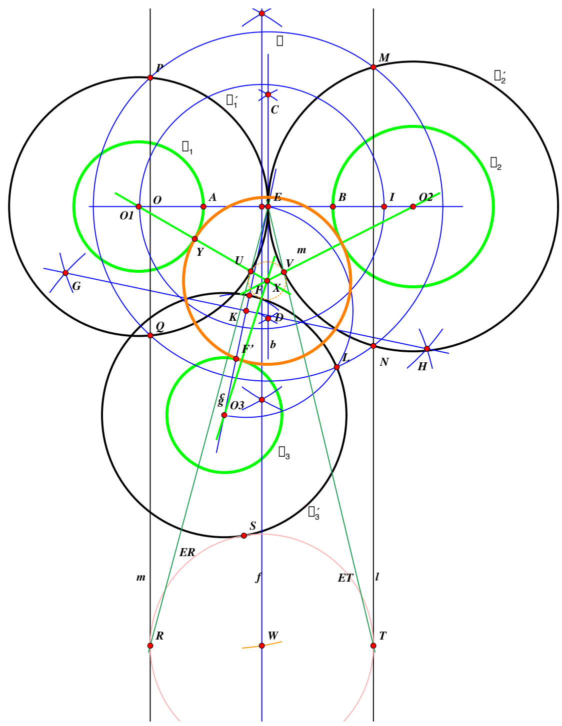

Dynamische Geometrie

von Ulrich Kortenkamp

Als Folkmar Bornemann mich fragte, ob ich einen Artikel zu den Mitteilungen in der Serie „Streifzug durch die Mathematik“ beisteuern könnte, beschrieb er das Ziel der Serie unter anderem mit dem Satz „Was bewegt einen Nachwuchswissenschaftler, nach der Promotion der Mathematik treu zu bleiben und auf eine Laufbahn als Forscher zu setzen?“ Diese Frage war der Grund für mich, spontan zuzusagen, denn die Antwort darauf interessiert mich selbst ganz besonders. Was hält mich *in der Mathematik, was ist so reizvoll an der Forschung, dass doppelte und dreifache Gehälter mich nicht locken können? Man möge mir also im Folgenden den teils sehr persönlichen Ton verzeihen, wenn ich beschreibe, welche Mathematik mich seit fünf Jahren beschäftigt. Ich möchte daran exemplarisch einige besonders bindende Momente hervorheben und einen kleinen Ausblick in die Zukunft wagen.*

Im August 1996, ich war damals Mitarbeiter an der TU Berlin in der Arbeitsgruppe von Prof. Günter Ziegler, schaffte es Jürgen Richter-Gebert, zu der Zeit ebenfalls in dieser Arbeitsgruppe, mich für sein Projekt „Cinderella“ zu gewinnen. Cinderella war und ist ein Computerprogramm für Geometrie, eine Computerversion von Zirkel und Lineal (und Papier).[1] Ich hatte von diesem Programm schon Anfang 1995 gehört, es existierte zu dem Zeitpunkt eine Version für NeXT-Rechner.[2] Da ich zu jener Zeit Fragestellungen über Punkt- und Geradenarrangements untersuchte, war ich sehr interessiert an einer solchen Software: Haben sie schon einmal versucht, mehr als 10 Geraden auf einem Blatt Papier unterzubringen, von denen sich keine drei in einem Punkt schneiden?

Dennoch hielt ich die Implementation einer solchen Software für nicht außerordentlich kompliziert und daher dachte ich nicht, dass man sich wissenschaftlich damit auseinandersetzen könnte. Als es gut ein Jahr nach dem Diplom darum ging, ein definitives Thema für meine Dissertation festzulegen, entschloss ich mich zunächst gegen Cinderella und für ein Thema aus der kombinatorischen Geometrie.[3]

Der plötzliche Sinneswandel kam durch eine mutwillig herbei geführte Notsituation zustande: Jürgen Richter-Gebert hatte zugesagt, eine Java-Version von Cinderella vorzuführen, ohne dass diese existierte – und es waren noch drei Wochen Zeit bis zu dem Vortrag. Wir konnten zu dem Zeitpunkt noch kein Java, aber wir wussten, dass wir es gemeinsam schaffen würden, und so legten wir in einer Hauruckaktion den Grundstein für viele Jahre gemeinsamer Arbeit.[4] Der Prototyp des „neuen Cinderella“ war rechtzeitig vorführbereit.

1 `http://www.cinderella.de`; siehe auch *Mitteilungen* 3–2000. [Anm. d. Red.]
2 Diese erste Version stammte von Jürgen Richter-Gebert und Henry Crapo.
3 Nachbarschaftliche Polytope.
4 Dieser Just-In-Time-Arbeitsstil ist bis heute geblieben, zum Leidwesen unserer Familien.

```
1: A=FreePoint(1,1);
2: B=FreePoint(5,3);
3: C=Circle(A,B);
4: D=Circle(B,A);
5: E=IntersectionCC(C,D);
6: F=IntersectionCC(C,D);
7: G=Join(A,B);
8: H=Join(E,F);
9: I=Meet(G,H);
```

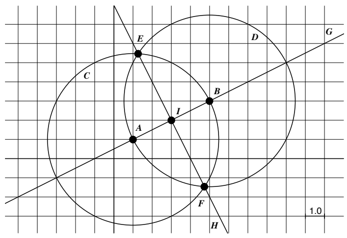

Abbildung 1: Eine Konstruktionsvorschrift und eine dazu passende Konstruktionsinstanz

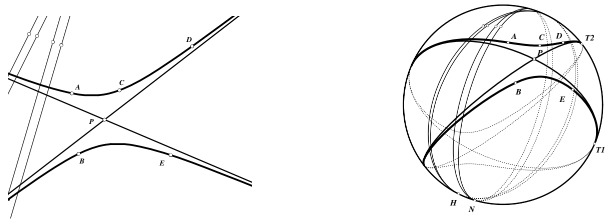

Abbildung 2: Die Tangentenkonstruktion im Unendlichen an eine Hyperbel ergibt die beiden Asymptoten. Links die Ansicht in der „normalen", euklidischen Ebene, rechts die Kugelprojektion mit Zugriff auf unendlich ferne Punkte.

Dennoch verstand ich erst Anfang 1997, warum selbst einfache Geometrieprogramme zu tatsächlich interessanten, nicht-trivialen Fragestellungen führen. Ich möchte die Problematik und ihre vorläufige Lösung hier darstellen, da es tatsächlich ein wunderschöner Streifzug durch die Mathematik zwischen Monge und Riemann ist.

Konstruktionen

In jedem Geometrieprogramm gibt es verschiedene Konstruktionselemente – Punkte, Geraden, Kreise, Kegelschnitte, Polygone, Zahlen, ... – die mit Hilfe von Konstruktionsvorschriften – Verbindungsgerade, Schnittpunkt, Lotgerade, Parallele, Winkelhalbierende, Kreis durch drei Punkte, Kegelschnitt durch fünf Punkte, ... – zusammengefügt werden. Dadurch können, ausgehend von freien Elementen, Konstruktionen durchgeführt werden, die durch die Konstruktionsvorschriften gewisse Eigenschaften haben. Werden nun im Nachhinein die Koordinaten der freien Elemente verändert, so soll der Computer die anderen, abhängigen, Elemente so nachjustieren, dass die Konstruktion immer noch den Konstruktionsvorschriften genügt (Abbildung 1).

Selbst wenn wir uns auf ganz einfache Elemente und Vorschriften beschränken, sagen wir, Punkte und Geraden, Schnittpunkte und Verbindungsgeraden, so müssen wir schon bei der Implementierung einige Spezialfälle beachten. Was geschieht mit den Schnittpunkten paralleler Geraden? Wir könnten ja in einer Konstruktion mit vier freien Punkten A, B, C, D und zwei Verbindungsgeraden $a = \overline{AB}$ und $b = \overline{CD}$ den Schnittpunkt $E = a \cap b$ einfügen, und *danach* die freien Punkte so verschieben, dass a parallel zu b liegt. Soll E dann einfach verschwinden? Was ist, wenn andere Konstruktionsteile an E hängen?

Die Lösung dieser Probleme ist einfach und allgemein bekannt; die Forderung, Spezialfälle zu eliminieren, führt sofort zur Projektiven Geometrie. Axiomatisch sind damit diese Spezialfälle wegdiskutiert, und über homogene Koordinaten ist auch die Implementierung *straightforward*.[5] Auch in der alten Cinderella-Version war dies schon so gelöst und es war auch dort schon möglich, eine Konstruktion sowohl in der euklidischen Ebene als auch auf einer Kugelprojektion zu bearbeiten, primal und polar (Abbildung 2).

Weitere Konstruktionsvorschriften hinzuzufügen birgt nicht wirklich neue Schwierigkeiten, doch spätestens mit dem Wunsch, Kreise als geometrische Objekte zu benutzen, wird es komplex. Nicht nur, weil Schnitte von Kreisen mit Geraden oder anderen Kreisen verschwinden können – hier ist der Übergang von $\mathbb{R}$ zu $\mathbb{C}$ die naheliegende Lösung. Das wirkliche Problem hier ist dynamischer Natur; nur durch die mögliche Bewegung von Anfangselementen kommt es zustande. Dieser Übergang vom „Ausrechnen" zum Bewegen kann mit dem Übergang von der *Arithmetik* zur Untersuchung von *Funktionen* verglichen werden, und wie wir bald sehen, passt dieser Vergleich perfekt.

Nehmen wir eine einfache Konstruktion – ein Kreis C um A durch B, eine Gerade a durch E und F, und den Schnittpunkt G von C mit a ... stopp! „Welcher Schnittpunkt?" werden Sie zu Recht – es gibt ja zwei, wenn a überhaupt C schneidet – fragen, aber diese Frage beantworten wir ja schon automatisch während wir konstruieren, indem wir ihn mit der Maus auswählen. Kein Problem also.

Nun bewegen wir A, also verschieben wir auch C, und beobachten den Schnittpunkt G. Nun muss der Rechner für jede neue Position von A bestimmen, wo G liegt. Aber genau hier fangen die Probleme an, denn die Entscheidung, die wir zuvor mit der Maus gefällt haben, muss jetzt der Rechner treffen, und zwar möglichst nicht „irgendwie", sondern nach sinnvollen, mathematisch begründeten Kriterien. Ein solches Kriterium wäre die Forderung nach kontinuierlichem (stetigen) Verhalten: Kleine Bewegungen der freien

5 Erstaunlicherweise gibt es dennoch nur wenige Programme für dynamische Geometrie auf der Basis projektiver Geometrie.

Punkte sollen nur kleine Bewegungen der abhängigen Punkte verursachen, eine Art ε-δ-Kriterium für „geometrische Funktionen“. Als Begründung für die Wahl dieser Stetigkeitsforderung wollen wir zunächst nur die offensichtliche Sinnhaftigkeit heranziehen – kleine Bewegungen sollen halt keine großen Sprünge hervorrufen, das wäre unintuitiv und auch unpraktisch (wir möchten ja als Vorteil geometrischer Zeichenprogramme die leichte Nachjustierbarkeit von Zeichnungen hervorheben). Später werden wir sehen, dass die Konsequenzen aus der Stetigkeit so wünschenswert sind, und die daraus resultierende Theorie so gut zusammenpasst, dass wir keine weitere Begründung mehr brauchen.[6]

Der Wunsch nach stetigem Verhalten mag einfach sein, die Erfüllung desselben ist wesentlich schwieriger. Schauen wir uns die Situation genauer an: Die beiden Schnittpunkte des Kreises mit der Geraden können durch das Lösen einer quadratischen Gleichung gefunden werden und haben die Form $P^{\pm} = P \pm \sqrt{\alpha}X$. Die Auswahl eines der Punkte entspricht also der Wahl zwischen + und − in dieser Darstellung. Was passiert, wenn wir nun einfach dieses Vorzeichen für die weiteren Rechnungen (während des Verschiebens) beibehalten?

Dieses Beibehalten eines Vorzeichens kann als Vergabe von Orientierungen für alle Punkte, Geraden und Kreise interpretiert werden, und genau auf diesem Wege kann man sich auch klar machen, dass sich die abhängigen Elemente so nicht immer stetig verhalten *können.* Stattet man die Geraden und Kreise mit einer Orientierung aus, so schneidet der Kreis die Gerade in einem Schnittpunkt P^+ von rechts nach links und im anderen Schnittpunkt P^- von links nach rechts, jeweils in Richtung der Gerade gesehen. Ebenso kann man die Schnitte zweier orientierter Kreise unterscheiden.

Betrachten wir nun zwei (orientierte) Kreise gleicher Größe, deren Mittelpunkte auf einem weiteren Kreis gleicher Größe liegen, und ihren Schnittpunkt D, der nicht gleich dem Kreismittelpunkt A ist (Abbildung 3). Bewegt man nun den Mittelpunkt des einen Kreises durch den Mittelpunkt des anderen, so „springt“ D auf den ursprünglich vermiedenen Schnittpunkt A. Die Länge des Sprunges ist der Durchmesser des Kreises, also völlig unabhängig von der Bewegung des Mittelpunktes. Dies zeigt, dass Stetigkeit nicht über Orientierungen zu erreichen ist.[7]

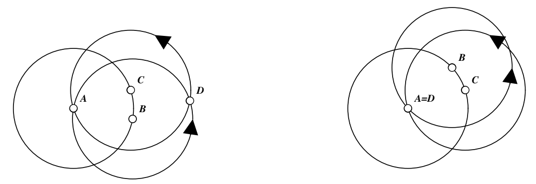

Abbildung 3: Drei Kreise mit gleichem Radius, die zeigen, dass Orientierungen allein nicht für Stetigkeit genutzt werden können. Wählt man anhand der Orientierung links den Punkt D aus, so springt er nach einer kleinen Bewegung auf A.

Mehrdeutigkeiten

Schauen wir noch genauer hin: Algebraisch sind die verschiedenen Lösungen der Gleichungssysteme nicht zu unterscheiden (höchstens semi-algebraisch, aber Orientierungen haben wir ja gerade schon als ungeeignet erkannt). Sämtliche Geometrieprogramme lösen diese Mehrdeutigkeiten dadurch auf, dass sie sich implizit für eine *Funktion* entscheiden, die für jede Position der Eingangselemente genau einen Wert für die durch *Relationen* gegebenen abhängigen Elemente auswählt.

Doch genau diese Eindeutigkeit verträgt sich nicht mit Stetigkeit: Nehmen wir die Standardkonstruktion für eine Winkelhalbierende, die sich problemlos mit Zirkel und Lineal, also mit Schnittpunkten von Kreisen und Geraden, konstruieren lässt. Wenn wir die erste der beiden Geraden ℓ_1 und ℓ_2, die den halbierten Winkel bestimmen, um den Geradenschnittpunkt P rotieren lassen, dann dreht sich die Winkelhalbierende b, sofern stetig, mit halber Winkelgeschwindigkeit. Nach einer vollen Umdrehung von ℓ_1 hat b eine halbe Umdrehung beschrieben, ist also, wenn wir Orientierungen außer Acht lassen, wieder in ihrer ursprünglichen Lage.

Iterieren wir aber die Winkelhalbierendenkonstruktion, so erhalten wir eine „Winkelviertelnde“ (oder -achtelnde, -sechzehntelnde, …), die wir mit c bezeichnen. Drehen wir nun ℓ_1 einmal um P erhalten wir eine zweite Konfiguration der Ausgangskonstruktion, in der alle freien Elemente in der gleichen Position liegen, aber die abhängigen Elemente, namentlich c, in einer anderen Lage sind. Tatsächlich können wir durch mehrfaches Iterieren für eine Konstruktion sogar exponentiell (in der Anzahl der Konstruktionsschritte) viele „passende“ Instanzen erzeugen.

6 Eine weitere, historische, Begründung möchte ich zunächst nur kurz erwähnen: Poncelet [6] ging in seinen Überlegungen ebenfalls von der Kontinuität aus (das Poncelet’sche Kontinuitätsprinzip), weil er nicht akzeptieren konnte, dass geometrische Sätze ihre Allgemeingültigkeit jemals verlieren könnten, sofern sie mehr als nur sporadisch wahr sind.

7 Der orientierungsbasierte Ansatz scheitert erst recht bei der Einbeziehung von Kegelschnitten: zwei Kegelschnitte haben acht mögliche (orientierte) Schnittpunkte, liefern aber nur zwei Bit Information durch ihre Orientierungen. Es können also nur vier Möglichkeiten unterschieden werden.

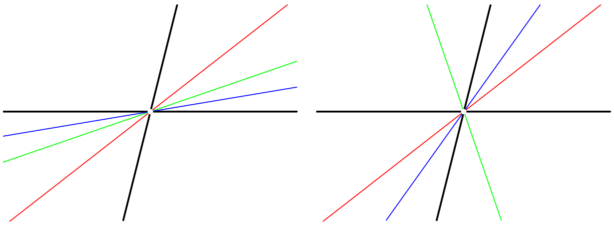

Abbildung 4: Zwei verschiedene Instanzen einer iterierten Winkelhalbierendenkonstruktion.

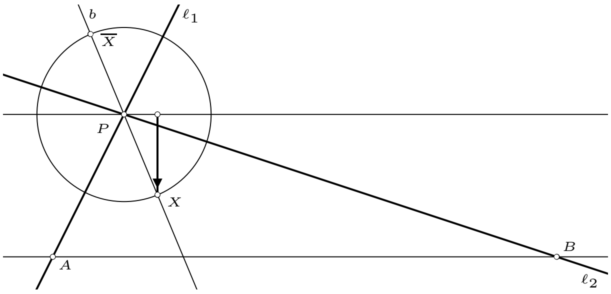

Abbildung 5: Eine Konstruktion von zwei festen Punkten A und B und einem freien Punkt $P = (x, y)$, den Verbindungsgeraden $\ell_1 = \overline{PA}$ und $\ell_2 = \overline{PB}$, einem Kreis C_0 mit festem Radius um P, einer Winkelhalbierenden b von ℓ_1 und ℓ_2 und dem Schnittpunkt X von b mit C_0.

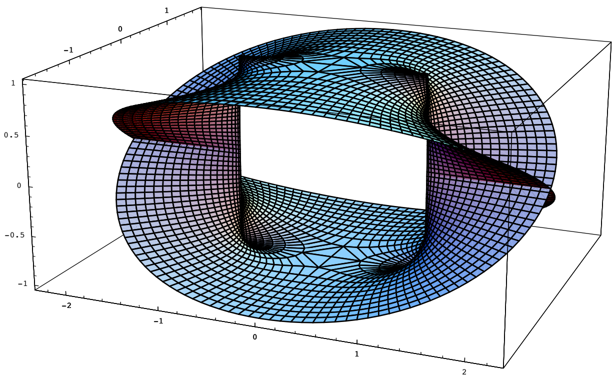

Abbildung 6: Eine von der obigen Konstruktion erzeugte Riemannsche Fläche.

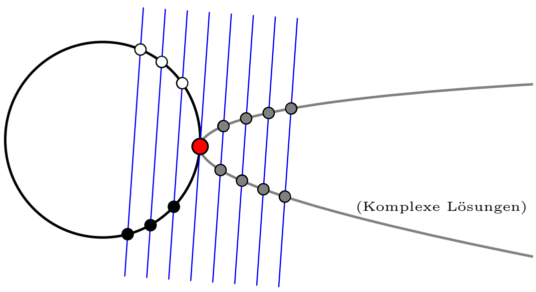

Abbildung 7: Die Schnitte einer Geraden mit einem Kreis lassen sich in der Tangentialsituation nicht unterscheiden. Daher können sie anscheinend auch nach dem Passieren dieser Situation nicht mehr unterschieden werden.

Diese Betrachtungen zeigen, dass der Wunsch nach Stetigkeit zwangsläufig die Zulassung von Mehrdeutigkeit nach sich zieht. Und ab hier können wir elegant zu dem Ansatz überleiten, der hinter *Cinderella* steckt: Der übliche Weg, mehrdeutige Relationen in Funktionen umzuwandeln, ist die Erweiterung des Definitionsbereiches. Wir werden solch eine Erweiterung nicht explizit, aber implizit durchführen.

Ich möchte dies an einem schönen Beispiel illustrieren: In Abbildung 6 sehen wir den Graph des y-Abstandes von X und P, aufgetragen nach der Position (x, y) von P. In diesem Graph kann man erkennen, dass es möglich ist, stetig von der in Abbildung 5 angegebenen Instanz zu der Instanz zu gelangen, wo der Schnittpunkt X auf dem anderen Schnittpunkt $\overline{X}$ von b mit C_0 liegt.

Dieses Beispiel (und der Hinweis auf Riemannsche Flächen) führt uns direkt zu einer Modellierung, die tatsächlich beweisbar zu kontinuierlichem Verhalten führt. Zunächst müssen wir uns von der diskreten Natur des ursprünglichen Problems lösen – wir können nicht von Kontinuität sprechen, wenn wir nur diskrete Instanzen von Konstruktionen (etwa durch eine Anzahl von Mauspositionen gegeben) betrachten. Statt einem Sprung eines Punktes P von einer Position A in die Position B betrachten wir die Bewegung von A nach B über eine lineare Interpolation $P(\lambda) = (1-\lambda)A + \lambda B$, $\lambda \in [0, 1]$, und wandeln so die Bewegung über eine Folge von Stützstellen in eine stückweise lineare Funktion um.

Wir fragen also, in welche Instanz man gelangt, wenn man eine Konstruktion von einer Startinstanz an einem Punkt $P(0) = A$ zu einer Endinstanz bei $P(1) = B$ bewegt, unter der Bedingung, dass die abhängigen Elemente sich als stetige Funktionen in λ darstellen lassen.

Ein Weg, diese Frage zu beantworten, ist es, diesen Weg tatsächlich in enger Schrittweite mit λ abzuschreiten und für jedes λ sämtliche Instanzen zu berechnen. Dann können wir – theoretisch – über eine geeignet gewählte Distanzfunktion die richtigen Instanzen miteinander verbinden und erhalten so die gesuchte Endinstanz. Das geht aber nur solange, wie die verschiedenen Instanzen weit genug voneinander entfernt sind – treffen diese aufeinander, so können wir eigentlich keine begründete Entscheidung mehr für den einen oder anderen Weg treffen.

Wir können diese Ungewissheit aber nicht akzeptieren: Ungewollte Sprünge treten genau in solchen „degenerierten“ Situationen auf, und zwar nicht bei den Elementen, deren Weg wir nicht entscheiden können, also den Schnittpunkten selbst, sondern bei von ihnen abhängigen Konstruktionsteilen. Diesen Effekt können wir bei iterierten Winkelhalbierenden beobachten, ohne dass wirklich klar wird, wo genau die falsche Entscheidung liegt, wir können ihn aber auch mit der in Abbildung 8 gezeigten Konstruktion direkt verständlich machen. Zwei zu Beginn noch unabhängige Schnittpunkte werden durch eine an beide angehängte Konstruktion miteinander gekoppelt (oder „verschränkt“, im quantenphysikalischen Sinne), und plötzlich büßen beide ihre freie Entscheidungsgewalt ein – beide müssen sich für die gleiche Richtung entscheiden.

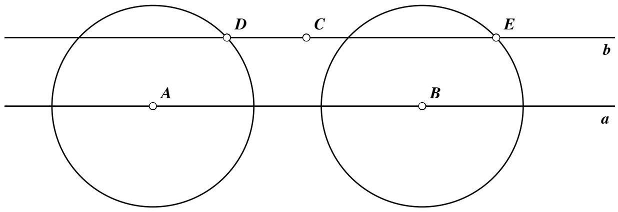

Abbildung 8: Zwei Kreise mit gleichem Radius und eine Parallele b zu a. Die Punkte D und E sind unabhängig, können aber durch eine an sie angehängte Konstruktion miteinander verschränkt werden, da sie in einer „Umgebung“ der Konstruktion, wenn C bewegt wird, stets den gleichen Abstand haben.

Die Lösung für dieses Problem der zusammenfallenden Wege ist denkbar einfach: Wir verhindern, dass die Wege jemals zusammenfallen. Die oben genannte Parametrisierung nach λ durch lineare Interpolation schrieb für λ einen Weg auf dem reellen Segment von 0 nach 1 vor. Da wir aber ohnehin mit komplexen Werten rechnen können müssen, erlauben wir uns nun einen beliebigen Weg von 0 nach 1 durch die komplexe Zahlenebene $\mathbb{C}$.

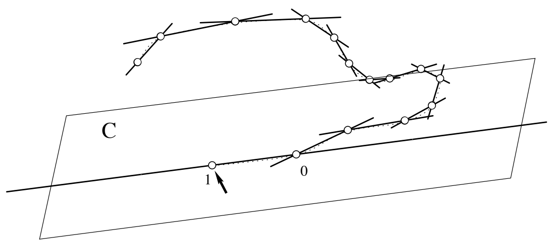

Abbildung 9: Komplexe Interpolation zwischen zwei Stützstellen.

Und diese neue Freiheit benutzen wir, um die verbotenen Tangentialsituationen (und andere Degeneriertheiten) über einen komplexen Umweg zu vermeiden. Und – und das ist wirklich erstaunlich – wir können dies immer tun, da alle diese Situationen diskret liegen, wenn es wenigstens eine Stelle gibt, wo es überhaupt etwas zu entscheiden gibt, also eine nichtdegenerierte Situation.

Warum dies? Die Antwort finden wir in der Funktionentheorie: Wir haben es bei den geometrischen Grundoperationen, die wir betrachten, mit Lösungen von polynomialen Gleichungssystemen zu tun, also *algebraischen* Funktionen. Ihre vollständigen analytischen Fortsetzungen sind holomorphe Funktionen auf den dazugehörigen Riemannschen Flächen, und für diese gilt (unter anderem) der Hauptsatz über holomorphe Funktionen: Diese sind dann und nur dann identisch, wenn sie auf einer Menge mit Häufungspunkt übereinstimmen. Übersetzt in unsere geometrische Situation heißt das zum Beispiel, dass eine Gerade in einer Konstruktion entweder immer tangential an einen Kreis ist oder nur sporadisch (bei linearer Bewegung eines der freien Punkte).[8]

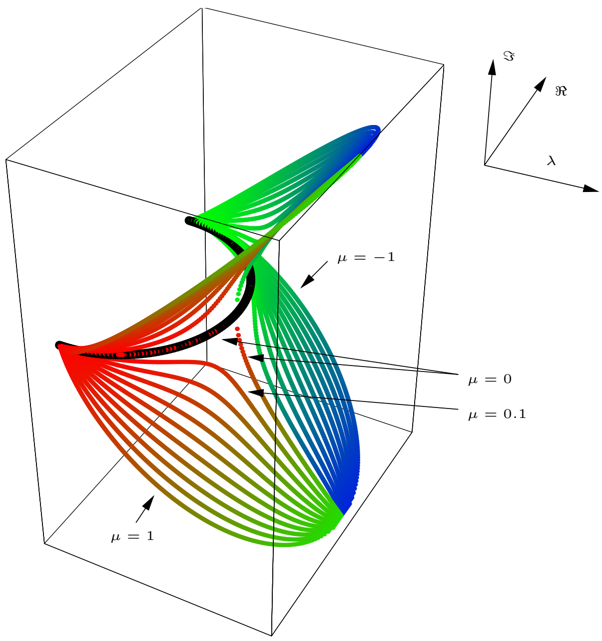

Abbildung 10: Das Zusammenfallen der beiden Wege in der Tangentialsituation wird durch verschiedene Umwege „aufgebogen“ und somit umgangen. Der Parameter μ gibt an, wie stark die Abweichung vom rein reellen Weg ist. Die dargestellten Wege treten beim Schnitt eines Kreises (für $\mu = 0$ zu erkennen) und einer Geraden auf.

Wir können also Stetigkeit in einem Geometriesystem erreichen, indem wir auf Riemannschen Flächen navigieren! Eine genauere Beschreibung dieses Ansatzes, inklusive dem Beweis, dass unter bestimmten Voraussetzungen dieser Weg unumgänglich ist, findet sich zum Beispiel in [5] oder meiner Dissertation [2].

8 Leider erst viel zu spät habe ich gelernt, dass Felix Klein diese Lösung seit jeher klar war: *„Das Prinzip der Kontinuität spricht dann nichts anderes aus, als daß eine analytische Funktion, die längs eines noch so kleinen Stückes ihres Bereiches verschwindet, überhaupt gleich Null ist.“* [1]

Implementationen

Unser persönlicher Durchbruch, der zu dem führte, was ich bisher skizziert habe, fand im Frühjahr 1998 statt. Die erste lauffähige Implementation des „komplexen Verfolgens“, *complex tracing*, funktionierte mit der damaligen Version von Cinderella erstmals. Das warf allerdings auch unsere Pläne zur Veröffentlichung des Programmes noch im selben Jahr über den Haufen, da wir fest entschlossen waren, diese Ergebnisse in die Endfassung zu integrieren. Vor allem, als uns klar wurde, dass wir nicht nur Stetigkeit sondern auch ganz andere Vorteile gewonnen hatten.

Alle abhängigen Elemente verhielten sich nun nicht nur stetig, sondern sogar analytisch (modulo numerischer Schwierigkeiten), und damit konnten wir ein System zur randomisierten Überprüfung von geometrischen Sätzen integrieren (welches den bis dato eingebauten, auf biquadratischen Finalpolynomen basierenden Beweiser ersetzte). Bei diesem Verfahren erzeugt das Programm viele zufällig gewählte Beispiele einer Konstruktion und überprüft an jedem dieser Beispiel die vermutete Aussage. Sollte ein Gegenbeispiel gefunden werden, so können wir die Aussage verwerfen, finden wir keines, so können wir mit hoher Wahrscheinlichkeit davon ausgehen, dass die Aussage stets wahr ist.

Obgleich unsere Implementierung aus Geschwindigkeitsgründen nicht so exakt ist, wie sie mathematisch sein könnte, kann dieser Überprüfer für einen extrem interessanten Teil von *Cinderella* benutzt werden: Geometrieaufgaben mit automatischen Hilfestellungen und Lösungskontrolle, ohne Einschränkung auf einen vorgegebenen Lösungsweg.

Dazu überprüfen wir, ob eine vom Schüler durchgeführte Konstruktion *immer* zum korrekten Ergebnis führt (welches der Autor der Aufgabe per Musterlösung vorgibt). Es müssen aber nicht alle Elemente der Schüler- und Lehrer-Konstruktion übereinstimmen, sondern nur die für das Endergebnis relevanten. Damit werden auch „kreative“ Lösungen als korrekt erkannt (Abbildung 11), selbst wenn nur Standardlösungen (Abbildung 1) zur Überprüfung herangezogen werden.

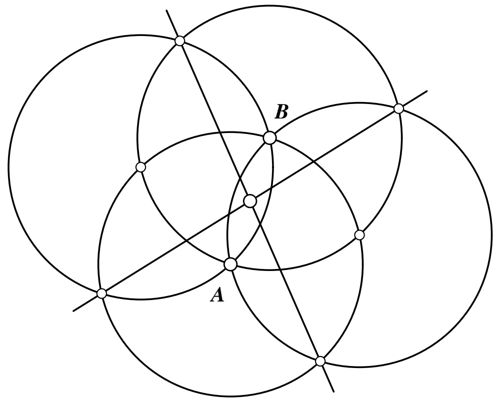

Abbildung 11: Eine originelle Konstruktion für den Mittelpunkt zweier Punkte A und B.

Zurück zur Theorie

Das Jahr 1998 wurde dadurch zu einer Belastungsprobe sondergleichen, aber wir haben es dennoch geschafft, neben den anderen üblichen Aufgaben im Universitätsbetrieb zusätzlich das Programm fertigzustellen. Und nun, nachdem es 1999 dann endlich erschienen ist, können wir uns weiter all den neuen Problemen zuwenden, die bei der Arbeit an Cinderella entstanden sind.

Die Implementation des *complex tracing* warf die Frage auf, ob es nicht eine bessere, einfachere, oder schnellere Möglichkeit gibt, die richtige Instanz nach einer Bewegung zu finden. Ganz grob möchte ich exemplarisch eines unserer Ergebnisse skizzieren. Dieses und andere sind in [10] zu finden.

Die Frage, die wir stellen, lautet: Gegeben eine Konstruktion aus Punkten und Geraden, die nur von den Grundoperationen Freier Punkt, Schnittpunkt, Verbindungsgerade und Winkelhalbierende Gebrauch macht, und zwei Instanzen A und B dieser Konstruktion, die sich nur in einem Element unterscheiden (eine um 90° gedrehte Winkelhalbierende). Wie schwierig ist es zu entscheiden, ob man stetig von A nach B gelangen kann, ohne über degenerierte Situationen zu laufen? Wir werden sehen, dass dieses Entscheidungsproblem zumindest NP-schwer ist.

Der Beweis funktioniert über eine Reduktion von 3-SAT auf das obige Erreichbarkeitsproblem. Dazu konstruieren wir Punkte $X_1, \ldots, X_n$ die als Variablen fungieren, und sorgen dafür, dass sie per Konstruktion auf ein Segment eingeschränkt sind, welches als das Intervall $[0, 1]$ interpretiert wird. Dann übersetzen wir die 3-SAT-Formel von der Aussagenlogik in ein Polynom, in welches wir die Variablen einspeisen. Dieses Polynom wertet nun aus, ob die Formel für eine gegebene Belegung erfüllt ist oder nicht. Geometrisch gesehen wird ein Ergebnis-Punkt $P(X_1, \ldots, X_n)$ konstruiert, welcher einen bestimmten Punkt Q entweder treffen kann oder sich stets in einem gewissen Abstand befindet.

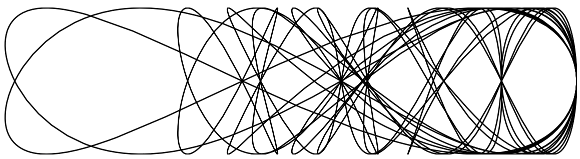

Abbildung 12: Die Ortskurve eines schwierig zu verfolgenden Punktes: Kombinieren wir die Transformation von 3-SAT in das Erreichbarkeitsproblem mit einem Binärzähler, so können wir beweisen, dass es schon NP-schwer ist, zu entscheiden, in welche Instanz eine Bewegung eines einzigen Punktes führt.

Nun lassen wir um diesen Punkt $P(X_1, \ldots, X_n)$ einen weiteren Punkt R auf einem Kreis rotieren,[9] welcher eine Gerade ℓ_1 bewegt. Die Konfiguration ist dabei so gewählt, dass der zweite bestimmende Punkt O der Gerade ℓ_1 nur dann von R umrundet werden kann, wenn die ursprüngliche Formel erfüllt werden kann. Wenn wir nun eine weitere Gerade ℓ_2 durch O legen und die Winkelviertelnde b zu ℓ_1 und ℓ_2 konstruieren, so sind wir fertig: b kann genau dann um 90° gedreht werden, wenn die 3-SAT-Formel erfüllbar ist.

Die Möglichkeit, Polynome und damit auch aussagenlogische Formeln in geometrische Konstruktionen zu codieren, lässt sich für viele weitere komplexitätstheoretische Ergebnisse ausnutzen, ein weiteres Beispiel zeigt Abbildung 12.

Und die Zukunft?

Und – was hält mich nun an der Universität?

Zunächst ist da der Wunsch, das Angefangene zum Abschluss zu bringen: große Teile dessen, was ich zusammen mit Jürgen Richter-Gebert gemacht habe, sind immer noch nicht endgültig aufgeschrieben; teils handelt es sich um Probleme, die zwar schon im Zuge der Implementation von Cinderella „gelöst“ worden sind, aber noch nicht vollständig mathematisch behandelt wurden, teils handelt es sich um Probleme, zu denen es bislang nur Ideen gibt, aber noch keine ausgearbeiteten Gedanken.

Und da sind die vielen neuen Probleme, die sich immer wieder aus der laufenden Arbeit heraus ergeben. Ein Hauptproblemfeld ist für mich derzeit die Kombination der Kontinuität mit Orientierungen. Es ist mir klar, dass ich wenige Absätze zuvor Orientierungen als nicht kompatibel mit Stetigkeit dargestellt habe – aber das ändert nichts daran, dass Orientierungen ein extrem wichtiges und notwendiges Konzept in der algorithmischen Geometrie sind. Nahezu jeder geometrische Algorithmus basiert auf Vorzeichenentscheidungen in der einen oder anderen Form, und die Frage ist, ob diese Entscheidungen in einer kontinuitäts-kompatiblen Art und Weise, zum Beispiel ohne das komplexe Verfolgen völlig zu zerstören, möglich sind. Dies ist ohne Zweifel ein wichtiges Problem, da die *Computational Geometry* derzeit immer öfter gekrümmte Objekte (zum Beispiel Arrangements von Ellipsen) untersucht, nachdem über viele Jahre nur lineare Elemente berücksichtigt wurden, und damit die Mehrdeutigkeitsprobleme der dynamischen Geometrie automatisch auftauchen.

Ein weiteres Gebiet erschließt sich gerade durch die jüngste Fortentwicklung von *Cinderella*. Für eine der nächsten Versionen planen wir die Ankoppelung physikalischer Simulationen an den Geometrie-Kern von Cinderella. Die damit verbundenen Berechnungen führen direkt in die Numerik partieller Differentialgleichungen, ein Thema, mit dem wir uns bisher nicht auseinandersetzen mussten. Immerhin können wir hier auf die Arbeiten vieler Kollegen zurückgreifen.

Und natürlich ist da noch die unvermeidliche Frage nach den höheren Dimensionen. Wann kann man mit echter 3-D-Unterstützung in *Cinderella* rechen? Die Antwort ist schon seit längerem „in frühestens drei Jahren“. Obwohl viele der Methoden aus dem zweidimensionalen auch höherdimensional funktionieren, bedarf eine solche Version noch vieler Arbeit. Nicht nur die Mathematik muss erweitert werden, sondern auch die Konzepte für die Benutzeroberfläche.

Lehre

Der zweite Grund – neben der Forschung – weshalb ich an einer Hochschule zu bleiben möchte, ist die Lehre. Genauso wie ich gerne lerne – und Forschung besteht zum größten Teil aus Lernen – gebe ich das erworbene Wissen gerne weiter. Der beste Ort dafür ist und bleibt die Universität, allen „virtuellen“ modernen Möglichkeiten zum Trotz. Ich freue mich jede Woche aufs Neue, die interessierten Studenten in der Vorlesung „Dynamische Geometrie“, die ich dankenswerter Weise halten darf, zu treffen.

Ein weiterer Punkt ist natürlich aber auch der Einsatz sogenannter neuer Medien in der (universitären und schulischen) Lehre, zur Unterstützung und Ergänzung des Präsenzunterrichts. Durch die Arbeit an *Cinderella* habe ich Kontakte zu vielen Mathematikdidaktikern knüpfen dürfen, und wurde dadurch auf viele Aspekte aufmerksam, die in der Diplomausbildung leider fehlen. Der begonnenen Annäherung der beiden Gesellschaften DMV und GDM sehe ich mit Spannung entgegen, die Anfänge sind vielverspre-

9 Wir können den Satz des Thales benutzen, um ohne einen als geometrisches Objekt gegebenen Kreis einen Punkt auf eine Kreisbahn zu zwingen.

chend! Für mich persönlich sehe ich hier eine Aufgabe (siehe auch [3]): Damit das volle Potenzial der rechnerunterstützten Lehre erschlossen wird, bedarf es der richtigen Software, die wissenschaftlich korrekt und auf dem neuesten Stand ist[10] und in der Umsetzung keine unnötigen neuen Hürden einführt.[11] Wenn dann noch die richtigen didaktischen Konzepte, die diese Möglichkeiten ausschöpfen, hinzukommen, können wir auf eine bessere Ausbildung hoffen. Für dieses Ziel brauchen wir aber eine bessere Kommunikation zwischen Fachdidaktik und Mathematik als sie derzeit leider üblich ist.

Anwendung und Freiheit

Forschung und Lehre allein wären aber zu wenig. Die Anwendung des Geforschten und die Brauchbarkeit des Gelehrten außerhalb des Campus ist für mich der Rettungsanker, der mir die Gelassenheit gibt, der es bedarf, um den Verlockungen der „freien“ Wirtschaft zu entgehen. Es ist gut und wichtig zu wissen, dass die Qualifikationen, die man an der Universität erwirbt und weitergibt, auch dort zählen, wo es tatsächlich keine Freiheit gibt, sondern Businesspläne und Shareholder-Value.

Einer der Gründe für die „Kommerzialisierung“ von *Cinderella* [8] [9] war der Zwang dazu, das Programm in eine Form zu bringen, die auch außerhalb des universitäts-lokalen Rechnernetzes überlebt. Gleichzeitig wäre es aber nicht möglich gewesen, ein Projekt wie *Cinderella* privatwirtschaftlich durchzuführen: Nur über unsere wissenschaftliche Freiheit konnten wir uns über viele Forderungen (von potenziellen Anwendern und Verlegern) zunächst hinwegsetzen und die Punkte durchsetzen, die uns am Herzen lagen. Nur dadurch konnten wir uns auf die Lösung der mathematischen Probleme konzentrieren und Pfuscherei verhindern. Auch das ist der Auftrag der Universitäten: ein Umfeld zu bieten, in dem nicht in vorderster Linie an Wirtschaftlichkeit gedacht werden muss, um so auf lange Sicht bessere Qualität zu liefern. Wir sollten darauf achten, dass dies im (vernünftigen!) Bestreben nach mehr Wirtschaftlichkeit nicht geopfert wird.

Geständnis

Es sind also diese drei Haken – Forschung, Lehre und (freie) Anwendung – die mich seit der Promotion an der Universität halten. Ich muss aber an dieser Stelle gestehen, dass ich nicht „der Mathematik treu“ geblieben bin, wie in der Einleitung geschrieben. Tatsächlich bin ich sogar schon während der Promotion zum Informatiker mutiert, und auch jetzt bin ich an einem Institut für Informatik beschäftigt. Allerdings – ich fühle mich noch immer der Mathematik zugehörig, und empfinde die Trennung von Mathematik und Informatik an vielen Stellen als künstlich und unnötig. Es mag für die praktische und technische Informatik nicht gelten, aber die theoretische Informatik ist und bleibt zu einem Großteil Mathematik. Ich bin froh, dass es in Deutschland auch noch viele gemeinsame Fachbereiche der Mathematik und Informatik gibt, und kann mir nur wünschen, dass der Austausch zwischen den Gebieten nicht nur an der FU Berlin gut funktioniert.

Literatur

[1] Felix Klein. *Elementarmathematik vom höheren Standpunkt aus. Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen*, Band 2. Springer-Verlag, Berlin, 1925, Nachdruck 1968.

[2] Ulrich Kortenkamp. *Foundations of Dynamic Geometry.* Dissertation, ETH Zürich, Institut für Theoretische Informatik, November 1999. `http://www.cinderella.de/papers/diss.pdf`

[3] Ulrich Kortenkamp. The future of mathematical software. In *Proceedings of MTCM 2000.* Springer-Verlag. To appear, 2001.

[4] Ulrich Kortenkamp and Jürgen Richter-Gebert. A dynamic setup for elementary geometry. In *Proceedings of MTCM 2000.* Springer-Verlag. To appear, 2001.

[5] Ulrich H. Kortenkamp and Jürgen Richter-Gebert. Grundlagen Dynamischer Geometrie. In *Oberwolfach-Tagungsband.* Franzbecker. To appear, 2001.

[6] Jean-Victor Poncelet. *Traité des propriétés projectives des figures.* Gauthier-Villars, 1822.

[7] Oswald Giering. *Visualisierung und klassische Geometrie Mitteilungen der DMV* 1–2001.

[8] Jürgen Richter-Gebert and Ulrich H. Kortenkamp. *The Interactive Geometry Software Cinderella.* Springer-Verlag, Heidelberg, 1999. `http://www.cinderella.de.`

[9] Jürgen Richter-Gebert and Ulrich H. Kortenkamp. *Cinderella – die interaktive Geometriesoftware.* HEUREKA-Klett Softwareverlag, Stuttgart, January 2000.

[10] Jürgen Richter-Gebert and Ulrich H. Kortenkamp. Complexity issues in Dynamic Geometry. In *Proceedings of the Smale Fest 2000.* To appear 2001.

Adresse des Autors
Dr. Ulrich Kortenkamp
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
`kortenkamp@inf.fu-berlin.de`

10 Damit man auch wirklich das vermittelt, was man vermitteln möchte, und nicht ein schwaches, verzerrtes Abbild der Realität – siehe dazu auch den Artikel von Oswald Giering [7]!

11 Bloß weil Mathematik manchmal schwierig ist, darf die dazugehörige Software nicht schwierig sein