

# The use of GeoGebra software for typesetting mathematical text

## Abstract

*It is important to make the right decision on how to draw accompanying pictures when typesetting mathematical text. One option is the use of dynamic geometry software. This article deals with the possibilities of exporting pictures from GeoGebra software and the problems encountered during this process. Although GeoGebra enables effortless creation of a dynamic geometry figure, there are limited possibilities of exporting completed figures in the form of pictures. It is particularly complicated to achieve a uniform appearance of pictures created using a variety of Graphics View zooms. The basic requirement here is to ensure unity in the line thickness and font size of legends for pictures in upcoming publications. We have proposed and programmed a sophisticated module ensuring the export of figures from GeoGebra software. These improvements enable the effective creation of illustrations with the desired uniform appearance.*

## 1. Introduction

The typesetting of mathematical text is a common activity. In certain situations, mathematical text itself is insufficient for the needs of readers. It needs to be accompanied by visual schemes or diagrams. Adding pictures to text is seen to be particularly important in geometry. Here, graphical representation is often not only part of the problem or the method of solving a problem but also the solution itself. It is then crucial to make the right decision on how to create these pictures. When choosing a suitable way to create pictures, it is essential to consider the aspect of inserting the completed picture into the typeset in the highest possible quality, not only from the point of view of image resolution but also with regards to the individual appearance of pictures inserted into the upcoming publication.

One of the major aesthetic requirements is the use of a constant font size for all of the pictures, whilst keeping this size relative to other text appearing in the given typeset. Furthermore, it is necessary to ensure that the thickness of lines with a certain significance (i.e. not purposely highlighted) is identical in all pictures. The final requirement is the optimal “density” of a picture in the typeset so that a picture is neither too empty nor overcrowded with too much information cramped into a small space. Although it is easy to enlarge or scale down pictures as a whole, it is complicated to maintain (or even find) the same line thickness and font size when editing in this way. For this reason, it is wise to maintain a single font size and line thickness for pictures at the time of export. Any subsequent change of size should be carried out on all pictures included in the typeset or not carried out at all.

Currently, there is a wide range of techniques that can be used to create pictures. One way is to draw the required scheme using vector graphics software. Such an approach offers huge potential due to its universal character, providing opportunities to create schemes extending beyond mathematics. However, this software does not abide to mathematical laws, which have to be added explicitly. If entry conditions are changed, the whole drawing can easily fall apart. This is considered a significant drawback to the technique. An advanced method of picture creation is drawing in a LaTeX environment (for example using Metafont, PSTricks package and other similar). This option is rather demanding as individual parts of a drawing have to be created using

text commands. Another possibility for creating pictures to accompany mathematical text (particularly geometric) is the use of dynamic geometry software.

Dynamic geometry software (DGS) is designed for the construction and analysis of tasks and problems in geometry [1]. Its major benefit is the automatic re-working of a construction after modifying any parameter (e.g. point position) whilst maintaining all dependants and conditions in the construction [2]. This feature is important when creating pictures designated for typesetting in a situation where an almost completed construction needs to be edited, due to, for example, clarity or more efficient use of space in the upcoming publication. The following text deals with the possibilities for exporting pictures from GeoGebra software but sets out by describing our method of investigation.

## **2. Method of investigation**

GeoGebra software documentation had been expected to provide the basic source of information for our investigation. However, we discovered that the issue of picture export is currently not sufficiently documented in these sources, not even in the theses by the main author of the software [3, 4]. Alternatively, we inclined to the practical testing of export.

### **2.1. Testing picture export**

To test how well picture export from GeoGebra software works, we used a set of approximately 180 figures. Our aim was to reveal mistakes and risk factors that might occur while using various methods of picture export. We checked the appearance of exported pictures after manipulating with the following parameters:

- Graphics View zoom
- GeoGebra window size
- Font size in the figure
- Line thickness of elements of a figure

At the same time, we focused on extreme situations, where, for example, Graphics View had been highly zoomed in or, contrarily, zoomed out before figure creation. Where a particular export method allowed, we also tested the influence of various subsequent export settings on the resulting picture.

### **2.2. Method of importing pictures into typeset**

Created pictures were imported into the typeset in the LaTeX environment and in MS Word software. For this import, the size of the picture in the typeset had to be adjusted. The ratio of the size of the picture in the typeset to the original picture size will be referred to in the following text as the *import ratio*. We did not adjust the import ratio for each picture separately, but we set a single import ratio which was implemented on all pictures in a series<sup>1</sup>. In the LaTeX environment, import ratios were set using the *scalebox* command. For typeset carried out in MS Word software, the size of a graphic element was modified using the *Picture scale* function.

---

<sup>1</sup> This condition mainly results from a lack of time for real typesetting of pictures, where experimenting with the individual import ratio of each picture would be very time-consuming.

### 3. Storing and rendering figures in GeoGebra

GeoGebra stores parts of a figure in computer memory as abstract models [3]. Apart from a mathematical definition, individual elements of a figure have allocated properties essential for rendering on screen. Some of the more significant ones are visibility of a given object, title, description, color, viewing size (point diameter or line thickness) and others. Figures are viewed in so-called Graphics View, which is a visible and, in a certain way, zoomed part of the workspace. The size of Graphics View changes in a linear relationship to the GeoGebra window size.

When rendering a figure in Graphics View, point size and line thickness can be edited using settings for the given object. The size of texts viewed in Graphics View can be set in the appropriate program menu. Such chosen values are respected when rendering a figure on screen, regardless of any Graphics View zoom in or zoom out, as shown in Figure 1. Only the “density” of the viewed figure changes whilst zooming in and zooming out in Graphics View. That is to say only the amount of objects in a certain part of a screen’s workspace changes, not the size of texts, radius of points or line thickness.

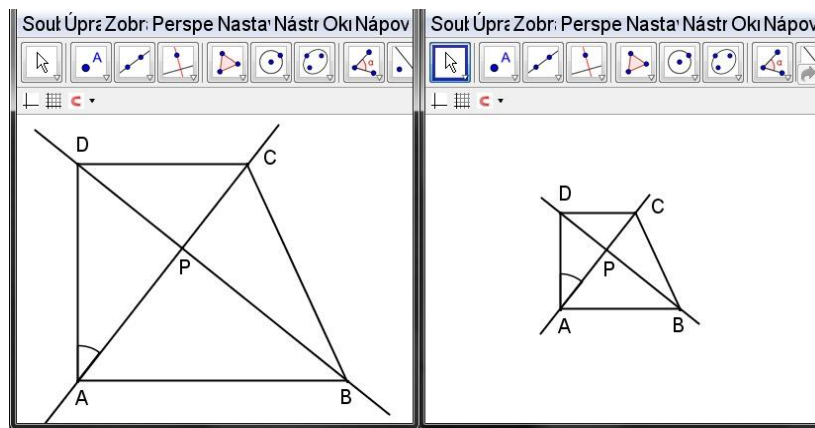


Figure 1: *Line thickness and font size when zooming Graphics View in and out*

### 4. Problems when exporting pictures from GeoGebra

Whilst studying the acquisition of pictures from GeoGebra, we focused on methods of export that enable the creation of a bitmap or vector picture. These were as follows:

- Creating a screenshot
- Export Graphics View as Picture
  - Scale in cm
  - Size in pixels
- Export Graphics View to Clipboard

#### 4.1. Creating a Screenshot

In GeoGebra, as in most applications, the PrintScreen function can be used to create a screenshot. A picture initiated in this way is a bitmap one with a resolution identical to the resolution of the highlighted part of the screen. This is a relatively low resolution for printing pictures. Hence, where standard print quality is preserved, there is a limit to the maximum size of a picture. A picture with

a maximum possible size obtainable from the figure in Figure 1 (on the left) when respecting a standard print resolution of 300 DPI is shown in Figure 2.

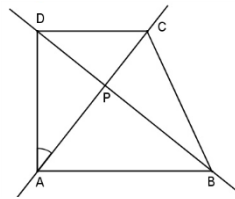


Figure 2: Picture created from the figure in Figure 1 (on the left) using PrintScreen function, import ratio 1.0

#### 4.2. Export Graphics View to Clipboard

A static picture can be obtained from GeoGebra via Export Graphics View. The resulting picture will contain that part of the workspace visible on screen as Graphics View. The coordinate axes and grid will be displayed providing they are displayed in Graphics View.

The use of Export Graphics View to Clipboard leads to the creation of a bitmap picture with a sufficient resolution for printing. However, any change in Geogebra window size on the screen results in the size of the resulting picture being modified at a rate that is not directly proportional to the change in window size.

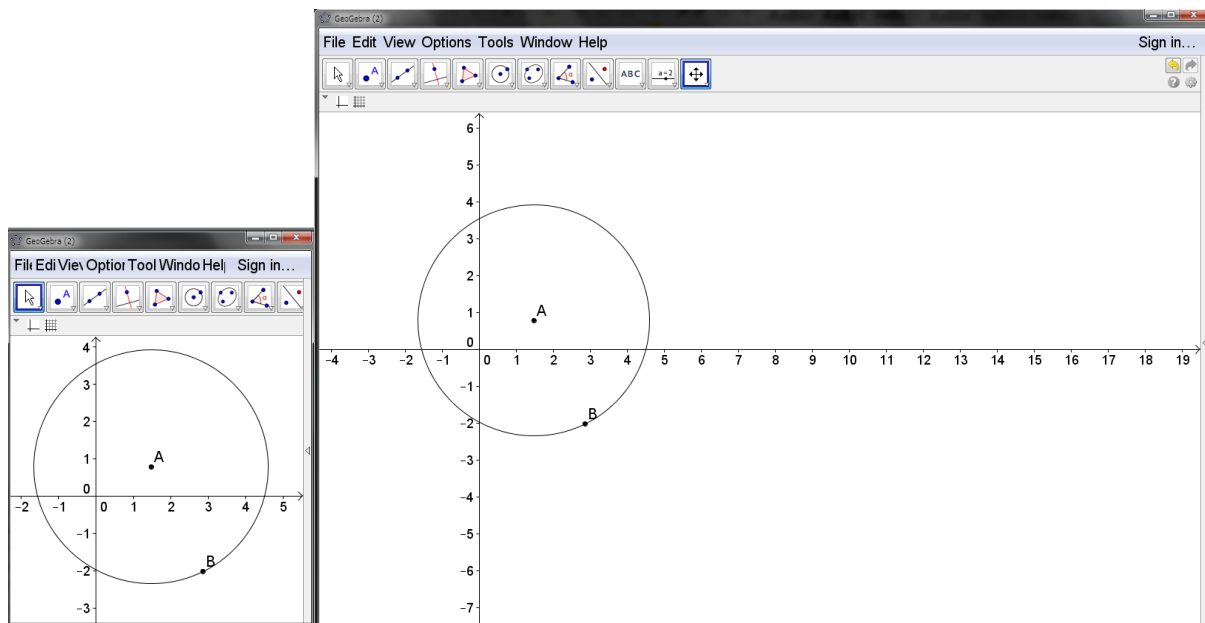


Figure 3: Changing GeoGebra window size with Graphics View

Figure 3 shows windows with the same figure. These two windows differ only in size on the screen (and hence Graphics View size). Using the Graphics View to Clipboard function, two pictures were obtained and subsequently imported into the typeset with the same import ratio, creating figures 4 and 5.

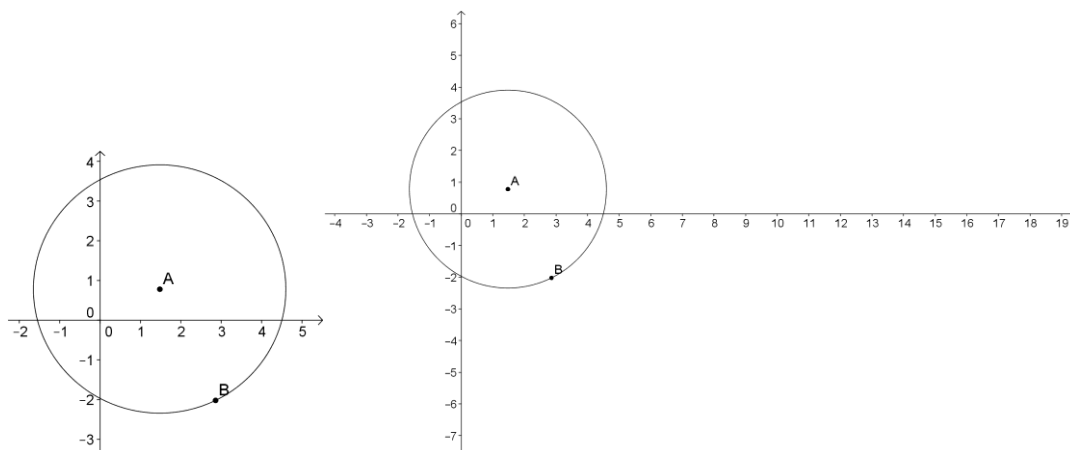


Figure 4 and 5: *Export via Graphics View to Clipboard, import ratio 0.15*

Although the same method of export and import was used for both pictures, there are clear differences in the size of individual elements of the figure. Despite carrying out an analysis of the exported pictures, we were unable to find out what parameters the resulting size of individual pictures depends on and how changes in size can be eliminated.

### 4.3. Export Graphics View as Picture

Export to files carrying graphic information is enabled by saving Graphics View in the form of a vector or bitmap picture. There are a number of various formats to be chosen from when exporting: EPS, PNG, PDF, SVG a EMF.

EPS format is vector and pictures exported in this format can be enlarged and scaled down without affecting quality – unlike pictures exported into files of bitmap formats [4]. It is natively supported by the LaTeX typesetting environment but it is not possible to import them into OpenOffice Writer text editor. PNG format is bitmap with loss-free compression and Hohenwarter claims it is used on web pages and in MS Word text editor [4]. Apart from that, it is supported by OpenOffice Writer and can be used in the LaTeX system if an appropriate source code compiler is used. Pictures in SVG, EMF and PDF format are vector but their import into text editors and the LaTeX system is complicated: PDF type pictures cannot be imported into MS Word or OpenOffice Writer and pictures in EMF format are not supported by LaTeX. SVG files are not supported by any of the above mentioned programs; for that reason we have disregarded SVG files in our further analysis.

Currently there are two possible ways to export Graphics View as Picture from GeoGebra: via *Scale in cm* option and via *Size in pixels* option.

#### Export via Scale in cm option

When exporting via Scale in cm option with an export ratio of 1:1, one unit on the coordinate axis has a length of one centimetre in the resulting picture. Our investigation leads us to believe that this is ensured by a “vector copy” of the Graphics View and the subsequent change in size of the created picture. When using this method, the resulting resolution of a figure with a highly zoomed out

Graphics View is several times larger than a figure with a zoomed in Graphics View<sup>2</sup>. When viewing both pictures at 100% size, line thicknesses, point diameters and font size differ greatly. Font size, line thickness and point size in the exported picture are indirectly proportional on the screen to the displayed distance of two adjacent marks on coordinate axes. However, this does not apply when exporting a picture to an EPS type file, where line thickness remains constant despite changes in point diameter size and font size.

Because the PNG format appears to be the most appropriate for the purposes of our analysis, our investigation will go on to deal solely with export to this file type. Export to a file of EPS vector format has been dealt with in the monograph [5].

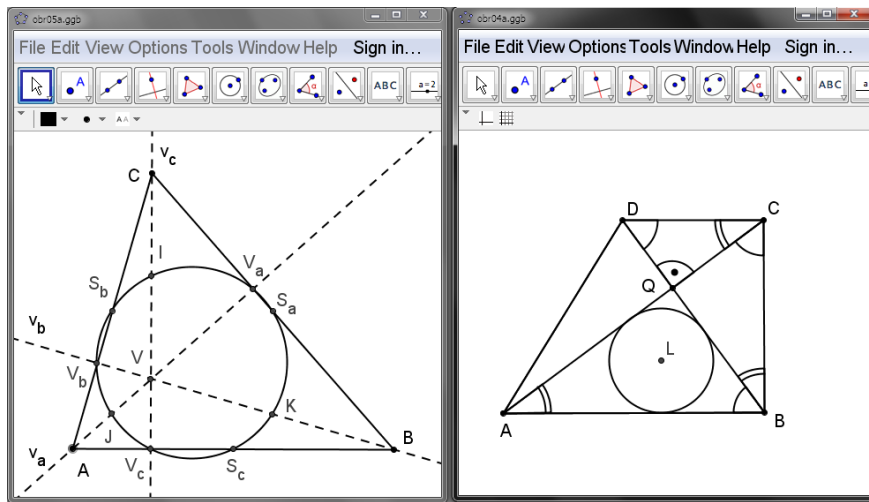


Figure 6: *GeoGebra windows with initial figures*

Figure 6 shows windows with two figures which look very similar from a technical point of view. Because they were created separately, their Graphics View zoom differs. The pictures were exported to a PNG type file with an export ratio of 1:1 and subsequently inserted into the typeset of this text with a single import ratio, leading to the creation of figures 7 and 8.

---

<sup>2</sup> Resolution can only be considered when exporting to PNG format; the given findings can also be applied on export to EPS, EMF and PDF formats

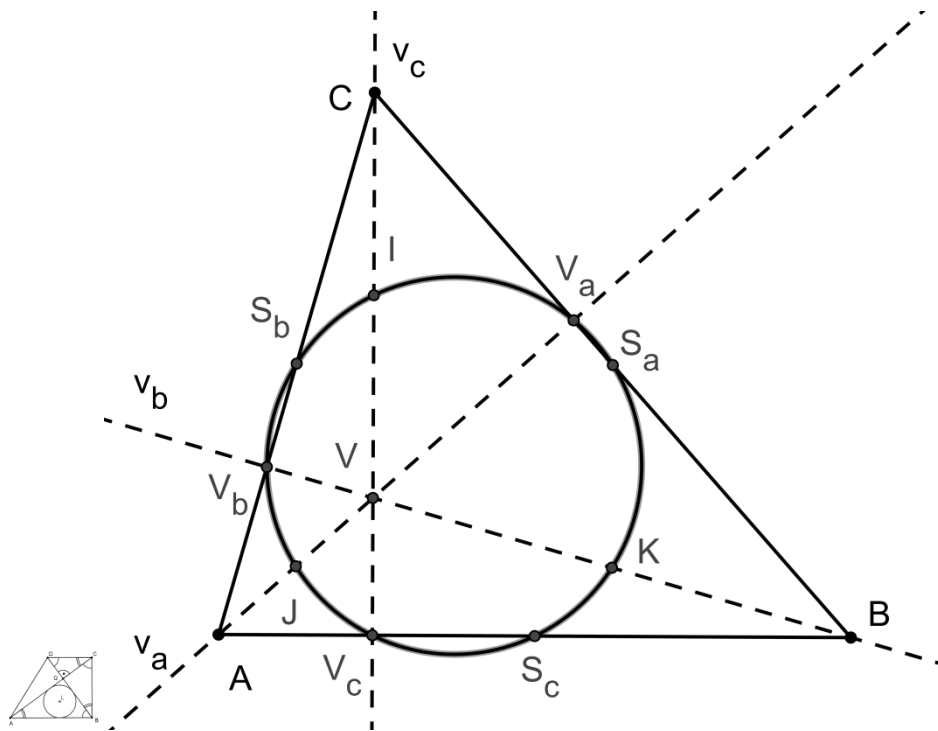


Figure 7 and 8: *Export via Scale in cm, export ratio 1:1, import ratio 1.0*

As figures 7 and 8 clearly illustrate, export via Scale in cm with export ratio set at 1:1 does not ensure a uniform picture size. When exporting a picture from GeoGebra via Scale in cm, the desired export ratio  $a : b$  can be set, changing the size of the resulting picture. This function carries out an appropriate change to the size of the “vector copy” Graphics View, so the  $b$  unit on the coordinate axis has a length of  $a$  centimetres in the resulting picture. At the same time, point diameter, line thickness and font size change in direct proportion to the selected export ratio.

In our analysis, we also focused on the question of how to make use of these findings to ensure a uniform font size, point diameter and line thickness for pictures. We discovered that, given the same Graphics View dimensions for figures, it is enough to ensure a uniform size of resulting pictures in centimetres. That can be done by defining one reference figure with a known export ratio of  $a_0 : b_0$  and appropriately setting an export ratio of  $a_k : b_k$  for the other figures. The resulting picture will then have the same dimensions as the reference figure. If the Graphics View dimensions differ, calculation of the export ratio  $a_k : b_k$  must also take into account the ratio between the Graphics View dimensions for the reference and produced figures. Although our testing found this procedure to be useable, it cannot be particularly recommended due to the time-consuming calculation of export ratios.

### **Export via Size in pixels option**

Export via Size in pixels option is possible only to bitmap formats (PNG, SVG). The use of this type of export enables the resolution of the resulting picture to be explicitly defined. This can again be ensured by a “vector copy” of the Graphics View and subsequent change in picture size in order

to preserve the required resolution. The point diameter, line thickness and font size change in direct relation to the required picture resolution.

If figures having the same Graphics View dimensions are exported in this way and provided the same required resolution of resulting pictures is set, these pictures will have matching font size and line thickness. Here, the Graphics View zooming of the original figures is irrelevant. However, if the Graphics View is of a different size to the figures, font size and line thickness will also differ among pictures.

The figures from Figure 3 were gradually exported using the Size in pixels method, having defined a single width for the resulting pictures. This led to the creation of two pictures which were imported to this typeset with the same import ratio, with Figures 9 and 10 emerging. The differing font size and line thickness of these two pictures is clear.

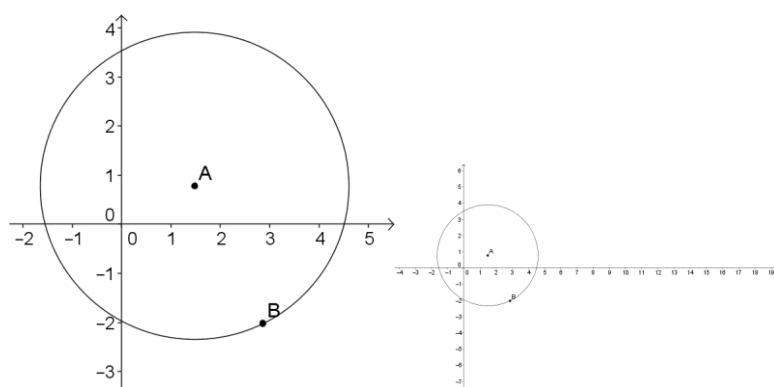


Figure 9 and 10: *Export via Size in pixels, required width of pictures 600 pixels, import ratio 1.0*

In our analysis, we again focused on the question of how to ensure a uniform font size, point diameter and line thickness for all pictures created from figures with a differing Graphics View size. We discovered the need to ensure a uniform ratio of Graphics View size and required size of resulting picture for all exported figures. If we call Graphics View width  $x$  and resulting picture width in pixels  $w$ , then  $x_k / w_k$  must be constant for all pictures. This can be achieved by appropriately setting the required resolution of the resulting picture. Although our testing found this procedure useable and simpler than the Scale in cm method, it cannot be particularly recommended due to the time-consuming calculation of appropriate required resolution and bearing in mind it can only export to bitmap formats.

## 5. Possibilities of simplifying export

The preceding text has analysed methods enabling the export of figures from GeoGebra in the form of bitmap and vector pictures, focusing on the ability to generate pictures of uniform appearance. Not having managed to find a method meeting our requirements, we went on to deal with the question of how to simplify the export process. We are of the opinion that it would be optimal if the user had the chance to choose how many units of length (in centimetres) are to represent a particular part of the Graphics View width in the exported picture. This data should be constant for all



exported figures regardless of Graphics View size and zoom. For this, the possibility of saving exported figures in vector picture formats needs to be preserved.

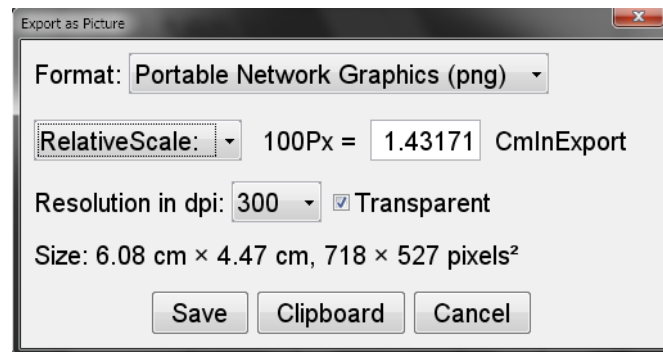


Figure 11: *Export dialog for our proposed export method*

We decided to download GeoGebra source codes from the GeoGebra developers' website and analyse them. We discovered that our proposed solution is feasible. From a programming point of view, it does not involve a brand new method of export but the current Export Graphics View as Picture method could be used and modified. We implemented the export proposal and the export dialog for our solution is illustrated in Figure 11. The user inputs the number of centimetres to be represented by 100 pixels of the Graphics View in the picture. As confirmed by testing, the input of uniform data ensures the same font size, point size and line thickness for all pictures created using this method. Figure 12 shows a mutual comparison of line thicknesses and font sizes of figures from Figure 6.

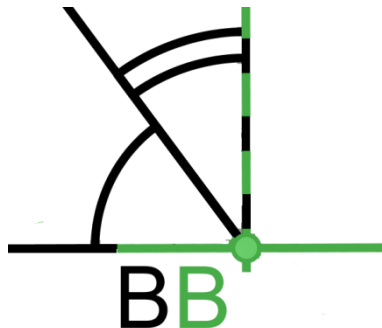


Figure 12: *Comparison of font size and line thickness of two pictures (CmInExport value set at 5.72 for both)*

## 6. Conclusions

Having analysed methods of obtaining bitmap and vector pictures from GeoGebra software, we were unable to find a method which would enable efficient creation of pictures with a sufficient resolution and uniform appearance. After analysing current GeoGebra source codes, we proposed and implemented an improvement to the export of figures, which allows the efficient creation of high-quality pictures with uniform appearance. We sent the source code of this improvement to the GeoGebra authors with a request for inclusion in the main software development branch. Although

our request has not yet been evaluated, we believe that this improvement will be included in future versions of the software and the export of pictures from GeoGebra will become more efficient.

## 7. Acknowledgements

The research was supported by the project GAJU 017/2013/S.

## 8. References

1. Sträßer, R. (2002). Research on Dynamic Geometry Software (DGS) - an introduction. In *Zentralblatt für Didaktik der Mathematik*. **34**(3). Retrieved from: <http://www.springerlink.com/content/j733470n67l36678/fulltext.pdf>
2. Kortenkamp, U. (1999). *Foundations of Dynamic Geometry* (Doctoral dissertation). Swiss Federal Institute of Technology Zurich, Switzerland. Retrieved from: <http://kortenkamps.net/papers/1999/diss.pdf>
3. Hohenwarter, M. (2002). *GeoGebra: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene* (Master's thesis). University of Salzburg, Austria. Retrieved from: [http://www.geogebra.org/publications/diplomarbeit\\_geogebra.pdf](http://www.geogebra.org/publications/diplomarbeit_geogebra.pdf)
4. Hohenwarter, M. (2006). *GeoGebra – didaktische Materialien und Anwendungen für den Mathematikunterricht* (Doctoral dissertation). University of Salzburg, Austria. Retrieved from: [http://www.geogebra.org/publications/mhohen\\_diss.pdf](http://www.geogebra.org/publications/mhohen_diss.pdf)
5. Günzel, M. et al. (2012). *Integrace elektronických prostředí pro počítačem podporovanou výuku matematiky*. University of South Bohemia, Budweis, Czech Republic.