ALGEBRAIC MODELLING USING A DYNAMICALLY LINKED GEOMETRY AND COMPUTER ALGEBRA SYSTEM

Reinhard Oldenburg

University of Education Heidelberg, Germany

Algebraic modelling is the process of describing situations in terms of variables and equations. If the situation at hand has a geometric interpretation this process can be supported by software that offers simultaneous algebraic and geometric representations. This can be achieved by dynamically linking a computer algebra system and a dynamic geometry system. A prototype FeliX of such a system is presented and it is discussed how the new technical possibilities can influence the learning process.

ON THE IMPORTANCE OF ALGEBRAIC MODELLING

Algebra and geometry have a rich common history and often insight into problems of one domain could be gained by translating it to the other domain. With the advent of the computer this balance of the two fields has shifted somewhat to the side of algebra: Computers are algebraic machines, they calculate terms either numerically or symbolically. Therefore, using a computer for some kind of mathematics, may it be geometry, optimization or statistical analysis requires an algebraic (in the widest sense) formulation of the problem at hand. Often, this is only needed when a program is implemented, not when it is used. We all know that dynamic geometry systems give users the illusion of manipulating geometric objects directly. Nevertheless, there is an underlying algebraic model and at times the algebra scratches the surface, for example in the way computations of angles are handled.

Geometry can play an important role in the pupil's development of algebraic thinking. French (2002) devotes an entire chapter to suggestions exploiting this connection. Duval (1999) as cited by Hohenwater (2006) emphasises the importance of this connection by stating that "There is no true understanding in mathematics for students who do not incorporate into their cognitive architecture the various registers of semiotic representations used to do mathematics."

Several studies have shown that the use of multiple representations can support the learning process of algebraic concepts, see e.g. Stacey et al. (2004) and the references given there (especially in chapter 6). However, most of these studies focus on the functional aspect of the algebra-geometry correspondence. This is far from being unexpected as the functional concept of variation neatly fits with the dynamic interaction with a software system. The work reported here aims at extending this successful use of multiple representations to the relational aspects of algebra.

Observations of the practice of experts in applied mathematics (engineers) have shown that they need to be very flexible in modelling situations by equations and in interpreting equations in the given situation. This involves both a functional understanding (how does a value change if another is varied) as well as a relational under-

standing (what are the constraints imposed by the geometry of the object or by physical laws). The same holds true for the applications of mathematics at the secondary school level. In physics lessons students are required to combine the knowledge on a vertically falling object encoded as an equation $y = y_0 - \frac{1}{2} g t^2$ with the description of a horizontal movement $x = x_0 + v_0 t$ by eliminating variables. Another subject is the calculation of currents, resistances and voltages in electric circuits which requires them to algebraically combine equations. A study in Germany (Beckmann 2000) revealed that many students lack the mathematical competencies to handle such physical applications.

In the literature algebraic modelling is almost completely seen from the functional point of view (e.g. Janvier 1996). While this functional approach certainly is of outstanding importance (both for applications as well as for understanding of concepts) it has the tendency to neglect relational aspects.

We conclude that algebraic modelling is an important ability that is worth to support by specialised learning environments. The pedagogical rationale is to enhance students' understanding of algebra and especially equations as a tool for modelling that allows one to express relations and study their meaning. The research questions derived from this are: a) How should a software system be designed that uses geometry to let students explore equations and foster their mental link between algebra and geometry? b) How do students work with such software, what strategies do they develop? c) What impact has the work with the software on the students' concept development?

ALGEBRAIC DYNAMIC GEOMETRY SYSTEMS

Computer tools such as computer algebra systems are perfect tools for experienced algebraic modellers but for beginners they lack interactivity. The user has to pose "what if..." questions in a very concise form (e.g. by deriving equations and plotting them in dependence of a parameter). On the other hand dynamic geometry systems are very explorative environments but they lack the power of algebra. We propose therefore an integration of these two kinds of software.

The kind of system we have in mind shall be called an algebraic DGS or ADGS for short. An ADGS shall offer an algebra window and a geometry window that interact and update each other mutually. These windows shall offer full CA and DG functionality.

The bidirectional connection shall be established on two levels:

On the level of the configuration (algebraically: coordinates) we have the operations of object creation and modification and of entering arbitrary coordinates.

On the level of the geometric relations (algebraically: equations) the user shall be able to impose, relax or modify them in geometric or algebraic form. Especially, the

user shall be allowed to enter arbitrary equations and inequalities that have to be respected during dragging.

An important point is that the dragging behaviour shall be governed by a simple rule: The user may forbid the movement of certain objects (fixed objects). The others move in such a way that the equations hold. This is, at least in principle, all that needs to be said to explain the systems behaviour.

A prototype of such a system has been called FeliX and it has been first realized in 2002 as an add-on for Mathematica, followed by a version for MuPAD in 2005.

Apart from FeliX there seems to be no other system that aims at this tight integration. Geogebra (Hohenwater 2006) is purely functional, GeometryExpressions (Todd 2006) offers (a subset of) geometric constraints and calculates loci algebraically but lacks the opportunity to enter and investigate algebraic relations freely. Yet other research systems aim at automated geometric theorem proving. None of these systems can claim to offer unlimited multiple representations.

THE PROTOTYPE FELIX

FeliX for the commercial computer algebra system MuPAD is invoked from a MuPAD worksheet and the MuPAD window is accessible all the time. Within the worksheet interface one has full read and write access to all objects, coordinates, equations, object creation operations and so on. For example, one may use the MuPAD programming language to automate certain constructions or to make animations. The MuPAD worksheet is the right place to do advanced algebraic manipulations starting from data constructed using the geometric tools. However, for the quick inspection of equations or the input of new equations it is more convenient not to have to switch the window. Therefore, all this information is present in the FeliX window itself as well. It is divided into a sub-window for geometry on the left and the combination of an object browser (for reading and changing coordinates, colours, fixed property, names) and an equation browser (to read, enter and modify equations and inequalities). Moreover, there is a MuPAD input line and output area that enables teletype style interactions with the MuPAD kernel without selecting the MuPAD window. This is convenient if the screen is too small to show both windows in a reasonable size at the same time.

To enter equations, the user, at least at the moment, has to use 1-d math input. Coordinate variables of objects are written for points P as x[P] and y[P], for circles as x[C], y[C], r[C] and for lines as coefficients in the equation $ax+by=c$, i.e. as a[L], b[L], c[L]. Further objects are segments, vectors, and curves (which may be function graphs, parametric curves or implicit curves).

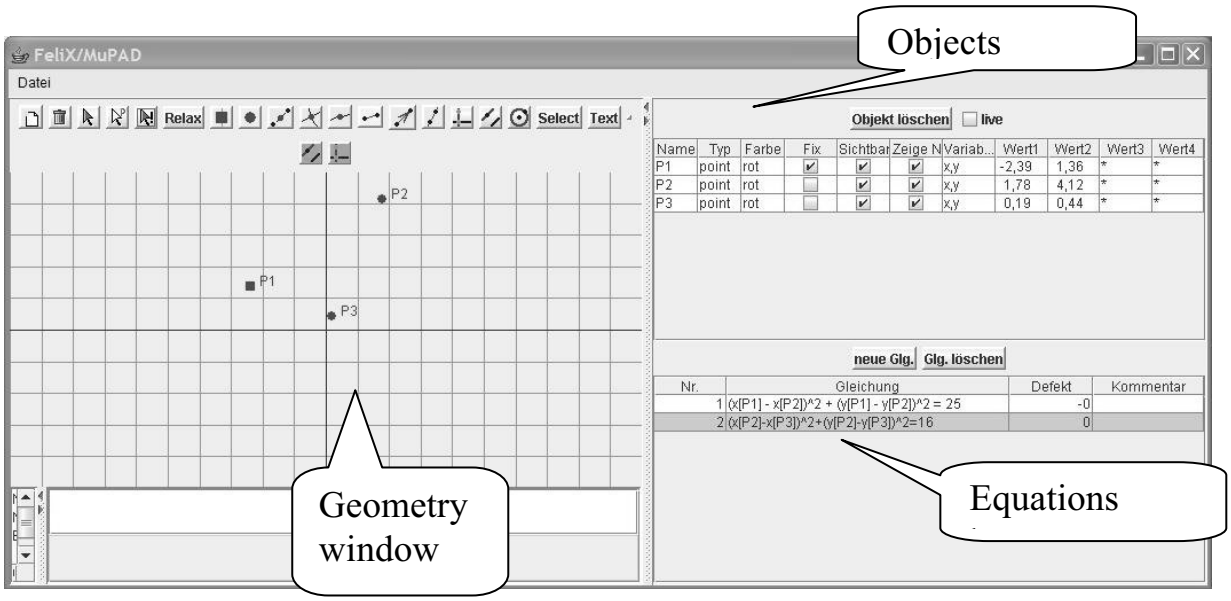

Fig. 1: The structure of the FeliX main window

In the example shown in Fig. 1 the construction consists just of three points. One may e.g. enter the equations `x[P1]+x[P2]=2*x[P3]` and `y[P1]+y[P2]=2*y[P3]` to declare P3 to be the midpoint of P1 and P2 (in the screenshot some other equations are shown in the equation browser). Or one changes the second equation to `y[P1]+2*y[P2]=3*y[P3]` and observes the impact of this "rule".

MODELLING EXAMPLES

Here we list some further examples of modelling tasks that can be handled with FeliX in a novel way. We start with Snellius' law of refraction:

Many physical situations can be modelled by a set of equations. The construction on the right shall be interpreted as a light ray S1 that hits a drop of rain and gets refracted to direction S2. Besides the obvious geometrical incidence relations we only have one physical equation that governs the physical behaviour: Snellius' law. It can be entered almost in the form found in textbooks:

```
sin(angle(S1,Normal))=
1.4*sin(angle(S2,Normal))
```

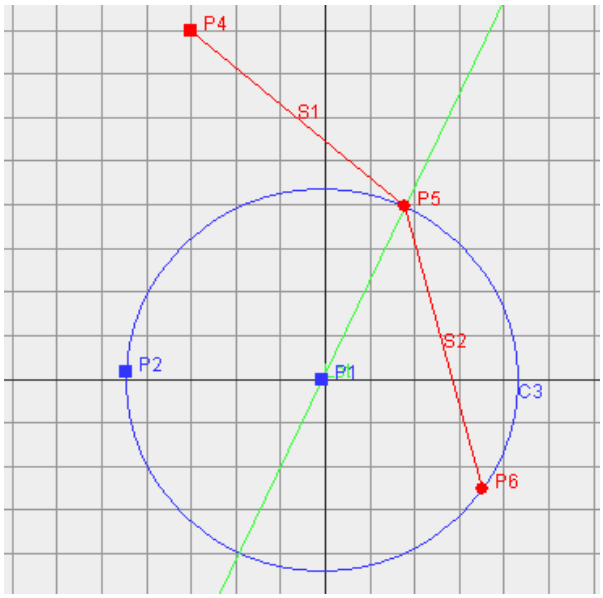

Fig. 2: Rain drop

This example shows that modelling with equations in FeliX is not restricted to algebraic equations but can include transcendental equations as well.

Modelling coins

Coins on a table and circles in a dynamic geometry system behave quite differently: In the DGS the radius of a circle may change and the circles may overlap. Students that have mastered the theorem of Pythagoras can add the algebraic rules that are needed to model the physical coin situation.

If there are two circles C1, C2 the following set of equations can be entered into FeliX to make the situation quite realistic:

`r[C1]=2` Fix the first radius
`r[C2]=2` Fix the second radius
`(x[C1]-x[C2])^2+(y[C1]-y[C2])^2>4^2` Distance of midpoints of the circles must be greater than 4.

With these equations in force one may drag one circle with the mouse and push the other around the window. Moreover, one may wish to set the x-axes as a "floor" by demanding `y[C1]>2, y[C2]>2`.

This example shows that FeliX can handle inequality constraints as well. This is a very important feature, because thinking in inequalities is not adequately supported by current educational software. Related problems for students are to restrict point to move only in a half-circle or inside a lens shaped region.

A LEGO robot

It is important to model situations of interest to the students. A nice example is the analysis of the mechanism of the following little Lego "robot" which is capable of moving along a thread. Observing the real robot is very interesting because its movement is quite involved. The hand goes slowly backwards and then goes forward very quickly. Moreover, it seems not to describe a circle. We will use FeliX to analyze this mechanism.

The natural geometric model (Fig. 4) consists of a point P1 of the axes of the motor, a circle C3 that describes the orbit of the end of the “elbow”, a fixed point P5 where the “lower arm” segment must pass through and a point P7 which is the position of the robot's hand. The distance between P4 (on C3) and P7 must be constraint to be of some fixed value. Plugging this information into FeliX one gets the model of the mechanism that can be moved by dragging P7 or P4. The mechanism can be deformed by moving the circle or by moving P5. FeliX is able to calculate the equation of the orbit of the robot's hand symbolically. In the case at hand, the equation starts

with $x^6 - 4x^5 \cdot x[P1] - 2x \cdot x[P5] + 3x^4 y^2 - \ldots$ which illustrates that FeliX calculates the orbit symbolically including its dependence on parameters like x[P1].

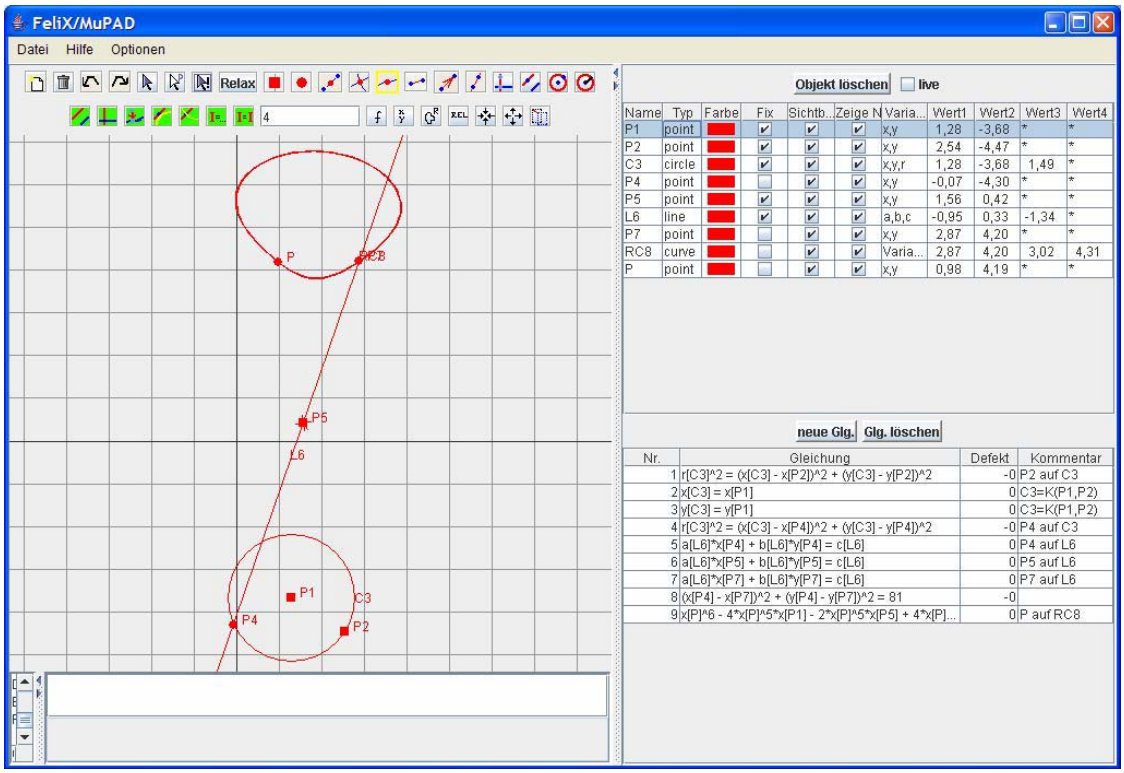

Fig. 4

A 3-bar linkage

Three-bar linkages have been studied very intensively because they generate complex mathematics from a very simple situation. Such a linkage can be realised e.g. by the following LEGO construction.

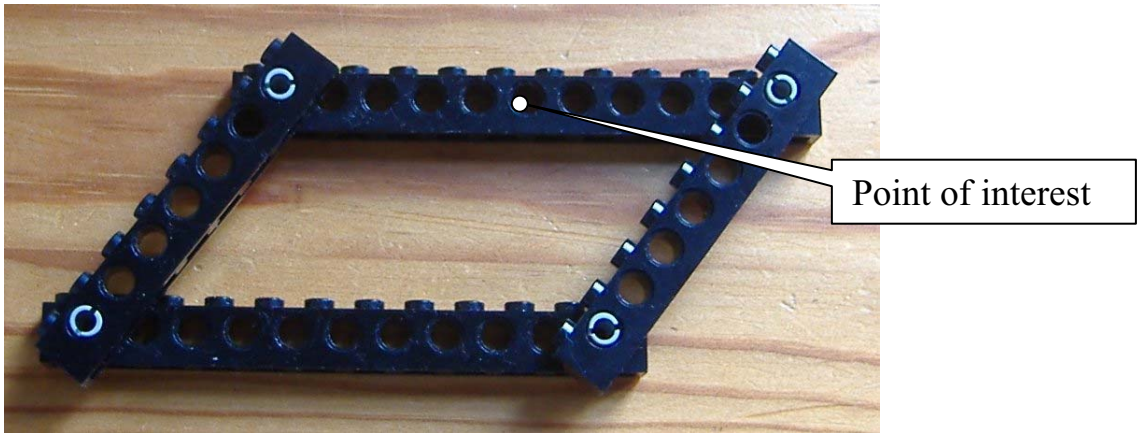

Fig. 5: A 3-bar linkage build with 4 LEGO bricks. The lower horizontal segment is considered to be fixed and is therefore not counted as a bar.

One asks for the orbit of the midpoint of the middle segment (see Fig. 5). As long as the linkage forms a parallelogram this is simply a circle, but the linkage has a second configuration in which this midpoint moves on a lemniscate. To investigate this model one simply needs to describe the distance relations realized by the rigid segments. The model then shows precisely the same behaviour as the real linkage and

FeliX can calculate the symbolic orbit. To cast it into a more readable form we substitute simple values for the coordinates of the endpoints of the lower segment:

- `subs(FelixCurves[1][3],xc["P1"]=-8,xc["P2"]=8,yc["P1"]=0,yc["P2"]=0)`

$$X^2 + Y^2 - 64 \cdot 4 \cdot X^4 + 8 \cdot X^2 \cdot Y^2 - 256 \cdot X^2 + 4 \cdot Y^4 + 768 \cdot Y^2 = 0$$

As the result is printed in factorized form it is easy to see that the orbits consist of two components, the circle and the lemniscate.

A TEACHING SEQUENCE

In the sections above the technical features of an ADGS have been illustrated. This kind of software can be used in many places in the curriculum from the introduction of Cartesian coordinates and equations to advanced algebraic geometry of curves. As a concrete example we will consider a sequence that has been taught in a 9th grade classroom after introduction of the theorem of Pythagoras. The students had modest experience with a traditional DGS (Euklid Dynageo). The introductory example was to create a point P and enter the equation x[P2]+2*y[P]=4. Then students had to observe how P can be moved with the mouse. They easily found that P is restricted to a straight line. They typically checked the validity of the equation at various positions of P. Only a minority used knowledge about linear equations from the 8th grade from the beginning. In subsequent tasks the students had to modify the equation such that the point comes closer to the origin or that its line has a greater slope. During these activities several students made the mistake not to modify the equation but to add a second equation. As a result, the point was rigid and didn't move at all. This observation brought up lively discussions. The validity of the equations was checked several times and finally the students explained themselves that the points is now at the intersection of the lines defined by the two equations. Then some recalled knowledge about systems of linear 2 by 2 equations and restated the explanation as the point being the unique solution of the system of equations.

This episode from the first lesson with FeliX illustrates typical learning trajectories: In a first approach the students check the dragging behaviour of objects against the equations by calculating if the equations are satisfied. (In fact, I was really surprised to see how many calculations were performed in the beginning. This shows that a very crucial phase of the instrumentation process concerns the building of confidence in the tool. By doing these vast amounts of calculations the students both build up this confidence and fostered their mental geometry-algebra link.) Only after the numerical approach students incorporate concepts like "linear equation" in their argumentation. This pattern could be observed with every new kind of problem introduced, with the "numerical phase" shortening as the confidence in FeliX grew.

The sequence went one with some nonlinear examples and with example that related two points. E.g. the students had to achieve that A is always on the left of B or that B is always exactly one unit "higher" than A. In these examples the facility of FeliX to declare some objects as fixed so that they don't move in response to the movement of

other objects was introduced. This technique was very quickly adopted by the students and most of them used it systematically to explore the situation. This observation, the episode with the rigid point subject to two linear equations and further observations e.g. with triangles with sides of fixed lengths show that in a ADGS environment the concept of "degrees of freedom" becomes crucial for the understanding of the behaviour of the construction. It is interesting that the physical term "degrees of freedom" is much better suited to describe this situation than the static mathematical term of "dimension of a solution manifold".

In the next part of the sequence the students had to model geometric distance problems. This started of course with the basic problem of setting the distance between two points to a prescribed value. As can be expected, the students at first made no link to the theory learned before but tried to add simply the x-distance and the y-distance with several students asking about the way the absolute value function can be used in FeliX. This misconception turned out not to be viable when dragging the point. Some students even used rulers on the computer's screen to convince themselves. But then the Pythagorean theorem was used to solve this and a vast set of related problems including e.g. restricting points to lie on or inside or outside a circle, restricting a point to lie inside the intersection of two circles, making a triangle rigid using the SSS congruence theorem, exploring parabolas and ellipses as loci of points.

CONCLUSION

In this paper we have concentrated on FeliX as a modelling tool. There are other features of such a system. From a purely geometric point of view it is interesting to investigate how a construction behaves if certain constraints are set. E.g. one may investigate which kind of constraints suffice to make a quadrilinear rigid. As one does not have to specify a construction sequence, many interesting problems (e.g. trisection of an angle) can be solved without doing constructions.

A first evaluation of FeliX based upon the sequence described above took place early in 2006 with 9^{th} graders from a german "Gymnasium". As reported the students were very engaged in formulating equations and in interpreting them. In a final free-answer questionnaire they stated their impression that they better understood what equations between the coordinates of points express. Further results were that they judged FeliX to be easily usable and that the link between algebra and geometry is very clear.

FeliX is free but it requires the commercial computer algebra system MuPAD (www.mupad.com) to run. FeliX can be downloaded from http://www.ph-heidelberg.de/wp/oldenbur/felix.

REFERENCES

A. Beckmann: Bereitet der Mathematikunterricht auf die Mathematik im Physikunterricht vor? *mathematica didactica* 23 1 (2000), 3-23.

R. Duval: Representation, vision, and visualization: Cognitive functions in mathematical thinking. Basic issues for learning. In: Hitt (Ed.); Santos (Ed.): Proceedings of the twenty-first annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education Bd. 1, 1999, p. 3-26

D. French: Teaching and learning algebra. London, New York 2002.

M. Hohenwater: Geogebra. www.geogebra.at

M. Hohenwater: GeoGebra – didaktische Materialien und Anwendungen f¨ur den Mathematikunterricht. Salzburg 2006.

C. Janvier: Modeling and the initiation into Algebra. In: N. Bednarz et al. (Eds.): Approaches to Algebra. Dordrecht 1996.

R. Oldenburg: Bidirektionale Verknüpfung von Computeralgebra und dynamischer Geometrie, *Journal für Mathematikdidaktik* 26 (2005), 249-273.

K. Stacey et al.: The Future of the Teaching and Learning of Algebra. Boston, London 2004.

Ph. Todd: Geometry Expressions. www.geometryexpressions.com.