer Simplex S mit dem Mittelpunkt $x_0 = (0,0)^T$ 124
3.5 Bestimmung der Interpolationspunkte für die Surrogate-Funktion 127

4.1 3-Punkte-Beispieltrajektorie für die Testfunktion f_c; Startpunkte: $\{\{-0.4, 0.9\}, \{-0.7, -1.3\}, \{0.8, -1.3\}\}$ 131
4.2 2-Punkte-Beispieltrajektorie für die Testfunktion f_e; Startpunkte: $\{\{0.6, 1.\}, \{0.5, -1.\}\}$ 132
4.3 Beispieltrajektorie für die Testfunktion f_t; Startpunkt: $\{0.49, 0.25\}$ 134
4.4 1-Punkt-Beispieltrajektorie für die Testfunktion f_g; Startpunkt: $\{0.6, 1.0\}$ 134
4.5 Stammfunktion des Tschebyscheff-Polynomes T_6 139
4.6 Laguerre-Polynom L_4 140
4.7 Restriktionsmenge für das zweidimensionale Testproblem 1 . . 147
4.8 Beispieltrajektorie für die Testfunktion f_1; Startpunkt: $\{3.0, 1.5\}$ 147
4.9 Restriktionsmenge für das zweidimensionale Testproblem 2 . . 149
4.10 Beispieltrajektorie für die Testfunktion f_2; Startpunkt: $\{3.0, 1.5\}$ 149
4.11 Restriktionsmenge für das zweidimensionale Testproblem 3 . . 151
4.12 Beispieltrajektorie für die Testfunktion f_3; Startpunkt: $\{1.2, 1.8\}$ 151

Tabellenverzeichnis

4.1 Vergleich der Trajektorien (Anzahl der gefundenen kritischen Punkte) 141
4.2 Vergleich der Trajektorien / Anzahl der Komponenten 141
4.3 Mehrpunkttrajektorien für die Testfunktion f_s 141
4.4 Vergleich der Gitterpunkt-Strategien 143
4.5 Vergleich der Gitterpunkt-Strategien 144
4.6 Notwendige Berechnungen der vorgestellten Verfahren 144
4.7 Gegenüberstellung von KNV und AFSV 145
4.8 Trefferquote für die restringierten Probleme 153

Einführung

Newton - der Name eines der berühmtesten Wissenschaftler aller Zeiten ist unter anderen mit der Newton-Methode zur Lösung von nichtlinearen Gleichungssystemen verbunden. Besonders interessant - wegen ihrer globalen schnellen Konvergenz - ist die gedämpfte Form der Newton-Methode.

Wird die gedämpfte Newton-Methode für eine zweimal stetig differenzierbare Funktion $f : \mathbb{R}^n \to \mathbb{R}$ auf das Gleichungssystem

$$\nabla f(x) = 0$$

angewandt, so führt dies (vgl. Abschnitt 1.3) zur Euler-Diskretisierung der Lösungskurve $\mathbf{x}(\alpha)$ folgender implizieter Differentialgleichung:

$$\frac{\partial}{\partial \alpha} \nabla f(\mathbf{x}(\alpha)) = -\nabla f(\mathbf{x}(\alpha)) \qquad | \qquad \mathbf{x}(0) = x^0.$$

Für die Lösungskurve $\mathbf{x}(\alpha)$ können unter anderem folgende Eigenschaften bewiesen werden:

1. Die Richtung des Gradienten $\nabla f(x)$ bleibt entlang der Kurve konstant,
2. Die Kurve $\mathbf{x}(\alpha)$ führt vom Startpunkt x^0 zu einem kritischen Punkt x^* der Funktion f $(\nabla f(x^*) = 0)$.

Die Eigenschaften der Newtonschen Lösungskurve wurden von Branin in [6] zum Anlaß genommen, eine erweiterte Newton-Trajektorie (auch Branin-Trajektorie genannt) zu definieren, die alle Lösungen des Gleichungssystems $\nabla f(x) = 0$ enthält:

$$T(f) := \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda \in \mathbb{R}} \ \nabla f(x) = \lambda \nabla f\left(x^0\right) \right\}.$$

Die Rekonstruktion der Trajektorie $T(f)$ wird demnach nicht nach der Entdeckung eines kritischen Punktes abgebrochen, sondern weiter fortgesetzt

(Stichwort: Kontinuitätsmethoden), bis, nach Möglichkeit, alle Kandidaten für gesuchte globale Extrema gefunden wurden.

Die geometrische Beschaffenheit der Trajektorie kann sehr kompliziert sein (vgl. Abschnitt 1.2.3). Die Trajektorie kann z.B. Verzweigungspunkte enthalten und/oder aus mehreren Komponenten bestehen. Die vollständige Rekonstruktion der Trajektorie ist deshalb oft mit Schwierigkeiten verbunden.

Von Diener und Schaback wurde in [15] eine Methode vorgeschlagen, verschiedene Trajektorienkomponenten mit Hilfe von rekursiv konstruierten Hilfstrajektorien zu verbinden. In der Diplomarbeit [3] wurde diese Methode implementiert und ausführlich getestet. Es wurde dabei festgestellt, daß, trotz guter theoretischer Resultate [18], die untersuchte Methode in nicht wenigen Fällen versagt (vgl. Beispiel 96). In der vorliegenden Arbeit wurden deshalb folgende Verbesserungs- und Ergänzungsmöglichkeiten vorgestellt und getestet:

1. Trajektorien mit mehreren Startpunkten (vgl. Abschnitt 1.4.3),
2. Verallgemeinerte rekursive Konstruktion der Hilfstrajektorien (vgl. Abschnitt 2.1.3),
3. Neue Strategien zur Bestimmung der Gitterpunkte (Startpunkte für die Hilfstrajektorien, vgl. Abschnitt 2.2).

Die beschriebene Trajektorie mit mehreren Startpunkten kann als Verallgemeinerung der Newton-Trajektorie aufgefaßt und als Richtungsfeld-Trajektorie dargestellt werden. Die Richtung des Gradienten entlang der Trajektorie ist in diesem Fall also nicht mehr konstant, sondern stimmt mit einem vorgegebenen Richtungsfeld überein. Die Richtungsfeld-Trajektorien enthalten natürlich alle kritischen Punkte der Zielfunktion f und können ohne Einschränkung zur Lösung unrestringierter Optimierungsprobleme eingesetzt werden.

Dies gilt im allgemeinen nicht für Optimierungsprobleme mit Restriktionen. Die klassische Newton-Trajektorie und auch die eingeführte Trajektorie mit mehreren Startpunkten enthalten in der Regel nicht die Randpunkte der Restriktionsmenge, die die notwendige Optimalitätsbedingung 1. Ordnung erfüllen und als Lösungen des gestellten Problems in Frage kommen.
Für restringierte Optimierungsprobleme wurde deshalb eine Richtungsfeld-Trajektorie definiert, die außer den kritischen Punkten der Zielfunktion auch die interessanten Randpunkte enthält (vgl. Abschnitt 1.4.4).

Für die Rekonstruktion der Trajektorien wurde außer dem klassischen kontinuierlichen Newton-Verfahren (KNV) und dem kontinuierlichen Quasi-Newton-Verfahren (KQNV) auch ein neues ableitungsfreies Surrogate-Verfahren

(vgl. Abschnitt 3.4) beschrieben, implementiert und getestet. Der Einsatz des Surrogate-Verfahrens ermöglicht die Lösung von Optimierungsproblemen mit einer komplizierten Zielfunktion, bei deren die Berechnung bzw. Auswertung des Gradienten mit viel Aufwand verbunden oder sogar unmöglich ist. Anhand von einigen Funktionswerten in geeignet gewählten Interpolationspunkten wird anstelle der Zielfunktion eine quadratische Modellfunktion berechnet und eine lokale Näherung der gesuchten Trajektorie konstruiert. Die Modellfunktion wird dann entlang der Trajektorie von Schritt zu Schritt neu angepaßt.

Die Arbeit wurde durch viele Beispiele ergänzt, die einerseits Schwierigkeiten der Trajektorien-Verfahren aufzeigen und die Eigenschaften der eingeführten Trajektorien veranschaulichen sollten. Anderseits sollte dadurch dem Leser erleichtert werden, die vorgestellten Verfahren zu implementieren und zu testen. Die rekursive Konstruktion, als schwierigster und aufwendigster Teil des Verfahrens, wurde deshalb detailliert beschrieben und objektorientiert modelliert.

Ich möchte mich ganz herzlich bei Prof. Dr. Klaus Glashoff für die Betreuung dieser Arbeit bedanken.

Außerdem möchte ich meiner Frau Jola von ganzen Herzen für das Verständnis und die Liebe danken, mit der sie jede Phase dieser Arbeit mitgetragen hat.

Kapitel 1

Trajektorien in der globalen Optimierung

1.1 Problemstellung

Gegenstand dieser Arbeit ist das folgende Problem:

Problem 1 *(Unrestringiertes Optimierungsproblem)*
Gegeben sei eine stetig differenzierbare Zielfunktion $f \in C^1(\mathbb{R}^n)$

$$f : \mathbb{R}^n \to \mathbb{R}.$$

Gesucht ist ein Punkt $x^* \in \mathbb{R}^n$*, so daß für alle* $x \in \mathbb{R}^n$ *gilt:*

$$f(x^*) \leq f(x).$$

Viele Probleme aus der Technik, Physik, Ökonomie, Medizin und anderen Wissenschaften lassen sich als solche Probleme formulieren. Oft treten zusätzliche Beschränkungen an die Variablen und/oder Relationen zwischen solchen auf, so daß die Zielfunktion nur in einem bestimmten Bereich definiert bzw. betrachtet wird.

Problem 2 *(Restringiertes Optimierungsproblem)*
Gegeben seien:
- eine stetig differenzierbare Funktion $c \in C^1(\mathbb{R}^n)$

$$c : \mathbb{R}^n \to \mathbb{R},$$

- die Menge der zulässigen Punkte $M \subset \mathbb{R}^n$

$$M = \{x \in \mathbb{R}^n | \quad c(x) \geq 0\}$$

und eine stetig differenzierbare Zielfunktion $f \in C^1(M)$.
Gesucht ist ein zulässiger Punkt $x^* \in M$, *so daß für alle* $x \in \mathbb{R}^n$ *gilt:*

$$f(x^*) \leq f(x).$$

Es werden hier Probleme mit einer Ungleichheitrestriktion betrachtet. Die vorgestellten Methoden können leicht auf Probleme mit mehreren Restriktionen übertragen werden (s. Abschnitt 1.4.4).

Die Behandlung dieser klassischen Optimierungsprobleme ist theoretisch gut ausgearbeitet. Es gibt für Praktiker eine Reihe von Verfahren, die leicht implementierbar und auf entsprechend formulierte Aufgaben anwendbar sind. Für die unrestringierten Aufgaben (Problem 1) zählen hierzu vor allem das Newton-Verfahren, die Quasi-Newton-Verfahren und die Verfahren der konjugierten Richtungen. Für die restringierten Verfahren (Problem 2) kommt z.B. die SQP-Methode (Sequentiel Quadratic Programming) zum Ansatz.
Zur Lösung des Optimierungsproblems mit Hilfe der oben genannten Methoden verwendet werden die notwendigen und hinreichenden Bedingungen für die lokale Optimalität der gesuchten Lösung. Für unsere Betrachtungsweise ist lediglich die notwendige Bedingung 1. Ordnung von Bedeutung. Für weitere Bedingungen wird auf klassische Handbücher und Skripte zur nichtlinearen Optimierung hingewiesen [20] [23][26].

Zuerst wird das unrestringierte Optimierungsproblem 1 untersucht.

Satz 3 *(notwendige Bedingung 1. Ordnung).*
Ist x^* *lokales Minimum von* $f \in C^1(\mathbb{R}^n)$, *so gilt:*

$$\nabla f(x^*) = 0. \tag{1.1}$$

Definition 4 *Ist die notwendige Bedingung 1. Ordnung für eine Funktion* f *und einen Punkt* x^* *erfüllt, so ist* x^* *ein* ***kritischer*** *oder* ***stationärer Punkt*** *von* f.

Die Menge der kritischen Punkte der Funktion f wird im weiteren mit $Krit(f)$ bezeichnet.

Die notwendige Bedingung 1. Ordnung für unrestringierte Probleme schließt die Existenz einer Abstiegsrichtung s (d.h. $\nabla f(x^*)^T s < 0$) in Punkt x^* aus. Bei einem restringierten Problem (2) wäre die Forderung (1.1) für die Randpunkte $x_c \in \partial M$ des zulässigen Bereiches M zu stark. In einem solchen Fall muß lediglich sichergestellt werden, daß die Menge der bzgl. der zulässigen Menge M zulässigen Richtungen keine Abstiegsrichtung enthält.

Definition 5 $x^* \in M$ *heißt* ***regulärer Punkt*** *von* M, *wenn* $\nabla c(x^*) \neq 0$.

Definition 6 *$s \in \mathbb{R}^n$ heißt **zulässige Abstiegsrichtung** für f in einem regulären Punkt x^* bzgl. der zulässigen Menge M, wenn*

$$\nabla f(x^*)^T s < 0 \tag{1.2}$$
$$\nabla c(x^*)^T s > 0. \tag{1.3}$$

Für restringierte Optimierungsprobleme kann folgende notwendige Bedingung 1. Ordnung aufgestellt werden, die die Existenz einer zulässigen Abstiegsrichtung s in einem vermeintlichen Minimum x^* ausschließt:

Satz 7 *(notw. Bedingung 1. Ordnung für restr. Optimierungsprobleme). Ist $x^* \in \partial M$ lokales Minimum von f auf M und ist x^* ein regulärer Punkt von M, so gibt es eine Zahl λ, so daß gilt*

$$\nabla f(x^*) - \lambda \nabla c(x^*) = 0. \tag{1.4}$$

Der Lagrange-Ansatz bietet einen anderen Zugang zu möglichen optimalen Punkten, die auf dem Rand ∂M der zulässigen Menge M liegen. Die Lagrange-Funktion wird hierbei als Summe der Zielfunktion $f(x)$ und der mit dem variablen Lagrange-Faktor λ multiplizierten Restriktion $c(x)$ gebildet:

$$L(x, \lambda) := f(x) - \lambda c(x). \tag{1.5}$$

Ein Randpunkt $x^* \in \partial M$, der die notwendige Bedingung 1. Ordnung für die restringierte Optimierungsprobleme erfüllt, kann dann im Paar mit dem entsprechenden Lagrange-Parameter $\lambda^* \in \mathbb{R}$ als kritischer Punkt der Lagrange-Funktion aufgefaßt werden.

Satz 8 *(Lagrange Charakterisierung)*
Sei x^* ein regulärer Punkt von M.
Ein Paar (x^*, λ^*) ist genau dann ein kritischer Punkt der Lagrange-Funktion $L(x, \lambda)$, wenn x^* ein Randpunkt der zulässigen Menge M ist, der die notwendige Bedingung 1. Ordnung für restringierte Optimierungsprobleme erfüllt.

Beweis. Der Gradient der Lagrange-Funktion läßt sich wie folgt berechnen:

$$\nabla L(x^*, \lambda^*) = \begin{pmatrix} \nabla f(x^*) - \lambda^* \nabla c(x^*) \\ c(x^*) \end{pmatrix}.$$

Die Bedingung $\nabla L(x^*, \lambda^*) = 0$ ist dann äquivalent mit

$$\nabla f(x^*) - \lambda \nabla c(x^*) = 0 \wedge x^* \in \partial M.$$

■

1.2 Trajektorien

1.2.1 Motivation und theoretischer Ansatz

Die globale Optimalität eines gefundenen kritischen Punktes x^* ist im allgemeinen nur bei den zusätzlichen (sehr starken) Voraussetzungen für das Problem wie z.B. Konvexität der Zielfunktion sichergestellt.
Gibt es mehrere, möglicherweise viele kritische Punkte, so müssen Wege gesucht werden, um nicht nur eine lokale, sondern die global beste Lösung des Optimierungsproblems zu finden.
Möglich ist es, die Suche immer wieder von verschiedenen, nach einem Zufallsprinzip ausgesuchten Startpunkten zu wiederholen. (s. Monte-Carlo-Methoden) oder während der Suche ab und zu einen Zufallsschritt auszuführen, um eventuell aus einer vermeintlichen Gasse herauszukommen (s. genetische Verfahren, Stichwort: Mutationen).
Bei beiden Methoden muß letztendlich der Zufall entscheiden, ob die richtige globale Lösung gefunden wird. Eine richtige Parameterwahl für das ausgewählte Verfahren sichert allerdings große statistische Zuverlässigkeit.
In dieser Arbeit wurde ein anderer Ansatz gewählt. Um möglichst alle vermeintlichen Kandidaten für das globale Optimum zu finden, wird eine Verbindung (Pfad, Trajektorie) zwischen den einzelnen kritischen Punkten konstruiert. Die Trajektorie wird hier allgemein mit einer Hilfsfunktion H charakterisiert, die folgende Voraussetzungen erfüllen soll:

Definition 9 *Eine Funktion $H : \mathbb{R}^n \to \mathbb{R}^{n-1}$ heißt eine **bzgl. der Funktion f trajektorieninduzierende Hilfsfunktion** falls*

(i) die Funktion H stetig differenzierbar ist und

(ii) alle kritischen Punkte der Zielfunktion f von der Funktion H in den Nullvektor abgebildet werden:

$$\forall_{x \in Krit(f)} H(x) = 0 \tag{1.6}$$

Die induzierte Trajektorie $T_H(f)$ wird als die Nullmenge der Funktion H definiert:

Definition 10 *Für eine stetig differenzierbare Zielfunktion $f \in C^1(\mathbf{M})$ und eine Hilfsfunktion H, die die Bedingungen (i) und (ii) erfüllt, wird die Trajektorie $T_H(f)$ definiert als:*

$$T_H(f) := \{x \in \mathbb{R}^n | \quad H(x) = 0\}.$$

Bemerkung 11 *Aus der Bedingung (1.6) folgt direkt, daß die eingeführte Trajektorie $T_H(f)$ alle kritischen Punkte der Zielfunktion enthält.*

Bei einer restringierten Aufgabe kommen außer den kritischen Punkten auch lokal optimalen Randpunkte als mögliche Optimallösung in Frage. Um die optimalen Randpunkte zu finden, könnte eine entsprechende Trajektorie für die Lagrange-Funktion konstruiert werden. Ein anderer Weg mit einer geeignet (s. Bedingung 1.6) definierten Hilfsfunktion $H(x)$, die außer den kritischen Punkten auch lokal optimalen Randpunkte in den Nullvektor abbildet, wird im Abschnitt 1.4.4 vorgeschlagen und untersucht. In den übrigen Teilen der Arbeit wird der unrestringierte Fall betrachtet.

1.2.2 Eigenschaften und Charakterisierung

Die lokalen Eigenschaften der Trajektorie $T_H(f)$ in einem Punkt $x \in T_H(f)$ sind vom Rang der Jacobi-Matrix $DH(x)$ der Funktion H in diesem Punkt abhängig. Aus diesem Grund werden die Begriffe eines regulären bzw. singulären Punktes und Wertes für die Funktion H eingeführt.

Definition 12 *Sei $H : \mathbb{R}^n \to \mathbb{R}^{n-1}$ eine stetig differenzierbare Funktion. Ein Punkt $x \in M$ heißt ein **regulärer Punkt** von H, wenn die Jacobi-Matrix $DH(x)$ existiert und den maximalen Rang hat:*

$$Rang\,(DH(x)) = n - 1.$$

*Ein Punkt $y \in \mathbf{R}^{n-1}$ heißt ein **regulärer Wert** von H, wenn alle Urbilder von y reguläre Punkte von H sind:*

$$\forall_{z \in H^{-1}(y)} Rang\,(DH(z)) = n - 1.$$

*Punkte und Werte heißen **singulär**, wenn sie nicht regulär sind.*

Unter bestimmten Voraussetzungen kann dann bewiesen werden, daß die Trajektorie $T_H(f)$ lokal einer eindimensionalen Lösungskurve einer bestimmten Differentialgleichung entspricht. Aus dem Satz über implizite Funktionen folgt:

Satz 13 *Ist $x^0 \in T_H(f)$ ein regulärer Punkt von H, so existiert ein offenes Intervall $J = (-\delta, \delta)$, $\delta > 0$ und eine differenzierbare parametrisierte Kurve $x(\alpha) : J \to \mathbb{R}^n$, so daß für alle $\alpha \in J$ gilt:*

$$x(0) = x^0 \tag{1.7}$$

$$H(x(\alpha)) = 0 \tag{1.8}$$

$$Rang\,(DH(x(\alpha))) = n - 1 \tag{1.9}$$

$$\dot{x}(\alpha) \neq 0. \tag{1.10}$$

Für den Tangentenvektor $\dot{x}(\alpha)$ der Kurve $x(\alpha)$ gilt außerdem:

$$DH(x(\alpha))\,\dot{x}(\alpha) = 0. \tag{1.11}$$

Beweis. Da die Jacobi-Matrix $DH(x^0)$ im regulären Punkt x^0 den vollen Rang hat, gibt es einen Index $i \in \{1, ..., n\}$ so daß die Untermatrix D_i der Matrix $DH(x^0)$, die durch Weglassen der i-ten Spalte entsteht, nichtsingulär ist:

$$D_i := \left(\frac{\partial H(x^0)}{\partial x_1}, ..., \frac{\partial H(x^0)}{\partial x_{i-1}}, \frac{\partial H(x^0)}{\partial x_{i+1}}, ..., \frac{\partial H(x^0)}{\partial x_n} \right).$$

Wie folgt wird eine Erweiterung $\bar{H}(x, \alpha)$ der Funktion $H(x)$ konstruiert:

$$\bar{H}(x, \alpha) := \begin{pmatrix} H(x) \\ x_i - x_i^0 - \alpha \end{pmatrix}.$$

Die Jacobi-Matrix $D\bar{H}(x^0)$ der erweiterten Funktion $\bar{H}(x, \alpha)$ ist dann gegeben durch:

$$D\bar{H}(x^0) = \left(\frac{\partial \bar{H}(x^0)}{\partial x}, \frac{\partial \bar{H}(x^0)}{\partial \alpha} \right) = \begin{pmatrix} DH(x^0) & 0 \\ e_i^T & -1 \end{pmatrix}.$$

Aus der Nichtsingularität der Matrix D_i folgt dann die Nichtsingularität der Matrix $\frac{\partial \bar{H}(x^0)}{\partial x}$. Das Funktionensystem $\bar{H}(x, \alpha) = 0$ kann also nach x aufgelöst werden. In dieser Weise entsteht eine parametrisierte Kurve $x(\alpha)$, die alle Bedingungen (1.7) bis (1.10) erfüllt.
Die Gleichung (1.11) folgt direkt durch die Differenzierung der Gleichung (1.8). ■

Die Differentialgleichung (1.11) wird hier als verallgemeinerte Newton-Gleichung und die Lösungskurve $x(\alpha)$ als **verallgemeinerte Newton-Kurve** bezeichnet. Für weitere Ausführungen ist es hierbei bequem, die Lösungskurve nach der Bogenlänge s zu parametriesieren [24]:

$$ds = \left[\sum_{j=1}^{n} \left(\frac{dx_j(\alpha)}{d\alpha} \right)^2 \right]^{\frac{1}{2}} d\alpha,$$

wobei mit x_j die j-te Komponente von x bezeichnet.

Bemerkung 14 *Nach der Gleichung (1.11) liegt der Tangentenvektor $\dot{x}(s)$ entlang der reparametrisierten Kurve $x(s)$ immer im Kern der Jacobi-Matrix $DH(x(s))$,*

$$\dot{x}(s) \in \ker\left(DH(x(s))\right).$$

Aufgrund der Bogenlängeparametrisierung gilt:

$$\|\dot{x}(s)\| = 1.$$

Ohne Beschränkung der Allgemeinheit kann weiterhin eine positive Orientierung angenommen werden

$$\det \begin{pmatrix} DH(x(s)) \\ \dot{x}(s)^T \end{pmatrix} > 0.$$

Diese Eigenschaft berechtigt folgende Definition, die speziell bei der numerischen Rekonstruktion der theoretisch definierten Trajektorien ihre Anwendung findet.

Lemma 15 *Für jede $n-1 \times n$ Matrix A vollen Ranges existiert ein eindeutiger Vektor $t(A)$ mit folgenden Eigenschaften:*

$$\begin{aligned} At(A) &= 0 \\ \|t(A)\| &= 1 \\ \det \begin{pmatrix} A \\ t(A)^T \end{pmatrix} &> 0. \end{aligned}$$

Der Vektor $t(A)$ heißt ***der von der Matrix A induzierte Tangentenvektor.***

Korollar 16 *Die Kurve $x(s)$ kann als eine lokale Lösung der folgenden Anfangswertaufgabe definiert werden (vgl. [1]):*

$$\dot{x}(s) = t(DH(x(s))) \quad | \quad x(0) = x^0. \tag{1.12}$$

Die rechte Seite der aufgestellten Differentialgleichung (1.12) ist genau dann richtig definiert, wenn die Jacobi-Matrix $DH(x)$ den vollen Rang hat:

$$Rang(DH(x)) = n - 1. \tag{1.13}$$

Die Bedingung (1.13) gilt nur für reguläre Punkte der Funktion H und ist nicht direkt von der Eigenschaften der Funktion f oder deren Gradienten ∇f abhängig. In der Arbeit [1] wurden folgende geometrische Eigenschaften der Lösungskurve $x(s)$ bewiesen.

Satz 17 *Wenn 0 ein regulärer Wert von H ist, dann ist die Kurve $x(\alpha)$ definiert auf dem ganzen $\mathbb{R}$ und erfüllt genau eine von folgenden zwei Bedingungen:*

(i) die Kurve $x(s)$ ist diffeomorph zu einem Kreis

(ii) die Kurve $x(s)$ ist diffeomorph zu einer Gerade.

Die Trajektorie $T_H(f)$ kann als Erweiterung der Lösungskurve $x(s)$ für die nicht regulären Punkte der Hilfsfunktion H aufgefaßt werden und wird als **verallgemeinerte Newton-Trajektorie** bezeichnet.

Bemerkung 18 *Falls 0 ein regulärer Wert von H ist, stellt die Kurve $x(s)$ eine Komponente der Trajektorie $T_H(f)$.*

Korollar 19 *Die Trajektorie $T_H(f)$ ist auch für die Punkte der Nullmenge von H definiert, wo die Jacobi-Matrix $DH(x)$ singulär ist und die Kurve $x(\alpha)$ nicht mehr definiert ist. In einem solchen Punkt ist die Trajektorie $T_H(f)$ nicht mehr lokal eindimensional.*

1.2.3 Numerische Ansätze

Auf der Suche nach den kritischen Punkten der Zielfunktion f wird also eine verallgemeinerte Newton-Trajektorie $T_H(f)$ konstruiert. Um ausgehend von einem Anfangspunkt x^0 der Trajektorie $T_H(f)$ folgend zu kritischen Punkten zu gelangen, muß die Trajektorie numerisch rekonstruiert werden.

Eine einfache Realisierung dieser Aufgabe wäre, die Differentialgleichung (1.12) mit einem erprobtem aus einer Programmbibliothek stammenden Verfahren numerisch zu integrieren. Da allerdings außer der Jacobi-Matrix DH die Hilfsfunktion H zur Verfügung steht, ist der im Kapitel 3 beschriebene Prädiktor-Korrektor Ansatz zur Lösung des nichtlinearen Gleichungssystem $H(x) = 0$ besser für die Rekonstruktion der Trajektorie $T_H(f)$ geeignet [25]. Der Zugang zur Trajektorie $T_H(f)$ über das nichtlineare Gleichungssystem $H(x) = 0$ ist daher, aus numerischer Sicht, dem Zugang über die Differentialgleichung (1.12) vorzuziehen (vgl. Abschnitte 1.3 und 1.4).

Im folgenden Beispiel werden weitere mögliche Schwierigkeiten, die bei der numerischen Rekonstruktion einer verallgemeinerten Newton-Trajektorie auftreten können, angedeutet. Für eine einfache Testfunktion werden hierzu einige typische Strukturen klassischer Newton-Trajektorien aufgezeigt.

Beispiel 20 *Sei die folgende Zielfunktion $f_a : \mathbb{R}^2 \to \mathbb{R}$ mit der Variablen $z = (x, y)^T \in \mathbb{R}^2$ gegeben:*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y.$$

Der Gradient der Funktion f_a und damit auch die kritischen Punkte können leicht berechnet werden:

$$\nabla f_a(z) = \left(x^2 - 1, 1 - y^2\right).$$

Die Funktion besitzt also vier kritische Punkte:

$$z_i^* = (\pm 1, \pm 1)^T, \quad i = 1, ..., 4.$$

Mit Hilfe der Hesse-Matrix $D^2 f_a(x)$,

$$D^2 f_a(x) = \begin{pmatrix} 2x & 0 \\ 0 & -2y \end{pmatrix},$$

kann der Charakter der kritischen Punkte z_i^* festgestellt werden. Durch Untersuchung des Vorzeichens der Eigenwerte der Matrix $D^2 f_a(x)$ erweisen sich der Punkt $z_2^* = (1, -1)^T$ als lokales Minimum, der Punkt $z_4^* = (-1, 1)^T$ als lokales Maximum und die Punkte $z_1^* = (1, 1)^T$ und $z_i^* = (-1, -1)^T$ als zwei Sattelpunkte.
Abhängig von dem Ausgangspunkt $z_0 = (x_0, y_0)^T$ kann die Hilfsfunktion $H(x)$ wie folgt konstruiert werden:

$$H(z) = \frac{\partial f_a(z_0)}{\partial y}\left(x^2 - 1\right) - \frac{\partial f_a(z_0)}{\partial x}\left(1 - y^2\right).$$

Die Bedingungen aus der Definition 9 sind hier trivialerweise erfüllt. Für den Ausgangspunkt z_0 gilt dann außerdem

$$H(z_0) = 0.$$

Auf diese Weise wird sichergestellt, daß außer den kritischen Punkten auch der Ausgangspunkt z_0 der Trajektorie gehört. Die klassische Newton-Trajektorie ist dann (vgl. Abschnitt 1.3 und insbesonders Beispiel 29) definiert durch:

$$T(f_a, z_0) = T_H(f_a).$$

Es können beispielweise, abhängig vom Punkt z_0, folgende Newton-Trajektorien

konstruiert werden:

$$
\begin{aligned}
& z_0^1 = (1.2, 0.6) \\
T_1 := T\left(f_a, z_0^1\right) &= \left\{ z \in \mathbb{R}^2 \middle| \quad H_1(z) := 0.64x^2 + 0.44y^2 - 1.08 = 0 \right\} \\
& z_0^2 = (0.4, 0.6) \\
T_2 := T\left(f_a, z_0^2\right) &= \left\{ z \in \mathbb{R}^2 \middle| \quad H_2(z) := 0.64x^2 - 0.84y^2 + 0.2 = 0 \right\} \\
& z_0^3 = (1.0, 0.0) \\
T_3 := T\left(f_a, z_0^3\right) &= \left\{ z \in \mathbb{R}^2 \middle| \quad H_3(z) := x^2 - 1 = 0 \right\} \\
& z_0^4 = (0.2, 0.2) \\
T_4 := T\left(f_a, z_0^4\right) &= \left\{ z \in \mathbb{R}^2 \middle| \quad H_4(z) := 0.96x^2 - 0.96y^2 = 0 \right\} \\
& z_0^5 = (0.21, 0.2) \\
T_5 := T\left(f_a, z_0^5\right) &= \left\{ z \in \mathbb{R}^2 \middle| \quad H_5(z) := 0.96x^2 - 0.9559y^2 - 0.0041 = 0 \right\}.
\end{aligned}
$$

Die Newton-Trajektorie $T(f_a, z_0)$ kann also eine Ellipse (T_1), eine Hyperbel mit waagerecht (T_2) oder senkrecht (T_5) verlaufenden Flügel, zwei parallele (T_3) oder orthogonalen (T_4) Geraden werden.
Die Trajektorien T_1,T_2 und T_3 werden in der Abbildung (1.1) dargestellt. Die Trajektorien T_4 und T_5 werden in den Bildern (1.2) und (1.3) getrennt gezeigt.

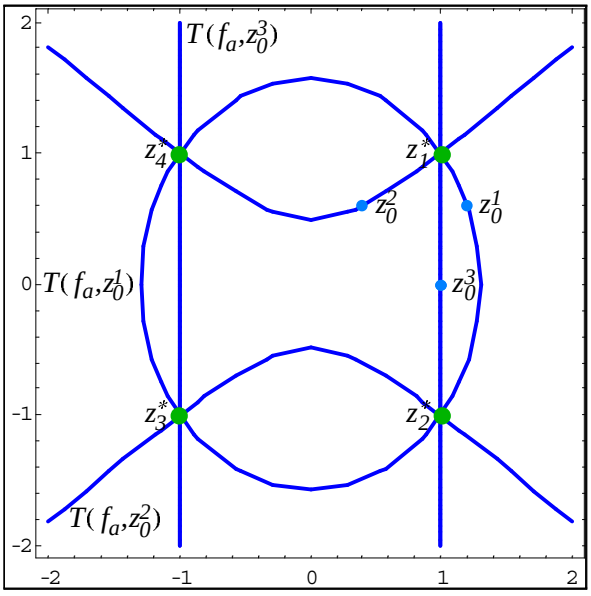

Abbildung 1.1: Beispieltrajektorien T_1,T_2 und T_3 für die Funktion f_a

Die vorgestellten Trajektorien beinhalten natürlich auch alle kritischen Punkte der Funktion f_a. Die geometrische Beschaffenheit der Trajektorien ist aber durchaus unterschiedlich.

Trajektorienkomponenten

Während die Trajektorien T_1 und T_4 zusammenhängend sind, bestehen die Trajektorien T_2,T_3 und T_5 aus zwei getrennten Komponenten. Es ist bei einem praktischen Optimierungsproblem oft schwer vorherzusehen, wieviele Komponenten die definierte Trajektorie bilden. Die Suche nach den Trajektorienkomponente ist für ein numerisches Rekonstruktionsverfahren ein schwieriges Problem.
Im Kapitel 2 werden Methoden vorgestellt, die es ermöglichen, mittels rekursiv konstruierten Verbindungstrajektorien die Komponenten der Haupttrajektorie zu finden. Im Abschnitt 1.4.3 wird außerdem eine neue Trajektorienklasse eingeführt, die mehrere vorgegebenen Startpunkte enthält. Auf diese Weise ist es dann in der Regel möglich, einige (und nicht nur eine!) Trajektorienkomponenten bereits beim ersten Anlauf zu rekonstruieren.

Zyklische und nicht zyklische Komponenten

Die einzelnen Trajektorienkomponenten sind unter bestimmten Voraussetzungen entweder zu einem Kreis (Trajektorie T_1) oder einer Gerade (Trajektorien T_2,T_3 und T_5) diffeomorph (s. Satz 17).
Die numerische Behandlung der zyklischen Komponenten erfordert einen *Zyklustest*, der mit Hilfe charakteristischer Punkte erkennt, ob die Komponente vollständig rekonstruiert wurde. Die nichtzyklische Komponenten können anderseits, wegen ihrer unendlichen Länge, nicht vollständig rekonstruiert werden. Aus diesem Grunde wird die Suche auf einen Bereich B beschränkt, in dem alle kritische Punkte vermutet werden. Besitzt die Zielfunktion nur endlich viele kritische Punkte, so existiert ein konvexer Bereich $B \subset \mathbb{R}^n$, für den gilt:

$$Krit\,(f) \subset B.$$

Singuläre Punkte und Sprünge zwischen den Komponenten

Ein anderes Problem stellen Verzweigungspunkte dar. Wie an dem Beispiel der Trajektorie T_4 zu erkennen ist (vgl. Abbildung 1.2), können solche Punkte dann auftreten, wenn ein Trajektorienpunkt ein singuläres Punkt der Hilfsfunktion $H\,(z)$ ist. Für die Trajektorie T_4 und den Trajektorienpunkt

$\check{z} = (0,0)^T$ gilt hier

$$DH_4(\check{z}) = (1.92\check{x}, 1.92\check{y}) = (0,0),$$

so daß der Rang der Matrix $DH_4(\check{z})$ gleich $Null$ ist. Für die anderen Trajektorien kann man zeigen, daß der Punkt $\check{z} = (0,0)^T$ auch ein singulärer Punkt der Hilfsfunktion H ist. Da $\check{z}$ aber dann nicht zur Trajektorie gehört, stellt dies in diesen Fälle keine Schwierigkeit dar.

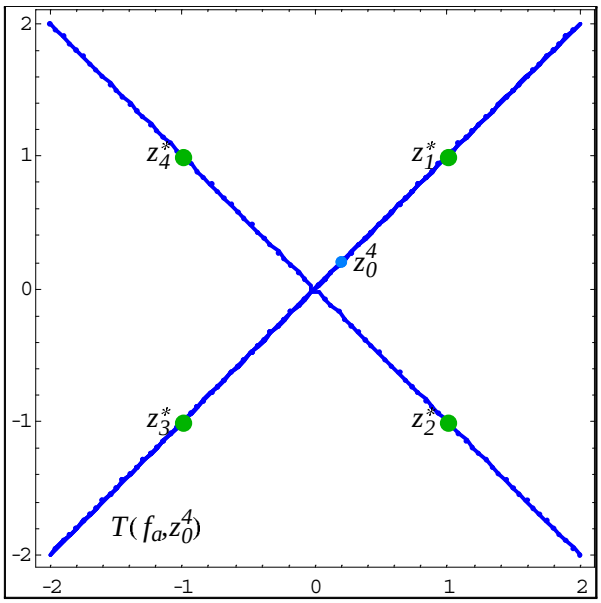

Abbildung 1.2: Beispieltrajektorie T_4 für die Zielfunktion f_a

Um Verzweigungspunkte bei einer numerisch berechneten Trajektorie zu vermeiden, reicht es, im allgemeinen den Ausgangspunkt ein wenig zu stören (s. Abbildung 1.3). Die auf diese Weise neu definierte Trajektorie hat keine Verzweigungspunkte mehr.
Demnach gibt es problematische Stellen, an denen verschiedene Trajektorienkomponenten nahe beieinander verlaufen. Es besteht dann die Gefahr, daß das Rekonstruktionsverfahren fälschlicherweise von einer Komponente auf die andere springt. Dank einer geeigneten Schrittlängesteuerung (vgl. Abschnitt 3.2.2) kann dies vermieden werden. Das Verfahren ist allerdings deutlich langsamer.

Ist es für eine definierte Trajektorie möglich, sie abzufahren, so können alle kritischen Punkte für das Problem gefunden werden. Durch den einfachen Vergleich der Funktionswerte kann dann der global optimale Punkt bestimmt werden.

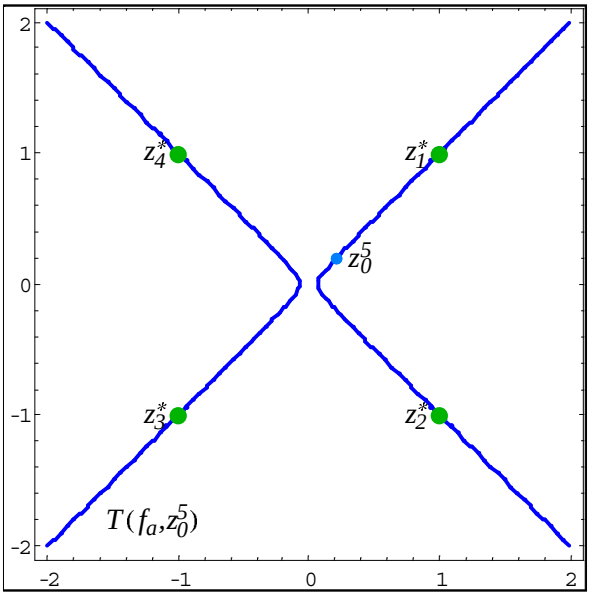

Abbildung 1.3: Beispieltrajektorie T_5 für die Zielfunktion f_a

Die kritischen Punkte werden mit Hilfe eines Trajektorienverfahrens im allgemeinen nur sehr ungenau bestimmt. Es geht in diesem Fall vielmehr darum, sich einen Gesamtüberblick über die Eigenschaften der Zielfunktion zu verschaffen und die Struktur des Problems zu erkennen. Die gefundenen kritischen Punkte können dann als "gute" Startpunkte für die klassischen lokal konvergenten Optimierungsverfahren genommen werden.

1.3 Klassische Newton-Trajektorie

1.3.1 Zugang über eine Differentialgleichung

Der Name der im folgenden konstruierten Trajektorie kommt von der bekannten Methode, die in Form einer im weiteren angegebenen Iterationsformel (1.14) zur Lösung des Gleichungssystems (1.1) angewandt werden kann. Sei f eine zweimal stetig differenzierbare Funktion $f : \mathbb{R}^n \to \mathbb{R}$ mit den Gradienten ∇f und der Hessematrix $D^2 f$. Die Newton-Iterationsformel ist dann gegeben durch:

$$x_{i+1} := x_i - p_i D^2 f(x_i)^{-1} \nabla f(x_i). \tag{1.14}$$

Der durch die Punkte $\{x_i\}_{i=0,1,\ldots}$ bestimmte Polygonzug kann als eine Euler-Diskretisierung mit der Schrittlänge p_i der durch folgende Anfangswertauf-

gabe definierten Kurve $\mathbf{x}(\alpha)$ angesehen werden:

$$\dot{x}(\alpha) = -D^2 f(x(\alpha))^{-1} \nabla f(x(\alpha)) \qquad | \qquad x(0) = x^0. \tag{1.15}$$

Es ist in diesem Kontext üblich, an Stelle der Differentialgleichung 1.15 folgende Aufgabe zu betrachten (vgl. [16]):

$$\dot{x}(\alpha) = -adjD^2 f(x(\alpha)) \nabla f(x(\alpha)) \qquad | \qquad x(0) = x^0. \tag{1.16}$$

Mit $adjD^2 f$ wird die Adjungierte der Hesse-Matrix $D^2 f$ bezeichnet, die auch für diejenige Punkte existiert, für die die Hesse-Matrix singulär und damit nicht invertierbar ist. Ist anderseits die Hesse-Matrix $D^2 f$ regulär, so kann die Inverse $(D^2 f)^{-1}$ wie folgt dargestellt werden:

$$\left(D^2 f\right)^{-1} = \frac{1}{\det D^2 f} adjD^2 f.$$

Für die Hilfsfunktion $H(x) := G^T \nabla f(x)$, mit einer beliebigen vorgegebenen $n \times n-1$ Matrix G vollen Ranges, wurde in [3] gezeigt, daß der Tangentenvektor $t(DH(x))$ der Jacobi-Matrix $DH(x)$, wie folgt dargestellt werden kann:

$$t(DH(x)) = \sigma \frac{adjD^2 f(x(\alpha)) \nabla f(x(\alpha))}{\|adjD^2 f(x(\alpha)) \nabla f(x(\alpha))\|}, \qquad \sigma = \pm 1.$$

Wird die Kurve $\mathbf{x}(\alpha)$ nach der Bogenlänge s parametrisiert, so kann die Anfangswertaufgabe (1.16) auf die Differentialgleichung (1.12) zurückgeführt werden.

Die impliziete Form der Differentialgleichung 1.16 mit den entsprechenden Anfangswertbedingung

$$\frac{\partial}{\partial \alpha} \nabla f(x(\alpha)) = -\nabla f(x(\alpha)) \qquad | \qquad x(0) = x^0$$

führt dann zu folgenden Eigenschaften der Lösungskurve $\mathbf{x}(\alpha)$ (vgl. [18]):

Satz 21 *Für die Lösungskurve $\mathbf{x}(\alpha)$ der Newtonschen Differentialgleichung 1.15 gilt:*

(i) die Richtung des Gradienten $\nabla f(x(\alpha))$ bleibt entlang der Kurve unverändert:

$$\forall_{x(\alpha) \in \mathbf{x}(\alpha)} \nabla f(x(\alpha)) = \nabla f\left(x^0\right) e^{-\alpha},$$

(ii) mit $\alpha \to \infty$ führt die Kurve $\mathbf{x}(\alpha)$ von dem Startpunkt x^0 zu einem kritischen Punkt der Funktion f.

Der Pfad, der entlang der Lösungskurve $\mathbf{x}(\alpha)$ von dem Startpunkt x^0 zu einem kritischen Punkt x^* der Funktion f führt, wird als Newton-Kurve bezeichnet. In den nächsten Absätzen werden die genannten Eigenschaften der Newton-Kurve genutzt, um die klassische Newton-Trajektorie zu definieren, die alle kritische Punkte enthält (s. [19]).

1.3.2 Zugang über ein nichtlineares Gleichungssystem

Die Menge der kritischen Punkte der Zielfunktion $f: \mathbb{R}^n \to \mathbb{R}$ wird durch die Abschwächung des notwendigen Optimalitätskriteriums (vgl. Satz 3) erweitert. Hierzu werden die n Gleichungen des nichtlinearen Gleichungssytems (1.1) durch $n-1$ linear unabhängige Gleichungskombinationen ersetzt:

$$\nabla f(x) = 0 \to \begin{bmatrix} g_1^T \nabla f(x) = 0 \\ \dots \\ g_{n-1}^T \nabla f(x) = 0 \end{bmatrix}. \tag{1.17}$$

Das Gleichungssystem (1.17) kann mit Hilfe einer beliebigen $n \times n-1$ Matrix G vollen Ranges

$$\begin{aligned} G &= (g_1, ..., g_{n-1}) \\ Rang(G) &= n-1 \end{aligned}$$

oder der entsprechend definierten Hilfsfunktion $H: \mathbb{R}^n \to \mathbb{R}^{n-1}$ kurz geschrieben werden:

$$H(x) := G^T \nabla f(x) = 0. \tag{1.18}$$

Die Lösungsmenge $T_G(f)$ des auf diese Weise abgeschwächten Gleichungssystem (1.17) enthält natürlich alle kritischen Punkte der Zielfunktion f:

$$Krit(f) \subset T_G(f) := \{x \in \mathbb{R}^n | \qquad H(x) = 0\}. \tag{1.19}$$

Definition 22 *Für eine $n \times n-1$ Matrix G mit*

$$Rang(G) = n-1$$

ist die ***durch die Matrix G induzierte Trajektorie*** *gegeben mit:*

$$T_G(f) := \left\{x \in \mathbb{R}^n | \quad H(x) := G^T \nabla f(x) = 0\right\}. \tag{1.20}$$

Bemerkung 23 *Die Bedingung $G^T \nabla f(x) = 0$ ist zur Aussage äquivalent, daß der Gradient $\nabla f(x)$ im Kern der Matrix G liegt. Die Trajektorie $T_G(f)$ ist durch den eindimensionalen Kern der Matrix G eindeutig bestimmt. Für alle $n \times n-1$ Matrizen G vollen Ranges mit dem gleichen Kern sind auch die durch diese Matrizen induzierten Trajektorien gleich!*

Der aufgrund der Rangbedingung eindimensionale Kern ist aber durch einem beliebigen Vektor

$$0 \neq g \in \ker(G)$$

eindeutig festgelegt. Diese Eigenschaft begründet folgende Definition:

Definition 24 *Für einen Vektor $g \in \mathbb{R}^n$ und einer Matrix G mit*

$$\begin{aligned} Rang(G) &= n-1 \\ g &\in \ker G^T, \end{aligned}$$

*ist die **durch den Vektor g induzierte Trajektorie** gegeben durch:*

$$T_g(f) := T_G(f). \tag{1.21}$$

*g wird im Weiteren ein **induzierender Vektor** genannt.*

Als induzierender Vektor $g \in \mathbb{R}^n$ kann der Gradient der Funktion f in einem nichtkritischen Punkt x_0 ($\nabla f(x_0) \neq 0$) genommen werden:

$$g := \nabla f(x_0). \tag{1.22}$$

Der Punkt x_0 kann dann als Startpunkt für das Rekonstruktionsverfahren benutzt werden (vgl. Beispiel 20). In diesem Fall ist die Trajektorie $T(f)$ durch den Vektor g oder äquivalent durch den Startpunkt x_0 eindeutig bestimmt und wird daher weiter mit $T_g(f)$ bzw. mit $T(f, x_0)$ bezeichnet.

Satz 25 *Sei $g := \nabla f(x_0) \neq 0$ und f eine zweimal stetig differenzierbare Funktion $f \in C^2(\mathbb{R}^n)$. Für die durch den Vektor g induzierte Trajektorie $T_g(f)$ können dann folgende Eigenschaften bewiesen werden:*

(i) Die Richtung des Gradienten ∇f ist für alle Punkte der Trajektorie $T_g(f)$ gleich der Richtung des induzierenden Vektors g:

$$\forall_{x \in T_g(f)} \exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda g.$$

(ii) Ist der Gradient ∇f der Funktion f in einem Punkt x linear abhängig von g, so gehört x damit zur Trajektorie $T_g(f)$

$$\exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda g \Rightarrow x \in T_g(f).$$

(iii) Die Trajektorie $T_g(f)$ ist eine Erweiterung der im vorigen Abschnitt definierten Newton-Kurve $\mathbf{x}(\alpha)$:

$$\mathbf{x}(\alpha) \subset T_g(f).$$

Beweis. Die Aussagen (i) und (ii) folgen direkt aus der Bemerkung (23). Nach dem Satz 21 bleibt die Richtung des Gradienten $\nabla f(x)$ der Zielfunktion f entlang der Kurve $\mathbf{x}(\alpha)$ unverändert. Da der Gradient $\nabla f(x_0)$ im Startpunkt x_0 als induzierender Vektor g gewählt wurde, folgt für alle Punkte x der Lösungskurve $\mathbf{x}(\alpha)$:

$$\forall_{x \in \mathbf{x}(\alpha)} \exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda \nabla f(x_0) = \lambda g$$

und mit (ii) dann auch

$$\forall_{x \in \mathbf{x}(\alpha)} x \in T_g(f).$$

■

Definition 26 *Die Trajektorie $T_g(f)$,*

$$T_g(f) := \{x \in \mathbb{R}^n | \quad \nabla f(x) \ \textit{parallel zu}\ g\},$$

*wird als **klassische Newton-Trajektorie** bezeichnet und ist mit der durch Vektor g induzierten Trajektorie identisch (vgl. Definition 24).*

Im Abschnitt 1.4 wird eine andere erweiterte Familie der Newton-Trajektorien eingeführt.

Konstruktion des Gleichungssystems

Im folgenden wird für einem vorgegebenen Vektor $g = \nabla f(x_0)$ eine $n \times n-1$ Matrix G bestimmt für die gilt:

$$\ker G = span\{g\}.$$

Diese Bedingung sichert die Zugehörigkeit des Ausgangspunktes x_0 zur Trajektorie $T_G(f)$.

Direkte Konstruktion Ist ein induzierender Vektor $g \in \mathbb{R}^n$ und eine Orthonormalbasis $\hat{Q} := \{\hat{q}_1, ..., \hat{q}_n\}$ mit $g^T \hat{q}_n \neq 0$ gegeben, so kann ein Gleichungssystem (1.17) auf folgende Weise konstruiert werden:

$$\hat{G}^T \nabla f(x) = \begin{bmatrix} \left\langle \left(g^T \hat{q}_1\right) \hat{q}_n^T - \left(g^T \hat{q}_n\right) \hat{q}_1^T, \nabla f(x) \right\rangle = 0 \\ \dots \\ \left\langle \left(g^T \hat{q}_{n-1}\right) \hat{q}_n^T - \left(g^T \hat{q}_n\right) \hat{q}_{n-1}^T, \nabla f(x) \right\rangle = 0 \end{bmatrix}. \tag{1.23}$$

Die Spalten $\hat{g}_i$, $i = 1, ..., n-1$ der Matrix $\hat{G}$ können also nach folgender Vorschrift bestimmt werden:

$$\hat{g}_i := \left(g^T \hat{q}_i\right) \hat{q}_n - \left(g^T \hat{q}_n\right) \hat{q}_i. \tag{1.24}$$

Die konstruierte Matrix $\hat{G}$ hat die in der Definition (24) geforderte Eigenschaften:

Lemma 27 *Die Matrix $\hat{G}$ hat den vollen Rang und ist orthogonal zum induzierenden Vektor g:*

$$\begin{aligned} Rang\left(\hat{G}\right) &= n-1 \\ \hat{G}^T g &= 0 \end{aligned}$$

Beweis. Indirekt wird hier angenommen, daß die Matrix G nicht vollen Ranges ist

$$Rang(G) < n-1.$$

Dann gibt es eine Linearkombination der Spalten $\hat{g}_i$, $i = 1, ..., n-1$ mit den reellen Koeffizienten $\alpha_i \in \mathbb{R}$, $i = 1, ..., n-1$, die nicht alle gleichzeitig gleich *Null* sind, für die folgende Bedingung gilt:

$$\begin{aligned} \sum_{i=1}^{n-1} \alpha_i \hat{g}_i &= \sum_{i=1}^{n-1} \alpha_i \left(\left(g^T \hat{q}_i\right) \hat{q}_n - \left(g^T \hat{q}_n\right) \hat{q}_i \right) \\ &= \sum_{i=1}^{n-1} \alpha_i \left(g^T \hat{q}_i\right) \hat{q}_n - \sum_{i=1}^{n-1} \alpha_i \left(g^T \hat{q}_n\right) \hat{q}_i = 0 \end{aligned}$$

Da die Spalten $\hat{q}_i$ der Matrix Q linear unabhängig sind und $g^T \hat{q}_n \neq 0$ gefordert wurde, müssen alle Koeffizienten α_i, $i = 1, ..., n-1$ gleichzeitig verschwinden. Dies ist aber äquivalent der linearen Unabhängigkeit der Vektoren $\hat{g}_i$ und widerspricht der indirekten Annahme.

Für die Spalten $\hat{g}_i$, $i = 1, ..., n-1$ gilt dann außerdem:

$$\hat{g}_i^T g := \left(g^T \hat{q}_i\right)\left(\hat{q}_n^T g\right) - \left(g^T \hat{q}_n\right)\left(\hat{q}_i^T g\right) = 0.$$

■

Korollar 28 *Für die Trajektorie*

$$T_{\hat{G}}(f) := \left\{ x \in \mathbb{R}^n | \quad \hat{G}^T \nabla f(x) = 0 \right\},$$

gilt dann:

$$x_0 \in T_{\hat{G}}(f).$$

Beweis. Für den vorgegebenen Punkt x_0 gilt:

$$\hat{G}^T \nabla f(x_0) = \hat{G}^T g = 0.$$

■

Beispiel 29 *Für eine zweidimensionale Zielfunktion $f : \mathbb{R}^2 \to \mathbb{R}$, eine Matrix $Q = I_2$ und einen Vektor $g := \nabla f(z_0)$ ist dann die Trajektorie $T_g(f, z_0)$ (vgl. Beispiel 20) gegeben mit:*

$$T(f, z_0) = \left\{ z \in \mathbb{R}^2 | \quad \frac{\partial f(z_0)}{\partial y}\frac{\partial f(z)}{\partial x} - \frac{\partial f(z_0)}{\partial x}\frac{\partial f(z)}{\partial y} = 0 \right\}. \tag{1.25}$$

Aus der Gleichung (1.25) folgt für jeden Trajektorienpunkt z

$$\hat{G}^T \nabla f(x) := \left(\frac{\partial f(z_0)}{\partial y}, -\frac{\partial f(z_0)}{\partial x}\right) \nabla f(z) = 0$$

so daß mit

$$\hat{g}_1 := \left(\frac{\partial f(z_0)}{\partial y}, -\frac{\partial f(z_0)}{\partial x}\right)^T$$

der Gradient $\nabla f(z)$ orthogonal zum Vektor $\hat{g}_1$ und damit parallel zum Gradienten $\nabla f(z_0)$ liegt.

Beispiel 30 *Für eine dreidimensionale Zielfunktion* $f:\mathbb{R}^3 \to \mathbb{R}$ *mit der Variablen* $z=(u,v,w)^T$, *einer Matrix* $Q=I_3$ *und einem Vektor* $g:=\nabla f(z_0)$ *ist dann die Trajektorie* $T_g(f,z_0)$ *gegeben mit:*

$$T(f,z_0)=\left\{ z\in\mathbb{R}^3 | \quad \begin{pmatrix} \frac{\partial f(z_0)}{\partial w}\frac{\partial f(z)}{\partial u}-\frac{\partial f(z_0)}{\partial u}\frac{\partial f(z)}{\partial w} \\ \frac{\partial f(z_0)}{\partial w}\frac{\partial f(z)}{\partial v}-\frac{\partial f(z_0)}{\partial v}\frac{\partial f(z)}{\partial w} \end{pmatrix}=0 \right\}. \tag{1.26}$$

Aus der Gleichung (1.26) folgt für jeden Trajektorienpunkt z

$$\hat{G}^T\nabla f(x):=\begin{pmatrix} \frac{\partial f(z_0)}{\partial w} & 0 & -\frac{\partial f(z_0)}{\partial u} \\ 0 & \frac{\partial f(z_0)}{\partial w} & -\frac{\partial f(z_0)}{\partial v} \end{pmatrix}\nabla f(z)=0$$

so daß mit

$$\begin{aligned} \hat{g}_1 &: = \left(\frac{\partial f(z_0)}{\partial w},0,-\frac{\partial f(z_0)}{\partial u}\right)^T \\ \hat{g}_2 &: = \left(0,\frac{\partial f(z_0)}{\partial w},-\frac{\partial f(z_0)}{\partial v}\right)^T \end{aligned}$$

der Gradient $\nabla f(z)$ *orthogonal zu Vektoren* $\hat{g}_1$ *bzw.* $\hat{g}_2$ *liegt.*
Aus der in der Vorschrift (1.23) aufgestellten Forderung

$$\nabla f(z_0)^T\hat{q}_n=g^T\hat{q}_n\neq 0$$

folgt dann

$$\frac{\partial f(z_0)}{\partial w}\neq 0.$$

Die Vektoren $\hat{g}_1$ *und* $\hat{g}_2$ *sind linear unabhängig. Für den Rang der Matrix* $\hat{G}$ *gilt also*

$$Rang\left(\hat{G}\right)=2.$$

Der Gradient $\nabla f(z)$, *der im eindimensionalen Kern der Matrix* G *liegt, ist also parallel zum Gradienten* $\nabla f(z_0)$ *(vgl. Bemerkung 23).*
Für die Vektoren $\hat{g}_1$ *und* $\hat{g}_2$ *gilt außerdem:*

$$\hat{g}_1^T\hat{g}_2=\frac{\partial f(z_0)}{\partial u}\frac{\partial f(z_0)}{\partial v}.$$

Das Skalarprodukt $\hat{g}_1^T\hat{g}_2$ *ist in der Regel ungleich Null; d.h. Vektoren* $\hat{g}_1$ *und* $\hat{g}_2$ *sind nicht orthogonal!*

Konstruktion mittels Spiegelung In manchen Fällen ist es von Vorteil, wenn die Matrix $G = (g_1, ..., g_{n-1})$ orthonormal ist. Die Spalten $g_1, ..., g_{n-1}$ müssen also so konstruiert werden, daß die folgende Bedingung erfüllt ist:

$$g_j g_k = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases} .$$

Dies ist bei der Matrix $\hat{G}$ nicht immer der Fall (s. Beispiel 30). Die Konstruktion einer orthogonalen Matrix G ist aufwendiger und kann z.B. mittels einer Orthogonaltransformation S einer beliebigen orthonormalen Basis $Q := \{q_1, ..., q_n\}$ durchgeführt werden. Wird der Basisvektor q_n auf den normierten induzierenden Vektor $\tilde{g} := \frac{g}{\|g\|}$ abgebildet

$$Sq_n = \tilde{g}, \tag{1.27}$$

so stellen die Abbilder der anderen Basisvektoren $q_1, ..., q_{n-1}$ die Spalten der gesuchten orthonormalen Matrix

$$G := (g_1 := Sq_1, ..., g_{n-1} := Sq_{n-1}).$$

Ist zufälligerweise $q_n = \tilde{g}$, so ist keine Transformation der Basisvektoren $q_1, ..., q_{n-1}$ notwendig. Die Matrix G ist dann definiert durch $G := (q_1, ..., q_{n-1})$. Ist $q_n \neq \tilde{g}$, so kann als Transformation S die Householder-Spiegelung $I_n - 2ww^T$ an der zu einem Einheitsvektor w ($\|w\| = 1$) orthogonalen Hyperebene $w^\perp$ genommen werden. Aus der geforderten Bedingung (1.27) kann der Vektor w abgeleitet werden (vgl. [2]):

Lemma 31 *Ist ein normierter Vektor $\tilde{g}$ und eine orthonormale Basis $Q := \{q_1, ..., q_n\}$ mit $q_n \neq \tilde{g}$ gegeben, so existiert mit $w := \frac{\tilde{g} - q_n}{\|\tilde{g} - q_n\|}$ eine eindeutige Spiegelung $\tilde{S} := I - 2ww^T$ mit der Eigenschaft (1.27).*

Für die mit Hilfe der Spiegelung $\tilde{S}$ konstruierte Matrix $\tilde{G}$ können folgende Eigenschaften nachgewiesen werden:

Korollar 32 *Ist ein normierter induzierender Vektor $\tilde{g}$ und eine orthonormale Basis $Q := \{q_1, ..., q_n\}$ gegeben und wird eine Spiegelung $\tilde{S} := I - 2ww^T$, die die Bedingung (1.27) erfüllt, konstruiert, so gilt für die Matrix*

$$\tilde{G} := (\tilde{g}_1 := \tilde{S}q_1, ..., \tilde{g}_{n-1} := \tilde{S}q_{n-1})$$

$$\begin{aligned} \tilde{G}^T \tilde{G} &= I_{n-1} \\ \tilde{G}^T g &= 0. \end{aligned}$$

Beweis. Da $\tilde{S}$ eine Orthonormaltransformation ist, folgt für die Matrix $\tilde{G}$ und den induzierenden Vektoren g:

$$\left(\tilde{G},\tilde{g}\right)^T\left(\tilde{G},\tilde{g}\right)=\left(\tilde{S}Q\right)^T\left(\tilde{S}Q\right)=Q^T\tilde{S}^T\tilde{S}Q=Q^TQ=I_n.$$

■

Das Gleichungssystem (1.17) kann also wie folgt aufgestellt werden:

$$\tilde{G}^T\nabla f\left(x\right)=\begin{bmatrix}\left\langle\tilde{S}q_1,\nabla f\left(x\right)\right\rangle=0\\ \ldots\\ \left\langle\tilde{S}q_{n-1},\nabla f\left(x\right)\right\rangle=0\end{bmatrix}. \tag{1.28}$$

Korollar 33 *Für die Trajektorie*

$$T_{\tilde{G}}\left(f\right):=\left\{x\in\mathbb{R}^n|\quad\tilde{G}^T\nabla f\left(x\right)=0\right\},$$

gilt dann:

$$x_0\in T_{\tilde{G}}\left(f\right).$$

Beweis. Für den vorgegebenen Punkt x_0 gilt:

$$\tilde{G}^T\nabla f\left(x_0\right)=\tilde{G}^Tg=0.$$

■

Beispiel 34 *Sei* $f:\mathbb{R}^3\rightarrow\mathbb{R}$ *eine dreidimensionale Zielfunktion mit der Variablen* $z=(u,v,w)^T$ *und die Matrix* $Q=I_3$ *gegeben. Seien weiterhin Vektoren* g *und* $\tilde{g}$ *definiert durch:*

$$\begin{aligned}g&=\left(g_u,g_v,g_w\right):=\nabla f\left(z_0\right)\\ \tilde{g}&=\left(\tilde{g}_u,\tilde{g}_v,v_w\right):=\nabla f\left(z_0\right)/\left\|\nabla f\left(z_0\right)\right\|.\end{aligned}$$

Der die gesuchte Spiegelung $\tilde{S}$ *induzierende Vektor* w *ist dann gegeben mit:*

$$w=\left(\begin{array}{ccc}\tilde{g}_u,&\tilde{g}_v,&\tilde{g}_w-1\end{array}\right)^T/\left\|\left(\begin{array}{ccc}\tilde{g}_u,&\tilde{g}_v,&\tilde{g}_w-1\end{array}\right)\right\|.$$

Die Matrix $\tilde{S}$ *der entsprechenden Housholder-Spiegelung kann wie folgt berechnet werden:*

$$\tilde{S}=I_3-\frac{2}{\left\|\left(\begin{array}{ccc}\tilde{g}_u,&\tilde{g}_v,&\tilde{g}_w-1\end{array}\right)\right\|^2}\begin{pmatrix}\tilde{g}_u^2&\tilde{g}_u\tilde{g}_v&\tilde{g}_u\left(\tilde{g}_w-1\right)\\ \tilde{g}_v\tilde{g}_u&\tilde{g}_v^2&\tilde{g}_v\left(\tilde{g}_w-1\right)\\ \left(\tilde{g}_w-1\right)\tilde{g}_u&\left(\tilde{g}_w-1\right)\tilde{g}_v&\left(\tilde{g}_w-1\right)^2\end{pmatrix}.$$

Ist der Gradient $\nabla f(z_0)$ der Zielfunktion f im Punkt x_0 beispielsweise gleich

$$\nabla f(z_0) = \begin{pmatrix} a, & a, & 0 \end{pmatrix}^T, \qquad 0 \neq a \in \mathbb{R}$$

so können der Vektor w und die Matrix $\tilde{S}$ explizit angegeben werden

$$w = \begin{pmatrix} \frac{1}{2}, & \frac{1}{2}, & -\frac{\sqrt{2}}{2} \end{pmatrix}^T$$

$$\tilde{S} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{2}}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{pmatrix}.$$

Für die Trajektorie $T_g(f, z_0)$ induzierende Matrix $\tilde{G}$

$$\tilde{G} := (\tilde{S}e_1, \tilde{S}e_2) = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{2}}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}^T,$$

sind die geforderten Bedingungen trivial erfüllt:

$$\begin{aligned} \tilde{G}^T\tilde{G} &= I_2 \\ \tilde{G}^T\nabla f(z_0) &= 0. \end{aligned}$$

Die Trajektorie $T_g(f, z_0)$ ist dann gegeben mit:

$$T(f, z_0) = \left\{ z \in \mathbb{R}^3 | \quad \begin{pmatrix} \frac{1}{2}\frac{\partial f(z)}{\partial u} - \frac{1}{2}\frac{\partial f(z)}{\partial v} + \frac{\sqrt{2}}{2}\frac{\partial f(z)}{\partial w} \\ -\frac{1}{2}\frac{\partial f(z)}{\partial u} + \frac{1}{2}\frac{\partial f(z)}{\partial v} + \frac{\sqrt{2}}{2}\frac{\partial f(z)}{\partial w} \end{pmatrix} = 0 \right\}.$$

Bemerkung 35 *Die entsprechende Bedingung $g^T\hat{q}_n \neq 0$ ist hierbei nicht mehr relevant. Die orthonormale Basis $Q := \{q_1, ..., q_n\}$ kann also beliebig gewählt werden.*

1.3.3 Zugang über eine vorgegebene Richtung

Die Trajektorie $T_g(f)$ kann auch als die Menge der Punkte angesehen werden, in denen die Richtung des Gradienten der Zielfunktion f mit der vorgegebenen Richtung übereinstimmt (s. Satz 25):

$$T_g(f) = \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda g \right\}. \tag{1.29}$$

Da der Gradient $\nabla f(x)$ der Zielfunktion f in kritischen Punkten der Nullvektor ist, gehören auch diese Punkte (man wähle $\lambda = 0$) zu der auf solche Art und Weise definierten Menge (s. Abbildung 1.4, vgl. Beispiel 20).

Anderseits kann leicht gezeigt werden, daß jeder nichtkritische Punkt einer klassischen Newton-Trajektorie eindeutig zugeordnet werden kann.

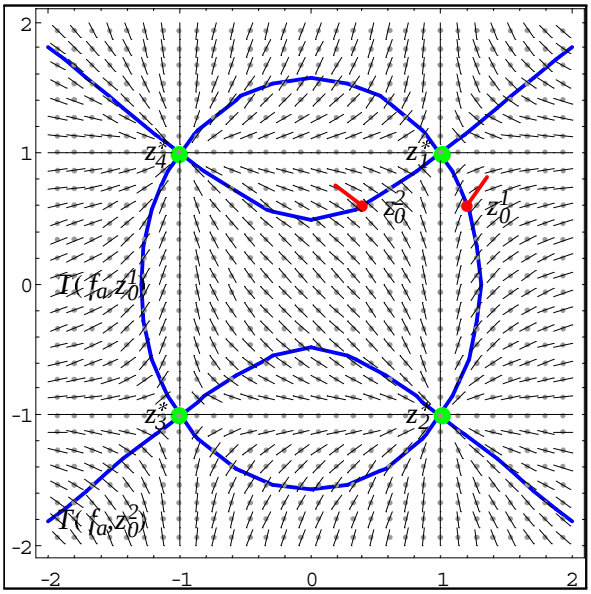

Abbildung 1.4: Beispieltrajektorien T_1 und T_2 im Gradientenfeld der Testfunktion f_a

Korollar 36 *Die Schnittmenge zwei verschiedener klassischen Newton-Trajektorien ist mit der Menge der kritischen Punkte der Zielfunktion f identisch:*

$$\left(T_{g_1}(f) \neq T_{g_2}(f)\right) \Rightarrow Krit(f) = T_{g_1}(f) \cap T_{g_2}(f).$$

Beweis. Aus (1.19) folgt direkt $Krit(f) \subset T_{g_1}(f) \cap T_{g_2}(f)$. Existiert ein Punkt $x \in T_{g_1}(f) \cap T_{g_2}(f)$ mit $\nabla f(x) \neq 0$, so gilt

$$\lambda_1 g_1 = \nabla f(x) = \lambda_2 g_2$$

und mit $\lambda_1 \lambda_2 \neq 0$ folgt

$$g_1 \| g_2.$$

Die klassischen Newton-Trajektorien $T_{g_1}(f)$ und $T_{g_2}(f)$ sind aber genau dann verschieden, wenn $g_1 \nparallel g_2$. Dies führt zum Wiederspruch. ■

Bemerkung 37 *Weil die induzierende Funktion H für die verallgemeinerten Newton-Trajektorien $T_H(f)$ (s. Definition 10) mit unterschiedlichen Ansätzen (vgl. Abschnitte 1.4.3 oder 1.4.4) gewählt werden kann, gilt die Behauptung aus dem Korollar 36 nur für die klassischen Newton-Trajektorien und* ***nicht*** *für die verallgemeinerte Newton-Trajektorien $T_H(f)$.*

Die vorgegebene Richtung g ist vor allem dann von direkter Bedeutung, wenn ein Ausgangspunkt für die Rekonstruktion der Trajektorie bestimmt werden muß (vgl. 1.22). Auch ein bekannter oder mittels klassischen Optimierung berechneter kritischer Punkt kann als Ausgangspunkt für das Rekonstruktionsverfahren genutzt werden (vgl. [33]).

1.4 Richtungsfeld-Trajektorie

Die Richtungs des Gradienten der Zielfunktion f bleibt entlang der klassischen Newton-Trajektorie konstant (vgl.26). Unter Umständen kann auch eine variable Richtung $g(x)$ zugelassen werden. Diese Betrachtungsweise wird hier als Anlaß zur Definition und Untersuchung durch ein Richtungsfeld induzierten Trajektorien genommen.

1.4.1 Zugang über ein Richtungsfeld

Indem die konstante Richtung g durch ein vorgebenes Richtungsfeld $r : \mathbb{R}^n \to \mathbb{R}^n$ ersetzt wird, wird hier eine neue Trajektorie $T_r(f)$ als die Menge der Punkte definiert, in denen die Richtung des Gradienten der Zielfunktion f mit dem Richtungsfeld r übereinstimmt.

Definition 38 *Sei r eine stetige Abbildung $r : \mathbb{R}^n \to \mathbb{R}^n$, die keine Nullstellen hat:*

$$r^{-1}(0) = \emptyset. \tag{1.30}$$

Die durch das Richtungsfeld $r(x)$ induzierte Trajektorie $T_r^o(f)$ (kurz Richtungsfeld-Trajektorie) ist definiert als:

$$T_r^o(f) := \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda(x) \in \mathbb{R}} \nabla f(x) = \lambda(x)\, r(x) \right\}.$$

Die neu definierte Trajektorie $T_r^o(f)$ enthält alle kritischen Punkte der Zielfunktion $f(x)$. Im Falle eines konstanten Richtungsfeldes $r(x) \equiv g$ entspricht die Trajektorie $T_r^o(f)$ der durch Vektor g induzierten klassischen Newton-Trajektorie $T_g(f)$.
Es ist nicht immer möglich ein Richtungsfeld zu konstruieren, das gewünschte Eigenschaften (vgl. Abschnitte 1.4.3 und 1.4.4) und gleichzeitig keine Nullstellen hat. Mit einer Umformulierung der Definition 38 werden deshalb die Nullstellen des Richtungsfeldes zugelassen und in die Trajektorie mitaufgenommen:

Definition 39 *Sei r eine stetige Abbildung $r : \mathbb{R}^n \to \mathbb{R}^n$ (mit Nullstellen). Die* ***durch das Richtungsfeld*** $r(x)$ ***induzierte Trajektorie*** $T_r(f)$ *ist definiert als:*

$$T_r(f) := \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda(x) \in \mathbb{R}} \nabla f(x) \, \|r(x)\| = \lambda(x) \, r(x) \right\}. \tag{1.31}$$

Die Nullstellen der Funktion $r(x)$ sind hier in der Trajektorie $T_r(f)$ enthalten. Ein Richtungsfeld mit unendlich vielen Nullstellen ist deshalb, für den praktischen Ansatz, wenig sinnvoll. Besitzt das Richtungsfeld nur endlich viele Nullstellen, so können diese, isolierte Ein-Punkt-Komponenten der Trajektorie bilden. Solche Komponenten sind aber für unsere Anwendung uninteressant und können deshalb vernachlässigt werden. Im Zusammenhang mit der Richtungsfeld-Trajektorie wird im weiteren stets die erweiterte Definition 39 gemeint.
Im folgenden Lemma wird gezeigt, daß der Parameter $\lambda(x)$ als Norm des Gradienten $\nabla f(x)$ berechnet werden kann.

Lemma 40 *Sei eine stetige Abbildung $r : \mathbb{R}^n \to \mathbb{R}^n$ und eine durch $r(x)$ induzierte Trajektorie $T_r(f)$ gegeben, dann gilt:*

$$\begin{aligned} Krit(f) &\subset T_r(f) \\ \forall_{x \in T_r(f)} \nabla f(x) \, \|r(x)\| &= \pm \|\nabla f(x)\| \, r(x). \end{aligned}$$

Beweis. Betrachtet werden die kritischen Punkte der Zielfunktion f, die Nullstellen des Richtungsfeldes r und die übrigen Trajektorienpunkte.
Für jeden kritischen Punkt x^* ist die Bedingung (1.31) mit $\lambda(x^*) := \|\nabla f(x^*)\| = 0$ trivialerweise erfüllt:

$$\nabla f(x^*) \, \|r(x^*)\| = 0 = \lambda(x^*) \, r(x^*).$$

Für eine Nullstelle x^r des Richtungsfeldes ist die Wahl des Parameters $\lambda(x^r)$ nicht relevant. Insbesondere kann also auch die Norm des Gradienten angenommen werden:

$$\lambda(x^r) := \|\nabla f(x^r)\|.$$

Ist allerdings für ein Trajektorienpunkt $x \in T_r(f)$ der Wert des Richtungsfeldes ungleich *Null*, so folgt:

$$\frac{\nabla f(x)}{\lambda(x)} = \frac{r(x)}{\|r(x)\|}.$$

Für den Parameter $\lambda(x)$ gilt dann:

$$\lambda(x) = \pm \|\nabla f(x)\|.$$

■

1.4.2 Zugang über ein nichtlineares Gleichungssystem

Auch die modifizierte Trajektorie kann als Lösungsmenge eines nichtlinearen Gleichungssystems bzw. als Nullmenge einer bestimmten Hilfsfunktion $H_r : \mathbb{R}^n \to \mathbb{R}^{n-1}$ dargestellt werden. Hierzu wird im weiteren (s. Abschnitt *Konstruktion des Gleichungssystems*) eine variable Matrix $G_r(x)$ mit folgenden Eigenschaften konstruiert:

$$\forall_{x \in \mathbb{R}^n} \quad r(x) \in \ker G_r^T(x) \tag{1.32}$$

$$\forall_{x \in \mathbb{R}^n} \quad Rang\left(G_r(x)\right) = n-1. \tag{1.33}$$

Das Gleichungssystem (1.17) kann mit Hilfe der Matrix $G_r(x)$ allgemeiner dargestellt werden:

$$G_r^T(x) \nabla f(x) = 0. \tag{1.34}$$

Die Hilfsfunktion $H_r(x)$ wird dann wie folgt definiert (vgl. Vorschrift 1.18):

$$H_r(x) := \begin{cases} G_r^T(x) \nabla f(x), & r(x) \neq 0 \\ 0, & r(x) = 0 \end{cases}. \tag{1.35}$$

Die mögliche Unstetigkeitsstellen der Funktion $H_r(x)$ in den (endlich vielen) Nullstellen des Richtungsfeldes $r(x)$ können von der Trajektorie $T_r^o(f)$ isoliert werden und sind deshalb für unsere Betrachtung uninteressant.
Die als Nullmenge der Funktion $H_r(x)$ definierte, verallgemeinerte Newton-Trajektorie $T_{H_r}(f)$ (vgl. Definition 10) stimmt mit der durch das Richtungsfeld $r(x)$ induzierten Trajektorie $T_r(f)$ überein.

Satz 41 *Für ein Richtungsfeld $r : \mathbb{R}^n \to \mathbb{R}^n$ und die mit Hilfe der Vorschrift (1.35) aufgestellte Funktion H_r, gilt:*

$$T_r(f) = T_{H_r}(f).$$

Beweis. Da die Nullstellen der Abbildung $r(x)$ trivialerweise zu beiden Trajektorien $T_r(f)$ und $T_{H_r}(f)$ gehören, wird im weiteren $r(x) \neq 0$ vorausgesetzt.
Für jeden Punkt $x \in T_{H_r}(f)$ ist die Bedingung

$$H_r(x) := G_r^T(x) \nabla f(x) = 0$$

mit der Aussage äquivalent, daß der Gradient $\nabla f(x)$ im Kern der variablen Matrix $G_r(x)$ liegt:

$$\nabla f(x) \in \ker G_r^T(x).$$

Der entsprechende Richtungsvektor $r(x)$ liegt ebenfalls im Kern von $G_r(x)$ (s. Bedingung 1.32). Da der Kern der Matrix $G_r(x)$ überall eindimensional sein soll (s. Bedingung 1.33) und $r(x) \neq 0$, ist der Gradient $\nabla f(x)$ linear abhängig von $r(x)$, so daß $x \in T_r(f)$ folgt.

Andererseits gilt für jeden $x \in T_r(f)$ mit $r(x) \neq 0$ auch

$$G_r^T(x) \nabla f(x) = \pm \frac{\|\nabla f(x)\|}{\|r(x)\|} G_r^T(x) r(x) = 0,$$

so daß $H_r(x) = 0$ und damit $x \in T_{H_r}(f)$ folgt. ■

Konstruktion des Gleichungssystems

Um eine variable Matrix $G_r(x)$ und damit auch eine Funktion H_r für ein Richtungsfeld $r : \mathbb{R}^n \to \mathbb{R}^n$ zu konstruieren, wird zuerst eine beliebige orthonormale Basis $Q = \{q_1, ..., q_n\}$ von $\mathbb{R}^n$ ausgesucht. Die Richtungsfunktion $r(x)$ wird dann bzgl. der Basis Q in n Koordinatenfunktionen $r_i : \mathbb{R}^n \to \mathbb{R}^1$ zerlegt :

$$\begin{aligned} r(x) &= \sum_{i=1}^{n} r_i(x) q_i \\ r_i(x) &= q_i^T r(x). \end{aligned}$$

Die Konstruktionsmethoden aus dem Abschnitt (1.3) können dann leicht übertragen werden.

Direkte Konstruktion Das nichtlineare Gleichungssystem (1.23) wird umgeschrieben in:

$$\begin{gathered} \left\langle r_1(x) q_n^T - r_n(x) q_1^T, \nabla f(x) \right\rangle = 0 \\ \cdots \\ \left\langle r_{n-1}(x) q_n^T - r_n(x) q_{n-1}^T, \nabla f(x) \right\rangle = 0. \end{gathered} \tag{1.36}$$

Für die entsprechend definierte Matrix,

$$\hat{G}_r(x) := (r_1(x) q_n^T - r_n(x) q_1^T, ..., r_{n-1}(x) q_n^T - r_n(x) q_{n-1}^T),$$

liegt die Richtung $r(x)$ stets im Kern der Matrix $\hat{G}_r(x)$.

Lemma 42 *Die Matrix $\hat{G}_r(x)$ ist überall orthogonal zu $r(x)$*

$$\hat{G}_r(x)^T r(x) = 0.$$

Beweis. Sei $\hat{g}_l(x)$ die l-te Spalte der Matrix $\hat{G}_r(x)$, dann gilt:

$$\begin{aligned}\hat{g}_l^T \hat{r}(x) &= \left(r_l(x)\, q_n^T - r_n(x)\, q_l^T\right)^T \sum_{i=1}^{n} r_i(x)\, q_i \\ &= r_l(x)\, r_n(x) - r_n(x)\, r_l(x) = 0.\end{aligned}$$

■

Mit der variablen Matrix $\hat{G}_r(x)$ ist dann eine entsprechende Hilfsfunktion $\hat{H}_r$ gegeben durch:

$$\hat{H}_r(x) := \hat{G}_r^T(x) \nabla f(x) = 0. \tag{1.37}$$

Es soll hier beachtet werden, daß für die Nullstellen des Richtungsfeldes $r(x)$ die Bedingung $\hat{H}_r(x) = 0$ mit $\hat{G}_r(x) = 0_{n,n-1}$ natürlicherweise erfüllt ist.

Bemerkung 43 *Die folgende Bedingung*

$$r_n(x) := \langle r(x), q_n \rangle \neq 0,$$

kann leider im allgemeinen nicht für jeden Punkt $x \in B$ gesichert werden, so daß die Matrix $\hat{G}_r(x)$ nicht notwendigerweise überall vollen Rang hat. Das kann zu numerischen Instabilitäten des Rekonstruktionsverfahren führen (vgl. Abschnitt 3.1).

Konstruktion mittels Spiegelung Die hier vorgestellte Konstruktionsprozedur basiert auf einer variablen Householder-Spiegelung

$$\tilde{S}_r(x,p) : \mathbb{R}^{n \times n} \to \mathbb{R}^n.$$

Ist das Richtungsfeld $r : \mathbb{R}^n \to \mathbb{R}^n$ in einem Punkt $x \in \mathbb{R}^n$ ungleich $Null$, so kann ein normierter Richtungsvektor $\tilde{r}(x)$ konstruiert werden:

$$\tilde{r}(x) := \frac{r(x)}{\|r(x)\|}.$$

Mit dem variablen Vektor $w(x)$,

$$w(x) := \frac{\tilde{r}(x) - q_n}{\|\tilde{r}(x) - q_n\|},$$

wird die Spiegelung $\tilde{S}_r(x,p)$ wie folgt definiert:

$$\tilde{S}_r(x,p) := I_n - 2w(x)\, w^T(x).$$

Wie im Lemma 31 bereits gezeigt wurde, wird dabei der Basis-Vektor q_n auf $\tilde{r}(x)$ gespiegelt:

$$\tilde{S}(x, q_n) = \tilde{r}(x).$$

Die Bilder der anderen Basisvektoren stellen dann die Spalten der gesuchten variablen orthonormalen Matrix $\tilde{G}_r(x)$,

$$\tilde{G}_r(x) := (\tilde{S}(x, q_1), ..., \tilde{S}(x, q_{n-1})). \tag{1.38}$$

Lemma 44 *Sei $r : \mathbb{R}^n \to \mathbb{R}^n$ ein Richtungsfeld und $Q = \{q_1, ..., q_n\}$ eine orthonormale Basis von $\mathbb{R}^n$. Für die entsprechend der Vorschrift (1.38) konstruierte variable Matrix $\tilde{G}_r(x)$ gilt überall:*

$$\begin{aligned} \tilde{G}_r^T(x)\,\tilde{G}_r(x) &= I_{n-1} \\ \tilde{G}_r^T(x)\,r(x) &= 0. \end{aligned}$$

Beweis. Für jeden einzelnen Punkt x kann der Beweis des Lemmas (31) direkt übertragen werden. ■

Das nichtlineare Gleichungssystem (1.28) wird umgeschrieben in:

$$\tilde{G}_r^T(x)\,\nabla f(x) = \begin{bmatrix} \left\langle \tilde{S}(x, q_1), \nabla f(x) \right\rangle = 0 \\ ... \\ \left\langle \tilde{S}(x, q_{n-1}), \nabla f(x) \right\rangle = 0 \end{bmatrix}. \tag{1.39}$$

1.4.3 Richtungsfeld-Trajektorie mit mehreren vorgegebenen Startpunkten

Wie im Beispiel 20 gezeigt wurde, kann eine verallgemeinerte Newton-Trajektorie $T_H(f)$ aus mehreren Komponenten bestehen. Der in Kapitel 2 vorgestellte rekursive Rekonstruktionsalgorithmus führt nicht immer zu allen Trajektorienkomponenten. In einem solchen Fall werden möglicherweise nicht alle kritischen Punkte der Zielfunktion $f(x)$ gefunden. Insbesondere kann es passieren, daß die Trajektorie nur in einer kleinen Region des Bereichs, der durchsucht werden soll, steckenbleibt.
Um diese Situation weitgehend auszuschließend, wird hier eine Richtungsfeld-Trajektorie definiert, die außer den gesuchten kritischen Punkten bis zu $n+1$ vorgegebene Punkte x_k, $k = 0, ..., m \leq n$ enthält. Die vorgegebenen Punkte x_k können als Startpunkte für mehrere Komponenten der Trajektorie genutzt werden. Die Punkte x_k sollten als Ecken eines m-dimensionalen Polyeders in $\mathbb{R}^n$ gewählt werden (s. Bemerkung 51). Werden die Punkte den Vorkenntnissen und Wünschen entsprechend verteilt, so wird damit sichergestellt, daß die Trajektorie $T_H(f)$ mit den Punkten x_k auch besonders interessante Regionen, wo die kritischen Punkte vermutet werden, erreicht.

Richtungsfeld

Konvexe Kombination vorgegebener Richtungen Sollen die Punkte x_k, $k = 0, ..., m \leq n$ der Trajektorie angehören, so wird ein Richtungsfeld $\widehat{r}(x)$ als konvexe Kombination der Gradienten $\nabla f(x_k)$ der Zielfunktion f in den vorgegebenen Punkten x_k bestimmt:

$$\widehat{r}(x) := \sum_{k=0}^{m} \mu_k(x) \nabla f(x_k).$$

Die Bestimmung der Koeffizienten $\mu_k(x)$ und damit der Richtung $\widehat{r}(x)$ in einem Punkt $x \in \mathbb{R}^n$ kann in zwei Schritten erfolgen:

1. Der Punkt x wird mit Hilfe einer orthogonalen Abbildung P auf die durch die Punkte x_k aufgespannte lineare Mannigfaltigkeit $M_{x_0,x_1,...,x_m}$, projiziert:

$$M_{x_0,x_1,...,x_m} := \left\{ x \in \mathbb{R}^n | \quad x = \sum_{k=0}^{m} \mu_k(x) x_k \wedge \sum_{k=0}^{m} \mu_k = 1 \right\}.$$

2. Der projizierte Punkt $P(x)$ wird dann als konvexe Kombination der vorgegebenen Punkte x_k dargestellt:

$$P(x) := \sum_{k=0}^{m} \mu_k(x) x_k. \tag{1.40}$$

Dies führt zum folgenden Approximationsproblem:

Problem 45 *Für $m+1$ vorgegebene Punkte $x_0, ..., x_m \in \mathbb{R}^n$, $m \leq n$ werden konvexe Koeffizienten $\mu_0, ..., \mu_m \in \mathbb{R}$, $\sum_{k=0}^{m} \mu_k = 1$ gesucht, für die gilt:*

$$\left\| x - \sum_{k=0}^{m} \mu_k(x) x_k \right\| \to \min.$$

Mit v_i, $i = 1, ..., m$ werden im weiteren die Differenzenvektoren $x_i - x_0$ und mit $G_{v_1,...,v_m}$ die Gramsche Matrix $G_{v_1,...,v_m}$ bezeichnet:

$$G_{v_1,...,v_m} := \begin{pmatrix} v_1^T v_1 & \cdots & v_1^T v_m \\ \vdots & & \vdots \\ v_m^T v_1 & \cdots & v_m^T v_m \end{pmatrix}.$$

Die Parameter $\mu_1(x), ..., \mu_m(x)$ können als Lösung des folgenden linearen Gleichungssystems bestimmt werden:

$$G_{v_1,...,v_m} \begin{pmatrix} \mu_1(x) \\ \vdots \\ \mu_m(x) \end{pmatrix} = \begin{pmatrix} v_1^T(x - x_0) \\ \vdots \\ v_m^T(x - x_0) \end{pmatrix}. \tag{1.41}$$

Der Parameter $\mu_0(x)$ ist dann durch folgende Bedingung bestimmt:

$$\mu_0(x) = 1 - \sum_{k=1}^{m} \mu_k.$$

Bemerkung 46 *Sind die Vektoren $v_1, ..., v_m$ linear unabhängig, so ist die Gramsche Matrix $G_{v_1,...,v_m}$ regulär, und die Parameter $\mu_1(x), ..., \mu_m(x)$ sind eindeutig bestimmbar.*

Unter der Regularitätsvoraussetzung der Matrix $G_{v_1,...,v_m}$ stimmt das Richtungsfeld $\widehat{r}(x)$ in den vorgegebenen Punkten x_k mit der Richtung der Gradienten $\nabla f(x_k)$ der Funktion f überein (vgl. Satz 49).

Beispiel 47 *Für die zweidimensionale Zielfunktion $f_a : \mathbb{R}^2 \to \mathbb{R}$ aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y,$$

*werden **zwei** beliebige Punkte z_0, $z_1 \in \mathbb{R}^2$ ausgesucht. Für die in diesen Punkten vorgegebene Richtungen gilt:*

$$g_i := \nabla f(z_i) = \left(x_i^2 - 1, 1 - y_i^2\right), \qquad i = 0, 1.$$

Mit dem Vektor $v := z_1 - z_0$ und der 1×1 Matrix $G_v := v^2$ können die Koeffizienten $\mu_1(z)$ und $\mu_0(z)$ leicht bestimmt werden durch:

$$\begin{aligned} \mu_1(z) &= \frac{v^T(z - z_0)}{v^T v}, \\ \mu_0(z) &= 1 - \mu_1(z) = \frac{v^T(z_1 - z)}{v^T v}. \end{aligned}$$

Das Richtungsfeld ist dann gegeben mit:

$$\widehat{r}(z) = \frac{v^T(z_1 - z)}{v^T v} g_0 + \frac{v^T(z - z_0)}{v^T v} g_1.$$

Für die Punkte $z_0 := (-2,-1)^T$ *und* $z_1 := (1,-2)^T$ *ergibt sich dann*

$$\begin{array}{ll} v = (3,-1)^T & G_v = (10) \\ \mu_0 = \frac{1}{10}(5-3x+y) & g_0 = (3,0)^T \\ \mu_1 = \frac{1}{10}(-5-3x+y) & g_1 = (0,-3)\,. \end{array}$$

Das in der Abbildung 1.5 dargestellte Richtungsfeld $\widehat{r}(x)$ *ist dann gegeben durch:*

$$\widehat{r}(x) = \frac{3}{10}\begin{pmatrix} 5-3x+y \\ -5-3x+y \end{pmatrix}.$$

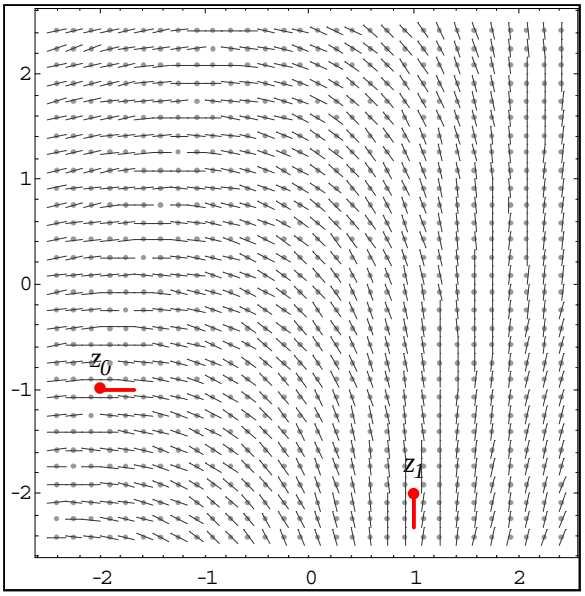

Abbildung 1.5: Richtungsfeld für die Richtungsfeld-Trajektorie mit zwei Startpunkten

Beispiel 48 *Für die zweidimensionale Zielfunktion* $f_a : \mathbb{R}^2 \to \mathbb{R}$ *aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y,$$

werden hier ***drei*** *Punkte* $z_0, z_1, z_2 \in \mathbb{R}^n$ *vorgegeben:*

$$\begin{array}{rcl} z_0 & = & (-2,-1)^T \\ z_1 & = & (1,-2)^T \\ z_2 & = & (0,1)^T\,. \end{array}$$

Für die Vektoren v_1, v_2 und die Gramsche Matrix G_{v_1,v_2} ergibt sich dann:

$$v_1 = (3,-1)^T \quad v_2 = (2,2)^T,$$

$$G_v = \begin{pmatrix} 10 & 4 \\ 4 & 8 \end{pmatrix}.$$

Die Koeffizienten $\mu_i(x)$ und die Richtungsvektoren g_i können dann leicht berechnet werden:

$$\begin{array}{ll} \mu_0 = \frac{1}{8}(1-3x-y) & g_0 = (3,0)^T \\ \mu_1 = \frac{1}{8}(5+x+3y) & g_1 = (0,-3)^T \\ \mu_2 = \frac{1}{4}(1+x-y) & g_2 = (-1,0). \end{array}$$

Das in der Abbildung 1.6 dargestellte Richtungsfeld $\widehat{r}(x)$ ist dann gegeben mit:

$$\widehat{r}(z) = \frac{1}{4}\begin{pmatrix} -1-5x-3y \\ -3-3x+3y \end{pmatrix}.$$

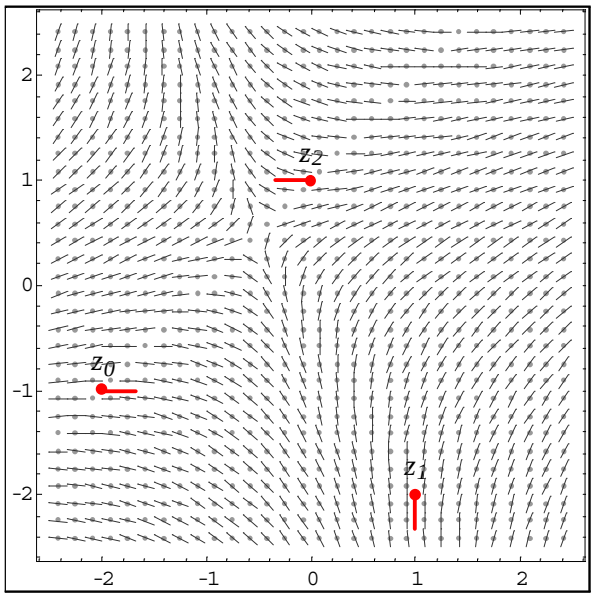

Abbildung 1.6: Richtungsfeld für die Richtungsfeld-Trajektorie mit drei Startpunkten

Im Satz 49 wird bewiesen, daß die durch das Richtungsfeld $\widehat{r}(x)$ induzierte Trajektorie $T_{\widehat{r}}(f)$ außer den kritischen Punkte der Zielfunktion f auch die vorgegebenen Punkte x_k enthält .

Satz 49 *Für die vorgegebenen Punkte $x_0, x_1, ..., x_m$ sei die Gramsche Matrix $G_{v_1,...,v_m}$ regulär. Für die Trajektorie*

$$T_{\hat{r}}(f) := \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda(x)\in\mathbb{R}} \nabla f(x) \left\| \hat{r}(x) \right\| = \lambda(x) \hat{r}(x) \right\},$$

gilt dann:

(i) Die Trajektorie enthält alle kritischen Punkte:

$$Krit(f) \subset T_{\hat{r}}(f).$$

(ii) Die Trajektorie enthält alle vorgebenen Punkte:

$$\forall_{k=1,...,m} x_k \in T_{\hat{r}}(f).$$

Beweis. Die Behauptung (i) folgt direkt aus dem Lemma 40.
Um die Behauptung (ii) zu beweisen, werden zuerst die variablen Koeffizienten $\mu_0(x), \mu_1(x), ..., \mu_m(x)$ bestimmt.
Für den Ausgangspunkt x_0 ist die rechte Seite des Gleichungssystems (1.41) gleich $Null$, so daß aufgrund der Regularität der Matrix $G_{v_1,...,v_m}$ die Parameter $\mu_1(x), ..., \mu_m(x)$ verschwinden müssen. Der Parameter $\mu_0(x_0)$ ist dann gleich 1:

$$\mu_k(x_0) = \begin{cases} 0, & k = 1, ..., m \\ 1, & k = 0 \end{cases}.$$

Für einen vorgegebenen Punkt x_j ($j \in \{1, 2, ..., m\}$) nimmt das Gleichungssystem (1.41) folgende Form an

$$\begin{pmatrix} v_1^T v_1 & \cdots & v_1^T v_m \\ \vdots & & \vdots \\ v_m^T v_1 & \cdots & v_m^T v_m \end{pmatrix} \begin{pmatrix} \mu_1(x) \\ \vdots \\ \mu_m(x) \end{pmatrix} = \begin{pmatrix} v_1^T (x_j - x_0) \\ \vdots \\ v_m^T (x_j - x_0) \end{pmatrix} = \begin{pmatrix} v_1^T v_j \\ \vdots \\ v_m^T v_j \end{pmatrix}.$$

Die j-te Spalte der Gramschen Matrix $G_{v_1,...,v_m}$ stimmt dann mit der rechten Seite des Gleichungssystems überein, so daß mit

$$\mu_k(x_0) = \begin{cases} 0, & k \neq j \\ 1, & k = j \end{cases},$$

die eindeutige Lösung des Systems gefunden wurde.

Für jeden vorgegebenen Punkt x_j ($j \in \{0, 1, ..., m\}$) gilt also:

$$\hat{r}(x_j) := \sum_{k=0}^{m} \mu_k(x_j) \nabla f(x_k) = \nabla f(x_j)$$

und der Punkt gehört damit der Trajektorie $T_{\hat{r}}(f)$. ■

Effiziente Konstruktion des Richtungsfeldes Die Lösung eines linearen Gleichungssystem (1.41) in jedem Punkt x kann sehr umständlich sein. Da sich die Koeffizienten des Systems nicht ändern, ist es naheliegend, das System in eine einfachere Form zu bringen. Mittels Gram-Schmidt-Orthogonalisierung der Vektoren v_i wird also zunächst ein Orthonormalsystem $\{w_1, ..., w_{m\prime}\}$ konstruiert. Die vorgegebenen Punkte $x_1, ..., x_m$ können dann mit x_0 und den Basisvektoren $w_1, ..., w_{m\prime}$, wie folgt dargestellt werden:

$$x_i = x_0 + \sum_{k=0}^{m'} v_i^T w_k w_k.$$

Die Richtungsvektoren $g_1, ..., g_m$ müssen dabei in die neue Basisdarstellung übertragen werden:

$$g_i = g_0 + \sum_{k=0}^{m'} v_i^T w_k \check{g}_k.$$

Algorithm 50 *Konstruktion eines Orthonormalsystems.*

for $k = 1, 2, ..., m$		
	$p_k \leftarrow g_k - g_0$;	
$m' \leftarrow m$;		
$w_1 \leftarrow \frac{v_1}{\|v_1\|}$;		
$\check{g}_1 \leftarrow \frac{p_1}{\|v_1\|}$;		
for $j = 2, 3, ...m$		
	$u_j \leftarrow v_j - \sum_{i=1}^{j-1} v_j^T w_i w_i$;	
	if $(\|u_j\| == 0)$	
	then	*schließe Punkt* x_j *aus*
		$-- m'$;
	else	$w_j \leftarrow \frac{u_j}{\|u_j\|}$;
		$\check{g}_j \leftarrow \left(p_j - \sum_{i=1}^{j-1} v_j^T w_i \check{g}_i\right) / \|u_j\|$;

Bemerkung 51 *Sind die Differenzenvektoren $v_1, ..., v_m$ linear abhängig, so ist die Anzahl $m\prime$ der auf beschriebene Art und Weise konstruierten orthogonalen Basisvektoren $w_1, ..., w_{m\prime}$, kleiner als m.*
In einem solchen Fall kann es unter Umständen geschehen, daß nicht alle vorgegebenen Punkte mit der Trajektorie erreicht werden.

Die lineare Mannigfaltigkeit $M_{x_0,x_1,...,x_m}$ wird mit Hilfe des gewonnenen Orthonormalsystems $w_1, ..., w_{m\prime}$ wie folgt beschrieben (vgl. Lemma 52). Es

sei:

$$M_{x_0,w_1,...,w_{m'}} := \left\{ x \in \mathbb{R}^n \middle| \quad x = x_0 + \sum_{k=1}^{m'} \gamma_k(x)\, w_k \right\}.$$

Lemma 52 *Die Mannigfaltigkeiten $M_{x_0,x_1,...,x_m}$ und $M_{x_0,w_1,...,w_{m'}}$ sind identisch:*

$$M_{x_0,w_1,...,w_{m'}} = M_{x_0,x_1,...,x_m}.$$

Beweis. Jeder Vektor w_k läßt sich als lineare Kombination der Vektoren $v_1, ..., v_m$ darstellen. Für jeden beliebigen Punkt $x \in M_{x_0,w_1,...,w_{m'}}$ gilt also:

$$x = x_0 + \sum_{k=1}^{m'} \gamma_k(x)\, w_k = x_0 + \sum_{k=1}^{m} \alpha_k(x)\, v_k = x_0 + \sum_{k=1}^{m} \alpha_k(x)\,(x_k - x_0).$$

Mit den Parametern

$$\mu_k(x) = \begin{cases} \alpha_k, & k = 1, ..., m \\ 1 - \sum_{k=1}^{m} \alpha_k, & k = 0 \end{cases}$$

folgt dann direkt:

$$x = \sum_{k=0}^{m} \mu_k(x)\, x_k \wedge \sum_{k=0}^{m} \mu_k = 1.$$

Da die Vektoren $w_1, ..., w_{m'}$ ebenfalls eine Basis für das Vektorensystem $v_1, ..., v_m$ bilden, läßt sich auch jeder Vektor v_k als Lineare Kombination der Vektoren $w_1, ..., w_{m'}$ darstellen. Für jeden beliebigen Punkt $x \in M_{x_0,x_1,...,x_m}$ läßt sich die Zugehörigkeit x zu $M_{x_0,w_1,...,w_{m'}}$ auf dem gleichen Weg zeigen. ■

Die Orthogonalprojektion $P(x)$ (vgl. 1.40) kann mit dem Ausgangspunkt x_0 und den Basis-Vektoren $w_1, ..., w_{m'}$ einfacher dargestellt werden:

$$P(x) = x_0 + \sum_{k=1}^{m'} \gamma_k(x)\, w_k.$$

Die Koeffizienten $\gamma_0(x), ..., \gamma_{m'}(x)$ können auf folgende Weise direkt berechnet werden:

$$\gamma_k(x) = (x - x_0)^T w_k.$$

Die Lösung eines linearen Gleichungssystems ist hier nicht mehr erforderlich!

Der neue Ansatz führt zum folgenden Richtungsfeld $\check{r} : \mathbb{R}^n \to \mathbb{R}^n$

$$\check{r}(x) := g_0 + \sum_{k=1}^{m\prime} \gamma_k(x)\, \check{g}_k.$$

Für die Punkte $\check{x}_i := x_0 + w_i$, $i = 1, ..., m'$ kann leicht folgendes gezeigt werden:

$$\gamma_k(\check{x}_i) = w_i^T w_k = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}.$$

Die neuen Richtungsvektoren $\check{g}_i$, $i = 1, ..., m'$ entsprechen also den vom Richtungsfeld $\check{r}(x)$ in den Punkten $\check{x}_i$ vorgegebenen Richtungen $\check{r}(\check{x}_i)$. Das auf diese Weise konstruierte Richtungsfeld $\check{r}(x)$ stimmt mit dem ursprünglichen Richtungsfeld $\widehat{r}(x)$ überein.

Satz 53 *Die Richtungsfelder $\widehat{r}(x)$ und $\check{r}(x)$ sind identisch*

$$\widehat{r}(x) \equiv \check{r}(x).$$

Beweis. Es wurde bereits bewiesen (s. Lemma 52), daß die Mengen $M_{x_0,x_1,...,x_m}$ und $M_{x_0,w_1,...,w_{m\prime}}$ gleich sind.
Die Orthogonalprojektion eines beliebigen Punktes $x \in \mathbb{R}^n$ auf die jeweilige Menge bildet den Punkt x auf den gleichen Punkt $x^{\perp} \in M_{x_0,w_1,...,w_{m\prime}}$. Es reicht also die Übereinstimmung beider Richtungsfelder auf der Menge $M_{x_0,x_1,...,x_m}$ bzw. $M_{x_0,w_1,...,w_{m\prime}}$ zu zeigen.
Sei ein beliebiger Punkt $x \in M_{x_0,w_1,...,w_{m\prime}}$ gegeben

$$x = x_0 + \sum_{k=1}^{m\prime} \gamma_k(x)\, w_k = x_0 + \sum_{k=1}^{m} \alpha_k(x)\, v_k = x_0 + \sum_{k=1}^{m} \alpha_k(x)\,(x_k - x_0).$$

Da die Vektoren $g_1, ..., g_{m\prime}$ mit Hilfe der gleichen Transformationen aus $p_1, ..., p_m$ entstanden sind wie $w_1, ..., w_{m\prime}$ aus $v_1, ..., v_m$, gilt dann auch entsprechend:

$$\sum_{k=1}^{m\prime} \gamma_k(x)\, g_k = \sum_{k=1}^{m} \alpha_k(x)\, g_k = x_0 + \sum_{k=1}^{m} \alpha_k(x)\,(g_k - g_0).$$

Durch den Richtungsfeld $\widehat{r}(x)$ wird in x folgende Richtung gesetzt:

$$\check{r}(x) := g_0 + \sum_{k=1}^{m\prime} \gamma_k(x)\, g_k = g_0 + \sum_{k=1}^{m} \alpha_k(x)\,(g_k - g_0).$$

Mit den Parametern

$$\mu_k(x) = \begin{cases} \alpha_k, & k = 1, ..., m \\ 1 - \sum_{k=1}^{m} \alpha_k, & k = 0 \end{cases}$$

folgt dann direkt

$$\widehat{r}(x) = \sum_{k=0}^{m} \mu_k(x)\, g_k.$$

■

Beispiel 54 *Für die zweidimensionale Zielfunktion* $f_a : \mathbb{R}^2 \to \mathbb{R}$ *und drei Punkte* $z_0, z_1, z_2 \in \mathbb{R}^n$ *aus dem Beispiel 48*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y,$$

$$\begin{aligned} z_0 &= (-2,-1)^T \\ z_1 &= (1,-2)^T \\ z_2 &= (0,1)^T, \end{aligned}$$

sind die Vektoren v_1, v_2 *und die orthogonalen Basisvektoren* w_1 *und* w_2 *gegeben mit:*

$$\begin{array}{ll} v_1 = (3,-1)^T & v_2 = (2,2)^T \\ w_1 = \frac{\sqrt{10}}{10}(3,-1)^T & w_2 = \frac{\sqrt{10}}{10}(1,3)^T. \end{array}$$

Die Koeffizienten $\gamma_i(x)$ *und die Richtungsvektoren* $\check{g}_i$ *können dann leicht berechnet werden:*

$$\begin{array}{ll} \gamma_1(x) = \frac{\sqrt{10}}{10}(5 + 3x - y) & \check{g}_1 = -\frac{\sqrt{10}}{10}(3,3)^T \\ \gamma_2(x) = \frac{\sqrt{10}}{10}(5 + x + 3y) & \check{g}_2 = -\frac{\sqrt{10}}{10}(3.5,-1.5)^T. \end{array}$$

Für das Richtungsfeld $\check{r}(x)$ *ergibt sich:*

$$\begin{aligned} \check{r}(x) &= (3,0)^T - \frac{1}{10}(5 + 3x - y)(3,3)^T - \frac{1}{10}(5 + x + 3y)(3.5,-1.5)^T \\ &= \frac{1}{4}\begin{pmatrix} -1 - 5x - 3y \\ -3 - 3x + 3y \end{pmatrix}. \end{aligned}$$

Wie erwartet, stimmt das Richtungsfeld $\check{r}(x)$ *mit dem Richtungsfeld* $\hat{r}(x)$ *aus dem Beispiel 48 überein.*

Konstruktion des Gleichungssystem

Die Konstruktion des Gleichungssystems (1.34) für die Richtungsfeld-Trajektorien mit mehreren Startpunkten kann etwas vereinfacht werden.
Bei der direkten Konstruktion der Matrix $\hat{G}(x)$ wird sie durch konvexe Kombination der konstanten Matrizen $\hat{G}(x_k)$, $k = 0, ..., m$ ersetzt. Das Richtungsfeld $\widehat{r}(x)$ und damit auch die Trajektorie $T_{\widehat{r}}(f)$ bleiben unverändert.
Da eine konvexe Kombination der orthogonalen Matrizen nicht notwendigerweise orthogonal bleibt, ist der gleiche Weg bei der Konstruktion der Matrix $\tilde{G}(x)$ nicht möglich. In diesem Fall kann lediglich der Vektor $w(x)$ durch normierte konvexe Kombination der Vektoren $w(x_k)$, $k = 0, ..., m$ ersetzt werden.

Direkte Konstruktion Die in den vorgegebenen Punkten berechneten Gradienten $\nabla f(x_k)$ werden hier als lineare Kombination der Basisvektoren $\{q_1, ..., q_n\}$ dargestellt:

$$\nabla f(x_k) = \sum_{i=1}^{n} q_i^T \nabla f(x_k)\, q_i, \qquad k = 0, ..., m.$$

Die Richtungsfunktion $\widehat{r}$ wird bzgl. der Basis Q in n Koordinatenfunktionen $\widehat{r}_i : \mathbb{R}^n \to \mathbb{R}^1$ zerlegt:

$$\widehat{r}(x) = \sum_{k=0}^{m} \widehat{r}_i(x)\, q_i.$$

Die Funktionen $\widehat{r}_i(x)$ können wie folgt bestimmt werden:

$$\begin{aligned} \widehat{r}(x) &= \sum_{k=0}^{m} \mu_k(x) \nabla f(x_k) = \sum_{k=0}^{m} \mu_k(x) \sum_{i=1}^{n} q_i^T \nabla f(x_k)\, q_i \\ &= \sum_{i=1}^{n} \left(\sum_{k=0}^{m} \mu_k(x)\, q_i^T \nabla f(x_k) \right) q_i. \end{aligned}$$

Damit ergibt sich:

$$\widehat{r}_i(x) = \sum_{k=0}^{m} \mu_k(x)\, q_i^T \nabla f(x_k).$$

Das nichtlineare Gleichungssystem (1.36) wird umgeschrieben in:

$$\hat{G}(x)^T \nabla f(x) = \left(\sum_{k=0}^{m} \mu_k(x)\, \hat{G}_k \right)^T \nabla f(x) = 0. \tag{1.42}$$

Das Gleichungssystem (1.42) wird hier durch konvexe Kombination der konstanten Matrizen $\hat{G}_k$ bestimmt, die den vorgegebenen Punkten x_k zugeordneten werden können

$$\hat{G}_k^T := \begin{pmatrix} q_1^T \nabla f(x_k)\, q_n^T - q_n^T \nabla f(x_k)\, q_1^T \\ \cdots \\ q_{n-1}^T \nabla f(x_k)\, q_n^T - q_n^T \nabla f(x_k)\, q_{n-1}^T \end{pmatrix}.$$

Lemma 55 *Die Matrix $\hat{G}(x)$ ist überall orthogonal zu $\hat{r}(x)$:*

$$\hat{G}(x)^T \hat{r}(x) = 0.$$

Beweis. Sei $\hat{g}_{kl}$ die l-te Spalte der Matrix $\hat{G}_k$ und $\hat{g}_l(x)$ die l-te Spalte der Matrix $\hat{G}(x)$, dann gilt:

$$\begin{aligned} g_{kl}^T \widehat{r}(x) &= \left(q_l^T \nabla f(x_k)\, q_n^T - q_n^T \nabla f(x_k)\, q_l^T\right) \sum_{i=1}^{n} \widehat{r}_i(x)\, q_i \\ &= q_l^T \nabla f(x_k)\, \widehat{r}_n(x) - q_n^T \nabla f(x_k)\, \widehat{r}_l(x) \end{aligned}$$

und es folgt

$$\begin{aligned} \hat{g}_l^T \widehat{r}(x) &= \sum_{j=0}^{m} \mu_j(x) \left(q_l^T \nabla f(x_k)\, \widehat{r}_n(x) - q_n^T \nabla f(x_k)\, \widehat{r}_l(x)\right) \\ &= \widehat{r}_l(x)\, \widehat{r}_n(x) - \widehat{r}_n(x)\, \widehat{r}_l(x) = 0 \end{aligned}$$

■

Bemerkung 56 *Durch geeignete Wahl der Basisvektoren $\{q_1, ..., q_n\}$ kann gewährleistet werden, daß die Bedingung*

$$\forall_{k=0,1,...,n} q_n^T \nabla f(x_k) \neq 0$$

erfüllt ist und damit die Matrizen $\hat{G}_k$ vollen Rang haben (s. 27). Dies gilt aber nicht für jede konvexe Kombination der Matrizen $\hat{G}_k$ und somit auch nicht überall für die Matrix $\hat{G}(x)$.

Beispiel 57 *Für die zweidimensionale Zielfunktion $f_a : \mathbb{R}^2 \to \mathbb{R}$ aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y,$$

und zwei vorgegebene Punkte $z_0 = (-2,-1)^T$ *und* $z_1 = (1,-2)^T$ *wurde in Beispiel 47 folgendes Richtungsfeld aufgestellt:*

$$\widehat{r}(x) = \frac{3}{10}\begin{pmatrix} 5-3x+y \\ -5-3x+y \end{pmatrix}.$$

Die entsprechende Trajektorie $T_{\widehat{r}}(f)$ *läßt sich hierbei mittels einer variablen* 1×2 *Matrix* $\hat{G}(x)$

$$\hat{G}(x) = \frac{3}{10}(5+3x-y, 5-3x+y),$$

wie folgt beschreiben:

$$T(f, z_0, z_1) = \left\{ z \in \mathbb{R}^2 | \quad \hat{G}^T(x) \nabla f(x) = 0 \right\}.$$

Für die in der Abbildung 1.7 dargestellte Trajektorie $T(f, z_0, z_1)$ *gilt also:*

$$(5+3x-y)\left(x^2-1\right) + (5-3x+y)\left(1-y^2\right) = 0$$

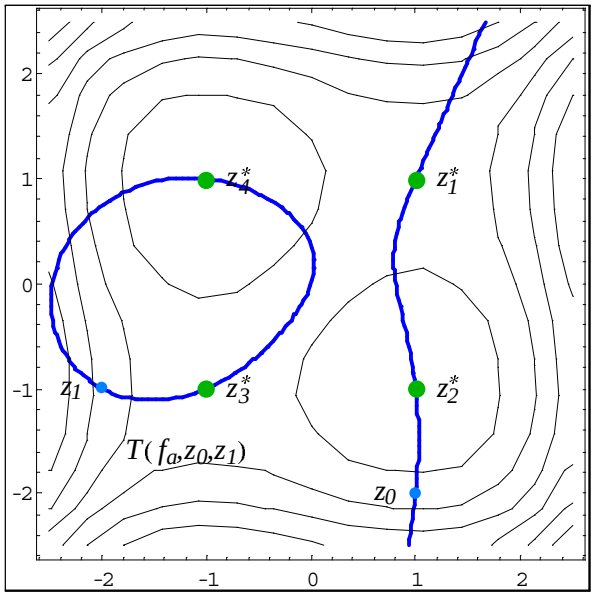

Abbildung 1.7: Richtungsfeld-Trajektorie mit zwei vorgegebenen Punkten z_0 und z_1 im Konturplot der Zielfunktion f_a.

Beispiel 58 *Für die zweidimensionale Zielfunktion* $f_a : \mathbb{R}^2 \to \mathbb{R}$ *aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y,$$

und drei vorgegebene Punkte $z_0 = (-2,-1)^T$, $z_1 = (1,-2)^T$ *und* $z_2 = (0,1)^T$ *wurde in Beispiel 48 folgendes Richtungsfeld aufgestellt:*

$$\widehat{r}\,(x) = \frac{1}{4}\begin{pmatrix} -1 - 5x - 3y \\ -3 - 3x + 3y \end{pmatrix}.$$

Für die in der Abbildung dargestellte Trajektorie $T\,(f, z_0, z_1, z_0)$ *gilt also:*

$$(3 + 3x - 3y)\left(x^2 - 1\right) - (1 + 5x + 3y)\left(1 - y^2\right) = 0$$

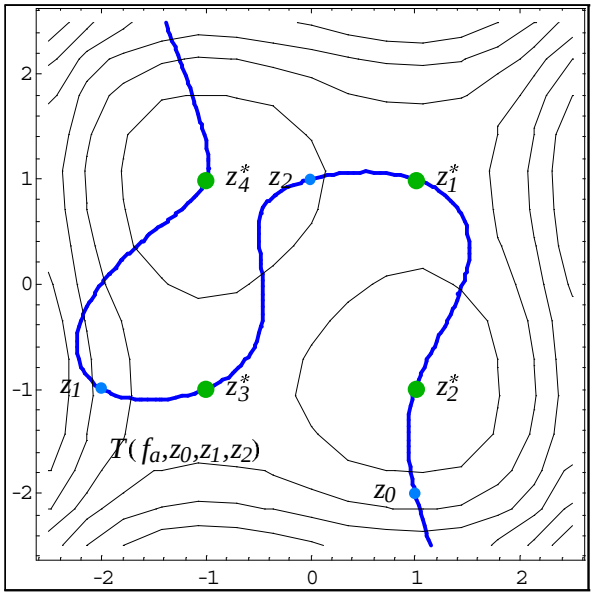

Abbildung 1.8: Richtungsfeld-Trajektorie mit drei vorgegebenen Punkten z_0, z_1 und z_2 im Konturplot der Zielfunktion f_a.

Konstruktion mittels Spiegelung Für die vorgegebenen Punkte x_k und normierten Richtungen $\tilde{g}_k$ werden die konstanten Vektoren w_k konstruiert:

$$w_k := \frac{\tilde{g}_k - q_n}{\|\tilde{g}_k - q_n\|}.$$

Mit dem variablen Vektor

$$w(x) := \sum_{k=0}^{m} \mu_k(x)\, w_k$$

wird dann die Spiegelung $\tilde{S}_r(x,p)$ definiert durch:

$$\tilde{S}_r(x,p) := I_n - 2w(x)\, w^T(x).$$

Die folgende Eigenschaft der Spiegelung $\tilde{S}_r(x,p)$

$$\tilde{S}(x_k, q_n) = \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|},$$

ist trivialerweise erfüllt. Dies erlaubt die Konstruktion der orthogonalen variablen Matrix $\tilde{G}(x)$ durch die Spiegelung der Basisvektoren $q_1, ..., q_{n-1}$

$$\tilde{G}(x) := \tilde{S}_r(x, q_1), ..., \tilde{S}_r(x, q_{n-1})).$$

Das Richtungsfeld $\hat{r}(x)$ ist zwar nicht mehr orthogonal zur konstruierten Matrix $\tilde{G}(x)$. Ein neues Richtungsfeld $\tilde{r}(x)$ kann aber wie folgt konstruiert werden:

$$\tilde{r}(x) = \tilde{S}_r(x, qn).$$

Das neue Richtungsfeld $\tilde{r}(x)$ stimmt in den vorgegebenen Punkten x_k, $k = 0, ..., m$ mit dem ursprünglichen Feld $\hat{r}(x)$ überein. Die neue Trajektorie $T_{\tilde{r}}(f)$ enthält somit die Punkte x_k.

Beispiel 59 *Für die zweidimensionale Zielfunktion $f_a : \mathbb{R}^2 \to \mathbb{R}$ und drei vorgegebene Punkte z_0, z_1 und z_2 aus dem Beispiel 48 wurde das nach dem Prinzip der variablen Spiegelung aufgestellte Richtungsfeld $\tilde{r}(x)$ in Abbildung 1.9 dargestellt.*
Die entsprechende Trajektorie $T(f, z_0, z_1, z_2)$ ist in der Abbildung 1.10 skizziert. Obwohl die konstruierte Trajektorie $T(f, z_0, z_1, z_2)$ die Punkte z_0, z_1, z_2 enthält, stimmt sie nicht mit der Trajektorie aus Beispiel 58 überein (vgl. Abbildung 1.8).

1.4.4 Richtungsfeld-Trajektorie für restringierte Probleme

Bei einem restringierten Problem müssen die Randpunkte des Zulässigkeitsbereiches besonderes berücksichtigt werden. Die notwendige Optimalitätsbedingung 1. Ordnung für restringierte Probleme (s. Abschnitt 1.1) kann

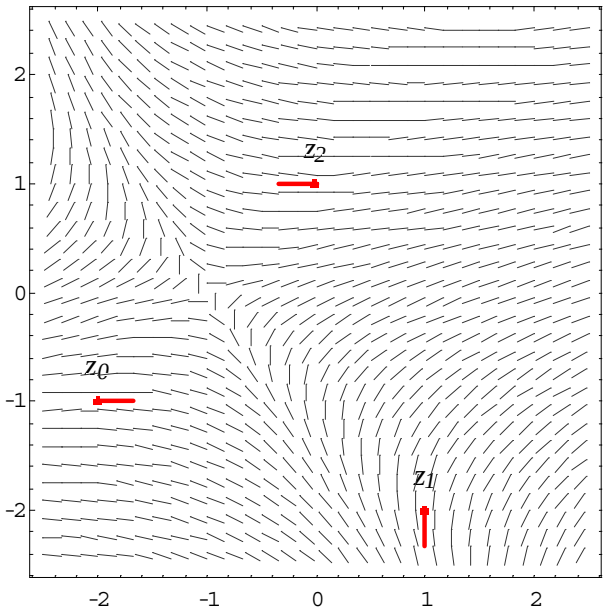

Abbildung 1.9: Richtungsfeld variablen Spiegelung mit drei vorgegebenen Punkten z_0, z_1 und z_2 und die Zielfunktion f_a.

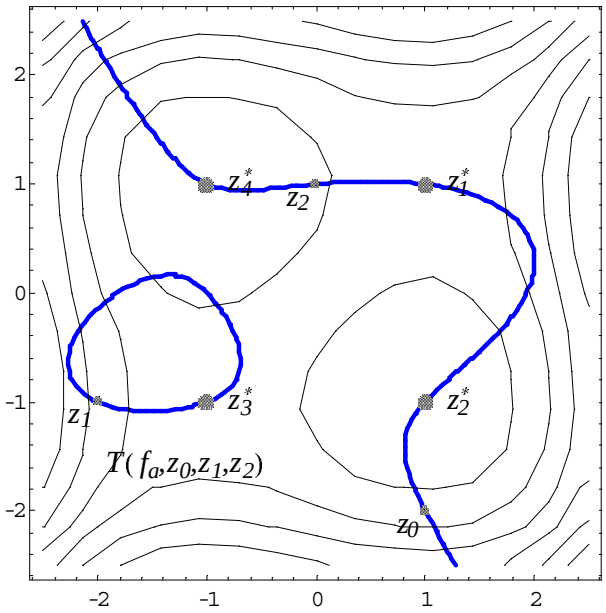

Abbildung 1.10: Richtungsfeld-Trajektorie variablen Spiegelung mit drei vorgegebenen Punkten z_0, z_1 und z_2 im Konturplot der Zielfunktion f_a.

als eine Richtungsbedingung für den Gradienten $\nabla f(x)$ der Zielfunktion f aufgefaßt werden:

$$\exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda \nabla c(x).$$

In diesem Abschnitt wird eine Trajektorie definiert, die außer den kritischen Punkte der Zielfunktion auch die Randpunkte des Zulässigkeitsbereiches enthält, die notwendige Optimalitätsbedingung erfüllen.

Richtungsfeld

Um in einer Trajektorie $T_r(f)$ die aus Sicht der Optimierungsaufgabe interessanten Randpunkte zu erfassen, wird ein trajektorieninduzierendes Richtungsfeld $r_c(x)$ wie folgt definiert:

Definition 60 *Seien eine restringierte Optimierungsaufgabe 2 mit der Zielfunktion $f(x)$ und der Restriktion $c(x)$ sowie ein Anfangspunkt $x_0 \in \mathbb{R}^n$ gegeben.*
Ein Richtungsfeld $r_c(x_0)$ heißt für die gestellte Optimierungsaufgabe ***trajektorieninduzierend*** *falls folgendes gilt:*

1. *Die Richtung $r_c(x_0)$ stimmt mit der Richtung des Gradienten $\nabla f(x_0)$ der Funktion f im Punkt x_0 überein:*

$$\exists_{\lambda \in \mathbb{R}} \nabla f(x_0) = \lambda r_c(x_0).$$

2. *Die Richtung $r_c(x_c)$ stimmt mit der Richtung des Gradienten $\nabla c(x_c)$ der Restriktion c für alle Randpunkte x_c überein:*

$$\forall_{x_c \in \partial M} \exists_{\lambda \in \mathbb{R}} \nabla c(x_c) = \lambda r_c(x_c).$$

Das Richtungsfeld $r_c(x)$ kann als nichtlineare Kombination des Gradienten $\nabla f(x_0)$ der Zielfunktion f in dem vorgegebenen Ausgangspunkt $x_0 \in M \setminus \partial M$ und des Gradienten $\nabla c(x)$ der Randfunktion $c(x)$ in Punkt x konstruiert werden:

$$r_c(x) := c(x) \nabla f(x_0) + (c(x_0) - c(x)) \nabla c(x). \tag{1.43}$$

Die Koeffizienten $c(x)$ und $(c(x_0) - c(x))$ sind hierbei vom Wert der Randfunktion $c(x)$ abhängig.
Es sind auch andere Ansätze möglich, beispielsweise:

$$r_{c,p}(x) := c(x)^p \nabla f(x_0) + (c(x_0)^p - c(x)^p) \nabla c(x), \qquad p \in \mathbb{R}_+. \tag{1.44}$$

Im weiteren wird der Einfachheit halber stets die Vorschrift 1.43 eingesetzt.

Bemerkung 61 *Der Ausgangspunkt x_0 darf nicht auf dem Rand liegen. Gilt nämlich $x_0 \in \partial M$ und damit $c(x_0) = 0$, so ist das Richtungsfeld $r_c(x)$ auf dem gesamten Rand ∂M gleich Null.*

Probleme mit mehreren Restriktionen

Für die Probleme mit mehreren aber endlich vielen Restriktionen $c_1(x), ..., c_m(x)$ und den entsprechenden Zullässigkeitsbereich:

$$M = \left\{ x \in \mathbb{R}^n | \quad \forall_{j \in \{1,...,m\}} \quad c_j(x) \geq 0 \right\},$$

kann leicht eine Funktion $c(x)$ konstruiert werden, deren Gradient $\nabla c(x)$ in den regulären Randpunkten $x_c \in \partial M$ orthogonal zum Rand ∂M der Menge M liegt. Dies gilt dann, wenn die Funktion $c(x)$ in der Umgebung von x_c auf dem Rand ∂M konstant ist. Für das Produkt der Restriktionsfunktionen $c_1(x), ..., c_m(x)$ ist diese Bedingung beispielsweise erfüllt. Auf der Grundlage einer solchen Funktion kann dann ein geeignetes Richtungsfeld $r_c(x)$ anhand der Vorschrift (1.43) bzw. (1.44) konstruiert werden.

Bemerkung 62 *Die Nullmenge der Funktion $c(x)$ muß hierbei nicht mit dem Rand des Zullässigkeitsbereiches übereinstimmen sondern lediglich die regulären Randpunkte x_c enthalten. Die Zullässigkeit eines Kandidates für die globale Lösung des Optimierungsproblems muß in jedem Fall anhand der Restriktionen $c_j(x) \geq 0$, $j \in \{1, ..., m\}$ geprüft werden. Die Bedingung $c(x) \geq 0$ ist im algemeinen Fall weder ausreichend noch notwendig!*

Trajektorie für restrinigertes Optimierungsproblem

Die entsprechend durch das Richtungsfeld $r_c(x)$ induzierte Trajektorie $T_{r_c}(f)$ enthält den Startpunkt x_0 und alle Punkte aus dem Zulässigkeitsbereich, die die notwendige Optimalitätsbedingung der restringierten Aufgabe erfüllen.

Satz 63 *Sei $r_c(x)$ das für das restringierte Optimierungsproblem 2 konstruierte Richtungsfeld. Für die Trajektorie $T_{r_c}(f)$ gilt dann:*

(i) Die Trajektorie enthält den Ausgangspunkt x_0.

(ii) Die Trajektorie enthält alle Randpunkte x_c^*, für die die Optimalitätsbedingung (1.4) erfüllt ist:

(iii) Die Trajektorie enthält alle kritischen Punkte:

$$Krit(f) \subset T_{r_c}(f).$$

Beweis.

(i) Für $x = x_0$ folgt:

$$r_c(x_0) = c(x_0)\nabla f(x_0) + (c(x_0) - c(x_0))\nabla c(x_0) = c(x_0)\nabla f(x_0)$$

und die Trajektorienzugehörigkeitsbedingung (1.31) ist trivialerweise erfüllt.

(ii) Für jeden Randpunkt x_c gilt (wegen $c(x_c) = 0$):

$$r_c(x_c) := c(x_c)\nabla f(x_0) + (c(x_0) - c(x_c))\nabla c(x_c) = c(x_0)\nabla c(x_c).$$

Für die gesuchten Randpunkte x_c^* gilt außerdem (vgl. Bedingung 1.4)

$$\nabla f(x_c^*) - \lambda \nabla c(x_c^*) = 0,$$

so daß mit

$$c(x_0)\nabla f(x_c^*) = c(x_0)\lambda \nabla c(x_c^*) = \lambda r_c(x_c^*)$$

die Bedingung (1.31) erfüllt ist.

(iii) Die kritischen Punkte der Zielfunktion gehören nach dem Lemma 40 zur durch $r_c(x)$ induzierten Trajektorie.

■

Die praktische Anwendung der Richtungsfeld-Trajektorien für restringierte Probleme wird hier am Beispiel zweier einfacher Randfunktionen vorgestellt ist aber im Prinzip auf andere Optimierungsprobleme mit einer beliebigen Restriktion $c(x)$ übertragbar (vgl. Abschnitt 4.5).

Beispiel 64 *(n-dimensionale Kugel)*
Mit der Randfunktion

$$c_{K(s,\rho)}(x) := \rho^2 - (x - s)^2$$

läßt sich eine kugelförmige Zulässigkeitsmenge mit dem Mittelpunkt s und dem Radius ρ beschreiben. Das entsprechende Richtungsfeld kann wie folgt berechnet werden:

$$r_c(x) := \left(\rho^2 - (x - s)^2\right)\nabla f\left(x^0\right) - 2\left((x - s)^2 - (x_0 - s)^2\right)(x - s).$$

Für die zweidimensionale Zielfunktion f_a aus dem Beispiel 20, den Ausgangspunkt $z_0 = (0,0)^T$ und die Randfunktion

$$c_{K((0,0),2)} = 4 - x^2 - y^2, \tag{1.45}$$

wurde das entsprechende Richtungsfeld

$$r_c(x) := \left(4 - x^2 - y^2\right) \begin{pmatrix} -1 \\ 1 \end{pmatrix} - 2\left(x^2 + y^2\right) \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.46}$$

in der Abbildung 1.11 dargestellt. Für das restringierte Problem gibt es außer

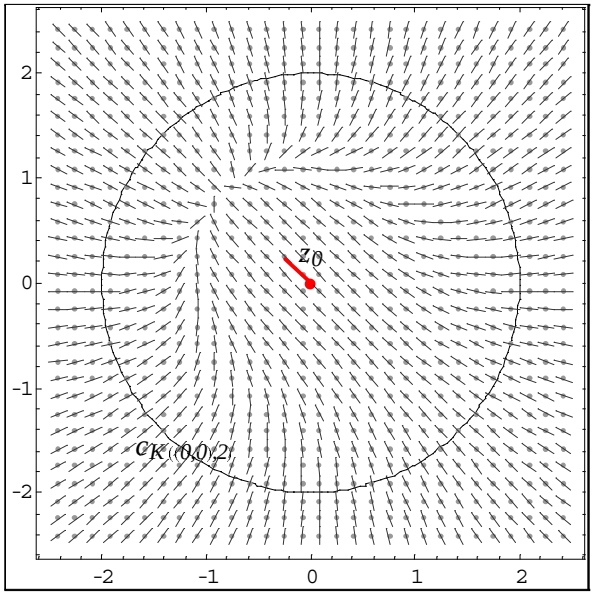

Abbildung 1.11: Richtungsfeld für ein restringiertes Optimierungsproblem mit kreisförmigem Zulässigkeitsbereich

den vier kritischen Punkten

$$z_i^* = (\pm 1, \pm 1)^T, \quad i = 1, ..., 4,$$

noch sechs weitere Randpunkte $z_i^*, \quad i = 5, ..., 10,$

$$\begin{aligned} z_{6,9}^* &= \left(\mp\sqrt{2}, \pm\sqrt{2}\right)^T \\ z_{5,10}^* &= \left(\pm\sqrt{2-\sqrt{3}}, \pm\sqrt{2+\sqrt{3}}\right)^T \\ z_{7,8}^* &= \left(\pm\sqrt{2+\sqrt{3}}, \pm\sqrt{2-\sqrt{3}}\right)^T, \end{aligned}$$

die als Kandidaten für das globale Minimum bzw. das globale Maximum in Betracht kommen. Die Kandidatenpunkte können durch Lösung des folgenden Gleichungssystems (vgl. Lagrange-Charakterisierung - Satz 8) berechnet werden:

$$\begin{aligned} y\left(1-x^2\right)+x\left(1-y^2\right) &= 0 \\ 4-x^2-y^2 &= 0. \end{aligned}$$

Für die in der Abbildung 1.12 dargestellte, durch das Richtungsfeld $r_c(x)$ induzierte Trajektorie $T_{r_c}(f_a)$ gilt hier:

$$\left(4-x^2-y^2+2x\left(x^2+y^2\right)\right)\left(1-x^2\right)-\left(4-x^2-y^2-2y\left(x^2+y^2\right)\right)\left(1-y^2\right)=0.$$

Die Trajektorie $T_{r_c}(f_a)$ enthält alle möglichen Kandidatenpunkte z_i^*, $i = 1, ..., 10$. Der Vergleich der Funktionswerte in den gefundenen zulässigen

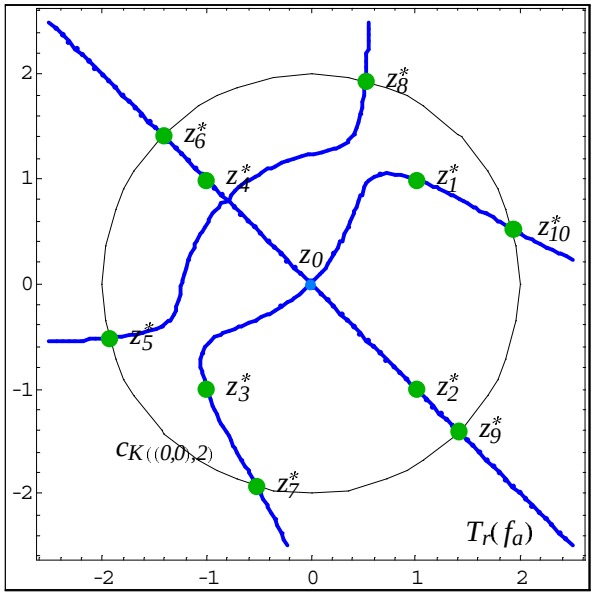

Abbildung 1.12: Richtungsfeld-Trajektorie für das restringierte Optimierungsproblem mit kreisformigem Zulässigkeitsbereich

Punkten z_i^*, führt zur folgenden Erkenntnis über die globalen Minima und Maxima für das aufgestellte restringierte Problem:

1. *Globales Minimum:*

$$z_2^*, \text{ mit dem Funktionswert } f_a(z_2^*) = -\frac{4}{3}.$$

2. Globales Maximum

$$z_4^* \quad \textit{mit dem Funktionswert } f_a\left(z_2^*\right) = \frac{4}{3}.$$

Beispiel 65 *Ein n-dimensionaler Simplex*
Der Rand eines n-dimensionaler Simplex in $\mathbb{R}^n$ *kann mit Hilfe von* $n+1$ *Hyperebenen*

$$\eta_{a_i,b_i} := \left\{ x \in \mathbb{R}^n | \quad a_i^T \left(x - b_i\right) = 0 \right\}$$

als Teilmenge der Nullmenge folgender Funktion (vgl. Bemerkung 62) bestimmt werden:

$$c_D\left(x\right) := \prod_{i=\mathbf{0}}^{n} a_i^T \left(x - b_i\right).$$

Die Vektoren a_i *sind dabei als normale (orthogonale) Vektoren der Hyperebenen* η_{a_i,b_i}, *die in das Innere des Simplex zeigen, zu wählen. Als Vektoren* b_i *können beliebige Punkte der Hyperebenen* η_{a_i,b_i} *gewählt werden. Die Randfunktion* $c_D\left(x\right)$ *ist ein n-dimensionaler Polynom* $n+1$*-ten Grades.*
Für einen Randpunkt x_c *des zulässigen Bereiches ist die Richtung* $r_c\left(x\right)$ *durch folgende Bedingung bestimmt:*

$$r\left(x\right) = \begin{cases} a_i, & x \in \eta_{a_i,b_i} \wedge x \notin \eta_{a_j,b_j}, \quad j \neq i \\ 0, & x \in H_{a_i} \wedge x \in \eta_{a_j,b_j}, \quad j \neq i \end{cases}.$$

Für die zweidimensionale Zielfunktion f_a *aus dem Beispiel 20, den Ausgangspunkt* $z_{\mathbf{0}} = \left(0,0\right)^T$ *und die durch Vektoren*

$$\begin{array}{ll} a_1 = \left(0,1\right)^T & b_1 = \left(0,-2\right)^T \\ a_2 = \left(2,-1\right)^T & b_2 = \left(-1,0\right)^T \\ a_3 = \left(-2,-1\right)^T & b_2 = \left(1,0\right)^T, \end{array}$$

induzierte Randfunktion

$$c_D\left(x\right) = \left(2x + y - 2\right)\left(2x - y + 2\right)\left(y + 2\right), \tag{1.47}$$

wurde das entsprechende Richtungsfeld $r_c\left(x\right)$ *in der Abbildung 1.11 dargestellt. Für das restringierte Problem gibt es außer den vier kritischen Punkten*

$$z_i^* = \left(\pm 1, \pm 1\right)^T, \quad i = 1, ..., 4,$$

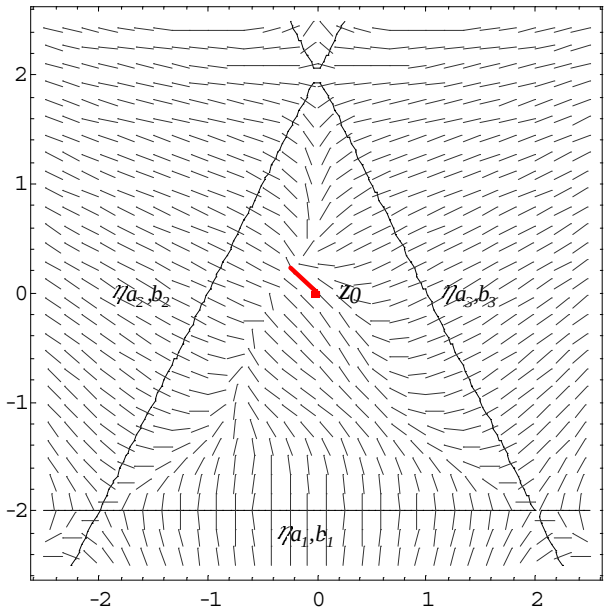

Abbildung 1.13: Richtungsfeld für ein Optimierungsproblem mit dreieckigem Zulässigkeitsbereich

noch sechs weitere Randpunkte $z_i^*, \quad i = 5, ..., 10$ *sowie drei Simplexecken* $z_i^*, \quad i = 11, ..., 13$, *die als Kandidaten für das globale Minimum bzw. das globale Maximum in Betracht kommen.*
Die Punkte $z_i^*, \quad i = 5, ..., 10$ *können, abhängig von dem Rand, auf dem sie liegen, durch Lösung eines der folgenden Gleichungssysteme (vgl. Lagrange-Charakterisierung - Satz 8) berechnet werden.*
Für die Randpunkte der Hyperebene η_{a_1,b_1} *gilt die Forderung:*

$$\begin{cases} 1 - x^2 = 0 \\ y + 2 = 0 \end{cases} . \tag{1.48}$$

Die Bedingung 1.48 führt zum folgenden Kandidatenpunkten:

$$\begin{aligned} z_5^* &= (-2, -1)^T \\ z_6^* &= (-2, 1)^T . \end{aligned}$$

Für die Randpunkte der Hyperebene η_{a_2,b_2} *gilt die Forderung:*

$$\begin{cases} -(1 - x^2) + 2(1 - y^2) = 0 \\ 2x - y + 2 = 0 \end{cases} . \tag{1.49}$$

Die Bedingung 1.49 führt zu folgenden Kandidatenpunkten:

$$z^*_{7,9} = \left(\frac{1}{7}\left(-8 \pm \sqrt{15}\right), \frac{2}{7}\left(-1 \pm \sqrt{15}\right) \right)^T .$$

Für die Randpunkte der Hyperebene η_{a_3,b_3} gilt die Forderung:

$$\begin{cases} -(1-x^2) - 2(1-y^2) = 0 \\ 2 - 2x - y = 0 \end{cases} . \tag{1.50}$$

Die Bedingung 1.50 führt zu folgenden Kandidatenpunkten:

$$z^*_{8,10} = \left(\frac{1}{9}\left(8 \mp \sqrt{19}\right), \frac{2}{9}\left(1 \pm \sqrt{19}\right) \right)^T .$$

Die Eckpunkte $z^*_i, \quad i = 11, ..., 13$ sind gegeben durch

$$\begin{aligned} z^*_{11} &= (0,2)^T , \\ z^*_{12} &= (2,-2)^T , \\ z^*_{13} &= (-2,-2)^T . \end{aligned}$$

Die durch das Richtungsfeld $r_c(x)$ induzierte Trajektorie $T_{r_c}(f_a)$ enthält alle möglichen Kandidatenpunkte (vgl. Abbildung 1.14).

Der Vergleich der Funktionswerte in den gefundenen zulässigen Punkten $z^*_i, \quad i = 2, 3, 5, ..., 13$ (z^*_1 und z^*_4 sind nicht zulässig) führt zu folgenden Erkenntnis über die globalen Minima und Maxima für das aufgestellte restringierte Problem:

1. *Globales Minimum:*

$$z^*_2, \text{ mit dem Funktionswert } f_a(z^*_2) = -\frac{4}{3}.$$

2. *Globale Maxima*

$$z^*_5 \text{ und } z^*_{12} \text{ mit dem Funktionswert } f_a(z^*_2) = \frac{4}{3}.$$

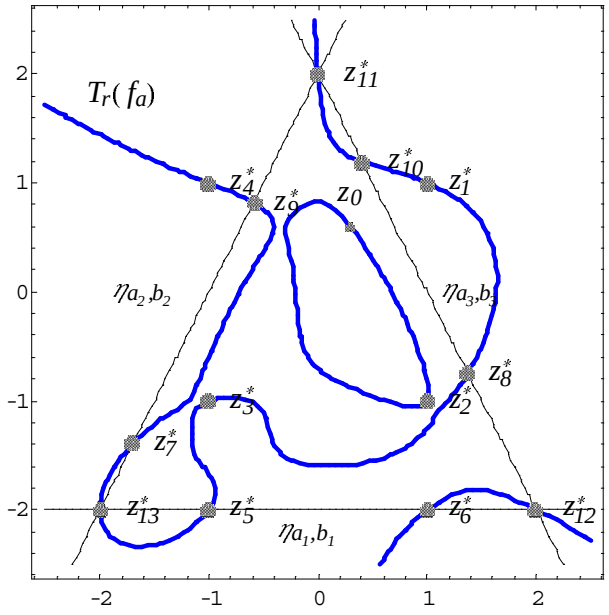

Abbildung 1.14: Richtungsfeld-Trajektorie für das restringierte Optimierungsproblem mit dreieckigem Zulässigkeitsbereich

Kapitel 2

Suche nach den Trajektorienkomponenten

2.1 Rekursive Konstruktion des Verfahrens

Wie bereits erwähnt wurde, sind die vorgestellten Trajektorien nicht immer zusammenhängend und bestehen deshalb möglicherweise aus mehreren Komponenten. Eine mögliche Strategie, Verbindungstrajektorien zwischen den Trajektorienkomponenten zu konstruieren, wurde in [15] vorgestellt. Die topologische und geometrische Eigenschaften der Verbindungstrajektorien wurden in [17] theoretisch untersucht.
Die Grundidee dieser Methode ist, die speziellen Unterprobleme aufzustellen, deren Dimension um eins niedriger ist als die Dimension des Hauptproblems. Für die Unterprobleme werden Hilfstrajektorien konstruiert, die dann die ursprünglichen Trajektorienkomponenten miteinander verbinden.
Da die Verbindungstrajektorien selber auch aus mehreren Komponenten bestehen können, wird ein Netz aus den rekursiv konstruierten Trajektorien gebaut. Auf diese Weise wird der Versuch unternommen, alle Komponenten der Haupttrajektorie und damit auch alle im Sinn der Optimierungsaufgabe interessanten Punkte zu erreichen.

2.1.1 Newton-Blätter

Mit $Q = (q_1, ..., q_n)$ wird eine beliebige orthogonale $n \times n$ Matrix $\left(Q^T Q = I\right)$ eingeführt und mit $Q_k := (q_1, ..., q_l)$ die auf l-Spalten reduzierte $n \times l$ Teilmatrix der Matrix Q bezeichnet. Ausgehend vom Gleichungssystem (1.17) für klassische Newton-Trajektorie werden die Eigenschaften der Lösungsmenge

des folgenden Gleichungssystems

$$Q_l^T \nabla f(x) = 0 \tag{2.1}$$

untersucht.

Definition 66 *Sei $f \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ und Q eine orthogonale $n \times n$ Matrix. Die Lösungsmenge des Gleichungssystems (2.1) mit $l = n-k$, $(k = 0, ..., n-1)$*

$$\mathcal{F}_Q(f) := \left\{ x \in \mathbb{R}^n \middle| \quad Q_{n-k}^T \nabla f(x) = 0 \right\}$$

*wird als k-**Newton-Blatt der Zielfunktion f bzgl. der Matrix Q** bezeichnet.*

Mit der QR-Zerlegung einer regulären $n \times n$ Matrix $A \in \mathbb{R}^{n \times n}$

$$A = QR,$$

kann die Definition 66 auf reguläre Matrizen erweitert werden. Die orthogonale Matrix Q und obere Dreiecksmatrix R (mit positiven Diagonaleinträgen) sind hierbei eindeutig bestimmt [27].

Definition 67 *Sei $f \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ und $A = QR$ eine reguläre Matrix. Die Lösungsmenge des Gleichungssystems (2.1) mit $l = n-k$, $(k = 0, ..., n-1)$*

$$\mathcal{F}_A(f, k) := \left\{ x \in \mathbb{R}^n \middle| \quad Q_{n-k}^T \nabla f(x) = 0 \right\}$$

*wird als k-**Newton-Blatt der Zielfunktion f bzgl. der Matrix A** $(k = 0, ..., n-1)$ bezeichnet.*

Es können dann folgende Beziehungen nachgewiesen werden.

Korollar 68 *Die Menge der kritischen Punkte der Funktion f ist ein 0-Newton-Blatt bzgl. jeder regulärer Matrix $A = QR$:*

$$\mathcal{F}_A(f, 0) = Krit(f).$$

Beweis. Die Bedingung $Q^T \nabla f(x) = 0$ ist genau dann erfüllt, wenn $\nabla f(x) = 0$. ■

Korollar 69 *Seien eine $n \times n-1$ Matrix G vollen Ranges ($Rang(G) = n-1$) und ein Vektor $0 \neq g \in \ker G^T$ gegeben.*
Die durch die Funktion $H(x) := G^T \nabla f(x)$ induzierte klassische Newton-Trajektorie $T_H(f)$ ist ein 1-Newton-Blatt der Funktion f bzgl. $A = (G, g)$:

$$\mathcal{F}_A(f, 1) = T_H(f).$$

Beweis. Mit der QR-Zerlegung der Matrix A kann die Matrix G wie folgt dargestellt werden:

$$G = QR_{n-1}.$$

Mit R_{n-1} wird hier $n \times n-1$ Matrix bezeichnet, die durch Weglassen der letzten Spalte aus der Matrix R entsteht. Da die letzte Zeile der reduzierten Matrix R_{n-1} aus lauter *Nullen* besteht, kann auch diese weggelassen werden $(R_{n-1} \rightarrow R_{n-1,n-1})$. Für die Matrix G gilt dann:

$$G = Q_{n-1}R_{n-1,n-1}.$$

Mit der Matrix R ist auch die reduzierte Matrix $R_{n-1,n-1}$ regulär, so daß folgt:

$$Q_{n-1}^T \nabla f(x) = 0 \Leftrightarrow G^T \nabla f(x) = 0.$$

■

Beispiel 70 *Für eine zweidimensionale Zielfunktion* $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ *und einen Vektor* $g := \nabla f(z_0)$ *(vgl. Beispiel 29) ist die die klassische Newton-Trajektorie* $T(f, z_0)$ *induzierende Matrix* A *gegeben durch:*

$$A = \begin{pmatrix} \frac{\partial f_a(z_0^1)}{\partial y} & \frac{\partial f_a(z_0^1)}{\partial x} \\ -\frac{\partial f_a(z_0^1)}{\partial x} & \frac{\partial f_a(z_0^1)}{\partial y} \end{pmatrix}.$$

Mit der orthogonalen Matrix Q *und der diagonalen Matrix* R *:*

$$Q = \frac{1}{\|\nabla f_a(z_0)\|} A \qquad \textit{und} \qquad R = \|\nabla f_a(z_0)\| \, I_2$$

ist dann die QR-Zerlegung der Matrix A *gegeben.*
Für den 1-Newton-Blatt $\mathcal{F}_A(f, 1)$ *der Funktion* f *bzgl. der Matrix* A *ergibt sich dann (vgl. Formel 1.25):*

$$\begin{aligned} \mathcal{F}_A(f,1) &= \left\{ x \in \mathbb{R}^n \middle| \quad Q_1^T \nabla f(x) = 0 \right\} \\ &= \left\{ x \in \mathbb{R}^n \middle| \quad \frac{\partial f(z_0)}{\partial y}\frac{\partial f(z)}{\partial x} - \frac{\partial f(z_0)}{\partial x}\frac{\partial f(z)}{\partial y} = 0 \right\} \\ &= T(f, z_0). \end{aligned}$$

Unter der Voraussetzung lokaler Regularität der Abbildung $H_{n-k}(x) := Q_{n-k}^T \nabla f(x)$,

$$Rang\left(DH_{n-k}(x)\right) = Rang\left(Q_{n-k}^T D^2 f(x)\right) = n-k,$$

kann gezeigt werden (vgl. Abschnitt 1.2.2), daß das k-Newton-Blatt $\mathcal{F}_A(f,k)$ der Funktion f eine lokal k-dimensionale nichtlineare Mannigfaltigkeit darstellt. Mit Hilfe der orthogonalen Matrizen Q_{n-k} werden anderseits gleichzeitig auch lineare Mannigfaltigkeiten der Dimension $l = n-k$ in $\mathbb{R}^n$ definiert:

$$\mathcal{M}_A(t,k) := \{x \in \mathbb{R}^n | \quad x = t + Q_l \xi\}.$$

Bemerkung 71 *Der Punkt t wird als Ausgangspunkt für eine Verbindungstrajektorie benutzt und wird weiter als* ***Gitterpunkt*** *bezeichnet. Die Gitterpunkte werden während der Rekonstruktion einer Trajektorie nach einer bestimmten Strategie (vgl. Abschnitt 2.2) bestimmt.*

Die Menge der kritischen Punkte der eingeschränkten Funktion $f|_{\mathcal{M}_A(t,k)}(x)$ kann als Schnitt der Mannigfaltigkeit $\mathcal{M}_A(t,k)$ mit dem entsprechenden k-Newton-Blatt $\mathcal{F}_A(f,k)$ dargestellt werden.

Lemma 72 *Seien eine Funktion $f \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ und eine orthogonale Matrix $Q \in \mathbb{R}^{n \times n}$ gegeben.*
Für die Funktion $\phi_t^l(\xi) := f|_{\mathcal{M}_A(t,k)}(x) = f(t + Q_{n-k}\xi)$, $\xi \in \mathbb{R}^{n-k}$ gilt:

$$\begin{array}{c} \nabla \phi_t^l(\xi) = 0 \\ \Updownarrow \\ x = t + Q_{n-k}\xi \in \mathcal{M}_A(t,k) \cap \mathcal{F}_A(f,k). \end{array}$$

Beweis. Nach der Kettenregel kann man den Gradientenvektor $\nabla \phi_t^l(\xi)$ der eingeschränkten Funktion ϕ_t^l als Projektion des Gradienten $\nabla f(x)$ der Zielfunktion f darstellen:

$$\nabla \phi_t^l(\xi) = Q_{n-k}^T \nabla f(x).$$

Die Stationaritätsbedingung $\nabla \phi_t^l(\xi) = 0$ ist also genau dann erfüllt, wenn der Gradient $\nabla f(x)$ der Zielfunktion f im Kern der Matrix Q_{n-k} liegt. ■

2.1.2 Rekursive Konstruktion für klassische Newton-Trajektorien

Die klassische Newton-Trajektorie aus dem Abschnitt 1.3 wurde auf drei äquivalenten Wegen konstruiert:

1. Mit Hilfe einer Matrix $G = (g_1, ..., g_{n-1})$

$$T_G(f) := \left\{x \in \mathbb{R}^n | \quad G^T \nabla f(x) = 0\right\}$$

2. Mit Hilfe eines Vektors $0 \neq g \in \ker G^T$

$$T_g(f) = \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda \mathbf{g} \right\}$$

3. Mit Hilfe einer Funktion $H(x)$

$$T_H(f) := \{ x \in \mathbb{R}^n | \quad H(x) = 0 \}.$$

Konstruktion der Unterprobleme

Sei $g \in \mathbb{R}^n$ ein von Null verschiedener Vektor aus dem Kern der Matrix G. Mit $\eta_g(t)$ wird die zum Vektor g normale Hyperebene, die den Gitterpunkt t enthält, bezeichnet:

$$\eta_g(t) := \left\{ x \in \mathbb{R}^n | \quad g^T(x-t) = 0 \right\}. \tag{2.2}$$

Lemma 73 *Für die Matrix $A = (G, g)$ entspricht die Hyperebene $\eta_g(t)$ der Mannigfaltigkeit $\mathcal{M}_A(t, n-1)$*

$$\eta_g(t) = \mathcal{M}_A(t, n-1). \tag{2.3}$$

Beweis. Mit der QR-Zerlegung der Matrix A, kann die Matrix G als lineare Kombination der Spaltenvektoren $q_1, ..., q_{n-1}$ der Matrix Q dargestellt werden (vgl. Korollar 69):

$$G = Q_{n-1} R_{n-1,n-1}.$$

Daraus folgt, daß der Spaltenvektor q_n zusammen mit dem Vektor g im Kern der Matrix G liegt und deshalb sich wie folgt darstellen läßt:

$$q_n = \sigma \frac{g}{\|g\|}, \qquad \sigma \in \{-1, 1\}.$$

Die Äquivalenz der beiden Mengen $\eta_g(t)$ und $\mathcal{M}_A(t, n-1)$ folgt dann trivial. ■

Aus dem Lemma (72) folgt, daß die Trajektorie $T_g(f)$ und die Hyperebene $\eta_g(t)$ sich genau in den kritischen Punkten der auf die Hyperebene $\eta_g(t)$ eingeschränkten Zielfunktion $f|_{\eta_g(t)}(x)$ schneiden. Gelingt es, diese Punkte zu finden, so könnten sie als Startpunkte für die neuen Komponenten der Trajektorie $T_g(f)$ benutzt werden. Dies führt zum folgenden Hilfsproblem:

Problem 74 *Sei $\phi_t : \mathbb{R}^{n-1} \to \mathbb{R}$ gegeben durch:*

$$\phi_t(\xi) := f|_{\eta_g(t)}(x) = f(t + G\xi).$$

Gesucht sind die kritischen Punkte der Funktion ϕ_t:

$$\nabla \phi_t(\xi) = 0.$$

Beispiel 75 *Für die zweidimensionale Zielfunktion* $f_a : \mathbb{R}^2 \to \mathbb{R}$ *und den Startpunkt* z_0^1 *aus dem Beispiel 20*

$$\begin{aligned} z_0^1 &= (1.2, 0.6) \\ f_a(z) &= \frac{1}{3}\left(x^3 - y^3\right) - x + y, \end{aligned}$$

ist der induzierende Vektor $g = \nabla f\left(z_0^1\right)$ *gegeben durch:*

$$g = (0.44, 0.64)^T.$$

Eine geignete 2×1 *Matrix* G *kann hier direkt angegeben werden:*

$$G = (0.64, -0.44)^T.$$

Für einen Gitterpunkt $t = (t_1, t_2)$ *kann die Hilfsfunktion* $\phi_t(\xi)$ *aus dem Problem 74 wie folgt berechnet werden:*

$$\phi_t(\xi) = \frac{1}{3}\left((t_1 + 0.64\xi)^3 - (t_2 - 0.44\xi)^3 - (t_1 + 0.64\xi) + (t_2 - 0.44\xi)\right).$$

Für die kritischen Punkte ξ^* *der Funktion* $\phi_t(\xi)$ *gilt:*

$$\phi_t'(\xi^*) = 0.64\,(t_1 + 0.64\xi^*)^2 + 0.44\,(t_2 - 0.44\xi^*)^2 - 1.08 = 0.$$

Das Hilfsproblem hat also abhängig vom Punkt t zwei, eine oder keine Lösung. Das bedeutet, daß die Hyperebene $\eta_g(t)$*, die in diesem Fall einer Gerade entspricht, und die Trajektorie* $T_g(f)$ *sich entsprechend in zwei, einem oder keinem Punkt schneiden.*

Für das Hilfsproblem 74 wird eine klassische Newton-Trajektorie $T(\phi_t)$ konstruiert. Die Trajektorie $T(\phi_t)$ enthält alle kritischen Punkte der Funktion ϕ_t und verbindet deshalb alle Komponenten der Haupttrajektorie $T_g(f)$, die die Hyperebene $\eta_g(t)$ schneiden. Die Verbindungstrajektorie $T(\phi_t)$ ist allerdings selber nicht notwendigerweise zusammenhängend.

Die Spalten $g_1, ..., g_{n-1}$ der Matrix G wurden bis jetzt als schlicht linear unabhängig gewählt. Um die entsprechenden Hilfsprobleme für die rekursiv definierte Verbindungstrajektorien aufzustellen, wird im weiteren eine orthonormale Basis $\mathcal{Q} = \{q_1, ..., q_n\}$ mit der Eigenschaft $q_n = \frac{g}{\|g\|}$ konstruiert. Die Basis $\mathcal{Q}$ kann mittels des Gram-Schmidt-Orthogonalisierungsverfahrens oder äquivalent durch QR-Zerlegung der Matrix $A = (G, g)$ gewonnen werden (vgl. Korollar 69 und Lemma 73).

Die lineare Mannigfaltigkeit $\mathcal{M}_A(t,k)$ kann dann als Schnittmenge der zu den Vektoren $q_{n-k+1},...,q_n$ normalen Hyperebenen dargestellt werden:

$$\mathcal{M}_A(t,k) = \bigcap_{i=n-k+1}^{n} \eta_{q_i}(t). \tag{2.4}$$

Für die auf dem Niveau $n-k$, $(0 \le k < n-1)$ gesuchte Verbindungstrajektorie $T_G^{n-k}(f,t)$ wird auf folgende Weise ein Hilfsproblem definiert

Problem 76 *Hilfsproblem auf dem Rekursionsniveau $l = n-k$.*
Für die reguläre Matrix $A = (G,g)$ und einen Gitterpunkt $t \in \mathbb{R}^n$ seien die Mannigfaltigkeit $\mathcal{M}_A(t,k)$ und die Hilfsfunktion $\phi_t^l : \mathbb{R}^l \to \mathbb{R}$,

$$\phi_t^l(\xi) := f|_{\mathcal{M}_A(t,k)}(x) = f\left(t + Q_l^T \xi\right), \tag{2.5}$$

gegeben.
Gesucht sind die kritischen Punkte der Funktion ϕ_t^l:

$$\nabla \phi_t^l(\xi) = 0.$$

Rekursionsschema für klassische Newton-Trajektorien

Das Lemma 72 zeigt auf, wie ein Trajektoriennetz der Haupt- und Verbindungstrajektorien rekursiv aufgebaut werden kann. Im weiteren führen wir folgende Bezeichnungen ein:

$T^k(f,t)$-die auf dem Rekursionsniveau k ausgehend vom Gitterpunkt t rekonstruierte Verbindungstrajektorie

$\mathcal{P}\left(T^k(f,t)\right)$-die Menge der auf der Trajektorie $T^k(f,t)$ gefundenen Gitterpunkte

$\mathcal{C}\left(T^k(f,t)\right)$-die Menge der auf der Trajektorie $T^k(f,t)$ gefundenen kritischen Punkte

Das Rekursionschema kann dann wie folgt aufgebaut werden:

Rekursionsschema 77 *(für klassische Newton-Trajektorien)*

1. *Ausgehend von dem Ausgangspunkt x^0 wird eine Komponente der gesuchten Trajektorie $T(f)$ rekonstruiert und die auf der Komponente liegenden kritischen Punkte der Zielfunktion identifiziert. Nach einer bestimmten Strategie werden gleichzeitig Gitterpunkte für die neuen auf dem Rekursionsniveau $n-1$ liegenden Verbindungstrajektorien $T^{n-1}(f,t)$ bestimmt und gespeichert.*

2. *Ausgehend von dem jeweiligen Gitterpunkt t werden die Verbindungstrajekorien $T^{n-1}(f,t)$ rekonstruiert. Die gefundenen kritischen Punkte $\mathcal{C}(T^{n-1}(f,t))$ können dann als Startpunkte für die Rekonstruktion der neuen Komponenten der Haupttrajektorie $T(f)$ benutzt werden.*

3. *Die Verbindungstrajektorien $T^{n-1}(f,t)$ enthalten natürlich neue Gitterpunkte $\mathcal{P}(T^{n-1}(f,t))$, die als Startpunkte für die auf dem Rekursionsniveau $n-2$ liegenden Verbindungstrajektorien $T^{n-2}(f,t)$ benutzt werden können.*

4. *Wird allgemein eine Verbindungstrajektorie $T^k(f,t)$ auf dem Rekursionsniveau k rekonstruiert, so werden die gefundenen kritischen Punkte $\mathcal{C}\left(T^k(f,t)\right)$ als Startpunkte für die Verbindungstrajektorien auf dem Rekursionsniveau $k+1$ (vgl. Lemma 72) und die Gitterpunkte $\mathcal{P}\left(T^k(f,t)\right)$ als Startpunkte für die Verbindungstrajektorien auf dem Rekursionsniveau $k-1$ benutzt.*

Verbindungstrajektorien

Das Problem 76 impliziert folgende Trajektorie $T_G^l(f,t)$:

$$T_G^l(f,t) := \left\{ x \in \mathcal{M}_A(t,k) \mid \quad Q_{l-1}^T \nabla f(x) = 0 \right\}. \tag{2.6}$$

Lemma 78 *Die Trajektorie $T_G^l(f,t)$ ist die Lösungsmenge des folgenden Gleichungssystems von k linearen und $l-1$ nichtlinearen Gleichungen:*

$$\begin{cases} (q_{l+1},...,q_n)^T (x-t) = 0 \\ (q_1,...,q_{l-1})^T \nabla f(x) = 0 \end{cases}$$

und enthält alle kritischen Punkte von $\phi_t^l(\xi)$ bzw. $f|_{\mathcal{M}_A(t,k)}(x)$.

Beweis. Die k lineare Gleichungen sind äquivalent zu der Zugehörigkeit jedes Trajektorienpunktes x der Schnittmenge $\mathcal{M}_A(t,k)$:

$$x \in \mathcal{M}_A(t,k) \Longleftrightarrow (q_{l+1},...,q_n)^T (x-t) = 0.$$

Die $l-1$ nichtlinearen Gleichungen folgen direkt aus der Definition der Trajektorie $T_G^l(f,t)$.
Da die Funktion $f|_{\mathcal{M}_A(t,k)}(x)$ nur für $x = t+\xi$ aus $\mathcal{M}_A(t,k)$ definiert ist, gilt für den kritischen Punkt x^*:

$$t + \xi^* =: x^* \in \mathcal{M}_A(t,k).$$

Andererseits gilt für einen kritischen Punkt ξ^* der Funktion ϕ_t^k auch:

$$0 = \nabla\phi_t^l(\xi^*) = Q_l^T \nabla f(x^*) \Rightarrow Q_{l-1}^T \nabla f(x) = 0.$$

Der Punkt x^* gehört also zu der Trajektorie $T_G^l(f,t)$. ■

Die folgende Eigenschaft sichert, daß das rekursive Verfahren auf dem Niveau $l = 1$ abgeschlossen werden kann und nicht in eine endlosschleife geriet.

Korollar 79 *Die Trajektorie $T_G^1(f,t)$ ist eine durch das Gleichungssystem*

$$(g_{2,\ldots,g_n})^T (x - t) = 0$$

definierte Gerade und besteht deshalb aus einer Komponente.

Beispiel 80 *Für das im Beispiel 75 definierte Hilfsproblem ist die entsprechende Hilfstrajektorie $T_G^1(f_a,t)$ definiert durch:*

$$0.44(x - t_1) + 0.64(y - t_2) = 0.$$

oder äquivalent durch:

$$(x,y) = (t_1 + 0.64\xi, t_2 - 0.44\xi).$$

Für die Punkte der Trajektorie $T_G^1(f_a,t)$ kann die Ableitung $\phi_t'(\xi)$ der Funktion $\phi_t(\xi)$ wie folgt dargestellt werden:

$$\phi_t'(\xi) = 0.64x^2 + 0.44y^2 - 1.08.$$

Die kritischen Punkte der Hilfsfunktion $\phi_t(\xi)$ gehören also gleichzeitig der Hilfstrajektorie $T_G^1(f_a,t)$ und der Haupttrajektorie $T(f_a, z_0^1)$ Die Hilfstrajektorie $T_G^1(f_a,t)$ ist in diesem Fall eine Gerade und entspricht der Hyperbene $\eta_g(t)$ (vgl. Beispiel 20).

Die bei der Rekonstruktion der Trajektorie $T_G^l(f,t)$ entdeckten kritischen Punkte der Funktion ϕ_t^l können als Startpunkte für neue Komponenten der Verbindungstrajektorie $T_G^{l+1}(f,t)$ benutzt werden.

Satz 81 *Für einen beliebigen Punkt $t \in \mathbb{R}^n$ und die rekursiv definierten Trajektorien $T_G^l(f,t)$ und $T_G^{l+1}(f,t)$ gilt:*

$$x^* = t + B\xi^* \in M_Q(t,k) \cap T_G^{l+1}(f,t)$$
$$\Updownarrow$$
$$\nabla\phi_t^k(\xi^*) = 0.$$

Beweis. Aus dem Lemma (78) folgt:

$$\begin{array}{c} x^* \in M_Q(t,k) \cap T_G^{n-k+1}(f,t) \\ \Updownarrow \\ \left\{ \begin{array}{c} (q_{n-k+1,\ldots,}q_n)^T (x-t) = 0 \\ (q_1,\ldots,q_{n-k-1})^T \nabla f(x) = 0. \end{array} \right. \\ \Updownarrow \\ \nabla \phi_t^l(\xi^*) = 0. \end{array}$$

■

2.1.3 Rekursive Konstruktion für Richtungsfeld-Trajektorien

Die Richtungsfeld–Trajektorie aus dem Abschnitt (1.4) wurde ebenfalls auf drei äquivalenten Wegen konstruiert:

1. Mit Hilfe des Richtungsfeldes $r(x)$

$$T_r(f) = \left\{ x \in \mathbb{R}^n | \quad \exists_{\lambda \in \mathbb{R}} \nabla f(x) = \lambda r(x) \right\}$$

2. Mit Hilfe einer variablen, zum Richtungsfeld $r(x)$ orthogonalen Matrix $G(x) = (g_1(x), \ldots, g_{n-1}(x))$, $G(x)^T r(x) = 0$

$$T_G(f) := \left\{ x \in \mathbb{R}^n | \quad G(x)^T \nabla f(x) = 0 \right\}$$

3. Mit Hilfe einer Funktion $H(x)$

$$T_H(f) := \left\{ x \in \mathbb{R}^n | \quad H(x) = 0 \right\}.$$

Konstruktion der Unterprobleme

Die variable Matrix $G(x)$ kann hier **nicht** zur Konstruktion der Hilfsprobleme benutzt werden. Stattdessen wird eine orthogonale Matrix Q eingesetzt:

$$\begin{array}{rcl} Q & = & (q_1, \ldots, q_n) \\ Q^T Q & = & Q Q^T = I. \end{array}$$

Die Matrix Q ist **von der Matrix** $G(x)$ **nicht abhängig.** In jedem Punkt $x \in \mathbb{R}$ kann aber der Richtungsvektor $r(x) \in \mathbb{R}^n$ und die variable $n \times n-1$

Matrix $G(x)$ als Linearkombination der Spalten $q_1, ..., q_n$ der orthogonalen Matrix Q dargestellt werden:

$$(G(x), r(x)) = Q^T C(x).$$

Die Koeffizientenmatrix $C(x) = (\psi_1, ..., \psi_n)$ können leicht berechnet werden:

$$\psi_i = \begin{cases} Qg_i(x), & i = 1, ..., n-1 \\ Qr(x) & i = n. \end{cases}$$

Für den Vektor $q := q_n$ und einen Punkt $t \in \mathbb{R}^n$ wird jetzt die auf die Hyperebene $\eta_q(t)$ eingeschränkte Zielfunktion $\phi_t(\xi) := f|_{\eta_q(t)}(x)$ betrachtet und ein entsprechendes Optimierungsproblem definiert:

Problem 82 *Für die Zielfunktion f sei $\phi_t : \mathbb{R}^{n-1} \to \mathbb{R}$ gegeben durch:*

$$\phi_t(\xi) = f\left(t + Q_{n-1}^T \xi\right).$$

Gesucht sind die kritischen Punkte der Funktion ϕ_t:

$$\nabla \phi_t(\xi) = 0.$$

Für das Hilfsproblem 82 können Trajektorien konstruiert werden, die alle kritischen Punkte der Funktion ϕ_t enthalten. Der Gradient $\nabla f(x)$ der Zielfunktion liegt dann im Kern der Matrix Q_{n-1}^T,

$$\nabla \phi_t(\xi) = 0 \Leftrightarrow Q_{n-1}^T \nabla f(x) = 0.$$

Bemerkung 83 *Da die variable Richtung $r(x)$ im allgemeinen nicht im Kern der Matrix Q_{n-1}^T liegt, gehören die kritischen Punkte der Funktion ϕ_t nicht automatisch zu der Haupttrajektorie $T_r(f)$.*

Das Lemma 72 findet hier also keine Anwendung!

Beispiel 84 *Für die zweidimensionale Zielfunktion $f_a : \mathbb{R}^2 \to \mathbb{R}$ und den Startpunkt z_0^1 aus dem 20*

$$\begin{aligned} z_0^1 &= (1.2, 0.6) \\ f_a(z) &= \frac{1}{3}\left(x^3 - y^3\right) - x + y, \end{aligned}$$

wird hier die orthogonale Matrix $Q = I_2$ gewählt. Für einen Gitterpunkt $t = (t_1, t_2)$ kann die auf die Hyperebene $\eta_{(0,1)} = \{(x, y) \in IR^2 | \quad y - t_2 = 0\}$ eingeschränkte Hilfsfunktion $\phi_t(\xi)$ wie folgt berechnet werden:

$$\phi_t(\xi) = \frac{1}{3}\left((t_1 + \xi)^3 - {t_2}^3 - (t_1 + \xi) + t_2\right).$$

Für die kritischen Punkte ξ^* der Funktion $\phi_t(\xi)$ gilt:

$$\phi_t'(\xi^*) = (t_1 + \xi^*)^2 - 1 = 0.$$

Daraus folgt für die gefundenen kritischen Punkte (x^*, y^*) der Funktion $f_a|_{\eta_{(0,1)}(t)}(z)$

$$(x^*, y^*) = \left(\dot{t}_1 + \xi^*, t_2\right) = (\pm 1, t_2).$$

Falls $t_2 \neq \pm 1$ gehören die Punkte (x^*, y^*) nicht der Haupttrajektorie $T(f_a, z_0^1)$.

Die Konstruktion der Hilfsprobleme für die klassischen Newton-Trajektorien (vgl. Problem 76) kann hier nicht direkt übertragen werden. Es muß also eine geeignete Verbindungstrajektorie $T_r^{n-1}(f)$ konstruiert werden, die außer den kritischen Punkten des Hilfsproblems auch noch die Schnittpunkte der Trajektorie $T_r(f)$ und der Hyperebene $\eta_q(t)$ enthält.
Dies gilt für eine Richtungsfeld-Trajektorie $T_r^{n-1}(f,t)$ des Hilfsproblems dann, wenn das Richtungsfeld $r(x)$ auf die Hyperebene $\eta_q(t)$ projiziert wird:

$$T_r^{n-1}(f,t) := \left\{ x \in \eta_q(t) \mid \quad \exists_{\lambda(x)\in\mathbb{R}} \nabla\phi_t(\xi) = \lambda(x) Q_{n-1}^T r(x) \right\}.$$

Diese Vorgehensweise kann auf alle Rekursionsniveaus verallgemeinert werden. Mit der bereits eingeführten Bezeichnungen der Schnittmenge $\mathcal{M}_A(t,k)$ (vgl. 2.4) und der eingeschränkten Funktion $\phi_t^l(\xi)$ (vgl. 2.5) führt dies zu folgenden Definition 85 (vgl. auch 2.6).

Definition 85 *Die Verbindungstrajektorie $T_r^l(f)$ auf dem Rekursionsniveau $l = n - k$ wird allgemein definiert durch:*

$$T_r^l(f,t) := \left\{ x \in \mathcal{M}_A(t,k) \mid \quad \exists_{\lambda(x)\in\mathbb{R}} \nabla\phi_t^l(\xi) = \lambda(x) Q_l^T r(x) \right\}.$$

Gilt in einer Umgebung $U(t) \subset \mathbb{R}^n$ des Punktes $t \in \mathbb{R}^n$:

$$\forall_{x\in U(t)} \quad q_l^T r(x) \neq 0, \tag{2.7}$$

so kann der Parameter $\lambda(x)$ und die Trajektorie $T_r^l(f,t)$ wie folgt dargestellt werden (vgl. 2.6):

$$\begin{aligned} \lambda(x) &= \frac{q_l^T \nabla f(x)}{q_l^T r(x)}, \\ T_r^l(f,t) &:= \left\{ x \in \mathcal{M}_A(t,k) \mid \quad Q_{l-1}^T \nabla f(x) = \lambda(x) Q_{l-1}^T r(x) \right\}. \end{aligned}$$

In diesem Fall kann das Lemma 78 dierekt übertragen werden:

Lemma 86 *Die Trajektorie $T_r^l(f,t)$ ist die Lösungsmenge des folgenden Gleichungssystems von k linearen und $l-1$ nichtlinearen Gleichungen:*

$$\begin{cases} (q_{l+1},...,q_n)^T (x-t) = 0 \\ (q_1,...,q_{l-1})^T \nabla f(x) = \lambda(x) (q_1,...,q_{l-1})^T r(x) \end{cases}$$

und enthält alle kritischen Punkte von $\phi_t^l(\xi)$ bzw. $f|_{\mathcal{M}_A(t,k)}(x)$.

Beweis. Vgl. 78. ■

Bemerkung 87 *Die Bedingung (2.7) muß allerdings für jedes Rekursionsniveau l geprüft werden. Dieser Aufwand kann reduziert werden, in dem die Bedingung (2.7) durch folgende ersetzt wird:*

$$\forall_{x \in U(t)} \quad q_1^T r(x) \neq 0. \tag{2.8}$$

Die Trajektorie $T_r^l(f,t)$ ist dann die Lösungsmenge des folgenden Gleichungssystems:

$$\begin{cases} (q_{l+1},...,q_n)^T (x-t) = 0 \\ (q_2,...,q_l)^T \nabla f(x) = \lambda(x) (q_2,...,q_l)^T r(x) \end{cases}$$

Für die neudefinierten Richtungsfeld-Verbindungstrajektorien können dann die gewünschten Eigenschaften bewiesen werden:

Satz 88 *Für die Verbindungstrajektorien $T_r^l(f,t)$, $(1 \leq l < n)$ gilt:*

(i) die Trajektorie $T_r^{n-1}(f)$ enthält alle kritischen Punkte der eingeschränkten Zielfunktion ϕ_t aus dem Problem 82:

$$(\nabla \phi_t(\xi) = 0) \Rightarrow \left(x := t + Q_{n-1}^T \xi \in T_r^{n-1}(f)\right)$$

(ii) alle Schnittpunkte der Trajektorie $T_r(f)$ und der Hyperebene $\eta_q(t)$ gehören zu der Trajektorie $T_r^{n-1}(f)$:

$$\left(x \in T_r(f) \cap \eta_q(t)\right) \Rightarrow x \in T_r^{n-1}(f)$$

(iii) alle Schnittpunkte der Trajektorie $T_r^{l+1}(f)$ und zu der Hyperebene $\eta_{q_l}(t)$ gehören der Trajektorie $T_r^l(f)$

$$\left(x \in T_r^{l+1}(f,t) \cap \eta_q(t)\right) \Rightarrow x \in T_r^l(f).$$

Beweis.

(i) Mit der Wahl $\lambda = 0$ ist die Behauptung trivial bewiesen.

(ii) Für jeden Trajektorienpunkt $x := t + Q_{n-1}^T \xi \in T_r(f) \cap \eta_q(t)$, gibt es eine Zahl $\lambda(x)$, so daß gilt:

$$\begin{gathered} \nabla \phi_t(\xi) = Q_{n-1}^T \nabla f(x) = \lambda(x) Q_{n-1}^T r(x) \\ \Downarrow \\ x \in T_r^{n-1}(f). \end{gathered}$$

(iii) Für die Punkte der Trajektorie $T_r^{l+1}(f,t)$ gilt:

$$Q_{l+1}^T \nabla f(x) = \nabla \phi_t^{l+1}(\xi) = \lambda(x) Q_{l+1}^T r(x).$$

Insbesondere für die Schnittpunkte der Trajektorie $T_r^{l+1}(f,t)$ und der Hyperebene $\eta_{q_l}(t)$ gilt also:

$$\nabla \phi_t^l(\xi) = Q_l^T \nabla f(x) = \lambda(x) Q_l^T r(x).$$

■

Die bei der Rekonstruktion der Trajektorie $T_r^l(f,t)$ gefundenen Schnittpunkte von $T_r^{l+1}(f,t)$ und $\eta_{q_l}(t)$ können als Startpunkte auf der Suche nach neuen Komponenten der Trajektorie $T_r^{l+1}(f,t)$ benutzt werden.

Bemerkung 89 *Die Trajektorie $T_r^{n-1}(f)$ verbindet alle Komponenten der Haupttrajektorie $T_r(f)$, die die Hyperebene $\eta_q(t)$ schneiden und rekursiv: Die Trajektorie $T_r^l(f,t)$ verbindet alle Komponenten der Trajektorie $T_r^{l+1}(f,t)$, die die Hyperebene $\eta_{q_l}(t)$ schneiden.*

Rekursionsschema für Richtungsfeld-Trajektorien

Auf der Grundlage des Satzes 88 kann ein allgemeines Rekursionsschema für Richtungsfeld-Trajektorien formuliert werden:

Rekursionsschema 90 *(für Richtungsfeld-Trajektorien)*

1. *Ausgehend von dem Ausgangspunkt x^0 wird eine Komponente der gesuchten Trajektorie $T_r(f)$ rekonstruiert und die auf der Komponente liegenden kritischen Punkte der Zielfunktion identifiziert. Nach einer bestimmten Strategie werden gleichzeitig Gitterpunkte für die neuen auf dem Rekursionsniveau $n-1$ liegenden Verbindungstrajektorien $T_r^{n-1}(f,t)$ bestimmt und gespeichert.*

2. *Ausgehend von dem jeweiligen Gitterpunkt t werden die Verbindungstrajekorien $T_r^{n-1}(f,t)$ rekonstruiert. Die Schnittpunkte der Verbindungstrajektorie mit der Haupttrajektorie $T_r(f)$ werden erkannt und als Startpunkte für die Rekonstruktion der neuen Komponenten benutzt.*

3. *Die Verbindungstrajektorien $T_r^{n-1}(f,t)$ enthalten natürlich neue Gitterpunkte, die als Startpunkte für die auf dem Rekursionsniveau $n-2$ liegenden Verbindungstrajektorien $T_r^{n-2}(f,t)$ benutzt werden können.*

4. *Wird allgemein eine Verbindungstrajektorie $T_r^k(f,t)$ auf dem Rekursionsniveau k rekonstruiert, so werden die Schnittpunkte mit der Trajektorie $T_r^{k+1}(f,t)$ gesucht und als Startpunkte für die Verbindungstrajektorien auf dem Rekursionsniveau $k+1$ benutzt.*
   Die auf der Trajektorie $T_r^k(f,t)$ gefundenen Gitterpunkte dienen als Startpunkte für die Verbindungstrajektorien auf dem Rekursionsniveau $k-1$.

Erkennung der Schnittpunkte

Die Schnittpunkte von $T_r^{l+1}(f,t)$ und $\eta_{q_l}(t)$ sind hier keine kritischen Punkte der Hilfsfunktion $\phi_t^l(\xi)$, können aber mit Hilfe des Faktors $\lambda(x)$ (vgl. Definition 85) charakterisiert werden. Bei der Rekonstruktion der Trajektorie der Trajektorie $T_r^l(f,t)$ kann der Faktor $\lambda(x)$ (vgl. Bedingungen 2.7 bzw. 2.8) mit wenig Aufwand mitbestimmt werden. Um die Erkennung der gesuchten Schnittpunkte zu ermöglichen wird hier eine Abtastfunktion $\tau_l(x) : \mathbb{R}^n \to \mathbb{R}$ eingeführt:

$$\tau_l(x) := q_{l+1}^T \nabla f(x) - \lambda(x)\, q_{l+1}^T r(x).$$

Satz 91 *Ist für einen Trajektorienpunkt $x \in T_r^l(f,t)$ die Abtastfunktion $\tau_l(x)$ gleich Null, so gehört der Punkt auch zu der Trajektorie $T_r^{l+1}(f,t)$:*

$$(\tau_l(x) = 0) \Rightarrow x \in T_r^{l+1}(f).$$

Beweis. Für jeden Trajektorienpunkt $x \in T_r^l(f,t)$ gilt:

$$x \in T_r^l(f,t) \Rightarrow x \in \eta^l(t) \Rightarrow x \in \eta^{l+1}(t)$$

$$Q_l^T \nabla f(x) = \lambda(x)\, Q_l^T r(x).$$

Mit der Bedingung $\tau_l(x) = 0$ folgt dann auch:

$$\begin{aligned}
\nabla\phi_t^{l+1}(\xi) &= Q_{l+1}^T \nabla f(x) = (Q_l, q_{l+1})^T \nabla f(x) \\
&= \left(Q_l^T \nabla f(x), q_{l+1}^T \nabla f(x)\right)^T = \left(\lambda(x)\, Q_l^T r(x), \lambda(x)\, q_{l+1}^T r(x)\right)^T \\
&= \lambda(x)\, Q_{l+1}^T r(x).
\end{aligned}$$

■

2.2 Strategien zur Bestimmung der Gitterpunkte

Eine wichtige Rolle im vorgestellten Rekursionsschema spielt die Strategie der Bestimmung der Ausgangspunkten für die Verbindungstrajektorien. In [15] wurde die Strategie der Berührungspunkte vorgeschlagen. Auch die Konstruktion eines äquidistanten bzw. angepaßten inäquidistanten Netzes ist möglich. Dies wird mit Hilfe der entsprechend definierten Gitterpunkte in den Abschnitten 2.2.2 und 2.2.3 vorgeführt. Die Strategien werden im weiteren an einem Beispielproblem diskutiert.
Für die Konstruktion des Trajektoriennetzes werden Hyperebenen benötigt, die zu einer bestimmten Richtung q orthogonal liegen. Hierzu werden zunächst folgende Bezeichnungen eingeführt.

Definition 92 *Sei* $\mathcal{H}_q = \left\{ \eta_q(t) \mid \quad t \in \mathbb{R}^n \right\}$ *der Raum der zu einem Vektor* q *orthogonalen Hyperebenen. Der Raum* $\mathcal{H}_q$ *ist mit folgenden Homöomorphismus* $\phi : \mathcal{H}_q \to \mathbb{R}$ *homöomorph zu* $\mathbb{R}$*:*

$$\phi\left(\eta_q(t)\right) \to q^T t$$

Mit $\partial\phi(\eta)$ *wird die Menge der Randpunkte der Bildmenge* $\phi(\eta) \subset \mathbb{R}$ *bezeichnet.*
Mit $\pi(t) = \eta_q(t)$ *ist weiter eine stetige Abbildung von* $\mathbb{R}^n$ *in* $\mathcal{H}_q$ *gegeben .*

2.2.1 Strategie der Berührungspunkte

Ein Berührungspunkt t der Trajektorie $T(f)$ bzgl. der Richtung q wird wie folgt charakterisiert.

Definition 93 *Ein Trajektorienpunkt* $t \in T(f)$ *heißt* ***Berührungspunkt von*** $T(f)$ ***bezüglich der Richtung*** q*, wenn folgende Bedingung erfüllt ist:*

$$\exists_{U(t)\subset\mathbb{R}^n} \mid \quad \phi(\pi(t)) = q^T t \in \partial\phi\left(\pi\left(U(t) \cap T(f)\right)\right).$$

Mit $\eta_q(t)$ *ist die* ***berührende Hyperebene*** *der Familie* $\mathcal{H}_q$ *bzgl. der Trajektorie* $T(f)$ *gegeben.*

Es gibt also eine offene Umgebung $U(t) \subset \mathbb{R}^n$ von t, so daß die Hyperebene $\eta_q(t)$ ein Randpunkt der in $\mathcal{H}_q$ abgebildeten Schnittmenge $U(t) \cap T(f)$ ist (s. Abbildung 2.1).

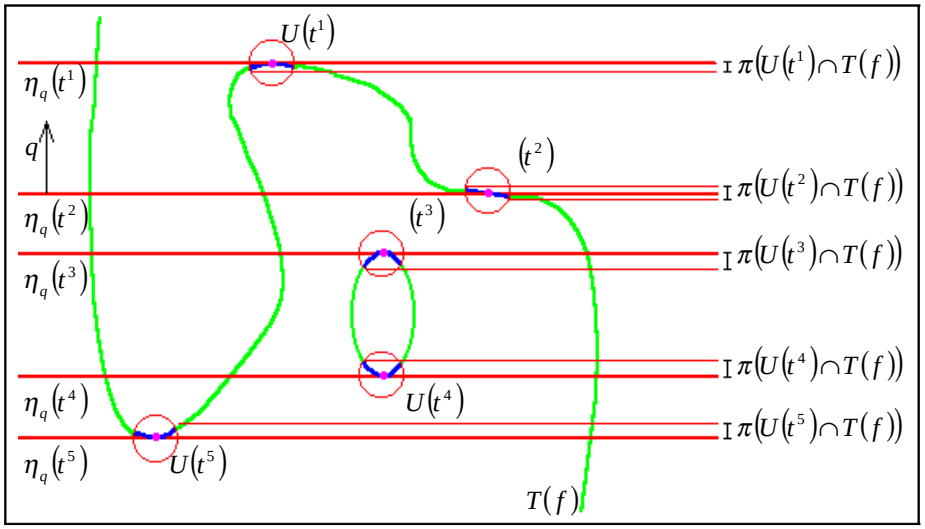

Abbildung 2.1: Berührungspunkte (t^1, t^3, t^4, t^5) einer Trajektorie bzgl. der Richtung q. t^2 ist kein Berührungspunkt der Trajektorie $T(f)$.

Da die Berührungspunkte gleichzeitig auch Trajektorienpunkte sind, können sie während der Rekonstruktion der Trajektorie mit wenig Aufwand mitbestimmt werden. Folgt man der Trajektorie $T(f)$ in der Nähe eines Berührungspunktes t, so nähert man sich der berührenden Hyperebene $\eta_q(t)$. Man erreicht sie dann im Punkt t und entfernt sich dann wieder von ihr (s. Abbildung 2.2). Man bleibt vorerst auf der gleichen Seite der Hyperebene. Die Hyperebene wird nicht im Berührungspunkt t von der Trajektorie $T(f)$ geschnitten (vgl. Abbildung 2.1).

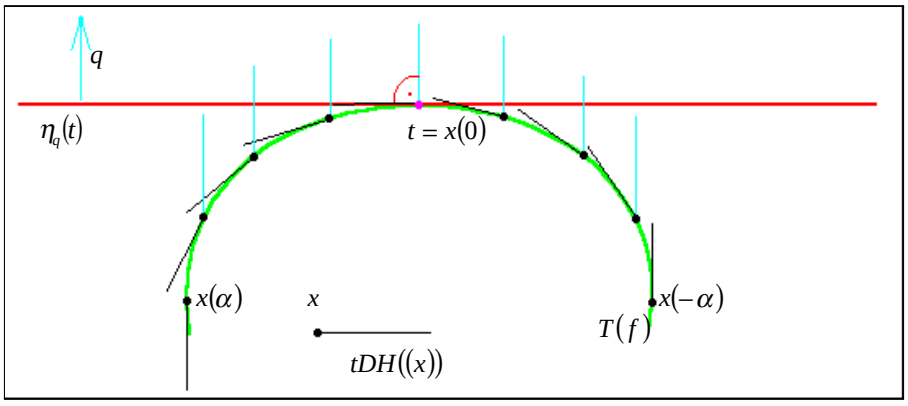

Abbildung 2.2: Suche nach einem Berührungspunkt

Um die Berührungspunkte während der Rekonstruktion der Trajektorie zu erkennen, werden hier Abtastfunktionen $\phi_q, \tau_q : T(f) \to \mathbb{R}$ definiert. Die Berührungspunkte sind dann entsprechend Extrema oder Nullstellen dieser Funktionen.

Lemma 94 *Sei $t^* \in T(f)$ ein Trajektorienpunkt und $x(\alpha)$, $x(0) = t^*$ eine lokale Parametrisierung der Trajektorie $T(f)$ im Punkt t^*. Seien weiter die Abtastfunktionen gegeben durch:*

$$\begin{aligned} \phi_q(x(\alpha)) &:= x(\alpha)^T q \\ \tau_q(x(\alpha)) &:= \dot{x}(\alpha)^T q. \end{aligned}$$

Der Punkt t^* ist ein Berührungspunkt der Trajektorie $T(f)$ bezüglich der Richtung q genau dann, wenn

(i) die Abtastfunktion ϕ_q in t^* ein lokales Extremum annimmt:

$$\exists_{\varepsilon>0}\forall_{0<\alpha<\varepsilon} \left(\phi_q(t^*) - \phi_q(x(\alpha))\right)\left(\phi_q(t^*) - \phi_q(x(-\alpha))\right) \leq 0,$$

(ii) wenn die Abtastfunktion τ_q in t^* ihr Vorzeichen wechselt:

$$\exists_{\varepsilon>0}\forall_{0<\alpha<\varepsilon} \tau_q(x(\alpha))\,\tau_q(x(-a)) \leq 0.$$

Beweis. (i) Die Hyperebenen $\eta_q(t)$ sind gleichzeitig auch Niveaumenge der Hilfsfunktion ϕ_q:

$$\begin{aligned} x, y &\in \eta_q(t) \Rightarrow \\ \phi_q(x) &= x^T q = x^T q + (t-x)^T q + (y-t)^T q = y^T q = \phi_q(x). \end{aligned}$$

Da die Trajektorie $T(f)$ die Hyperebene $\eta_q(t^*)$ im Punkt t^* erreicht, aber nicht schneidet, muß die Hilfsfunktion $\phi_q(x(\alpha))$ in t^* ihren lokalen Extremalwert annehmen.
(ii) Die Behauptung folgt durch Ableitung direkt aus (i).
Das Vorzeichen der Hilfsfunktion τ_q ist vom Cosinus-Wert des Winkels zwischen dem Tangentialvektor $\dot{x}(\alpha)$ und dem vorgegebenen Vektor q abhängig. Die beiden Vektoren sind im Punkt t^* orthogonal. Auf der einen Seite des Punktes t^* ist der Winkel spitz und auf der anderen stumpf, so daß das Vorzeichen der Funktion $\tau_q(x(\alpha))$ sich in t^* ändern muß. ■

Bemerkung 95 *Sei $\mathcal{P}_q(T(f))$ die Menge der Berührungspunkte von $T(f)$ bezüglich der Richtung q. Unter bestimmten Voraussetzungen kann für einen beschränkten Bereich $B \subset \mathbb{R}^n$ bewiesen werden [15], daß die Hyperebenen $\pi(\mathcal{P}_q(T(f)))$ alle Komponenten der Trajektorie $T(f)$ verbinden. Aufgrund dieses Ergebnisses ist es naheliegend, die Berührungspunkte als Startpunkte für die Hilfsprobleme zu wählen.*

Im weiteren wird die rekursive Konstruktion anhand eines zweidimensionalen Beispiels noch einmal erklärt. Zuerst werden einige weitere Bezeichnungen eingeführt:

c^i - kritische Punkte der Zielfunktion,

t^j - Berührungspunkte der Trajektorie $T_g(f)$ bezüglich der Richtung g,

T^j - entsprechend für den Berührungspunkt t^j konstruierte Verbindungstrajektorie

s^k - mittels der Verbindungstrajektorien gefundene Ausgangspunkte zur Rekonstruktion der neuen Komponenten der Haupttrajektorie $T(f)$,

K^k - entsprechend vom Ausgangspunkt s^k konstruierte Komponente der Trajektorie $T(f)$

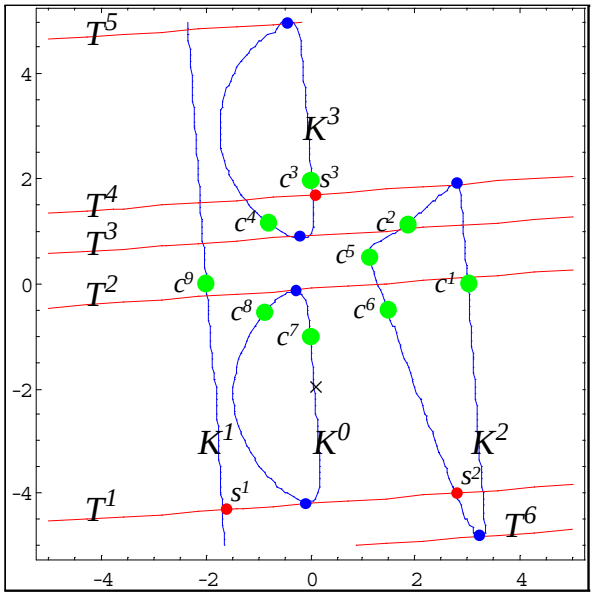

Abbildung 2.3: Trajektoriennetz zur Strategie der Berührungspunkte

Beispiel 96 *Beispiel zur Strategie der Berührungspunkte*
Die Trajektorie $T(f)$ besteht aus vier Komponenten, die mit Hilfe sechs eindimensionaler Hilfstrajektorien verbunden sind (vgl. Abbildung 2.3). Die Rekonstruktion der Trajektorie nahm folgenden Verlauf :

1. *Ausgehend vom Startpunkt x_s wurde die zyklische Komponente K^0 rekonstruiert. Dabei wurden zwei kritische Punkte c^1 und c^2 sowie zwei Berührungspunkte t^1 und t^2 gefunden.*

2. *Für die gefundenen Berührungspunkte wurden dann entsprechende Hilfsprobleme aufgestellt und die Verbindungstrajektorien T^1 und T^2 konstruiert. Sechs neue Ausgangspunkte wurden dabei entdeckt.*

3. *Ausgehend von den Punkten s^1 und s^2 konnten zwei neue Komponenten K^1 und K^2 der Haupttrajektorie gefunden werden. Die anderen Punkte führten nicht zu neuen Komponenten.*

4. *Die Komponente K^1 ist im untersuchten Bereich nicht zyklisch und enthält einen neuen kritischen Punkt c^3, aber keinen Berührungspunkt.*

5. *Die Komponente K^2 ist zyklisch und enthält vier neue kritische Punkte $c^4, ..., c^7$ und zwei Berührungspunkte t^3 und t^4. Für die neuen Berührungspunkte wurden die Verbindungstrajektorien T^3 und T^4 konstruiert und drei neue Ausgangspunkte entdeckt.*

6. *Ausgehend vom Punkt s^3 konnte dann die letzte Komponente K^3 der Haupttrajektorie gefunden werden, die die restlichen kritischen Punkte c^8 und c^9 der Zielfunktion sowie zwei weitere Berührungspunkte t^5 und t^6 enthält.*

7. *Im weiteren Verlauf des Verfahrens wurden noch die Verbindungstrajektorien T^5 und T^6 konstruiert und vier neue Ausgangspunkte entdeckt, die aber nicht mehr zu einer neuen Komponente der Trajektorie $T(f)$ führten.*

Bemerkung 97 *Zum Verlauf des Verfahrens und zur dargestellten Trajektorie läßt sich folgendes anmerken:*

a) *Eine Trajektorie $T(f)$ kann zyklische und/oder nichtzyklische Komponenten enthalten.*

b) *Jede zyklische Komponente enthält mindestens zwei Berührungspunkte bezüglich jeder beliebigen Richtung q.*

c) *Mit Hilfe der kritischen Punkte und Berührungspunkte kann leicht geprüft werden, ob eine Komponente zyklisch ist. Der erste während der Rekonstruktion der Komponente gefundene kritische Punkt bzw. Berührungspunkt sollte mit den neu entdeckten Punkten verglichen werden. Wird eine Übereinstimmung festgestellt, so kann die Rekonstruktion der Komponente abgeschlossen werden.*

d) Wird der erste während der Rekonstruktion der Komponente gefundene kritische Punkt bzw. Berührungspunkt mit allen bisher entdeckten verglichen, so kann die wiederholte Rekonstruktion einer bereits gefundenen und rekonstruierten Komponente sofort gestoppt werden.

e) Obwohl das gesamte Trajektoriennetz, d.h. die Haupttrajektorie $T(f)$ gemeinsam mit dem Verbindungstrajektorien $T^1, ..., T^6$, zusammenhängend ist, werden nicht immer alle Komponenten der Haupttrajektorie gefunden.
Wenn als Startpunkt für das Verfahren beispielweise ein Punkt aus der Komponente K^2 ausgesucht wird, so werden die anderen Komponenten nicht entdeckt. Die Komponente K^2 enthält nämlich keine Berührungspunkte.
Selbst wenn als Startpunkt ein Punkt aus der Komponente K^1 oder K^3 gewählt wird, kann die Komponente K^0 nicht erreicht werden, weil die Verbindungstrajektorien $T^3, ..., T^6$ sie nicht treffen.

2.2.2 Strategie des äquidistanten Netzes

Da die Strategie der Berührungspunkte nicht immer zum Erfolg führt (vgl. Bemerkung 97), kann diese durch die äquidistante Netzbildung ersetzt oder ergänzt werden. Mit der vorgegebenen Gitterbreite δ werden die Gitterpunkte wie folgt bestimmt.

Definition 98 *Ein Trajektorienpunkt $t \in T(f)$ heißt* ***Gitterpunkt von*** *$T(f)$* ***bezüglich der Richtung*** *q* ***und der Gitterbreite*** *δ, falls der Wert der Funktion ϕ_q (vgl. Lemma 94) sich als ganzzahlige Vielfaches m von δ darstellen läßt:*

$$\phi_q(t) = m\delta.$$

Wie Berührungspunkte können auch die Gitterpunkte leicht während der Rekonstruktion der Trajektorie $T(f)$ erkannt werden.

Beispiel 99 *Beispiel zur Strategie des äquidistanten Netzes*
Das äquidistante Netz für die Trajektorie $T(f)$ aus dem Beispiel 96 wurde hier mit der Gitterbreite $\delta = 1.5$ bestimmt. Die Rekonstruktion der Trajektorie $T(f)$ nahm diesmal folgenden Verlauf:

1. *Ausgehend vom Startpunkt x_s wurde die zyklische Komponente K^0 rekonstruiert. Dabei wurden zwei kritische Punkte c^1 und c^2 sowie vier Gitterpunkte t^1 und t^{1b} sowie t^2 und t^{2b} gefunden.*

2. *Die Gitterpunkte t^1 und t^{1b} sowie die Punkte t^2 und t^{2b} führen zu Verbindungstrajektorien T^1 und T^2. Auf den Trajektorien T^1 und T^2 wurden sechs neue Ausgangspunkte entdeckt.*

3. *Ausgehend von den Punkten s^1 und s^2 konnten zwei neue Komponenten K^1 und K^2 gefunden werden. Die anderen Punkte führten nicht zu neuen Komponenten.*

4. *Die Komponente K^1 ist in dem untersuchten Bereich nicht zyklisch, enthält aber fünf weitere Gitterpunkte $t^2, ..., t^7$ sowie einen kritischen Punkt c^3.*

5. *Die Komponente K^2 ist zyklisch und enthält vier neue kritische Punkte $c^4, ..., c^7$ sowie weitere Gitterpunkte, die aber nicht mehr zu neuen Verbindungstrajektorien führen.*

6. *Mit Hilfe der Verbindungstrajektorie T^5 konnte dann auch die letzte Komponente K^3 erreicht werden, die die restlichen kritischen Punkte c^9 und dann c^8 der Zielfunktion enthält.*

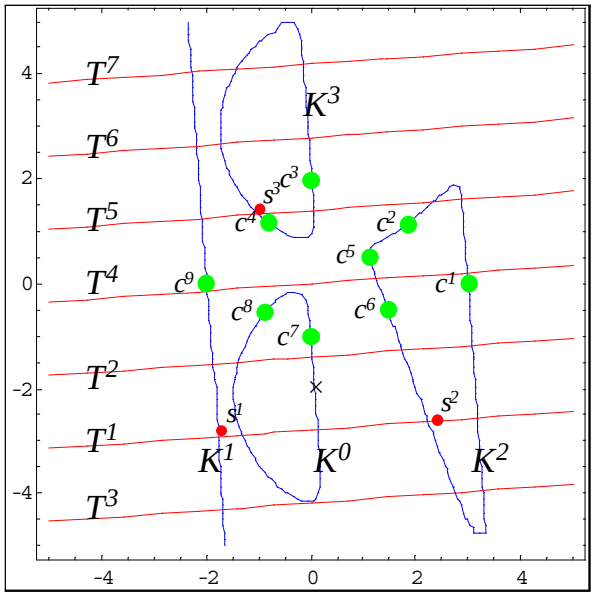

Abbildung 2.4: Trajektoriennetz zur Strategie des äquidistanten Netzes

Bemerkung 100 *Zum Verlauf des Verfahrens und zur dargestellten Trajektorie läßt sich folgendes anmerken:*

a) Die Schnittpunkte der Haupttrajektorie $T(f)$ mit einer Verbindungstrajektorie sind gleichzeitig auch die Gitterpunkte von $T(f)$ bezüglich der Richtung q und der Gitterbreite δ. Die Verbindungstrajektorie kann also von jeder schneidenden Komponente der Trajektorie $T(f)$ gefunden werden.

b) Schneidet eine Verbindungstrajektorie zwei verschiedene Komponenten, so kann immer die eine Komponente von der anderen aus erreicht werden. Ist das Trajektoriennetz zusammenhängend, so kann als Startpunkt für das Rekonstruktionsverfahren ein beliebiger Trajektorienpunkt genommen werden.

2.2.3 Strategie des inäquidistanten Netzes

Werden kritische Punkte in bestimmten Regionen des vorgegebenen Bereiches vermutet, so kann ein geeignetes Raster als endliche Teilmenge der Punkte aus B definiert werden:

$$X = \{x_j \in B | \quad j = 1, 2, ..., J\}.$$

Das Netz der Verbindungstrajektorien wird dann so konstruiert, daß es in den interessanten Regionen dichter und außerhalb gröber ist.

Definition 101 *Ein Trajektorienpunkt $t \in T(f)$ heißt* ***Gitterpunkt von $T(f)$ bezüglich des Rasters*** *X, falls der Wert der Funktion ϕ_q dem Wert in einem der Punkte des Rasters gleich ist:*

$$\exists_{x_j \in X} | \quad \phi_q(t) = \phi(x_j).$$

Das Rekonstruktionsverfahren verläuft hier ähnlich wie im vorherigen Beispiel 99 (s. Abbildung 2.5). Ausgehend von der Komponente K^0 werden zuerst vier Verbindungstrajektorien und dann die Komponenten K^1 und K^2 gefunden. Gitterpunkte für weitere Verbindungstrajektorien werden auf der Komponente K^1 entdeckt. Eine weitere Verbindungstrajektorie führt dann zur Komponente K^3. Die Aussagen der Bemerkung 100 sind auch hier gültig.

Bemerkung 102 *Die Punkte des Rasters X können im allgemeinen nicht als Startpunkte für die Verbindungstrajektorien eingesetzt werden. Nur auf dem niedrigsten Rekursionsniveau $l = 1$ gehören sie zu den Verbindungstrajektorien.*

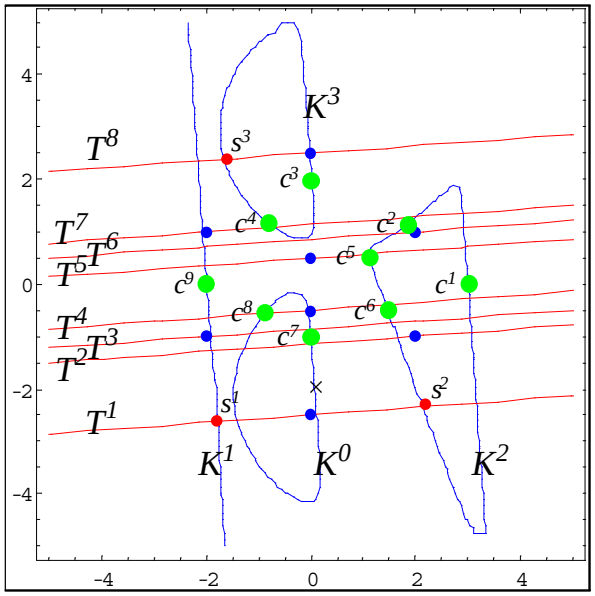

Abbildung 2.5: Trajektoriennetz zur Strategie des inäquidistanten Netzes

2.2.4 Vergleich der Gitterpunktstrategien

In diesem Absatz werden die Vor- und Nachteile verschiedener Gitterpunkt-Strategien diskutiert. Zunächst werden folgende Bezeichnungen eingeführt:

B-Strategie	Strategie der Berührungspunkte
A-Strategie	Strategie des äquidistanten Netzes
C-Strategie	Strategien des inäquidistanten Netzes,
M-Strategie	mixed Strategie (BA oder BC)

1. Zussamenhängigkeit des gesamten Netzes.
   Für die B-Strategie kann unter bestimmten Voraussetzungen bewiesen werden (vgl. [15]), daß das Netz zusammenhängend ist. Dies gilt dann natürlich auch für die M-Strategie. Für die Strategie A und C gibt es keine entsprechenden Aussagen.

2. Erreichbarkeit der Komponenten.
   Wie im Beispiel 96 gezeigt wurde, können nicht immer und von jedem Startpunkt alle Komponenten der Haupttrajektorie erreicht werden. Dies gilt auch dann, wenn das Netz zusammenhängend ist. Im Abschnitt 2.2.2 wurde gezeigt, daß diese Gefahr bei einem dichten A-Netz

bzw. C-Netz zwar unwahrscheinlich gemacht, aber nicht ausgeschlossen werden kann.

3. Schmale (bzgl. einer bestimmten Richtung) Komponenten.
   Die Dichte des B-Netzes kann nicht beeinflußt und auch nicht abgeschätzt werden. Die Position der Berührungspunkte hängt alleine von der Gestalt der Trajektorie ab. Es kann immer passieren, daß zwischen den Verbindungstrajektorien große Bereiche frei bleiben und nicht durchsucht werden.
   Die Dichte eines A- bzw.- C-Netzes kann dagegen frei vorgegeben werden. Trotzdem kann es passieren, daß die Trajektorie in der Ecke des vorgegebenen Bereiches, der durchsucht werden sollte, steckenbleibt.

4. Die Anzahl der Verbindungstrajektorien.
   Da die Gitterbreite in der A-Strategie üblicherweise nicht sehr groß gewählt wird, führt diese Strategie meistens zur größeren Anzahl der Verbindungstrajektorien.
   Bei höherdimensionalen Problemen, insbesondere wenn das Rekursionsschema weit nach unten abgearbeitet werden soll, wächst die Anzahl der Probleme sehr schnell.
   Der Aufwand zur Erstellung eines A-Netzes ist im allgemeinen deutlich höher, als daß bei der B-Strategie der Fall wäre. Im Fall der C-Strategie ist der Aufwand nicht mehr so groß. Auch die M-Strategie mit einem groben A- bzw. C-Netz kann in diesem Bezug empfohlen werden.

5. Das Problem, der nah beieinander liegenden Verbindungstrajektorien.
   Im Fall der B-Strategie passiert es nicht selten, daß verschiedene Berührungspunkte (die vielleicht sogar auf verschiedenen Komponenten der Haupttrajektorie liegen) nah beieinander liegende Verbindungstrajektorien induzieren. Die aufwendige Rekonstruktion solcher Trajektorien führt in meisten Fällen nicht mehr zu neuen Komponenten der Haupttrajektorie.
   Im Fall der B- bzw. C-Strategie sind die Verbindungstrajektorien entweder identisch oder mindestens auf die Gitterbreite δ entfernt. Der zusätzliche Aufwand braucht hier nicht befürchtet werden.

2.3 Graphentheoretische Betrachtung

In diesem Abschnitt wird das rekursiv konstruierte Trajektoriennetz als ein Digraph (gerichteter Graph) dargestellt. Anhand der Beispiele werden Eigenschaften der Graphen untersucht. Anschließend wird das Problem der

Rekonstruktion des Trajektoriennetzes als Problem der Konstruktion eines erzeugendes Baumes dargestellt. Dies wird im nächsten Abschnitt erlauben, geeignete Algorithmen zur Rekonstruktion des Trajektoriennetzes zu finden. Es werden hier einige Begriffe der Graphentheorie verwendet, die aber nicht explizit eingeführt werden. Der Leser wird auf das Handbuch [28] verwiesen. Ist ein Trajektoriennetz gegeben, so wird auf folgende Wiese ein Digraph generiert.

1. Jede Komponente der einzelnen Trajektorien ist eine Graphecke (Graphknoten).

2. Die Komponenten, die zur gleichen Trajektorie gehören, werden in Knotengruppen zusammengefaßt (s. Abbildung 2.6).

3. Die Gruppen sind, abhängig vom Rekursionsniveau der entsprechenden Trajektorien, entweder unter- oder nebeneinander angeordnet.

4. Die nach unten verlaufenden Bögen (gerichteten Kanten) führen vom Knoten a_i^l zum Knoten a_j^{l-1}, wenn auf der entsprechenden Komponente K_i^l ein Gitterpunkt liegt, der als Ausgangspunkt der Komponente K_j^{l-1} benutzt werden kann.

5. Die nach oben verlaufenden Bögen führen vom Knoten a_i^l zum Knoten a_j^{l+1} , wenn die Komponenten K_i^l und K_j^{l-1} sich schneiden. Es besteht dann ein Übergang von der Komponente K_j^{l-1} der Verbindungstrajektorie auf dem Rekursionsniveau $l-1$ zur Komponente K_i^l der ursprünglichen Trajektorie auf dem Rekursionsniveau l.

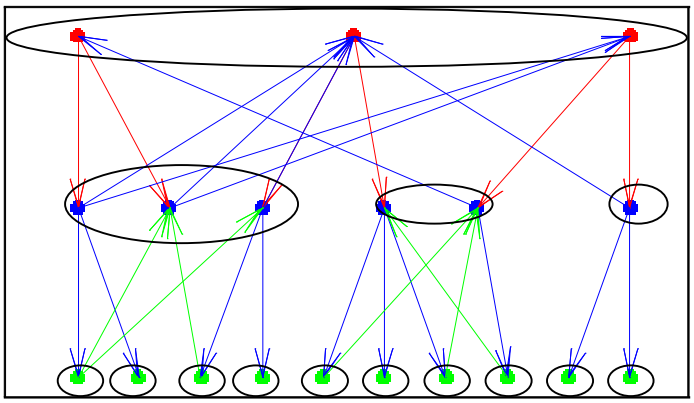

Abbildung 2.6: Digraph des rekursiv konstruierten Trajektoriennetzes

Bemerkung 103 *Es ist zu beachten, daß die auf dem gleichen Rekursionniveau konstruierten Komponenten nicht unbedingt zur gleichen Trajektorie gehören. Jeder Gitterpunkt kann zu einem neuen Hilfsproblem und damit auch zu einer neuen Verbindungstrajektorie führen.*
Die Knoten, die gemeinsame Nachfolger haben, entsprechen den Komponenten einer Trajektorie; die Nachfolgerknoten entsprechen dann Verbidungstrajektorien.
Die anderen Knoten können unter Umständen verschiedenen Trajektorien angehören, auch wenn sie auf dem gleichen Rekursionsniveau liegen.

Strategie der Berührungspunkte

Für das Trajektoriennetz aus dem Beispiel 96 entsteht dann folgender Digraph G_{BP} (s. Abbildung 2.7):

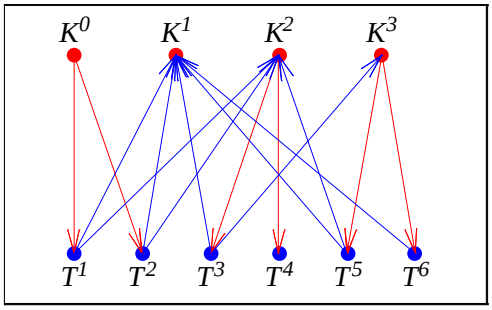

Abbildung 2.7: Digraph G_{BP} für das nach Beispiel der Strategie der Berührungspunkte erzeugte Trajektoriennetz

Die vier Knoten in der oberen Reihe entsprechen den vier Komponenten der Haupttrajektorie. Die sechs Knoten der unteren Reihe stellen die sechs Verbindungstrajektorien dar. Die nach unten verlaufenden Bögen zeigen, wie die Verbindungstrajektorien ausgehend von den Berührungspunkten der entsprechenden Komponenten der Haupttrajektorie konstruiert wurden. Die nach oben verlaufenden Bögen zeigen, welche Komponenten der Haupttrajektorie sich mit entsprechenden Verbindungstrajektorien schneiden.
Ein Verfahren kann das Trajektoriennetz nur übereinstimmend mit dem entsprechenden Digraph rekonstruieren. Ein möglicher Verlauf der Rekonstruktion, ausgehend von der Komponente $K^{\mathbf{0}}$, wurde hier dargestellt. Die Bögen, die zu bereits rekonstruierten Komponenten führen, wurden hier weggelassen.

Auf diese Weise wurde für den Digraphen G_{BP} ein erzeugender Baum $B_{BP}(K^{\mathbf{0}})$ konstruiert (s. Abbildung 2.8). Ein erzeugender Baum enthält

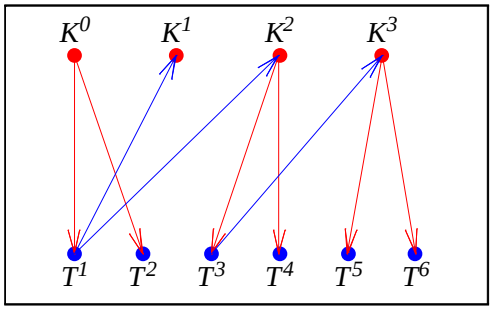

Abbildung 2.8: Erzeugender Baum $B_{BP}(K^0)$ der von der Komponente K^0 mittels der Verbindungstrajektorien erreichbaren Komponenten

alle Knoten des ursprünglichen Graphen und zeigt, daß das rekonstruierte Trajektoriennetz alle Komponenten der gesuchten Trajektorie enthält.

Die Frage, ob für einen gegebenen Digraphen ein erzeugender Baum iterativ (ausgehend von einer gewählten Ecke) konstruierbar ist, ist also der Frage nach der Erreichbarkeit aller Trajektorienkomponenten äquivalent. Um diese Fragestellung zu untersuchen, werden hier weitere graphentheoretische Begriffe eingeführt.

Definition 104 *Wenn wir jeden Bögen eines Digraphen durch eine ungerichtete Kante ersetzen, so erhalten wir den* ***zugehörigen Mutligraphen****. Ein Digraph heißt* ***zusammenhängend****, wenn der zugehörige Multigraph zusammenhängend ist. Ein Digraph heißt* ***stark zusammenhängend****, wenn jeder Knoten von jedem anderen aus in Übereinstimmung mit der Orientierung der Kanten erreichbar ist.*

Es kann leicht geprüft werden, daß der Digraph G_{BP} zwar zusammenhängend, aber nicht stark zusammenhängend ist. Der Knoten K^1 kann zwar von jeden anderen Graphecke aus erreicht werden, der Weg von dem Knoten aus ist aber nicht möglich (s. Abbildung 2.9'). Die anderen Knoten werden nicht erreicht.

Ausgehend von einem der Knoten K^2 oder K^3 kann der Knoten K^0 nicht erreicht werden. Die konstruierten Bäume $B_{BP}(K^2)$ (s. Abbildung 2.10) und $B_{BP}(K^3)$ (s. Abbildung 2.11) enthalten nicht alle Knoten des Digraphen G_{BP}.

Bemerkung 105 *Im Fall der Richtungsfeld-Trajektorie mit mehreren Startpunkten (vgl. Abschnitt 1.4.3) stehen mehrere Ausgangspunkte zur Verfügung. Das rekonstruierte Trajektoriennetz entspricht dann der Vereinigung*

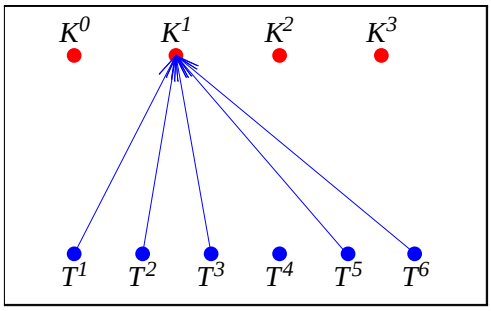

Abbildung 2.9: Die Komponente K^1 wird durch mehrere Verbindungstrajektorien geschnitten, besitzt aber keine Berührpunkte.

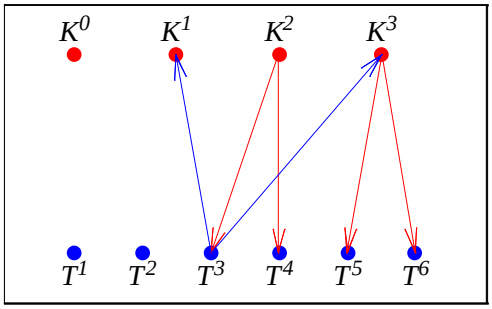

Abbildung 2.10: Baum $B_{BP}(K^2)$ der von der Komponente K^2 mittels der Verbindungstrajektorien erreichbaren Komponenten.

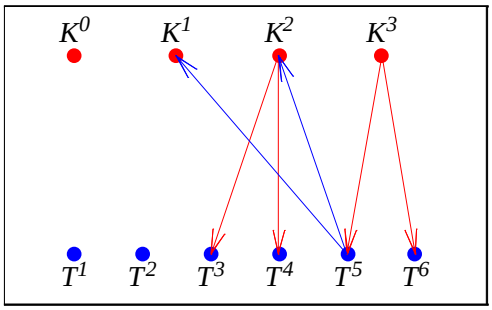

Abbildung 2.11: Baum $B_{BP}(K^3)$ der von der Komponente K_3 mittels der Verbindungstrajektorien erreichbaren Komponenten.

der Netze, die ausgehend von einzelnen Punkten rekonstruiert werden. Durch geeignete Wahl der Ausgangspunkte kann zwar gesichert werden, daß die rekonstruierte Trajektorie bestimmte Regionen und damit wahrscheinlich auch dort plazierte kritische Punkte erreicht. Dies impliziert allerdings nicht, daß wirklich alle Komponenten der Trajektorie und alle kritischen Punkte der Zielfunktion gefunden werden.

Strategie des äquidistanten bzw. inäquidistanten Netzes

Es wurde bereits angesprochen (vgl. Bemerkung 100), daß die Gitterpunkte eines äquidistanten bzw. inäquidistanten Netzes gleichzeitig auch die Schnittpunkte der Trajektorie mit den entsprechenden Verbindungstrajektorien sind. Jede Verbindungstrajektorie kann also von beliebigen sie schneidenden Komponenten erreichbar werden. Der Digraph G_{AN} des Trajektoriennetzes aus dem Beispiel 99 kann deshalb durch einen zugehörigen ungerichteten Multigraphen M_{AN} (s. Abbildung 2.12) ersetzt werden.

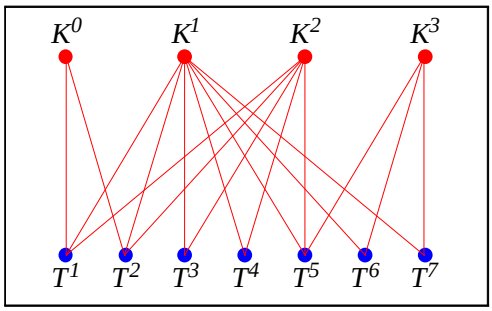

Abbildung 2.12: Ungerichteter Graph für das nach Beispiel der Strategie des äquidistanten Netzes erzeugtes Trajektoriennetz

Die Begriffe **zusammenhängend** und **stark zusammenhängend** sind dann äquivalent, so daß der erzeugender Baum von jedem Knoten aus konstruiert werden kann!

Abschließende Bemerkungen

Die graphentheoretische Betrachtunsweise ermöglichte eine klare Vorstellung der Abläufe der rekursiven Konstruktion des Trajektoriennetzes. Es ist dabei ersichtlich geworden, wann und warum das Trajektoriennetz nicht rekonstruiert werden kann. Es konnte untersucht werden, welche Rolle dabei die Strategie der Gitterpunkte spielt.

Anderseits aber gibt diese Betrachtungsweise auch Aufschlüsse über die Struktur des Problems. Die aus der Graphentheorie bekannten Datenstrukturen, wie Inzidenzlisten, Adjazenzlisten oder Adjazenzmatrizen [28], können dann leicht übertragen werden.
Die effizienten Algorithmen der Graphentheorie, wie BFS (breadth first search) oder DFS (depth first search) können angewandt werden.

2.4 Rekursionsalgorithmus

In diesem Abschnitt werden zwei mögliche Algorithmen zur Rekonstruktion des Trajektoriennetzes beschrieben. Die Algorithmen werden mittels der Modellierungssprache UML (unified modelling language) objektorientiert dargestellt (vgl. [4][22]). Hierzu werden die wichtigen Klassen Komponente, Punktliste und Schicht skizziert. Die Beziehungen (Assoziationen und Aggregationen) zwischen den Klassen werden in Klassendiagrammen graphisch dargestellt. Das entsprechende Hauptprogramm wird anschließend ausführlich beschrieben.

2.4.1 BFS Algorithmus (breadth first search)

Auf der Basis des *breadth first search* Algorithmus zur Bestimmung kürzester Wege in einem Graph wird hier ein Verfahren zur Rekonstruktion des Trajektoriennetzes aufgebaut. Abhängig vom Rekursionsniveau der Trajektorie werden deren Komponenten entsprechenden Schichten zugeordnet. Die Schichten, die Listen von Ausgangspunkten der Trajektorienkomponenten enthalten, werden dann nach der absteigenden Ordnung bearbeitet.

Klassen

Zunächst werden Objekte der Klasse **Punkt** als Vektoren der vorgegebenen Länge n definiert und entsprechende Operationen (Addition, Multiplikation mit einem Skalar, Skalarprodukt usw. implementiert).
Die Klasse **PunktListe** stellt dann für ein n-dimensionales Problem verkettete Listen von n-dimensionalen Punkten zur Verfügung. Außer der Abfrage, ob die Liste leer ist, stehen noch die Operation der Rückgabe *getPunkt()*, das Löschen *delPunkt()* und Hinzufügen des ersten Elements *addPunkt(*Punkt*)* sowie das Hinzufügen einer anderen Liste *addList*(PunktListe) zur Verfü-

gung.

PunktListe	
Punkt	x
PunktListe	*Rest*;
	PunktListe();
Boolean	*IstLeer()*;
Punkt	*getPunkt*();
	delPunkt();
	addPunkt(Punkt);
	addList(PunktListe);

Die Objekte der Klasse **Komponente** sind die Bausteine des Rekonstruktionsverfahrens. Identifiziert durch den Ausgangspunkt, entsprechen sie den Knoten des Rekursionsgraphen. Der Attribut *Rekursionsniveau* zeigt, wo sich das Objekt in der Schichtstruktur des Graphen befindet, und die Listen *Rauf* und *Runter* entsprechen den Adjazenzlisten mit den Nachbarn, die auf den höher oder niedriger gesetzten Schicht liegen. Mit Hilfe der Methode *IstNeu(PunktListe)*; kann anhand der Liste *KontrollPunkte*; geprüft werden, ob die Komponente bereits rekonstruiert wurde.

Komponente	
Punkt	*Ausgangspunkt*;
int	*Rekursionsniveau*;
PunktListe	*Rauf*;
PunktListe	*Runter*;
	Komponente(Punkt]);
	~*Komponente*();
Boolean	*IstNeu(PunktListe)*;
PunktListe	*getRauf*();
PunktListe	*getRunter*();

In einem Objekt der Klasse **Schicht** werden die Komponentenobjekte zusammengefaßt, die dem gleichen Rekursionsniveau entsprechen. Die PunktListe *KontrollPunkte* enthält die Ausgangpunkte der Komponenten, die bereits bearbeitet wurden. In der Liste *StartPunkte* werden die Ausgangspunk-

te gespeichert, die noch nicht bearbeitet wurden.

Schicht	
int	*Rekursionsniveau*;
PunktListe	*KontrollPunkte*;
PunktListe	*StartPunkte*;
	Schicht();
	Reset();
Boolean	*IstAbgearbeitet()*;
PunktListe	*getKontrollPunkte*();
	addKontrollPunkt(Punkt)
Punkt	*getStartPunkt()*;
	addStartPunkt();
	delStartPunkt()

Die Assoziationen und Aggregationen der definierten Klassen wurden mit UML-üblicher Notation in der Abbildung 2.13 graphisch dargestellt.

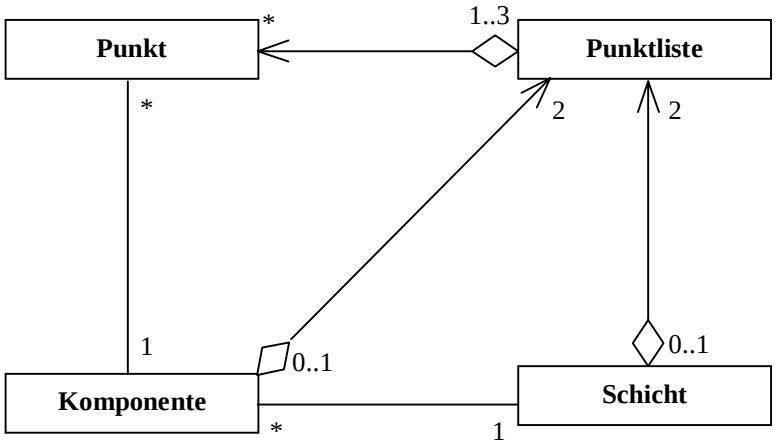

Abbildung 2.13: Klassendiagramm mit der UML-Notation

Assoziationen:

Komponente/Punkt

Von einem Punkt wird genau eine Komponente rekonstruiert.
Jede Komponente kann von mehreren Punkten aus erreicht werden.

Komponente/Schicht
Einer Schicht können mehrere Komponente zugeordnet werden.
Jede Komponente liegt in einer bestimmten Schicht.

Aggregationen (Teil-von-Beziehung)**:**

Punkt/Punktliste
Punktliste kann mehrere Punkte enthalten.
Ein Punkt kann zu einer bis drei Listen gehören:
Schicht→*StartPunkte* oder *KontrollPunkte*
Komponente→*rauf* oder *runter*
und auch zu der Liste *KritischePunkte*.

Punktliste/Komponente
Jede Komponente enthält zwei Punktlisten.
Jede Punktliste gehört zu höchstens einer Komponente.

Punktliste/Schicht
Jede Schicht enthält zwei Punktlisten.
Jede Punktliste gehört zu höchstens einer Schicht.

Algorithmus

Mit der gegebenen Dimension des Problems n kann der Verlauf des Algorithmus wie folgt beschrieben werden.

(1-3) Zu Anfang wird eine PunktListe *KritischePunkte* sowie n Objekte $S[i]$, $i = 0, ..., n-1$ der Klasse **Schicht** initialisiert. In die Liste *StartPunkte* des Objektes $S[n-1]$ wird ein (bzw. mehrere) vorgegebener *Punkt* x_0 aufgenommen.

(4) Der Ablauf des Verfahrens wird dann mit Hilfe der Niveauzahl l gesteuert. Diese Zahl zeigt auf die höchstgelegene Schicht $S[l]$, die zu diesem Zeitpunkt eine nichtleere Liste *StartPunkte* enthält.

(5-18) Nach und nach werden alle *StartPunkte*-Listen bearbeitet.

(6-9) Für jeden neuen Punkt der Listen wird eine Komponente konstruiert und der Ausgangspunkt aus der Liste *StartPunkte* entfernt und der Liste *KontrollPunkte* hinzugefügt.

(10-16) Es wird kontrolliert, ob die Komponente bereits bearbeitet wurde. Wenn dies nicht der Fall ist, werden die Listen *Rauf* und *Runter* bei entsprechenden Schichten den *StartPunkte* Listen hinzugefügt.

(17-18) Anschließend wird die Komponente wieder gelöscht und der neue Wert der Niveauzahl l vor dem neuen Durchlauf ermittelt. Der Vorgang wird solange wiederholt bis die Niveauzahl l das minimale Rekursionsniveau unterschreitet.

Algorithm 106 *BFS-Rekonstruktion des Trajektoriennetzes*

```
    Initialisierung
1     PunktListe      KritischePunkte;
2     Schicht      S[]=new Schicht[n];
3     S[n].addStartPunkt(x0);
4     int      l=n;
5   while (l ≥ 0)
6     Punkt      x=S[l].getStartPunkt();
7     Komponente      K=Komponente(x);
8     S[l].delStartPunkt();
9     S[l].addKontrollPunkt(x);
10    PunktListe      KontrollPunkte=S[l].getKontrollPunkte();
11    if (K.IstNeu(KontrollPunkte))
12    then  if (l > 0)
13          then  S[l − 1].addStartPunkte(K.getRunter());
14          if (l < n − 1)
15          then  S[+ + l].addStartPunkte(K.getRauf());
16          else  KritischePunkte.addList(K.getRauf());
17    K::~Komponente();
18    while (l ≥ 0   ||   S[l].IstAbgearbeitet()) l − −;
```

2.4.2 DFS Algorithmus (depth first search)

Während der BFS Algorithmus den Rekursionsgraphen "der Breite nach" rekonstruiert, geht der "depth first search" Algorithmus jeweils so weit wie möglich in den Graphen hinein also mit dem Rekursionsniveau nach unten. Wurde das minimale Rekursionsniveau erreicht oder wurden keine Gitterpunkte mehr gefunden, so können keine Verbindungstrajektorien konstruiert werden. Der Weg in die Tiefe des Graphen ist nicht mehr möglich. In diesem Fall kommt man zum Mutterknoten zurück und versucht andere Zweige abzufahren. Wurde ein Zweig vollständig durchsucht, so werden die gesammelten Informationen nicht mehr gebraucht und können aus dem Speicher der Klasse **Schicht** entfernt werden. Die Objekte dieser Klasse entsprechen dann also den Verbindungstrajektorien.

Algorithmus

Mit der gegebenen Dimension des Problems n kann der Verlauf des Algorithmus wie folgt beschrieben werden.

(1-3) Zur Anfang wird eine PunktListe *KritischePunkte* sowie n Objekte $T[i]$, $i = 0, ..., n-1$ der Klasse **Schicht** initialisiert. In die Liste *StartPunkte* der Haupttrajektorie $T[n-1]$ wird ein (bzw. mehrere) vorgegebener Punkt x_0 aufgenommen.

(4) Der Ablauf des Verfahrens wird dann mit Hilfe der Niveauzahl l gesteuert. Die Zahl zeigt auf die in der Rekursionshierarchie niedrigstgelegene Trajektorie $T[l]$, die zum diesen Zeitpunkt eine nichtleere Liste *StartPunkte* enthält.

(6-9) Der erste Punkt wird dann aus der Liste *StartPunkte* gelesen, entfernt und der Liste *KontrollPunkte* hinzugefügt.

(10-16) Führt dieser Punkt zu einer neuen Komponente der Trajektorie $T[l]$, so werden die Listen *Rauf* und *Runter* der entsprechenden Trajektorien $T[l+1]$ und $T[l-1]$ hinzugefügt.

(17-18) Anschließend wird die Komponente wieder gelöscht und der neue Wert der Niveauzahl l vor dem neuen Durchlauf ermittelt. Wurde eine Verbindungstrajektorie $T[l]$ abgearbeitet (d.h. die Liste $T[l].StartPunkte$ ist leer), so wird die nicht mehr gebraucht und das korrespondierende Objekt kann zurückgesetzt werden.

(5-18) Der Vorgang wird solange wiederholt bis die Liste $T[n-1].StartPunkte$ der Haupttrajektorie $T[n-1]$ abgearbeitet ist.

Algorithm 107 *DFS-Rekonstruktion des Trajektoriennetzes*

```
    Initialisierung
1     PunktListe      KritischePunkte;
2     Schicht      T[]=new Schicht[n];
3     T[n].addStartPunkt(x_0);
4     int      l=n;
5   do
6     Punkt      x=T[l].getStartPunkt();
7     Komponente      K=Komponente(x);
8     T[l].delStartPunkt();
9     T[l].addKontrollPunkt(x);
10    PunktListe      KontrollPunkte=T[l].getKontrollPunkte();
11    if (K.IstNeu(KontrollPunkte))
12    then  if (l < n)
13          then  T[l + 1].addStartPunkte(K.getRauf());
14          else  KritischePunkte.addList(K.getRauf());
15          if (l > 1)
16          then  T[− − l].addStartPunkte(K.getRunter());
17    K::~Komponente();
18    while (T[l].IstAbgearbeitet()  ||  l ≤ n) T[l + +].Reset();
    while (l ≤ n)
```

2.4.3 Vergleich der vorgestellten Algorithmen

Die Komplexität der Algorithmen ist im wesentlichen durch die aufwendige Rekonstruktion der einzelnen Trajektorienkomponenten bestimmt. Wird das gesamte Trajektoriennetz rekonstruiert, so ist die Komplexität der beiden Algorithmen vergleichbar. Die Vorteile des DFS-Algorithmus liegen dann in folgenden Eigenschaften:

1. Durch die Berücksichtigung der Zusammnehänge zwischen den einzelnen Komponenten der Verbindungstrajektorien werden nur die zur einem bestimmten Zeitpunkt relevanten Informationen gespeichert. Die Speicherkapazität wird hierdurch dynamisch optimal verwaltet.

2. Die neukonstruierten Komponenten einer Verbindungstrajektorie werden lediglich mit den anderen bereits konstruierten Komponenten derselben Trajektorie verglichen und nicht wie beim BFS-Algorithmus mit allen Komponenten der auf dem entsprechenden Rekursionsniveau liegenden Trajektorien. Das DFS-Verfahren wird dadurch etwas schneller.

In einigen Fällen kann aber das BFS-Verfahren günstiger sein, da das Rekursionsniveau dabei möglichst hoch gehalten wird. Das BFS-Verfahren führt deshalb im allgemeinen schneller zu gesuchten kritischen Punkten. Wird die Rekonstruktion des Trajektoriennetzes abgebrochen, so ist die Anzahl der gefundenen kritischen Punkten im Vergleich zum DFS-Verfahren oft größer. Ist die Anzahl der vorhandenen kritischen Punkte bekannt, so kann dann das BFS-Verfahren abgebrochen werden, noch bevor das gesamte Trajektoriennetz rekonstruiert wurde.

Kapitel 3

Ausgewählte Verfahren zur Trajektorienverfolgung

In diesem Abschnitt werden Verfahren vorgestellt, mit deren Hilfe eine zusammenhängende Trajektorienkomponente einer verallgemeinerten Newton-Trajektorie $T(f)$,

$$T(f) := \{x \in \mathbb{R}^n | \quad H(x) = 0\},$$

rekonstruiert werden kann (vgl. Definition 10). Die Hilfsfunktion $H : \mathbb{R}^n \rightarrow \mathbb{R}^{n-1}$ sei dabei stetig differenzierbar.
Die Aufgabe ist, einer Komponente der Trajektorie, ausgehend von einem Punkt x_0, so lange zu folgen, bis entweder der Rand des gegebenen Bereiches B, der alle kritischen Punkte enthalten soll, erreicht oder ein Zyklus entdeckt wird. Im ersten Fall wird der Kurve in anderer Richtung gefolgt; danach und im zweiten Fall wird ein Startpunkt für eine neue Komponente gesucht. Zur Verfolgung der Trajektorie wurde hier der Prädiktor-Korrektor Ansatz gewählt.

3.1 Prinzip des Prädiktor-Korrektor-Verfahrens

Seien $x_0, x_1, ..., x_i$, die im Laufe des Verfahrens gefundenen Punkte einer Trajektorienkomponente. Mit Hilfe der gewählten Prädiktormethode wird dann ein Prädiktorpunkt $x_{i+1,0}$ als Näherungswert eines weiteren auf der Kurve liegenden Punktes x_{i+1} bestimmt. Mit Hilfe einer geeigneten Korrektormethode kann dann der Punkt x_{i+1} mit vorgegebenen Genauigkeit berechnet werden. Die Konvergenz des Korrektorsverfahrens wird durch Schrittweitensteuerung gesichert.

3.1.1 Prädiktormethode

Als Prädiktormethode kann die Euler-Methode zur numerischen Integration der Differentialgleichung (1.12)

$$\dot{x}(\alpha) = t(DH(x(\alpha))) \qquad | \qquad x(0) = x^0.$$

genommen werden. Ein Prädiktorpunkt ist dann iterativ gegeben durch:

$$x_{i+1,0} := x_i + p \cdot t(DH(x_i)),$$

wobei mit p die Schrittweite des Prädiktorschrittes bezeichnet ist. Für die $n-1 \times n$ Jacobi-Matrix $DH(x_i)$ kann der Tangentenvektor $t(DH(x_i))$ (s. Definition 15) leicht aus der QR-Zerlegung der transponierten Jacobi-Matrix gewonnen werden.

Lemma 108 *Sei eine QR-Zerlegung der transponierten Jacobi-Matrix $DH(x_i)^T$ in einem regulären Punkt x_i der Funktion $H(x)$ gegeben:*

$$DH(x_i)^T = Q_i \begin{pmatrix} R_i \\ 0 \end{pmatrix}^T.$$

Dabei ist $Q_i := (q_{i,1}, ..., q_{i,n})$ eine $n \times n$ orthogonale Matrix $\left(Q_i^T Q_i = I\right)$ und R eine $n-1 \times n-1$ obere Dreiecksmatrix mit den Diagonalelementen $r_{i,i}$. Für den Tangentenvektor $t(DH(x_i))$ der Matrix $DH(x_i)$ folgt dann:

$$\begin{aligned} t(DH(x_i)) &= \sigma q_{i,n}, \\ \sigma &= \det Q^T \operatorname{sign}\left(\prod_{i=1}^{n-1} r_{i,i}\right). \end{aligned}$$

Beweis. Jede Spalte der Matrix $DH(x_i)$ kann als lineare Kombination der Spalten $q_{i,1}, ..., q_{i,n-1}$ der orthogonalen Matrix Q dargestellt werden. Für die letzte Spalte $q_{i,n}$ der Matrix Q gilt also:

$$DH(x_i)^T q_{i,n} = 0.$$

Auf Grund der Regularität des Punktes x_i ist der Rang der Jacobi-Matrix $DH(x_i)$ gleich $n-1$, so daß die Vektoren im Kern der Jacobi-Matrix alle linear abhängig sind. Der Tangentenvektor $t(DH(x_i))$ ist also laut Definition 15 bis auf die Orientierung σ gleich der letzten Spalte der Matrix Q:

$$t(DH(x_i)) = \sigma q_{i,n}, \qquad \sigma = \pm 1.$$

Die Orientierung σ läßt sich dann wie folgt bestimmen:

$$\begin{aligned}\sigma &= \operatorname{sign}\left(\det\begin{pmatrix} DH(x_i) \\ q_{i,n} \end{pmatrix}\right) = \det Q^T \operatorname{sign}\left(\det\begin{pmatrix} R^T, 0 \\ e_n^T \end{pmatrix}\right) \\ &= \det Q^T \operatorname{sign}\left(\prod_{i=1}^{n-1} r_{i,i}\right).\end{aligned}$$

■

Bemerkung 109 *Für die Determinante der orthogonalen Matrix Q gilt:*

$$\det Q^T = \pm 1.$$

Das Vorzeichen der Determinante kann bei der Wahl des Orthogonalisierungsverfahren auf $+1$ (Givensrotationen) oder $(-1)^{c(n)}$ mit einer dimensionsabhängigen Konstante $c(n)$ (Householder-Spiegelungen) festgelegt werden.

3.1.2 Korrektormethode

Im Korrektorschritt wird ein Trajektorienpunkt x_{i+1} gesucht, dessen Abstand zum Prädiktorpunkt $x_{i+1,0}$ möglichst klein ist. Es wird also eine Lösung des folgenden Minimierungsproblems gesucht:

$$\min_{x \in T(f)} \|x_{i+1,0} - x\|.$$

Die notwendige Bedingung für eine Lösung x^* des Problems lautet:

$$\begin{aligned} H(x^*) &= 0 \\ x^* - x_{i+1,0} &= DH(x^*)^T \lambda, \qquad \lambda \in \mathbb{R}^{n-1}. \end{aligned} \tag{3.1}$$

Die Bedingung (3.1) sagt, daß der Differenzenvektor $x^* - x_{i+1,0}$ als lineare Kombination der Zeilen der Matrix $DH(x^*)$ dargestellt werden kann und damit orthogonal zum Tangentenvektor $t(DH(x^*))$ liegen muß (s. Abbildung 3.1).

Die Bedingung (3.1) kann also durch folgende Gleichung ersetzt werden:

$$t(DH(x^*))^T (x^* - x_{i+1,0}) = 0.$$

Der Trajektorienpunkt x_{i+1} ist als die Lösung des folgenden nichtlinearen Gleichungssystems gegeben:

$$\begin{cases} H(x) = 0 \\ t(DH(x))^T (x - x_{i+1,0}) = 0. \end{cases} \tag{3.2}$$

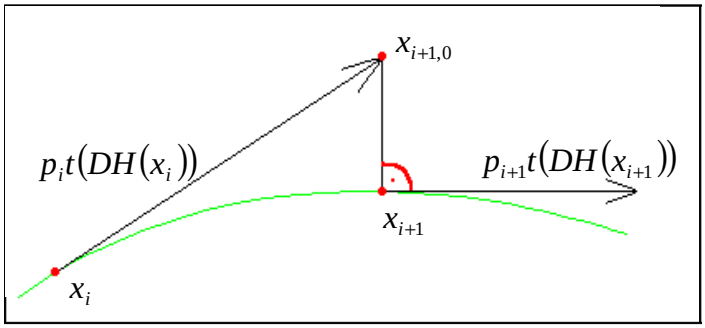

Abbildung 3.1: Prädiktor-Korrektor-Schritt

Zur Lösung des Gleichungssystems wird in den meisten Fällen das Newton-Verfahren oder Quasi-Newton-Verfahren eingesetzt. Eine neue Möglichkeit für die "analytisch schwierigen" Zielfunktionen bietet das Ableitungsfreie Surrogate-Verfahren. Die angegebenen Verfahren basieren alle auf dem Prädiktor-Korrektor Schema und werden im weiteren näher beschrieben.
Die Verfahren lassen sich ohne Einschränkung sowohl zur Rekonstruktion der klassischen Newton-Trajektorie als auch der eingeführten Richtungsfeld-Trajektorien einsetzten.

3.2 Kontinuerliches Newton-Verfahren (KNV)

Das kontinuierliche Newton-Verfahren ist die übliche Realisierung des Prädiktor-Korrektor Schema zur Rekonstruktion der Newton-Trajektorie. Die Jacobi-Matrix $DH(x)$ wird hierbei in jedem Schritt explizit analytisch oder numerisch berechnet. Nachdem der Tangentenvektor $t(DH(x))$ und die Schrittlänge p für den Prädiktorschritt bestimmt wurden, kann der Prädiktorpunkt $x_{i+1,0}$ mittels des Korrektorverfahrens an die Trajektorie projiziert werden.

3.2.1 Korrektor

Sei mit $\hat{H} : \mathbb{R}^n \to \mathbb{R}^n$ die linke Seite des Gleichungssystems (3.2) bezeichnet:

$$\hat{H}(x) := \begin{cases} H(x) \\ t(DH(x))^T (x - x_{i+1,0}) . \end{cases}$$

Der erste Schritt des Newton-Verfahrens zur Lösung des Gleichungssystems $\hat{H}(x) = 0$ ist dann gegeben durch:

$$\begin{aligned} x_{i+1,1} &= x_{i+1,0} - D\hat{H}(x_{i+1,0})^{-1} \hat{H}(x_{i+1,0}) \\ &= x_{i+1,0} - \begin{pmatrix} DH(x_{i+1,0}) \\ t(DH(x_{i+1,0}))^T \end{pmatrix}^{-1} \begin{pmatrix} H(x_{i+1,0}) \\ 0 \end{pmatrix}. \end{aligned} \tag{3.3}$$

Die Vorschriftsformel (3.3) läßt sich mit Hilfe der Pseudoinverse der Jacobi-Matrix $DH(x_{i+1,0})$ vereinfachen.

Definition 110 *(Moore,Penrose)*
Sei A eine reguläre $m \times n$ Matrix, dann heißt die $n \times m$ Matrix A^+,

$$A^+ := A^T \left(AA^T\right)^{-1},$$

*die **Pseudoinverse** von A.*

Lemma 111 *Für eine reguläre $n-1 \times n$ Matrix A mit der Pseudoinverse A^+ gilt:*

$$\begin{pmatrix} A \\ t(A)^T \end{pmatrix}^{-1} = \left(A^+, t(A)\right).$$

Beweis. Aus der Definition des Tangentenvektors $t(A)$ folgt:

$$\begin{aligned} At(A) &= 0 \\ t(A)^T t(A) &= 1. \end{aligned}$$

Aus der Definition der Pseudoinverse A^+ folgt weiterhin:

$$\begin{aligned} AA^+ &= AA^T \left(AA^T\right)^{-1} = I_{n-1} \\ t(A)^T A^+ &= t(A)^T A^T \left(AA^T\right)^{-1} = \left(At(A)\right)^T \left(AA^T\right)^{-1} = 0 \end{aligned}$$

Daraus ergibt sich dann:

$$\begin{pmatrix} A \\ t(A)^T \end{pmatrix} \left(A^+, t(A)\right) = \begin{pmatrix} AA^+ & At(A) \\ t(A)^T A^+ & t(A)^T t(A) \end{pmatrix} = I_n.$$

■

Der erste Schritt des Newton-Verfahrens zur Lösung des Gleichungssystems $\hat{H}(v) = 0$ lautet also:

$$\begin{aligned} x_{i+1,1} &= x_{i+1,0} - \left(DH(x_{i+1,0})^+, t(DH(x_{i+1,0}))\right) \begin{pmatrix} H(x_{i+1,0}) \\ 0 \end{pmatrix} \\ &= x_{i+1,0} - DH(x_{i+1,0})^+ H(x_{i+1,0}). \end{aligned}$$

Dies kann zum Anlaß für ein Korrektorverfahren genommen werden. Entsprechend durch:

$$x_{i+1,j+1} = x_{i+1,j} - DH\left(x_{i+1,j}\right)^{+} H\left(x_{i+1,j}\right)$$

ist das bekannte Newton-Verfahren zur Lösung des unterbestimmten nichtlinearen Gleichungssystems $H(x) = 0$ beschrieben. Das Verfahren liefert die Lösung von $H(x) = 0$, die dem Prädiktorpunkt $x_{i+1,0}$ am Nächsten liegt. Dieses Verfahren stimmt mit dem üblichen Newton-Verfahren zur Lösung der Gleichungssysteme überein, indem die Inverse der Jacobi-Matrix durch die Pseudoinverse ersetzt wird. Für das Verfahren kann die lokale quadratische Konvergenz bewiesen werden [30].

Mittels der QR-Zerlegung der transponierten Jacobi-Matrix $DH(x)^T$ läßt sich die Pseudoinverse $DH(x)^{+}$ leicht bestimmen.

Lemma 112 *Sei mit einer orthogonalen $n \times n$ Matrix Q und einer oberen $n-1 \times n-1$ Dreiecksmatrix R*

$$DH(x)^T = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

dann gilt:

$$DH(x)^{+} = Q \begin{pmatrix} \left(R^T\right)^{-1} \\ 0 \end{pmatrix}.$$

Beweis. Aus der Definition der Pseudoinverse folgt direkt:

$$\begin{aligned} DH(x)^{+} &= DH(x)^T \left(DH(x)\, DH(x)^T\right)^{-1} \\ &= Q \begin{pmatrix} R \\ 0 \end{pmatrix} \left(\left(R^T, 0\right) Q^T Q \begin{pmatrix} R \\ 0 \end{pmatrix}\right)^{-1} \\ &= Q \begin{pmatrix} R \\ 0 \end{pmatrix} \left(R^T R\right)^{-1} = Q \begin{pmatrix} RR^{-1}\left(R^T\right)^{-1} \\ 0 \end{pmatrix} \\ &= Q \begin{pmatrix} \left(R^T\right)^{-1} \\ 0 \end{pmatrix}. \end{aligned}$$

■

Der Korrektorschritt $k_{i+1,j} = -DH\left(x_{i+1,j}\right)^{+} H\left(x_{i+1,j}\right)$ kann dann auf folgende Weise berechnet werden:

1. Durch Vorwärtssubstitution wird zunächst die Lösung y^* des Gleichungssystems $R^T y = H(x)$ berechnet.

2. Der Korrektorschritt wird dann durch orthogonale Projektion bestimmt:

$$k_{i+1,j} = Q \begin{pmatrix} y^* \\ 0 \end{pmatrix}.$$

3.2.2 Schrittlängensteuerung

Die Steuerung der Prädiktorschrittweite p ist sehr wichtig für die effiziente Trajektorienrekonstruktion. Es wird deshalb versucht, die Schrittlänge p möglichst groß zu halten. Um aber alle Krümmungen der gesuchten Trajektorie $T(f)$ richtig wiederherzustellen und die Konvergenz des Korrektorverfahrens zu sichern, muß der berechnete Prädiktorpunkt stets genügend nah der Trajektorie liegen. Mit Hilfe von verschiedenen Tests kann dies kontrolliert werden.

Residualtest

In erster Linie wird die Entfernung des Prädiktorpunktes von der Trajektorie geprüft. Eine gute Approximation dieser Entfernung ist die Länge des ersten Korrektorschrittes $\|k_{i,1}\|$. Ist also der erste Korrektorschritt zu groß:

$$\|k_{i,j}\| > k_{\max},$$

so wird der Prädiktorschritt nicht akzeptiert. Die Schrittlänge p wird halbiert (oder in einem anderen Verhältnis verkürzt $p_{neu} = \alpha p$, $\alpha < 1$) und der Prädiktorschritt wiederholt.

Kontraktionstest

Die Konvergenz des Newton-Verfahrens ist nur lokal (d.h. in der Nähe der Trajektorie) gesichert. Es ist deswegen erforderlich, das Konvergenzverhalten des Korrektorverfahrens in jedem Schritt zu prüfen. Hierzu wird die Kontraktionsrate $\varkappa(k_{i,j}, k_{i,j+1})$ zweier nacheinander folgender Korrektorschritte getestet

$$\varkappa(k_{i,j}, k_{i,j+1}) = \frac{\|k_{i,j}\|}{\|k_{i,j+1}\|}.$$

Stellt man keine ausreichende Konvergenz des Korrektorverfahren fest:

$$\varkappa(k_{i,j}, k_{i,j+1}) > \varkappa_{\max},$$

so liegt der Prädiktorschritt wahrscheinlich zu weit weg von der Trajektorie. Das Korrektorverfahren wird abgebrochen. Der Prädiktorschritt muß verkürzt wiederholt werden.

Winkeltest

Es ist anschaulich klar, daß bei einer sehr krümmenden Trajektorie die Prädiktorschrittlänge kleiner gewählt werden muß, um all ihre Krümmungen zu rekonstruieren. Vor einem neuen Prädiktorschritt wird deshalb immer die Richtung des neuen und des alten Tangentenvektors miteinander verglichen (s. Abbildung 3.2). Ist der Winkel zwischen den beiden Richtungen zu groß:

$$t\left(DH\left(x_i\right)\right)^T t\left(DH\left(x_{i+1}\right)\right) < \cos\left(\gamma_{\max}\right),$$

wird der letzte Schritt nicht akzeptiert und muß komplett mit kleineren p wiederholt werden.

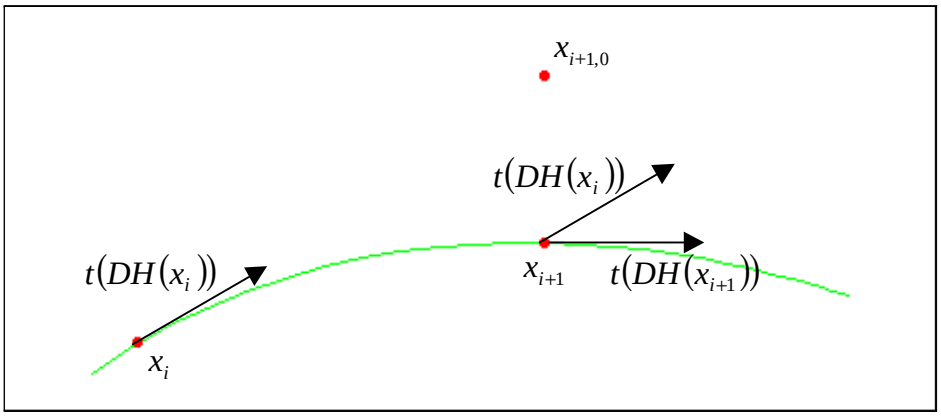

Abbildung 3.2: Winkeltest bei der Rekonstruktion einer Trajektorie

In diese Weise können auch bei der Trajektorienrekonstruktion Sprünge zwischen zwei nah beieinander liegenden Trajektorienkomponenten (bzw. zwei verschiedenen Zweigen einer Komponente) vermieden werden (s. Abbildung 3.3).

Zusammenfassung

Wurde der Schritt in allen Tests akzeptiert, so kann die Schrittweite p womöglich größer gewählt werden ($p_{neu} := \beta p$, $\beta > 1$). Die neue Schrittweite wird aber erst im nächsten Schritt eingesetzt. Der akzeptierte Schritt wird nicht wiederholt.
Sollte im Laufe des Verfahrens die Schrittweite zu kurz werden ($p < p_{\min}$), so muß die Rekonstruktion der Trajektorienkomponente abgebrochen werden. Die Rekonstruktion wird von einem weiteren Ausgangspunkt mit einer neuen Komponente fortgesetzt (s. Kapitel 2). Wird das Verfahren vorzeitig

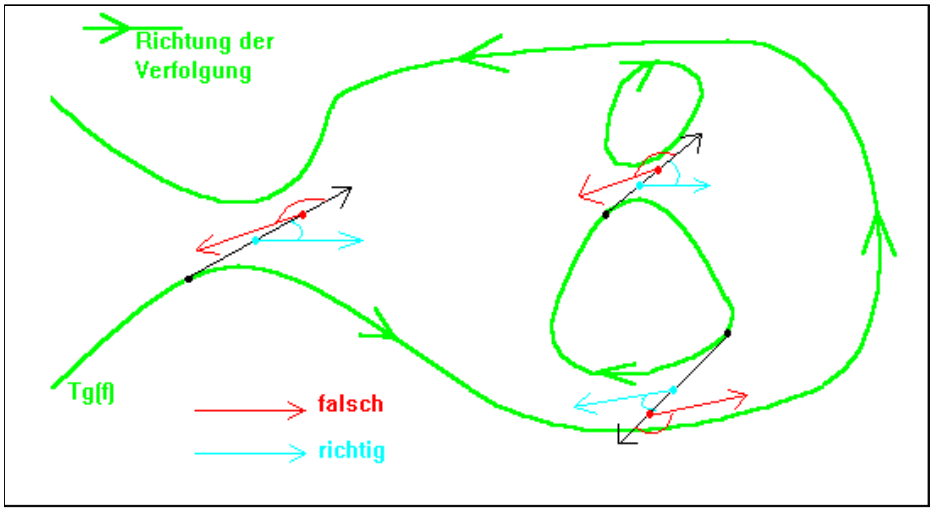

Abbildung 3.3: Sprünge bei der Rekonstruktion einer Trajektorie

abgebrochen (d.h. bevor alle kritische Punkte gefunden werden konnten), so müssen die Parameter $(k_{\max}, \varkappa_{\max}, \gamma_{\max})$ und/oder Startpunkte geändert werden.

3.3 Kontinuierliches Quasi-Newton-Verfahren (KQNV)

Das im Abschnitt 3.2 beschriebene Verfahren benötigt in jedem Prädiktor- und Korrektorschritt eine Auswertung der Jacobi-Matrix $DH(x)$ der Hilfsfunktion $H(x)$. Um diese Matrix genau zu bestimmen, muß die Hesse-Matrix $D^2 f(x)$ der Zielfunktion $f(x)$ berechnet werden. Dies ist oft mit viel Programmieraufwand und langer Berechnungszeit verbunden.
Zuerst müssen nämlich alle Ableitungsfunktionen f_{x_i,x_j} symbolisch berechnet und implementiert werden. Die Auswertung der vielen Ableitungsfunktionen in mehreren Punkten kann dazu sehr aufwendig sein. Dann müssen noch die Jacobi-Matrix DH und der induzierte Tangentenvektor $t(DH)$ berechnet und bei Bedarf die Korrektur durchgeführt werden. Dadurch wird der Aufwand weiter erhöht.
Es werden deshalb Aufdatierungsmethoden entwickelt, die o.g. Schwierigkeiten vermeiden und den Aufwand des Verfahrens möglicherweise vermindern. Die für den Ausgangspunkt der Trajektorie berechnete oder approximierte Jacobi-Matrix DH wird in jedem Schritt nach dem Gebrauch nicht verwor-

fen, sondern so modifiziert, so daß die sogenannte Quasi-Newton-Bedingung erfüllt ist. Die so modifizierte Matrix ist für den neuen Prädiktor- bzw. Korrektorpunkt eine für anschlägige Zwecke gut geeignete (nicht unbedingt bzgl. der Euklidischen Norm gute) Approximation der Jacobi-Matrix DH.

3.3.1 Broyden Aufdatierungsformel

Die Aufdatierungsmethode wird anhand des gedämpften Newton-Verfahrens

$$x_{i+1} = x_i - \alpha_i DG\left(x_i\right)^{-1} G\left(x_i\right) \tag{3.4}$$

zur Lösung des Gleichungssystems $G\left(x\right) = 0$ für eine glatte Abbildung $G \in C^1\left(\mathbb{R}^n, \mathbb{R}^n\right)$ beschrieben. Um den hohen numerischen Aufwand des Verfahrens zu reduzieren, kann die Jacobi-Matrix $DG\left(x_i\right)$ in der Newton-Vorschriftsformel (3.4) durch eine Approximation A_i ersetzt werden:

$$x_{i+1} = x_i - \alpha_i A_i^{-1} G\left(x_i\right).$$

Aus der Taylorentwicklung von G in x_i

$$G\left(x_{i+1}\right) = G\left(x_i\right) + DG\left(x_i\right)\left(x_{i+1} - x_i\right) + O\left(\left\|x_{i+1} - x_i\right\|^2\right)$$

folgt, daß für nahliegende Punkte x_{i+1} und x_i, eine Approximation A_i der Jacobi-Matrix $DG\left(x_i\right)$ sinnvollerweise folgende Sekanntenbedingung (Quasi-Newton Bedingung) erfüllen sollte:

$$A_i\left(x_{i+1} - x_i\right) = G\left(x_{i+1}\right) - G\left(x_i\right). \tag{3.5}$$

Die Menge der Matrizen, die für die vorgegebenen Differenzenvektoren s_i und z_i

$$\begin{aligned} s_i &= x_{i+1} - x_i \\ z_i &= G\left(x_{i+1}\right) - G\left(x_i\right) \end{aligned}$$

die Sekanntenbedingung erfüllen wird hier mit $Q\left(s_i, z_i\right)$ bezeichnet:

$$Q\left(s_i, z_i\right) := \left\{A \in L\left(\mathbb{R}^n\right) \mid \quad As_i = z_i\right\}.$$

Zur Berechnung der Matrix A_i werden dann üblicherweise eine Ausgangsmatrix A_{i-1} und die Vektoren s_i und z_i benutzt:

$$A_i := \Phi\left(A_{i-1}, s_i, z_i\right).$$

In diesem Fall wird A_i die **Aufdatierungsmatrix** und Φ die **Aufdatierungsformel** genannt. Hier und im folgenden wird vorausgesetzt, daß die Punkte x_{i+1} und x_i verschieden sind. Sind die Punkte gleich, ist die Aufdatierung der Jacobi-Matrix nicht notwendig.

Definition 113 *Seien $x_0 \in \mathbb{R}^n$ und $A_0 \in L(\mathbb{R}^n)$ gegeben.*
Seien weiterhin x_i und A_i $(i = 1, 2, ...)$ rekursiv definiert durch:

$$\begin{aligned} x_{i+1} &: = x_i + \alpha_i A_i^{-1} G(x_i) \\ A_{i+1} &: = \Phi(A_i, s_{i+1}, z_{i+1}). \end{aligned}$$

Ist für die Matrizen A_i die Sekanntenbedingung erfüllt, so heißt das durch die Rekursionsvorschrift definierte Verfahren ***Quasi-Newton-Verfahren.***

In [7] wurde von Broyden folgende Aufdatierungsformel vorgeschlagen:

$$A_{i+1}^B := \Phi^B(A_i, s_{i+1}, z_{i+1}) = A_i + \frac{(z_i - A_i s_i) s_i^T}{s_i^T s_i}. \tag{3.6}$$

Für das entsprechende Quasi-Newton-Verfahren

$$x_{i+1} := x_i + \left(A_i^B\right)^{-1} G(x_i)$$

konnte in [14] superlineare Konvergenz bewiesen werden. Das Verfahren hat sich als sehr erfolgreich bei der Lösung nichtlinearer Gleichungssysteme erwiesen, insbesondere dann, wenn keine zusätzlichen Informationen über die Beschaffenheit der Jacobi-Matrix $DG(x)$ vorhanden sind.

3.3.2 Broyden Aufdatierungsformel bei der Trajektorienrekonstruktion

Die Broydensche Aufdatierungsformel kann auch für unterbestimmte Gleichungssysteme

$$G(x) = 0, \qquad G \in C^1(\mathbb{R}^n, \mathbb{R}^m), m < n$$

zur Aufdatierung der nicht quadratischen Jacobi-Matrizen $DG(x)$ mittels der Vektoren $s \in \mathbb{R}^n$ und $z \in \mathbb{R}^m$ angewendet werden:

$$A^B := \Phi^B(A, s, z) = A + \frac{(z - As) s^T}{s^T s}.$$

Die Broydensche Matrix A^B ist dabei stets die beste Approximation der Ausgangsmatrix A bzgl. der Menge $Q(s, z)$ in der Frobenius-Norm $\|\|_F$.

Satz 114 *Seien $A \in L(\mathbb{R}^n, \mathbb{R}^m)$, $0 \neq s \in \mathbb{R}^n$ und $z \in \mathbb{R}^m$ gegeben.*
Die Approximationsaufgabe

$$\min_{B \in Q(s,z)} \|B - A\|_F = \min_{B \in Q(s,z)} \left(\sum_{k=1}^{n} \|b_k - a_k\|^2 \right)^{1/2} \tag{3.7}$$

besitzt eine eindeutige Lösung $\hat{B} = A^B$.

Beweis. Seien mit $a_k, b_k, \hat{b}_k$ entsprechend die Zeilen der Matrizen $A, B, \hat{B}$ und mit z_k die k-te Komponente des Vektors z bezeichnet.
Mit $q_k\,(s,z)$ $(k=1,...,n)$ wird dann die Menge der Vektoren $b \in \mathbb{R}^n$ gemeint, für die die entsprechende Sekanntenbedingung erfüllt ist:

$$q_k\,(s,z) := \left\{ b \in \mathbb{R}^n | \quad b^T s = z_k \right\}.$$

Die Approximationsaufgabe (3.7) kann getrennt bzgl. der Zeilen b_k der Matrix B gelöst werden:

$$\min_{b_k^T \in q_k(s,z)} \|b_k - a_k\|^2 .$$

Gesucht ist also ein Vektor $\hat{b}_k^T$ auf der Hyperebene $q_k\,(s,z)$, der dem Punkt a_k möglichst nahe liegt. Dies kann eindeutig durch die Projektion des Vektors a_k^T auf die Hyperebene $q_k\,(s,z)$ gelöst werden.
Es kann leicht nachgeprüft werden, daß die Zeilen $\hat{b}_k$ der Broydenschen Matrix A^B, die orthogonale Projektion von a_k auf $q_k\,(s,z)$ sind. Es gilt nämlich:

$$\hat{b}_k s = \left(a_k + \frac{z_k - a_k s}{s^T s} s^T \right) s = z_k \Longrightarrow \hat{b}_k \in q_k\,(s,z)$$

und weiter für einen beliebigen Vektor $b \in q_k\,(s,z)$:

$$\left(b^T - \hat{b}_k\right)\left(\hat{b}_k^T - a_k^T\right) = \frac{z_k - a_k s}{s^T s}\left(b^T - \hat{b}_k\right) s = 0.$$

■

Für das eigentliche Problem der Rekonstruktion der Trajektorie $T\,(f)$ sei die Jacobi-Matrix $DH\,(x_0)$ der Hilfsfunktion H in einem Ausgangspunkt x_0 gegeben. Um eine Punktfolge $\{x_i\}$ entlang der entsprechenden Trajektorienkomponente zu erzeugen, werden die Aufdatierungsmatrizen $A_{i,j}^B$ in jedem Prädiktor- und Korrektorpunkt konstruiert.
Mit folgenden Bezeichnungen kann die Broydensche Aufdatierungsformel für den Prädiktor- und Korrektorschritt des Trajektorienverfahrens einfach dargestellt werden:

$$\begin{aligned} w\,(x,y) &:= \frac{z - As}{\|s\|} = \frac{H\,(y) - H\,(z) - A\,(y-x)}{\|y-x\|} \\ v(x,y) &:= \frac{s}{\|s\|} = \frac{(y-x)}{\|y-x\|}. \end{aligned}$$

Broyden-Formel für den Prädiktorschritt

Für den Prädiktorschritt $p_i := p \cdot t\left(A_i^B\right)$ der Länge p,

$$x_{i+1,0} = x_i + p_i,$$

kann die neue Matrix $A_{i+1,0}^B$ wie folgt berechnet werden:

$$A_{i+1,0}^B := A_i^B + w\left(x_i, x_{i+1,0}\right) v\left(x_i, x_{i+1,0}\right)^T.$$

Die Hilfsvektoren $w\left(x_i, x_{i+1,0}\right)$ und $v\left(x_i, x_{i+1,0}\right)$ sind hierbei gegeben durch:

$$\begin{aligned} w\left(x_i, x_{i+1,0}\right) &= \frac{H\left(x_i\right) - H\left(x_{i+1,0}\right)}{p} \\ v\left(x_i, x_{i+1,0}\right) &= t\left(A\right). \end{aligned}$$

Bemerkung 115 *In einem Prädiktorschritt liegt der Vektor Δx der Veränderung des Bezugspunktes,*

$$\Delta x := x_{i+1,0} - x_i = pt\left(A_i^B\right),$$

im Kern der Ausgangsmatrix A_i^B (tangentialer Schritt). Die Pseudoinverse $\left(A_i^B\right)^+$ wird also bei diesem Aufdatierungsschritt immer nur auf einen neuen Kern projiziert [25].

Broyden-Formel für den Korrektorschritt

Für den Korrektorschritt $k_{i,j} := -\left(A_{i,j}^B\right)^+ H\left(x_{i,j}\right)$ des Trajektorienverfahrens

$$x_{i,j+1} = x_{i,j} + k_{i,j}$$

kann die neue Matrix $A_{i,j+1}^B$ wie folgt berechnet werden:

$$A_{i,j+1}^B := A_{i,j}^B + w\left(x_{i,j}, x_{i,j+1}\right) v\left(x_{i,j}, x_{i,j+1}\right)^T.$$

Die Hilfsvektoren $w\left(x_i, x_{i+1,0}\right)$ und $v\left(x_i, x_{i+1,0}\right)$ sind hierbei gegeben durch:

$$\begin{aligned} w\left(x_i, x_{i+1,0}\right) &= \frac{H\left(x_{i,j+1}\right)}{\|k_{i,j}\|} \\ v\left(x_i, x_{i+1,0}\right) &= \frac{k_{i,j}}{\|k_{i,j}\|}. \end{aligned}$$

Bemerkung 116 *In einem Korrektschritt liegt der Vektor Δx der Veränderung des Bezugspunktes,*

$$\Delta x := x_{i,j+1} - x_{i,j} = -\left(A_{i,j}^B\right)^+ H\left(x_{i,j}\right),$$

orthogonal zum Kern der Ausgangsmatrix $A_{i,j}^B$ (Newton-Schritt). Der Kern der aufdatierten Matrix $A_{i,j+1}^B$ wird sich hierbei nicht ändern.

3.3.3 Fehlerkontrolle der aufdatierten Matrizen

Wie es im vorherigen Abschnitt 3.3.2 bereits beschrieben wurde, sind in einem Prädiktor-Korrektor Verfahren zwei verschiedene Aufdatierungsmodelle vorgesehen. Auf Grund der angesprochenen Eigenschaften dieser Aufdatierungsschritte (vgl. Bemerkungen 115 und 116) sollte nicht auf eins der beiden Modelle verzichten werden. Auch dann nicht, wenn die Prädiktorpunkte nah bei der Trajektorie liegen sollten und nur eine geringe Korrektur notwendig wäre.
Es ist deshalb naheliegend, beide Aufdatierungsmöglichkeiten zu mischen [24]. In [25] wurde beschrieben, wie man die gemischte Aufdatierung gleichzeitig, also in einem Aufdatierungsschritt für den tangentialen und den Newton-Schritt effizient durchführen kann.

Ob die mit Hilfe der Aufdatierungsformel berechnete Approximation der Jacobi-Matrix $DH(x)$ den Anforderungen entspricht und eine korrekte Rekonstruktion der Trajektorie zuläßt, ist direkt nur sehr schwer zu prüfen. Mit Hilfe des im Abschnitt 3.2.2 beschriebenen Kontraktionstests kann die "Güte" der Approximation indirekt geprüft werden.
Ist nämlich die Aufdatierungsmatrix eine sehr schlechte Approximation der Jacobi-Matrix $DH(x)$, so ist es unwahrscheinlich, daß das Korrektorverfahren konvergiert. In diesem Fall sollte der Prädiktorschritt gekürzt oder die Jacobi-Matrix $DH(x)$ mit einer anderen Methode (z.B. Differenzenverfahren) genauer bestimmt werden.

Anderseits, wenn die Konvergenz des Korrektor-Verfahrens festgestellt und ein gute Näherung für den neuen Trajektorienpunkt gefunden wurde, so kann auch die Aufdatierungsmatrix akzeptiert werden.

3.3.4 QR-Aufdatierung

Die Berechnung des Tangentenvektors $t(DH(x))$ und des Korrektorschrittes $-DH(x_{i+1,j})^{+} H(x_{i+1,j})$ kann mit Hilfe der QR-Zerlegung der Jacobi-Matrix $DH(x)$ effizient bewältigt werden. Die QR-Zerlegung einer gegebenen Matrix A kann mit Hilfe der Householder-Transformation oder der Givensrotationen durchgeführt werden.
Um den Aufwand der Zerlegung in jedem Schritt zu vermeiden, ist es angebracht das Aufdatierungsverfahren direkt auf die Matrizen Q und R anzuwenden.

Die QR-Aufdatierung mit Hilfe der Givensrotationen wurde in [1] ausführlich beschrieben und deren Stabilitätseigenschaften untersucht.
Ein möglicher Implementierungsweg wird hier kurz skizziert.

Givensrotationen in der QR-Zerlegung

Givensrotation $\mathcal{G}_{kl}(\phi)$:$\mathbb{R}^n \to \mathbb{R}^n$ (auch ebene Rotation genannt) ist eine lineare Abbildung, die als Rotation in der Ebene $span\{e_k, e_l\}$ um den Winkel ϕ interpretiert werden kann. Diese orthogonale Transformation ist mit Hilfe einer Matrix $G_{kl}(\phi)$ beschrieben, die sich von der Einheitsmatrix I_n nur durch folgende Einträge unterscheidet:

$$\begin{aligned} g_{kk} &= g_{ll} = \cos(\phi) \\ g_{kl} &= -g_{lk} = \sin(\phi). \end{aligned}$$

Wendet man die Rotation auf einen beliebigen Vektor x an, so ändern sich nur die Einträge x_k und x_l, die anderen Einträge bleiben unverändert. Für $y := G_{kl}x$ gilt also:

$$y_i = \begin{cases} x_k \cos(\phi) + x_l \sin(\phi), & i = k \\ -x_k \sin(\phi) + x_l \cos(\phi), & i = l \\ x_i, & sonst. \end{cases}$$

Um eine $n \times n-1$ Matrix A^T in eine orthogonale $n \times n$ Matrix Q und eine obere $n-1 \times n-1$ Dreiecksmatrix R zu zerlegen, werden auf beide Seiten der Startformel

$$Q_0^T A^T = R_0 \qquad (Q_0 := I_n; \quad R_0 := A^T),$$

$n(n-1)/2$ Rotationen $\mathcal{G}_{kl}(\phi)$ $(k = 1, ..., n-1; \quad l = k+1, ..., n)$ angewendet

$$(G_{n-1,n}() ... G_{1,3}() G_{1,2}() I_n) A^T = G_{n-1,n}() ... G_{1,3}() G_{1,2}() A^T.$$

In jedem Schritt der Zerlegung werden die Matrizen Q und R auf folgende Weise modifiziert:

$$\begin{aligned} Q_{neu} &= G_{kl} Q_{alt} \\ R_{neu} &= G_{kl} R_{alt}. \end{aligned}$$

Die Matrix G_{kl} wird dabei so gewählt, daß das Element r_{lk}, der Matrix R *Null* wird. Die Reihenfolge der Transformation sichert, daß die bereits auf *Null* gesetzten Einträge der Matrix R nicht mehr geändert werden. Auf diese Wiese entstehen die gesuchten Matrizen Q und R:

$$\begin{aligned} Q &= (G_{n-1,n}() ... G_{1,3}() G_{1,2}() I_n)^T \\ R &= G_{n-1,n}() ... G_{1,3}() G_{1,2}() A^T. \end{aligned}$$

Die Matrix Q ist als Produkt der orthogonalen Matrizen G_{kl} natürlich orthogonal.

Bemerkung 117 *Da in jedem Schritt lediglich zwei Zeilen der Matrizen Q und R neu berechnet werden müssen, ist der Gesamtaufwand der QR-Zerlegung $O(n^3)$.*

Givensrotation in der QR-Aufdatierung

Problem 118 *Sei mit Q und R die QR-Zerlegung für eine $n \times n-1$ Matrix A^T gegeben. Mit den Vektoren $v \in \mathbb{R}^n$ und $w \in \mathbb{R}^{n-1}$ sei weiterhin der Prädiktor- bzw. der Korrektorschritt (s. Abschnitt 3.3.2) beschrieben. Mit B wird hier die Broydensche Aufdatierungsmatrix bezeichnet:*

$$B := A + wv^T.$$

Gesucht werden:
eine orthogonale $n \times n$ Matrix Q_B und eine obere $n-1 \times n-1$ Dreiecksmatrix R_B, für die gilt:

$$Q_B^T B^T = R_B.$$

Mit den gegebenen Matrizen Q und R gilt hier für die Matrix B:

$$Q^T B^T = Q^T A^T + Q^T v w^T = R + Q v w^T. \tag{3.8}$$

Die neuen Matrizen Q_B und R_B können wie folgt effizient bestimmt werden:

Aufdatierungsschema 119 *(QR-Aufdatierung)*

1. *Auf beiden Seiten der Formel (3.8) werden Givensrotationen angewendet, so daß der Term Qvw^T eine einfache Form annimmt, aber die Struktur der Matrix R dabei nicht komplett zerstört wird.*

2. *Mit Hilfe weiterer geeignet gewählter Rotationen wird die Dreiecksstruktur der Matrix R wieder rekonstruiert*

Zunächst werden $n-1$ Rotationen $\hat{G}_{n-1,n}(), ..., \hat{G}_{2,3}(), \hat{G}_{1,2}()$ eingesetzt, die den Vektor $u := Qv$ auf die x_1-Achse projizieren:

$$\hat{u} := \hat{G}_{1,2}() \hat{G}_{2,3}() ... \hat{G}_{n-1,n}() u = (\hat{u}_1, 0, ..., 0)^T.$$

Mit dem Vektor $\hat{u}$ und den Matrizen $\hat{Q}$ und $\hat{R}$

$$\begin{aligned} \hat{Q} &: = \hat{G}_{1,2}() \hat{G}_{2,3}() ... \hat{G}_{n-1,n}() Q \\ \hat{R} &: = \hat{G}_{1,2}() \hat{G}_{2,3}() ... \hat{G}_{n-1,n}() R, \end{aligned}$$

nimmt die Formel (3.8) folgende Gestalt:

$$\hat{Q}B^T = \hat{R} + \hat{u}w^T.$$

Die Matrix $\hat{R}$ ist keine Dreiecksmatrix mehr. Die Einträge der Nebendiagonale $(r_{21}, r_{32}, ..., r_{n,n-1}.)$ haben sich durch die Anwendung der Givensrotationen geändert. Die Matrix $\hat{R}$ hat also die Hessenbergsche Form. Alle Einträge der Matrix $\hat{u}w^T$, ausgenommen der ersten Zeile, sind gleich $Null$. Die Matrix $\hat{R} + \hat{u}w^T$ hat also ebenfalls Hessenbergsche Form.

Mit Hilfe der geeignet gewählten Rotationen $\tilde{G}_{1,2}(), \tilde{G}_{2,3}(), ..., \tilde{G}_{n-1,n}()$ können dann die Elemente der besagten Nebendiagonale wieder auf $Null$ gesetzt werden. Die gesuchten Matrizen Q_B und R_B sind dann gegeben durch:

$$\begin{aligned} Q_B &: = \tilde{G}_{n-1,n}() ... \tilde{G}_{2,3}() \tilde{G}_{1,2}() \hat{Q} \\ R_B &: = \tilde{G}_{n-1,n}() ... \tilde{G}_{2,3}() \tilde{G}_{1,2}() \left(\hat{R} + \hat{u}w^T\right). \end{aligned}$$

Bemerkung 120 *Da in diesem Fall lediglich* $2(n-1)$ *Givensrotationen durchgeführt werden mußten, ist der Gesamtaufwand der QR-Aufdatierung* $O(n^2)$.

3.4 Ableitungsfreies Surrogate-Verfahren

In manchen praktischen Optimierungsproblemen ist die analytische bzw. numerische Auswertung der Ableitungen der Zielfunktion nicht möglich oder unverhältnismäßig teuer.
Ein Standard-Verfahren zur ableitungsfreien Lösung klassischer Optimierungsprobleme (nur ein und nicht alle kritische Punkte wird gesucht) wurde von Nelder und Mean in [29] vorgeschlagen. In diesem Simplex-Verfahren wird das Minimum der Zielfunktion durch Wertvergleiche und Austauschschritte erreicht. Ein möglicher Ansatz zur Lösung der Probleme mit mehreren lokalen Minima wäre, die Suche immer wieder von einem neuen Ausgangspunkt zu starten. Es ist aber offensichtlich, daß ein solches Verfahren im allgemeinen nicht sehr effizient wäre.
In den letzten Jahren wurden neue ableitungsfreie Verfahren entwickelt, die mit Hilfe der Surrogate-Technik einen kritischen Punkt der Zielfunktion finden [5].
In jedem Iterationsschritt des auf der Basis der Surrogate-Technik implementierten Verfahrens wird die Zielfunktion anhand der gesammelten Informationen durch eine Surrogate-Funktion (Modellfunktion) ersetzt und diese dann minimiert. Für den gewonnenen vermeintlich optimalen Punkt wird die Zielfunktion ausgewertet. Die Qualität der Modellfunktion wird anschließend

geprüft und lokal verbessert.
In den Arbeiten [11][8][9][10] wurden die Theorie und praktische Umsetzung dieser Technik mit Hilfe der multidimensionalen quadratischen Interpolation vorgestellt. In [12] [13][32][31] wurden geeignete Verfahren der multidimensionalen Lagrange- und Newton-Interpolation beschrieben.
In der vorliegenden Arbeit wird vorgeschlagen, die Surrogate-Technik für den Trajektorien-Ansatz zu nutzen. Für die Modellfunktion der Zielfunktion kann dann als Näherung der klassischen Newton-Trajektorie bzw. Richtungsfeld-Trajektorie ein Trajektorienstück approximiert werden. Entlang der Trajektorie wird das Modell kontinuierlich angepaßt und die Trajektorie stückweise rekonstruiert. Die Qualität der aufgestellten Modelle wird bei der Schrittlängensteuerung berücksichtigt.

3.4.1 Modellbildung

Aufgabenstellung

Die Zielfunktion $f(x)$ wird lokal in einem Punkt z_0 durch eine quadratische Modellfunktion $m(z_0 + s)$ ersetzt:

$$m(z_0 + s) := f(z_0) + g^T s + \frac{1}{2} s^T H s. \tag{3.9}$$

Der Vektor $g \in \mathbb{R}^n$ und die symmetrische $n \times n$ Matrix H werden hier nicht als Gradient $\nabla f(z_0)$ und Hesse-Matrix $D^2 f(z_0)$ der Zielfunktion f berechnet oder approximiert [11]. Das Modell wird stattdessen als Lösung des folgenden Interpolationsproblems bestimmt.

Problem 121 *Sei eine endliche Menge I der Interpolationspunkte gegeben*

$$I = \{z_0, ..., z_{M-1}\} \subset \mathbb{R}^n.$$

Für die Zielfunktion f seien weiterhin die Funktionswerte $f(z_0), ..., f(z_{M-1})$ bekannt.
Gesucht ist eine multidimensionale quadratische Funktion $m(x)$ der Form (3.9), für die folgende Interpolationsbedingung erfüllt ist:

$$m(z_i) = f(z_i), \quad z_i \in I. \tag{3.10}$$

Um die Existenz und Eindeutigkeit der Lösung zu sichern, muß die Menge I der Interpolationspunkte gewisse Bedingungen erfüllen.
Zuerst sollte die Anzahl der Interpolationspunkte M der Anzahl der gesuchten Koeffizienten der Modellfunktion $m(x)$ gleichen.

Definition 122 *Die Menge der Interpolationspunkte $I \subset \mathbb{R}^n$ heißt* ***vollständig****, falls für die Anzahl der Interpolationspunkte M gilt:*

$$M = 1 + n + \binom{n+1}{2} = \binom{n+2}{2}. \tag{3.11}$$

Für die Existenz und die Eindeutigkeit der Interpolante ist die Bedingung (3.11) im allgemeinen nicht ausreichend. Liegen die Punkte beispielweise auf einer Hyperebene (einer Gerade, einem Kreis, Hyperbel, etc), so ist die Interpolante entweder nicht eindeutig oder existiert gar nicht.
Mit Hilfe der geeignet gewählten quadratischen Basispolynome $p_i(x)$, $i = 1, ..., M$ kann das Interpolationsproblem 121 zu folgenden linearen Gleichungssystem modifiziert werden:

$$P(I)\,\alpha = \begin{pmatrix} p_1(z_1) & \cdots & p_M(z_1) \\ \vdots & & \vdots \\ p_1(z_M) & \cdots & p_M(z_M) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = \begin{pmatrix} f(z_1) \\ \vdots \\ f(z_M) \end{pmatrix}. \tag{3.12}$$

Die Interpolationsbedingung (3.10) ist dann für die entsprechende Linearkombination $m(x)$ der Basispolynome $p_i(x)$ erfüllt:

$$m(x) = \sum_{i=1}^{M} \alpha_i p_i(x).$$

Das gegebene Interpolationsproblem ist also für beliebig vorgegebene bzw. berechnete Funktionswerte $f(z_i)$, $z_i \in I$ genau dann eindeutig lösbar, wenn die entsprechende Vandermond-Determinante $\det P(I)$ nicht singulär ist [12][31].

Durch geeignete Wahl der Interpolationspunkte kann dies in einem Ausgangspunkt der gesuchten Trajektorienkomponente leicht gesichert werden.

Definition 123 *Die Menge der Interpolationspunkte I heißt* ***gleichmäßig verteilt*** *bzgl. der Polynom-Basis P, falls die Vandermond-Determinante nicht singulär ist:*

$$\det P(I) \neq 0. \tag{3.13}$$

Satz 124 *Seien Interpolationspunkte $I = \{z_1, ..., z_M\} \subset \mathbb{R}^n$ und Interpolationswerte $f(I) = \{f(z_1), ..., f(z_M)\} \subset \mathbb{R}$ bekannt. Sei weiterhin mit $P = \{p_1(x), ..., p_M(x)\}$ eine endliche Menge linear unabhängiger Basispolynome gegeben.*

Die Menge I ist bzgl. P gleichmäßig verteilt genau dann, wenn ein eindeutiger Parametersatz $\alpha = \{\alpha_1, ..., \alpha_M\}$ existiert, so daß für die lineare Kombination der Basispolynome

$$m(I) = \sum_{i=1}^{M} \alpha_i p_i(x),$$

die Interpolationsbedingung (3.10) erfüllt ist.

Beweis. Die Behauptung folgt direkt aus der Erkenntnis, daß das lineare Gleichungssystem (3.12) genau dann eindeutig lösbar ist, wenn die Vandermond-Determinante $\det P(I)$ nicht singulär ist. ■

Bemerkung 125 *Für die vollständige bzgl. einer beliebigen Polynom-Basis P gleichmäßig verteilte Menge I ist die eindeutige Existenz der Interpolante auf jedem Fall gesichert. Die Bedingung (3.13) ist dann natürlich nicht von der Wahl der Basispolynome abhängig.*

Für die Interpolationspunkte, die während der Rekonstruktion der Komponente in die Modellierung hinzugefügt werden, ist die Bedingung (3.13) (insbesondere für höherdimensionale Probleme) nicht immer leicht zu prüfen und damit auch zu sichern (s. Abschnitt 3.4.1).
Da die Rekonstruktion der Trajektorie und deshalb auch die Modellanpassung nicht notwendigerweise mit größter Genauigkeit durchgeführt werden müssen (vgl. Abschnitt 1.2.3), ist es im Falle der ungeeignet gewählten Interpolationspunkte zulässig, eine sub-quadratische Modellierung vorzunehmen. Es werden dann nicht alle quadratischen Terme bzw. Basispolynome berücksichtigt.
In [32] wurde ein auf der Newton-Interpolation aufgebautes Verfahren beschrieben, mit dessen Hilfe für die vorgegebenen Punkte $z_1, ..., z_M$ und Funktionswerte $f_1, ..., f_M$ eine möglichst vollständige sub-quadratische Interpolante bestimmt werden kann.

Konstruktion Newton-Basispolynome

Für die Veränderlichen $x = (\xi_1, ..., \xi_n)$ und eine Konstante ξ_0 werden n-dimensionale quadratischen Monome $m_{ij}(x)$ wie folgt konstruiert:

$$m_{ij}(x) = \xi_i \xi_j, \qquad i, j = 0, ..., n, \quad i \leq j.$$

Für die nachfolgende Konstruktion der Newton-Basispolynome können an Stelle der quadratischen Monome $m_{ij}(x)$ auch andere Basispolynome genommen werden.

Die gewählten Basispolynome werden dann bzgl. einer bestimmten Ordnung sortiert und entsprechend mit den Interpolationspunkten den Listen N^0, N^1, N^2 und Z^0, Z^1, Z^2 zugeordnet:

1. In die Listen N^0 und Z^0 kommt die konstante Funktion $m_{00}(x)$ und der Interpolationspunkt z_0.

2. In die Listen N^1 und Z^1 werden die lineare Funktionen $m_{0j}(x)$ zusammen mit den Punkten $z_1, ..., z_n$ eingeordnet.

3. In die Listen N^2 und Z^2 kommen dann die echt-quadratischen Monome $m_{ij}(x)$ $i, j = 1, ..., n, \quad i \leq j$ und die restlichen Interpolationspunkte $z_{n+1}, ..., z_{M-1}$.

Für die Punkte $z_0, ..., z_{M-1}$ werden in drei Schritten die Newtonsche Basispolynome $N_1^0(x)$, $N_1^1(x), ..., N_n^1(x)$, und $N_1^2(x), ..., N_{M'}^2(x)$, $M' < M-n$ auf folgende Weise konstruiert (vgl. Beispiel 127):

1. **Schritt** - das Basispolynom $N_1^0(x)$ wird bestimmt (im Punkt z_0 auf 1 normiert), die anderen Basispolynome $N_j^l(x)$ werden im Punkt z_0 auf *Null* gesetzt:

$N_1^0(x) \leftarrow N_1^0(x) / N_1^0(z_0)$
für alle Elemente der Listen N^1 und N^2 $(k = 1, 2)$
$N_j^k(x) \leftarrow N_j^k(x) - N_j^k(z_0) N_1^0(x)$

   Für den Punkt z_0 und die Polynome $N_j^l(x)$ gilt dann:

$$N_j^l(z_0) = \begin{cases} 1, & l = 0, \quad j = 1 \\ 0, & sonst. \end{cases}$$

2. **Schritt** - den Basispolynomen $N_i^1(x)$ werden Punkte aus der Liste Z^1 zugeordnet (vgl. Bemerkung 126) und in der temporären Liste Z^{TEMP} gespeichert. Die Polynome $N_i^1(x)$ werden in den entsprechenden Punkten z_i^1 auf 1 normiert. Die anderen Polynome $N_j^1(x)$ $(j \neq i)$ und $N_j^2(x)$

(aber nicht $N_1^0(x)$!) werden in diesen Punkten auf $Null$ gesetzt.

$Z^{TEMP} \leftarrow \emptyset$
für $i = 1, ..., n$
wähle ein Element z_i^1 der Liste $Z^1 \setminus Z^{TEMP}$, so daß $N_i^1(z_i^1) \neq 0$ $N_i^1(x) \leftarrow N_i^1(x) / N_i^1(z_i^1)$
für alle Elemente der Liste N^1, ausgenommen $N_i^1(x)$, $(j \neq i)$
$N_j^1(x) \leftarrow N_j^1(x) - N_j^1(z_i^1) N_i^1(x)$
für alle Elemente der Liste N^2
$N_j^2(x) \leftarrow N_j^2(x) - N_j^2(z_i^1) N_i^1(x)$
$Z^{TEMP} \leftarrow z_i^1$
$Z^1 \leftarrow Z^{TEMP}$

Für die Punkte z_i^1 und die Polynome $N_j^1(x)$ bzw. $N_j^2(x)$ gilt dann:

$$\begin{aligned} N_j^1(z_i^1) &= \begin{cases} 1, & j = i \\ 0, & sonst. \end{cases} \\ N_j^2(z_i^1) &= 0. \end{aligned}$$

3. **Schritt** -den Basispolynomen $N_i^2(x)$ werden Punkte aus der Liste Z^2 zugeordnet und in der temporären Liste Z^{TEMP} gespeichert. Die Polynome $N_i^2(x)$ werden in den entsprechenden Punkten z_i^2 auf 1 normiert. Die anderen Polynome $N_j^2(x)$ $(j \neq i)$ (aber nicht $N_1^0(x)$ und $N_j^1(x)$!) werden in diesen Punkten auf $Null$ gesetzt. Die Polynome $N_1^0(x)$ und $N_j^1(x)$ werden in diesem Schritt nicht mehr modifiziert.

$Z^{TEMP} \leftarrow \emptyset$
für $i = 1, ..., M'$
wähle ein Element z_i^2 der Liste $Z^2 \setminus Z^{TEMP}$, so daß $N_i^2(z_i^2) \neq 0$ $N_i^2(x) \leftarrow N_i^2(x) / N_i^2(z_i^2)$
für alle Elemente der Liste N^2, ausgenommen $N_i^2(x)$, $(j \neq i)$
$N_j^2(x) \leftarrow N_j^2(x) - N_j^2(z_i^2) N_i^2(x)$
$Z^{TEMP} \leftarrow z_i^2$
$Z^2 \leftarrow Z^{TEMP}$

Für die Punkte z_i^2 und die Polynome $N_j^2(x)$ gilt dann:

$$N_j^2(z_i^2) = \begin{cases} 1, & j = i \\ 0, & sonst. \end{cases}$$

Bemerkung 126 *Die Zuordnung der Punkte könnte nach den Prinzip des größten Polynomwertes (vgl. Spaltenpivotsuche der Gauß-Elimination) erfolgen:*

$$\max_{z_i^l \in Z^l \setminus Z^{TEMP}} N_i^l \left(z_i^l\right).$$

Wird für ein Basispolynom $N_j^l(x)$ kein geeigneter Punkt aus der Liste Z^l gefunden, so muß bei der Modellbildung auf das Polynom verzichtet werden. In einem solchem Fall ist nur eine sub-quadratische Interpolation möglich. Die abgestoßenen Polynome werden an das Ende der Liste verschoben und können bei der nächsten Modellbildung wieder angenommen werden.
Es soll dabei beachtet werden, daß die ursprüngliche Anordnung der Punkte in der Listen Z^1 und Z^2 sich im Laufe des Verfahrens möglicherweise ändern wird. Die nicht zugeordneten Punkte werden aus der Listen entfernt.

Beispiel 127 *Seien folgende Interpolationspunkte gewählt:*

$$I := \{(0,0), (0,1), (1,0), (2,0)\,(1,1)\,(0,2)\}.$$

Als Basispolynome werden die quadratischen Monome $m_{ij}(z)$ gewählt:

$$\left\{c, x, y, x^2, xy, y^2\right\}.$$

Die Polynome und die Interpolationspunkte werden den Listen zugeordnet:

$$\begin{array}{cc} N^0 = \{c\} & Z^0 = \{(0,0)\} \\ N^1 = \{x, y\} & Z^1 = \{(0,1), (1,0)\} \\ N^2 = \{x^2, xy, y^2\} & Z^2 = \{(2,0)\,(1,1)\,(0,2)\}. \end{array}$$

Die Newton-Basispolynome werden dann wie folgt bestimmt:

1. *Das-Newton-Polynom $N_1^0(z)$ wird als konstante Funktion gleichgesetzt:*

$$N_1^0(z) = \frac{c}{c} = 1.$$

   Da die übrigen Basispolynome $N_j^k(z)$, $k = 1, 2$ im Punkt z_0 gleich Null sind, ändern sie sich in diesem Schritt nicht:

$$N_j^k(z) = N_j^k(z) - 0 \cdot N_1^0(z).$$

2. *Da für das Polynom $N_1^1(z) = x$, der Punkt $z_1^1 = (0,1)$ ein Nullstelle ist, darf er nicht als Kehrpunkt genommen werden. Stattdessen wird der*

Punkt $z_2^1 = (1,0)$ *den Polynom* $N_1^1(z)$ *zugeordnet. Die Basispolynome werden dann wie folgt modifiziert:*

$$\begin{aligned} N_1^1(z) &= \frac{x}{1} = x \\ N_2^1(z) &= y - 0x = y \\ N_1^2(z) &= x^2 - 1x = x^2 - x \\ N_2^2(z) &= xy - 0x = xy \\ N_3^2(z) &= y^2 - 0x = y^2 \end{aligned}$$

Es soll hier beachtet werden, daß die Reihenfolge der Punkte sich geändert hat:

$$I := \{(0,0),(1,0),(0,1),(2,0)(1,1)(0,2)\}$$

Dem Polynom $N_2^1(z)$ *wird der Punkt* $z_1^1 = (0,1)$ *zugeordnet. Die Basispolynome werden dann wie folgt modifiziert:*

$$\begin{aligned} N_2^1(z) &= \frac{y}{1} = y \\ N_1^1(z) &= x - 0y = x \\ N_1^2(z) &= x^2 - x - 0y = x^2 - x \\ N_2^2(z) &= xy - 0y = xy \\ N_3^2(z) &= y^2 - 1y = y^2 - y \end{aligned}$$

3. *In drittem Schritt werden lediglich Polynome* $N_j^2(z)$ *modifiziert. Dies führt dann zu der folgenden Newton-Basis:*

$$\begin{aligned} N_1^0(z) &= 1 \\ N_1^1(z) &= x \\ N_2^1(z) &= y \\ N_1^2(z) &= \frac{1}{2}\left(x^2 - x\right) \\ N_2^2(z) &= xy \\ N_3^2(z) &= \frac{1}{2}\left(y^2 - y\right) \end{aligned}$$

Newton-Interpolation

Das Interpolationsgleichungssystem (3.12) nimmt in diesem Fall folgende Form an:

$$\begin{pmatrix} 1 & 0 & 0 \\ N^0(Z^1) & I_n & 0 \\ N^0(Z^2) & N^1(Z^2) & I_{M'} \end{pmatrix} \begin{pmatrix} \alpha^0 \\ \alpha^1 \\ \alpha^2 \end{pmatrix} = \begin{pmatrix} f(Z^0) \\ f(Z^1) \\ f(Z^2) \end{pmatrix},$$

wobei $\alpha^0, f(Z^0) \in \mathbb{R}$, $\alpha^1, f(Z^1) \in \mathbb{R}^n$ und $\alpha^2, f(Z^2) \in \mathbb{R}^{M'}$. Mit $f(Z^i)$ bzw. $N^k(Z^i)$ werden die Werte der Zielfunktion f bzw. der Newton-Basispolynome N_j^k in den Interpolationspunkten aus der Liste Z^i kurz bezeichnet.
Die Koeffizienten α^0, α^1 und α^2 können mit wenig Aufwand rekursiv bestimmt werden:

$$\begin{aligned}
\alpha^0 &\leftarrow f(z_0) \\
\alpha_i^1 &\leftarrow f\left(z_i^1\right) - \alpha^0 N^0\left(z_i^1\right), \qquad i = 1, ..., n \\
\alpha_i^2 &\leftarrow f\left(z_i^2\right) - \alpha^0 N^0\left(z_i^2\right) - \left(\alpha^1\right)^T N^1\left(z_i^2\right), \qquad i = 1, ..., M'.
\end{aligned}$$

Korollar 128 *Die quadratische Interpolante $m[Z^0, Z^1, Z^2]$ einer Funktion f ist dann gegeben mit:*

$$m\left[Z^0, Z^1, Z^2\right] = \alpha^0 + \sum_{i=1}^{n} \alpha_i^1 N_i^1(x) + \sum_{i=1}^{M'} \alpha_i^2 N_i^2(x)$$

Bemerkung 129 *Unter der zusätzlichen Voraussetzung, daß die Basispolynome $m_{ij}(x)$ entsprechend der Ordnung den Listen N^0, N^1, N^2 zugeordnet sind (N^0-Konstante, N^1-lineare Terme), so sind auch die entsprechenden Newton-Polynome $N_1^0(x)$, $N_1^1(x), ..., N_n^1(x)$ konstant bzw. linear (vgl. Beispiel 127).*
Ein lineares Modell der Zielfunktion f (vgl. Abschnitt 3.4.2) kann also ohne zusätzlichen Aufwand bestimmt werden:

$$m\left[Z^0, Z^1\right] = \alpha^0 + \sum_{i=1}^{n} \alpha_i^1 N_i^1(x).$$

Beispiel 130 *Für die zweidimensionale Zielfunktion f_a aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y$$

und die Interpolationspunkte aus dem Beispiel 127

$$I := \{(0,0), (1,0), (0,1), (2,0)\,(1,1)\,(0,2)\}$$

sind die Funktionswerte gegeben mit:

$$f_a(I) = \left\{0, -\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 0, -\frac{2}{3}\right\}.$$

Mit den im Beispiel 127 bereits konstruierten Newton-Basispolynomen

$$\begin{aligned} N_1^0(z) &= 1 \\ N_1^1(z) &= x \\ N_2^1(z) &= y \\ N_1^2(z) &= \frac{1}{2}\left(x^2 - x\right) \\ N_2^2(z) &= xy \\ N_3^2(z) &= \frac{1}{2}\left(y^2 - y\right) \end{aligned}$$

können die Koeffizienten α^0, α^1 und α^2 leicht bestimmt werden:

$$\begin{aligned} \alpha^0 &= 0 \\ \alpha_1^1 &= -\frac{2}{3} - 0 \cdot N^0\left(z_1^1\right) = -\frac{2}{3} \\ \alpha_i^1 &= \frac{2}{3} - 0 \cdot N^0\left(z_2^1\right) = \frac{2}{3} \\ \alpha_1^2 &= \frac{2}{3} - 0 \cdot N^0\left(z_1^2\right) + \frac{2}{3}N_1^1\left(z_1^2\right) - \frac{2}{3}N_2^1\left(z_1^2\right) = 2 \\ \alpha_2^2 &= 0 - 0 \cdot N^0\left(z_2^2\right) + \frac{2}{3}N_1^1\left(z_2^2\right) - \frac{2}{3}N_2^1\left(z_2^2\right) = 0 \\ \alpha_3^2 &= -\frac{2}{3} - 0 \cdot N^0\left(z_3^2\right) + \frac{2}{3}N_1^1\left(z_3^2\right) - \frac{2}{3}N_2^1\left(z_3^2\right) = -2. \end{aligned}$$

Das lineare und quadratische Modell sind dann gegeben mit:

$$\begin{aligned} m\left[Z^0, Z^1\right] &= \frac{2}{3}(y - x) \\ m\left[Z^0, Z^1, Z^2\right] &= \frac{2}{3}(y - x) + x^2 - x - y^2 + y \\ &= \frac{5}{3}(y - x) + x^2 - y^2 \end{aligned}$$

Es kann leicht nachgeprüft werden, daß das lineare Modell $m\left[Z^0, Z^1\right]$ in den Punkten z_i, $i = 0..2$, und das quadratische Modell in den Punkten z_i, $i = 0..5$, mit der Zielfunktion $f_a(z)$ übereinstimmen.

Konstruktion Lagrange-Basispolynome

Ist die Menge der Interpolationspunkte vollständig und gleichmäßig verteilt, so ist die Newton-Interpolation dem auf den Lagrange-Basispolynomen basierenden Standardverfahren äquivalent [32][31]. Für die gegebene Menge I

gilt dann für die Lagrange-Basispolynome $L_i(x)$

$$L_i(z_j) = \begin{cases} 1, & j = i \\ 0, & sonst. \end{cases} \tag{3.14}$$

Für das quadratische Interpolationsproblem 121 lassen sich die Lagrange-Basispolynome $L_i(x)$ mit wenig Aufwand in drei Schritten aus den Newton-Basispolynomen $N_j^l(x)$ gewinnen:

1. **Schritt** - das Basispolynom $L_0(x)$ wird durch Modifizierung vom Newton-Basispolynom $N_1^0(x)$ mit Hilfe der skalierten Basispolynome $N_j^1(x)$ und $N_j^2(x)$ bestimmt.

$L_0(x) \leftarrow N_1^0(x)$
für alle Elemente der Liste N^1
$L_0(x) \leftarrow L_0(x) - L_0\left(z_j^1\right) N_j^1(x)$
für alle Elemente der Liste N^2
$L_0(x) \leftarrow L_0(x) - L_0\left(z_j^2\right) N_j^2(x)$

   Für das Polynom $L_0(x)$ und die Punkte aus der Listen Z^1 und Z^2 gilt dann:

$$L_0\left(z_j^l\right) = 0, \quad l \in \{1, 2\}$$

2. **Schritt** - die Basispolynome $L_i(x)$, $i = 1, ..., n$ werden durch Modifizierung der Newton-Basispolynome $N_i^1(x)$ mit Hilfe der skalierten Basispolynome $N_j^2(x)$ bestimmt.

$L_i(x) \leftarrow N_i^1(x)$
für alle Elemente der Liste N^2
$L_i(x) \leftarrow L_i(x) - L_0\left(z_j^2\right) N_j^2(x)$

   Für die Polynome $L_i(x)$, $i = 1, ..., n$ und die Punkte aus der Liste Z^2 gilt dann:

$$L_i\left(z_j^2\right) = 0.$$

3. **Schritt** - Setzt man die übrigen Polynome $L_i(x)$, $i = n+1, ..., M-1$ den Newton-Basispolynomen aus der Liste N^2 gleich,

$L_i(x) \leftarrow N_{i-n}^2, \qquad i = n+1, ..., M-1$

   so kann dann für alle Lagrange-Basispolynome $L_i(x)$ leicht die Bedingung (3.14) gezeigt werden.
   Die abgestoßenen Newton-Polynome $N_{M'+1}^2(x), ..., N_{M-n-1}^2(x)$, und Lagrange-Polynome $L_{M'+n+1}(x), ..., L_{M-1}(x)$ sind in allen Interpolationspunkten gleich *Null*.

Beispiel 131 *Für die Interpolationspunkte z_j^k aus dem Beispiel 127*

$$I := \{(0,0),(1,0),(0,1),(2,0)(1,1)(0,2)\}$$

und die entsprechenden Newton-Basispolynome N_j^k

$$\begin{aligned}
&N_1^0(z) = 1\\
&N_1^1(z) = x, \quad N_2^1(z) = y\\
&N_1^2(z) = \tfrac{1}{2}(x^2 - x), \quad N_2^2(z) = xy, \quad N_3^2(z) = \tfrac{1}{2}(y^2 - y)
\end{aligned}$$

werden die Lagrange-Basispolynome wie folgt bestimmt:

1. *Das Lagrange-Basispolynom $L_0(z)$ wird in mehreren Schritten entsprechend modifiziert:*

$$\begin{aligned}
L_0(z) &= N_1^0(z) = 1\\
L_0(z) &= 1 - 1 \cdot x\\
L_0(z) &= (1 - x) - 1 \cdot y\\
L_0(z) &= (1 - x - y) - (-1) \cdot \frac{1}{2}(x^2 - x)\\
L_0(z) &= \left(1 - x - y + \frac{1}{2}(x^2 - x)\right) - (-1) \cdot xy\\
L_0(z) &= \left(1 - x - y + \frac{1}{2}(x^2 - x) + xy\right) - (-1) \cdot \frac{1}{2}(y^2 - y)
\end{aligned}$$

   Das Lagrange-Basispolynom $L_0(z)$ ist also gegeben durch:

$$L_0(z) = 1 + xy + \frac{1}{2}\left(x^2 + y^2 - 3(x + y)\right)$$

2. *Die Lagrange-Basispolynome $L_1(z)$ und $L_2(z)$ können auf folgende Weise konstruiert werden:*

$$\begin{aligned}
&L_1(z) = N_1^1(z) = x && L_2(z) = N_2^1(z) = y\\
&L_1(z) = x - 2 \cdot \tfrac{1}{2}(x^2 - x) && L_2(z) = y - 0 \cdot \tfrac{1}{2}(x^2 - x)\\
&L_1(z) = (2x - x^2) - 1 \cdot xy && L_2(z) = y - 1 \cdot xy\\
&L_1(z) = 2x - x^2 - xy - 0 \cdot \tfrac{1}{2}(y^2 - y) && L_2(z) = (y - xy) - 2 \cdot \tfrac{1}{2}(y^2 - y).
\end{aligned}$$

   Die Lagrange-Basispolynome $L_1(z)$ und $L_2(z)$ sind dann gegeben mit:

$$\begin{aligned}
L_1(z) &= 2x - x^2 - xy\\
L_2(z) &= 2y - y^2 - xy.
\end{aligned}$$

3. *Für die Lagrange-Basispolynome* $L_3(z)$, $L_4(z)$ *und* $L_5(z)$ *gilt:*

$$\begin{aligned} L_3(z) &= N_1^2(z) = \frac{1}{2}\left(x^2 - x\right) \\ L_4(z) &= N_2^2(z) = xy \\ L_5(z) &= N_3^2(z) = \frac{1}{2}\left(y^2 - y\right). \end{aligned}$$

Bemerkung 132 *Ist die Menge I der Interpolationspunkte nicht gleichmäßig verteilt, so bilden die konstruierten Newton- bzw. Lagrange-Basispolynome keine vollständige Basis der quadratischen Polynome. Die reduzierten Polynombasen*

$$\begin{aligned} \bar{N} &= \left\{N_1^0(x), N_1^1(x), ..., N_n^1(x), N_1^2(x), ..., N_{M'}^2(x)\right\} \\ \bar{L} &= \left\{L_0(x), ..., L_{M'+n}(x)\right\} \end{aligned}$$

bilden mit den reellen Koeffizienten $\alpha^0, \alpha_i^1, \alpha_i^2, \alpha_i \in \mathbb{R}$ *linearen Räume* $\Pi\left(\bar{N}\right)$ *und* $\Pi\left(\bar{L}\right)$ *der sub-quadratischen Polynome:*

$$\begin{aligned} \Pi\left(\bar{N}\right) &= \left\{p(x) \mid \quad p(x) = \alpha^0 + \sum_{i=1}^{n} \alpha_i^1 N_i^1(x) + \sum_{i=1}^{M'} \alpha_i^2 N_i^2(x)\right\} \\ \Pi\left(\bar{L}\right) &= \left\{p(x) \mid \quad p(x) = \sum_{i=0}^{M'+n} \alpha_i L_i(x)\right\}. \end{aligned}$$

Satz 133 *Die Polynomräume* $\Pi\left(\bar{N}\right)$ *und* $\Pi\left(\bar{L}\right)$ *sind identisch:*

$$\Pi\left(\bar{N}\right) = \Pi\left(\bar{L}\right).$$

Beweis. Die Lagrange-Basispolynome $L_i(x)$ lassen sich als lineare Kombination der Newton-Basispolynome $N_j^k(x)$ darstellen. Damit folgt:

$$\Pi\left(\bar{L}\right) \subset \Pi\left(\bar{N}\right).$$

Anderseits gilt aber:

$$\left|\bar{N}\right| = \left|\bar{L}\right| = M' + n + 1,$$

so daß die Dimension beiden Räume gleich sein muß:

$$\dim \Pi\left(\bar{N}\right) = \dim \Pi\left(\bar{L}\right) = M' + n + 1.$$

Daraus folgt unmittelbar die Behauptung. ■

Lagrange-Interpolation

Das Interpolationsgleichungssystem (3.12) nimmt bei der Lagrange-Interpolation die einfachste mögliche Form an:

$$I_{M'+n+1}\alpha = f(I).$$

Die Koeffizienten α_i entsprechen also den Funktionswerten $f(z_i)$ und die quadratische Lagrange-Interpolante $m[I]$ einer Funktion f ist dann gegeben mit:

$$m[I] = \sum_{i=0}^{M'-+n} f(z_i) L_i(x).$$

Beispiel 134 *Für die zweidimensionale Zielfunktion f_a aus dem Beispiel 20*

$$f_a(z) = \frac{1}{3}\left(x^3 - y^3\right) - x + y$$

und die Interpolationspunkte aus dem Beispiel 127

$$I := \{(0,0),(1,0),(0,1),(2,0)(1,1)(0,2)\}$$

sind die Funktionswerte gegeben mit:

$$f_a(I) = \left\{0, -\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 0, -\frac{2}{3}\right\}.$$

Mit den im Beispiel 131 bereits konstruierten Lagrange-Basispolynomen $L_i(z)$

$$\begin{aligned}
L_0(z) &= 1 + xy + \frac{1}{2}\left(x^2 + y^2 - 3(x+y)\right)\\
L_1(z) &= 2x - x^2 - xy\\
L_2(z) &= 2y - y^2 - xy\\
L_3(z) &= \frac{1}{2}\left(x^2 - x\right)\\
L_4(z) &= xy\\
L_5(z) &= \frac{1}{2}\left(y^2 - y\right).
\end{aligned}$$

kann das quadratische Modell sofort konstruiert werden:

$$\begin{aligned}
m[I] &= -\frac{2}{3}\left(2x - x^2 - xy\right) + \frac{2}{3}\left(2y - y^2 - xy\right)\\
&\quad + \frac{2}{3}\frac{1}{2}\left(x^2 - x\right) - \frac{2}{3}\frac{1}{2}\left(y^2 - y\right)\\
&= \frac{5}{3}(y - x) + x^2 - y^2
\end{aligned}$$

Die Lagrange-Interpolante $m[I]$ und die Newton-Interpolante $m[Z^0, Z^1, Z^2]$ (vgl. Beispiele 134 und 130) sind bei den konstruierten Basispolynomen auch im subquadratischen Fall identisch.

Bemerkung 135 *Da die Lagrange-Basispolynome $L_i(x)$ $i = 0, 1, ..., n$ im allgemeinen nicht mehr konstant oder linear sein müssen, ist die Trennung des linearen $m[Z^0, Z^1]$ und des (sub-) quadratischen Modells $m[Z^0, Z^1, Z^2]$ der Zielfunktion f bzgl. der Lagrange Basis nicht möglich.*

3.4.2 Modellanpassung

In jedem neuen Schritt des Rekonstruktionsverfahrens sollte das Modell der Zielfunktion neu angepaßt werden. Die Menge I der Interpolationspunkte wird deshalb ständig modifiziert und aus diesem Grund in Form einer Liste implementiert. Die $n+1$ neue Interpolationspunkte werden am Anfang der Liste als $z_0, z_1, ..., z_n$ hinzugeschrieben, die anderen Punkte werden entsprechend verschoben und die $n+1$ Interpolationspunkte, die am Ende der Liste standen, entfernt.
Die neuen Punkte werden als Ecken eines n-dimensionalen Simplex gewählt, dessen Ecke z_0 auf der Trajektorie $T(f)$ und die Seite $\langle z_1, ..., z_n \rangle$ orthogonal zur Trajektorie liegt. Auf diese Weise wird sichergestellt, daß das subquadratische Newton-Modell $m[Z^0, Z^1, Z^2]$ (vgl. Korollar 128) mindestens vollständig linear ist. Das Modell $m(x)$ enthält also alle linearen Terme N_i^1, auch wenn einige Koeffizienten α_i^1 *Null* sein dürfen.

Ausgangssimplex

Für die Bestimmung erster Interpolationspunkte wird ein n-dimensionaler Simplex $S := \{s_0, ..., s_n\}$ mit dem Mittelpunkt x_0 benötigt. Die $n+1$ Ecken $s_0, ..., s_n$ zusammen mit den Kantenmittelpunkten s_{ij},

$$s_{ij} = \frac{s_i + s_j}{2}, \qquad i \neq j, \quad i, j = 0, ..., m,$$

bilden eine vollständige, bzgl. jeder beliebigen Polynom-Basis P gleichmäßig verteilte Interpolationsmenge.

Beispiel 136 *Für einen Punkt $x_0 = (0)_n$ sind die Ecken $s_0, ..., s_n$ eines regelmäßigen Simplex S gegeben mit*

$$\begin{aligned} s_0 &:= -\left(\frac{\sqrt{n}}{n}\right)_n \\ s_j &:= \left(\frac{1-\sqrt{1+n}}{n^{\frac{3}{2}}}\right)_n + \frac{n\sqrt{1+n}}{n^{\frac{3}{2}}} e_j, \qquad j = 1, ..., n. \end{aligned}$$

Mit $(a)_n$ wird hier ein $n-$dimensionaler Vektor bezeichnet, dessen Einträge alle gleich a sind. Im zweidimensionalen Fall (s. Abbildung 3.4) ist der Simplex S gegeben mit:

$$S = \left\{ \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right)^T, \left(\frac{1-\sqrt{3}}{2\sqrt{2}}, \frac{1+\sqrt{3}}{2\sqrt{2}} \right)^T, \left(\frac{1+\sqrt{3}}{2\sqrt{2}}, \frac{1-\sqrt{3}}{2\sqrt{2}} \right)^T \right\}.$$

Es gilt weiterhin:

$$\|s_j - x_0\| = \|s_j\| = 1, \qquad j = 0, ..., n.$$

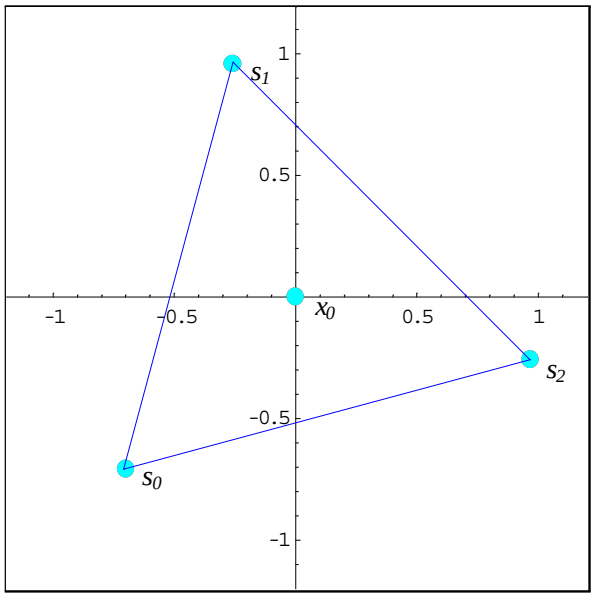

Abbildung 3.4: Regulärer Simplex S mit dem Mittelpunkt $x_0 = (0,0)^T$

Bestimmung neuer Interpolationspunkte

Seien x_{i-1} und x_i zwei im Rekonstruktionsverfahren nacheinander gefundene Trajektorienpunkte. Die Interpolationspunkte $z_0, z_1, ..., z_n$ werden dann wie folgt bestimmt.
Der Simplex S wird mit Hilfe einer Transformation $\mathcal{S}_i$ so gespiegelt, daß die folgenden normierten Differenzenvektoren d_i und $\mathcal{S}_i h$ sich überdecken:

$$d_i = \frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|} \qquad h = \frac{x_0 - s_0}{\|x_0 - s_0\|}$$
$$\mathcal{S}_i h = d_i.$$

Mit Hilfe einer Dehnung $\mathcal{D}_i$ mit dem Zentrum s_0 und den Dehnungskoeffizienten $p \in \mathbb{R}_+$ (abhängig von der Schrittlänge des Rekonstruktionsverfahrens) wird dann der Simplex S auf die gewünschte Größe gebracht:

$$\|\mathcal{D}_i \mathcal{S}_i (s_{i.} - x_0)\| = p.$$

Anschließend wird die Ecke s_0 des Simplex S durch eine Translation $\mathcal{T}_i$ auf den Punkt x_i verschoben:

$$\mathcal{T}_i \mathcal{D}_i \mathcal{S}_i s_0 = x_i.$$

Die neuen Interpolationspunkte $z_0, z_1, ..., z_n$ sind dann gegeben mit

$$z_j := \mathcal{T}_i \mathcal{D}_i \mathcal{S}_i s_j, \qquad j = 0, ..., n.$$

Beispiel 137 *Für den $n-$dimensionalen Simplex S aus dem Beispiel 136 ist der Vektor h gleich dem Vektor $-s_0$:*

$$h = -s_0 = \left(\frac{\sqrt{n}}{n}\right)_n.$$

Die Transformationen $\mathcal{T}_i$, $\mathcal{S}_i$ und $\mathcal{D}_i$ sind dann gegeben durch:

$$\begin{aligned} u_j &\leftarrow \mathcal{S}_i s_j = s_j - 2ww^T s_j, \qquad w := \frac{d_i - h}{\|d_i - h\|} \\ v_j &\leftarrow \mathcal{D}_i u_j = p u_j \\ z_j &\leftarrow \mathcal{T}_i v_j = v_j + x_i. \end{aligned}$$

Die Interpolationspunkte $z_0, z_1, ..., z_n$ können mit geringem Aufwand bestimmt werden durch:

$$\begin{aligned} z_0 &= x_i \\ z_j &= x_i + p\left(s_j - 2ww^T s_j\right). \end{aligned}$$

Im zweidimensionalen Fall seien die Trajektorienpunkte x_{i-1} und x_i und die gewünschte Schrittlänge p gegeben mit:

$$\begin{aligned} x_{i-1} &= (0.9, 1)^T \\ x_i &= (1, 1)^T \\ p &= 0.2. \end{aligned}$$

Die Differenzenvektoren d_i und w_i sind gegeben durch:

$$\begin{aligned} d_i &= (1, 0)^T \\ w_i &= \frac{\left(1 - \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)^T}{\sqrt{2 - \sqrt{2}}}. \end{aligned}$$

Die Matrix $S_i = I - 2w_i w_i^T$ der Spiegelung $\mathcal{S}_i$ kann leicht berechnet werden:

$$\begin{aligned} S_i &= I - 2\frac{\left(1-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)\left(1-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)^T}{2-\sqrt{2}} \\ &= \frac{1}{2}\begin{pmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{pmatrix}. \end{aligned}$$

Es gilt dann:

$$S_i h = \frac{1}{4}\begin{pmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{pmatrix}\begin{pmatrix} \sqrt{2} \\ \sqrt{2} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = d_i.$$

Die gespiegelten Vektoren $u_j = \mathcal{S}_i s_j$ sind also gegeben mit:

$$\begin{aligned} u_1 &= \frac{1}{2}\begin{pmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{pmatrix}\left(\frac{1-\sqrt{3}}{2\sqrt{2}}, \frac{1+\sqrt{3}}{2\sqrt{2}}\right)^T = \frac{1}{2}\left(1, -\sqrt{3}\right)^T \\ u_2 &= \frac{1}{2}\begin{pmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{pmatrix}\left(\frac{1+\sqrt{3}}{2\sqrt{2}}, \frac{1-\sqrt{3}}{2\sqrt{2}}\right)^T = \frac{1}{2}\left(1, \sqrt{3}\right)^T. \end{aligned}$$

Die neuen Interpolationspunkte z_0, z_1 und z_2 können dementsprechend wie folgt bestimmt werden:

$$\begin{aligned} z_0 &= (1,1)^T \\ z_1 &= (1,1)^T + 0.2 \cdot \frac{1}{2}\left(1, -\sqrt{3}\right)^T = \left(1.1, 1 - \frac{\sqrt{3}}{10}\right)^T \\ z_2 &= (1,1)^T + 0.2 \cdot \frac{1}{2}\left(1, \sqrt{3}\right)^T = \left(1.1, 1 + \frac{\sqrt{3}}{10}\right)^T. \end{aligned}$$

Der neue Trajektorienpunkt x_{i+1} wird dann auf der durch Punkte $z_1, ...z_n$ festgelegten Hyperebene und im zweidimensionalen Fall auf der Gerade $z_1 z_2$ gesucht. Dabei werden sowohl das lineare Modell $m[Z^0, Z^1]$ (Punkt x^l_{i+1}) als auch das quadratische Modell $m[Z^0, Z^1, Z^2]$ (Punkt x^q_{i+1}) berücksichtigt (s. Abbildung 3.5).

Modellkontrolle

Die Qualität der aufgestellten quadratischen Modelle wird hierbei durch Vergleich mit dem linearen Modell geprüft. Der Rekonstruktionsschritt wird

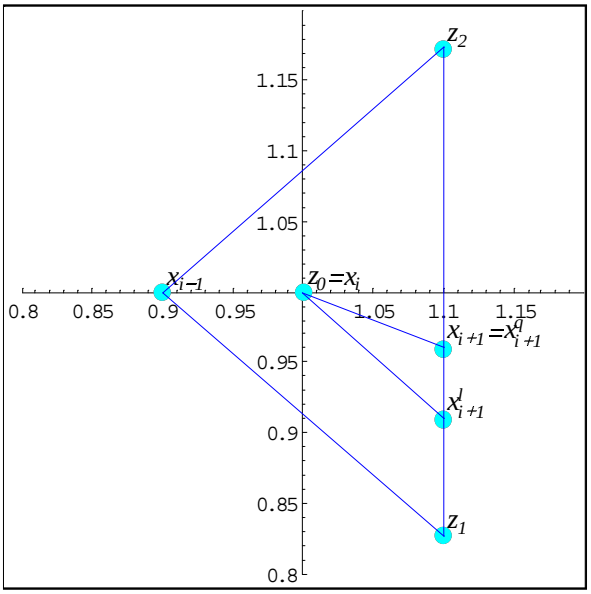

Abbildung 3.5: Bestimmung der Interpolationspunkte für die Surrogate-Funktion

also nur dann akzeptiert, wenn die mit Hilfe der beiden Modelle gefundenen Trajektorienpunkte x^l_{i+1} und x^q_{i+1} nah beinander liegen, bzw. der Winkel zwischen den entsprechenden Rekonstruktionsschritten $h^l = x^l_{i+1} - x_i$ und $h^q = x^q_{i+1} - x_i$ nicht zu groß ist (s. Abbildung 3.5).
Wird eine deutliche Diskrepanz festgestellt, so muß der Rekonstruktionsschritt verkürzt wiederholt werden.

Kapitel 4

Numerische Ergebnisse

Die wichtigste Frage, die nach der abgeschlossenen Rekonstruktion einer Trajektorie gestellt werden muß, lautet:

Wurden alle relevanten (kritischen) Punkte
für das vorgegebene Problem gefunden?

Die positive Antwort auf diese Frage kann im allgemeinen nicht gesichert werden. Stattdessen wird die Frage gestellt:

Mit welchem der beschriebenen Ansätze
sind die besten Ergebnisse zu erzielen?

In diesem Kapitel werden die numerischen Testergebnisse für die vorgeschlagenen Ansätze unter Berücksichtigung grundlegender Fragestellungen vorgestellt. Hierzu werden die in den Abschnitten 1.3 und 1.4.4 beschriebenen klassischen Newton-Trajektorien und die Richtungsfeld-Trajektorien mit mehreren Startpunkten gegenübergestellt (s. Abschnitt 4.2). Für die zufällig gewählten Startpunkte werden die entsprechenden Trajektorien konstruiert und die Anzahl der Trajektorienkomponenten und der gefundenen kritischen Punkte bestimmt.
Für die Rekonstruktion der Trajektorien werden die in Abschnitt 2.2 vorgeschlagenen Strategien eingesetzt und verglichen. Die numerischen Testergebnisse hierzu werden in Abschnitt 4.3 vorgestellt.

In Abschnitt 4.4 werden die Einsatzmöglichkeiten und der Berechnungsaufwand der in Kapitel 3 beschriebenen Verfahren diskutiert und verglichen.

Zum Schluß werden kurz Optimierungsprobleme mit Restriktionen behandelt. Es wird geprüft, ob mit der in Abschnitt 1.4.4 vorgeschlagenen Richtungsfeld-Trajektorie die relevanten Randpunkte des zulässigen Bereiches erreicht werden (s. Abschnitt 4.6).

Die Testprobleme für die Verifikation der gestellten Aussagen wurden den Arbeiten [19] und [21] entnommen.

4.1 Testprobleme ohne Restriktionen

4.1.1 Zweidimensionale Testfunktionen

1. Die Six-hump-camelback Funktion $f_c : \mathbb{R}^2 \to \mathbb{R}$ ist definiert durch:

$$f_c(z) = \frac{1}{3}x^6 - 2.1x^4 + 4x^2 + xy - 4y^2 + 4y^4.$$

Diese Testfunktion hat 15 kritische Punkte in dem Bereich $B = [-2.5, 2.5]^2$. Für den Minimalabstand $\delta_{\min}$ zwischen den kritischen Punkten gilt:

$$\delta_{\min} > 0.3.$$

Die Funktion f_c ist von unten, aber nicht von oben beschränkt und besitzt ein globales Minimum:

$$z_{\min} = \begin{pmatrix} 0, & 0 \end{pmatrix}^T, \quad f_c(z_{\min}) = 0 \ .$$

Der Gradient der Testfunktion ist gegeben durch:

$$\nabla f_c(z) = \begin{pmatrix} 2x^5 - 8.4x^3 + y, & x - 8y + 16y^3 \end{pmatrix}^T.$$

2. Die Testfunktion $f_e : \mathbb{R}^2 \to \mathbb{R}$ ist definiert durch:

$$f_e(z) = \sum_{i=1}^{m} \alpha_i e^{\lambda_i \left(\|z - b_i\|_2^2 \right)},$$

mit $m = 5$ und den Vektorparammetern:

$$\begin{aligned} \alpha &= \begin{pmatrix} 2, & 3, & 1, & 4, & 2 \end{pmatrix}^T \\ \lambda &= \begin{pmatrix} -1, & -2, & -3, & -3, & -2 \end{pmatrix}^T. \end{aligned}$$

Die Knotenpunkte b_i sind gegeben durch

$$\begin{aligned} b_1 &= \begin{pmatrix} -2, & 0 \end{pmatrix}^T \\ b_2 &= \begin{pmatrix} 3, & 0 \end{pmatrix}^T \\ b_3 &= \begin{pmatrix} 1, & 2 \end{pmatrix}^T \\ b_4 &= \begin{pmatrix} 0, & 2 \end{pmatrix}^T \\ b_5 &= \begin{pmatrix} 0, & -1 \end{pmatrix}^T. \end{aligned}$$

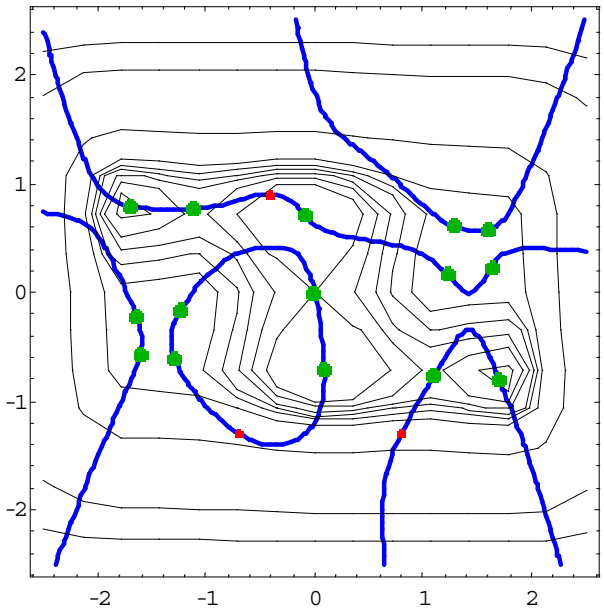

Abbildung 4.1: 3-Punkte-Beispieltrajektorie für die Testfunktion f_c;
Startpunkte: $\{\{-0.4, 0.9\}, \{-0.7, -1.3\}, \{0.8, -1.3\}\}$

Diese Testfunktion f_e hat neun kritische Punkte in dem Bereich $B = [-5, 5]^2$.
Für den Minimalabstand $\delta_{\min}$ zwischen den kritischen Punkten gilt:

$$\delta_{\min} > 0.95.$$

Die Funktion f_e ist sowohl von unten als auch von oben beschränkt und besitzt ein globales Minimum:

$$z_{\min} \cong \left(\ 1.10, \ \ 0.52 \ \right)^T, \quad f_e\left(z_{\min}\right) \cong 0.0046 \ ,$$

und ein globales Maximum:

$$\left(\ 0.01, \ \ 2.00 \ \right)^T, \quad f_e\left(z_{\max}\right) \cong 4.0524 \ .$$

Der Gradient der Testfunktion ist gegeben durch:

$$\nabla f_e\left(z\right) = 2 \sum_{i=1}^{m} \alpha_i \lambda_i e^{\lambda_i \left(\|x - b_i\|_2^2 \right)} \left(x - b_i\right).$$

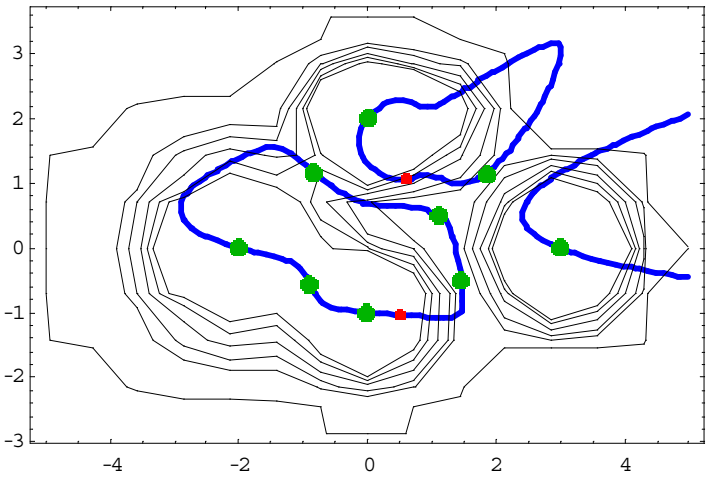

Abbildung 4.2: 2-Punkte-Beispieltrajektorie für die Testfunktion f_e; Startpunkte: $\{\{0.6, 1.\}, \{0.5, -1.\}\}$

3. Die Testfunktion $f_t : [0,1]^2 \to \mathbb{R}$ ist die Fehlerfunktion einer zweidimensionalen linearen Tschebyscheff-Approximation und ist definiert durch:

$$\begin{aligned} f_t(z) \;=\; & -x(1-x)\,y(1-y) + c_1 \sin(\pi x)\sin(\pi y) \\ & + c_2\left(\sin(\pi x)\sin(3\pi y) + \sin(3\pi x)\sin(\pi y)\right) \\ & + c_3 \sin(3\pi x)\sin(3\pi y) \\ & + c_4\left(\sin(\pi x)\sin(5\pi y) + \sin(5\pi x)\sin(\pi y)\right) \end{aligned}$$

mit dem Parametervektor

$$c = \left(\; 0.066581,\;\; 0.002503,\;\; 0.000086,\;\; 0.000559 \;\right)^T.$$

Diese Testfunktion f_t hat 53 kritische Punkte in dem Bereich $B = [0,1]^2$.
Für den Minimalabstand $\delta_{\min}$ zwischen den kritischen Punkten gilt:

$$\delta_{\min} > 0.05.$$

Die Funktion f_t ist im allgemeinen weder von unten noch von oben beschränkt und besitzt keine globale Extrema. Da die Funktion f_t hier nur auf dem Quadrat B definiert und auf dem Rand ∂B des Bereiches B konstant ist:

$$f|_{\partial B}(z) = 0,$$

besitzt sie vier globale Minima:

$$\begin{array}{ll} z^1_{\min} \cong \left(\ 0.35,\ \ 0.35\ \right)^T & z^2_{\min} \cong \left(\ 0.65,\ \ 0.35\ \right)^T \\ z^3_{\min} \cong \left(\ 0.35,\ \ 0.65\ \right)^T & z^4_{\min} \cong \left(\ 0.65,\ \ 0.65\ \right)^T \\ f\left(z^i_{\min}\right) \cong -0.0002984 & \end{array}$$

und vier globale Maxima:

$$\begin{array}{ll} z^1_{\max} \cong \left(\ 0.18,\ \ 0.50\ \right)^T & z^2_{\max} \cong \left(\ 0.50,\ \ 0.18\ \right)^T \\ z^3_{\max} \cong \left(\ 0.82,\ \ 0.50\ \right)^T & z^4_{\max} \cong \left(\ 0.50,\ \ 0.82\ \right)^T \\ f\left(z^i_{\max}\right) \cong 0.0003052. & \end{array}$$

Der Gradient der Testfunktion sind gegeben durch:

$$\begin{aligned} \frac{\partial f_t(z)}{\partial x} &= (2x-1)\,y\,(1-y) + \pi c_1 \cos(\pi x)\sin(\pi y) \\ &\quad + \pi c_2\left(\cos(\pi x)\sin(3\pi y) + 3\cos(3\pi x)\sin(\pi y)\right) \\ &\quad + \pi c_3 \cos(3\pi x)\sin(3\pi y) \\ &\quad + \pi c_4\left(\cos(\pi x)\sin(5\pi y) + 5\cos(5\pi x)\sin(\pi y)\right) \end{aligned}$$

$$\begin{aligned} \frac{\partial f_t(z)}{\partial y} &= (2y-1)\,x\,(1-x) + \pi c_1 \sin(\pi x)\cos(\pi y) \\ &\quad + \pi c_2\left(\sin(3\pi x)\cos(\pi y) + 3\sin(\pi x)\cos(3\pi y)\right) \\ &\quad + \pi c_3 \sin(3\pi x)\cos(3\pi y) \\ &\quad + \pi c_4\left(\sin(5\pi x)\cos(\pi y) + 5\sin(\pi x)\cos(5\pi y)\right) \end{aligned}$$

4. Die Goldstein-Price-Funktion $f_g : \mathbb{R}^2 \to \mathbb{R}$ (s. Problem 4.2 in [21]) ist definiert durch:

$$\begin{aligned} f_g(z) &= \left(1 + (x+y+1)^2\left(19 - 14x + 3x^2 - 14y + 6xy + 3y^2\right)\right) \\ &\quad \cdot \left(30 + (2x-3y)^2\left(18 - 32x + 12x^2 + 48y - 36xy + 27y^2\right)\right). \end{aligned}$$

Diese Standardtestfunktion hat neun kritische Punkte in dem Bereich $B = [-2.0, 2.0]^2$. Für den Minimalabstand $\delta_{\min}$ zwischen den kritischen Punkten gilt:

$$\delta_{\min} > 0.25.$$

Die Funktion f_c ist von unten, aber nicht von oben beschränkt und besitzt ein globales Minimum:

$$z_{\min} = (0, -1)^T, \quad f_c(z_{\min}) = 3.$$

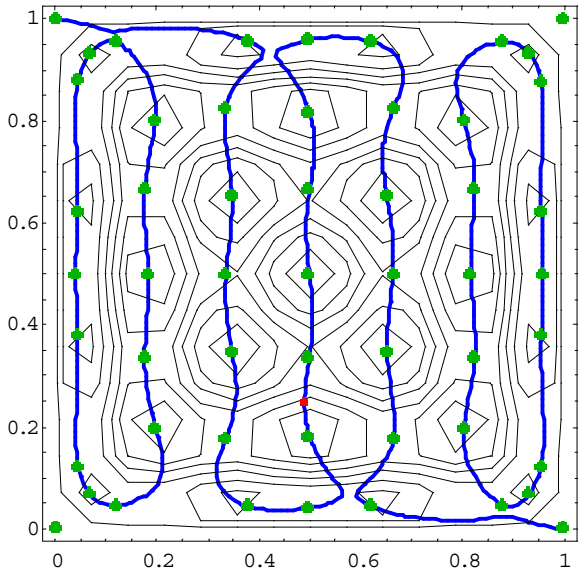

Abbildung 4.3: Beispieltrajektorie für die Testfunktion f_t;
Startpunkt: $\{0.49, 0.25\}$

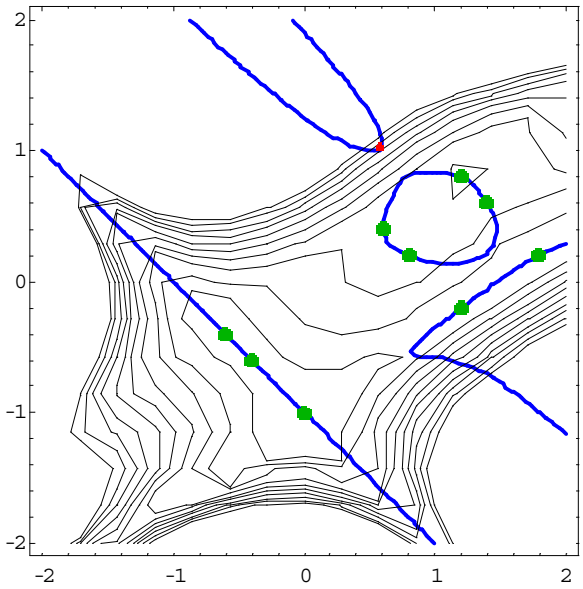

Abbildung 4.4: 1-Punkt-Beispieltrajektorie für die Testfunktion f_g;
Startpunkt: $\{0.6, 1.0\}$

4.1.2 Mehrdimensionale Testfunktionen

1. Die vierdimensionalen Shekel-Testfunktionen $f_{s,m} : \mathbb{R}^4 \to \mathbb{R}$ sind definiert durch:

$$f_{s,m}(x) = -\sum_{i=1}^{m} \left(\|x - a_i\|_2^2 + c_i \right)^{-1},$$

mit dem Parametervektor:

$$c = \left(\begin{array}{cccccccccc} 0.1, & 0.2, & 0.2, & 0.4, & 0.4, & 0.6, & 0.3, & 0.7, & 0.5, & 0.5 \end{array} \right)^T.$$

Die Knotenpunkte a_i sind gegeben durch:

$$\begin{array}{ll} a_1 = \left(\begin{array}{cccc} 4, & 4, & 4, & 4 \end{array} \right)^T, & a_6 = \left(\begin{array}{cccc} 2, & 9, & 2, & 9 \end{array} \right)^T \\ a_2 = \left(\begin{array}{cccc} 1, & 1, & 1, & 1 \end{array} \right)^T, & a_7 = \left(\begin{array}{cccc} 5, & 5, & 3, & 3 \end{array} \right)^T \\ a_3 = \left(\begin{array}{cccc} 8, & 8, & 8, & 8 \end{array} \right)^T, & a_8 = \left(\begin{array}{cccc} 8, & 1, & 8, & 1 \end{array} \right)^T \\ a_4 = \left(\begin{array}{cccc} 6, & 6, & 6, & 6 \end{array} \right)^T, & a_9 = \left(\begin{array}{cccc} 6, & 2, & 6, & 2 \end{array} \right)^T \\ a_5 = \left(\begin{array}{cccc} 3, & 7, & 3, & 7 \end{array} \right)^T, & a_{10} = \left(\begin{array}{cccc} 7, & 3.6, & 7, & 3.6 \end{array} \right)^T. \end{array}$$

Die Shekel-Funktionen $f_{s,m}$ mit den üblichen Werten von $m = 5$, 7 und 10 werden in der Literatur auch kurz mit $SQRN5$, $SQRN7$ bzw. $SQRN10$ bezeichnet. Die Anzahl der kritischen Punkte in dem Bereich $B = [0, 12]^4$ sowie den Minimalabstand $\delta_{\min}$ dazwischen kann für diese Funktionen aus der folgenden Tabelle abgelesen werden:

m	AKP	$\delta_{\min} >$
5	11	0.55
7	13	0.90
10	21	0.75

.

Die Funktionen $f_{s,m}$ sind sowohl von oben als auch von unten beschränkt und besitzt je ein globales Maximum und ein globales Minimum:

$$\begin{array}{c} m = 5 \\ x_{\min} \cong \left(\begin{array}{cccc} 4.000037, & 4.000133, & 4.000037, & 4.000133 \end{array} \right)^T \\ f_{s,5}(x_{\min}) \cong -10.1532 \\ x_{\max} \cong \left(\begin{array}{cccc} 2.402862, & 2.494488, & 2.402862, & 2.494491 \end{array} \right)^T \\ f_{s,5}(x_{\max}) \cong -0.2707. \end{array}$$

$$m = 7$$

$$x_{\min} \cong \left(\begin{array}{cccc} 4.000573, & 4.000689, & 3.999490, & 3.999606 \end{array} \right)^T$$
$$f_{s,7}\left(x_{\min}\right) \cong -10.4029$$
$$x_{\max} \cong \left(\begin{array}{cccc} 2.436458, & 2.528014, & 2.132607, & 2.224163 \end{array} \right)^T$$
$$f_{s,7}\left(x_{\max}\right) \cong -0.3534.$$

$$m = 10$$

$$x_{\min} \cong \left(\begin{array}{cccc} 4.000747, & 4.000593, & 3.999663, & 3.999510 \end{array} \right)^T$$
$$f_{s,10}\left(x_{\min}\right) \cong -10.5364$$
$$x_{\max} \cong \left(\begin{array}{cccc} 2.496845, & 2.326900, & 2.248851, & 2.078920 \end{array} \right)^T$$
$$f_{s,10}\left(x_{\max}\right) \cong -0.4251.$$

Die Gradientenfunktionen der Testfunktion $f_{s,m}$ sind gegeben durch:

$$\nabla f_{s,m} = 2 \sum_{i=1}^{m} \left(\|x - a_i\|_2^2 + c_i \right)^{-2} (x - a_i) .$$

2. Die Testfunktion $f_p : \mathbb{R}^8 \to \mathbb{R}$ ist definiert durch:

$$f_p(x) = \prod_{i=1}^{4} \|x - a_i\|_2^2 .$$

Die Knotenpunkte a_i, $i = 1, ..., 4$ sind gegeben durch:

$$\begin{aligned} a_1 &= \left(\begin{array}{cccccccc} 1, & 1, & 2, & 2, & 0, & -1, & 0, & -1 \end{array} \right)^T \\ a_2 &= \left(\begin{array}{cccccccc} -2, & -2, & -1, & 1, & 1, & 0, & 1, & 2 \end{array} \right)^T \\ a_3 &= \left(\begin{array}{cccccccc} 0, & -1, & 0, & 0, & -2, & 1, & -2, & -2 \end{array} \right)^T \\ a_4 &= \left(\begin{array}{cccccccc} -1, & 2, & 0, & 1, & 1, & -2, & 1, & -1 \end{array} \right)^T \end{aligned}$$

Die Testfunktion f_p hat sieben kritische Punkte in dem Bereich $B = [-3, 3]^8$. Für den Minimalabstand $\delta_{\min}$ zwischen den kritischen Punkten gilt:

$$\delta_{\min} > 0.75.$$

Die Funktion f_p ist von unten, aber nicht von oben beschränkt und besitzt vier globale Minima in den Knotenpunkten a_i. Der Gradient der Testfunktion f_p ist gegeben durch:

$$\nabla f_p(x) = 2 \sum_{i=1}^{4} \left((x - a_i) \prod_{\substack{j=1 \\ j \neq i}}^{4} \|x - a_i\|_2^2 \right) .$$

3. Testfunktion $f_d : \mathbb{R}^n \to \mathbb{R}$ ist definiert durch:

$$f_d(x) = \sum_{i=1}^{d} \arctan\left(\|x - a_i\|_2^2\right).$$

Die Anzahl der Knotenpunkte und die Punkte selbst sind nach einem Zufallsprinzip gewählt. Die Anzahl der kritischen Punkte der Funktion f_d sowie deren Minimalabstand $\delta_{\min}$ sind von den Knotenpunkten abhängig. Alle kritischen Punkte der Funktion f_d liegen in der konvexen Hülle der Knotenpunkte a_i:

$$B = \langle\ a_1,\ \ ...,\ \ a_d\ \rangle.$$

Die Funktion f_d ist von unten und von oben beschränkt. Die Funktion besitzt zwar globale Minima, aber keine globale Maxima. Der Gradient der Testfunktion ist gegeben durch:

$$\nabla f_d(x) = \sum_{i=1}^{d} \frac{2(x - a_i)}{1 + \|x - a_i\|_2^4}.$$

Eine Familie $n-$dimensionaler Testfunktionen

Im weiteren wird hier eine Familie $\mathcal{F}^B$ n-dimensionaler Testfunktionen folgender Form konstruiert:

$$\mathcal{F}^B := \left\{ f^B : \mathbb{R}^n \to \mathbb{R} | \qquad f^B(x) = \phi(x_1) + \sum_{i=1}^{n-1} \left(x_{i+1} - \psi_i(x_i)\right)^2 \right\},$$

mit den eindimensionalen reellwertigen Parameterfunktionen ϕ, $\psi_1, ..., \psi_{n-1}$. Die partielle Ableitungsfunktionen sind dann gegeben durch:

$$\begin{aligned}
\frac{\partial f^B(x)}{\partial x_1} &= \phi'(x_1) - 2(x_2 - \psi_1(x_1))\,\psi_1'(x_1) \\
\frac{\partial f^B(x)}{\partial x_2} &= 2(x_2 - \psi_1(x_1)) - 2(x_3 - \psi_2(x_2))\,\psi_2'(x_2) \\
&\ ... \\
\frac{\partial f^B(x)}{\partial x_{n-1}} &= 2\left(x_{n-1} - \psi_{n-2}(x_{n-2})\right) - 2\left(x_n - \psi_{n-1}(x_{n-1})\right)\psi_{n-1}'(x_{n-1}) \\
\frac{\partial f^B(x)}{\partial x_n} &= 2\left(x_n - \psi_{n-1}(x_{n-1})\right).
\end{aligned}$$

Durch Rückwärtssubstitution kann das Gleichungssystem $\nabla f^B(x) = 0$ leicht gelöst werden. Für die kritischen Punkte x^* der Testfunktion f^B gilt dann:

$$\begin{aligned} x_n^* &= \psi_{n-1}\left(x_{n-1}^*\right) \\ &\ldots \\ x_2^* &= \psi_1\left(x_1^*\right) \\ \phi'(x_1) &= 0. \end{aligned}$$

Die Testfunktion f^B hat also genau so viele kritische Punkte wie die Parameterfunktion ϕ. Für jeden kritischen Punkt gilt außerdem:

$$f^B(x^*) = \phi\left(x_1^*\right).$$

Ist die Funktion ϕ von unten beschränkt und besitzt ein globales Minimum x_1^*, so ist die Funktion $f^B(x)$ auch von unten beschränkt und besitzt ein globales Minimum:

$$x^* = \left(\; x_1^*, \;\; \psi_1\left(x_1^*\right), \;\; \ldots, \;\; \psi_{n-1}\left(x_{n-1}^*\right) \; \right)^T.$$

1. Für die eindimensionalen Parameterfunktionen

$$\begin{aligned} \phi(\xi) &= \frac{1}{32}\left(\frac{32}{7}\xi^7 - \frac{48}{5}\xi^5 + \frac{18}{3}\xi^3 - x\right) \\ \psi_i(\xi) &= \frac{1}{3}\sin(\xi), \end{aligned}$$

hat die entsprechende $f_b \in \mathcal{F}^B$ folgende Eigenschaften:

(a) Die Ableitungsfunktion der Parameterfunktion $\phi(\xi)$ ist der Tschebyscheff-Polynom sechsten Grades:

$$\phi'(\xi) = T_6(\xi) = \frac{1}{32}\left(32\xi^6 - 48\xi^4 + 18\xi^2 - 1\right)$$

und besitzt folgende sechs reelle Nullstellen:

$$\xi_k^* = \cos\left(\frac{2k+1}{12}\pi\right), \qquad k = 0, \ldots, 5.$$

Die Funktion f_b hat also sechs kritische Punkte.

(b) Es kann leicht nachgeprüft werden, daß alle kritischen Punkte der Funktion f_b in dem Bereich $B = [-1, 1]^n$ liegen.

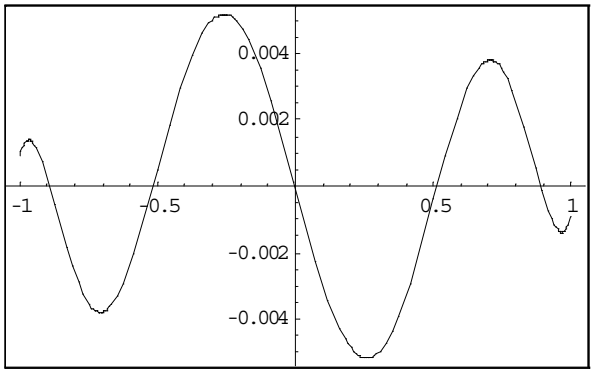

Abbildung 4.5: Stammfunktion des Tschebyscheff-Polynomes T_6

(c) Die Parameterfunktion ϕ hat ein globales Minimum (vgl. Abbildung 4.5):

$$\xi_{\min} \cong 0.2588176, \quad \phi(\xi_{\min}) \cong -0.0051746.$$

Die Funktion f_b besitzt ein globales Minimum:

$$x_{\min} \cong [\ 0.2588176, \quad \ldots\ ], \quad f_b(x_{\min}) \cong -0.0051746.$$

2. Für die eindimensionalen Parameterfunktionen

$$\begin{aligned} \phi(\xi) &= \frac{1}{24}\left(\xi^4 - 16\xi^3 + 72\xi^2 - 96x + 24\right) \\ \psi_i(\xi) &= 10 - \xi, \end{aligned}$$

hat die entsprechende $f_b \in \mathcal{F}^B$ folgende Eigenschaften:

(a) Die Parameterfunktion $\phi(\xi)$ ist der Laguerr-Polynom vierten Grades und hat folgende drei Extrema (s. Abbildung 4.6):

$$\begin{aligned} \xi_1^* &= 0.935822 \\ \xi_2^* &= 3.305410 \\ \xi_3^* &= 7.758770 \end{aligned}$$

Die Funktion f_b hat also drei kritische Punkte.

(b) Es kann leicht nachgeprüft werden, daß alle kritischen Punkte der Funktion f_b in dem Bereich $B = [0, 10]^n$ liegen.

(c) Die Parameterfunktion ϕ hat ein globales Minimum (vgl. Abbildung 4.6):

$$\xi_{\min} \cong 7.758770, \quad \phi(\xi_{\min}) \cong -9.82295.$$

Die Funktion f_b besitzt ein globales Minimum:

$$x_{\min} \cong [\ 7.758770, \quad \ldots\ ], \quad f_b(x_{\min}) \cong -9.82295.$$

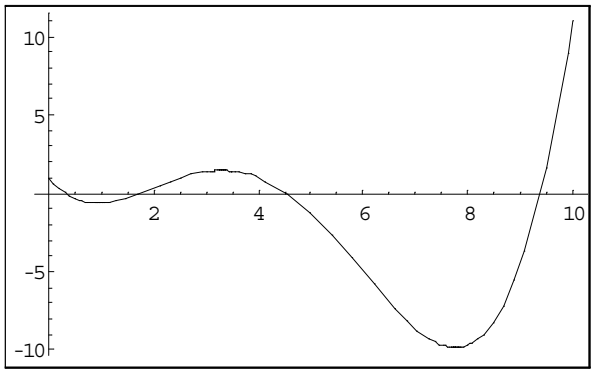

Abbildung 4.6: Laguerre-Polynom L_4

4.2 Gegenüberstellung der Trajektorien mit mehreren Startpunkten

Anhand der zweidimensionalen Testfunktionen f_c (neun kritische Punkte) und f_e (15 kritische Punkte) wurden in jeweils 100 Durchläufen mit den zufallsgenerierten Startpunkten die $1-$ bis $3-$Punkt-Trajektorien verglichen. Zusätzlich wurden 100 (M)3-Punkt-Trajektorien berechnet, wobei die Startpunkte aus drei disjunkten gleichmäßig verteilten Sektoren gewählt wurden. Als Kennzahlen werden hier die Trefferquote TQ und die Anzahl der Komponenten AK wie folgt definiert:

TQ : Anzahl der gefundenen kritischen Punkte,

AK : Anzahl der Komponenten der Haupttrajektorie.

Die Durchschnittswerte der Kennzahlen TQ und AK wurden mit der entsprechenden Standardabweichung in der Tabellen 4.1 und 4.2 dargestellt.

Testfunktion	GP-Startegie	$1-P$	$2-P$	$3-P$	$M3-P$
f_c	A-Strategie	13.6 ± 3.6	13.5 ± 3.2	13.6 ± 3.2	14.7 ± 4.0
	B-Strategie	12.5 ± 4.4	12.8 ± 3.6	13.1 ± 3.2	14.6 ± 4.0
f_e	A-Strategie	8.2 ± 1.8	8.7 ± 1.2	8.6 ± 1.2	8.9 ± 0.5
	B-Strategie	7.6 ± 2.5	8.0 ± 2.0	8.0 ± 1.2	8.5 ± 1.3

Tabelle 4.1: Vergleich der Trajektorien (Anzahl der gefundenen kritischen Punkte)

Testfunktion	$1-P$	$2-P$	$3-P$	$M3-P$
f_c	1.7 ± 1.2	3.2 ± 1.2	3.3 ± 1.1	3.2 ± 1.1
f_e	2.3 ± 1.1	3.1 ± 1.0	3.1 ± 1.1	3.0 ± 1.1

Tabelle 4.2: Vergleich der Trajektorien / Anzahl der Komponenten

Für die vierdimensionale Testfunktion f_s (elf kritische Punkte) wurden jeweils zwanzig $1-$ und $4-$Punkt-Trajektorien berechnet. Die durchschnittliche Trefferquote TQ und die Gesamtlaufzeit LZ der Rekonstruktion wurden in der Tabelle 4.3 präsentiert.

In Abschnitt 1.4.3 wurde die Richtungsfeld-Trajektorie mit mehreren vorgegebenen Punkten eingeführt. Durch die Vorgabe von bis zu $n+1$ (n - Dimension des Problems) Startpunkten sollte dem Anwender die Möglichkeit gegeben werden, direkten Einfluß auf den Verlauf der Trajektorie zu nehmen. Die Trefferquote TQ des Trajektorienverfahrens (Anzahl der gefundenen kritischen Punkte) sollte dadurch verbessert werden. Anhand der durchgeführten Tests konnte dies tatsächlich bestätigt werden (vgl. Tabellen 4.1, 4.4 und 4.3).

Die Vorteile der Mehr-Punkt-Trajektorien konnten durch das folgende Experiment gezeigt werden. Für die $5-$Punkt-Trajektorien der Testfunktion f_s wurden gleichmäßig in dem gesamten Bereich verteilte zufällig generierte

	$1-P$		$4-P$	
Testfunktion	TQ	LZ [min]	TQ	LZ [min]
f_s	9.8 ± 2.2	24.4 ± 17.1	10.5 ± 0.9	86.1 ± 22.9

Tabelle 4.3: Mehrpunkttrajektorien für die Testfunktion f_s

Startpunkte genommen:

$$\begin{aligned} s_0 &= \left(1.8,\ 1.6,\ 1.9,\ 1.4 \right)^T, \\ s_1 &= \left(11.0,\ 4.4,\ 4.8,\ 3.9 \right)^T, \\ s_2 &= \left(4.2,\ 10.0,\ 2.6,\ 4.0 \right)^T, \\ s_3 &= \left(2.8,\ 5.9,\ 7.6,\ 3.6 \right)^T, \\ s_4 &= \left(3.1,\ 3.7,\ 5.6,\ 9.0 \right)^T. \end{aligned}$$

Die fünf Punkte führten sofort zu fünf Komponenten, die alle elf kritischen Punkten der Zielfunktion enthielten. In diesem Fall konnte die Suche nach der Rekonstruktion der Hauptkomponenten abgebrochen werden. Die rekursive Netzkonstruktion war also nicht mehr notwendig. Die Berechnungszeit betrug 65 Sekunden!

Die Einbindung von zusätzlichen Startpunkten ist allerdings mit zusätzlichem Aufwand verbunden: der Grad der Nichtlinearität der Hilfsfunktion H steigt. Dies äußert sich im allgemeinen in höherer Anzahl AK der Trajektorienkomponenten (vgl. Tabellen 4.2 und 4.3) oder längerer Bearbeitungszeit (vgl. Tabelle 4.3).

4.3 Gegenüberstellung vorgeschlagener Gitterpunkt-Strategien

Theoretische Vorteile und Nachteile verschiedener Gitterpunktstrategien wurden bereits im Abschnitt 2.2 ausführlich diskutiert. In diesem Abschnitt werden die numerischen Ergebnisse der Strategie der Berührungspunkte (B-Strategie) und der Strategie des äquidistanten Netzes (A-Strategie) gegenübergestellt. Die Versuchsreihen mit den 1- bis 3-Punkt-Trajektorien für die zweidimensionalen Testfunktionen f_e und f_g werden hierfür kurz zusammengefaßt.
Durch geeignete Wahl der Dichte δ des äquidistanten Netzes wurde hierbei der Versuch unternommen, den Aufwand des Verfahrens für die beiden Strategien vergleichbar zu halten. Nichtsdestotrotz ist dieser von der geometrischen Beschaffung der Trajektorie und damit von der Wahl der vorgegebenen Startpunkte abhängig und aus diesem Grunde nicht gleich zu halten.

Als Kennzahlen werden die Erfolgsquote EQ und der Rekursionsaufwand RA wie folgt definiert:

EQ : Anzahl der Testdurchläufe (von insgesamt hundert) in denen alle kritische Punkte gefunden wurden.

		A-Strategie		B-Strategie	
Testfunktion	Trajektorie	EQ	RA	EQ	RA
f_c	$1-P$	86	5.3 ± 2.0	74	4.6 ± 2.9
f_c	$2-P$	74	6.0 ± 1.4	59	5.5 ± 2.9
f_c	$3-P$	78	6.0 ± 1.5	68	6.7 ± 3.1
f_c	$M3-P$	80	6.0 ± 1.3	76	7.4 ± 2.9
f_e	$1-P$	84	5.5 ± 2.2	70	4.2 ± 3.7
f_e	$2-P$	89	6.1 ± 1.5	71	4.9 ± 2.9
f_e	$3-P$	87	6.0 ± 1.5	74	5.8 ± 3.5
f_e	$M3-P$	97	6.1 ± 1.2	85	6.4 ± 2.6

Tabelle 4.4: Vergleich der Gitterpunkt-Strategien

RA : Durchschnittliche Anzahl der rekonstruierten Hilfstrajektorien $\pm$ Standardabweichung

Die Kennzahlen EQ und RA der beiden Strategien können aus Tabelle 4.4 abgelesen werden.

Die A-Strategie hat für die untersuchten Probleme bessere Ergebnisse geliefert (vgl. Tabellen 4.4 und 4.8). Die Trefferquote war im Vergleich zu der B-Strategie in allen Fällen besser und der Aufwand in den meisten Fällen nur geringfügig größer. Die B-Strategie führte zwar im allgemeinen zu weniger Hilfstrajektorien, in einigen Einzelfällen war der Rekursionaufwand sehr groß. Dies resultierte in der größeren Standardabweichung der Größe RA.

Einen großen Einfluß auf die Erfolgsquote des Verfahrens kann auch die Richtung der Hilfstrajektorien haben. Im Fall der in [15] vorgeschlagenen rekursiven Konstruktion für die klassischen Newton-Trajektorie (vgl. Abschnitt 2.1.2) stimmt diese mit der Richtung des Gradienten in dem vorgegebenen Ausgangspunkt überein und kann nur im Fall der 1-Punkt-Trajektorien angewendet werden. Im Fall der für die Richtungsfeld-Trajektorien vorgeschlagenen allgemeineren Konstruktion (vgl. Abschnitt 2.1.3) ist die Richtung durch die Wahl des Orthogonalssystems Q induziert und damit frei festzulegen. Die Erfolgsquote für die 1-Punkt-Trajektorien mit der waagerecht, senkrecht und klassisch gewählten Hilfstrajektorien wurden in der Tabelle 4.5 dargestellt.

Die beste Wahl der Richtung der Hilfstrajektorien könnte auch während der Berechnung der ersten Komponenten, abhängig von deren Lage ermittelt werden.

		waagerecht	senkrecht	klassisch
Testfunktion	Gitterpunkt-Strategie	EQ	EQ	EQ
f_c	A-Strategie	94	78	98
	B-Strategie	86	62	78
f_e	A-Strategie	94	74	100
	B-Strategie	84	56	80

Tabelle 4.5: Vergleich der Gitterpunkt-Strategien

	Zielfunktionswerte	**Gradienten**	**Hesse-Matrix**
KNV	berechnet	berechnet	berechnet
KQNV	berechnet	berechnet	approximiert
SV	berechnet	approximiert	approximiert

Tabelle 4.6: Notwendige Berechnungen der vorgestellten Verfahren

4.4 Komplexitätsvergleich vorgestellter Verfahren

In diesem Kapitel werden drei Verfahren vorgestellt, mit deren Hilfe ein in den Kapiteln 1 und 2 definiertes Trajektoriennetz rekonstruiert werden kann. Die Verfolgung der Trajektorienkurve basiert hierbei auf dem Prädiktor-Korrektor-Prinzip (s. 3.1).
Alle drei Verfahren sind auf gleiche Weise rekursiv aufgebaut (vgl. Abschnitt 2.4) und führen deshalb zu gleichen Trajektorienkomponenten.
Da die Verfahren nur in Ausnahmefällen abgebrochen werden müssen (vgl. Abschnitte 3.2.2, 3.3.3, 3.4.2), sind die erzielten Ergebnisse meistens identisch. Unabhängig von dem Verfahren werden also immer die gleichen kritischen Punkte gefunden.
Der Unterschied zwischen den Verfahren liegt in der Art und Weise, wie die Information über die Zielfunktion gewonnen und eingesetzt wird. Eine Übersicht wurde in der Tabelle 4.6 zusammengestellt.

In der Diplomarbeit [3] wurden die klassischen Verfahren, KNV und KQNV, im Einsatz für 1-Punkt-Trajektorien ausführlich getestet. Das KNV hat sich in den meisten Fällen als vorteilhaft erwiesen. Im Vergleich mit Surrogate-Verfahren sind die Vorteile vom KNV noch größer. Obwohl der Berechnungsaufwand pro Schritt für die beiden Verfahren vergleichbar ist, ist die Anzahl der für die korrekte Rekonstruktion der Trajektorie notwendigen Schritte für KNV kleiner und damit auch die Bearbeitungszeit viel kürzer. Hierzu wurde die Rekonstruktion der $1-$Komponenten Trajektorien der Testfunktionen f_c, f_e und f_t verglichen. Die Anzahl der Rekonstruk-

	KNV			AFSV		
Testfunktion	AS	LZ [s]	LZ/AS	AS	LZ [s]	LZ/AS
f_c	85	1.5	0.018	510	10.6	0.021
f_e	420	11.8	0.028	1631	35.3	0.022
f_t	570	10.8	0.019	1464	31.6	0.022

Tabelle 4.7: Gegenüberstellung von KNV und AFSV

tionsschritte (AS), die Bearbeitungszeit (LZ) und der Berechnungsaufwand pro Schritt (LZ/AS) wurden in der Tabelle 4.7 dargestellt.

Die Effizienz des Verfahrens ist von vielen Faktoren abhängig. Die Struktur des Problems, aber auch die gewählten Parameter der Schrittlängensteuerung spielen hierbei eine wichtige Rolle.
Es ist nicht möglich, pauschal zu entscheiden, welches Verfahren für ein bestimmtes Problem am besten geeignet ist. Eine Faustregel ist, möglichst alle Informationen, die zur Verfügung stehen, zu nutzen. Die Hesse-Matrix und/oder der Gradient der Zielfunktion sollten nach Möglichkeit genau ausgewertet werden. Eine Approximation dieser Werte ist im Zweifelsfall nur dann zu empfehlen, wenn die genaue Auswertung nicht möglich oder sehr aufwendig ist.

4.5 Testprobleme mit Restriktionen

4.5.1 Zweidimensionale Testfunktionen

1. Für das folgende Optimierungsproblem (vgl. Problem 3.3 in [21]):

$$f_1(z) = -25(x-2)^2 - (y-2)^2 \rightarrow \min!$$

$$\begin{aligned} 2 - x + 3y &\geq 0 \\ 2 + x - y &\geq 0 \\ 6 - x - y &\geq 0 \\ -2 + x + y &\geq 0, \end{aligned}$$

kann eine richtungsfeldinduzierende Funktion $c_1(z)$ wie folgt definiert werden (vgl. Bemerkung 62):

$$c_1(z) = (2 - x + 3y)(2 + x - y)(6 - x - y)(-2 + x + y).$$

Der Zulässigkeitsbereich des Problems wurde in der Abbildung 4.7 dargestellt. Für das restringierte Problem gibt es außer den kritischen

Punkt:

$$z_1^* = (2,2)^T,$$

noch vier weitere Randpunkte $z_i^*, \quad i = 2, ..., 5$:

$$\begin{aligned} z_2^* &= \left(2\frac{3}{113}, \frac{1}{113}\right)^T \\ z_3^* &= \left(1\frac{12}{13}, 3\frac{12}{13}\right)^T \\ z_4^* &= \left(2\frac{1}{13}, 3\frac{12}{13}\right)^T \\ z_5^* &= \left(1\frac{12}{13}, \frac{1}{13}\right)^T \end{aligned}$$

sowie vier Ecken $z_i^*, \quad i = 6, ..., 9$:

$$\begin{aligned} z_6^* &= (0,2)^T \\ z_7^* &= (2,0)^T \\ z_8^* &= (2,4)^T \\ z_9^* &= (5,1)^T \end{aligned}$$

des Restriktionsbereiches, die als Kandidaten für das globale Minimum bzw. das globale Maximum in Betracht kommen (s. Abbildung 4.8). Das globale Minimum und Maximum für das aufgestellte restringierte Problem sind gegeben mit:
Globales Minimum:

$$z_9^*, \text{ mit dem Funktionswert } f_1(z_9^*) = -226.$$

Globales Maximum:

$$z_1^* \text{ mit dem Funktionswert } f_1(z_1^*) = 0.$$

2. Für das folgende Optimierungsproblem (vgl. Problem 3.3 in [21]):

$$f_2(z) = -(x-4)^2 - (y-1)^2 \rightarrow \min!$$

$$\begin{aligned} 6 - x &\geq 0 \\ y - 1 &\geq 0 \\ 5 - y &\geq 0 \\ -4 + x + (y-3)^2 &\geq 0, \end{aligned}$$

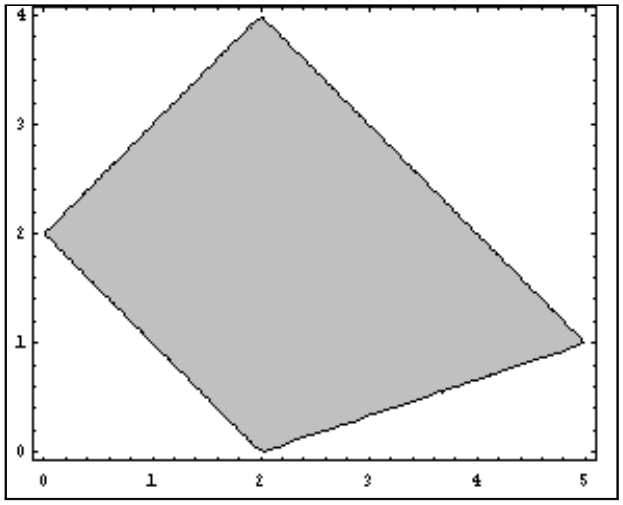

Abbildung 4.7: Restriktionsmenge für das zweidimensionale Testproblem 1

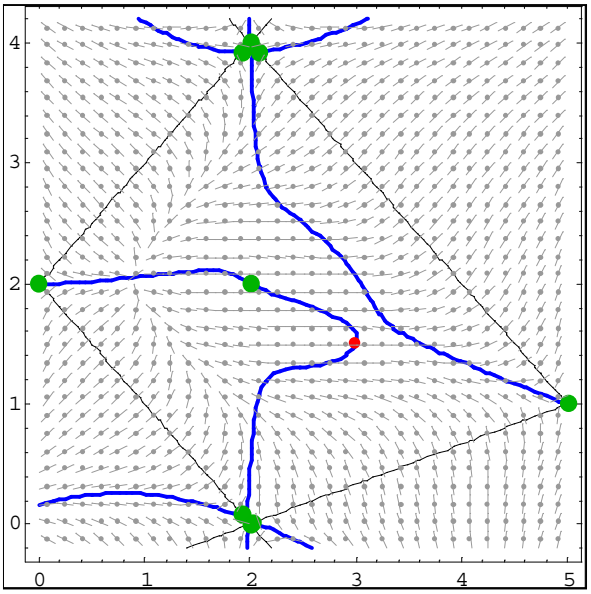

Abbildung 4.8: Beispieltrajektorie für die Testfunktion f_1;
Startpunkt: $\{3.0, 1.5\}$

kann eine richtungsfeldinduzierende Funktion $c_2(z)$ wie folgt definiert werden (vgl. Bemerkung 62):

$$c_2(z) = (6-x)(y-1)(5-y)\left(-4+x+(y-3)^2\right).$$

Der Zulässigkeitsbereich des Problems wurde in der Abbildung 4.9 dargestellt. Für das restringierte Problem gibt es außer den kritischen Punkt:

$$z_1^* = (4,1)^T,$$

noch zwei weitere Randpunkte $z_i^*, \quad i = 2,3:$

$$\begin{aligned} z_2^* &= (4,5)^T \\ z_3^* &= (3.30257, 2.16488)^T \end{aligned}$$

sowie vier Ecken $z_i^*, \quad i = 4,...,7:$

$$\begin{aligned} z_4^* &= (0,1)^T \\ z_5^* &= (0,5)^T \\ z_6^* &= (6,1)^T \\ z_7^* &= (6,5)^T \end{aligned}$$

des Restriktionsbereiches, die als Kandidaten für das globale Minimum bzw. das globale Maximum in Betracht kommen (s. Abbildung 4.10). Das globale Minimum und Maximum für das aufgestellte restringierte Problem sind gegeben mit:
Globales Minimum:

$$z_5^*, \text{ mit dem Funktionswert } f_2(z_5^*) = -32.$$

Globales Maximum:

$$z_1^* \text{ mit dem Funktionswert } f_2(z_1^*) = 0.$$

3. Für das folgende Optimierungsproblem (vgl. Problem 4.6 in [21]):

$$\begin{aligned} f_3(z) &= -x-y \rightarrow \min! \\ 0 &\leq x \leq 3 \\ 0 &\leq y \\ y &\leq 2x^4 - 8x^3 + 8x^2 + 2 \\ y &\leq 4x^4 - 32x^3 + 88x^2 - 96x + 36, \end{aligned}$$

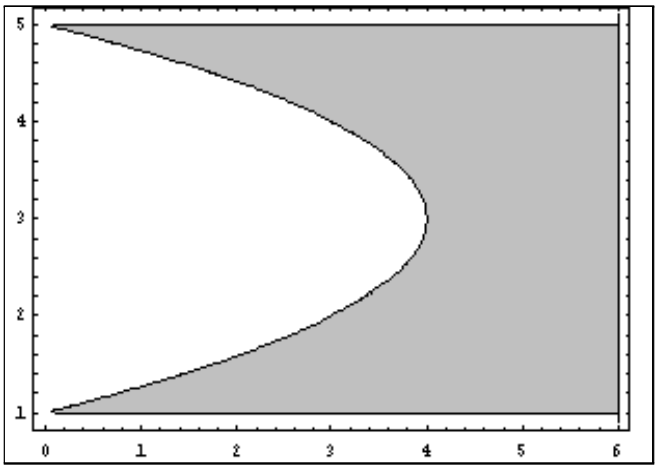

Abbildung 4.9: Restriktionsmenge für das zweidimensionale Testproblem 2

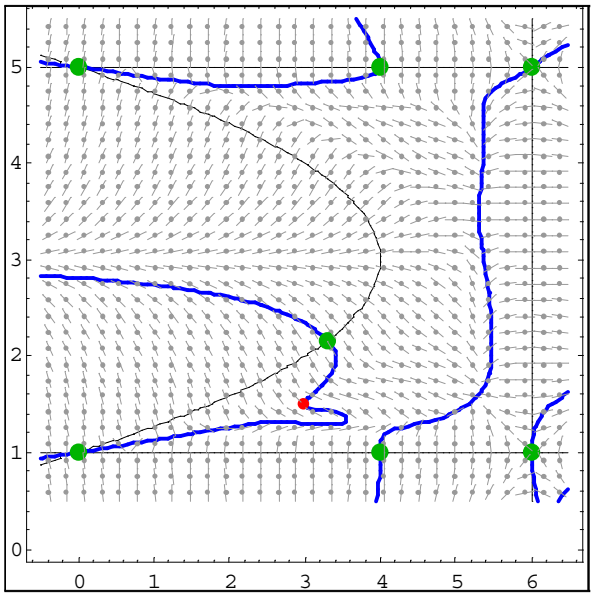

Abbildung 4.10: Beispieltrajektorie für die Testfunktion f_2;
Startpunkt: $\{3.0, 1.5\}$

kann eine richtungsfeldinduzierende Funktion $c_3(z)$ wie folgt definiert werden (vgl. Bemerkung 62):

$$\begin{aligned} c_3(z) &= x(3-x)\,y\left(2x^4-8x^3+8x^2+2-y\right) \\ &\quad \cdot\left(4x^4-32x^3+88x^2-96x+36-y\right). \end{aligned}$$

Der Zulässigkeitsbereich des Problems wurde in der Abbildung 4.11 dargestellt. Für das restringierte Problem gibt es keine kritischen Punkte. Als Kandidaten für das globale Minimum bzw. das globale Maximum kommen drei Randpunkte $z_i^*, \quad i = 1, ..., 3$:

$$\begin{aligned} z_1^* &= (0.97010, 0.01473)^T \\ z_2^* &= (1.93040, 2.03610)^T \\ z_3^* &= (2.96715, 0.01670)^T \end{aligned}$$

sowie sieben Ecken $z_i^*, \quad i = 4, ..., 10$:

$$\begin{aligned} z_4^* &= (0,0)^T \\ z_5^* &= (0,2)^T \\ z_6^* &= (0.61160, 3.44210)^T \\ z_7^* &= (1,0)^T \\ z_8^* &= (1.59962, 2.82036)^T \\ z_9^* &= (2.32950, 3.17830)^T \\ z_{10}^* &= (3,0)^T \end{aligned}$$

des Restriktionsbereiches in Betracht (s. Abbildung 4.12). Das globale Minimum und Maximum für das aufgestellte restringierte Problem sind gegeben mit:
Globales Minimum:

$$z_9^*, \text{ mit dem Funktionswert } f_3(z_9^*) = -5.5079.$$

Globales Maximum:

$$z_4^* \text{ mit dem Funktionswert } f_3(z_4^*) = 0.$$

4.5.2 Mehrdimensionale Testfunktionen

1. Die zweidimensionalen Testprobleme 1 und 2 werden hier zusammengefaßt (vgl. Problem 3.3 in [21]):

$$f_4(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) + f_2(x_3, x_4) \rightarrow \min!$$

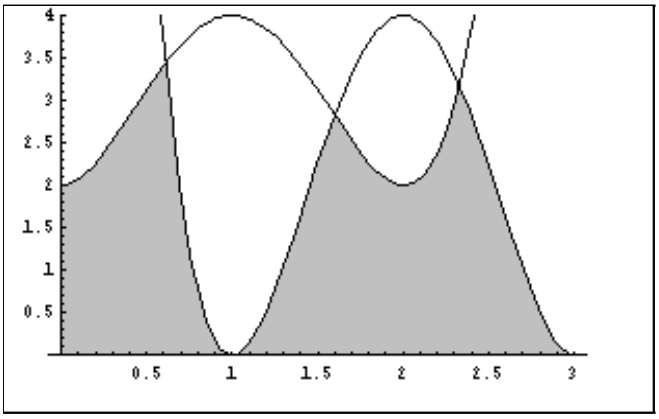

Abbildung 4.11: Restriktionsmenge für das zweidimensionale Testproblem 3

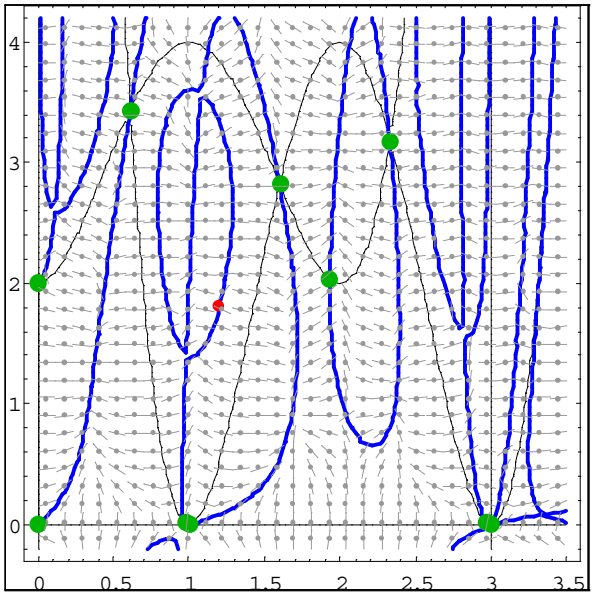

Abbildung 4.12: Beispieltrajektorie für die Testfunktion f_3;
Startpunkt: $\{1.2, 1.8\}$

$$\begin{aligned} 2 - x_1 + 3x_2 &\geq 0 \\ 2 + x_1 - x_2 &\geq 0 \\ 6 - x_1 - x_2 &\geq 0 \\ -2 + x_1 + l2 &\geq 0 \\ 6 - x_3 &\geq 0 \\ x_4 - 1 &\geq 0 \\ 5 - x_4 &\geq 0 \\ -4 + x_3 + (x_4 - 3)^2 &\geq 0 \end{aligned}$$

Eine richtungsfeldinduzierende Funktion $c_4(z)$ kann wie folgt definiert werden (vgl. Bemerkung 62):

$$c_4(x_1, x_2, x_3, x_4, x_5, x_6) = c_1(x_{1,}x_2)\, c_2(x_3, x_4).$$

Es gibt hier $9 \cdot 7 = 63$ Kandidatenpunkte. Das globale Minimum und Maximum für das aufgestellte restringierte Problem sind gegeben mit: Globales Minimum:

$$z^*_{9,5} = (5, 1, 0, 5)^T, \text{ mit dem Funktionswert } f_4\left(z^*_{9,5}\right) = -258.$$

Globales Maximum:

$$z^*_{1,1} = (2, 2, 4, 1)^T \text{ mit dem Funktionswert } f_4\left(z^*_{1,1}\right) = 0.$$

4.6 Trajektorienansatz in restringierten Optimierung

Wie im Abschnitt 1.4.4 bereits besprochen wurde, ermöglicht der Ansatz der Richtungsfeld-Trajektorie einen Zugang zur globalen Optimierung für restringierte Probleme. Hat der Zulässigkeitsbereich eine einfache Form (eine n-dimensionale Kugel, Würfel, etc.), so kann das Richtungsfeld mit wenig Aufwand bestimmt und eingesetzt werden. Auch für komplizierte oder auch zusammengesetzte zulässige Mengen können Trajektorien definiert und berechnet werden, solange ein geeignetes Richtungsfeld konstruiert werden kann (vgl. Testprobleme 2 und 3; Abbildungen 4.9 und 4.11).

Das Richtungsfeld und damit auch die durch das Richtungsfeld induzierte Trajektorie sind in der Regel auch außerhalb des zulässigen Bereiches definiert. Die Rekonstruktion der Trajektorie muß nicht und sollte auch nicht auf den zulässigen Bereich beschränkt werden. Es könnten sonst die Komponenten der Trajektorien nicht entdeckt werden, die die relevanten Randpunkte

	A-Strategie	B-Strategie
f_1 (7 krit. Punkte)	6.0 ± 1.1	5.6 ± 1.3
f_2 (9 krit. Punkte)	9.0 ± 0.0	8.5 ± 1.1
f_3 (10 krit. Punkte)	10.0 ± 0.0	10.0 ± 0.0

Tabelle 4.8: Trefferquote für die restringierten Probleme

enthalten aber hauptsächlich oder sogar vollständig außerhalb des zulässigen Bereiches verlaufen (vgl. Beispieltrajektorie für die Testfunktion f_2; Abbildung 4.10).

In diesem Zusammenhang sind also auch die Startpunkte interessant, die nicht der zulässigen Menge angehören (vgl. Beispieltrajektorie für die Testfunktion f_3; Abbildung 4.12). Lediglich die Randpunkte des zulässigen Bereiches dürfen nicht als Startpunkte für die Richtungsfeld-Trajektorie genommen werden (vgl. Bemerkung 61).

Die Trefferquote für die Rekonstruktion der im Abschnitt 1.4.4 definierten Trajektorie ist für die beschriebenen zweidimensionalen Beispielprobleme 1 bis 3 sehr zufriedenstellend (s. Tabelle 4.8).

Für die Probleme mit vielen Restriktionen (vgl. Testfunktion f_4) ist die Berechnung des Richtungsfeldes mit größerem Aufwand verbunden. Das Verfahren hat zwar auch in dem Fall der Testfunktion f_4 viele der kritischen Punkte entdeckt. Es wurden aber auch zusätzliche Punkte entdeckt, die als singuläre Punkte des Richtungsfeldes identifiziert wurden und für die Fragestellung nicht relevant waren. Für die Probleme mit vielen bzw. hoch nichtlinearen Restriktionen ist der Ansatz der Richtungsfeld-Trajektorie also wenig geeignet. Der Lagrange-Ansatz (vgl. Abschnitt 1.1) könnte vielleicht in einem solchen Fall Abhilfe schaffen.

Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde als Verallgemeinerung der klassischen Newton-Trajektorie der Begriff der durch ein Richtungsfeld induzierten Trajektorie eingeführt. Diese Verallgemeinerung der klassischen Newton-Trajektorie eröffnete neue Möglichkeiten, Einfluß auf den Verlauf der Trajektorie zu nehmen.

In diesem Zusammenhang wurden Trajektorien mit mehreren Startpunkten definiert (s. Abschnitt 1.4.3). Die aufgeführten Testrechnungen belegen, daß der Ansatz auch zur Verbesserung der Trefferquote des Trajektorienverfahrens führt. Die freie Wahl der Startpunkte ist insbesondere dann von Vorteil, wenn zusätzliche Informationen über die Verteilung der kritischen Punkte vorliegen (s. Abschnitt 4.2).

Für restringierte Optimierungsprobleme sind die Randpunkte der zulässigen Menge, die die notwendige Optimalitätsbedingung erster Ordnung erfüllen, von besonderem Interesse. Die konstruierten Richtungsfeld-Trajektorien enthalten außer den kritischen Punkten der Zielfunktion auch die genannten Randpunkte (s. Abschnitt 1.4.4). Der Ansatz der Richtungsfeld-Trajektorien ist aber nur für die restringierten Probleme zu empfehlen, für die ein einfaches Richtungsfeld mit keinen (oder wenigen) Singularitäten konstruiert werden kann.(s. Abschnitt 4.6).

Die rekursive Konstruktion des Trajektoriennetzes wurde für die eingeführten Richtungsfeld-Trajektorien angepaßt und mußte von der Richtung des Gradienten (die jetzt nicht mehr konstant sein muß) entkoppelt werden (s. Abschnitt 2.1.3). Die vorgegebenen Richtungen des konstruierten Gitters haben großen Einfluß auf die Erfolgsquote des Verfahrens. Die beste Richtung kann auch im Laufe der Berechnung ermittelt werden (s. Abschnitt 4.3).

Eine andere Möglichkeit, die Erfolgsquote des Verfahrens zu verbessern, liegt in der Strategie der Bestimmung der Gitterpunkte (Startpunkte für die Verbindungstrajektorien). Verglichen wurden die Strategien der Berührungspunkte und des äquidistanten Netzes. Die äquidistante Strategie hat sich bei den von uns behandelten Problemen als vorteilhaft herausgestellt (s. Ab-

schnitt 4.3). Stehen allerdings zusätzliche Informationen über die Verteilung der kritischen Punkte zur Verfügung, so könnten nicht äquidistante Netze bzw. hybride Strategien Vorteile bieten.

Ein besonderes Augenmerk wurde auf die Implementierung der beschriebenen Verfahren gelegt.
Zur rekursiven Konstruktion des Trajektoriennetzes wurden BFS (breadth first search) und DFS (depth first search) als objektorientierte Algorithmen beschrieben (s. Abschnitt 2.4).
Zur Rekonstruktion der Trajektorienkomponenten wurde das kontinuierliche Newton-Verfahren (vgl. Abschnitt 3.2), das kontinuierliche Quasi-Newton-Verfahren (vgl. Abschnitt 3.3) und das ableitungsfreie Surrogate-Verfahren (vgl. Abschnitt 3.4) vorgestellt. Die Implementierung der beschriebenen Methoden wird durch zahlreiche Beispiele erleichtert.

Die eingeführten Ideen und vorgeschlagenen Strategien ermöglichen einen vielseitigen Einsatz zur Lösung praktischer Probleme. Die lange Berechnungszeit könnte durch die effiziente Implementierung in einer leistungsstarken objektorientierten Sprache (z.B. C++) deutlich verkürzt werden. Die gezielte Wahl der Startpunkte würde die Trefferquote des Verfahrens zusätzlich verbessern.

Literaturverzeichnis

[1] E. L. Allgower and K. Georg. *Numerical Continuation Methods An Introduction*. Springer, 1990.

[2] R. Ansorge and H.J. Oberle. *Mathematik Für Ingenieure*. Akademie Verlag GmbH, 1994.

[3] L. Bajorski. Über Das Kontinuierliche Quasi-Newton-Verfahren. Master's thesis, Universität Hamburg, 1993.

[4] G. Bannert and M. Weitzel. *Objektorientierter Softwareentwurf mit UML*. Addison Wesley Longman Verlag GmbH, 1999.

[5] A. Booker, J.E. Jun. Dennis, P. D. Frank, D. B. Serafini, and V. Torczon. Optimization using surrogate objectives on a helicopter test example. *Prog. Syst. Control Theory.*, 24:49–58, 1998.

[6] F.H. Branin. A widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16:504–522, 1972.

[7] C. G. Broyden. A class of methods for solving nonlinear simultaneous equation. *Math. Comp.*, 19:577–593, 1965.

[8] A.R. Conn, K. Scheinberg, and Ph.L. Toint. On the convergence of derivative-free methods for unconstrained optimization. Invited presentation at the Powellfest, Cambridge, 1996.

[9] A.R. Conn, K. Scheinberg, and Ph.L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Math. Prog.*, 79:397–414, 1997.

[10] A.R. Conn, K. Scheinberg, and Ph.L. Toint. A derivative free optimization in practice. 1998.

[11] A.R. Conn and Ph.L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. To appear in Nonlinear Optimization and Applications, 1995.

[12] C. de Boor and A. Ron. On multivariate polynomial interpolation. *Constructive Approximation*, 6:287–302, 1990.

[13] C. de Boor and A. Ron. Computational aspects of polynomial interpolation in several variables. *Math. Comput.*, 58:705–727, 1992.

[14] J.E. Jun. Dennis and J.J. More. Quasi-newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[15] I. Diener. Trajectory nets connecting all critical points of a smooth function. *Math. Prog.*, 36:577–593, 1986.

[16] I. Diener. On the global convergence of path-following methods to determine all solutions to a system of nonlinear equations. *Math. Program.*, 39:181–188, 1987.

[17] I. Diener. Newton leaves and the continuous newton method. In Juergen (Ed.) et Al. Guddat, editor, *Parametric Optimization and Related Topics. III.*, pages 121–134. Frankfurt am Main: Peter Lang Verlag., 1993.

[18] I. Diener. Trajectory methods in global optimization. In Reiner et Al. Horst, editor, *Handbook of Global Optimization.*, pages 649–668. Dordrecht: Kluwer Academic Publishers., 1995.

[19] I. Diener and R. Schaback. An extended continuous newton method. *Journal of Optimization Theory and Applications*, 67:57–77, 1990.

[20] R. Fletcher. *Practical Methods of Optimization (Second Ed.).* John Wiley and Sons, Chichester, 1987.

[21] C. A. Floudas and P. A. Pardalos. *A Collection of Test Problems for Constrainde Global Optimization Algorithms.* Springer-Verlag, 1987.

[22] M. Fowler and K. Scott. *UML-Konzentriert, Die Standardobjektmodellierungssprache Anwenden.* Addison Wesley Longman Verlag GmbH, 1998.

[23] C. Geiger. *Optimierung.* Vorlesungsskript, Universitaet Hamburg, 1990.

[24] K. Georg. Numerical integration of the davidenko equation. In Springer Lecture Notes in Math., editor, *Numerical Solution of Non Linear Equations*, pages 117–127. Allgower E. and Glashoff K. and Peitgen H., 1981.

[25] K. Georg. *Zur Numerischen Realisieung Von Kontinuitaetsmethoden mit Praediktor-Korrektor-, Oder Simplizialen Verfahren.* PhD thesis, Universitaet Bonn, 1982.

[26] Ph.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, London, 1981.

[27] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[28] D. Jungnickel. *Graphen, Netzwerke und Algorithmen.* Wissenschaftsverlag Mannheim/Wien/Zürich, 1990.

[29] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[30] I.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, 1970.

[31] T. Sauer. Computational aspects of multivariate polynomial interpolation. *Adv. Comput. Math.*, 3, No.3:219–237, 1995.

[32] T. Sauer and Y. Xu. On multivariate lagrange interpolation. *Math. Comput.*, 64, No.211:1147–1170, 1995.

[33] C.M. Yang and J.L. Beck. Generalized trajectory methods for finding multiple extrema and roots of functions. *Journal of Optimization Theory and Applications*, 97:211–227, 1998.

Index

Berührungspunkt, 70
BFS Algorithmus, 88

DFS Algorithmus, 90
durch eine Matrix induzierte Trajektorie, 15
durch einen Vektor induzierte Trajektorie, 16
durch Richtungsfeld induzierte Trajektorie, 26

Gitterpunkt, 58

induzierender Vektor, 16
induzierter Tangentenvektor, 7

klassische Newton-Trajektorie, 17
kritischer Punkt, 2

Newton-Blatt, 56

regulärer Punkt, 5
regulärer Wert, 5
Richtungsfeld-Trajektorie, 25

singulärer Punkt, 5
singulärer Wert, 5
stationärer Punkt, 2

trajektorieninduzierende Hilfsfunktion, 4

verallgemeinerte Newton-Kurve, 6
verallgemeinerte Newton-Trajektorie, 8
Verbindungstrajektorien, 62

zulässige Abstiegsrichtung, 3

Lebenslauf

Persönliche Daten

Name	Bajorski
Vorname	Leszek
Geburtsdatum	3. November 1966
Geburtsort	Jaroslaw (Polen)
Familienstand	verheiratet, zwei Kinder

Schulausbildung

Sept. ´81 - Juni ´85	Allgemeinbildende Oberschule, Breslau
Juni ´85	Abschluß: Abitur (Polen)

Studium

Okt. ´85 - Juni ´89	Fach: Mathematik, Universität Breslau / Polen
Okt. ´86 - Juni ´89	Paralleles Studium Fach: Informatik, Universität Breslau
Okt. ´90 - Mai ´93	Fach: Angewandte Mathematik, Universität Hamburg
Mai ´93	Diplom im Studiengang Mathematik
Jan. ´96 - März 2000	Promotionsstudium am Fachbereich Mathematik der Universität Hamburg

Berufliche Tätigkeiten

Sept. ´93 - Dez. ´95	Wissenschaftlicher Mitarbeiter der Forschungsgruppe "Chirurgische Forschung" an der Medizinischen Universität zu Lübeck
April ´97 - März 2000	Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Universität Hamburg