International Journal of Computer Discovered Mathematics (IJCDM)
ISSN 2367-7775 ©IJCDM
March 2016, Volume 1, No.1, pp.45-61.
Received 25 December 2015. Published on-line 1 January 2016
web: http://www.journal-1.eu/

The Geometry Program C.a.R.

René Grothmann
Catholic University of Eichstätt Germany
e-mail: rene.grothmann@ku.de
web: http://www.rene-grothmann.de

Abstract. C.a.R (Compass and Ruler) is an advanced geometry program similar to other programs for "dynamic geometry", but with numerous features going beyond that category of software systems. The aim of this paper is to outline the intentions of C.a.R., and to provide hints for the best use of this program.

Keywords. Dynamic Geometry.

Mathematics Subject Classification (2010). 97G40

1. Geometry

Doing geometry with compass and ruler looks out of date now, especially when done on paper as in the old days. Like handwriting and pencil arithmetic it is certainly a basic technique that has been important to our culture, but that has lost its meaning somewhat. This is undeniably true. Today, we use computer programs to draw, from simple function plotters to advanced CAD systems or realistic 3D rendering machines. But geometry is more than simple sketches.

There can be no doubt that visual imagination is so basic that it cannot be learned early enough. It is one of the three columns of mathematics. Almost all mathematicians use it all the time, even in the most complex settings. To make mathematics, we use images in one, two or three dimensions, graphs of functions, curves and surfaces, as well as graphical representations of relations. Points, lines and circles are only a starting point.

The second column of mathematics are algebraic skills, i.e., everything that can be computed, including calculus. Geometry trains these skills too, in the form of angles, distances, and cartesian coordinates.

But there is an important third column, and it is logic. Logic controls our doing, and determines right from wrong, valid from invalid, randomness from generality, closeness from exactness. The Greeks have put much logical effort into the investigation of geometry. In fact, Euclid's book was used more than the two thousand years to teach logic. The Greeks started discussions and raised problems about the feasibility of constructions, or about the nature of their geometrical world, solved only more than two thousand years later.

Geometry packs the three columns of mathematics together in one subject. It is about geometrical objects in our visual world, it is about algebraic relations between these objects, and it is about logical statements about our world. Thus it is a pity to see its decline as a subject in schools.

To some extend, geometry is also related to motoric skills. With a compass and ruler, you actually have to do the construction with your hands. Dynamic geometry, lets you drag points around on the screen. This kind of motion sensing may be important to experience mathematics.

2. Dynamic Geometry

The term "dynamic geometry" has been coined by "Geometer's Sketchpad" ([3]), and is a trademark of that software. The program calls itself a "visualization software" and does geometry of the plane or the space, including curves or other mathematical drawings. As most programs of this kind it tries to be the one and all solution for the needs of the working teacher. There is even an implementation for a hand held device.

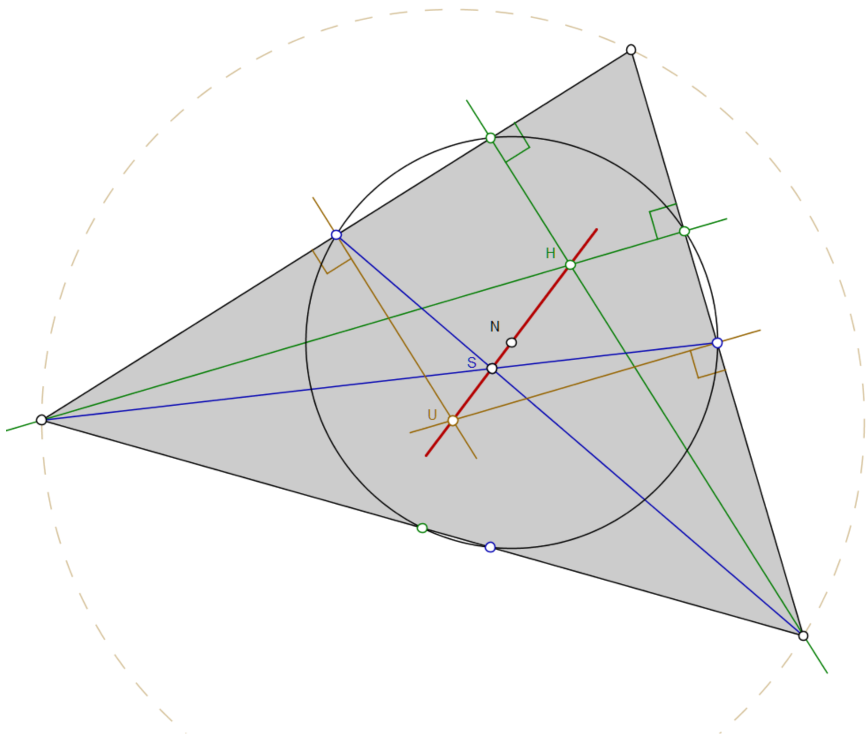


Figure 1. Dynamic Geometry

Figure 1: To explain the meaning of dynamic geometry, have a look at this construction. You see the famous Euler line in red passing through the intersection H of the heights and the center of the circle of circumference U, and you see the

Feuerbach circle around the midpoint N between H and U through the base point of one of the heights of the triangle. The circle passes through all other bases and through the midpoints of the sides, and some other points not show here. The center of gravity S is on Euler's line.

Is this by chance or is it a theorem? For a test, we can drag the corners of the triangle to some other position. This is shown in Figure 2.

The main feature of dynamic geometry is that we we able to drag one of the basic points of a construction, and see construction follow the this change. In our case, we can convince ourselves that the point S is "always" on the segment UH.

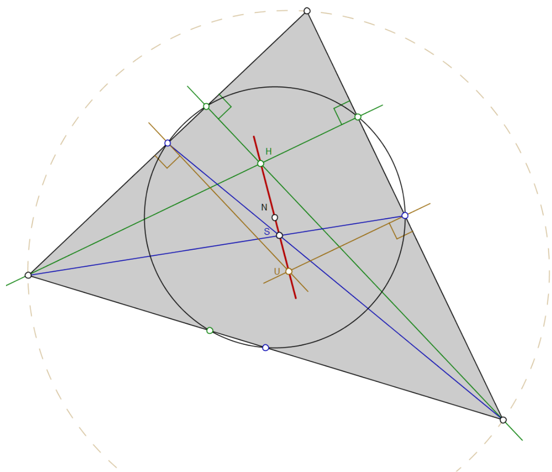


Figure 2. One point of the triangle at another position.

There are now dozens, if not hundreds, of geometry programs of the this kind. One important player is Geogebra ([4]), which also tries to collect a community of teachers around the software, providing and benefitting from fully prepared worksheets for the classroom. Some programs are free of costs, some are available on multiple platforms, including modern mobile devices. Others are more closed solutions.

There have been efforts to create file formats to exchange constructions between the different systems. The "Intergeo" project ([5]) was such an attempt. It also tried to set up a data server for the exchange of constructions and ideas. To some part, this has been a success.

3. C.a.R.

C.a.R. (Compass and Ruler, [1]) is a geometry program written by the author of this paper. The starting point of C.a.R. was to provide a portable Software using the Java language with the ability to easily present constructions on web pages. Another aim was to add features to dynamic geometry which were not available in other packages.

The user interface of C.a.R. is a simple interface with icons for construction steps and settings, a list of construction objects, and the main construction window (see Figure 3).

For the user, constructing consists of the following two steps:

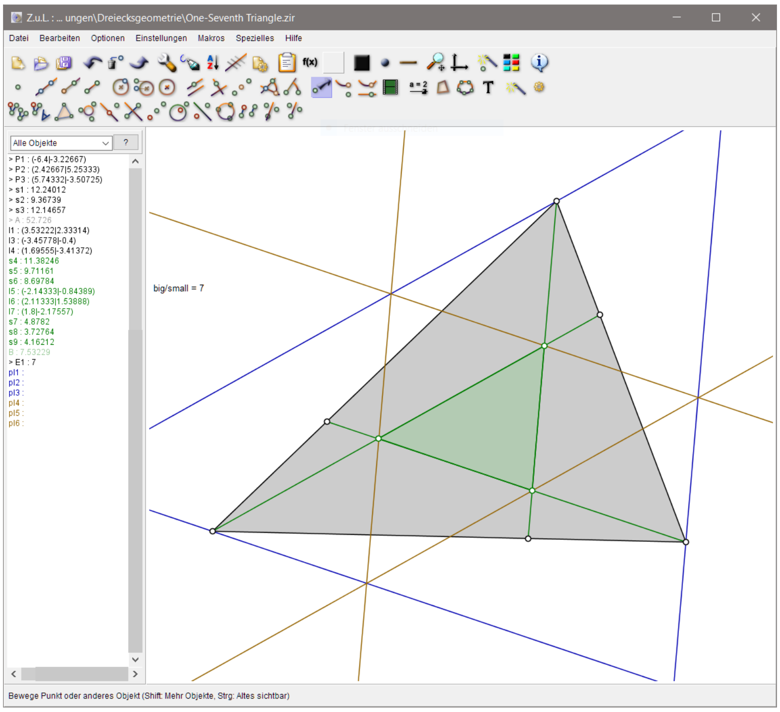


FIGURE 3. C.a.R. GUI

- Select a construction tool.
- Repeatedly apply the tool to the construction.

Applying a tool means to select the objects necessary to generate the new object one by one. The result will appear after the last parameter has been selected, and in a preview version before the last parameter is selected. E.g., the user selects the segment tool, and then a line segment will be generated after clicking two points.

Points are, by default, generated automatically as free points, as intersections of objects, or as points fixed to line objects, depending on where the user clicked. If objects have more than one intersection, the intersection closest to the clicked spot will be generated. The automatic creation of points as well as the selection of the tool before the selection of the parameters were unique to C.a.R. Now, almost all program use this natural method.

The program has been around since the times of the Atari ST albeit in a very different version. The first Java version of C.a.R. was written when the first version of Java came out in 1996. Until 2010, the program was developed to the current state. Around 2009, a new version of C.a.R. called CarMetal ([2]) was developed by Eric Hakenholz in France. The main difference are the non-modal settings for objects, yielding a different handling and a different outline of the interface. Both programs save constructions in a human readable XML format, but are not completely compatible.

One of the purposes of this paper is to give an overview of the features of C.a.R. Furthermore, we want to give hints on possible uses of C.a.R. for geometrical and for general mathematical education.

4. Tracks

As a first example of things that can be done with C.a.R., let us start with a question:

What is the set of points C such that

$$\mathrm{CA} = 2\,\mathrm{CB} \tag{1}$$

for two fixed points A and B?

Students learn that the set of points with CA = 2 would be a circle, and in an advanced class they would learn that the set of points with

$$\mathrm{CA} + \mathrm{CB} = 2$$

would be an ellipse with foci A and B. After this, (1) should appear as an obvious generalization. It is the kind of question that I would like to hear from students. It would be wonderful to have a classroom where such questions came up.

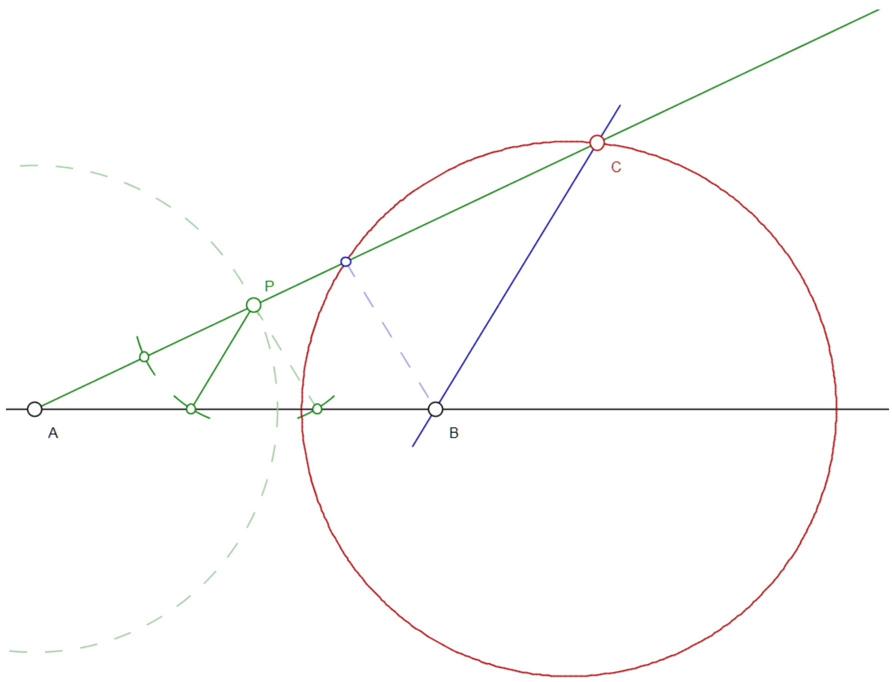


Figure 4. Circle of C with CA = 2 CB

Figure 4: C.a.R. can generate the track of the points C automatically in various ways. The following is one possible construction.

- We generate some fixed circle (the green dashed circle) around A and put a point P on it. It is then possible to move P only on the circle. P is said to be bound to the circle.
- Then we use a similarity construction to find C such that AC = 2 CB. Note that there are two solutions, both are on the red circle. The similarity construction is a challenge in itself.
- Finally, we let C.a.R. compute the track of points C, when P moves on the green dashed circle. Surprisingly, C moves along a circle.

How does C.a.R. produce the red circle? In C.a.R. and in other programs, this can also be done manually. For this, the point P must be moved using the mouse while C is marked to be tracked. If we do that we only get the right side of the red circle.

The automatic track feature of C.a.R does the following:

- It moves the point P into one direction until as long as possible.
- If the construction becomes invalid (in our case: the circle around P with radius AP/2 does no longer intersect the line AB) the movement is reversed and another branch of intersections is taken (in our case: the other intersection of the circle around P with the line).

This works even for very weird constructions. The geometry software Cindarella ([6]) does the same. It uses complex coefficients and different paths in the complex space to achieve the same result. Using complex spaces feels unnatural in plane geometry and has strange side effects. So C.a.R. avoids this.

If we want to generate the complete track manually we can do so in C.a.R. in the following way:

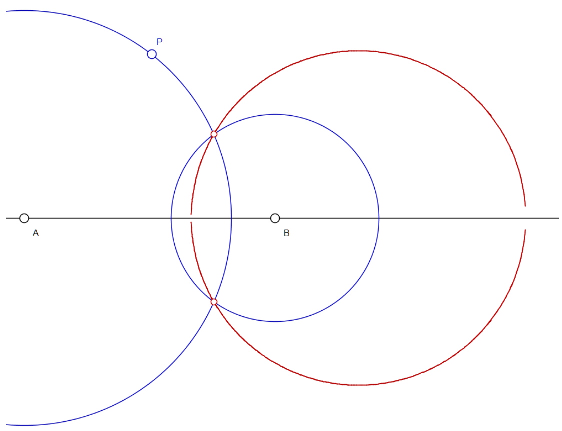


FIGURE 5. Circle of points with CA = 2 CB

- Generate a point P and a circle around A with radius AP.
- Generate a fixed circle around B and set its radius to $k/2$, where k is the name of the circle around A. (See below for expressions in C.a.R.) Alternatively, construct the circle with half the radius in a traditional way.
- Generate both intersections, and select them for the manual track tool. Move the point P properly to generate the tracks of both points.

Let us talk about proofs. Once we see the the circle, we should want to know why it has the desired property.

There are geometric proofs for the fact that the track is a circle. It is possible to use similarity, or to use results about the reflection with respect to a circle. However, in this case an analytic proof is easier. We want

$$(x - x_A)^2 + (y - y_A)^2 = 4\left((x - x_B)^2 + (y - y_B)^2\right)$$

Expanding and collecting the terms, we see that the equation is a quadratic function with equal coefficients for x^2 and y^2, and thus the solutions describe a circle.

These kind of proofs can generate a vivid discussion weather geometric or analytic proofs are better. But for theorems involving angles, almost always the geometric proof is much easier. An example is the chord theorem which states that the angle from each point to a fixed cord of the circle is always the same.

A polar set of a system of lines is a track such that the lines are tangent to the track. E.g., it is well known that the polar set of the middle perpendiculars of a fixed point and another point running on a line is a parabola.

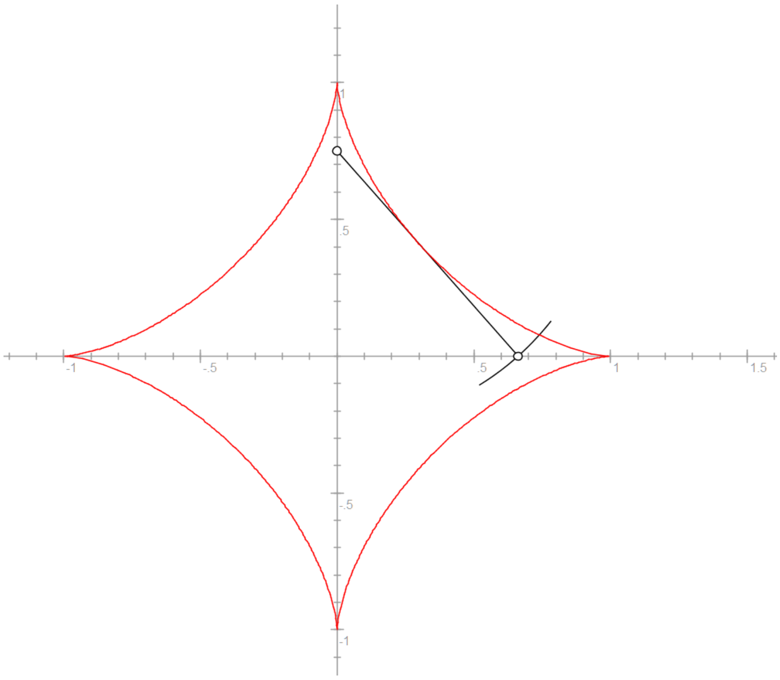


Figure 6. Polar set of the Leaning Ladder

Figure 6: This construction shows another example. The moving point runs on the vertical axis of the figure, and the black line has length 1. The figure is called the leaning ladder by obvious reasons. The curve is called the astroid. It is a nice task to compute the equation of the astroid. Using the computed expressions in the following section it would be possible to check the derived equation by comparing it to the track.

C.a.R. can do very complicated tracks simulating physical hinges and other physical systems. Using the computed constructions in the next section and the advanced objects, automatic tracks are a mighty feature.

Figure 7: For an example simulating a system of hinges, see this construction. The three segments have a fixed length, the centers of the circles are fixed, and the red point is the midpoint of the middle segment. It may be difficult to imagine the movement of the system. But C.a.R. animates the movement changing the direction of the turn and the intersections along the way to get the full 8-shaped track.

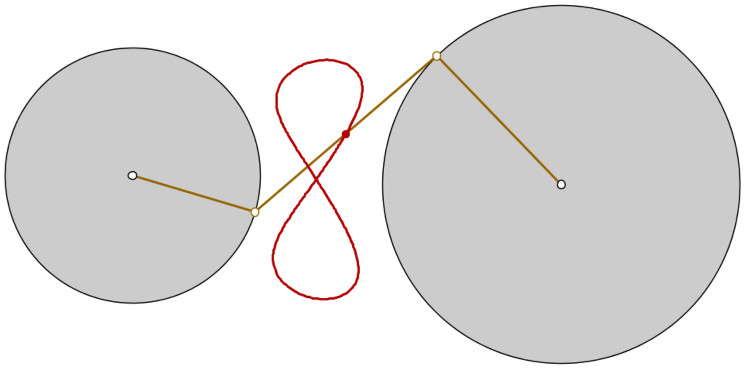


FIGURE 7. System of Hinges

Clearly, the equation for this track is not easy to derive. Using the complex plane, we need to solve

$$|z + re^{i\phi} - a| = r_1, \quad |z - re^{i\phi} - b| = r_2$$

for z and ϕ, given a, b, r, r_1 and r_2. After some transformations, this is essentially the same as

$$|1 - w_1| = \rho_1, \quad |1 - w_2| = \rho_2, \quad |w_1 - w_2| = \rho.$$

Though this is simply the construction of a triangle, it is not easy to derive any information on the track in the image from it. Trying to derive an algebraic formula for the track is even harder.

5. EXPRESSIONS, FUNCTIONS AND CURVES

Sooner or later the user of any geometry program will wish to do analytic geometry, or at least be able to enter and plot functions and curves given by an expression. The geometry program Geogebra even shifted the main emphasis from geometry to analytic constructions. It contains a tool for simple derivatives and integrals. The author of this paper prefers a true algebraic or numerical software for this purpose, like his program Euler Math Toolbox ([7]).

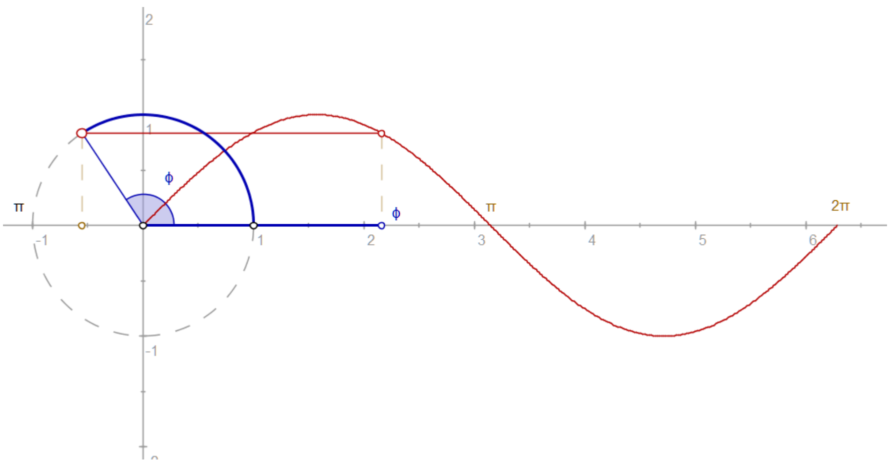


FIGURE 8. Demonstration of the Sine Function

Figure 8: This demonstration is an example for the mixture of geometrical and analytical content that is often required. The user can turn the red dot along the circle changing the blue angle ϕ. The blue arc and the blue line have equal lengths which is a transcendental transformation. The construction is not possible with compass and ruler.

The red graph could be a track of the small red dot. But it is a curve. This is much easier to handle for the program, and very easy for the user too. C.a.R. has the common functions of mathematics, including numerical integration and differentiation, and some special functions for geometrical constructions, like a "what if" simulation.

C.a.R. remains a geometric program still. The coordinates are always rectangular, squares are squares, and circles remain circles.

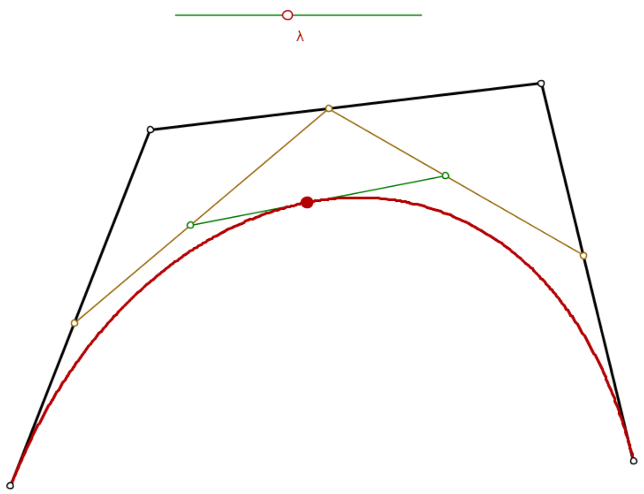


Figure 9. Construction of a Bezier Curve

Figure 9; The Bezier curve in this construction is done with a curve with parameter functions which depend on the four black points.

The construction of a Bezier curve can be achieved by subsequent partition of the segments connecting the support points. The same factor λ has to be used in all steps. In the figure, λ can be selected by the user with a slider. C.a.R. has special control sliders for this purpose which allow the input of numerical values. But, of course, a simple construction can be used instead of an expression slider.

If you look at this construction, you see that a track could have been used for the Bezier curve instead of the parameterized curve.

Figure 10: C.a.R. can also draw zero sets of functions $f(x, y)$.

$$Z = \{(x, y) : f(x, y) = 0\}.$$

In the image, we select the set Z of points P $= (x, y)$ with

$$\mathrm{AP} \cdot \mathrm{BP} = \mathrm{const.}$$

The user can drag the point P and the set is immediately re-computed accordingly. Note that the set consists of two branches if P gets close to A or B.

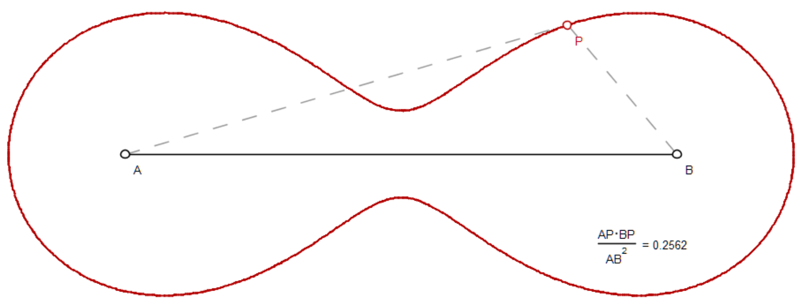


FIGURE 10. Set of all points with AP · BP = const

As is easily seen, this set is an algebraic curve of degree 4. It is again possible to generate it as a track using the same construction as in Figure 5. To construct numbers with fixed products, the Thales circle is one approach.

But in general it is not possible to construct the two or four points on the track which are on a given line through A. E.g., if we set

$$A = [0,0], \quad B = [0,1], \quad P = \left(x, \frac{x}{2}\right), \quad \text{const} = 2,$$

then we have to solve

$$\frac{25\,x^4}{16} - \frac{5\,x^3}{2} + \frac{5\,x^2}{4} = 4.$$

This cannot be solved with compass and ruler. The solution requires to take cube roots.

This kind of problems show the restrictions of constructions with compass and ruler. Moreover, it becomes obvious that the track feature of C.a.R. goes beyond classical geometric constructions.

6. MACROS

Macros not only enhance the capabilities of C.a.R., making very complex constructions possible, but also educate geometrical thinking. A macro is a construction that is done automatically from given parameter objects, usually points, generating other objects depending on these parameters. A very simple example is the middle perpendicular of two points.

To create a macro, the following steps are necessary.

- The constructions has to be done once.
- The parameter objects have to be selected. These are all objects that the macro needs to complete the construction.
- The target objects have to chosen. Only the objects which can be constructed from the selected parameters can be chosen at this stage in C.a.R.

The macro is saved under a name, and can optionally be included in the macro bar with an icon. It can contain an explaining text. Moreover, some parameters can be fixed by name. To run the macros, the user only needs to select argument objects of the correct type.

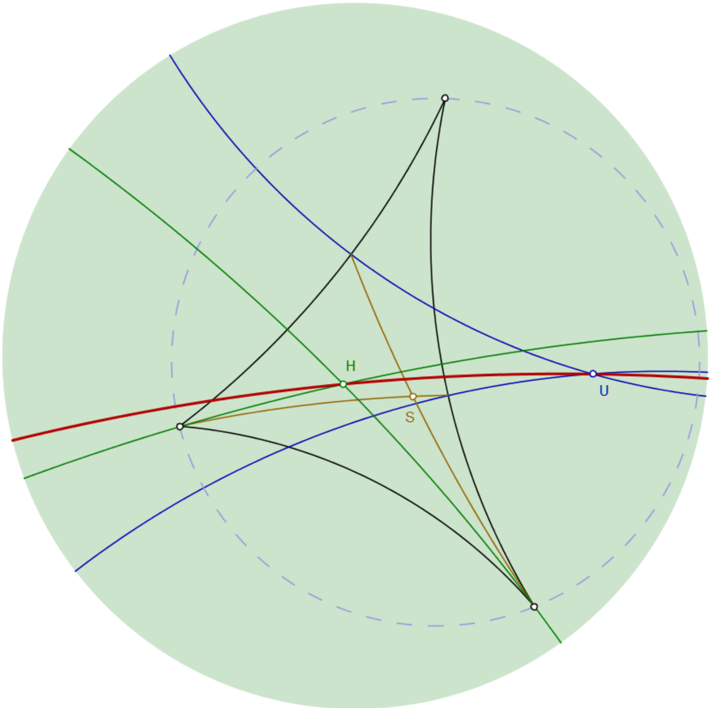


Figure 11. Euler's line in the Poincaré geometry

Figure 11: For an advanced example, macros for non-euclidean geometries are included in C.a.R. In the construction, it is shown that Euler's line does not work in the Poincaré geometry. This geometry is one model of a non-Euclidean geometry where the sum of the angles in a triangle is less than 180 degrees.

Other advanced sets of macros allow constructions in 3D. There are specialized programs for geometry in the space, like Cabri 3D ([8]) and Archimedes Geo3D ([9]). The constructions in C.a.R. are done in the plane and the coordinates are projected from the space to the plane. Special features of C.a.R. allow hidden lines in the case of simple spacial objects. Figure 12 shows an example. We consider this as a very unusual use of C.a.R.

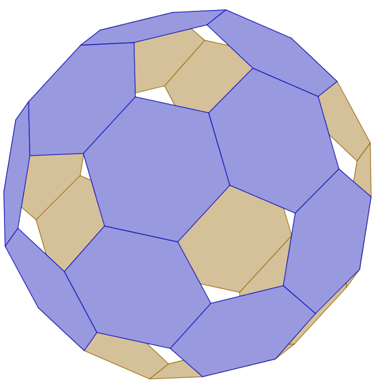


Figure 12. 3D in C.a.R.

Other, more useful macro sets contain functions for Bezier curves and other curves. It may be tempting to use C.a.R. as a tool for computer aided geometry (CAD). But this is not intended. Specialized programs have much more features and are easier to handle.

However, macros visualizing mathematical theories like projective geometry can be very useful. In fact, just to generate these macros can be very educative. In this matter, C.a.R. can be a tool to learn about dependencies and functions in geometry.

C.a.R. does also contain a language to describe geometric constructions. In this language, the description of macros looks like a function with parameters and results.

In C.a.R., constructions in macros can contain expressions. When the macros is applied to arguments, the objects referred in the expressions are replaced by the proper objects in the argument list. This allows very complex macros.

7. Advanced Objects

Like most programs, C.a.R. can generate quadratic curves from 5 points, or alternatively from less points and tangential lines, or with a quadratic formula. Figure 13 is a simple demonstration showing the theorem of Pascal. Of course, with simple projective arguments it is possible to reduce the problem to the case of a circle. But in general, quadratic curves appear so often that they deserve a special object.

Drawing a quadratic curve in the plane is done automatically and fast by C.a.R. in two parts by solving the equation for one variable. Other algebraic curves require more effort. See below for expressions, functions and curves.

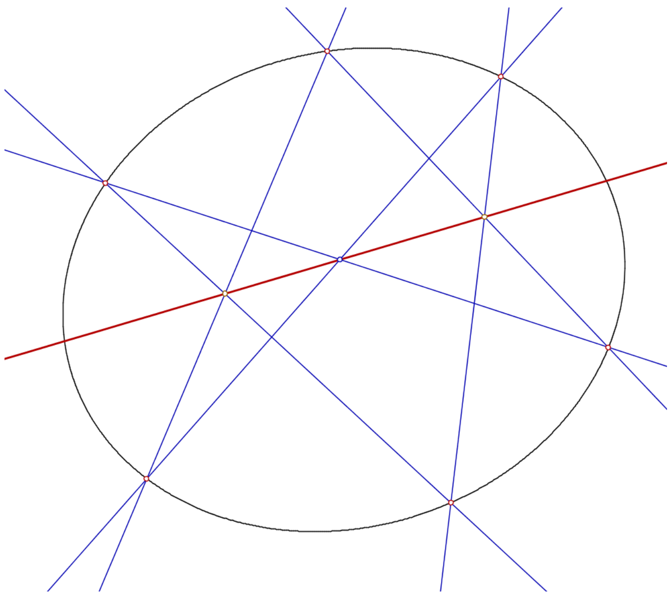


FIGURE 13. Pascal's theorem

It is interesting to learn how the program fixes points on curves. The objects in C.a.R. have a projection method which projects from a given mouse position to the object. In the case of circles or lines this is easy. For quadratic curves, however, the Lagrange method yields several longish solutions, which are computed and compared. The formulas for these solutions have been derived with an Algebra software. Curves given by more complex functions are treated as a sequence of line segments.

It is even more interesting to study the way that intersections of curves are computed. Since there is no available derivative, the Newton method is approximated by subsequent projection. A point close to the intersection is projected to one curve, then the result is projected to the other curve. From the projection paths tangential lines can be derived and the intersection of these lines is used to find the next approximation. This works quite well and the result depends as continuously as possible with numerical methods from the starting point, usually the mouse position.

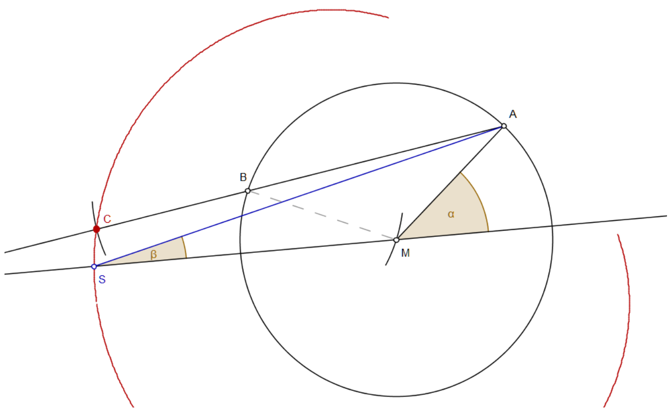


Figure 14. Sliding Trisection

Figure 14: The trisection in this construction is a another example of a construction out of the scope of circle and ruler constructions. It is a famous construction by the Greek mathematicians called a Neusis construction.

- The red track is the set of all points C such that MA = MB = MC, and C is on the line through A and B. This is not a circle! If it were a circle, we could construct the trisection with compass and ruler.
- The point S is the intersection with the track and the line, which passes through M and has the angle α to MA. Then it is easy to show that $3\beta = \alpha$.

If we want to use physical tools we need to mark the distance MA on a ruler and slide the ruler so that it passes through A and the two marks are on the circle and the line.

C.a.R. has also an iterative way to do this construction with an object referring to itself recursively. We do not want to go into the details of such strange constructions here.

Figure 15: The author often gets letters with "trisections". C.a.R. can then be used to check the error of the proposed construction. In the figure, you see an idea that is motivated by the Greek Neusis construction and seems to be new. We can easily construction three red points on the track above, and use the circle through the three points instead of the track.

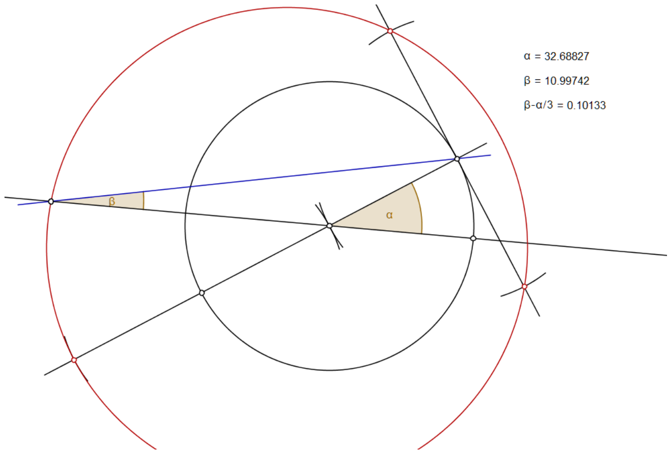
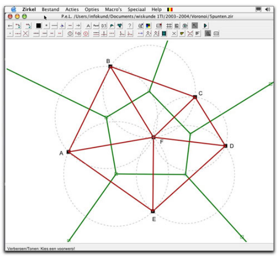


FIGURE 15. Approximative Trisection

The error can be computed and displayed by C.a.R. It is less than one tenth of a degree for angles less than 30 degrees, and it becomes much better for smaller angles. The author has computed the errors for various approximate trisections. This constructions is good, but not impressively good. Simpler constructions which trisect the cord are better. But it can probably been improved by taking three points on the track which are nearer to the intersection point S of figure 14.

8. JAVA

C.a.R. is written in Java and with good reasons. Only with Java, it was possible to produce a portable and yet powerful software for Windows, OSX and Linux. The Unicode support of Java helped to translate the program to many languages, including languages with completely different glyphs.



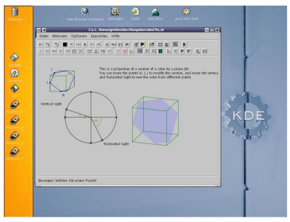


FIGURE 16. C.a.R. on OSX and Linux

Java as a language has several benefits.

- It produces to portable and small byte code. Thus Java code can be interpreted on all machines with a Java runtime environment (JRE), or can be compiled with a just-in-time compiler (JIT). The small code makes the language ideal for programs that are loaded over the net, such as applets on web pages and programs on thin clients. The lack of machine code allows sandboxes and safe programs.
- The language was designed to be object oriented from the start. It avoids error prone pointer arithmetic and cleans the memory with an automatic

garbage collector. Due to its clean design, the language is now a favorite for a first course in programming. The language is similar to C in many aspects, so that learners can easily switch to C++ or other programming languages.
- The JRE comes with a full library which allows to attack almost any project in Java, including projects interfacing with native code. A complete, cross platform API for graphical user interfaces is included.

It is a pleasure to develop a complex program like C.a.R. with Java and the IDE Eclipse.

However, Java as a language has got some problems recently. Even though one of the most successful mobile platforms, Android, is completely based on Java, the system is no longer used for net applications.

- The browsers on the mobile systems were never able to run Java applets, not even the browser on Android devices. Recently, most browsers gave up Java completely and replaced it with the inferior Javascript engine. Thus the nice web export of C.a.R. is now almost completely useless. But Java still runs on the desktop, on web services, and on embedded devices very successfully.
- There was never a Java version for the OSX tablets. Moreover, the Java language used on Android devices is not a complete implementation of Java, a fact that led to a law suit of the new owner of Java, Oracle, against Google, the maker of Android. In any case, Java programs must be ported to Android devices. This is not trivial for a program such as C.a.R.

Nevertheless, there are still numerous useful Java projects for the desktop. Java is still the most easy way to run code on Linux, Windows and OSX.

While the web export of C.a.R. is no longer useful, its capability to export graphics is central for the program. The program can export to PNG, PDF, EPS, SVG and FIG. Some export formats do not support the full range of transparency, however.

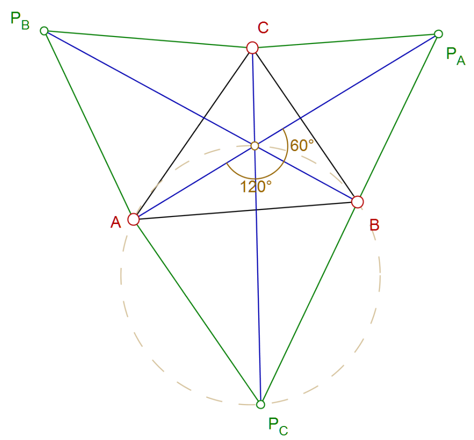


Figure 17. Scaled Graphics

The author prefers the PNG export. The vector formats seem to be preferable since they can be scaled. But in reality, each graphics must be designed with the proper size in mind. Simple scaling leads to thin lines, small points, or unreadable text. C.a.R. has an advanced dialog to set all these items for each export, and to preview the result.

Figure 17: This is an example of proper scaling. The text size fits to the rest of the page, and the graphics has a good enough resolution to be printed on normal print sizes. The position of the text elements had to be done precisely for the desired output size and resolution.

C.a.R. can also export in such a way that the text elements are done by Latex and laid over the graphics. Instead, the built in Latex display (done by HotEq) can be used for the display of formulas.

9. Discoveries and Proofs

In schools, proofs are not loved very much. They are thought as the most difficult topic in math, and they cannot be learned for tests easily but need to be practiced over a longer period of time. To be able to do proofs is indeed a matter of slow and continuous mathematical education. Students should learn that proofs can be achieved, even if they are hard to find, and that it is satisfying to find a proof. Stubbornness is one of the soft skills that mathematics can teach.

Geometry is one of the topics where proofs can easily be seen to be necessary, and where they are accessible to the student. This makes geometry an ideal tool for an education in mathematical thinking. Note, that we do not claim that it is the only tool. In other subjects, foremost in probability theory, it also possible to do less computing, and more modelling and thinking.

However, the proofs should be local. I.e., they should depend on other high level facts the students have already understood. A complete synthetic theory of geometry in the style of Euclid and Hilbert is certainly out of the scope of a general education, in schools and in universities.

Learning to do discoveries is even more important. And this is often done by experimenting. For experiments, a software like C.a.R. is an ideal tool. In other areas, an algebraic or numerical software like Euler Math Toolbox serves the same purpose.

Using a software has many benefits.

- With a software, it is possible to demonstrate that a mathematical result is indeed valid. E.g., in C.a.R., a construction may remain valid if the basic points are moved.
- Using the powerful track feature of C.a.R., it is also possible to answer "what if" questions. Sometimes, new results will spring into the eye. Often, the need for a proof becomes obvious.
- Only with a software, results involving advanced mathematical objects can be computed and demonstrated.

There is a new software called "Discoverer" (see [10] for details). It aims to discover mathematical relations in geometry. These relations can later be demonstrated using a software like C.a.R. The two programs complement each other well.

Last but not least, C.a.R. is a powerful tool to generate geometrical drawings for use in the classroom or in papers. If needed, C.a.R. can output vector formats which are open for further editing. Some formats are even human readable.

To summarize, a software for dynamic geometry or general geometrical software like C.a.R. should be used

- to discover,
- to demonstrate,
- to experiment,
- to learn.

C.a.R. helps to get rid of tedious things that any computer can do better, and move towards are more interesting thinking in mathematical education.

Supplementary Material

The enclosed file "Grothmann-CaR.zip" contains files in .zir format.

References

[1] C.a.R., http://car.rene-grothmann.de/.
[2] CarMetal, http://db-maths.nuxit.net/CaRMetal/
[3] Geometer's Sketchpad, http://www.dynamicgeometry.com/.
[4] Geogebra, http://www.geogebra.org/.
[5] Intergeo, http://i2geo.net/.
[6] Cindarella, http://www.cinderella.de/.
[7] Euler Math Toolbox, http://www.euler-math-toolbox.de/
[8] Cabri 3D, http://www.cabri.com/
[9] Archimedes Geo3D, http://raumgeometrie.de/drupal/
[10] Computer-Aided Education: Learning through Discovery, http://www.ddekov.eu/papers/2015-03-26_Confrence_Russia.pdf