



---

# Implementierung des Apollonischen Berührproblems

---

## Bachelorarbeit

an der Fakultät für Informatik und Mathematik  
der Universität Passau

**Prüfer:**

Prof. Dr. Tomas Sauer

Günther Reitberger  
Wintersemester 2013/14

## Zusammenfassung

Ein Kreis, der drei gegebene Kreise berührt, ist eine Lösung des Apollonischen Berührproblems. Die Kreise dürfen auch entartet sein, das heißt ein Punkt oder eine Gerade. Unterscheidet man also einen allgemeinen Kreis in einen nichtentarteten Kreis, einen Punkt und eine Gerade, so ergeben sich verschiedene Varianten des Apollonischen Berührproblems.

In dieser Arbeit wird für jede Konstellation des Berührproblems ein Algorithmus zur Lösung vorgestellt und eine detaillierte Betrachtung von Spezialfällen geliefert. Die Korrektheit der Ansätze wird ebenfalls bewiesen. Anschließend wird erläutert, warum die Ergebnisse der Implementierung numerischen Fehlern unterworfen sind. Diese zu umgehen oder zumindest einzuschränken ist ein besonderes Anliegen. Da trotzdem ungenaue Lösungen nicht zu vermeiden sind, wird die nichtlineare Optimierung vorgestellt, mit deren Hilfe eine Verringerung des Fehlers angestrebt wird. An Hand eines Beispiels wird zum einen die Wirkungsweise dieses Verfahrens verdeutlicht, zum anderen werden die Grenzen aufgezeigt.

Den Abschluss bildet ein kleines Résumé und es wird auf weitere Möglichkeiten hingewiesen, die Ergebnisse der Implementierung exakter zu gestalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Historischer Hintergrund . . . . .	1
1.2	Problemstellung und Arbeitsumfang . . . . .	2
<b>2</b>	<b>Vorstellung der Lösungsstrategien</b>	<b>4</b>
2.1	Auswahl möglicher Ansätze . . . . .	4
2.1.1	Elementargeometrischer Ansatz . . . . .	4
2.1.2	Inversion am Kreis . . . . .	4
2.1.3	Allgemeine Kreisgleichung . . . . .	5
2.1.4	Kegelschnitte . . . . .	5
2.1.5	Weitere Ideen . . . . .	6
2.2	Vorstellung der gewählten Algorithmen . . . . .	7
2.2.1	PPP . . . . .	7
2.2.2	CPP . . . . .	9
2.2.3	PPL . . . . .	11
2.2.4	PLL . . . . .	12
2.2.5	CLL . . . . .	14
2.2.6	CCL . . . . .	17
2.2.7	CPL . . . . .	18
2.2.8	CCP . . . . .	19
2.2.9	LLL . . . . .	21
2.2.10	CCC . . . . .	22
2.3	Korrektheit des Kegelschnittansatzes . . . . .	22
<b>3</b>	<b>Überlegungen zur numerischen Stabilität</b>	<b>23</b>
3.1	Gleichheitstest . . . . .	23
3.2	Kollinearitätstest . . . . .	24
3.2.1	Determinantenverfahren . . . . .	24
3.2.2	Normierungsverfahren . . . . .	25
3.2.3	Rotationsverfahren . . . . .	26
3.3	Auswahl der Kegelschnitte . . . . .	28
3.4	Berechnung des Radius der Umkreise . . . . .	28
3.5	Transformation in Normallagen . . . . .	29
3.6	Reihenfolge der Ausführung von Operationen . . . . .	30
3.7	Lösen einer quadratischen Gleichung . . . . .	30
<b>4</b>	<b>Nichtlineare Optimierung mit dem Newton-Verfahren</b>	<b>32</b>
4.1	Problemstellung . . . . .	32
4.2	Das Newton-Verfahren . . . . .	33
4.3	Anwendung des Newton-Verfahrens . . . . .	34
4.4	Gründe für die Verwendung des Newton-Verfahrens . . . . .	36

<i>Inhaltsverzeichnis</i>	iii
4.5 Gegenüberstellung der Ergebnisse mit und ohne Optimierung an Hand eines Beispiels . . . . .	36
4.6 Schwachstellen des Newton-Verfahrens . . . . .	40
<b>5 Fazit</b>	<b>42</b>
<b>Literaturverzeichnis</b>	<b>44</b>
<b>Abbildungsverzeichnis</b>	<b>45</b>

# 1 Einleitung

## 1.1 Historischer Hintergrund

Schon in der Antike beschäftigte sich mit Apollonius von Perge (ca. 260 - 170 v.Chr.) einer der größten Mathematiker seiner Zeit mit dem einfach anmutendem Problem, einen Kreis zu zeichnen, der drei gegebene Kreise berührt. Die Kreise dürfen dabei auch entartet sein, d.h. ein Punkt<sup>1</sup> oder eine Gerade<sup>2</sup>. Wie es sich für eine interessante Ideengeschichte zu einer historischen Aufgabe gehört, gingen die von Apollonius in seinem Werk *De Tactionibus* notierten Lösungen des Problems verloren. Das war aber nicht das Ende für die Kreise. Um 1600 veröffentlichte der französische Mathematiker Vieta (1540 - 1603) eine Arbeit, in der er die 10 Spezialfälle<sup>3</sup> einzeln behandelte und seine elementargeometrischen Ansätze aufeinander aufbauen lies. Obwohl damit wieder Lösungen für das Problem vorhanden waren, widmeten sich mit Josef Diaz Gergonne (1771 - 1859), René Descartes (1596 - 1650), Leonard Euler (1707 - 1783) und Carl Friedrich Gauß (1777 - 1855) weitere Größen der Mathematik dem Finden von Berührungskreisen. Die letzten drei verfolgten analytische Methoden, während Gergonne sich die Eigenschaften von Ähnlichkeits- und Potenzpunkten zu eigen machte. Auf Grund dieses großen Interesses der Mathematikerhistorie an dem Berührproblem, ist es als „Apollonisches Berührproblem“ nach seinem

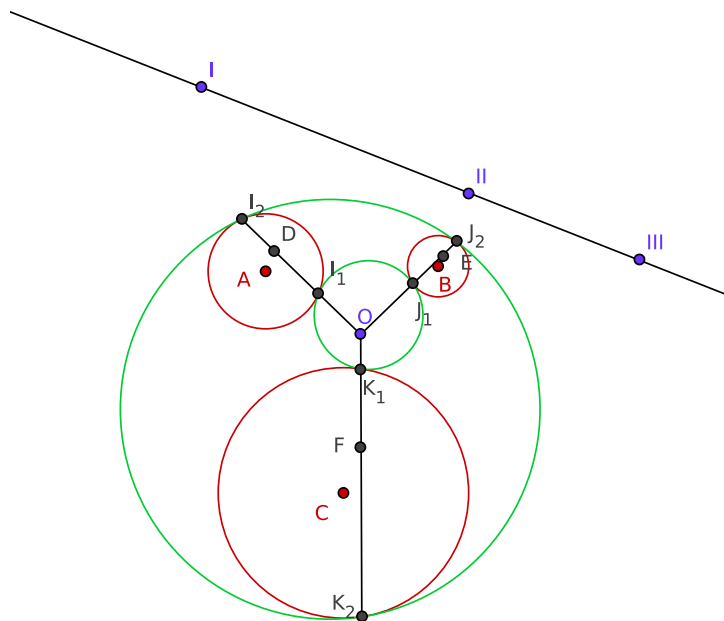


Abb. 1: Berührungskreise nach dem Verfahren von Gergonne

<sup>1</sup>ein Kreis mit Radius 0

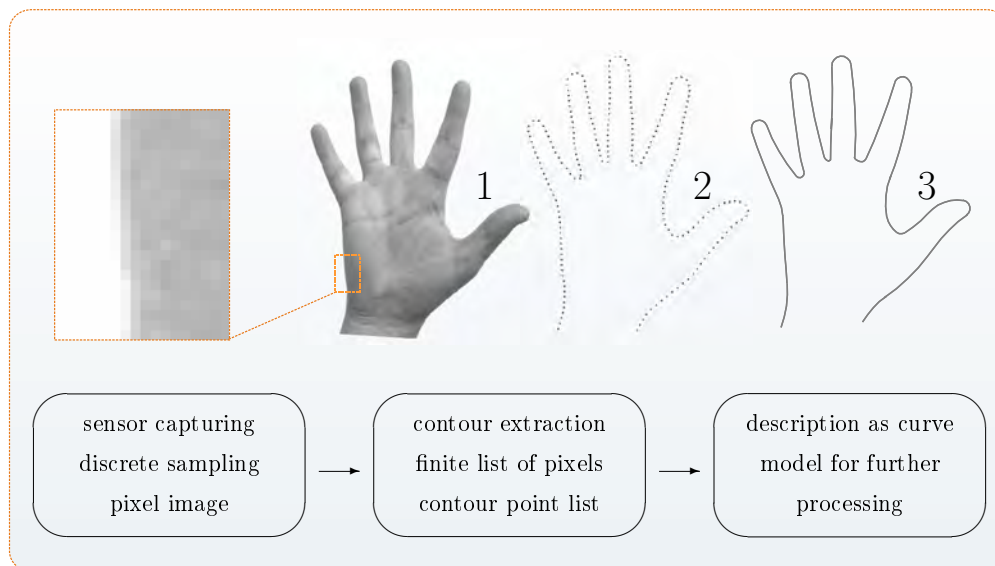
<sup>2</sup>ein Kreis mit Radius  $\infty$

<sup>3</sup>Für die Spezialfälle wurde ein allgemeiner Kreis entweder als Punkt, Gerade oder nichtentarteter Kreis betrachtet. Daraus ergeben sich verschiedene Fälle, die nachfolgend genauer aufgeschlüsselt werden.

Entdecker heute noch bekannt und ging unter diesem Namen in Sammlungen wie „100 Great Problems of Elementary Mathematics“ von Heinrich Dörrie [2] ein. Darin wird die Lösung von Gergonne kurz beschrieben, die mit Hilfe eines Potenzentrums  $O$  und einer Ähnlichkeitsachse, die durch die Ähnlichkeitspunkte  $I$ ,  $II$  und  $III$  bestimmt wird, Lösungskreise findet. Zu sehen ist das in Abb. 1. Darin sind die gegebenen Kreise mit Mittelpunkten  $A$ ,  $B$  und  $C$  rot eingezeichnet und die Berührkreise grün. Es sind nicht alle möglichen Berührkreise aufgeführt. Es gibt noch weitere Ähnlichkeitsachsen, die weitere Lösungen erzeugen.

## 1.2 Problemstellung und Arbeitsumfang

Das Apollonische Berührproblem ist keinesfalls auf seine historische Entwicklung und theoretische Bedeutung beschränkt. Auch in aktuellen Forschungen spielt diese Aufgabenstellung eine Rolle: So findet sie etwa in der Dissertation von Georg Maier zum Thema „Smooth Minimum Arc Paths“ oder kurz SMAP [6] Anwendung. Unter Einhaltung einer vom Benutzer einstellbaren Genauigkeit approximiert der SMAP-Algorithmus Punktsequenzen mit Kreisbogensplines<sup>4</sup>, die beweisbar kleinstmögliche Anzahl an Kurvensegmenten haben. Durch einen so genannten Toleranzkanal wird dabei die gewählte Genauigkeit kontrolliert. Die Aufgabenstellung wird in Abb. 2 veranschaulicht. Entscheidend für die algorithmische Umsetzung dieses Verfahrens ist die Berechnung von Kreisbögen, die den Toleranzkanal mehrmals und gegebenenfalls einen weiteren Kreisbogen berühren. Für polygonale Toleranzkanäle, wie sie in der Praxis meist auftreten, lässt sich diese Aufgabenstellung damit auf die Berechnung eines Kreises zurückführen, der



**Abb. 2:** Übersicht über die Motivation des DFG-Projekts

<sup>4</sup>Kreisbogensplines sind  $G^1$  stetige Kurven bestehend aus Kreisbögen und Strecken.

drei - teils entartete - Kreise berührt, also auf das Apollonische Berührproblem. Diese antike Fragestellung findet sich somit noch heute im Computer-Zeitalter wieder. So ist auch diese Arbeit im Kontext des Forschungsprojekts „Methoden für den Einsatz von Kreisbogensplines zur Darstellung hochgenauer digitaler Karten“<sup>5</sup>, das auf der genannten Dissertation [6] basiert, entstanden.

Die Aufgabe besteht jedoch nicht etwa darin, der Lösung von Gauß Konkurrenz zu machen und einen eleganteren Ansatz zu finden, sondern einen bereits entwickelten Algorithmus auszuwählen und zu implementieren. Damit muss man sich aber auch mit praktischen Problemen auseinandersetzen, die zum Beispiel ein Descartes ignoriert hat, indem er seine Berechnungen nicht mehr zu Ende führte, nachdem er gezeigt hatte, dass seine gefundenen Gleichungen prinzipiell lösbar sind. Dazu gehört auch die Tatsache, dass ein Computer nicht vollkommen exakt rechnet und daher die Ergebnisse fehlerbehaftet sind. Deshalb ist neben der korrekten Implementierung auch die numerische Stabilität ein großes Anliegen dieser Arbeit.

---

<sup>5</sup>Das Projekt wird von der Deutschen Forschungsgemeinschaft (DFG) gefördert (Fördernummer MA 5834/1-1, <http://gepris.dfg.de/gepris/projekt/225923277>)

## 2 Vorstellung der Lösungsstrategien

### 2.1 Auswahl möglicher Ansätze

Der Idee von Vieta folgend wurde der allgemeine Kreis in Punkt(P), Gerade(L) und nicht entartetem Kreis(C) unterschieden. Im Folgenden wird als Kreis nur die nicht entartete Version verstanden. Daraus ergeben sich 10 Spezialfälle, die separat behandelt werden. PPL steht dabei zum Beispiel für den Fall, dass der Berührungskreis zu zwei gegebenen Punkten und einer Geraden berechnet werden muss.

Generell ist zu sagen, dass es für die jeweiligen Konstellationen verschiedene Anzahlen von Lösungen gibt. Für den Fall CCC gilt, dass jeder Kreis sowohl von innen als auch von außen berührt werden kann, sodass sich maximal 8 nichtentartete Lösungen ergeben. Für die Fälle, in denen es weniger als 8 Lösungen gibt, sind die fehlenden Lösungen in den komplexen Bereich gewandert. Für den PPP-Fall etwa gibt es nur einen Berührungskreis, außer die drei Punkte sind kollinear. Dann entartet der Lösungskreis zu einer Geraden. Folgende Tabelle bildet eine Übersicht der 10 Fälle des Apollonischen Berührproblems unter der Angabe der Lösungen, die es im Allgemeinen für diesen Fall gibt.

Fall	PPP	CPP	PPL	PLL	CCP	CPL	LLL	CLL	CCL	CCC
Anz. d. Lsg.	1	2	2	2	4	4	4	8	8	8

Darüber hinaus kann man sich auch mit verschiedenen Ansätzen diesem Problem des Findens der Berührungskreise nähern. Nachstehend wird eine kleine Auswahl von möglichen Ansätzen vorgestellt, wobei aber nicht ins Detail gegangen wird. Röttgen-Burtscheidt geht in [8] ausführlicher darauf ein. Die jeweils ausgewählte Methode wird nachfolgend an den einzelnen Fällen genauer beschrieben.

#### 2.1.1 Elementargeometrischer Ansatz

Bei diesem Verfahren nähert man sich dem Problem mit Zirkel und Geodreieck. Zurückgegriffen wird auf geometrisches Grundwissen, wie der Konstruktion von Mittelsenkrechten, Loten oder Winkelhalbierenden. Hinzu kommen etwas fortgeschrittene Sätze, wie der Umfangswinkelsatz, der Sehnen-Satz und ähnliche. Vieta hat gezeigt, dass diese Mittel ausreichen, um das Apollonische Berührproblem zu lösen.

#### 2.1.2 Inversion am Kreis

Da der elementargeometrische Ansatz für die Fälle CCC, CCL, CCP, CPP oder PCL ziemlich aufwändig wird, kann man sich des Mittels der Inversion am Kreis bedienen, um die genannten Fälle auf den geometrisch leicht zu lösenden Fall CLL zurückzuführen, wobei die beiden Geraden parallel zueinander sind. Die Inversion am Kreis  $k$  mit Mittelpunkt  $M$  und Radius  $r$  ist eine bijektive Abbildung auf der Ebene ohne  $M$ . Dabei werden das Innere und das Äußere von  $k$  miteinander vertauscht, indem der Punkt  $P \neq M$  auf den Punkt  $P'$  mit folgenden Eigenschaften abgebildet wird:



1.  $P'$  liegt auf der Trägergeraden  $MP$
2.  $\overline{MP} * \overline{MP'} = r^2$

### 2.1.3 Allgemeine Kreisgleichung

Im Gegensatz zu den vorherigen konstruktiven Methoden geht man nun analytisch an das Problem heran und beschreibt einen Lösungskreis allgemein durch den Mittelpunkt  $M$  mit den Koordinaten  $x$  und  $y$  und den Radius  $r$ .

Für den CCC-Fall mit den Kreisen  $k_i$  mit Mittelpunkten  $M_i(x_i, y_i)$ , Radien  $r_i$  und  $i \in \{1, 2, 3\}$  muss eine Lösung  $(x, y, r)$  folgende Gleichungen erfüllen:

$$\begin{aligned}(x - x_1)^2 + (y - y_1)^2 &= (r \pm r_1)^2 \\(x - x_2)^2 + (y - y_2)^2 &= (r \pm r_2)^2 \\(x - x_3)^2 + (y - y_3)^2 &= (r \pm r_3)^2\end{aligned}$$

Das heißt, der Abstand von  $M$  zu  $M_i$  muss entweder gleich der Summe oder der Differenz der Radien sein, je nachdem ob der Lösungskreis  $k_i$  von außen oder einschließend berührt. Die Gleichungen sind in quadrierter Form angegeben, um die Berechnung der Wurzel zu vermeiden. Nun kann man dieses Gleichungssystem mit der Bedingung  $x^2 + y^2 = r^2$  nach den Unbekannten  $x$ ,  $y$  und  $r$  auflösen. Dadurch bekommt man auch die maximal 8 regulären Lösungen. Ebenso bilden diese drei Gleichungen einen Ansatz für alle Fälle, in denen ausschließlich Kreise und Punkte vorkommen, da nur die entsprechenden Radien für die Punkte auf 0 gesetzt werden müssen.

Für die Fälle, an denen auch Geraden beteiligt sind, müssen Kegelschnitte als Ortslinien von Punkten, die den gleichen Abstand zu zwei gegebenen Objekten haben, herangezogen werden. Jedoch beschränkt man sich hier im Gegensatz zum folgenden Kegelschnittansatz auf die Mittelsenkrechte, als eine solche Ortslinie für zwei Punkte und auf die Winkelhalbierende für zwei Geraden. Man kommt also ohne kompliziertere Kegelschnitte aus. Die Komplexität bleibt aber dadurch erhalten, dass man den Abstand der Punkte auf den Ortslinien zu den anderen Objekten berechnen muss, was im reinen Kegelschnittansatz nicht nötig ist, da dort nur die Ortslinien geschnitten werden.

### 2.1.4 Kegelschnitte

Der Ansatz mit den Kegelschnitten ist dem der allgemeinen Kreisgleichung sehr ähnlich, nur dass nun jeweils zwei Objekte betrachtet werden und die Ortslinie der Punkte mit gleichem Abstand zu den beiden Objekten aufgestellt wird. Schneidet man zwei solche Ortslinien, erhält man die Mittelpunkte der Berührungskreise. Die Ortslinien lassen sich als Kegelschnitte beschreiben.

Was der Ansatz mit der Kreisgleichung mit dem der Kegelschnitte noch gemeinsam hat, ist, dass bei beiden die zu lösenden Gleichungen ziemlich aufwändig werden können. Will man dies umgehen, so ist es ratsam, die Probleme zunächst zu transformieren, sodass einfachere Gleichungen entstehen. Ein Beispiel dafür wäre im PPL-Fall die Drehung der Objekte, bis die Gerade mit der x-Achse zusammenfällt. Am Ende muss die berechnete

Lösung wieder zurückgedreht werden, damit sie auch eine Lösung für die Ausgangslage darstellt.

### 2.1.5 Weitere Ideen

Wie schon im historischen Rückblick erwähnt, widmeten sich unter anderem die Mathematiker Gergonne, Descartes, Euler und Gauß dem Apollonischen Berührproblem und trugen Vorschläge zur Lösung bei. Die Konstruktion von Gergonne mit den Ähnlichkeits- und Potenzpunkten wurde in der Einleitung bereits vorgestellt und kann bei Dörrie in [2] nachgelesen werden. Die analytischen Ansätze von Descartes, Euler und Gauß werden für den CCC-Fall kurz bei Röttgen-Burtscheidt in [8] beschrieben. Alle drei beschreiben die gesuchte Lösung durch drei Unbekannte, die sie mit Hilfe eines Gleichungssystems bestimmen wollen, nutzen dabei jedoch völlig verschiedene Eigenschaften, die eine solche Lösung erfüllen muss. Dabei wird zumeist nur der Radius oder der Mittelpunkt berechnet. Hält man nämlich eine der beiden Komponenten, so kann man die andere Komponente leicht bestimmen.

Descartes liefert sogar zwei Lösungen. Die Erste nimmt die Heronsche Formel für den Flächeninhalt eines Dreiecks zur Hilfe. Mit den Dreiecksflächen, die von dem Lösungskreismittelpunkt und je zwei der Mittelpunkte der gegebenen Kreise aufgespannt werden, berechnet er den Radius des Berührkreises. Die zweite Lösung hat den Satz des Pythagoras als Grundlage.

Euler hingegen wählte einen trigonometrischen Zugang. Er verwendet in seiner Arbeit „Solutio facilis problematis, quo quaeritur circulus, qui datos tres circulos tangat“ den Cosinussatz und die Additionstheoreme. Euler geht in einem nachfolgendem Werk sogar einen Schritt weiter und löst das Apollonoische Berührproblem in der dreidimensionalen Variante.

Gauß gibt ebenfalls einen trigonometrisch motivierten Algorithmus an, der jedoch sehr abstrakt ist. Deshalb wird hier nur auf die Referenz verwiesen.

## 2.2 Vorstellung der gewählten Algorithmen

Die oben aufgezeigten Ansätze spielten eine Rolle bei den Überlegungen zur tatsächlichen Implementierung der jeweiligen Fälle des Apollonischen Berührproblems. Diese werden im Folgenden in Auszügen auf abstraktem Niveau dargestellt. Wenn von Abständen oder einem Skalarprodukt die Rede ist, dann sind der Euklidische Abstand bzw. das Standardskalarprodukt gemeint. Des Weiteren findet sich häufig eine C++ ähnliche Notation wieder. Wenn etwa ein Punkt  $P$  gegeben ist, so bezeichnen  $P.x$  oder  $P.y$  seine  $x$ - und  $y$ -Koordinaten. Es wird stets davon ausgegangen, dass die drei gegebenen Objekte verschieden sind, also nicht zum Beispiel zwei Punkte zusammenfallen. Nun noch zwei Definitionen von Begriffen, die in der Ausarbeitung verwendet werden.

**Definition 2.1** (Reguläre Lösung). *Als reguläre Lösung wird ein Lösungskreis bezeichnet, der nicht zu einem Punkt oder einer Geraden entartet ist. Es gibt Konstellationen, für die es keinen oder unendlich viele Berührkreise gibt. Diese werden kurz erwähnt, zu den regulären Lösungen zählen diese jedoch nicht. Das heißt, dass es maximal endlich viele reguläre Lösungen für die einzelnen Fälle geben kann.*

**Definition 2.2** (Normallage). *Wie schon bei der Vorstellung des Kegelschnittansatzes angedeutet wurde, werden die einzelnen Fälle stets so transformiert, dass sich die Berechnungen vereinfachen. Die Konstellation in derartig transformierter Lage wird Normallage oder normierte Lage genannt.*

Es wurde stets die Lösung mit den Kegelschnitten oder der allgemeinen Kreisgleichung gewählt, da sie sich softwaretechnisch gut umsetzen ließ. Dabei galt es, noch relativ übersichtliche quadratische Gleichungen aufzustellen und zu lösen. Bei den Kegelschnitten und der Kreisgleichung hat man die Formeln in der Hand und kann meist sofort an ihnen ablesen, was verbessert oder worauf geachtet werden muss. Andere Ansätze wie etwa von Gauß oder Descartes wurden nicht weiter verfolgt, da zum Beispiel der erste Ansatz von Descartes zu einer ziemlich aufwändigen Gleichung führt, die er nach einem Zitat selbst nicht auflösen wollte.

Ebenso waren für das Projekt, im Zuge dessen die Implementierung dieser Arbeit erstellt wurde, die Fälle LLL und CCC nicht notwendig, weshalb sie auch nicht implementiert wurden. Deren Lösung ist jedoch mit dem Wissen, das man auf Grund der behandelten Fälle erwerben wird, sehr einfach. Deswegen wird kurz darauf eingegangen, aber der Algorithmus nicht explizit ausgeführt.

Es wird stets angegeben, wie viele reguläre Lösungen es gibt. Auf den Beweis, dass die Anzahl richtig ist, wird verzichtet und stattdessen auf die geometrische Vorstellung verwiesen.

### 2.2.1 PPP

Für diesen Fall ist das Problem nichts anderes, als den Umkreis um das von den drei Punkten aufgespannte Dreieck zu finden und wurde elementargeometrisch schon zu Schulzeiten mit dem Schnittpunkt der Mittelsenkrechten gelöst. Der Ansatz mit den Kegelschnitten

führt zu demselben Algorithmus, da die Ortslinie der Punkte mit gleichem Abstand zu zwei gegebenen Punkten ebenfalls die Mittelsenkrechte der Verbindungslinie ist.

Es gibt stets genau eine nichtentartete Lösung, außer die Punkte sind kollinear. Dann entartet der Lösungskreis zu der Geraden, auf der die Punkte liegen.

Nach [5] Kapitel 3.5.3 berechnet sich der Umkreismittelpunkt  $U$  eines Dreiecks mit

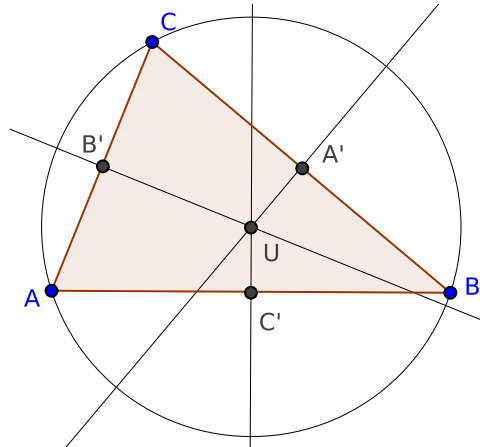


Abb. 3: Umkreis eines Dreiecks

den Eckpunkten  $A$ ,  $B$  und  $C$ , indem man zwei Mittelsenkrechten  $m_1$  und  $m_2$  wie folgt schneidet:

$$m_1 : C' + \lambda n_c, \quad \lambda \in \mathbb{R} \quad (1)$$

$$m_2 : B' + \mu n_b, \quad \mu \in \mathbb{R} \quad (2)$$

$$m_1 = m_2 \Leftrightarrow C' + \lambda n_c = B' + \mu n_b \Leftrightarrow \overrightarrow{B'C'} = \mu n_b - \lambda n_c$$

Dabei ist  $C'$  der Mittelpunkt der Seite  $[AB]$ ,  $B'$  der Mittelpunkt der Seite  $[AC]$  und  $n_b$  bzw.  $n_c$  sind die Normalenvektoren der Geraden  $AC$  bzw.  $AB$ . Illustriert wird dies in Abb. 3.

Daraus folgt mit den Eigenschaften des Skalarprodukts:

$$\langle \overrightarrow{AB}, \overrightarrow{B'C'} \rangle = \langle \overrightarrow{B'C'}, \overrightarrow{AB} \rangle = \langle \mu n_b, \overrightarrow{AB} \rangle - \langle \lambda n_c, \overrightarrow{AB} \rangle = \mu \langle n_b, \overrightarrow{AB} \rangle$$

und

$$\mu = \frac{\langle \overrightarrow{AB}, \overrightarrow{B'C'} \rangle}{\langle n_b, \overrightarrow{AB} \rangle}$$

Verwendet man  $\overrightarrow{B'C'} = C' - B' = \frac{1}{2}(A + B) - \frac{1}{2}(A + C) = \frac{1}{2}\overrightarrow{CB} = -\frac{1}{2}\overrightarrow{BC}$  und setzt dies mit  $\mu$  in (2) ein, erhält man

$$U = \frac{1}{2}(A + C - \frac{\langle \overrightarrow{AB}, \overrightarrow{BC} \rangle}{\langle n_b, \overrightarrow{AB} \rangle} * n_b)$$

Wie man dieser Formel leicht entnehmen kann, existiert der Mittelpunkt nur für den Fall, dass  $\langle n_b, \overrightarrow{AB} \rangle$  ungleich 0 ist, d.h. die Punkte  $A$ ,  $B$  und  $C$  nicht kollinear sind.

Der Radius des Lösungskreises lässt sich nun als der Abstand zu einem der drei Eckpunkte bestimmen.

### 2.2.2 CPP

Dieser Abschnitt beschreibt die Situation, dass die Berührkreise zu einem Kreis  $k$  mit Mittelpunkt  $M$  und Radius  $r$  und zwei Punkten  $P1$  und  $P2$  berechnet werden müssen. Dies ist sogleich um einiges komplizierter und aufwändiger als der PPP-Fall. Keine Lösung gibt es, wenn ein Punkt innerhalb von  $k$  liegt und der andere außerhalb. Ebenso trivial wird es, wenn beide Punkte auf  $k$  liegen, da dann der Kreis  $k$  selbst die Lösung bildet.

Die Berührkreismittelpunkte werden als Schnittpunkte der Ortslinien der Punkte, die gleichen Abstand zu einem Punkt und dem Kreis bzw. zu beiden Punkten haben, berechnet. Für die beiden Punkte weiß man bereits, dass die Mittelsenkrechte der Verbindungsstrecke dies erfüllt. Wie bei Röttgen-Burtscheidt in [8] Kapitel 5 beschrieben wird, ist die Ortslinie der Punkte, die von  $k$  und o.B.d.A.  $P1$  gleichen Abstand haben, eine Ellipse, wenn  $P1$  im Inneren von  $k$  liegt und eine Hyperbel, wenn  $P1$  außerhalb von  $k$  liegt. Befindet sich  $P1$  auf dem Kreis, so liegt der Mittelpunkt des Berührkreises auf der Geraden, die durch  $P1$  und  $M$  bestimmt ist.

Da aber die Formeln einer Ellipse bzw. einer Hyperbel in allgemeiner Lage relativ kompliziert sind, verschafft man sich eine Vereinfachung dadurch, dass man den Kreis und

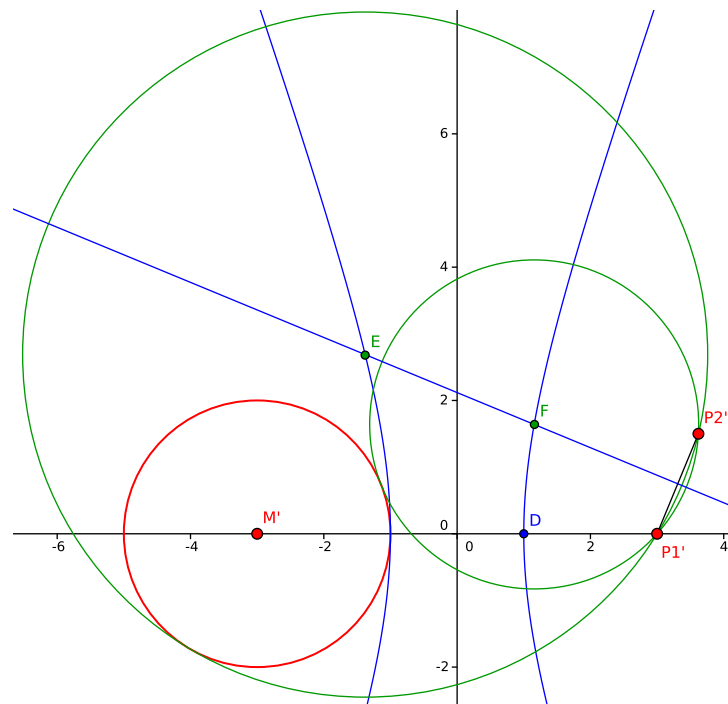


Abb. 4: CPP-Fall in Normallage mit Hyperbel

die beiden Punkte so dreht und verschiebt, dass der Mittelpunkt  $M$  und einer der beiden Punkte symmetrisch um den Ursprung auf der x-Achse liegen. Die Punkte und der Kreis in transformierter Lage sollen  $P1'$ ,  $P2'$  und  $k'$  mit Mittelpunkt  $M'$  heißen. Der Radius  $r$  bleibt unverändert. Damit sind die Hyperbel und die Ellipse in 1. Hauptlage und jeder Punkt  $P(x, y)$  darauf muss folgende Gleichung erfüllen:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \text{ für die Hyperbel, } \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \text{ für die Ellipse}$$

Dabei ist  $a := r/2$  und  $b := \sqrt{e^2 - a^2}$  für die Hyperbel und für die Ellipse  $b := \sqrt{a^2 - e^2}$ , wobei  $e$  der Abstand des Kreismittelpunktes  $M'$  zum Ursprung ist. Die Hyperbel und die Ellipse sind nur wohldefiniert, wenn  $a$  und  $b$  nicht 0 sind. Für  $a$  ist das trivialerweise erfüllt, da  $k$  ansonsten ein Punkt wäre, was ausgeschlossen ist. Liegt der Punkt, der die Hyperbel bzw. Ellipse mit aufspannt auf  $k$ , so gilt  $b = 0$ . Dies muss in einer Fallunterscheidung abgefangen werden und die Ortslinie ergibt sich dann als die Gerade durch  $M'$  und dem Punkt. In der transformierten Lage entspricht das der x-Achse.

Die Mittelsenkrechte zwischen  $P1'(x_1, y_1)$  und  $P2'(x_2, y_2)$  lässt sich in der Form  $y = m * x + t$  bestimmen mit

$$m = -\frac{x_2 - x_1}{y_2 - y_1} \text{ und } t = \frac{(x_2^2 - x_1^2) + (y_2^2 - y_1^2)}{2(y_2 - y_1)}.$$

Dies kann man sich mit elementargeometrischem Wissen überlegen oder auch bei Röttgen-Burtscheidt unter dem Kapitel „Analytische Methoden“ nachlesen. Nun hat man zwei Gleichungen mit zwei Unbekannten, die zusammen zu einer quadratischen Gleichung führen, die es zu lösen gilt:

$$(b^2 - a^2m^2)x^2 - 2a^2mtx - a^2(t^2 + b^2) = 0 \quad \text{für die Hyperbel}$$

$$(b^2 + a^2m^2)x^2 + 2a^2mtx + a^2(t^2 - b^2) = 0 \quad \text{für die Ellipse}$$

Ein Beispiel für den CPP-Fall in Normallage ist in Abb. 4 gegeben. Dabei sind die beiden roten Punkte und der rote Kreis gegeben. Die Mittelsenkrechte und die Hyperbel sind blau eingezeichnet, die beiden Lösungskreise grün.

Im Standardfall gibt es dabei 2 Lösungen. Liegt einer der beiden Punkte auf dem Kreis, gibt es nur eine Lösung und das ist der Schnittpunkt der Geraden durch den Mittelpunkt und den berührenden Punkt mit der Mittelsenkrechten. Bei der Berechnung der Mittelsenkrechten muss man noch darauf achten, dass die Steigung  $m$  für den Fall, dass  $P1'$  und  $P2'$  auf der x-Achse liegen, unendlich werden kann. Das bedeutet einfach, dass die x-Koordinate der Lösungen bestimmt ist durch die x-Koordinate des Mittelpunkts von  $[P1', P2']$ . Die y-Koordinate kann dann durch Einsetzen in die Formel für die Ellipse bzw. Hyperbel bestimmt werden.

Außerdem gilt es beim Auflösen der quadratischen Gleichung Acht zu geben, da dabei durch den Vorfaktor des quadratischen Terms geteilt wird. Das heißt, man muss prüfen, ob  $b^2 - a^2m^2 = 0$  bzw.  $b^2 + a^2m^2 = 0$  gilt. Dann fällt der quadratische Anteil weg und es ist nur noch eine lineare Gleichung übrig, die maximal eine Lösung liefert. Für die Ellipse ist ersichtlich, dass dies nicht der Fall sein kann, da ansonsten  $b^2$  Null ergeben müsste und das zuvor abgefangen wurde. Für die Hyperbel trifft die Bedingung genau dann zu, wenn die Steigung der Mittelsenkrechten  $m = \pm \frac{b}{a}$  erfüllt. Nun verwundert es sicherlich

auf den ersten Blick, warum für exakt diese Steigung eine Lösung wegfällt. Man kann sich zunächst einmal analytisch überlegen, dass die Mittelsenkrechte im Unendlichen parallel zu einem Hyperbelast verlaufen muss. Die Funktion  $f(x) = \pm \frac{b}{a} \sqrt{x^2 - a^2}$  beschreibt über einen geeigneten Definitionsbereich einen solchen Hyperbelast. Leitet man  $f$  nach  $x$  ab und berechnet darüber den Limes gegen Unendlich, so erhält man auch  $\pm \frac{b}{a}$  als Grenzwert. Geometrisch bedeutet  $m = \pm \frac{b}{a}$ , dass die Gerade durch  $P1'$  und  $P2'$  den Kreis  $k'$  berührt. Eben diese Gerade wäre als Kreis mit unendlichem Radius eine Lösung, fällt aber unter den regulären Lösungen weg.

Hat man nun die Mittelpunkte der Berührungskreise in transformierter Lage gefunden, müssen diese abschließend so zurücktransformiert werden, sodass sie auch eine Lösung in der Ausgangslage bilden. Die Radien können wieder durch den Abstand zu etwa  $P1$  berechnet werden.

### 2.2.3 PPL

Nachdem im obigen Abschnitt im Vergleich zum PPP-Fall ein Punkt durch einen Kreis ersetzt wurde, so wird dieser nun durch eine Gerade ersetzt. Keine Lösung gibt es, wenn sich die beiden Punkte in verschiedenen, von der Geraden aufgespannten Halbebenen befinden. Ebenso sind die Konstellationen uninteressant, in denen beide Punkte auf der Geraden liegen.

Die Lösungen werden wieder als Schnittpunkte von Kegelschnitten berechnet. Seien nun

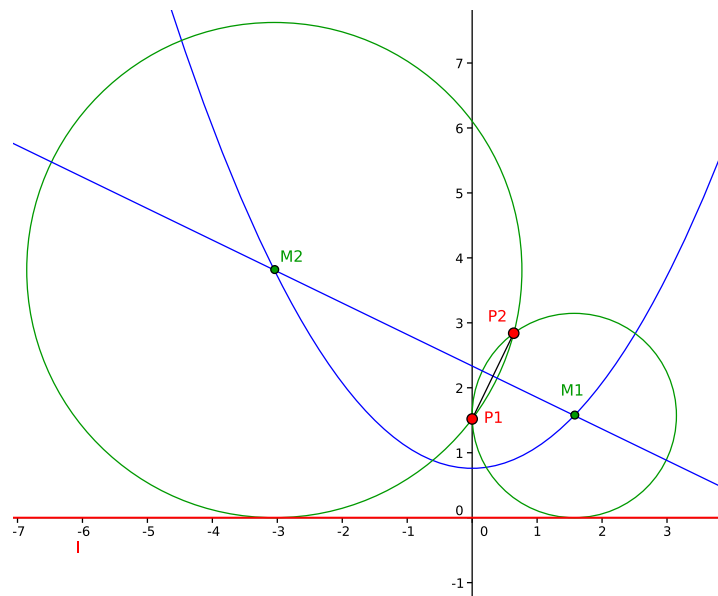


Abb. 5: PPL-Fall in Normallage

die Punkte  $P1$ ,  $P2$  und die Gerade  $l$  gegeben. Für  $P1$  und  $P2$  wurde bereits die Mittelsenkrechte als Ortslinie der Punkte ausgemacht, die gleichen Abstand zu  $P1$  und  $P2$  haben. In [8] von Röttgen-Burtscheidt kann man nun nachlesen, dass es eine Parabel

ist, die diese Eigenschaft für einen Punkt und eine Gerade erfüllt. Auch hier gilt, dass es angenehmer ist, mit einer Parabel der Form  $y = a * x^2 + b$  umzugehen, als in der allgemeineren Variante  $\frac{2p(x - x_s)}{(y - y_s)^2} = 1$  mit dem Scheitelpunkt  $S(x_s, y_s)$  und  $p$  als Abstand des Brennpunktes zur Leitlinie. Deshalb wird zunächst die Situation so transformiert, dass die Gerade mit der x-Achse zusammenfällt und o.B.d.A. der Punkt  $P1$  auf der y-Achse liegt. Die transformierten Punkte sollen  $P1'$  und  $P2'$  heißen. Die Parabel, die als Ortslinie für  $P1'$  und die x-Achse dient, lässt sich nun durch  $y = \frac{1}{2 * P1'.y} x^2 + \frac{P1'.y}{2}$  darstellen. Zusammen mit der Gleichung  $y = m * x + t$  für die Mittelsenkrechte hat man erneut zwei Gleichungen mit zwei Unbekannten, die man einsetzen und auflösen kann. Die resultierende quadratische Gleichung lautet nun:

$$\frac{1}{2 * P1'.y} x^2 - mx + \frac{P1'.y}{2} - t = 0$$

Damit erhält man die maximal zwei Mittelpunkte der Lösungskreise. Natürlich müssen diese wieder zurücktransformiert werden und der Abstand zu einem Punkt bestimmt den Radius.

Unterschlagen darf man jedoch nicht, dass die Parabel zu einer Geraden entarten kann. Das ist dann der Fall, wenn  $P1'$  auf der x-Achse liegt. Somit muss der Schnittpunkt der Mittelsenkrechten mit dem Lot zur x-Achse berechnet werden. Ebenso kann die Mittelsenkrechte zu einem Lot zur x-Achse entarten, wenn die Verbindungslinie durch  $P1'$  und  $P2'$  parallel zur x-Achse ist. Dann muss die Parabel mit diesem Lot geschnitten werden. In beiden Spezialfällen gibt es nur eine Lösung.

In Abb. 5 ist eine Illustration des PPL-Falls in normierter Lage zu sehen, d.h. die Gerade  $l$  fällt mit der x-Achse zusammen und der Punkt  $P1$  liegt auf der y-Achse. Die gegebenen Objekte  $P1$ ,  $P2$  und  $l$  sind rot eingezeichnet, die Mittelsenkrechte von  $[P1, P2]$  und die Parabel, die durch  $P1$  und  $l$  bestimmt wird, sind blau markiert. Die Lösungskreise mit ihren Mittelpunkten sind grün eingetragen.

#### 2.2.4 PLL

Der PLL-Fall stellt also die Situation dar, dass zwei Geraden und ein Punkt  $P$  gegeben sind. Zentral für diese Aufgabenstellung ist die Frage, ob sich die beiden Geraden schneiden oder nicht. Folgt man nämlich dem Ansatz der Kegelschnitte, so ist die Ortslinie der Punkte, die gleichen Abstand von beiden Geraden haben, in der parallelen Situation die Mittellinie, wohingegen die Winkelhalbierenden diese Eigenschaft erfüllen, wenn sich die Geraden schneiden.

Betrachten wir zunächst die parallele Situation. Transformiert man die Ausgangslage wie in Abb. 6 so, dass eine Gerade auf der x-Achse liegt, so ist die Aufgabe leicht zu lösen. Die beiden gegebenen Geraden und der Punkt  $P$  sind rot eingezeichnet. Liegt der Punkt nicht zwischen den beiden Geraden und auch nicht auf einer der beiden Geraden, gibt es keine Lösung. Liegt  $P$  hingegen auf einer der beiden Geraden, gibt es eine Lösung, nämlich den Punkt auf der blau gestrichelt eingezeichneten Mittellinie, der dieselbe x-Koordinate hat, wie  $P$ . Liegt nun  $P$  zwischen den beiden Geraden, wie in der



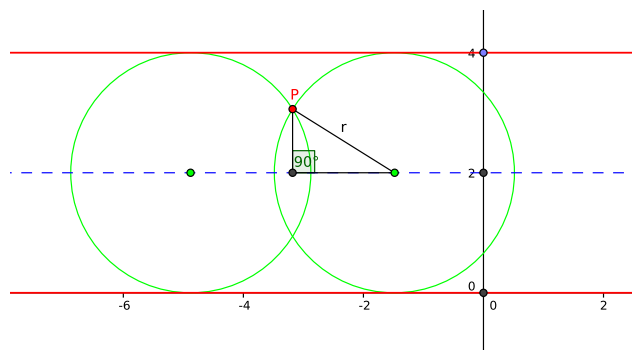


Abb. 6: PLL-Fall für parallele Geraden in normierter Lage

Abbildung, kann man die gesuchten Umkreismittelpunkte mit dem Satz von Pythagoras bestimmen, da der Abstand von  $P$  zur Mittellinie, als auch der Radius  $r$  als betragsmäßiger  $y$ -Achsenabschnitt der Mittellinie bekannt sind. Dies ist auch in der Abbildung angedeutet. Nun zu dem Fall, dass sich die Geraden schneiden.

Für einen Punkt und eine Gerade ist, wie schon im PPL-Fall gesehen, die Parabel die

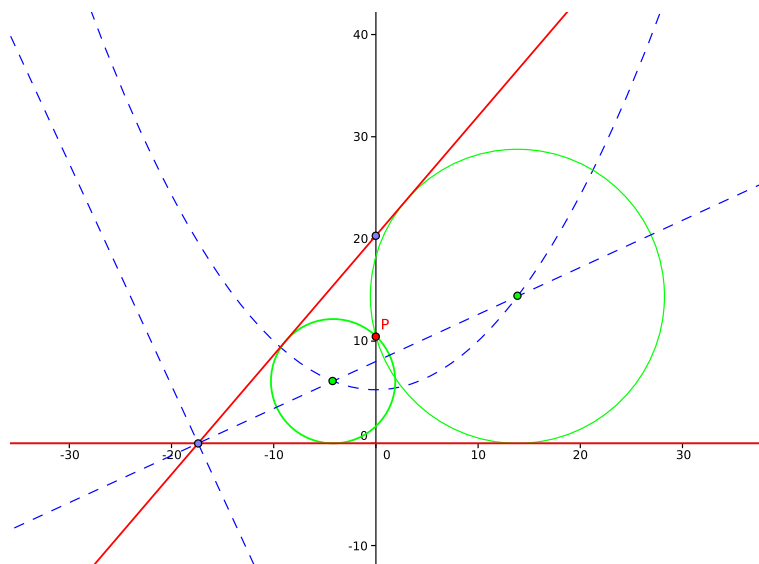


Abb. 7: PLL-Fall für sich schneidende Geraden in normierter Lage

geeignete Ortslinie. Will man erreichen, dass diese Parabel in Normallage ist, ist eine Transformation nötig. So wird die Ausgangslage gedreht und verschoben, sodass eine Gerade mit der  $x$ -Achse zusammenfällt und der Punkt auf der  $y$ -Achse liegt. Die Parabel und die Winkelhalbierenden sind in Abb. 7 blau gestrichelt eingezeichnet. Die Formel für die Parabel ist wie im PPL-Fall mit  $y = \frac{1}{2 * P.y} x^2 + \frac{P.y}{2}$  gegeben<sup>6</sup>. In [8] werden

<sup>6</sup>Während im PPL- und CPP-Fall noch zwischen Ausgangsobjekten und den in Normallage transformierten Objekten z.B. durch Notationen wie  $P$  und  $P'$  unterschieden wurde, so wird ab jetzt zur

die Formeln  $w_1 : y = \frac{m}{\sqrt{m^2 + 1} + 1}x$  und  $w_2 : y = -\frac{\sqrt{m^2 + 1} + 1}{m}x$  für die Winkelhalbierenden von zwei Geraden  $g : y = 0$  und  $h : y = mx$  entwickelt. In der Situation von Abb. 7 müssen diese Winkelhalbierenden aber noch in x-Richtung verschoben werden. Sei der Schnittpunkt der beiden Geraden  $S(x_s, 0)$ , dann ergeben sich die Formeln  $w_1 : y = \frac{m}{\sqrt{m^2 + 1} + 1}(x - x_s)$  und  $w_2 : y = -\frac{\sqrt{m^2 + 1} + 1}{m}(x - x_s)$ . Kürzt man die Steigung der Winkelhalbierenden mit  $m'$  ab und setzt die Winkelhalbierenden mit der Parabel gleich, so ist für die Bestimmung der Berührungsmittelpunkte folgende quadratische Gleichung zu lösen:

$$\frac{1}{2P \cdot y}x^2 - m'x + \frac{P \cdot y}{2} + m'x_s = 0$$

An ein paar Stellen gilt es noch etwas aufzupassen. Keine reguläre Lösung gibt es, wenn der Punkt mit dem Schnittpunkt der Geraden zusammenfällt. Liegt der Punkt  $P$  auf der x-Achse, degeneriert die Parabel zu der Senkrechten in  $P$  zur x-Achse, also zur y-Achse, da  $P$  auf dieser liegt. Deswegen ist der y-Achsenabschnitt der Winkelhalbierenden zu berechnen. Ist die Gerade, die nicht auf der x-Achse liegt, senkrecht zur x-Achse, dann können die Winkelhalbierenden nicht auf die Weise berechnet werden, wie es oben vorgestellt wurde. Für diesen Fall ist  $m'$  einfach auf 1 bzw. -1 zu setzen.

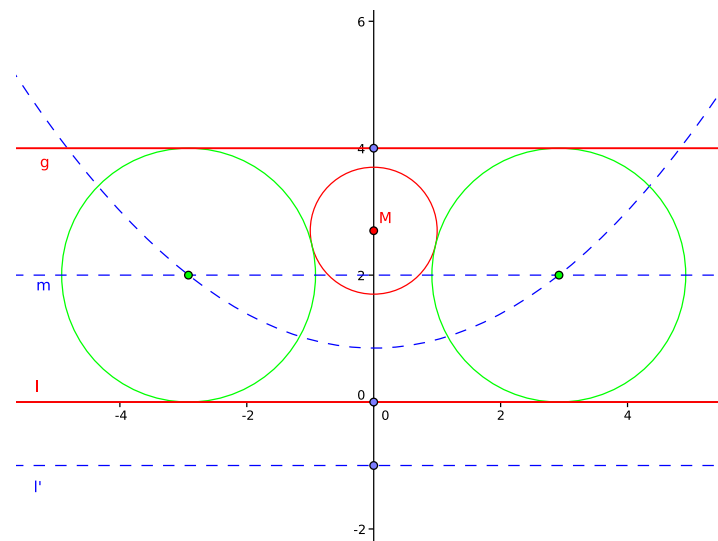
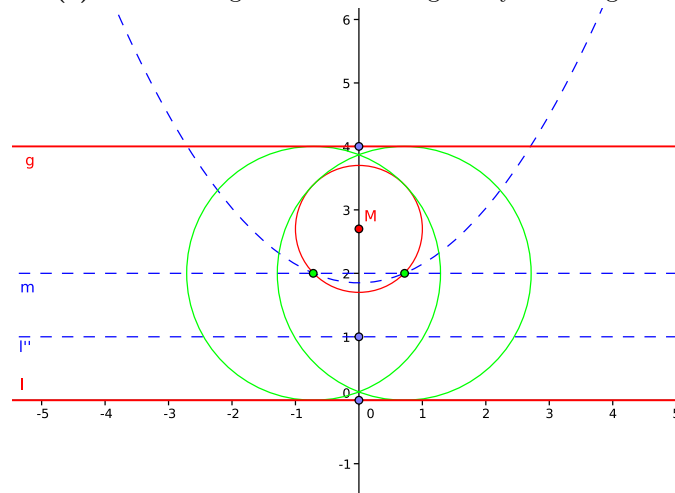
### 2.2.5 CLL

Der vorstehende Fall wird nun dadurch erweitert, dass anstatt des Punktes ein Kreis vorhanden ist. An der prinzipiellen Herangehensweise an das Problem ändert sich jedoch nichts und man wird sogar feststellen, dass man Teile wiederverwenden kann. Das sieht man auch schon daran, dass nach [8] Kapitel 5.2 die Ortslinie der Punkte, die gleichen Abstand von einer Geraden und einem Kreis haben, wie bei einem Punkt und einer Geraden die Parabel ist. Es ist nur eine Verschiebung der Geraden einmal um  $r$  und einmal um  $-r$  in Richtung des Lotes des Kreismittelpunkts auf die Gerade nötig.

Zunächst nehmen wir an, dass die Geraden parallel sind. Dann wird die Konstellation so transformiert, dass eine der beiden Geraden mit der x-Achse zusammenfällt und der Mittelpunkt des Kreises auf der y-Achse liegt. Dies ist auch in der Abb. 8a dargestellt mit den Geraden  $g$  und  $l$  und dem Kreis mit Mittelpunkt  $M$  und Radius  $r$ . Nun wird wie oben erwähnt die Gerade  $l$  um  $r$  in negative y-Richtung verschoben. Dies wird durch die Hilfslinie  $l'$  symbolisiert. Ebenso kann die Gerade  $l$  um  $r$  in positive y-Richtung verschoben werden, was in Abb. 8b durch die Gerade  $l''$  dargestellt wird. Geht man von der Definition einer Parabel als Menge der Punkte, die von einer gegebenen Geraden und einem Punkt den gleichen Abstand haben, aus und zieht die spezielle Lage des Punktes und der Geraden in Betracht, so ist die Parabel, die durch den Punkt  $M$  und die Gerade  $l'$  bzw.  $l''$  erzeugt wird, von der Form  $y = \frac{1}{2(M \cdot y \pm r)}x^2 + \frac{M \cdot y \mp r}{2}$ . Die Mittellinie der beiden Geraden sei  $m : y = y_m$ . Die x-Koordinaten der Mittelpunkte der Berührungskreise

---

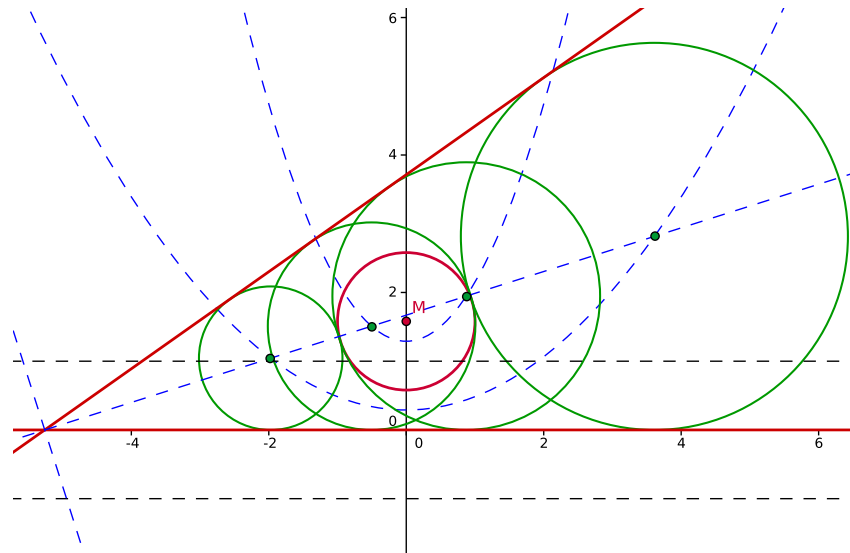
Vereinfachung darauf verzichtet.

(a) Verschiebung von  $l$  um  $r$  in negative  $y$ -Richtung(b) Verschiebung von  $l$  um  $r$  in positive  $y$ -Richtung**Abb. 8:** CLL-Fall für parallele Geraden in normierter Lage

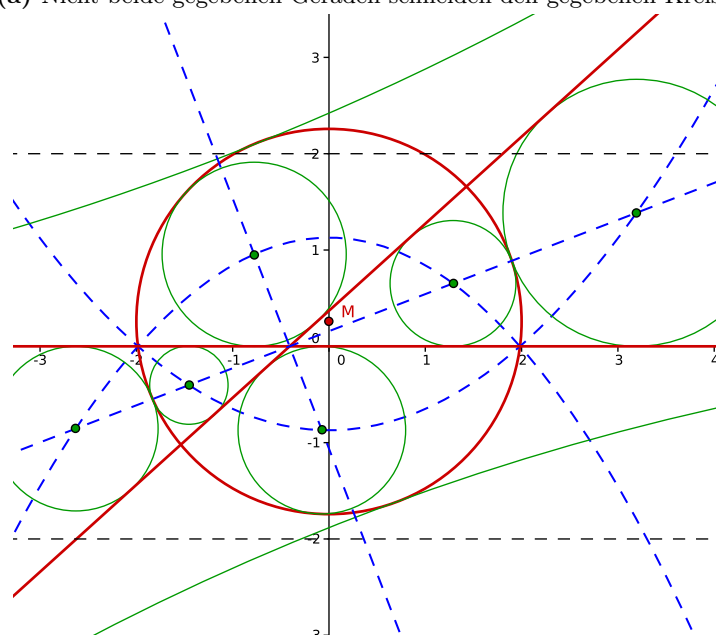
ergeben sich, wenn man die Mittellinie  $m$  mit der Parabel schneidet, d.h. die Gleichung

$$\frac{1}{2(M \cdot y \pm r)} x^2 + \frac{M \cdot y \mp r}{2} - y_m = 0$$

nach  $x$  auflöst. Die  $y$ -Koordinaten sind natürlich  $y_m$  und die Radien  $|y_m|$ . Dabei kann es zu maximal 4 regulären Lösungen kommen, die in den Abb. 8a und 8b als grüne Kreise eingezeichnet sind. In einem Spezialfall muss erneut betrachtet werden, wenn der Nenner in der Formel für die Parabel 0 wird, d.h. die Parabel zur Senkrechten zu  $l$  und  $g$  degeneriert. Dies ist nur möglich, wenn  $l$  den Kreis berührt. Wie man mit dieser Situation umgeht, wurde schon unter PLL beschrieben.



(a) Nicht beide gegebenen Geraden schneiden den gegebenen Kreis



(b) Beide gegebenen Geraden schneiden den gegebenen Kreis

Abb. 9: CLL-Fall für sich schneidende Geraden

Mit dem Wissen um die Verschiebung lässt sich der Fall, dass sich die Geraden schneiden, ebenfalls mit dem Ansatz für das analoge Problem im PLL-Fall lösen. Die Konstellation wird so transformiert, dass eine Gerade mit der x-Achse zusammenfällt, sich insbesondere die Geraden auf der x-Achse schneiden und der Mittelpunkt des Kreises auf der y-Achse liegt. Die Formeln für die Winkelhalbierenden können aus PLL übernommen werden und die Formeln für die Parabeln aus obigem Abschnitt für parallele Geraden. Abschließend wird je eine Winkelhalbierende mit einer Parabel geschnitten, d.h. es wird ein quadratisches Gleichungssystem gelöst, um die Schnittpunkte zu berechnen. Während im PLL-Fall nur maximal 2 reguläre Lösungen möglich waren, sind es nun 4, falls mindestens eine Gerade nicht den Kreis schneidet (siehe Abb. 9a) und 8, falls beide Geraden den Kreis schneiden (siehe Abb. 9b).

### 2.2.6 CCL

In diesem Fall sind zwei Kreise mit Mittelpunkten  $M1$  und  $M2$  und Radien  $r1$  und  $r2$  und eine Gerade  $l$  gegeben. Transformiert man die Lage so, dass die Gerade mit der x-Achse zusammenfällt und einer der beiden Mittelpunkte auf der y-Achse liegt, so kann sofort das Wissen aus den Abschnitten PPL und CLL angewandt werden. Die Mittelpunkte der Berührungskreise ergeben sich daher als Schnittpunkte der Parabeln, die durch den Mittelpunkt  $M1$  und einer Parallelen zur x-Achse mit Abstand  $r1$  bzw. den Mittelpunkt  $M2$  und einer Parallelen zur x-Achse mit Abstand  $r2$  gegeben sind. Zu sehen ist das in der Abb. 10, wobei jedoch die Parallelen zur x-Achse aus Übersichtsgründen ausgeblendet wurden.

Formal bedeutet dies, es werden die Parabeln

$$y = \frac{1}{2(M1.y \mp r1)}x^2 + \frac{M1.y \pm r1}{2}$$

und

$$y = \frac{1}{2(M2.y \mp r2)}(x - M2.x)^2 + \frac{M2.y \pm r2}{2}$$

geschnitten, was zur Lösung folgender quadratischen Gleichung führt:

$$\left(\frac{1}{M1.y \mp r1} - \frac{1}{M2.y \mp r2}\right)x^2 + 2\frac{M2.x}{M2.y \mp r2}x + M1.y \pm r1 - M2.y \mp r2 - \frac{M2.x^2}{M2.y \mp r2} = 0$$

Gesondert betrachtet werden müssen erneut Konstellationen, in denen ein Kreis die x-Achse berührt, d.h. der Mittelpunkt dieses Kreises auf einer Parallelen zur x-Achse mit Abstand  $r$  liegt. Dann wird aus der Parabel das Lot durch diesen Mittelpunkt auf die x-Achse. Außerdem ist zu beachten, dass zur Lösung der quadratischen Gleichung durch den Koeffizienten des  $x^2$ -Terms geteilt wird. Deshalb muss dieser Koeffizient auf 0 geprüft werden. Geometrisch bedeutet dies, dass der Abstand des jeweiligen Kreismittelpunkts zur jeweiligen Parallelen zur x-Achse gleich ist. Der dazugehörige Lösungskreis würde zu einer Geraden degenerieren. Dadurch fällt also der quadratische Teil der Gleichung weg und es muss nur eine lineare Gleichung gelöst werden. Keine Lösung gibt es, wenn sich beide Kreise komplett auf verschiedenen Seiten der x-Achse befinden.

Man kann in Abb. 10 auch sehen, dass es maximal 8 reguläre Lösungen für den CCL-Fall

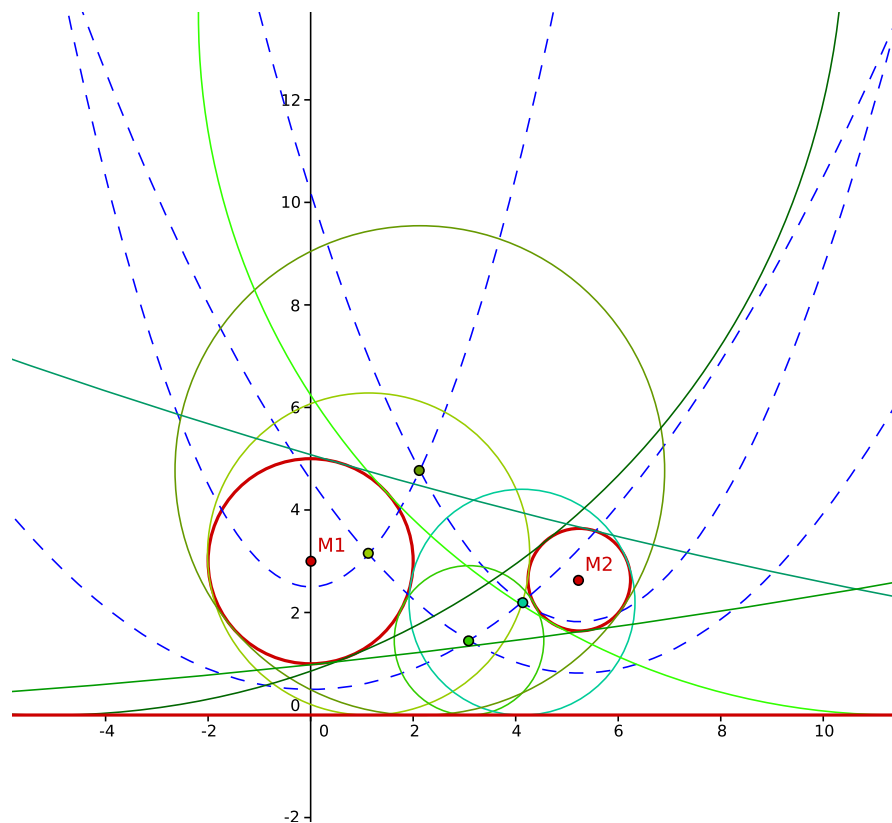


Abb. 10: CCL-Fall in transformierter Lage

gibt. Weniger Lösungen werden es, wenn einer der gerade erwähnten Spezialfälle auftritt.

### 2.2.7 CPL

Der CPL-Fall vereint nun alle drei möglichen Objekte. Es sind also eine Gerade  $g$ , ein Kreis  $k$  mit Radius  $r$  und Mittelpunkt  $M$  und ein Punkt  $P$  gegeben.

Die Lösungen werden wie im CCL-Fall ermittelt, indem die Lage zunächst so transformiert wird, dass die Gerade auf der x-Achse liegt und  $P$  auf der y-Achse. Dann werden die Parabeln, die durch  $P$  und die x-Achse bzw.  $M$  und einer Parallelen zur x-Achse mit Abstand  $r$  bestimmt werden, geschnitten. Die Formeln dazu sind analog zu denen im CCL-Abschnitt, nur dass der Radius eines Kreises auf 0 gesetzt wurde:

$$1. \text{ Parabel: } y = \frac{1}{2 * P.y} x^2 + \frac{P.y}{2}$$

$$2. \text{ Parabel: } y = \frac{1}{2(M.y \mp r)} (x - M.x)^2 + \frac{M.y \pm r}{2}$$

$$\text{quadratische Gleichung: } \left( \frac{1}{P.y} - \frac{1}{M.y \mp r} \right) x^2 + 2 \frac{M.x}{M.y \mp r} x + P.y - M.y \mp r - \frac{M.x^2}{M.y \mp r} = 0$$

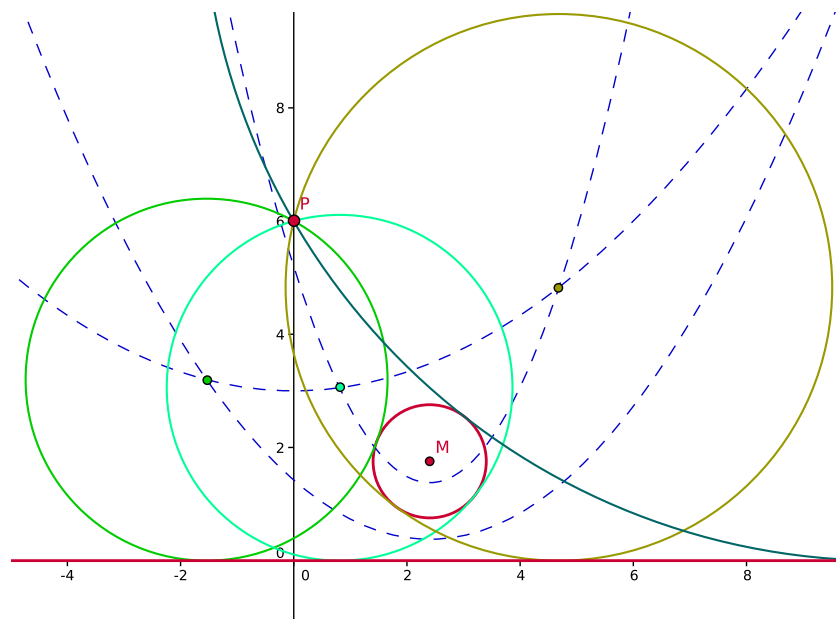


Abb. 11: CPL-Fall in normierter Lage

Dabei ist wieder Acht zu geben, nicht durch 0 zu teilen. Das heißt, es müssen die Situationen, in denen  $P$  auf der  $x$ -Achse liegt oder  $M$  auf einer parallelen Verschiebung der  $x$ -Achse, also die jeweilige Parabel zum Lot auf der  $x$ -Achse wird, gesondert behandelt werden. Außerdem vereinfacht sich die quadratische Gleichung zu einer Linearen, wenn der Koeffizient des quadratischen Anteils 0 ist. Keine Lösung erhält man, wenn der Punkt und der Kreis komplett in verschiedenen Halbebenen bzgl. der  $x$ -Achse liegen oder sich der Punkt im Kreis befindet, aber die Gerade den Kreis weder berührt noch schneidet. Unendlich viele Lösungen würde es geben, wenn sich der Punkt auf der Geraden befindet und der Kreis die Gerade in diesem Punkt berührt.

In der Abb. 11 wird der CPL-Fall mit den rot eingezeichneten gegebenen Elementen in transformierter Lage dargestellt. Die Parabeln sind blau gestrichelt angedeutet, wohingegen die Lösungskreise in Grünstufen eingezeichnet sind. Aus der Zeichnung und mit den Formeln kann man sich überlegen, dass es maximal 4 reguläre Lösungen gibt.

### 2.2.8 CCP

Für den Fall, dass zwei Kreise und ein Punkt gegeben sind und man nach dem Kegelschnitt-Ansatz vorgehen möchte, müsste man zwei Elemente der Menge {Ellipse, Hyperbel} schneiden. Im Allgemeinen ist es nun durch Drehung und Verschiebung nicht mehr möglich, die Ausgangslage so zu transformieren, dass sich beide Hyperbeln oder Ellipsen in Hauptlage befinden. Deshalb wurde hier der Ansatz mit den Kreisgleichungen gewählt. Man bezeichnet dazu den Mittelpunkt des gesuchten Berührkreises mit  $M(x, y)$  und den Radius mit  $r$ . Zusammen mit den beiden gegebenen Kreisen mit Mittelpunkten  $M_i(x_i, y_i)$

und Radien  $r_i$  mit  $i \in \{1, 2\}$  und dem Punkt  $P(x_3, y_3)$  erhält man nach der Idee des Ansatzes mit allgemeiner Kreisgleichung folgendes Gleichungssystem:

$$\begin{aligned}(x - x_1)^2 + (y - y_1)^2 &= (r \pm r_1)^2 \\(x - x_2)^2 + (y - y_2)^2 &= (r \pm r_2)^2 \\(x - x_3)^2 + (y - y_3)^2 &= r^2\end{aligned}$$

Es ergeben sich also maximal 4 reguläre Lösungen. Um erneut die nötigen Umformungen etwas übersichtlicher gestalten zu können, transformiert man die Ausgangslage derart, dass sich  $M_1$  auf der x-Achse befindet und  $P$  mit dem Ursprung zusammenfällt. Damit vereinfacht sich das obige Gleichungssystem folgendermaßen:

$$\begin{aligned}(x - x_1)^2 + y^2 &= (r \pm r_1)^2 & (1) \\(x - x_2)^2 + (y - y_2)^2 &= (r \pm r_2)^2 & (2) \\x^2 + y^2 &= r^2 & (3)\end{aligned}$$

Formt man die einzelnen Gleichungen etwas um und subtrahiert (1) von (2) erhält man die Gleichung  $ax + by + cr = d$  (4), wobei gilt:

$$\begin{aligned}a &= 2(x_1 - x_2) \\b &= -2 * y_2 \\c &= 2(\pm r_1 \mp r_2) \\d &= (x_1^2 - r_1^2) - (x_2^2 + y_2^2 - r_2^2)\end{aligned}$$

Ebenso erhält man die Gleichung  $a'x + c'r = d'$  (5) mit  $a' = 2 * x_1$ ,  $c' = \pm 2 * r_1$  und  $d' = x_1^2 - r_1^2$ , wenn man (3) - (1) berechnet.

Löst man die linearen Gleichungen (4) und (5) nach den beiden Unbekannten  $x$  und  $y$  auf, so erhält man:

$$\begin{aligned}x &= \frac{d' - c'r}{a'} \\y &= \frac{a'd - ad' + (ac' - a'c)r}{a'b}\end{aligned}$$

Setzt man diese  $x$  und  $y$  in die Formel  $x^2 + y^2 = r^2$  ein, kann man  $r$  berechnen und damit  $x$  und  $y$ .

Nun ist aber noch anzumerken, dass es für  $r$  als Lösung der quadratischen Gleichung im Allgemeinen zwei Lösungen gibt und es somit mit den Wahlmöglichkeiten in (1) und (2), bei denen man  $r_1$  bzw.  $r_2$  addieren oder subtrahieren kann, 8 Lösungen geben würde. Dies gibt aber geometrisch keinen Sinn, da es maximal 4 reguläre Lösungen geben kann. Also werden doppelte Lösungen erzeugt. Betrachtet man die Berechnung genauer, so kann man in der Tat sehen, dass eine Vorzeichenänderung von  $r_1$  keine neue Lösung erzeugt. Dies wird hier aber nicht genauer ausgeführt, kann jedoch leicht nachgerechnet werden. Wie bei den anderen Fällen soll auch hier kurz ein Blick darauf geworfen werden, wann es überhaupt Lösungen gibt und für welche Fälle weniger als 4 Lösungen vorhanden sind: Keine Lösung gibt es, wenn der Punkt innerhalb<sup>7</sup> eines Kreises ist, der andere Kreis

<sup>7</sup>Innerhalb bedeutet, dass der Abstand vom Mittelpunkt des Kreises zu dem Punkt kleiner als der Radius ist.



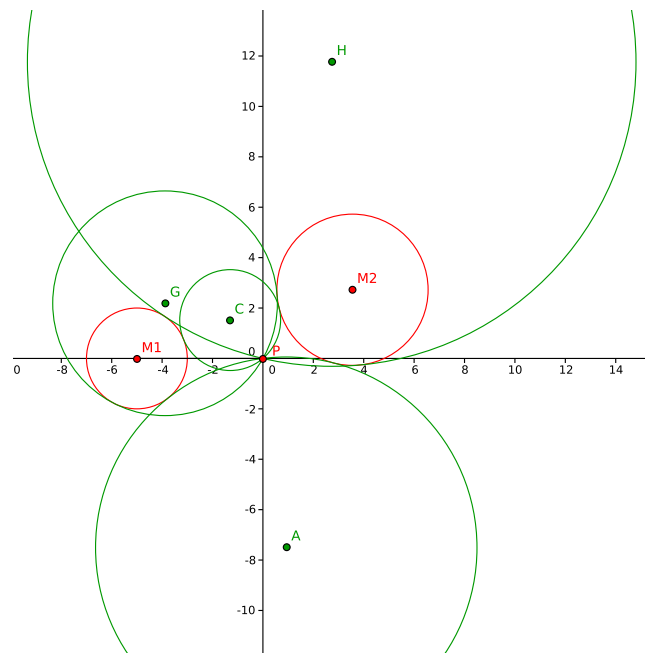


Abb. 12: CCP-Fall in transformierter Lage

diesen Kreis jedoch weder berührt noch schneidet. Umgekehrt existiert ebenfalls keine Lösung, wenn der Punkt außerhalb<sup>8</sup> des einen Kreises ist und der andere Kreis vollständig innerhalb des einen ist.

Unendlich viele Lösungen würde es geben, wenn der Punkt und der Berührungspunkt der Kreise, falls es ihn gibt, zusammenfallen.

Liegt der Punkt auf einem der beiden Kreise oder sind die Kreise konzentrisch, so gibt es nur maximal zwei Lösungen.

In Abb. 12 ist der CCP-Fall in transformierter Lage abgetragen. Die gegebenen Kreise und der Punkt sind rot eingezeichnet und die Lösungskreise grün.

### 2.2.9 LLL

Wie eingangs dieses Kapitels erwähnt, wurde der Fall von drei Geraden nicht implementiert und deshalb wird hier der Vollständigkeit halber nur kurz eine Lösungsidee vorgestellt. In den Fällen mit zwei Geraden wurde gezeigt, dass die Winkelhalbierenden die Ortslinien der Punkte sind, die von den jeweiligen Geraden gleichen Abstand haben. Zur Lösung des Problems müssen nun zwei solche Winkelhalbierendenpaare miteinander geschnitten werden. Deshalb gibt es im Allgemeinen auch 4 Lösungen. Transformiert man die Ausgangslage so, dass eine Gerade mit der x-Achse zusammenfällt, dann können sogar die unter PLL vorgestellten Formeln für die Winkelhalbierenden analog verwendet werden.

<sup>8</sup>Außerhalb bedeutet, dass der Abstand vom Mittelpunkt des Kreises zu dem Punkt größer als der Radius ist.

Dabei wurde jetzt davon ausgegangen, dass keine zwei Geraden parallel sind. Sind alle drei Geraden echt parallel, so gibt es keine Lösung. Sind zwei echt parallel, so kann die Mittellinie der parallelen Geraden mit einem Winkelhalbierendenpaar geschnitten werden.

### 2.2.10 CCC

Auch für die Situation von drei gegebenen Kreisen folgt nur eine kurze Beschreibung. Hierfür kann der CCP-Fall herangezogen werden. Bei der Berechnung muss nur der Radius des 3. Kreises miteinbezogen werden. Das führt auch dazu, dass es in der Regel 8 Lösungen des CCC-Falls gibt.

## 2.3 Korrektheit des Kegelschnittansatzes

Bei den Ansätzen mit Kegelschnitten wurde stets so vorgegangen, dass zwei der drei Kegelschnitte, die die Ortslinien aller Punkte beschreiben, die jeweils von zwei Objekten den gleichen Abstand haben, geschnitten wurden. Die Auswahl wurde dabei so getroffen, dass die Formeln, die die Kegelschnitte beschreiben, möglichst einfach<sup>9</sup> sind, um den Rechenaufwand zu verringern. Dabei stellt sich jedoch die Frage, ob die Wahl von zwei anderen Kegelschnitten nicht eine andere oder mehr Lösungen hervorrufen würden. Vorausgesetzt wird dazu, dass die jeweiligen Kegelschnitte die oben beschriebenen Ortslinieneigenschaft erfüllen. Bezeichne die drei gegebenen Objekte als  $A$ ,  $B$  und  $C$  und die zugehörigen Kegelschnitte als  $K_{A,B}$ ,  $K_{A,C}$  und  $K_{B,C}$ .

Dass ein Schnittpunkt  $S$  zweier solcher Kegelschnitte die Eigenschaft erfüllt, dass er von  $A$ ,  $B$  und  $C$  gleich weit entfernt ist, wird zunächst gezeigt. O.B.d.A wird angenommen, dass  $S$  ein Schnittpunkt von  $K_{A,B}$  und  $K_{A,C}$  ist. Dann folgt nach Voraussetzung, dass  $S$  als Element von  $K_{A,B}$  zu  $A$  und  $B$  den gleichen Abstand  $r$  besitzt. Ebenso folgt nach Voraussetzung, dass  $S$  als Element von  $K_{A,C}$  zu  $A$  und  $C$  den gleichen Abstand  $r'$  besitzt. Damit erhält man sofort, dass  $r = r'$  gilt und die Behauptung bewiesen ist.

Nun zu der Behauptung, dass die Schnittpunkte von o.B.d.A  $K_{A,B}$  und  $K_{A,C}$  alle gesuchten Punkte beschreiben. Angenommen die Behauptung stimmt nicht, dann bedeutet dies, dass es Punkte gibt, die von  $A$ ,  $B$  und  $C$  gleich weit entfernt sind, aber keine Schnittpunkte von  $K_{A,B}$  und  $K_{A,C}$  sind. Sei  $P$  ein solcher Punkt. Dann ist  $P$  aber nach Voraussetzung ein Element von  $K_{A,B}$ , da er insbesondere von  $A$  und  $B$  gleichen Abstand hat und  $K_{A,B}$  alle solche Punkte beschreibt. Analog gilt, dass  $P$  ein Element von  $K_{A,C}$  ist. D.h.  $P$  liegt auf  $K_{A,B}$  und  $K_{A,C}$ , ist aber kein Schnittpunkt der beiden Kurven, was sofort zu einem Widerspruch führt.

Damit wurde gezeigt, dass man bei den Kegelschnitten die einfacheren wählen und diese schneiden kann, ohne dass man Lösungen verliert.

---

<sup>9</sup>Einfach ist dabei ein intuitiver Begriff, der sich darauf bezieht, wie gut man eine Formel, die die Punkte eines Kegelschnitts beschreibt, für die weiteren Berechnungen verwenden kann. Das bezieht sich unter anderem auf den Grad der Formel und die Anzahl der Variablen. So ist etwa eine lineare Funktion „einfacher“ als  $\sqrt{x}$  oder  $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ .

### 3 Überlegungen zur numerischen Stabilität

Im Folgenden werden die Berechnungsmethoden noch einmal durchleuchtet und auf etwaige numerische Schwachstellen untersucht. Dies sind im Allgemeinen Fälle, bei denen es zu Auslöschungen und dadurch zu Verlust von Genauigkeit kommt oder bei denen durch sehr kleine Zahlen geteilt wird. Dann wird versucht, diese Probleme zu umgehen, oder zumindest weniger kritisch zu gestalten. Dazu muss man sich aber im Klaren sein, wie die Zahlen auf dem Rechner dargestellt werden, um zu verstehen, warum es zu diesen Problemen kommen kann. Es handelt sich um Gleitkommaarithmetik und die Zahlen sind von der Form  $.d_1 * \dots * d_m * 2^e$ . Dabei werden die Ziffern  $d_1 \dots d_m$  als Mantisse bezeichnet und  $e$  als der Exponent. Auf aktueller Hardware in double-Genauigkeit ist eine Mantissenlänge von  $52+1$ <sup>10</sup> und Exponentenlänge von 11 üblich.

Dadurch ergibt sich, dass die kleinste Zahl, die zu 1 addiert werden kann und das Ergebnis sich von 1 unterscheidet, die  $2^{-53} \approx 1.11 * 10^{-16}$  ist. Kleinere Zahlen fallen wegen der begrenzten Mantissenlänge unter den Tisch. Außerdem ist der Zahlenbereich beschränkt und es können natürlich nicht alle reellen Zahlen dargestellt werden. Bei Rechnungen, die eine höhere Genauigkeit als 16 Stellen benötigen würden, tritt deswegen ein relativer Fehler von  $\frac{1}{2}2^{1-m}$  allein auf Grund der Rundung auf die letzte Stelle auf. Wenn zwei ähnlich große Zahlen<sup>11</sup> voneinander abgezogen werden, entstehen beim Subtraktionsprozess führende 0en. Da die Zahlen auf dem Rechner stets normiert gespeichert werden, wird das berechnete Ergebnis so lange geschiftet, bis an der ersten Stelle eine von 0 verschiedene Ziffer steht. Dabei wird mit 0en von hinten aufgefüllt. Dies sind natürlich reine Phantasieziffern, die die Rechnung mit einem teils sehr großen relativen Fehler behaften. Dieser Effekt wird Auslöschung genannt. Mehr dazu kann bei Sauer in [9] nachgelesen werden.

Das sind die Hauptgründe, warum die Ergebnisse, die der Rechner liefert, nicht exakt sind. Man kann dies leider nicht vermeiden, aber man kann durch geschicktes Vorgehen den Fehler verringern. Eine kleine Auswahl solcher Ansätze wird nachfolgend aufgelistet.

#### 3.1 Gleichheitstest

Auf Grund der endlichen Genauigkeit der Zahlendarstellung auf dem Rechner ist eine exakte Berechnung meistens nicht möglich, weshalb ein Test auf Gleichheit (in C-Code „==“) stets mit einer gewissen Epsilon-Umgebung gestaltet werden muss. Soll etwa im Fall PPP überprüft werden, ob die drei Punkte kollinear sind und man dies mit der Determinante überprüfen will, so sieht das folgendermaßen aus:

$$|\det(\overrightarrow{P_1P_2}, \overrightarrow{P_1P_3})| < \epsilon$$

Es ist nicht sinnvoll, das  $\epsilon$  kleiner als  $1.11 * 10^{-16}$  zu wählen, weil es im Vergleich mit der 1 wegfallen würde. Dieses  $\epsilon$  wird auch häufig als „numerische Null“ bezeichnet und

---

<sup>10</sup>Das +1 rührt von einem sogenannten 'guard digit', das bei Gleitkommaoperationen für zusätzliche Genauigkeit sorgt und damit den Fehler der Operation deutlich verringert.

<sup>11</sup>Ähnlich groß bedeutet, die ersten paar Stellen dieser Zahlen, die Größenordnung und das Vorzeichen sind gleich.

wurde bei der Implementierung im Bereich  $10^{-14}$  gewählt.

### 3.2 Kollinearitätstest

An mehreren Stellen ist es nötig, zu prüfen, ob Vektoren kollinear sind. Ein Beispiel dafür ist der PPP-Fall, da der Lösungskreis zu einer Geraden degeneriert, falls die gegebenen Punkte kollinear sind. Eine Möglichkeit, dies zu prüfen, bildet die Determinante. Dabei stellt sich die Frage, ob die Determinante dafür die beste Methode ist. Um dies untersuchen zu können, ist die Rundungsfehleranalyse ein geeignetes Mittel. Generell ist dabei zu sagen, dass die Determinanten bei Numerikern als numerisch instabil gelten, da es dazu kommen kann, dass Zahlen gleichen Vorzeichens voneinander abgezogen werden und damit Auslöschung eintritt, wenn die Zahlen etwa gleich groß sind. Außerdem ist die Determinante stark anfällig gegen kleine Störungen. Nimmt man das Standardmodell der Gleitkommaarithmetik als Grundlage (siehe [9]), kann man alle Gleitkommaoperationen mit einem relativen Fehler von  $u = 2^{-m+1}$  nach oben abschätzen, wobei  $m$  die Mantissenlänge ist. Der Notation von [9] folgend, bezeichnet  $\odot$  mit  $\cdot \in \{+, -, *, /\}$  die auf dem Computer fehlerbehaftete Rechnung in Gleitkommaarithmetik.

#### 3.2.1 Determinantenverfahren

Geht man nun von den zweidimensionalen Punkten  $A$ ,  $B$  und  $C$  mit den Koordinaten  $x$  und  $y$  und den Verbindungsstrecken  $[AB]$  und  $[AC]$  aus, so berechnet sich der relative Fehler bei der Determinantenbestimmung wie folgt:

$$\frac{|\odot(\overrightarrow{AB}, \overrightarrow{AC}) - \det(\overrightarrow{AB}, \overrightarrow{AC})|}{|\det(\overrightarrow{AB}, \overrightarrow{AC})|}$$

wobei

$$\odot(\overrightarrow{AB}, \overrightarrow{AC}) = (B.x \odot A.x) \otimes (C.y \odot A.y) \ominus (B.y \odot A.y) \otimes (C.x \odot A.x)$$

die Berechnung der Determinante mit rundungsfehlerbehafteter Gleitkommaarithmetik ist. Schätzt man den relativen Fehler nach oben ab, erhält man

$$(4u + 6u^2 + 4u^3 + u^4) \frac{|(B.x - A.x) * (C.y - A.y)| + |(B.y - A.y) * (C.x - A.x)|}{|(B.x - A.x) * (C.y - A.y) - (B.y - A.y) * (C.x - A.x)|}$$

Dies zeigt, dass die obere Schranke für den Fehler beliebig groß wird, wenn die Vektoren beinahe kollinear sind. Zu sehen ist das an folgendem Minimalbeispiel in Matlab-Code:

```
% definiere A, B und C; B und C sind Vielfache von A
A = [2.123456789; 1.0]
B = 2*A
C = 1e6*A

% berechne die Matrix aus den Differenzvektoren
M = [B-A, C-A]

% bestimme die Determinante
det(M)           % 4.944058109540492e-10
```

Eigentlich sollte die Determinante 0 sein, da die Punkte kollinear sind. Aber durch die Subtraktion werden die Differenzvektoren linear unabhängig. Mit ungefähr  $5 * 10^{-10}$  ist das Ergebnis deutlich von der numerischen 0 entfernt.

### 3.2.2 Normierungsverfahren

Ein weiterer Ansatz wäre, die Differenzvektoren einfach zu normieren. Dann muss nur noch auf Gleichheit oder Gegengleichheit getestet werden. Im Folgenden wird ausschließlich der Test auf Gleichheit durchgeführt. Der Test auf Gegengleichheit verläuft analog. Da für die Bestimmung der Länge eines Vektors die Wurzel berechnet werden muss, ist zu überprüfen, ob die Wurzel als Gleitkommaoperation auch mit dem Rundungsfehler  $u$  abgeschätzt werden kann. Nach Mikhlin in [7] Kapitel I Abschnitt 1.4 ist dies möglich. Man kann nun auch den relativen Fehler bei dem Verfahren mit Normieren nach oben abschätzen und bekommt exemplarisch in der x-Koordinate

$$\left| \frac{\frac{\textcircled{xB}}{\sqrt{\textcircled{xB} \otimes \textcircled{xB} \oplus \textcircled{yB} \otimes \textcircled{yB}}} \ominus \frac{\textcircled{xC}}{\sqrt{\textcircled{xC} \otimes \textcircled{xC} \oplus \textcircled{yC} \otimes \textcircled{yC}}}}{\left| \frac{xB}{\sqrt{xB^2 + yB^2}} - \frac{xC}{\sqrt{xC^2 + yC^2}} \right|} - \left( \frac{xB}{\sqrt{xB^2 + yB^2}} - \frac{xC}{\sqrt{xC^2 + yC^2}} \right) \right|$$

$$\leq \left( \frac{(1+u)^2}{(1-u)^3} - 1 \right) \frac{\frac{|xB|}{\sqrt{xB^2 + yB^2}} + \frac{|xC|}{\sqrt{xC^2 + yC^2}}}{\left| \frac{xB}{\sqrt{xB^2 + yB^2}} - \frac{xC}{\sqrt{xC^2 + yC^2}} \right|}$$

mit  $\textcircled{xB} := B.x \ominus A.x$ ,  $\textcircled{xC} := C.x \ominus A.x$ ,  $\textcircled{yB} := B.y \ominus A.y$  und  $\textcircled{yC} := C.y \ominus A.y$ .

Analog dazu sind  $x_B$ ,  $x_C$ ,  $y_B$  und  $y_C$  definiert, nur mit exakter Berechnung.

Wenn man die y-Koordinate miteinbezieht, sieht man, dass auch hier der Fehler bei annähernder Kollinearität groß wird. Warum sollte man dennoch die Normierung wählen und nicht die Determinante? Das liegt an der Anwendung. Die Determinante ist sehr sprunghaft, wie oben schon erwähnt. D.h. ändert man die Werte eines Punktes leicht, kann es sein, dass sich die Determinante signifikant ändert. Das ist aber ein Verhalten, das bei diesem geometrischen Problem nicht erwünscht ist. Sei etwa  $B'(B.x + \delta, B.y)$  mit  $|\delta|$  relativ klein. Dann gilt:

$$\det(\overrightarrow{AB'}, \overrightarrow{AC}) = (B'.x - A.x)(C.y - A.y) - (C.x - A.x)(B.y - A.y) = \det(\overrightarrow{AB}, \overrightarrow{AC}) + \delta(C.y - A.y)$$

Daraus wird das Problem der Determinante schon ersichtlich. Wenn man nämlich die x-Koordinate von  $B$  ein bisschen ändert und  $|C.y - A.y|$  groß ist, ändert sich die Determinante stark, obwohl  $B$  unabhängig von  $|C.y - A.y|$  ist. Nun betrachten wir die normierten Vektoren unter der Annahme, dass sich  $B.x$  um  $\delta$  ändert:

$$\frac{B'.x - A.x}{AB'} - \frac{C.x - A.x}{AC} = \frac{\delta + B.x - A.x}{\sqrt{\delta^2 + 2\delta(B.x - A.x) + (B.x - A.x)^2 + (B.y - A.y)^2}} - \frac{C.x - A.x}{AC}$$

Nimmt man an, dass  $|\delta|$  sehr klein ist und die Punkte ausreichend voneinander entfernt sind, kann man folgern, dass

$$\sqrt{\delta^2 + 2\delta(B.x - A.x) + (B.x - A.x)^2 + (B.y - A.y)^2} \approx \overline{AB}$$

gilt und damit

$$\frac{B'.x - A.x}{AB} - \frac{C.x - A.x}{AC} = \frac{B.x - A.x}{AB} - \frac{C.x - A.x}{AC} + \frac{\delta}{AB}$$

Für den Vergleich der y-Koordinaten ändert sich durch die Addition von  $\delta$  auf B.x nur  $\overline{AB}$ :

$$\frac{B.y - A.y}{AB'} - \frac{C.y - A.y}{AC} \approx \frac{B.y - A.y}{AB} - \frac{C.y - A.y}{AC}$$

Die Addition von  $\delta$  entspricht also in der x- und y-Koordinate nur einer kleinen Änderung, die von der Länge von  $\overline{AB}$  abhängt, was einen nachvollziehbareren Zusammenhang darstellt. Wenn die Punkte A und B nämlich weit entfernt sind, wird die Auswirkung der Änderung immer schwächer, was in der Anwendung ein gewünschtes Verhalten ist. Dass ausschließlich die x-Koordinate verändert wurde, dient der Vereinfachung. Der Effekt ist ein Ähnlicher, wenn man beide Koordinaten leicht verändert, nur werden dann die Formeln und Zusammenhänge unübersichtlicher.

Beim Determinanten- und Normierungsverfahren wurden stets die Vektoren  $[AB]$  und  $[AC]$  auf Kollinearität überprüft. Dabei stellt sich die Frage, ob es geschickt ist, statisch diese zwei Vektoren zu nehmen, bzw. ob die Stabilität des Tests von der Wahl der Vektoren unabhängig ist. Dem ist natürlich nicht so. Betrachtet man dieses Problem unter dem Modell der Rückwärtsfehleranalyse (siehe [9]), so sieht man leicht ein, dass ein längerer Vektor unter dem gleichen absoluten Rückwärtsfehler besser konditioniert ist, als ein kurzer Vektor. Das heißt, seine Richtung ändert sich weniger.

### 3.2.3 Rotationsverfahren

Folgende Möglichkeit, auf Kollinearität zu testen, verwendet Drehmatrizen und berücksichtigt auch die Länge der Vektoren. Seien die zu untersuchenden Vektoren wie oben  $\overrightarrow{AB}$ ,  $\overrightarrow{AC}$  und  $\overrightarrow{BC}$  mit o.B.d.A.  $\overline{BC} \geq \overline{AB} \geq \overline{AC}$ . Da, wie schon beschrieben, die längeren Vektoren stabiler sind, werden für den Test der linearen Abhängigkeit die Vektoren  $\overrightarrow{BC}$  und  $\overrightarrow{AB}$  verwendet. Dazu wird eine Drehmatrix  $Q$  bestimmt, durch die  $\overrightarrow{BC}$  auf ein Vielfaches von  $e_1$  abgebildet wird. Wie eine solche Matrix  $Q$  bestimmt werden kann, wird im Abschnitt „Transformation in Normallagen“ erwähnt. Anschließend wird der Vektor  $\overrightarrow{AB}$  mit  $Q$  multipliziert:  $\overrightarrow{AB'} := Q * \overrightarrow{AB}$ . Sind  $\overrightarrow{AB}$  und  $\overrightarrow{BC}$  kollinear, so muss der Vektor  $\overrightarrow{AB}$  durch die Drehung, die er bei der Multiplikation erfährt, auch auf ein Vielfaches von  $e_1$  abgebildet werden. Die Überprüfung sieht am Ende so aus, dass getestet wird, ob  $\frac{\overrightarrow{AB'}.y}{\overrightarrow{AB'}.x}$  bis auf eine  $\varepsilon$ -Abweichung 0 ist. Durch die Division mit der x-Koordinate wird noch berücksichtigt, dass bei einem sehr langen Vektor eine kleine Abweichung in der

y-Koordinate weniger Gewicht hat, als bei einem sehr kurzen Vektor.

Dieses Verfahren gilt als sehr stabil. Zum einen wird die Länge der Vektoren berücksichtigt und zum anderen haben die Drehmatrizen Determinante 1 und sind somit gut konditioniert, d.h. die Multiplikation eines Vektors an die Drehmatrix erzeugt keinen großen Fehler.

```

% definiere A, B und C
A = [1.0; 2.123456789];
B = 1e5*A;
C = -200*A;

% berechne die von A, B und C aufgespannten Vektoren
AB = B-A;
AC = C-A;
BC = C-B;

% berechne deren Länge
lenAB = norm(AB,2); % 2.347117072905909e+05
lenAC = norm(AC,2); % 4.717752494065818e+02
lenBC = norm(BC,2); % 2.351834825399975e+05

% normiere die Vektoren
ABnorm = AB / lenAB; % (0.426050328525767, 0.904699462563721)
ACnorm = AC / lenAC; % (-0.426050328525767, -0.904699462563721)

M = [AB, AC];

% berechne die Rotationsmatrix
m = BC/norm(BC,2);
n = [-m(2), m(1)];
R = [n(2), -n(1); n(1), n(2)];
% rotiere AB
vec = R*AB; % (-2.347117072905909e+05, 0)

% Kollinearitätstests
det(M) % 0
ABnorm + ACnorm % (0.0000000000000001, 0.0000000000000001)
vec(2)/vec(1) % 0

% verändere B leicht
B(1) = B(1) + 1e-10;

% aktualisiere die Werte
AB = B-A;
BC = C-B;

lenAB = norm(AB,2); % 2.347117072905909e+05
lenAC = norm(AC,2); % 4.717752494065818e+02
lenBC = norm(BC,2); % 2.351834825399975e+05

ABnorm = AB / lenAB; % (0.426050328525768, 0.904699462563721)
ACnorm = AC / lenAC; % (-0.426050328525767, -0.904699462563721)

```

```

M = [AB, AC];
m = BC/norm(BC,2);
n = [-m(2), m(1)];
R = [n(2), -n(1); n(1), n(2)];
vec = R*AB;           % (-2.347117072905910e+05, 0)

% Kollinearitätstests
det(M)                % -0.0000000482813347
ABnorm + ACnorm      % (0.00000000000000005, 0.00000000000000001)
vec(2)/vec(1)         % 0

```

Veranschaulicht werden die gezeigten Verfahren in obigem Matlab-Codebeispiel. Die hinter den jeweiligen Zeilen notierten Zahlen beschreiben die an diesen Stellen berechneten Werte. Gegeben sind zunächst kollineare Punkte  $A$ ,  $B$  und  $C$ , für die alle drei Verfahren auch deutlich anzeigen, dass sie kollinear sind. Dann wird die  $x$ -Koordinate von  $B$  um  $10^{-10}$  geändert, was einer Änderung an der 15. gültigen Stelle entspricht. Dies ist ein sehr geringer Fehler und kann durch etwaige vorhergehende Berechnungen leicht entstehen. Die Determinante springt sofort auf einen Wert im Bereich von  $5 * 10^{-8}$ , was konsistent zur obigen theoretischen Beobachtung ist. Es wird also deutlich angezeigt, dass die Vektoren nicht kollinear sind. Beim normierten Ansatz hingegen ändert sich kaum etwas. Es wird immer noch im Bereich der numerischen Null angezeigt, dass die Vektoren kollinear sind. Das Rotationsverfahren zeigt sich gar völlig unbeeindruckt von der Änderung. Dies wird als robustes Verhalten gegenüber kleinen Fehlern bezeichnet und ist eine in der Anwendung gewünschte Eigenschaft.

### 3.3 Auswahl der Kegelschnitte

Bei dem Lösungsansatz mit den Kegelschnitten stehen drei solche zur Auswahl, von denen zwei ausgewählt wurden, um sie miteinander zu schneiden. Wie im Abschnitt „Korrektheit des Kegelschnittansatzes“ erwähnt wurde, wurden stets die „einfacheren“ Ortslinien miteinander geschnitten. Nun ist es aber im PPP-Fall so, dass drei Mittelsenkrechten gegeben sind. Es handelt sich also um drei Kegelschnitte gleicher Art. In der Numerik ist es ein Leitspruch, dass durch Theorie begründete Wahlfreiheit in der Praxis selten gegeben ist. Es ist numerisch ungünstig zwei beliebige Mittelsenkrechten auszuwählen und diese zu schneiden. Wie schon im vorangehenden Punkt ist es geschickter, die Mittelsenkrechten der beiden längeren Strecken zu schneiden. Bei längeren Strecken führt der gleiche absolute Fehler zu einer geringeren Veränderung der Mittelsenkrechten als bei kürzeren Strecken.

### 3.4 Berechnung des Radius der Umkreise

In den Fällen, in denen der Kegelschnittansatz gewählt wurde, wird zunächst der Berührungsmittelpunkt und im Anschluss durch den Abstand zu einer Komponente der Radius bestimmt. Da der Berührungskreis aber alle Komponenten möglichst exakt tangieren soll, be-



steht eine kleine Optimierung darin, den Abstand zu allen Komponenten zu bestimmen und den Radius als den Durchschnitt dieser Abstände zu wählen.

### 3.5 Transformation in Normallagen

Zur Lösung von Fällen, bei denen z.B. Hyperbeln in allgemeiner Lage vorkommen würden, wird erwähnt, dass das Problem zunächst so transformiert wird, dass sich die Hyperbeln in 1. Hauptlage befinden und damit leichter handhabbar sind. Leichter handhabbar heißt natürlich auch, dass auf Grund der Einfachheit der Formeln weniger Gleitkommaoperationen nötig sind und das Problem damit numerisch stabiler ist. Man muss aber auch erst noch überprüfen, ob die Transformationen an sich nicht numerisch instabil sind. Die vorhandenen Objekte werden dabei stets durch eine Drehung und eine Verschiebung derart neu positioniert, dass eine gegebene Gerade, die durch die beiden Punkte  $A$  und  $B$  bestimmt wird, mit der  $x$ -Achse zusammenfällt. Für die Drehung wird eine Rotationsmatrix verwendet, die die folgende Form hat:

$$R = \begin{pmatrix} n.y & -n.x \\ n.x & n.y \end{pmatrix} \text{ mit } n = \begin{pmatrix} -m.y \\ m.x \end{pmatrix} \text{ und } m = \frac{B - A}{\|B - A\|}$$

Dabei ist  $n$  also der normierte Normalenvektor gegen den Uhrzeigersinn zur gegebenen Geraden. Dass  $R * R^t = I$  und  $\det(M) = 1$  gilt und damit  $R$  eine Rotationsmatrix ist, ist leicht zu zeigen. Noch zu zeigen ist, dass die Gerade nach der Rotation parallel zur  $x$ -Achse ist.

Mit  $A' := R * A$  und  $B' := R * B$  folgt:

$$B'.y - A'.y = n^t * B - n^t * A = n^t * (B - A) = 0$$

Im Allgemeinen ist  $A'.y = n^t * A$  jedoch nicht 0, d.h. es ist noch eine Verschiebung in  $y$ -Richtung um  $-n^t * A$  nötig. Gegebene Punkte  $P$  werden also zu  $P' = R * P - \begin{pmatrix} 0 \\ n^t * A \end{pmatrix}$  transformiert.

Am Ende der Berechnung müssen die Berührungskreismittelpunkte  $M'$ , die in transformierter Lage berechnet wurden, natürlich wieder zurücktransformiert werden, d.h.  $M = R^t * (M' + \begin{pmatrix} 0 \\ n^t * A \end{pmatrix})$  bildet eine Lösung in der Ausgangslage.

Man macht durch die Transformation einen Fehler, wenn man die Ausgangslage zunächst verändert und dann nochmal einen, wenn man die Lösung zurückabbildet. Deshalb muss man sich überlegen, ob die Transformation kritisch für die Genauigkeit des Ergebnisses ist. Die Rotation geschieht durch Multiplikation mit einer Drehmatrix, was numerisch unkritisch ist, da die Matrix mit Determinante 1 gut konditioniert ist. Mehr zur Kondition einer Matrix kann in [9] nachgelesen werden. Dann bleibt nur noch die Verschiebung, die zu Auslöschung führen kann. Jedoch ist dafür keine Verschiebung z.B. des betroffenen Kegelschnitts nötig, weshalb diese Translation das Problem nicht instabiler macht. Deshalb ist die Transformation ein guter Ansatz, da die Berechnung stark vereinfacht wird.

### 3.6 Reihenfolge der Ausführung von Operationen

In der Numerik spielt die Reihenfolge der Ausführung von Operationen eine entscheidende Rolle. Während es beim theoretischen Rechnen im Körper  $\mathbb{R}$  für das Ergebnis unerheblich ist, wie man eine endliche Reihe aufsummiert oder ob man zur Berechnung des Terms  $a^2 - b^2$  die 3. Binomische Formel anwendet oder nicht, macht es in den auf dem Rechner darstellbaren Zahlen durchaus einen Unterschied.

Addiert man eine Reihe von Zahlen, so ist es von Vorteil, diese zu Beginn aufsteigend zu sortieren, sodass zunächst kleine Zahlen miteinander addiert werden. Würde man nämlich sofort die kleinste Zahl mit der Größten addieren, so besteht die Gefahr, dass signifikante Teile der kleinen Zahl im Vergleich zur großen Zahl auf Grund mangelnder gültiger Stellen in der Mantisse verschwinden. Diese numerische Überlegung war jedoch weniger relevant in der Implementierung des Apollonischen Berührproblems, da es meist nicht viele Summanden zu addieren galt. Jedoch sollte dieser Gedanke stets im Hinterkopf sein. Ungleich öfter war folgender verwandter Klassiker der Numerik relevant.

Bei den Berechnungen zu den einzelnen Fällen kommt es immer wieder zu Teilausdrücken der Form  $a^2 - b^2$ . Nun weiß man, dass dies nach der 3. Binomischen Formel dasselbe wie  $(a + b) * (a - b)$  ist. Ob das auch für Gleitkommaarithmetik gilt, wird in [9] Kapitel 2 Abschnitt 2 durch Vorwärtsfehleranalyse versucht zu klären. Dabei ergeben sich folgende Abschätzungen nach oben:

$$\frac{|(a \otimes a) \ominus (b \otimes b) - (a^2 - b^2)|}{|a^2 - b^2|} \leq u \left( 1 + (1 + u) \frac{a^2 + b^2}{|a^2 - b^2|} \right)$$

und

$$\frac{|(a \oplus b) \otimes (a \ominus b) - (a^2 - b^2)|}{|a^2 - b^2|} \leq 3u + 3u^2 + u^3 \sim 3u$$

Dabei sieht man, dass der relative Fehler bei  $a^2 - b^2$  für  $|a| \sim |b|$  beliebig schlecht werden kann, wohingegen die Abschätzung nach oben bei  $(a + b) * (a - b)$  beschränkt ist für eine kleine Konstante. Es muss jedoch noch auf die Anmerkung, die in [9] angeführt wird, hingewiesen werden, dass es sich hierbei um Abschätzungen nach oben handelt und daher kein so schlimmer Fehler angenommen werden muss. Daher bleibt es an der Realität, festzustellen, ob die Berechnung mit  $a^2 - b^2$  wirklich signifikant schlechter ist. Nach [9] ist dem jedoch so.

Daher wurde in der Implementierung stets die 3. Binomische Formel angewandt, wenn es Terme wie  $a^2 - b^2$  zu berechnen galt.

### 3.7 Lösen einer quadratischen Gleichung

Ist eine quadratische Gleichung der Form  $ax^2 + bx + c = 0$  gegeben, so hat man in der Schule gelernt, dass  $x_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  die Lösungen liefert, vorausgesetzt dass  $a \neq 0$  und  $b^2 - 4ac > 0$  gilt. Das Lösen einer quadratischen Gleichung ist im Allgemeinen in jedem der obigen Fälle außer PPP nötig. Ein Beispiel ist das Schneiden einer Ellipse und einer Winkelhalbierenden im PLL-Fall, was zu der Berechnung einer quadratischen

Gleichung führt. Deshalb ist es durchaus sinnvoll, sich zu überlegen, ob die in der Schule kennengelernte Methode ausreichend numerisch stabil ist oder ob es bessere Verfahren gibt.

Betrachtet man die sogenannte Mitternachtsformel unter dem bisher schon gewonnenen numerischen Grundwissen, so fällt auf, dass es zu Auslöschung kommen kann, wenn  $ac$  im Vergleich zu  $b^2$  relativ klein ist und damit  $\sqrt{b^2 - 4ac} \approx b$  gilt. Die Auslöschung kommt zu tragen, wenn  $-b + \sqrt{b^2 - 4ac}$  für  $b > 0$  berechnet wird bzw.  $-b - \sqrt{b^2 - 4ac}$  für  $b < 0$ . Um diese Auslöschung zu vermeiden, werden bei Horn in [3] die Formeln

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \quad \text{für } b > 0$$

und

$$x_1 = \frac{2c}{-b + \sqrt{b^2 - 4ac}}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{für } b < 0$$

vorgestellt. Die Korrektheit ist leicht nachzurechnen. Insbesondere vermeiden sie Auslöschung ohne dabei größeren Rechenaufwand zu erzeugen, weshalb dieser Löser für die quadratische Gleichung deutlich besser geeignet ist als die Mitternachtsformel. Nur die Implementierung bläht sich etwas auf, da nun nach  $b < 0$  und  $b > 0$  unterschieden werden muss.

Bei Kahan in [4] wird noch einen Schritt weitergegangen. Bei der Berechnung der Determinante wird quadriert und es werden zwei Zahlen multipliziert. Will man das Ergebnis der Determinante auf double-Genauigkeit haben und sind die Eingabewerte in double-Genauigkeit angegeben, so reicht es natürlich nicht, dass die Rechnung in double-Genauigkeit durchgeführt wird, da zu wenige gültige Ziffern für die Basis vorhanden sind. Anders ausgedrückt bedeutet dies, dass bei der Berechnung in double-Genauigkeit mehr als nur die letzten beiden Bits im Ergebnis fehlerbehaftet sein können. Will man erreichen, dass sich der Fehler auf das letzte Bit bzw. die letzten beiden Bits beschränkt, so ist nach einer Daumenregel die Berechnung in einer Arithmetik durchzuführen, die mindestens doppelt so viele signifikante Stellen besitzt als das gewünschte Resultat.

Nun gibt es Hardware bzw. Compiler, die eine solche ausreichende Genauigkeit<sup>12</sup> zur Verfügung stellen, jedoch bei weitem nicht alle. Man kann auch in Software mit einigen Tricks diese Genauigkeit simulieren, wie es ebenfalls in [4] dargelegt wird, was aber sehr aufwändig und teuer ist. Deshalb wurde dieser Ansatz in der Implementierung nicht umgesetzt und bei der oben vorgestellten Erweiterung der Mitternachtsformel verblieben.

---

<sup>12</sup>Zum Beispiel eine sogenannte „quadruple precision“.

## 4 Nichtlineare Optimierung mit dem Newton-Verfahren

### 4.1 Problemstellung

Mit den oben vorgestellten Algorithmen lässt sich für jede der Aufgabenstellungen eine Menge von Lösungen ermitteln. Bezieht man nun noch die Überlegungen zur numerischen Stabilität mit ein, so sind diese Lösungen im Allgemeinen robuster und exakter. Die Qualität einer solchen Lösung lässt sich mit Hilfe einer Fehlerfunktion ermitteln, die auf einen bestimmten Schwellwert geprüft wird. Für diese Fehlerfunktion gibt es mehrere Möglichkeiten. Eine besteht darin, die Abstände des Mittelpunkts eines Lösungskreises zu den Berührungspunkten mit den jeweiligen Objekten zu berechnen und davon den Radius des Lösungskreises zu subtrahieren. Dadurch erhält man drei Werte, die man zu einem Vektor zusammenfassen kann. Das angestrebte Ziel ist es, dass alle drei Komponenten dieses Vektors 0 ergeben, beziehungsweise zum Beispiel die 2-Norm dieses Vektors 0 ergibt. Die Fehlerfunktion ist also von der Form  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  und man sucht den Vektor  $(x, y, r)$  für den  $\|f(x, y, r)\|_2$  Null ergibt, wobei  $x$  und  $y$  die Koordinaten des Lösungskreismitelpunkts sind und  $r$  der Radius des Lösungskreises ist. In der Praxis ist es auf Grund von Rundungsfehlern sowohl bei der Berechnung des Lösungskreises als auch der Auswertung der Fehlerfunktion kaum möglich, die exakte Nullstelle zu finden. Deshalb versucht man, nur die Funktion  $\|f\|_2$  so weit zu minimieren, bis sie ein gewisses  $\varepsilon$  unterschritten hat. Die 2-Norm ist dabei willkürlich gewählt bzw. von der Anwendung abhängig. Ebenso sinnvoll wäre etwa die  $\infty$ -Norm, wenn man erreichen will, dass der Maximalfehler unterhalb der  $\varepsilon$ -Schranke sein soll. Dies ist ein Paradebeispiel für eine Aufgabenstellung im Bereich der Optimierung. Da die Abstandsberechnung, d.h. die Euklidische Norm, eine nichtlineare Funktion ist, betrachten wir im Folgenden Methoden der nichtlinearen Optimierung, um das eventuell fehlerbehaftete Ergebnis aus den oben beschriebenen Berechnungen im Anschluss etwas zu verbessern.

Sei nun der Vektor  $(x, y, r)$  die berechnete Lösung eines Apollonischen Berührproblems. Für die Fehlerfunktion wird, wie oben erwähnt, der Abstand der Lösung zu den jeweiligen gegebenen Objekten berechnet und mit dem Radius der Lösung verglichen. Berührt die Lösung einen Kreis  $k_i$  mit Mittelpunkt  $(x_i, y_i)$  und Radius  $r_i$ , dann bildet die Funktion  $f_i : \mathbb{R}^3 \rightarrow \mathbb{R} : (x, y, r) \rightarrow (x - x_i)^2 + (y - y_i)^2 - (r \pm r_i)^2$  eine Komponente der Fehlerfunktion. Das  $\pm$  ist so zu sehen, dass es darauf ankommt, ob  $k_i$  von außen oder einschließend berührt wird. Ist das berührte Objekt ein Punkt, so wird einfach  $r_i$  auf 0 gesetzt. Damit ergibt sich beispielsweise für den CCP-Fall mit den Kreisen  $k_i$  und  $i \in \{1, 2, 3\}$ , wobei  $k_3$  den Punkt repräsentiert und somit  $r_3 = 0$  gilt, folgende Fehlerfunktion:

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 : (x, y, r) \rightarrow \begin{pmatrix} f_1(x, y, r) \\ f_2(x, y, r) \\ f_3(x, y, r) \end{pmatrix} = \begin{pmatrix} (x - x_1)^2 + (y - y_1)^2 - (r \pm r_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 - (r \pm r_2)^2 \\ (x - x_3)^2 + (y - y_3)^2 - r^2 \end{pmatrix}$$

Eine Ideallösung  $(x, y, r)$  hat von den beiden Kreisen und dem Punkt den exakt gleichen Abstand, d.h. es gilt  $f(x, y, r) = 0$ . Davon gibt es im Allgemeinen mehrere. Der CCP-Fall hat davon 4, wenn kein Spezialfall vorliegt. Mit Hilfe nichtlinearer Optimie-

rungsverfahren wird nun versucht, eine fehlerbehaftete Lösung  $(x^{(0)}, y^{(0)}, r^{(0)})$  seiner Ideallösung so anzunähern, sodass  $\|f(x^{(k)}, y^{(k)}, r^{(k)})\|_2$  kleiner einem vorgegebenen  $\varepsilon$  ist, wobei  $(x^{(k)}, y^{(k)}, r^{(k)})$  die Annäherung ist. Dazu gibt es verschiedene Möglichkeiten, von denen einige von Alt in [1] vorgestellt werden. Darin wird in Verfahren unterschieden, die ohne die Ableitung auskommen, so wie das Verfahren von Nelder und Mead und welche, die die Ableitung entweder in expliziter Form oder angenähert benötigen. Ein sehr bekanntes Verfahren ist das Newton-Verfahren, das im nächsten Abschnitt kurz beschrieben wird.

## 4.2 Das Newton-Verfahren

Im eindimensionalen Fall hat das Newton-Verfahren eine sehr anschauliche geometrische Interpretation, wie man in Abb. 13 sehen kann. Es bedient sich nämlich der Ableitung, um sich der Nullstelle einer stetig differenzierbaren Funktion  $f$  iterativ zu nähern. Für einen Startwert  $x^{(0)} \in \mathbb{R}$  lautet die Iterationsvorschrift  $x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$ ,  $k \in \mathbb{N}_0$

Für eine einfache Nullstelle  $x \in \mathbb{R}$  einer zweimal stetig differenzierbaren Funktion  $f$  lässt

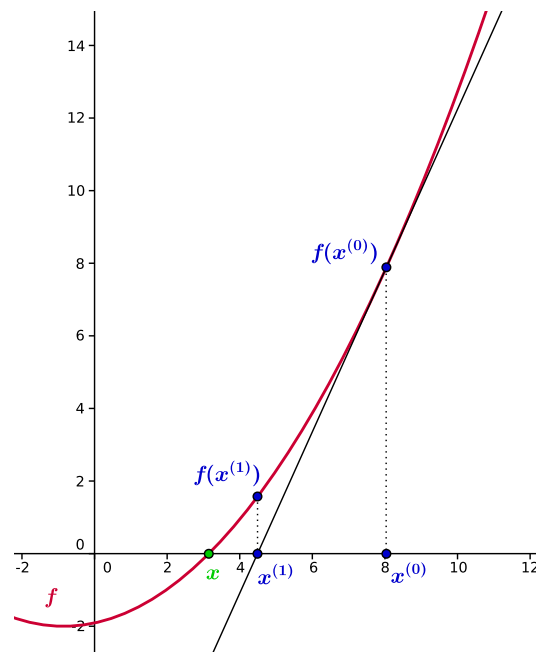


Abb. 13: Erster Iterationsschritt eines eindimensionalen Newton-Verfahrens

sich nach [9] Kapitel 10.4 zeigen, dass das Newton-Verfahren lokal quadratisch konvergent gegen  $x$  ist. Lokal quadratisch konvergent heißt, dass man für den Startwert  $x^{(0)}$  nahe genug an der Nullstelle  $x$  sein muss, damit das Verfahren garantiert konvergiert, jedoch sichert es dann auch zu, dass es sehr schnell konvergiert. Diese Aussagen gelten nach [9] nicht nur für den eindimensionalen Fall, sondern lassen sich auf  $n$  Dimensionen erweitern, womit es für unsere Anwendung interessant wird.

Es ist eine Abbildung  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  gegeben, die eine  $n$ -dimensionale, nichtlineare Gleichung darstellt und es soll eine Nullstelle von  $F$  berechnet werden. Ist  $F$  zweimal stetig differenzierbar und  $x \in \mathbb{R}^n$  eine einfache Nullstelle von  $F$ , d.h.  $F(x) = 0$  und  $\det J[F](x) \neq 0$ , dann konvergiert das Newton-Verfahren mit der Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} - \frac{F(x^{(k)})}{J[F](x^{(k)})}, \quad k \in \mathbb{N}_0, \quad x^{(0)} \in \mathbb{R}^n$$

lokal quadratisch gegen  $x$ . Dabei ist  $J[F] = \left[ \frac{\partial F_j}{\partial x_k} : j, k = 1, \dots, n \right]$  die Jacobi-Matrix der Abbildung  $F$ .

### 4.3 Anwendung des Newton-Verfahrens

Da für die oben definierte Fehlerfunktion  $f$  ebenfalls die Nullstelle berechnet, oder zumindest angenähert werden muss, wäre das Newton-Verfahren in der 3-dimensionalen Variante dafür geeignet. Der Iterationsstart  $x^{(0)}$  wäre die berechnete Lösung  $(x^{(0)}, y^{(0)}, r^{(0)})$  und die anzunähernde Nullstelle die Ideallösung  $(x, y, r)$ . Zu überprüfen ist noch, ob das Newton-Verfahren angewendet werden darf. Dazu muss  $f$  zweimal stetig differenzierbar sein. Das ist genau dann der Fall, wenn  $f_i$  zweimal stetig differenzierbar ist für alle  $i \in \{1, 2, 3\}$ . Dies wiederum ist erfüllt, wenn für alle  $i \in \{1, 2, 3\}$   $f_i$  zweimal partiell differenzierbar ist und alle zweiten partiellen Ableitungen stetig sind. Die partiellen Ableitungen existieren offensichtlich für alle  $f_i$  und sind Polynome. Daraus folgt, dass alle  $f_i$  zweimal stetig differenzierbar sind und damit auch  $f$ . Das heißt, das Newton-Verfahren darf angewandt werden und konvergiert lokal quadratisch. Die zugehörige Jacobi-Matrix zu obigem CCP-Beispiel lautet wie folgt:

$$J[f](x, y, r) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial r} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial r} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial r} \end{pmatrix} = \begin{pmatrix} 2(x - x_1) & 2(y - y_1) & -2(r \pm r_1) \\ 2(x - x_2) & 2(y - y_2) & -2(r \pm r_2) \\ 2(x - x_3) & 2(y - y_3) & -2r \end{pmatrix}$$

Der Konvergenzradius bestimmt, in welchem Bereich das Newton-Verfahren konvergiert. Außerhalb dieses Bereichs kann es divergieren oder oszillieren. Da sich dieser Konvergenzradius sehr schwer bestimmen lässt, besteht der praktische Ansatz darin, davon auszugehen, dass sich der Startwert und die Nullstelle innerhalb dieses Konvergenzbereichs befinden und das Newton-Verfahren ungeprüft anzuwenden. Da das Newton-Verfahren innerhalb des Konvergenzradius eine Kontraktion ist, muss das Ergebnis der Iteration auf jeden Fall mindestens so nah am Zielwert liegen, wie der Startwert, bzw. der Fehler darf in keinem Iterationsschritt im Vergleich zum Vorherigen größer werden. Testet man darauf und stellt fest, dass der Fehler größer geworden ist, so war der Startwert wohl nicht nahe genug an der Nullstelle und man muss das Ergebnis verwerfen.

Die Iterationsvorschrift besagt, dass durch die Jacobi-Matrix geteilt wird, also das Inverse

dieser Matrix berechnet werden muss. Dies umgeht man aus numerischen Gründen gerne, weshalb die Vorschrift wie folgt umgeformt wird:

- $J[f](z^{(k)}) * \Delta z = f(z^{(k)})$  mit  $z^{(k)} = (x^{(k)}, y^{(k)}, r^{(k)})$
- $z^{(k+1)} = z^{(k)} - \Delta z$

Bei dem  $(k+1)$ . Iterationsschritt mit  $k \in \mathbb{N}_0$  wird also zunächst die Fehlerfunktion  $f$  an der Stelle  $z^{(k)}$  ausgewertet und die dazugehörige Jacobi-Matrix berechnet. Anschließend wird das lineare Gleichungssystem  $J[f](z^{(k)}) * \Delta z = f(z^{(k)})$  gelöst und Iterationsvariable mit  $z^{(k+1)} = z^{(k)} - \Delta z$  aktualisiert. Mehr ist beim Newton-Verfahren nicht zu tun.

In der Theorie bildet meist ein  $\varepsilon$ -Kriterium die Abbruchbedingung für die Iteration. In vorliegendem Fall wäre etwa  $\|f(x^{(k)}, y^{(k)}, r^{(k)})\|_2 < \varepsilon$  eine solche Bedingung. In der Praxis kann es zu Divergenz, Oszillation oder auch numerischen Schwierigkeiten kommen, die dafür sorgen können, dass dieses  $\varepsilon$ -Kriterium nie erfüllt wird und damit eine Endlosschleife entstehen würde. Deshalb sind weitere Abbruchbedingungen notwendig. Sinnvoll ist eine eventuell experimentell bestimmte Obergrenze an Iterationsschritten, da das Newton-Verfahren, wenn es konvergiert, dies sehr schnell macht. Dadurch kann angenommen werden, dass bei Erreichen dieser Obergrenze einer der erwähnten Problemfälle eingetreten ist und die Iteration gestoppt werden kann. Ein weiteres Kriterium betrachtet das  $\Delta z$  im Vergleich zu  $z^{(k)}$ . Wird das  $\Delta z$ , also die Veränderung, die an der Iterationsvariablen  $z^{(k)}$  in einem Iterationsschritt vorgenommen wird, relativ klein, so kann auf Grund mangelnder gültiger Stellen in der Mantisse der Effekt eintreten, dass diese Änderung einfach wegfällt. Dies führt dazu, dass das Newton-Verfahren stagniert und sich auch der Fehler nicht mehr ändert. Würde man dies nicht abprüfen, iteriert das Verfahren unendlich lange oder eben bis zur Obergrenze an Iterationsschritten.

Eingangs wurde die Fehlerfunktion  $f$  nur für Fälle, die ausschließlich Kreise oder Punkte enthalten, aufgestellt. Die Situationen, in denen eine oder mehrere Geraden beteiligt sind, sollen aber auch nicht vernachlässigt werden. Den Abstand zu einer Geraden zu berechnen, ist etwas aufwändiger als zu einem Kreis oder einem Punkt. Dazu wird nämlich zunächst der Lösungskreismitelpunkt orthogonal auf die Gerade projiziert und anschließend der Abstand zu diesem Punkt berechnet. Sei  $g$  die gegebene Gerade mit normiertem Normalenvektor  $n$  und dem Hesse-Parameter  $\delta$ . Dann beschreibt die Funktion

$$h : \mathbb{R}^2 \rightarrow \mathbb{R}^2 : p \rightarrow \begin{pmatrix} n.x * \delta + (p \times n) * n.y \\ n.y * \delta - (p \times n) * n.x \end{pmatrix}$$

die Projektion eines Punktes  $p$  auf die Gerade  $g$ . Dazu wurde  $g = \{x \in \mathbb{R}^2 | \langle x, n \rangle = \delta\}$  mit der Menge  $\{p + \lambda * n | \lambda \in \mathbb{R}\}$  der Punkte, die sich auf der Senkrechten zu  $g$  durch  $p$  befinden, geschnitten. Der Fehler einer Lösung  $(x, y, r)$  bezüglich der Geraden  $g$  bestimmt sich durch

$$f_g : \mathbb{R}^3 \rightarrow \mathbb{R} : (x, y, r) \rightarrow (x - x_h)^2 + (y - y_h)^2 - r^2 \text{ mit } (x_h, y_h) = h(x, y).$$

Die dazugehörigen partiellen Ableitungen lauten wie folgt:

$$\frac{\partial f_g}{\partial x} = 2(x - x_h)(1 - n.y^2) + 2(y - y_h) * n.x * n.y$$

$$\frac{\partial f_g}{\partial y} = 2(x - x_h) * n.x * n.y + 2(y - y_h)(1 - n.x^2)$$

$$\frac{\partial f_g}{\partial r} = -2r$$

Damit können nun die Fehlerfunktion und die Jacobi-Matrix für Fälle, die Geraden enthalten, aufgestellt werden und das Newton-Verfahren angewandt werden.

#### 4.4 Gründe für die Verwendung des Newton-Verfahrens

Die Argumente für die Verwendung des Newton-Verfahrens sind eigentlich schon alle gefunden. Wenn man die Ableitung so einfach berechnen kann, wie in vorliegendem Fall, so ist es ratsam das Newton-Verfahren selbst zu verwenden, anstatt einem Verfahren, das ohne die direkte Ableitung zu berechnen funktioniert, diese aber annähern muss. Außerdem bekommt man sehr einfach Informationen über das Verhalten der Fehlerfunktion, die man zum Beispiel bei ableitungsfreien Verfahren ungenutzt lassen würde. Darüber hinaus ist das Newton-Verfahren nicht schwer zu implementieren und mit quadratischer Konvergenz sehr schnell.

#### 4.5 Gegenüberstellung der Ergebnisse mit und ohne Optimierung an Hand eines Beispiels

Die Implementierung des Newton-Verfahrens zur nichtlinearen Optimierung war dennoch mit nicht unerheblichem Aufwand verbunden. Deshalb soll nun an Hand eines Beispiels falls in C++ Code das Ergebnis vor und nach Anwendung der Optimierung verglichen werden, um zu prüfen, welchen Nutzen die Optimierung hat.

Dazu wurde der CCP-Fall untersucht. Die Ausgangslage wurde so gewählt, dass bis auf einen kleinen Rundungsfehler bei der Konstruktion der Ausgangslage eine der vier Lösungen der Einheitskreis sein sollte, der mit „scale“ multipliziert wurde. FPoint2 ist dabei nur die Implementierung eines zweidimensionalen Punktes. Die auskommentierten Zahlen geben die Werte der berechneten Variablen an.

```
// Skalierung um 1000
double scale = 1000.0;

// Initialisierung des 1. Kreises
FPoint2 center1 = scale * FPoint2(2.0 * sin(0.34300000000000024913), 2.0
    * cos(0.34300000000000024913)); // (672.6277024084329,
    1883.499926719604)
double r1 = scale * 1;

// Initialisierung des 2. Kreises
FPoint2 center2 = scale * FPoint2(2.0 * sin(4.0999999999997038813), 2.0 *
    cos(4.0999999999997038813)); // (-1636.554222128481,
    -1149.647893067022)
double r2 = scale * 1;
```



```
// Initialisierung des Punktes
FPoint2 pt = scale * FPoint2(0, 1); // (0, 1000)
```

Dann wurden die Berührungskreise nach dem in CCP vorgestelltem Algorithmus unter Berücksichtigung der Aspekte aus dem Abschnitt zur numerischen Stabilität berechnet. Zur jeweiligen Lösung wurde der Abstand ihres Mittelpunktes zu den gegebenen Objekten berechnet.

```
// Lösungen ohne Optimierung
circle 0 center: 0.0000000100681573 -0.0000000076645355 radius:
    999.999999999997726
// Distanzen der Lösung zu den einzelnen Objekten - sollten bei optimaler
// Lösung bei allen drei gleich sein
P 1000.0000000076645392 C1 1000.0000000038321559 C2 1000.0000000038326107

circle 1 center: -963.9408694691688879 733.8629583365278677 radius:
    1000.0054623760805725
P 1000.0054623742622653 C1 1000.0054623751709642 C2 1000.0054623751721010

circle 2 center: -260.3539503334410483 -1381.3850902476592637 radius:
    2395.5749054311027066
P 2395.5749054262696518 C1 2395.5749054276930110 C2 395.5749054228074328

circle 3 center: -2226.2372241081939137 115.2543805182641563 radius:
    2395.6015923260229101
P 2395.6015923348099932 C1 2395.6015923322220260 C2 395.6015923411068798
```

Nimmt man den ersten Lösungskreis, der den um 1000 skalierten Einheitskreis darstellen sollte, so sieht man, dass dieser ab der 8. Stelle nach dem Komma in der x-Koordinate des Mittelpunkts von der theoretischen Ideallösung 0 abweicht. Betrachtet man die Abstände des Lösungskreismittelpunkts zu den beiden Kreisen und dem Punkt, so stellt man fest, dass die Lösung auf 8 Stellen nach dem Komma beziehungsweise 12 gültige Stellen exakt ist. Das ist kein katastrophales Ergebnis, aber es ist schon mit einem deutlichen Fehler behaftet. Das liegt auch daran, dass das Beispiel die numerische Stabilität herausfordernd gewählt wurde. Die Skalierung mit 1000 sorgt schlichtweg dafür, dass es durch große Werte und mehreren gültigen Stellen vor dem Komma schwieriger ist, die gleiche absolute Genauigkeit einzuhalten, als bei betragsmäßig kleineren Zahlen. Zudem sind der Punkt und die beiden Kreismittelpunkte beinahe kollinear, was zu teils sehr großen Zwischenergebnissen bei der Berechnung nach der oben vorgestellten allgemeinen Kreisgleichung führt. Dadurch kommt es im Vergleich mit kleineren Zwischenergebnissen zu Genauigkeitsverlusten. Diese auf den ersten Blick etwas überraschende numerische Schwachstelle wurde empirisch gefunden, indem die Ergebnisse wieder besser wurden, nachdem die Komponenten deutlicher linear unabhängig gestaltet wurden.

Im Anschluss wurde die nichtlineare Optimierung nach dem Newton-Verfahren auf das berechnete Ergebnis angewandt.

```
// Ausgaben während der Optimierung
// Optimierung der 1. Lösung
// Fehlervektor vor Beginn der Optimierung
```

```

initial_error_vec 0.0000153291202878 0.0000153320586378
0.0000153295257519
// 2-Norm des Fehlervektors vor Beginn der Optimierung
initial_error 2.6552745821257244e-05

// erster Iterationsschritt und delta z
i 1 delta 0,0000000100680555 -0,0000000076644806 -0,0000000000002823
// Lösung nach einem Iterationsschritt
sol 0,0000000000001017 -0,0000000000000549 1000,0000000000000000
// Fehlervektor und 2-Norm davon nach einem Iterationsschritt
error_vec 0,0000000002196430 0,0000000002194156 0,0000000001097078
error 3.2927502126768967e-10

i 2 delta -0,0000000000000002 0,0000000000000001 -0,0000000000000549
sol 0,0000000000001019 -0,0000000000000550 1000,0000000000000000
error_vec 0,0000000002196430 0,0000000002196430 0,0000000001099352
error 3.2950236876310593e-10

i 3 delta 0,0000000000000001 -0,0000000000000001 -0,0000000000000549
sol 0,0000000000001018 -0,0000000000000548 1000,0000000000000000
error_vec 0,0000000002196430 0,0000000002194156 0,0000000001097078
error 3.2927502126768967e-10

i 4 delta -0,0000000000000002 0,0000000000000001 -0,0000000000000549
sol 0,0000000000001019 -0,0000000000000549 1000,0000000000000000
error_vec 0,0000000002196430 0,0000000002196430 0,0000000001098215
error 3.2946445571724325e-10

i 5 delta 0,0000000000000000 -0,0000000000000000 -0,0000000000000549
sol 0,0000000000001019 -0,0000000000000549 1000,0000000000000000
error_vec 0,0000000002196430 0,0000000002196430 0,0000000001098215
error 3.2946445571724325e-10

// Optimierung der 2. Lösung
initial_error_vec -0,0000036372669001 -0,0000036336066387
-0,0000036367036955
initial_error 6.2974932692603325e-06

i 1 delta 0,0000000023885362 -0,0000000018178258 -0,0000000000002650
sol -963,9408694715574484 733,8629583383457202 1000,0054623760807999
error_vec 0,0000000001687113 0,0000000002187335 0,0000000001078888
error 2.9655997741652492e-10

i 2 delta -0,0000000000000242 0,0000000000000267 -0,0000000000000377
sol -963,9408694715574484 733,8629583383457202 1000,0054623760807999
error_vec 0,0000000001687113 0,0000000002187335 0,0000000001078888
error 2.9655997741652492e-10

// Optimierung der 3. Lösung
initial_error 4.0107068373043258e-05

i 1 delta 1.3132997820352792e-08 error 5.8121732839811137e-10
i 2 delta 1.2513993382800079e-13 error 5.8121732839811137e-10

```

```
// Optimierung der 4. Lösung
initial_error 7.2921367038379685e-05

i 1 delta 2.388100356582607e-08 error 8.333116162455102e-10
i 2 delta 1.2084862768990689e-13 error 8.333116162455102e-10
```

In obiger Ausgabe wurden Zwischenergebnisse beim Optimierungsprozess festgehalten. Man sieht, dass in allen vier Fällen der Fehler verringert wird, jedoch auf verschiedene Weisen. Betrachtet man die erste Lösung, so sieht man, dass der erste Iterationsschritt sogleich eine Verbesserung des Fehlers von etwa  $2 * 10^{-5}$  auf  $3 * 10^{-10}$  bewirkt. Dann jedoch stagniert das Verfahren etwas und es treten oszillierende Effekte ein, bis sich der Fehler vom 4. auf den 5. Schritt nicht mehr ändert und dadurch die Iteration abbricht. Bei der 2. Lösung tritt auch im ersten Schritt eine ähnliche Verbesserung ein, dann jedoch stagniert der Fehler sofort. Die 3. und 4. Lösung zeigen analoges Verhalten zu dem der 2. Lösung und sind deshalb verkürzt mit der 1-Norm von  $\Delta z$  angegeben. Nun versuchen wir uns diese Beobachtungen zu erklären.

Dass der Fehler stagniert liegt daran, dass die Änderung  $\Delta z$ , wie sie oben genannt wurde, im Vergleich zu den Werten der Lösung zu klein wird und bei der Addition wegfällt. Dies wurde im Abschnitt zur Anwendung des Newton-Verfahrens zu dem Thema Abbruchkriterien theoretisch beschrieben und ist hier in der Ausführung zu beobachten. Für zu optimierende Werte von etwa  $10^3$  bis  $10^4$ , wie im Beispiel, tritt dieser Effekt ein, wenn das  $\Delta z$  in den Bereich von  $10^{-13}$  oder kleiner wandert. Das ist der Grund für das Abbruchkriterium, das die Fehler aus zwei aufeinanderfolgenden Iterationen betrachtet. Ohne diesem Kriterium würde die Schleife unendlich bzw. bis zur maximalen Iterationszahl laufen. Der oszillationsähnliche Effekt ist dadurch zu erklären, dass die Werte der ersten Lösung in verschiedenen Größenordnungen sind. Die x- und y-Koordinate sind zu Beginn im Bereich von  $10^{-8}$ , wohingegen der Radius etwa bei 1000 liegt. Dies hat zur Folge, dass die Änderungen, die das  $\Delta z$  vornimmt, in der x- und y-Koordinate noch signifikant sind, beim Radius jedoch schon ignoriert werden. Dass die Iteration irgendwann an den Punkt kommt, an dem es auf Grund von mangelnder Änderung des Fehlers abbricht, muss nicht sein und wurde in einigen Beispielen auch schon anders erfahren. Das heißt, es reicht auch nicht nur, die Fehler zweier aufeinanderfolgender Schritte zu betrachten, man muss auch das  $\Delta z$  im Verhältnis zur Größe der Werte im Auge haben. Bei den anderen Lösungen sind die Werte von vergleichbarer Größenordnung und das  $\Delta z$  springt gleichsam in einen Bereich, in dem es bei der Addition wegfällt. Das bedeutet jedoch leider nicht, dass für vergleichbare Werte der Oszillationseffekt nicht auch auftreten könnte.

Trotzdem sieht man, dass das Newton-Verfahren theoretisch konvergiert, die ursprünglichen Lösungen also alle innerhalb des Konvergenzbereiches sind.

Die optimierten Lösungen wurden ebenfalls mit den Abständen ausgegeben.

```
// Lösungen mit Optimierung
circle 0 center: 0,0000000000001019 -0,0000000000000549 radius:
    1000,0000000000000000
P 1000,0000000000000000 C1 1000,0000000000000000 C2 1000,0000000000000000
```

```

circle 1 center: -963,9408694715574484 733,8629583383457202 radius:
    1000,0054623760807999
P 1000,0054623760807999 C1 1000,0054623760809136 C2 1000,0054623760809136

circle 2 center: -260,3539503259851813 -1381,3850902533363296 radius:
    2395,5749054311027066
P 2395,5749054311027066 C1 2395,5749054311031614 C2 395,5749054311027066

circle 3 center: -2226,2372240946356214 115,2543805079417751 radius:
    2395,6015923260224554
P 2395,6015923260224554 C1 2395,6015923260224554 C2 395,6015923260224554

```

Zieht man erneut exemplarisch den ersten Lösungskreis heran, so erkennt man, dass dieser nun deutlich näher an  $(0, 0, 1000)$  liegt als die ursprüngliche Lösung. Des Weiteren unterscheiden sich die Abstände zu den anderen Objekten erst an der 13. Stelle nach dem Komma bzw. der 16. gültigen Stelle. Dies ist ein sehr gutes Ergebnis, da eine solche Abweichung als numerisch Null angesehen werden kann.

Die nichtlineare Optimierung hat also den Fehler noch einmal deutlich verringert und das mit einem sehr geringen Aufwand, da eine Iteration einfach zu berechnen ist und auch nur wenige Iterationen nötig waren. Dies war natürlich nur ein Beispiel und ist nicht ausreichend, um eine fundierte Aussage über den Nutzen der Nachoptimierung zu treffen. Jedoch wurden stichprobenartig für alle Fälle numerisch kritische Stellen untersucht und das Newton-Verfahren angewendet. Da die Optimierung keine Lösung liefert, die einen größeren Fehler hat, als die Initiallösung, führt die Nachiteration schlimmstenfalls dazu, dass kein signifikant besseres Ergebnis erzielt wurde, aber Laufzeitressourcen investiert wurden. Je nach Anwendung ist also zu entscheiden, ob die nichtlineare Optimierung notwendig ist, zumal festgestellt wurde, dass die Optimierung vor allem in den Standardfällen nötig ist, da in den Spezialfällen die vereinfachte Situation zu exakteren Ergebnissen führt. Kritisch sind vor allem die Übergänge in Spezialfälle, die aber gerade noch Standardfälle sind. Dort kann die Optimierung entscheidende Verbesserungen herbeiführen, wie es im Beispiel gesehen wurde.

#### 4.6 Schwachstellen des Newton-Verfahrens

In obigem Beispiel hat das Newton-Verfahren eine deutliche Verbesserung bewirkt. Das ist leider nicht immer der Fall. Erwähnt wurde schon, dass das Newton-Verfahren an seine Grenzen stößt, wenn das  $\Delta z$  im Vergleich zu  $z^{(k)}$  sehr klein wird. Ein weiterer Punkt ist die Tatsache, dass die Konvergenz des Verfahrens nicht gesichert ist. Die quadratische Geschwindigkeit der Iteration wird mit Lokalität gekauft. Ist also eine Lösung ziemlich schlecht und weit von der Nullstelle entfernt, so kann es sein, dass das Newton-Verfahren divergiert, wobei gerade in diesem Fall eine Verbesserung nötig wäre.

Wie es bei den Voraussetzungen zur Anwendung des Newton-Verfahrens heißt, muss das Ziel der Optimierung eine einfache Nullstelle sein. Bei einer doppelten Nullstelle ist das Verfahren nicht definiert und das Verhalten unbestimmt. Die Fehlerfunktion ist so gewählt, dass jede Nullstelle eine Lösung des Apollonischen Berührproblems darstellt. Tritt kein Spezialfall ein, d.h. gibt es für die zu berechnende Konstellation so viele Lösungen,

wie es im Allgemeinen für den dazugehörigen Fall gibt, so ist jede Nullstelle einfach und die Anwendung des Newton-Verfahrens scheitert zumindest nicht an diesem Punkt. Ist jedoch eine spezielle Konstellation gegeben, wenn z.B. im PLL-Fall für parallele Geraden der Punkte auf einer Geraden liegt, dann fallen zwei Lösungen zusammen und die Anzahl der Lösungen entspricht nicht der für diesen Fall angegebenen Maximalzahl. So hat auch die Fehlerfunktion doppelte Nullstellen. Für diese Lösung ist das Newton-Verfahren somit nicht anwendbar, da das Inverse der Jacobi-Matrix nicht existiert, bzw. das lineare Gleichungssystem nicht eindeutig lösbar ist. Jedoch war dieses Problem in der Praxis bisher tragbar, da die Spezialfälle für vereinfachte Situationen stehen und somit die Lösungen auf Grund einer nicht so aufwändigen Berechnung relativ exakt sind.

## 5 Fazit

Am Ende dieser Arbeit soll nun noch einmal kurz zusammengefasst werden, was alles behandelt und erreicht wurde. Am Anfang stand die Aufgabe, Berührungskreise für das Zusammensetzen eines Kreisbogensplines nach dem SMAP-Algorithmus zu berechnen. Für die Lösung dieser Aufgabenstellung wurde die Ideengeschichte zum Apollonischen Berührproblem betrachtet und es wurden dem Vieta folgend für entartete Kreise gesonderte Spezialfälle beschrieben, da sich dadurch die Implementierung speziell anpassen ließ. Auf diesem Wege konnten Schwierigkeiten besser aufgedeckt werden. So wurde Fall für Fall ein Algorithmus zur Berechnung der Lösungskreise vorgestellt und auch gezeigt, dass dadurch alle validen Lösungen bestimmt wurden.

Die strukturelle Unterscheidung in Spezialfälle war dem Gedanken der numerischen Stabilität geschuldet. Aber auch bei der Implementierung der einzelnen Konstellationen waren numerische Überlegungen allgegenwärtig und zeigten sich im Code etwa in einer auf den ersten Blick ungewöhnlichen Art des Auflöserns einer quadratischen Gleichung oder der permanenten Verwendung von  $\varepsilon$ -Umgebungen. Dies führte dazu, dass in vielen Fällen das Ergebnis schon ausreichend exakt war.

Um weitere Verbesserungen zu erzielen, wandte man sich dem Gebiet der nichtlinearen Optimierung zu. Dabei wurde ausgehend von einer berechneten Lösung mit Hilfe einer Fehlerfunktion nach dem Newton-Verfahren iterativ versucht, die Qualität der Lösung zu verbessern. Das heißt, der Fehler, mit dem ein Berührungskreis behaftet ist, soll nach der Optimierung kleiner sein als zuvor. An einem Beispiel konnte man sehen, wie durch die beschriebene Iteration eine ungenaue Lösung ziemlich exakt gestaltet wurde. Es wurden aber auch Grenzen aufgezeigt. Wenn etwa die Zahlen zu groß werden, kann der absolute Fehler nicht mehr eingehalten werden.

Die Fahnenstange der Möglichkeiten, eine numerisch besonders stabile und genaue Implementierung des Apollonischen Berührproblems zu finden, wurde mit den genannten Methoden natürlich bei weitem nicht ausgereizt. Dazu könnte man noch explizit die historisch entwickelten Lösungsansätze untersuchen und prüfen, ob diese nicht auf Grund ihrer Umsetzung in Software stabiler wären, als der gewählte Algorithmus. Des Weiteren gäbe es noch mehr numerische „Tricks“, die man anwenden könnte, um die Robustheit zu erhöhen und Effekte wie Auslöschung zu vermeiden. Zu guter Letzt wären für die Fälle andere Methoden der nichtlinearen Optimierung denkbar, in denen das Newton-Verfahren nicht angewandt werden kann, oder keine Verringerung des Fehlers bewirkt, obwohl es wünschenswert wäre. Unterschiedliche Verfahren mit unabhängigen Ansätzen könnten etwa für diese Situationen die nötige Genauigkeit besorgen.

Man sieht also, dass über diese Arbeit hinaus noch viele Punkte betrachtet werden könnten, nur ist die Ausführungszeit ebenfalls ein wichtiger Faktor. Die Methoden zur Berührungskreisberechnung werden im Zuge des SMAP-Algorithmus sehr häufig aufgerufen. Da die Berechnung der Lösungen kurz ist, kann auch eine sehr effiziente Newton-Optimierung, die nicht mehr als zwei oder drei Iterationen durchläuft, die Ausführungszeit signifikant erhöhen. Deshalb muss man abwägen, ob die Optimierung nötig ist und gegebenenfalls in welchem Rahmen. Nicht nur die numerische Exaktheit ist das alleinige Ziel, auch teils

konkurrierende Faktoren spielen eine große Rolle, inwiefern die Implementierung weiterhin verbessert wird.

Da die Numerik in ihrer Anwendung ein relativ neues Gebiet der Mathematik darstellt, gibt es in diesem Bereich sehr viel Bewegung. Befeuert wird dies unter anderem durch die Industrie, die in ihren Systemen immer mehr Präzision fordert, die zunächst einmal durch die Software bereitgestellt werden muss. Dies wird dazu führen, dass viele elementargeometrische Probleme unter diesem Aspekt neu untersucht werden und dadurch vielleicht nicht nur Apollonius eine zweite Renaissance erleben wird.

## Literatur

- [1] Walter Alt. *Nichtlineare Optimierung*. Vieweg, Braunschweig [u.a.], 2002.
- [2] Heinrich Dörrie. *100 Great Problems of Elementary Mathematics*. DOVER PUBLICATIONS, INC., NEW YORK, 1965.
- [3] Berthold K.P. Horn. *Numerically Stable Method for Solving Quadratic Equations*. Artikel, Massachusetts Institute of Technology, 2005.
- [4] William Kahan. *On the Cost of Floating-Point Computation Without Extra-Precise Arithmetic*. Work in progress, University of California, Berkeley, 2004.
- [5] F. Kappel. *Lineare Algebra I & II*. Vorlesungsskript, Karl-Franzens-Universität Graz, 2005/06.
- [6] Georg Maier. *Smooth Minimum Arc Paths*. SHAKER Verlag GmbH, Deutschland, 2010.
- [7] Solomon G. Mikhlin. *Error analysis in numerical processes*. Wiley, Chichester, 1991.
- [8] Johannes Röttgen-Burtscheidt. *Das Apollonische Berührproblem*. Sammlung von Lösungen, Köln, 2007.
- [9] T. Sauer. *Einführung in die Numerische Mathematik*. Vorlesungsskript, Universität Passau, 2013.



## Abbildungsverzeichnis

1	Berührungskreise nach dem Verfahren von Gergonne . . . . .	1
2	Übersicht über die Motivation des DFG-Projekts . . . . .	2
3	Umkreis eines Dreiecks . . . . .	8
4	CPP-Fall in Normallage mit Hyperbel . . . . .	9
5	PPL-Fall in Normallage . . . . .	11
6	PLL-Fall für parallele Geraden in normierter Lage . . . . .	13
7	PLL-Fall für sich schneidende Geraden in normierter Lage . . . . .	13
8	CLL-Fall für parallele Geraden in normierter Lage . . . . .	15
9	CLL-Fall für sich schneidende Geraden . . . . .	16
10	CCL-Fall in transformierter Lage . . . . .	18
11	CPL-Fall in normierter Lage . . . . .	19
12	CCP-Fall in transformierter Lage . . . . .	21
13	Erster Iterationsschritt eines eindimensionalen Newton-Verfahrens . . . . .	33

## Eidesstattliche Erklärung

Hiermit versichere ich, Günther Reitberger, an Eides statt, dass ich diese Bachelorarbeit selbstständig und ohne Benutzung anderer als der ausgewiesenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind.

Ich erkläre weiterhin, dass ich diese Bachelorarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Passau, den 4. April 2014

---

Günther Reitberger