RELATIONAL THINKING SUPPORTED BY AN ALGEBRAIC MODELING TOOL ON THE WEB

Reinhard Oldenburg

Universität Augsburg; reinhard.oldenburg@math.uni-augsburg.de

A new web-based implementation of the FeliX system that combines algebra and geometry in a way that supports relational thinking is introduced. It allows to explore the role of algebra for the modelling of geometrical relations. The rationale behind the system is described, its design principles based on mathematical logic are explained and some use-cases are described. Especially, the didactical implications of a new feature that incorporates minimization is elaborated. Examples range from elementary explorations of geometric and algebraic relations to advanced applications.

Keywords: Algebra, geometry, educational software, relational thinking.

INTRODUCTION

Algebra offers different thinking tools; among the most important are variables, expressions, functions and equations. Working with and thinking about functions is coined together into the famous concept of functional thinking, which dates back to the days of Felix Klein more than 100 years ago (Weigand et al., 2017). Despite its age, the concept is still of great importance, and the use of spreadsheets and dynamic geometry systems such as Geogebra put even more weight on this. Both types of software realize a functional view of computations: There are certain input elements (cells with numbers respectively basic objects) from which other objects are functionally dependent (cells with formula respectively constructed objects). Dependent objects may be the input of further calculations or constructions so that a directed graph of dependencies is constructed. When an input cell is changed, or a basic object is moved then following the route of this graph allows propagating recalculation through all dependent objects. Thus, existent technology uses and fosters functional thinking.

Functions have a direction: $x \mapsto f(x)$. This is the reason why they are so often useful in modelling causal chains or channels of information transport. However, in the real world, there are also many relations that are undirected. Consider economics: Are the wages a function of the prices or the prices a function of the wages? Or consider physics: Is the pressure of a gas a function of its volume, or is its volume a function of the pressure? In such examples, there is no clear direction and functional thinking falls short as a mental tool to model such situations. What is needed is relational thinking. This concept is not as widespread as functional thinking. Stephens (2006) has reviewed some of the literature with a focus on primary education where relational thinking shows up in being able to see the equal sign not only in an operational sense. In secondary education, relational understanding also links to understanding the meaning of the equal sign (Bardini et al., 2013). For the present paper, relational thinking is understood as the thinking that relates several quantities. The aspect that equations are relations or restrictions on values of variables is also elaborated in Drijvers (2011).

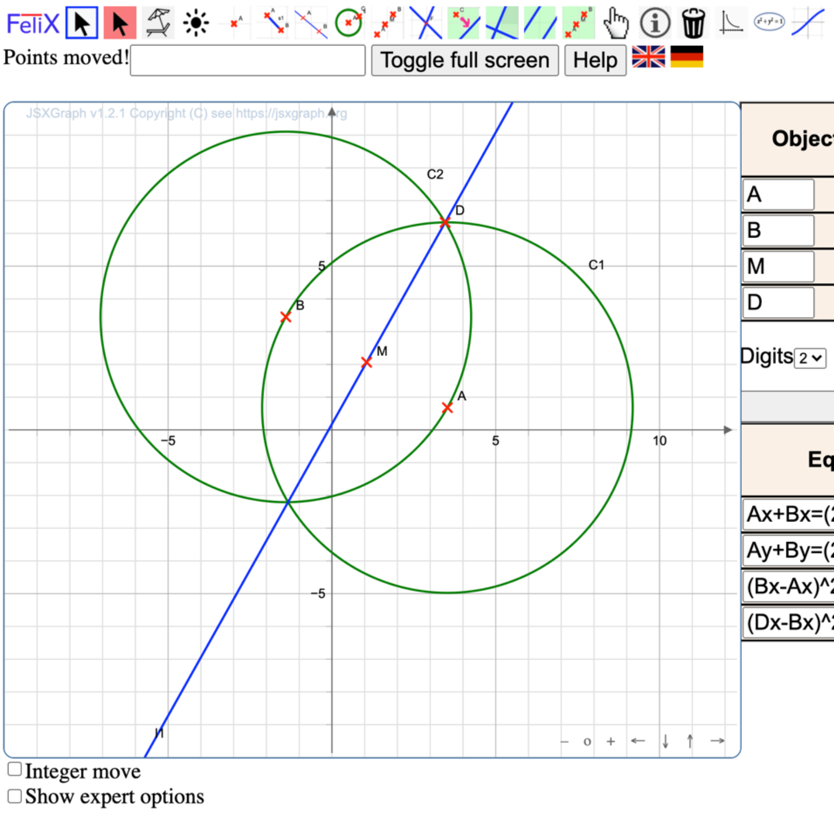
Given this background, the project described here addresses the following research questions:

1. Is it possible to design a consistent software environment that supports relational activities?
2. Is it possible to integrate relational and functional modelling tools?
3. Do examples of tasks exist that exhibit the power of relational thinking for modelling?

The methodology to answer all three questions is constructive existence proof. It is thus a theoretical and empirical paper where the empirical part consists of computer implementation experiments. Therefore, data collection is reduced to observing the behaviour of the implementation. Based on old ideas (Oldenburg, 2007), a new system (Oldenburg, 2021) called FeliX has been developed. The present paper describes the implementation and the possibilities it opens up to integrate relational and functional thinking. The paper first gives a short overview of the system from a user's point of view, then shortly explains the design choices. A short technical part explains central ideas of the implementation. The paper concludes with a small kaleidoscope of applications.

FELIX FROM THE USER'S PERSPECTIVE

The user interface of FeliX consists of three main components: A geometry view that shows part of the Euclidean plane with a Cartesian coordinate system (cf. Fig. 1). Points and other elementary geometric objects can be created, and points can be dragged. The second component is a table that shows all points and their current coordinates. Points may be moved with the mouse or by entering new coordinates in the coordinate table, i.e., these representations are bi-directionally linked. The last and most important component is an equation table that may in fact take equations, inequalities and expressions that are built up from the coordinate variables of the points.



Object	Value	🔒	👁
A	[3.53,0.68]	☐	☑
B	[-1.41,3.45]	☐	☑
M	[1.06,2.06]	☐	☑
D	[3.46,6.34]	☐	☑

Digits 2

New equation or expression

Equation/expression	Defect/Value	Valid
Ax+Bx=(2*Mx)	0	☑
Ay+By=(2*My)	0	☑
(Bx-Ax)^2+(By-Ay)^2=((Dx-Ax)^2+(	0	☑
(Dx-Bx)^2+(Dy-By)^2=((Ax-Bx)^2+	0	☑

Figure 1. The main components of the FeliX window are a Euclidean plane, a table of points and a table of equations/inequalities and expressions

If there is a point P, then its Cartesian coordinates are Px and Py. The equations and inequalities are respected while dragging. Assume one has three points A, B, C and enters the equations 2*Bx=Ax+Cx and 2*By=Ay+Cy then B will be the midpoint of A and C. Still, all three points can be dragged with the mouse. In fact, if one of the points is dragged the system has some freedom to adjust the other two points so that the equations are fulfilled. To make the behaviour more deterministic, one may set any of the points (temporarily) fixed using a checkbox in the object table. Such fixed points can still be dragged, but they will not move when the system tries to fulfil the equations. If, e.g., B is fixed, then A or C can be moved, and the other one follows as a reflection of the moved point. Equations and inequalities may be nonlinear and involve all usual mathematical functions such as the absolute value abs, trigonometric functions sin, cos, tan, exponential and logarithms and much more. There are helper functions to calculate distances, angles and lengths of segments. Moreover, some common equations (e.g., orthogonality or parallelism) may be set by using a button for convenience: This results in the appropriate equations being inserted just if they were entered by the user directly. Of course, all equations can be modified to explore the meaning of these algebraic relations. They can be set valid or invalid to explore their meaning. If the user enters contradictory equations, such as Ax=1 and Ax=3 then the system will show red defect values for those equations that cannot be satisfied. In this particular case, Ax would take the value 2 with defect 1 for both equations.

Figure 2 illustrates how FeliX can be used to model the classic sliding ladder problem. A "ladder" of length 8 with endpoints A and B is modelled by the equations Ax=0 (i.e., endpoint A is on the y-axis as a wall), By=0 (i.e., the other endpoint is on the x-axis as the ground). The formula for the length of the ladder was entered as Len(s1)=8 which FeliX automatically expands to the Pythagorean form given in Figure 2. Next, the midpoint M of the ladder was constructed. It can be moved with the mouse only on a curve, and, using Groebner basic methods, FeliX can calculate the equation of this curve and plot it. It is a circle which is optically highlighted by moving D to its centre (but D is not necessary in this example to create the circle!). Such graphs of curves that result from constraints on the freedom of a point are called "relation graphs" in FeliX language.

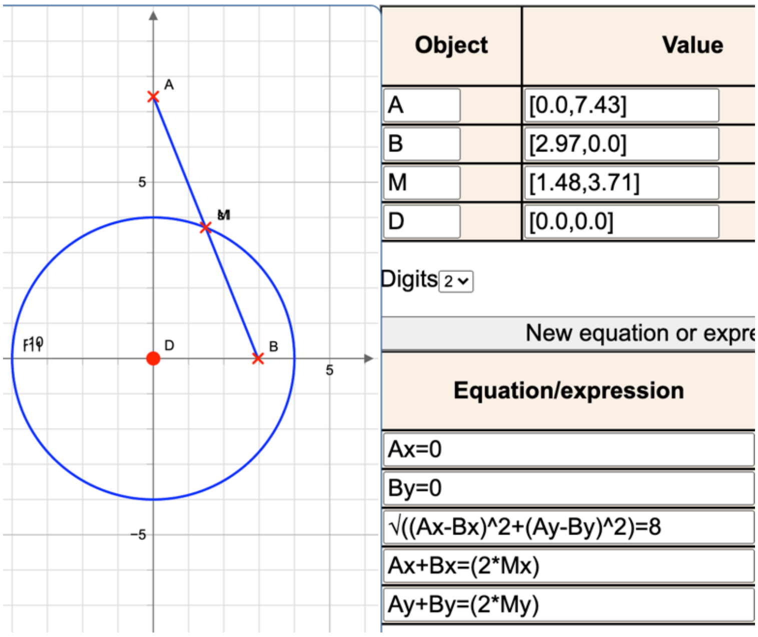


Figure 2. The sliding ladder problem

One could further construct an orthogonal line to the segment that passes through D. To do this, one has to construct a further point F, construct the line through D and F and enter the equations Dx=0, Dy=0, (Fx-Dx)*(Bx-Ax)+(Fy-Dy)*(By-Ay)=0. The last equations can also be set by using the green "declare orthogonal" button. Then the intersection of the line and the segment follows an algebraic curve of degree 6, which can be calculated but not yet plotted.

Figure 3 presents two other simple problems where the calculation of curves is interesting. On the left-hand side is an ellipse constructed from the defining equation that point C is on the ellipse if the distances to focal points A and B sum up to a fixed number. This fixed number is Dx in this case, so moving D deforms the ellipse.

The second example in Figure 3 starts from two segments that are set orthogonal. FeliX can calculate the curve that C can move on easily. As in Figure 2, the midpoint D of A and B is obsolete but was included in the figure to optically underpin that it is the centre of the Thales circle.

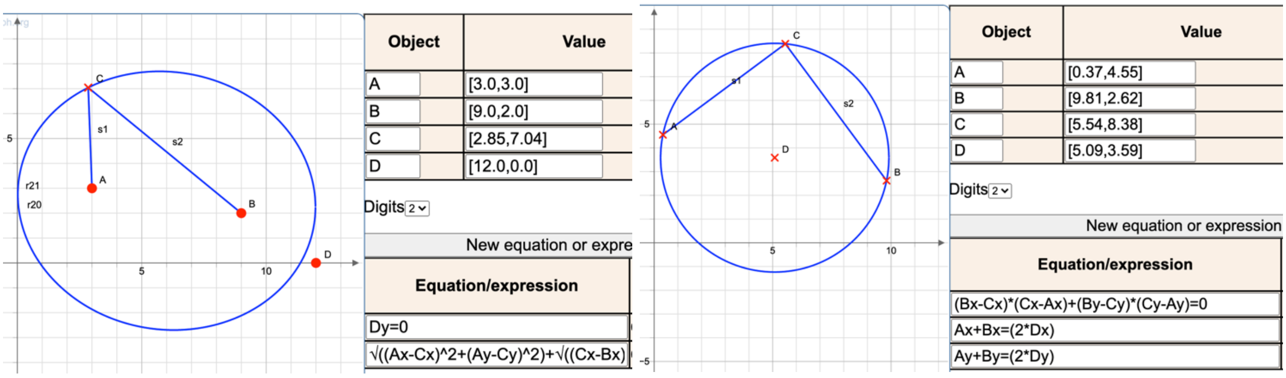


Figure 3. An ellipse from Len(s1)+Len(s2)=Dx and a Thales circle constructed from setting s1 and s2 orthogonal.

A feature not touched upon in the examples so far is that FeliX can handle inequalities. For example, one may enforce that two points always have at least a distance of 1 by entering (Ax-Bx)^2+(Ay-By)^2>1. This results in a construction where one of these points may be used to push the other around if they would come too close to each other.

Yet another feature is the ability to draw function graphs from expressions that are allowed to involve all other objects' variables. Using the "bind to" tool, points can be bound to lines, segments, circles, function graphs or relation graphs.

All these functionalities add up to a system that is very flexible in modelling geometric configurations and exploring the meaning of a wide range of algebraic relations.

DESIGN CHOICES

The design of FeliX follows from some very basic principles that have motivations both in mathematics and in didactics:

- Full Information: All information that governs the behaviour is fully visible on the screen. This is in contrast, e.g., with spreadsheets where one usually sees only the values in the cells, not at the same time the formulas (or, in formula view, vice versa).
- Based on concepts from mathematical logic: Object tables with the values of coordinates are essentially interpretations in the sense of mathematical logic (e.g., Hamilton, 1988): At each time, they assign numerical variables to all variables in such a way that, if possible, this is a model of the equations, i.e., that they are all fulfilled.
- Object creation and imposing relations between objects are independent operations. This allows a step-by-step specification so that the user can symbolize its knowledge about the situation dynamically as it evolves.
- Everything can be changed at all times. The order in which objects or equations are created has no impact on the behaviour. For example, the equations in the equation table are logically

connected by the "and" operator, which is commutative—any order of creation or order of imposing relations thus leads to equivalent configurations. Thus, no user will ever be caught in a deadlock because (s)he took a wrong decision at some time of the "construction" process.
- There is no restriction on the type of operations and functions that can be used.

It is interesting to compare these characteristics to those of dynamic geometry systems such as GeoGebra which differ much in all these points. Focusing on geometrical aspects, one may realize that the functional dynamic geometry is very different from the relational geometry described here. For example, to construct a triangle with sides 3,4,5 in a dynamic geometry system, one has to come up with a construction (which is a real problem for students that encounter it the first time), while, in relational geometry, one simply specifies the parts, removing *this* occasion for problem solving. From a pedagogical point of view, one may wonder what kind of geometry is more important to master. As compass and ruler constructions dominate in schools, functional geometry is more important there. However, in the professional world of computer-aided design systems such as AutoCAD and FreeCAD, the relational approach is dominant.

IMPLEMENTATION

The goal of FeliX's design is to have a system that has a clear and mathematically-defined semantics. Thus, it is basically an interface to a numerical constraint optimization algorithm. If, e.g., a point P shall be dragged to coordinates (x_0, y_0) then the expression $(P_x - x_0)^2 + \left(P_y - y_0\right)^2$ is minimized subject to the constraints given by the equations and inequalities. Coordinate values of fixed objects are inserted in the constraints, of course. At each point, the current configuration is used as the starting point for the search for a new solution. This approach is rather simple, but some caveats are in place: The solution process is done by a numerical algorithm (Powell, 1998) and hence may fail to find a solution, or there may be some noticeable inaccuracy. Moreover, some artefacts may result from this design. If a point is bound to a line that is fixed because some fixed points lie on it, then dragging the point is restricted in the sense that it will stay at the line and move to that point on the line that is closest to the mouse position. It may be the only sensible behaviour in this case.

Often there are many solutions, but only one will be found and realized. The equation that states that the lines through A,B and C,D are orthogonal leads to a scalar product equation that is not only fulfilled if the lines are orthogonal but also if A=B or C=D. Hence, such a degenerate solution may be found and realized by FeliX. There are two ways out: One may add an inequation that says that the points shall have some minimal distance. The other way is to use the "shake button" that randomly moves points and will often get one out of degenerate solutions.

Another issue is that of points going to infinity: Consider the lines through A,B and through C,D and construct the intersection point F. Set A,B,C fix and move D around C. The intersection F should move to infinity and come back from the opposite side. This may bring the solver into trouble. There is a more powerful move tool that sets the coordinates of the moved points and searches then for a solution. With this move tool, one may move through such degenerate situations.

The approach further implies a kind of existential quantification. If one has two circles and an intersection point, one cannot move the circles so far apart that they no longer intersect. Dragging mode will stop when they are tangent at their intersection point. Moreover, an equation like Ax=sqrt(Ay) will, as a side-effect, constrain A to the first quadrant of the coordinate system.

In contrast to the dynamic of moving objects, the calculation of relation curves for a restricted point P is done symbolically by calculating lexicographic Groebner bases (Cox et al., 2005) and eliminating non-fix auxiliary variables. In the resulting equation, only variables of fixed points and of the generating point P occur. The method of Groebner bases is suited only for polynomial equations. Thus, FeliX eliminates in a preparation step subexpression $a^{\frac{n}{m}}, n, m \in \mathbb{N}$ and replaces them with v^n where v is a new variable, and it adds the equations $v \geq 0 \wedge v^m = a$.

Taking these elements together, research question 1 is answered positively by this constructive proof of existence.

EXAMPLES OF OPTIMIZATIONS

FeliX has a powerful feature that is almost invisible from its user interface: The checkbox to set equations either valid or invalid (an invalid equation will be ignored while dragging) is also in place for expressions. In contrast to equations, expressions are set to invalid by default. Setting them to valid means that they will be minimized! This gives some more modelling possibilities and combines relational and functional thinking, as the following examples will show.

Figure 4 illustrates a possibility to explore the optimality of certain figures. Five points are created and connected by segments s1,s2,s3,s4,s5 to form a polygon. The user entered the equation abs(polyArea([A,B,C,D,E]))=50 which is expanded by FeliX using the Gauss formula for the area of a polygon. It is then interesting to move around the points and see how flexible a polygon with fixed area of 50 is. Entering Len(s1)+Len(s2)+Len(s3)+Len(s4)+Len(s5) as an expression displays the circumference, e.g., the polygon on the left in Figure 4 has area 50 and length 32.48. Setting the expression for the circumference valid, i.e., minimizing its value, immediately moves the points to form the shape on the right-hand side of Figure 4. Similar investigations can be undertaken, e.g., to find shapes that have maximal area under fixed circumference or to solve other optimization problems.

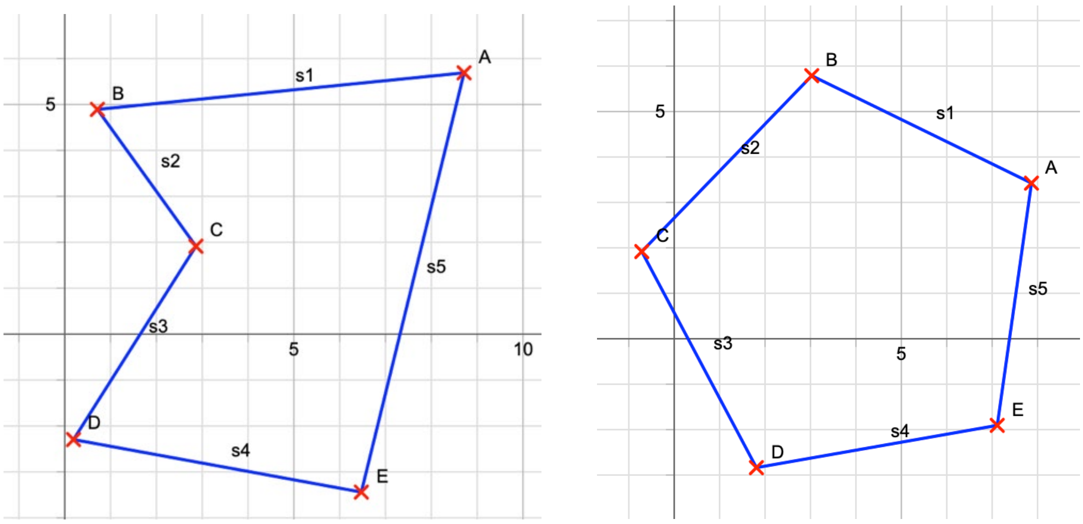


Figure 4. One click transforms the polygon of area 50 to one with the same area but minimal circumference

Figure 5 shows a discrete hanging chain. Points A and E are fixed (recall that they still can be moved explicitly with the mouse), and B,C,D are unrestricted points. The segments s1,s2,s3,s4 between AB, BC, CD, DE are all set to have length 3. Their midpoints F,G,H,I are constructed, and the expression Fy+Gy+Hy+Iy that corresponds to the potential energy of the chain is minimized. It is interesting how natural the chain behaves when, e.g., E is raised further or moved horizontally.

Another example shown in Figure 5 is a shortest-path problem which is also classic: The Fermat point is that point P inside a tringle ABC that minimizes the sum of the distances AP+BP+CP. FeliX allows students to observe the fact that at the optimal point, the three segments to A,B,C form angles of 120°. Of course, this is not yet the solution, but it may give hints to come up with a theoretical solution.

Both problems support a combination of functional and relational thinking. They combine the variability of a function value (which is minimized) with the invariance of a relation, which is preserved. The existence of these examples answers research questions 2 and 3 positively.

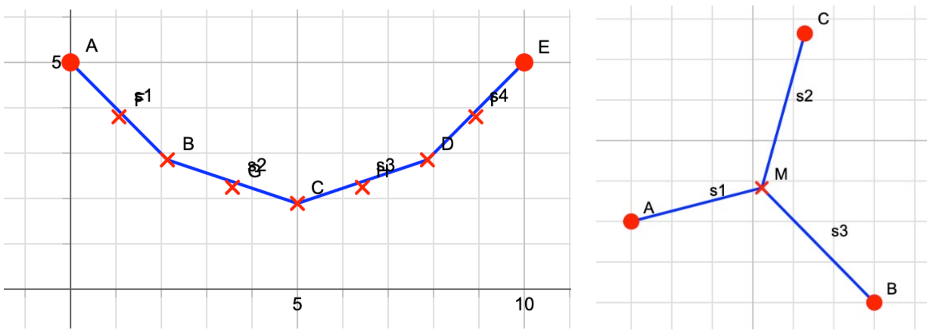


Figure 5. Discrete hanging chain and Fermat Point of a triangle

DISCUSSION AND OUTLOOK

The use of the old FeliX system 15 years ago was very limited because it was extremely difficult to install, but with the new web-based approach presented here accessibility of FeliX shall no longer be a problem. There are quite a number of areas were the work with FeliX might open promising perspectives. This final section will discuss some of these. Implementing the use of FeliX in schools is a complex task. First, teachers need to get an idea about what FeliX is and why its use could be rewarding. Next, one needs concrete ideas of how to introduce the system and what topics to use it for.

As a first contact with FeliX the problem of finding the midpoint of two points is very rewarding, because it exhibits most of the semantics of FeliX. Moreover, the equations that one needs are very easy and can either be entered by the students or created using the convenient tools. To explore und understand the midpoint relation it is useful to use FeliX's option "integer move". When this is activated, the dragged point moves discontinuously jumping only to points with integer coordinates. This eases mental calculations to check and understand what the system does. Moreover, modifying the midpoint equations is an interesting problem: How to get the 1:2 point that divides AB in this ratio? From there, one can, in principle, go on to find a form of the equation of a line.

A next activity may be to investigate one-dimensional equations. For example, one sets two points, A,B on the x-axis by Ay=0, By=0, and then relates A and B, e.g. by Ax+2*Bx=12 or by Ax*Bx=100. The same strategy can also be applied to equations that relate more than two variables, e.g., the "lens equation": When a lens forms a sharp image of an object, then there holds the relation $\frac{1}{f} = \frac{1}{a} + \frac{1}{b}$ between the focal length of the lens f and the distances a, b between then lens and object and between the lens and image. This interesting example has been used by Drijvers (2006) to illustrate the many roles variables can take when a computer algebra system is used, but the same consideration applies here: Setting a point fixed, e.g., turns it from a variable into a parameter.

Another field that can be explored with FeliX consists of the many relations between the various forms of quadrilinears. One may start with a general quadrilinear, possibly with its diagonals, and then may impose more and more relations and explore how rigid it becomes. As a last area of applications, the large field of mechanical linkages shall be mentioned. FeliX both provides easy ways to model and simulate them and to calculate relation curves of the movement of certain objects.

Of course, plenty of research lies ahead. One may ask if the experience with equations in this relational sense enhances students' performance on reversal error tasks (Rosnick & Clement, 1980).

ACKNOWLEDGEMENTS

It is a pleasure to thank Bernard Parisse and Luka Marohnić for much help on the giac computer algebra system and Alfred Wassermann for help with the jsxgraph library.

REFERENCES

Bardini. C., Oldenburg, R., Pierce, R., & Stacey, K. (2013). Technology prompts new understandings: The case of equality. *MERGA* 36, pp. 82–89.

Cox, D. A., Little, J., & O'Shea, D. (2005). *Using algebraic geometry*. Springer. https://doi.org/10.1007/b138611

Drijvers, P. (2006). Die variable Unbekannte. Facetten des Variablenbegriffs mit Computeralgebra erkunden [Include English title here]. *Mathematik lehren,* 136, 10–12.

Drijvers, P. (Ed.) (2011). *Secondary algebra education*. Sense Publisher. https://doi.org/10.1007/978-94-6091-334-1

Hamilton, A. G. (1988). *Logic for Mathematicians*. Cambridge University Press. https://doi.org/10.2307/2273209

Oldenburg, R. (2007). Algebraic modeling using a dynamically linked geometry and computer algebra system. In D. Pitta-Pantazi & G. Philippou (Eds.), *Proceedings of the fifth congress of the european society for research in mathematics education* (pp. 1480–1488). University of Cyprus.

Oldenburg, R. (2021, September). FeliX. https://myweb.rz.uni-augsburg.de/~oldenbre/jsfelix/F2d/jxfelix.html

Parisse, B. (2021, September). giac/Xcas 1.6.0-47 https://www-fourier.ujf-grenoble.fr/~parisse/giac.html

Powell, M. (1998). Direct search algorithms for optimization calculations. *Acta Numerica*, 7, 287–336.

Rosnick, P., & Clement, R. (1980). Learning without understanding: The effect of tutoring strategies on algebra misconceptions. *Journal of Mathematical Behaviour*, 3(1), 3–27.

Stephens, M. (2006). Describing and exploring the power of relational thinking. *MERGA* 29, Volume 2, pp. 479–486.

Wassermann, A. (2021, September). jsxgraph 1.2. https://jsxgraph.uni-bayreuth.de/wp/index.html

Weigand, H. G., McCallum, W., Menghini, M., Neubrand, M., Schubring, G., & Tobies, R. (2017). What is and what might be the legacy of Felix Klein? In G. Kaiser (Ed.), *Proceedings of the 13th International Congress on Mathematical Education*. (pp. 321–334) ICME-13 Monographs. Springer. https://doi.org/10.1007/978-3-319-62597-3_21