

Feli-X: ein Prototyp zur Integration von CAS und DGS

Reinhard Oldenburg

Angaben zur Veröffentlichung / Publication details:

Oldenburg, Reinhard. 2003. "Feli-X: ein Prototyp zur Integration von CAS und DGS." In Lehr- und Lernprogramme für den Mathematikunterricht: Bericht über die 20. Arbeitstagung des Arbeitskreises "Mathematikunterricht und Informatik" in der Gesellschaft für Didaktik der Mathematik e.V. vom 27. bis 29. September 2002 in Soest, edited by Peter Bender, Wilfried Herget, Hans-Georg Weigand, and Thomas Weth, 123–32. Hildesheim: Franzbecker.

● Feli-X: Ein Prototyp zur Integration von CAS und DGS

Reinhard Oldenburg, Göttingen

Das Beispiel der Parabelkonstruktion macht deutlich, dass die Alternative zwischen den Werkzeugen CAS und DGS unglücklich ist. Anhand eines Prototypen werden die Möglichkeiten eines integrierten CAS-DGS diskutiert.

1 Einleitung

Computeralgebrasysteme (CAS) und dynamische Geometrieprogramme (DGS) gehören zur selbstverständlichen Werkzeugsammlung des modernen Mathematikunterrichts. Nun gibt es einen beachtlichen Bereich mathematischer Inhalte, die mit beiden Werkzeugen bearbeitet werden können. Mit einem Programm, das beide Fähigkeiten vereint, könnte man sich die Qual der Wahl ersparen und zudem neue didaktische Wege eröffnen. In diesem Aufsatz soll ein solches Programm vorgestellt werden, das das CAS Mathematica und ein selbst geschriebenes DGS integriert.

Zunächst wird der didaktische Hintergrund beschrieben, aus dem sich die Zielsetzung einer solchen Integration ergeben hat. Dazu wird am Beispiel der Einführung von Parabeln die Alternative zwischen CAS und DGS herausgearbeitet. In der anschließenden Diskussion des Programms Feli-X wird dieses Beispiel wieder aufgenommen.

Die hier vorgestellte Arbeit entspringt einem Bild von Mathematik und Mathematikunterricht, das mit den folgenden Schlagworten umrissen werden kann:

- Mathematik ist eine ermächtigende Wissenschaft, und Schüler müssen den Zuwachs an eigener Macht erfahren können.
- Mathematik ist ein Werkzeug und benutzt Werkzeuge.
- Werkzeuge sind nicht passiv, sie prägen als Interaktionspartner die mathematischen Inhalte.
- Die Wahl des richtigen Werkzeugs ist eine zu erwerbende Kompetenz.

2 Parabeln

Es ist Konsens der Mathematikdidaktik, dass Parabeln als geometrische Objekte und nicht

nur als Graphen quadratischer Funktionen behandelt werden sollten. Bei Parabeln ist ihre Brennpunkt- bzw. Leitlinieneigenschaft zentral für ihre technische Anwendung und bietet somit reiche Vernetzungsmöglichkeit.

Eine Parabel in der Euklidischen Ebene E wird festgelegt durch einen Brennpunkt F und eine Leitgerade g (die F nicht enthält). Die zugehörige Parabel ist die Menge derjenigen Punkte, deren Abstand von g genauso groß ist wie ihr Abstand von F . Also: $\{P \in E \mid d(P, g) = d(P, F)\}$.

Diese mathematische Definition gilt es unterrichtlich umzusetzen. Im folgenden werden Erfahrungen mit verschiedenen Ansätzen dazu berichtet.

3 Parabeln mit DGS

Weitgehend dem Ansatz des innovativen Mathematikschulbuches Mathenetz folgend, habe ich zunächst andere, bekannte Abstandseigenschaften (Mittelsenkrechte, Kreis) wiederholt, um dann auf die Konstruktion der Parabel mittels Tangente (Abb. 1) zuzusteuern.

Leider hat kein einziger meiner Schüler diese Vorlage der Mittelsenkrechten genutzt. Statt dessen kamen Schüler auf verschiedene Varianten der Idee, einen Kreis von variablem Radius um F zu schlagen und eine Parallele zu g im Abstand des Kreisradius zu konstruieren. Die Schnittpunkte von Kreis und Parabel liegen dann natürlich auf der Parabel (Abb. 2).

Eine Analyse der kognitiven Anforderungen der beiden Wege fördert zu Tage, warum der zweite Weg einfacher ist: Er setzt nämlich die geforderte Eigenschaft geradliniger um. Von der geforderten Abstandseigenschaft der Parabelpunkte kann man durch gerichtete Argumentationsschritte unmittelbar die Richtigkeit der Konstruktion einsehen. Im ersten Fall

dagegen, ist es einfacher, von der fertigen Konstruktion rückwärts zu argumentieren.

Besonders interessant ist die, auch in Abb. 2 gezeigte und von mehreren Schülern parallel

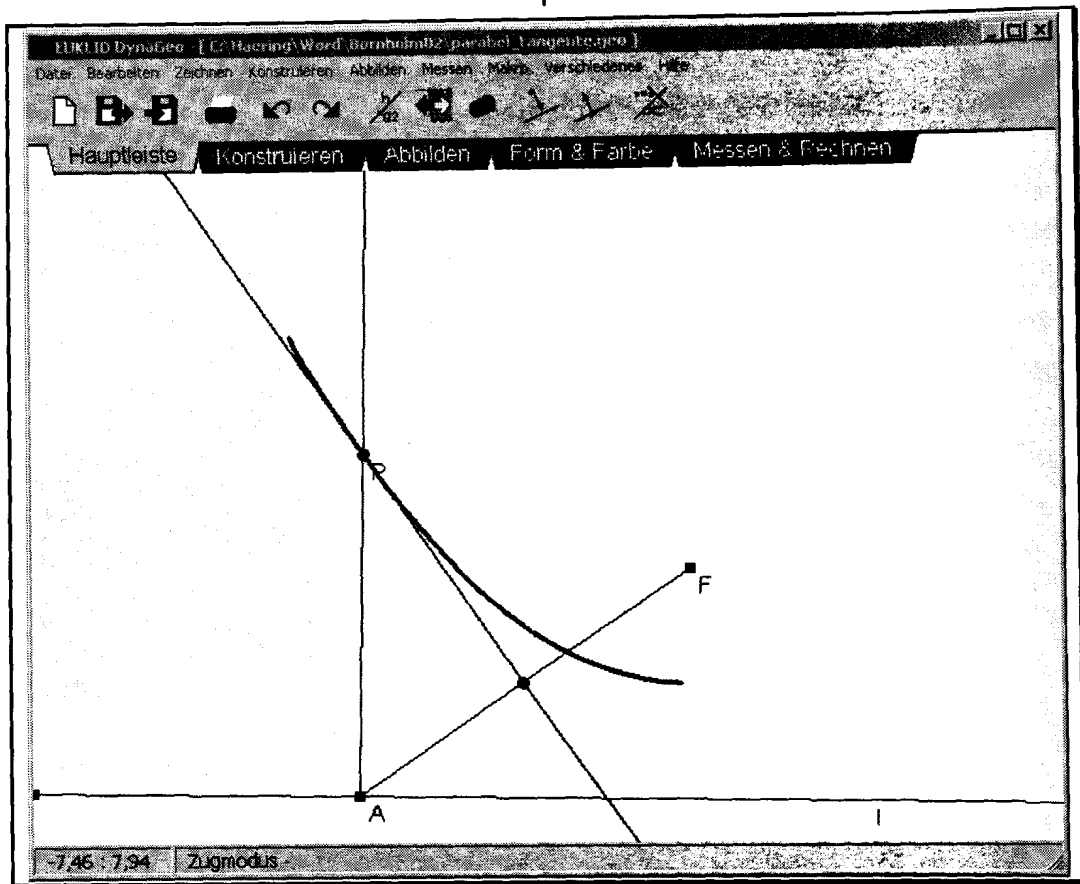


Abb. 1: Die Leitlinienkonstruktion der Parabel mit Tangente

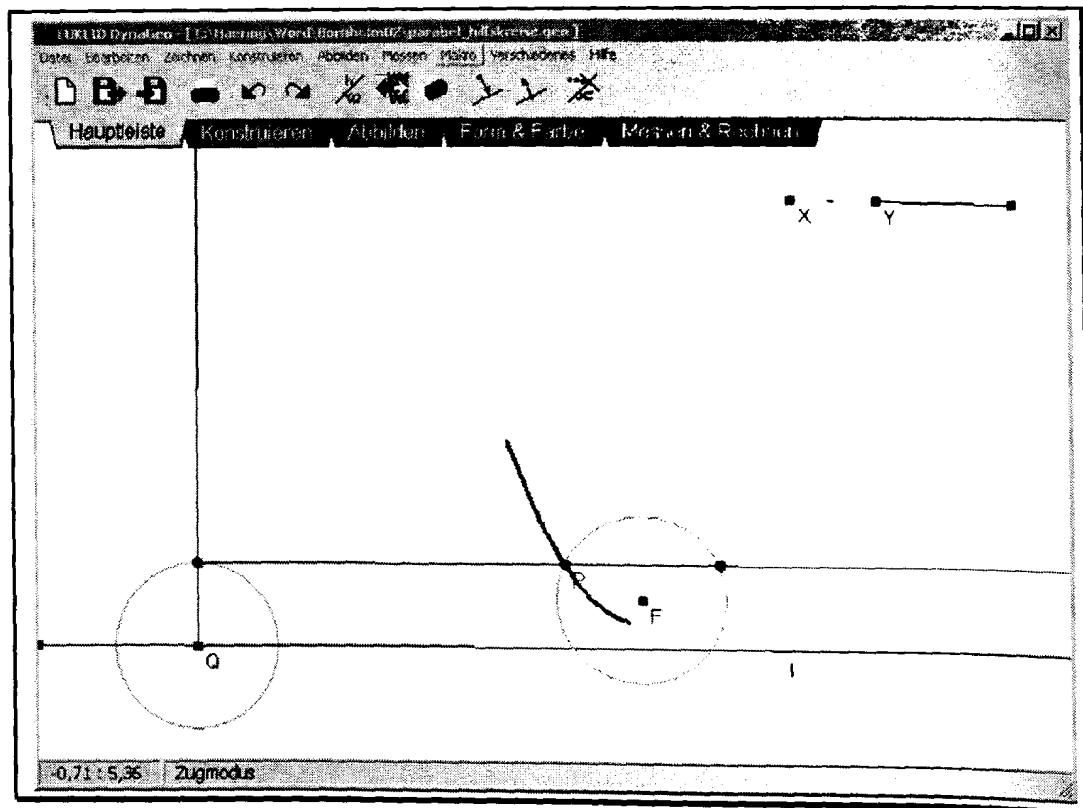


Abb. 2: Die Konstruktion der Parabel mit einem Schieberegler und Hilfskreisen

entdeckte Variante, den gemeinsamen Radius und Parallelenabstand durch einen selbstgebauten Schieberegler vorzugeben, also eine algebraische Variable einzuführen. Dies legt den Schluss nahe, dass ein algebraischer Zugang angemessen sein könnte.

4 Parabeln mit CAS

Da zum Zeitpunkt der Parabeleinführung der Satz des Pythagoras in der Regel zur Verfügung steht, kann die Definition der Parabel (bezogen auf ein Koordinatensystem) in Gleichungen übersetzt werden. Mit Derive kann das wie in Abb. 3 geschehen. Wenn man #2, #3 und #4 zeichnen lässt, erhält man das in Abb. 3 rechts gezeigte Bild.

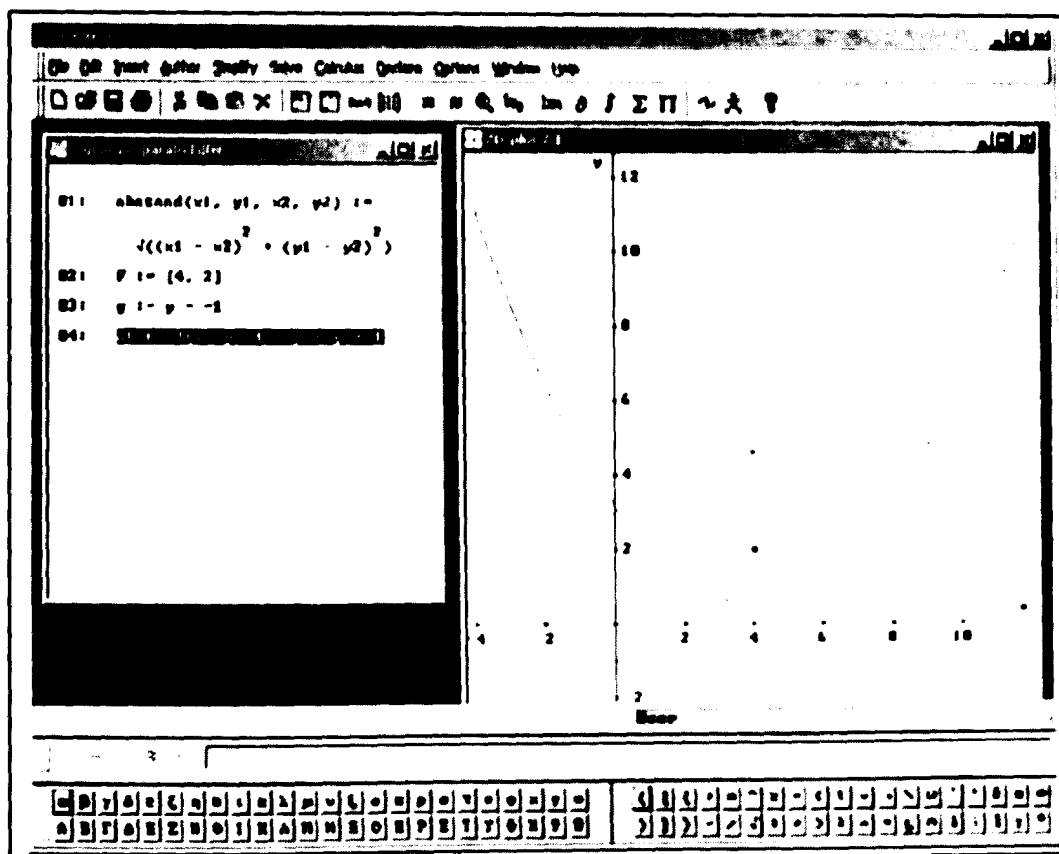


Abb. 3 Die Parabel in Derive

Bei diesem Zugang hatten einige Schüler Probleme, mit der abstrakten algebraischen Ausdrucksweise. Andererseits ermöglichte sie auch eine verhältnismäßig leichte Erweiterung auf die Abstandsdefinitionen anderer Kegelschnitte. Obwohl die Schüler diesen Vorteil sahen, erklärte in einer abschließenden Befragung die Mehrheit die Veranschaulichung der Parabel mit Euklid sei schöner. Hauptargument war, dass das fertige Bild noch mit der Maus verändert werden konnte.

5 CAS vs. DGS

Die bisherige Diskussion sollte zeigen, wie das jeweils gewählte Werkzeug schon die Problemwahrnehmung und erst recht die Lösung verändert. Die Werkzeuge bestimmen damit maßgeblich die ablaufenden Denkprozesse. Es ist damit zu rechnen, dass der Umgang mit einem bestimmten Werkzeug zu charakteristischen Fixationen führt.

Eine weitere bedenkenwertere Punkt ist die naheliegende Parallelität zwischen der DGS/CAS-Scheidung und der von Schwank (1996) vorgenommenen Unterscheidung der Präferenz für funktionalen/prädikativen Denkstil. Es könnte sich schon von daher als geboten erweisen, verschiedenen Schülern verschiedene Zugangsmöglichkeiten zu eröffnen.

Da aber nach dieser Theorie der jeweils nicht präferierte Stil auch gefordert werden kann, scheint ein integratives Werkzeug auch aus diesem Blickwinkel sinnvoll zu sein.

6 CAS und DGS mit Feli-X

Aus den oben dargestellten Überlegungen ergibt sich der Wunsch nach einem System, das CAS und DGS integriert. Mit dem Programm Feli-X steht ein Prototyp zur Verfügung.

gung, der als experimentelles System zwar noch nicht für Schüler geeignet ist, es aber erlaubt, erste Erfahrungen mit diesem Ansatz zu sammeln.

Gegenwärtig ist Feli-X in Mathematica geschrieben, da dieses CAS derzeit das einzige ist, dessen Kern über dokumentierte Schnittstellen verfügt, die die benötigten Dienste bereitstellen. Da langfristig der Umstieg auf ein anderes CAS geplant ist, ist die gegenwärtige Phase bewusst experimentell gehalten. Sie dient vor allem der Exploration und Auswahl von Eigenschaften, die dann in solider Form in einem zweiten System implementiert werden sollen.

In Kürze könnte man Feli-X durch die folgenden Eigenschaften umreißen:

- *Kleines, offenes Mathematica/Java-Programm*

Die Übersichtlichkeit des Quellcodes und die Offenheit soll es engagierten Anwendern erlauben, Feli-X als Bausteinkiste für eigene Lern- und Forschungsumgebungen zu verwenden.

- *Mathematica-Notebook und Grafikfenster parallel benutzbar und sich gegenseitig beeinflussend*

Dies ist die Kernidee der interaktiven Integration von CAS und DGS.

- *Mächtigkeit des CAS und seiner Sprache überall verfügbar, universelles Werkzeug, anpassbar*

Dem Paradigma des Werkzeugs folgend, soll Feli-X der Phantasie des Nutzers möglichst wenig Schranken auferlegen.

- *Gleichungen als zentrales Modellierungsmittel (teilweise mit Zusatzinformationen)*

Eine Mindestforderung ist, dass sich die interaktiv oder via Programm erstellte Konstruktion nicht nur in Form einer Konstruktionsbeschreibung sondern auch in algebraischer Form ausgeben lässt. Vom technischen Standpunkt aus erlaubt dies die Ankopplung eines automatischen Beweissystems. Vom didaktischen Standpunkt aus dient dies der Vernetzung von Geometrie und Algebra und ermöglicht verschiedenste Untersuchungen der Konstruktion. In Feli-X ist aber auch die Ankopplung an das Gleichungssystem bidirektional, d.h. Änderungen des Gleichungssystems bewirken unmittelbar eine Änderung der geometrischen Konfiguration.

- *Ansatz möglichst allgemein, spätere Spezialisierung*

Die Programmierung von Feli-X erfolgt so, dass zunächst Möglichkeiten geschaffen werden, und erst in einer zweiten Runde ihr Nutzen bewertet wird. Bei sinnvollen Möglichkeiten stellt sich dann auch die Frage nach der Optimierung der Performance.

- *Kapselung verschiedener DGS-Kerne*

Dem experimentellen Ansatz folgend wurde versucht, durch möglichst viele verschiedene Strategien Dynamik in die Konstruktion zu bringen.

- *Keine Unterscheidung von abhängigen und unabhängigen Punkten*

Wenn man die Konstruktion mit Gleichungen beschreibt, wird klar, dass die strenge Unterscheidung von abhängigen und freien Punkten, wie sie die traditionelle DGS trifft, keine mathematische Notwendigkeit ist. Feli-X erlaubt deshalb auch das Ziehen an abhängigen Punkten. Diese Möglichkeit hat technische und didaktische Konsequenzen, über die noch zu sprechen sein wird.

Abbildung 4 zeigt die typische Situation für die Arbeit mit Feli-X. Auf der rechten Seite sieht man das Mathematica-Notebook, auf der linken Seite das Grafikfenster. Zusätzlich ist in diesem Beispiel ein Schieberegler für „myphi“ zu sehen. Mit diesem setzt der Benutzer den Wert der Mathematica-Variable `myphi`, und deren Veränderung bewirkt in der abgebildeten Situation eine Rotation des Dreiecks im linken Fenster. Diese Rotation wurde dabei über eine Rotationsmatrix gesteuert, die im Mathematica-Notebook definiert wurde.

Die Bidirektionalität der Kommunikation beider Programmfenster ermöglicht es, geometrische Objekte auf zwei Arten zu verschieben (mit der Maus oder durch Angabe der neuen Koordinaten) und Variablen auf zwei Arten mit neuen Werten zu belegen (mit einer Mathematica-Zuweisung oder durch den Zugmodus der DGS).

Die analoge Bidirektionalität zwischen geometrischer Konstruktion und Gleichungssystem fordert eine Doppelung der graphisch relevanten Variablen: In den Gleichungen werden Symbole wie `xc[„P1“]` für die x-Koordinate des Punktes P1 verwendet. Die aktuellen Werte dieser Variablen werden aber getrennt gespeichert und sind als `XC[„P1“]` verfügbar. Die Funktion `InsertValues` erlaubt, in einem Term einige oder alle Varia-

blen durch ihre aktuellen Werte zu ersetzen. Dies muss aber von Hand geschehen.

er durch die nachfolgenden Gleichungen ohnehin umgelenkt wird.

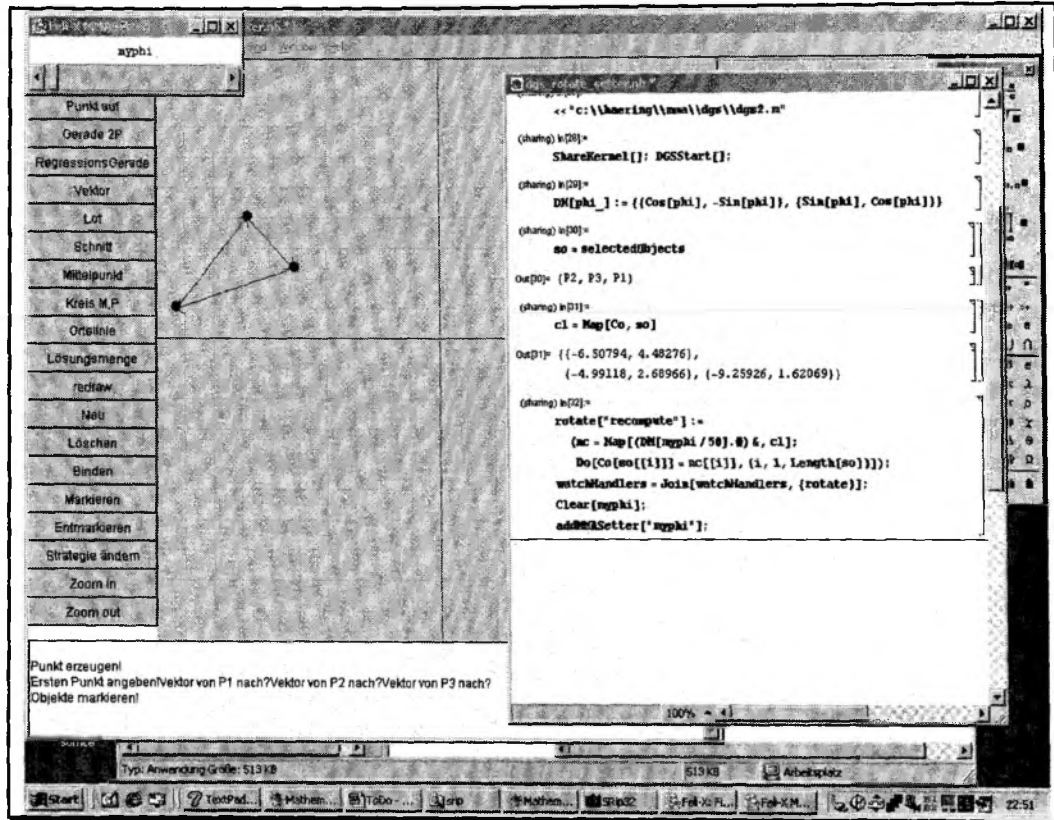


Abb. 4: Die Arbeitsumgebung von Feli-X

Die folgende Sequenz von Feli-X/Mathematica-Befehlen konstruiert eine Dynagraph Anwendung (vgl. Elschenbroich 2003), bei der ein Punkt (hier P1, auf die x-Achse festgelegt) mit der Maus gezogen wird, und ein zweiter Punkt auf einer zweiten Achse (hier P2 auf der um 2 in y-Richtung verschobenen Parallelen) gemäß einer Funktionsvorschrift (hier $x \mapsto x^2$) wandert.

```
P1=addObject[„point“, 0, 0];
P2=addObject[„point“, 1, 1];
addEquation[yc[P1]==0];
addEquation[yc[P2]==2];
addEquation[xc[P1]^2==xc[P2]];
```

Die ursprünglich für P2 angegebenen Koordinaten sind vollkommen bedeutungslos, da

In dieser Beschreibung wird nirgends verwendet, dass P1 unabhängig (ein Basispunkt) und P2 abhängig (ein konstruierter Punkt) ist. Folgerichtig erlaubt es Feli-X, an P2 zu ziehen, wobei sich dann P1 mitbewegt. Feli-X arbeitet also mit Gleichungen als Relationen. Herkömmliche DGS dagegen nehmen eine funktionale Modellierung dar. Dass dieser traditionelle Ansatz einengend wirkt, zeigt sich an den (allerdings sehr beschränkten) Möglichkeiten der Umdefinition von Punkten in Cabri und Euklid. Es sei allerdings nicht verschwiegen, dass die der relationale Ansatz eine Reihe von technischen und mathematischen Schwierigkeiten mit sich bringt. Einige davon bilden aber unmittelbar wichtige mathematischen Eigenschaften

Benutzeraktion	Reaktion von Feli-X
<pre>F=addObject[„point“, 3, 4]; g=addObject[„line“, 1, -4, 2]; P=addObject[„point“, 5, 5]; redrawDGS[];</pre> <p>Verschieben von P mit der Maus <pre>addEquation[dist[P, F]==dist[P, g]];</pre> Verschieben von P mit der Maus</p> <p>Button „Lösungsmenge“, dann P anklicken Verschieben von F mit der Maus</p>	<p>Es werden eine Gerade und zwei Punkte gezeichnet. P ist frei beweglich P springt (?!, nämlich auf die Parabel) P ist nur noch auf der (noch unsichtbaren) Parabel beweglich Es erscheint eine Parabel Die Parabel wird nachgezogen</p>

Tab. 1: Erzeugung einer Parabel mit Feli-X

ten und Einschränkungen ab, die auch didaktisch interessant sind. Beispielsweise machen sich nicht-eindeutige Zuordnungen und Einschränkungen im Bildbereich bemerkbar.

Wie gestaltet sich die oben beschriebene Einführung der Parabeln mit Feli-X? Tabelle 1 auf der Vorseite zeigt die Entwicklung.

Form durch versuchsweises Verschieben von P ertastet werden. Der Nutzer zieht dabei an P als dem Punkt, der ihn interessiert, nicht an einem Basispunkt, der lediglich Hilfsmittel in einer Konstruktion ist.

Einer der DGS-Kerne, die das oben beschriebene Ziehen an abhängigen Punkten erlaubt, heißt „FindMinimum“ nach der

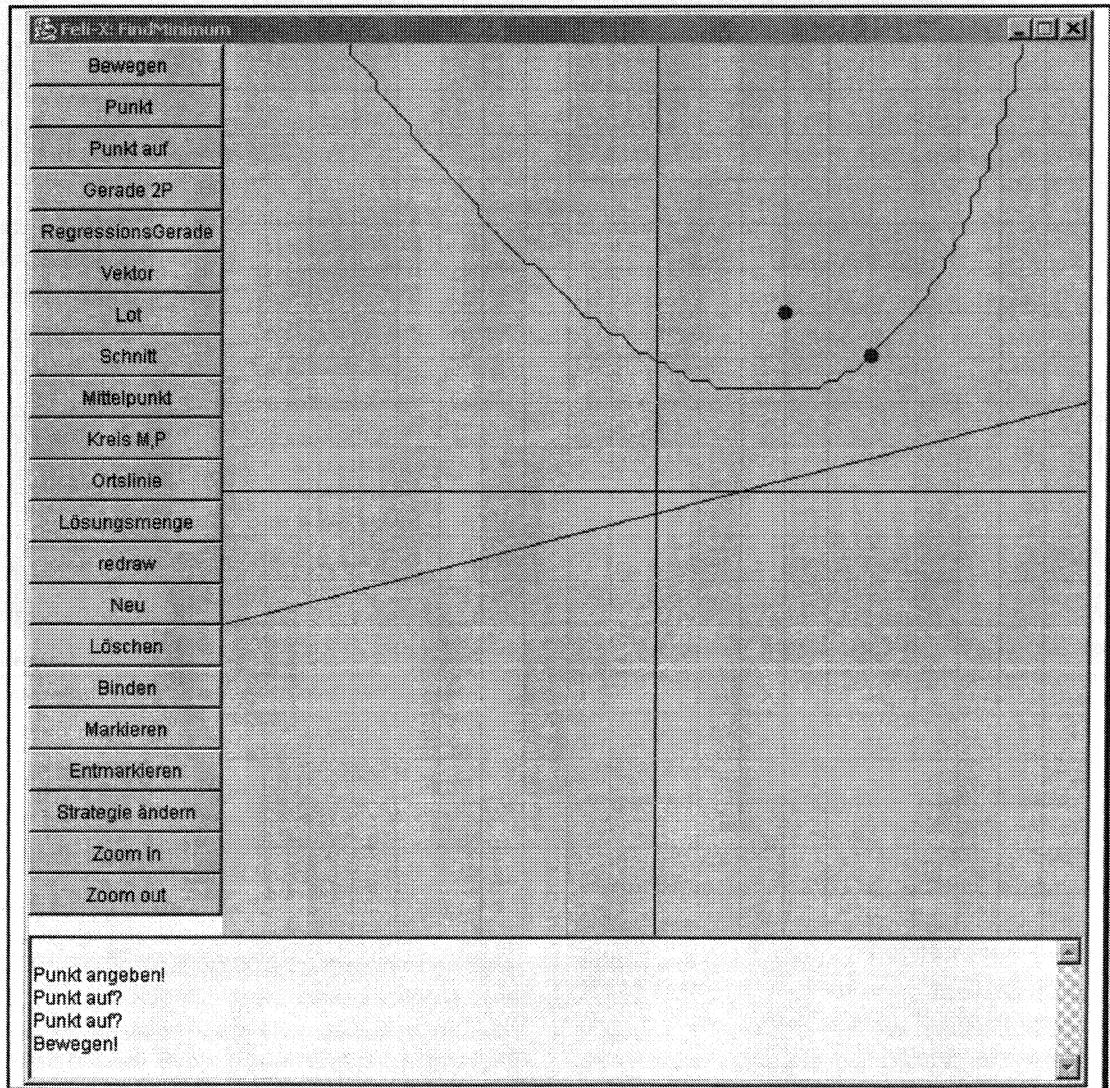


Abb. 5: Die Parabel in Feli-X

Das Endstadium der Konstruktion ist in Abb. 5 zu sehen. Die grobe Rasterung der Parabel ist das Resultat einer programmtechnischen Notlösung, die ein Scheitern von Mathematicas `ImplicitPlot` gerade an diesem Beispiel kompensiert. Im Unterscheid zur Ortslinienfunktion klassischer DGS (die es in Feli-X auch gibt) gibt es hier keinen Basispunkt, an dem gezogen werden könnte, um die Parabel abzufahren. Dafür aber kann, noch bevor die Parabel gezeichnet wird, ihre

gleichnamigen Mathematica-Funktion, die er benutzt. Er leistet noch etwas erstaunliches: Wenn man in der Abbildung 6 den rechten Kreismittelpunkt (der an die Gerade gebunden ist) nach links verschiebt bewegt sich der Kreisschnittpunkt, auf den ein Vektor als Marker gesetzt ist, stetig. Dieses Beispiel wird im Handbuch von Cinderella verwendet, um Unstetigkeit in traditionellen DGS aufzuzeigen.

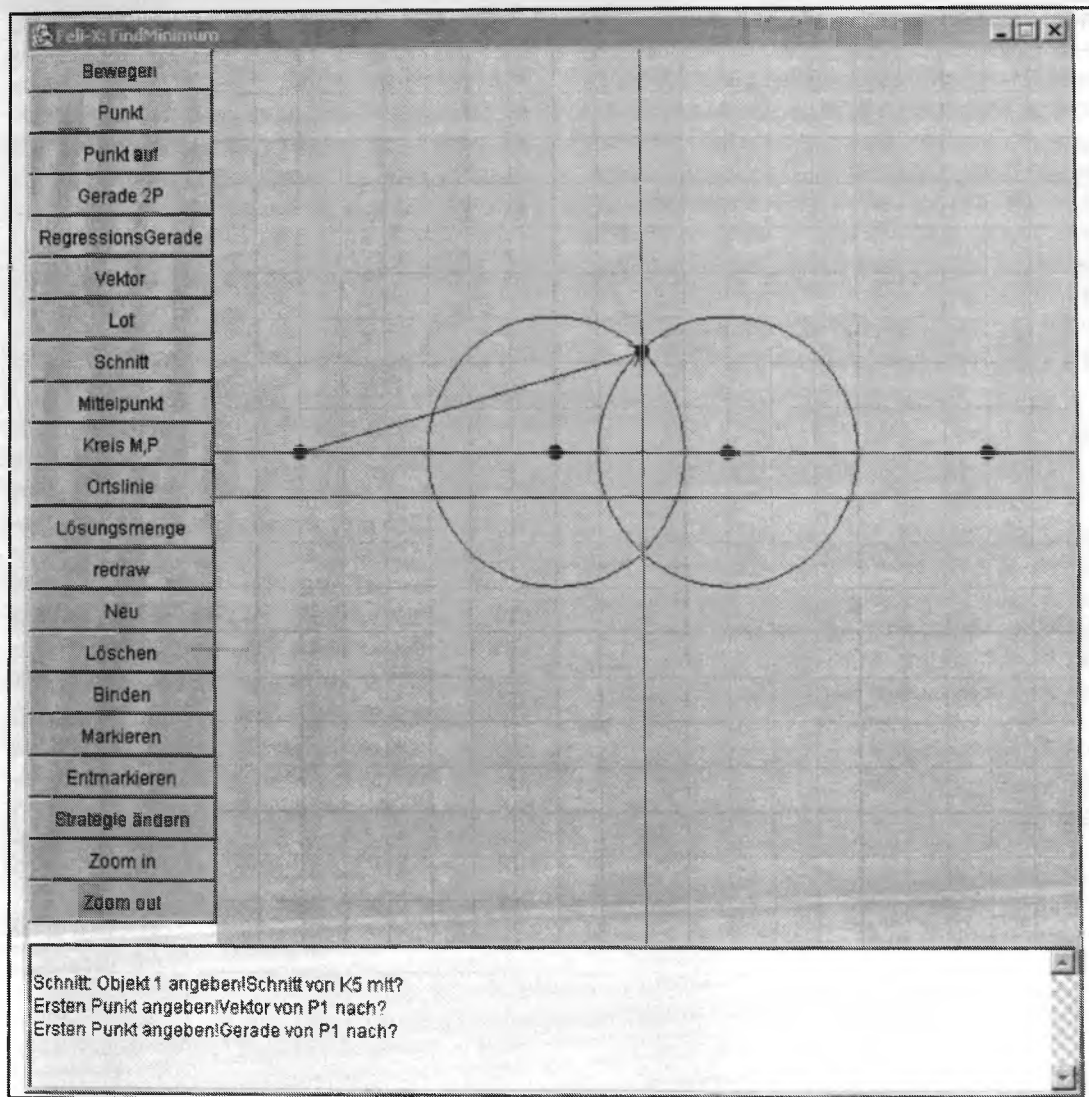


Abb. 6: Eine bekannte Stetigkeitsfalle für dynamische Geometrieprogramme im Fenster von Feli-X

7 Beispiel eines einfachen DGS-Kerns

Die Aufgabe eines DGS-Kerns im hier beschriebenen Rahmen ist, aus einer gegebenen Lösung des Gleichungssystems (alte Konfiguration) eine neue Lösung zu finden, bei der ein Teil der Variablen die durch die Mausbewegung gegebenen neuen Koordi-

naten hat. Dabei können die Bewegungen Einschränkungen unterliegen, beispielsweise kann ein an ein Objekt gebundener Punkt nicht an jede Stelle mit der Maus gezogen werden. Das legt nahe, das Finden der neuen Konfiguration als Optimierungsproblem aufzufassen. Die folgende Mathematica-Definition ist der vollständige Code des Kerns „FindMinimum“, der für die obigen Beispiele verwendet wurde, und der diese Idee realisiert:

```

DGmoveF[name_, Calt_, Cneu_] := Module[{eqs, sol, RV},
  RV = Map[#, InsertValues[#]] & , Vars];
  eqs = Join[Equations, {xc[name] == Cneu[[1]],
    yc[name] == Cneu[[2]]}];
  sol = Apply[FindMinimum,
    Append[Prepend[RV,
      Apply[Plus, Map[#[[1]] - #[[2]]^2 &, eqs]],
      MaxIterations -> 200]];
  UpdateFromRules[sol[[2]]]; ];

```

Tab. 2: DGS-Kern „FindMinimum“

Diese Funktion übernimmt den Namen des zu ziehenden Objekts (hier notwendig ein Punkt) sowie seine alten (hier gar nicht benötigt) und neuen Koordinaten. Zum aktuell gültigen Gleichungssystem werden die Bedingungen hinzugefügt, dass die neuen Koordinaten die vorgegebenen Werte annehmen sollen (Variable eqs). Zu allen Variablen wird ihr aktueller Wert ergänzt (Variable RV) und aus beiden Bestandteilen wird ein FindMinimum-Befehl geformt, der die eigentliche Berechnung vornimmt.

8 Phasenraum-Modell

Schon der Begriff „Dynamisches Geometriesystem“ legt nahe, die Bewegungen im Zugmodus im Sinne dynamischer Systeme der Physik zu deuten. Dazu gehört zunächst, dass man neben dem Konfigurationsraum auch den Impulsraum der Objekte betrachtet. Das wirft ein interessantes Licht auf die von Kortenkamp & Richter-Gebert (2000) mit Cinderella bekannt gemachte Stetigkeitsproblematik. Betrachten wir als Beispiel die Konstruktion der Winkelhalbierenden mit Hilfe dreier Kreise gleichen Radius (Abb. 7). Wenn man A in Richtung S mit konstanter Geschwindigkeit bewegt, bewegt sich auch P

Richtung S mit konstanter Geschwindigkeit. Das traditionelle Euklid, und auch die neueren Versionen in der Einstellung des deterministischen Zugmodus, zeigen ein überraschendes Verhalten, wenn A schließlich durch S hindurch läuft. Dann zweigt P nämlich orthogonal ab, biegt also aus die zweite Winkelhalbierende ein. Eine durch S und P eingezeichnete Winkelhalbierende würde in diesem Moment also springen. Der Punkt P zeigt hier im Ortsraum nichtdifferenzierbares, im Geschwindigkeitsraum sogar unstetiges Verhalten. Dies ist allgemein typisch:

In allen in der Cinderella-Diskussion untersuchten Beispielen geht eine Unstetigkeit im Ortsraum mit einer Nichtdifferenzierbarkeit im Impulsraum, und umgekehrt eine stetige, aber nicht glatte Stelle im Ortsraum mit einer Unstetigkeit im Impulsraum einher. Das legt den Verdacht nahe, dass die Einbeziehung des Impulsraums das Stetigkeitsproblem lösen könnte.

Eine naheliegende physikalische Interpretation der geometrischen Objekte eines DGS ist es, die Punkte als Massepunkte zu betrachten, die durch Stangen und Räder verbunden sind. Man erhält so ein Hamiltonsches System mit vielen Zwangsbedingungen. Betrachten wir die Mathematik dazu:

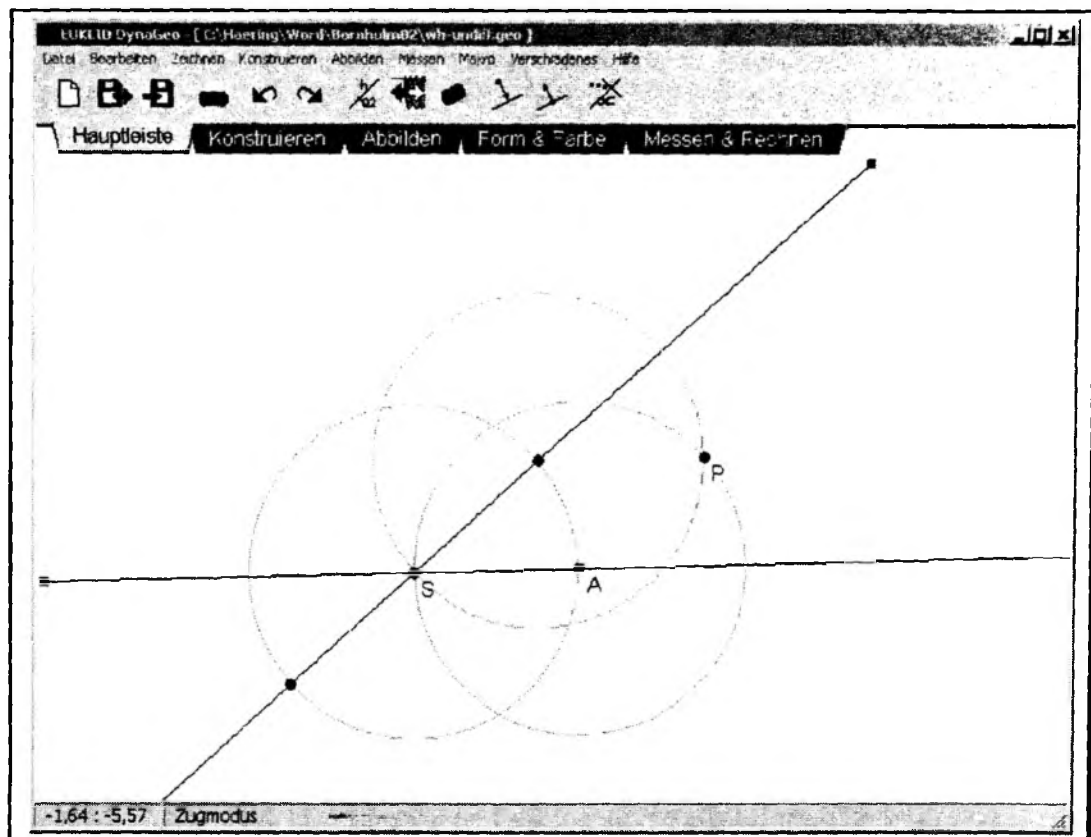


Abb. 7

Es seien $q_i, i=1\dots n$ die Koordinaten aller geometrischer Objekte (also z.B. auch die Radien von Kreisen usw.). Zwischen diesen Koordinaten bestehen aufgrund der Konstruktion Zwangsbedingungen (meistens algebraische Gleichungen): $g_j(\{q_i\})=0, j=1\dots k$. Man hat nun die Wahl der Masse aller Objekte m_i . Für die Bestimmung der Lagrangefunktion verzichten wir auf potentielle Energie, so dass sich als Lagrangefunktion ergibt:

$$L = \frac{1}{2} \sum_{i=1}^n m_i \dot{q}_i^2. \text{ Die Euler-Lagrange-Gleichungen unter den Zwangsbedingungen lauten}$$

$$\frac{d}{dt} \frac{\partial}{\partial \dot{q}_i} - \frac{\partial}{\partial q_i} = \sum_{j=1}^k \frac{\partial g_j}{\partial q_i} \lambda_j, \text{ für } i=1\dots n. \text{ In}$$

unserem einfachen Fall ergibt sich also ein-

$$\text{fach: } m_i \ddot{q}_i = \sum_{j=1}^{k+2} \frac{\partial g_j}{\partial q_i} \lambda_j. \text{ Um zu einem System erster Ordnung zu kommen, führt man Geschwindigkeiten } v_i \text{ ein und erhält als zu lösendes System:}$$

$\dot{q}_i = v_i, i=1\dots n$

$$\dot{v}_i = \frac{1}{m_i} \sum_{j=1}^{k+2} \frac{\partial g_j(\{q_i\})}{\partial q_i} \lambda_j$$

$$0 = g_j(\{q_i\}, t)$$

Dieses Anfangswertproblem ist bei jeder Mausbewegung zu lösen. Die Anfangswerte der Koordinaten q sind die Werte in der alten Konfiguration. Alle Geschwindigkeiten haben den Anfangswert 0. Man kann in dieses System, wenn man will, auch Informationen über die Konstruktion (Abhängigkeitsgraph) eingeben, indem man für die Koordinaten, die von der Bewegung nicht betroffen sind, die konstanten Koordinaten von vorneherein eingibt.

Für dieses algebraisch-differentielle Gleichungssystem kann man glücklicherweise auf eine ausgearbeitete Theorie (Hairer et al. 2002) zurückgreifen, die einem die Existenz einer eindeutigen Lösung garantiert, wenn die Jacobimatrix der Zwangsbedingungen maximalen Rang hat. Es hat noch keine systematische Untersuchung dieser Bedingung stattgefunden, aber einfache Beispiele zeigen, dass sie in der Regel erfüllt ist. Wenn dem so ist, hat man eine differenzierbare und damit sogar stetige Lösung auf dem Phasenraum. Diese Lösung der Stetigkeitsproblematik ist im Vergleich zu der von Cinderella zu sehen. Die in Cinderella vorgenommene Erweiterung auf komplexe Koordinaten steht in Analogie zur Erweiterung auf den Phasenraum im hier diskutierten Modell.

Die in Feli-X bis dato implementierte Version dieses Verfahrens ist leider sehr ineffizient, so dass nur die aller einfachsten Beispiele damit berechnet werden konnten. Je nach Wahl der Massen zeigt sich ein unterschiedliches Verhalten, das zum Teil gewöhnungsbedürftig ist. Beispielsweise kann man manchmal am Massenschwerpunkt die ganze Konstruktion translateren, oder durch andere Züge rotieren lassen. Es muss sich noch zeigen, welche Varianten von Schülern als natürlich empfunden wird.

9 Didaktische Bewertung

Das hier vorgestellte Programm kann einige Hoffnungen erfüllen:

- Die Verbindung von algebraisch/abstrakter und dynamisch/visueller Sichtweise wird geschaffen. Feli-X trägt damit zur Vernetzung bei. Es ermöglicht, klassische Inhalte auf mehr Kanälen zu transportieren als nichthybride Werkzeuge.
- Die Offenheit des Systems kann die Informationsflüsse transparent machen.
- Die Bedeutung der Gleichungen wird erfahrbar.
- Feli-X ist ein Baukasten, mit dem Dynamograph-Anwendungen schnell realisiert werden können. Es ermöglicht einen dynamischen Zugang zur Frage der Umkehrbarkeit einer Zuordnung.

Neben technischen Fragen wie der derzeit noch zu geringen Arbeitsgeschwindigkeit und gelegentlichen Programmabstürzen sollten vor allem die folgenden didaktischen Fragen in Zukunft genauer untersucht werden:

- Steht das Ziehen an beliebigen Punkten im Widerspruch zur didaktisch wertvollen Idee der geometrischen Konstruktion sukzessivem, also funktionalem Bauplan?
- Können Schüler sinnvoll mit der Beschreibung eines geometrischen Sachverhaltes durch ein (unübersichtliches?) Gleichungssystem umgehen?
- Stellt die Unterscheidung von Variabler (z. B. $x_C[P]$) und ihrem Wert (z. B. $XC[„P“]$, was zum jeweils aktuellen Wert evaluiert) eine kognitive Hürde dar? Oder ist es die Chance das Variablenkonzept neuartig zu visualisieren?

10 Weitere Ziele

Es sollen in Zukunft Schritte unternommen werden, um in absehbarer Zukunft Schüler mit einem solchen System arbeiten lassen zu können. Im einzelnen stehen an:

- Ausbau
 - Integralkurven, Vektorrechnung, Tangenten etc...
 - Anknüpfung an Beweissystem
 - Algorithmische algebraische Geometrie
 - Dynamisch angekoppelte Tabellenkalkulation
- Richtung Mathematik
 - Theorie des physikalischen Kerns (Stetigkeit, Eichtheorie, schneller numerischer Löser)
 - Homogene komplexe Koordinaten
- Richtung Schüler
 - Auswahl und Reifung geeigneter Kerne
 - Neuimplementation in MuPAD oder Maple
 - Verdeckte Linien, Farben, Makros,...
 - Dokumentation
 - Evaluation

Wenn die technische Umsetzung dieses Plans gelingt, können auch die oben aufgeworfenen didaktischen Fragen experimentell untersucht werden.

Literatur

- Cukrowicz, Jutta & Zimmermann, Bernd (2001): MatheNetz 9. Braunschweig: Westermann
- Elschenbroich, Hans-Jürgen (2003): Funktionen dynamisch entdecken. In diesem Band
- Hairer, Ernst et al. (2002): Geometrical Numerical Integration. New York: Springer
- Kortenkamp, Ulrich & Richter-Gebert, Jürgen (2000): Cinderella. Berlin: Springer
- Schwank, Inge (1996): Zur Konzeption prädikativer versus funktionaler kognitiver Strukturen und ihrer Anwendung. In: Zeitschrift für Didaktik der Mathematik 28, 168–183
- Wolfram, Stephen (2001): Mathematica. A system for doing Mathematics by Computer. Champaign: Wolfram Press