WinCAG – DYNAMIC GEOMETRY SOFTWARE FOR TEACHING AND LEARNING IMPROVING VISUALISATION REASONING AND SKILLS

DANIEL DINIS DA COSTA

Title:	*WinCAG* – Dynamic Geometry Software for Teaching
Designer:	Karl-Heinz Brakhage
Contact information:	Institute of Geometry and Numerical Mathematics – Aachen University of Technology, Germany Website: http://www.igpm.rwth-aachen.de/~brakhage/ Email: Brakhage@igpm.rwth-aachen.de
Type of product:	Dynamic Geometry Software package
Price:	-
ISBN:	-

Keywords: *WinCAG[1], descriptive geometry, dynamic geometry, student teachers, professional learning.*

1. Introduction

The aim of this report is to evaluate the capabilities of the dynamic geometry software WinCAG, which implements an online and interactive system for the teaching and learning of descriptive geometry (DG). WinCAG is mainly applied in classroom demonstrations and to export files. The literature reviewed so far shows that there is a relative consensus within the research community that this software does support learners in developing their three dimensional (3D) spatial understanding and dexterity (Pütz 2001).

The development of imaginative creativity and visualisation skills for the understanding of 3D-geometric objects and forms and their interrelation underlies the rationale of this software program. These skills in turn, will permit learners to interpret and communicate information and ideas graphically. Furthermore, the software exposes clearly the overriding difference between the WinCAG technique and hand-drawn design, which lies in the introduction of ‘motion’ to objects and forms to emphasise their underlying shapes projected in three dimensions. By introducing new parameters through the use of a mouse or predefined modifications, the results obtained are reliable and valid. The use of predefined modifications for presenting forms in motion enables this software package to assist in the teaching and learning of practical 3D geometry.

The report consists of five main sections. The first section gives a general description and history of WinCAG. It critically describes how the software operates and evaluates its intended purposes. In addition to assessing its functions and aims, the benefits for teacher professional learning of descriptive geometry are discussed, and potential disadvantages are identified.

2. A general description of the product: brief history of WinCAG

Until the past few years, the use of geometric equipment and instruments for teaching and learning descriptive geometry was limited. In the mid-1980s, the turning point was developments in information and communication technologies that permitted the design of new educational software such as WinCAG. Originally developed between 1985 and 1987, WinCAG was adapted from an old DOS-version specifically to teach descriptive geometry. WinCAG is based on parametric construction. From the early 1990s onwards, high-resolution PCs and Windows systems became more easily available. This meant that:

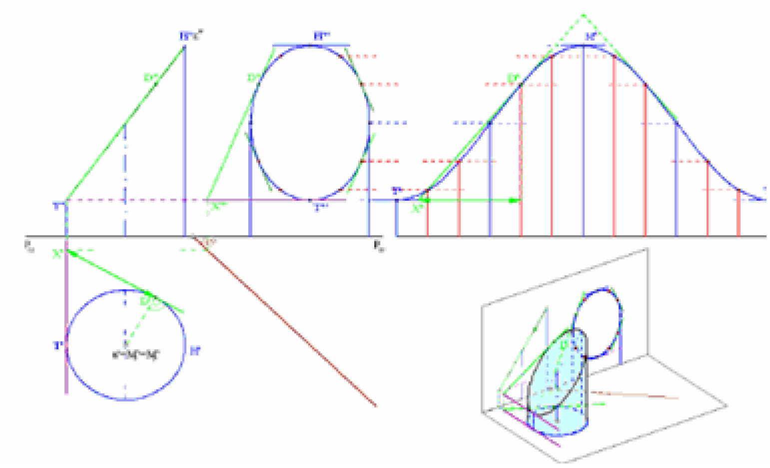
- Simple and complex objects could easily be animated;
- More and wider access to spatial visualisation training was available;
- Online modifications of projection and scaling could be achieved. This component appears to be the key for the visualization and thinking in 3D of projected shapes in two dimensions (2D); and
- The structure and development of representational systems allowed 3D applications of 3D in the teaching of descriptive geometry.

WinCAG uses concepts and geometrical elements such as points, lines, circles, cones, and parallel and perpendicular lines, and executes descriptive geometry exercises employing them (Monge 1847). These elements are often present in the WinCAG operations that follow.

3. What does WinCAG try to do? How does it operate?

The basic functions of WinCAG encompass three related steps. Firstly, it stores construction and related information. Secondly, it makes modifications online by dragging and moving around objects while the system memorises and displays each position the object or objects have occupied. Finally, it supplies the resolution (output) of ongoing changes in all meaningful situations (correct positions) as Brakhage demonstrated in figure 1.

Figure 1. A cylinder's section and development[2]



This depiction of a geometric module shows how to obtain an axonometric projection of a two-place projection of the cylinder using the layer technique, which allows teachers to visualise complex and pre-constructed positions of 3D geometry problems both already in motion and by clicking the mouse a few times to generate motion. However, this procedure falls short of redefining intersections for the design and display of projections of an object with the aim of setting it in motion, due to a lack of one geometric degree of freedom.

The 3D-geometric principles of this degree of freedom consist of commands and modifications. A user who applies these principles could easily generate and modify points provided that they have: *(i)* two degrees of freedom (a point by coordinates and a point relative to another point); *(ii)* one degree of freedom (a point on a line, a circle, a cone; a point given by a convex combination of two points; and a point at a certain distance from another point in a given direction (height); and *(iii)* fixed points which relate to a point of intersection, tangent point and nadir and midpoint. Other incidences such as placing a point on a circle or defining a point of intersection may lead to modifications at a later stage of geometric construction.

It is important to note that the commands for generating the *fixed points* are thus far too unreliable for the demonstration of motion due to the lack of at least one degree of freedom. Nevertheless, the designer proposed a command syntax with a degree of freedom for generating a *moving point* such as the point *P*, in figure 2, that is stored internally in a form of *P=definepoint (x, y)*, considering that *x* and *y* encompass the two-dimensional coordinates of point *P*. The next step consists of dragging *P* towards a new position, for which the command syntax is formulated as: $Modifypoint(P; x_0, x_1, dx, y_0, y_1, dy)$.

In examining figure 2, it is understood that when the point *P* moves, it produces a rectangle $[X_0X_1]x[Y_0Y_1]$ to x-direction *(dx)* and to y-direction *(dy)*. Should the point *P* touch the borders of the rectangle, it deflects as a result of the law that the angle of reflection is equal to the angle of incidence. The application of this law works in practice as exemplified more explicitly in figure 3 below.

Figure 2. Point P deflecting to different direction

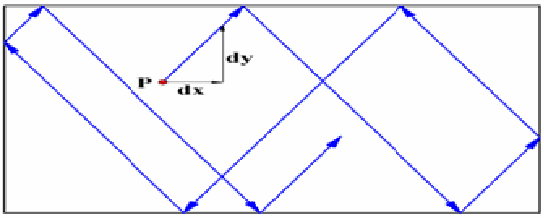
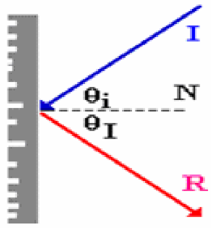
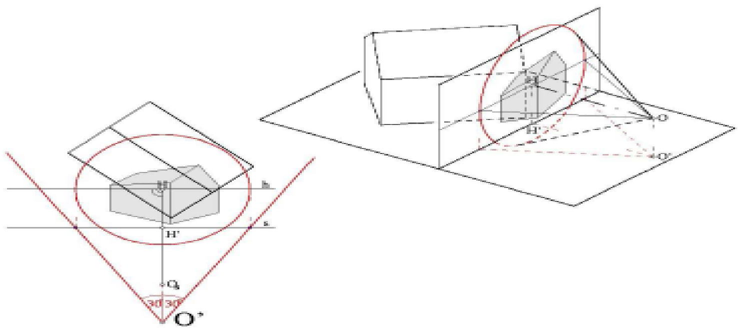


Figure 3. The law of reflection in practice



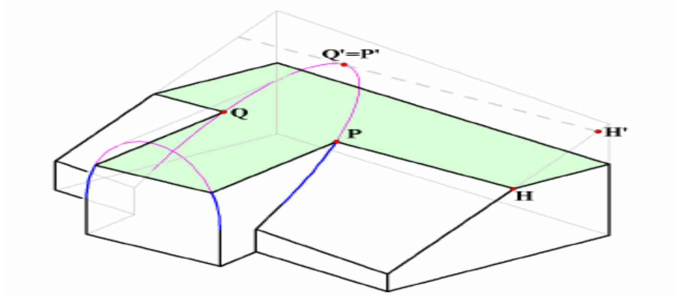
The object I ray moves in a straight line. When a ray collides with a reflecting object at an angle, it traces the path after deflection to what is known as a normal line *(N)*. The resultant *R* of these two gives the direction of the reflected ray. It follows that the angle of incidence is equal to the angle of reflection as a result of WinCAG's automatic detection in accordance with the law of reflection (see fig. 2). The resulting automatic modification can be achieved using a virtual command that alters the point of view (0') to influence a viewer's height, as figure 4 shows.

Figure 4. ModifyPoint in a practical example



In addition to the introduction of the element of height for the modifying point as the point of view, WinCAG offers a number of other possibilities for solving 3D geometry ambiguities, such as the intersection of a line with a conic double root, by determining the nearest point to the intended intersection point (special position) which is proposed in figure 5.

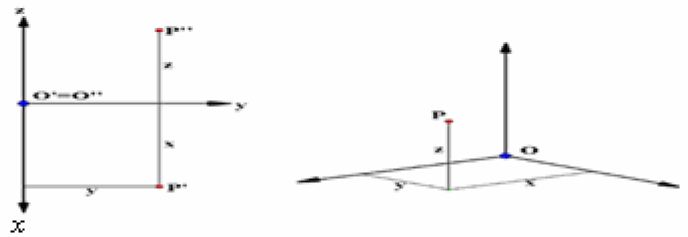
Figure 5. An application for removing ambiguities in special designing positions



This illustration is a typical example of unintended ambiguities which may occur whenever there is a double intersection of the horizontal line *H′* with the ellipse in the roof of the house. To solve this problem, the designer assumes that the point *P* is nearest to *H* and *Q* is further away from *H* or *P***,** since the use of the method of continuous change alone impairs the distinction between *P* and *Q* as a result of changing the height *(H)*. However, to enhance its capability, WinCAG uses the projection module in this exercise to solve 3D-geometric problems where a single method cannot.

WinCAG's use of the projection module aims to offer spatial demonstrations. In our view, projection modules underlie the principle of the top and front orthographic views from which a coordinate system stems. In other words, the assumption is that there is a definition of *0′=0″,* which axes theoretically are vertical and horizontal of which coordinates are computed to project *P′* and *P″* respectively, as shown in figure 6.

Figure 6. Representation of the axometric projection of point P

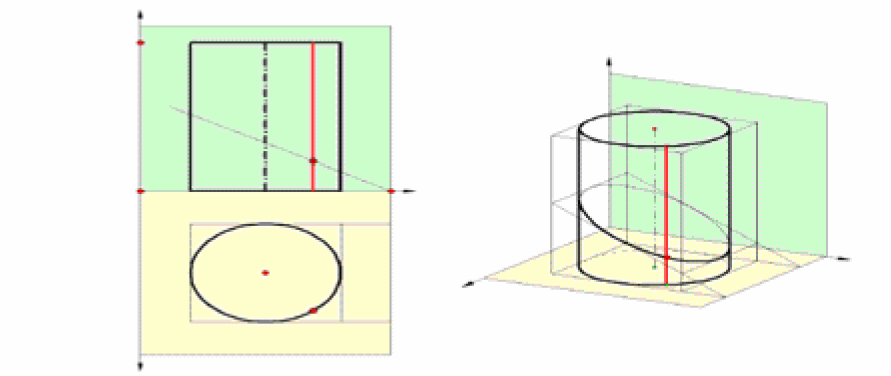


In addition to WinCAG's ability to compute the coordinates of *P (x, y, z)* and the parameter of the projection, the purpose of the exercise is to produce an automatic modification, where a point is represented by its 3D coordinates. The designer's command syntax for these coordinates and parameters is as follows:

P' = DefinePointRelative (O', x, y)
P" = Distancefrom (H, V; Z)
DefineMap (O', O, Kind, φ, θ, r, scale)
ModifyPoint (P'; Z_o, Z_1, dz)
ModifyMap ($\varphi_o, \varphi_1, \theta_o, \theta_1, d_\theta$).

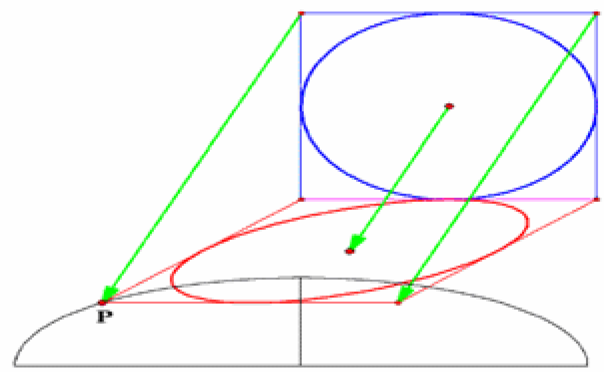
In the DefineMap command syntax, the *φ, θ* and *r* parameters depict spherical coordinates, of which distance *r* represents the perspective view among the other possible parameters alluded to in figures 1 and in 7. *ModifyMap* ($\varphi_o, \varphi_1, \theta_o, \theta_1\ d_\theta$) determines a change of the parameter in respect of the position, radius, and height of the cylinder, as well as the inclination of the cutting plane, position of the surface line, and parameter of the view.

Figure 7. An application of a projection of modeule with ModifyMap



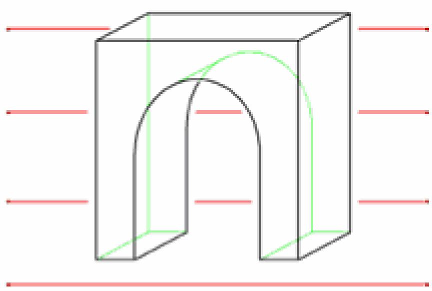
In examining the application of the projection module, it is important to understand that WinCAG also relies on other useful commands that are pertinent to the teaching and learning of 3D geometry. These commands are threefold. Firstly, an object conic section is often obtained as the affine or perspective image of a circle. A geometric transformation can then easily map the circumfluent square of the circle. Finally, it is possible to compute the inscribed section. Likewise, in figure 8, it is possible to understand the proposed application of these commands the automatic modification of point *P* that moves on an ellipse. Similar commands cater for a semicircle and its circumfluent rectangle.

Figure 8. Demonstration of special command for a point on an elliptical surface



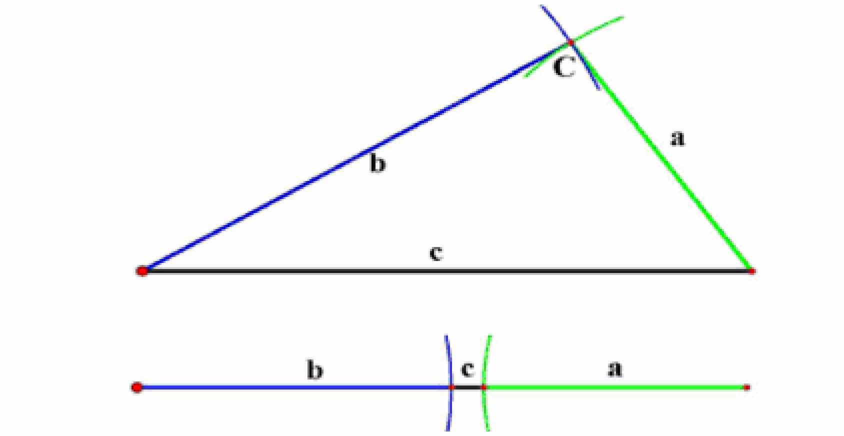
In addition to commands for designing circles and semi-circles, another command worth mentioning is relevant to visibility checks. It is interesting to note that this is a practical technique for creating a notion of spatial dimension and visibility in a planar object, as suggested in figure 9. The design of this command essentially aims to allow 3D effects in a normal 2D operation, making it an excellent tool for demonstrations or modelling either in classroom practice or in a remote setting. The designer's ultimate aim behind this 3D geometry command is an abstractive model of a third dimension that could provide students with a deeper understanding of forms and their functions.

Figure 9. Illustration of the generation of visibility



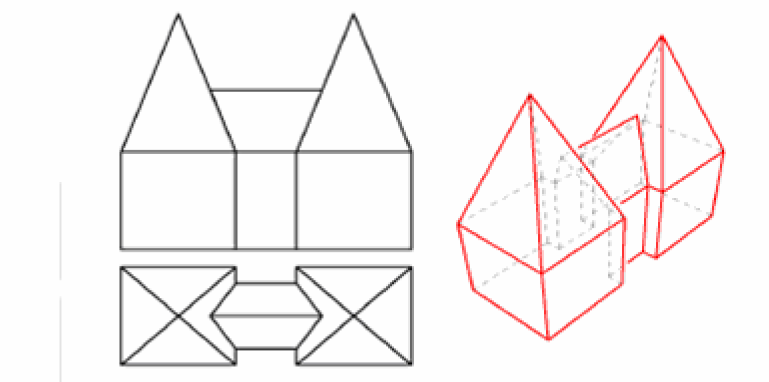
Turning to the aspect of flexibility, where students can achieve the same result using different methods of construction, it is also interesting to note that the software employs an alternative command, as in figure 10 in which a given point *C* results from the intersection of two circles which eventually do not touch each other. As a result, a link point is added on point *C* to superimpose *c*.

Figure 10. An alternative procedural technique for intersections



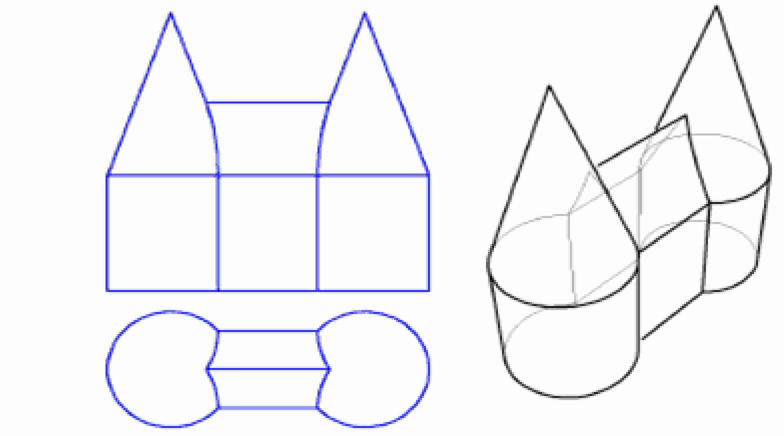
This alternative command is helpful in several ways, such as in suiting the different learning styles of users and meeting their needs in solving a problem to arrive eventually at the same result. Another command syntax for an orthographic projection of a building is applied to determine basic designs (triangular roof, pyramid and prism), from which a perspective view can be obtained by means of the projection module to arrive at the resolution suggested in figure 11. This particular procedure takes advantage of the users' prior knowledge and imaginative creativity because of the nature of the transition from a 3D to a 2D form and vice-versa.

Figure 11. Sketches of orthographic and perspective views



For this specific example, a similar procedure is undertaken through a command syntax in figure 12, which applies a circumfluent square, a tangent ellipse-ellipse and a tangent point-ellipse. In practice, the intersection of a roof-cone is based on five points of a conic section, where an appropriate operating scale can enlarge the perspective view.

Figure 12. Modified version of a building with conic section



The last two examples clearly show an extended application of this software. The extensions to these 3D-geometric constructions could indeed assist students' logical thinking and visualisation skills.

4. An evaluation of the product - How well does it achieve its purpose and/or for whom might WinCAG be beneficial?

WinCAG offers an array of techniques and capabilities for teaching and learning 3D geometry through problem-solving exercises. In other words, the program constitutes an asset for improving students' spatial visualisation skills and logical reasoning when performing WinCAG-based interactive 3D geometry learning. In addition to its reliable use for rapidly revising problems in classroom interaction, WinCAG is a functional tool in the context of self-directed learning, where the teacher's presence is minimal. WinCAG consists of a number of key characteristics that are potentially beneficial for 3D geometry student teachers during their professional learning activities. These benefits are as follows.

- 3D geometry problems can be resolved online;
- Time efficiency is ensured;
- Unlike hand-drawn design, the program permits students to retrieve and continue exercises from the point at which errors are detected;
- Interactivity and explanatory step-by-step design takes place;
- The animation of forms allows improved spatial visualisation of views and reflection of forms;
- Saving/storing, encrypting and sending solved problems to peers or teachers is part of online practice;
- Teacher educators/lecturers can measure student teachers' individual performance and monitor progress.

Central to the best use of this software package is the ability of student teachers to take electronic files of solved problems away from the classroom. These may contain pertinent information on layers, movements of objects and the process of construction. One additional advantage of this software is that both teacher trainers and student teachers can solve 3D geometry problems in the classroom and at home and send them in the form of bit-maps, encapsulated-postscript-files or simply text files. Likewise, Pütz (2001) indicated other similar potential benefits that as an instructional tool WinCAG could bring to the teaching and learning of descriptive geometry.

A recent study (Hannafin and Scott, 2001) investigating students' and teachers' attitudes and beliefs towards the use of similar dynamic geometry software yielded interesting findings. The findings indicated that teachers liked and valued the

Geometer's Sketchpad as a tool, but had conflicting opinions about the value of using it to support student-centred activities. Students were quite positive about self-controlled learning activities and their ability to use geometric shapes in motion. The results also suggested that changing teacher's attitudes about the most appropriate setting for learning proved to be more difficult than academics and practitioners have anticipated. However, this viewpoint may be disputed, because after all, it can be concluded that the program offers a number of possibilities and applications in teaching environments where teachers and learners make successful use of it.

We can conclude that dynamic geometry software is undoubtedly valuable and worthy of application as part of a fully functional and improved classroom practice in descriptive geometry in training teachers professionally. The software can make a recognisable contribution to purely traditional hand-drawn projective geometry and to the constructivist way of teaching and learning. The present author has worked as a descriptive geometry teacher for several years in secondary and further education in Mozambique, where traditional methods are still dominant. It would be particularly interesting to discover what the use of such dynamic software for teaching could contribute to classroom interaction in such a context.

5. Main shortcomings of WinCAG

The main shortcomings of this dynamic geometry package are fourfold. Firstly, it requires that the user should have a solid background in 3D descriptive geometry. Secondly, redefining intersections can be an issue for the design and display of projections of an object with the aim of setting it in motion. Thirdly, a limiting factor not to be underestimated concerns the use of memory space for the program. If stored in a local device this might slow or impair its applications, in which case memory upgrading would be necessary. Finally, this software would be more useful if it could be used with subsidiary multimedia devices and other geometry techniques such as problem solving and games to improve 2D and 3D visualisation skills, imaginative creativity and logical thinking, as opposed to merely using basic projective elements.

6. Conclusions

This report has examined the uses and capabilities of the dynamic geometry software, WinCAG, which has the potential to improve the quality of teaching and learning as a real animation tool with online presentation in developing imaginative creativity, systematic thinking and visualisation skills. The use of real animation allows teachers and learners to visualise procedures, online presentation facilitates teaching and learning interactions, saves time and offers accurate solutions.

However, it has to be recognised that WinCAG cannot fully substitute for classroom teaching, since the package supports only a limited range of interactions. Nevertheless, this program can help to develop imaginative creativity by encouraging students' systematic thinking in 2D and 3D, and by relating this to spatial visualisation skills in solving descriptive geometry problems in a practical manner.

WinCAG is therefore a didactic tool, which may contribute to substantive improvements in the way student teachers of descriptive geometry learn professionally. WinCAG could also be particularly useful for introducing new conceptual frameworks and strategies to the field of education in practical 3-D geometry in developing countries such as Mozambique.

References

BRAKHAGE, KH, (2001). WinCaG: Education Software for Geometry; Paper presented at *11th International Conference on Geometry and Graphics, 1-5, 2004*, *Guangzhou*. Aachen University of Technology.

HANNAFIN, RD., & SCOTT, B. N. (2001). Teaching and Learning with Dynamic Geometry Programs in Student-Centred Learning Environments: A Mixed Method Inquiry. *Computers the Schools*, 17, 121-141.

MONGE, G. (1847). Géometríe: Suivie d'une Theorie des Ombres et de la Perspective, *Extrait des papiers de l'auteur par M. Brisson*, Paris.

PÜTZ, C. (2001). Teaching Descriptive Geometry: Principles and Effective Methods Demonstrated by the Example of Monge Projection, *XV Conference on Graphics*, *Sao Paulo Brazil, November 5-9, 2001*.

About the author

The author is a PhD student in the School of Education, Communication and Language Sciences, University of Newcastle upon Tyne, England. Comments on the article can be directed to Email contact: d.d.da-costa@newcastle.ac.uk

---

[1] **W**indows **C**omputer **A**ided **G**eometry
[2] All figures from Brakhage (2001).