GCLC/WinGCLC

A Workbench for Geometry... and More...

— *system presentation* —

Predrag Janičić
Faculty of Mathematics
University of Belgrade

Special Semester on Gröbner Bases
Mini-Workshop on Gröbner Bases and Theorem Proving in Geometry
RICAM/RISC, Linz, May 14, 2006.

The main purposes of GCLC/WinGCLC:

- visualizing geometry (and not only geometry);
- producing digital mathematical illustrations of high quality;
- use in teaching geometry;
- use in studying geometry and as a research tool.

Name of the Game:

- Originally, a tool for producing geometrical illustrations for LaTeX, hence the name GCLC:

  "Geometry Constructions $\rightarrow$ LaTeX Converter".

GCLC: History and Releases

- First release in 1996, graphical interface for Windows in 2003
- Theorem prover built-in in 2006.
- Freely available releases for DOS, Windows, Linux
- `http://www.matf.bg.ac.yu/~janicic/gclc`
- Also available from emis (The European Mathematical Information Service) servers `http://www.emis.de/misc/index.html`

GCLC: Basic principles

- A construction is a formal procedure, not an image
- Producing mathematical illustrations should be based on "describing figures" rather than of "drawing figures"
- The image can be produces from a procedure, but not vice-versa!
- All instructions are given explicitly, within descriptions in GCLC language.

GCLC: Ruler and Compass

- A geometrical construction is a sequence of primitive construction steps, *elementary constructions*, performed by ruler and compass.

- GCLC provides support also for many non-constructible objects (not only geometrical — for instance, function graphs).

Features

- Support for a range of constructions and transformations;
- Conics, parametric curves, symbolic expressions, while-loops;
- User-friendly interface, interactive work, animations, traces;
- free, simple, easy to use, very small in size;
- import/export to different formats.
- built-in theorem prover.

GCLC language (part I):

- instructions for describing content;
- instructions for describing presentation;
- all of them are explicit, given within GCLC documents.

GCLC language (part II):

- basic definitions;
- basic constructions;
- transformations;
- drawing commands;
- commands for calculations, expressions, and loops;

GCLC language (part III):

- labelling and printing commands
- Cartesian commands
- low level commands
- commands for describing animations
- commands for the geometry theorem prover

Simple example (part I):

```
% fixed points
point A 15 20
point B 80 10
point C 70 90

% side bisectors
med a B C
med b A C
med c B A

% intersections of bisectors
intersec O_1 a b
intersec O_2 a c
distance d O_1 O_2
```

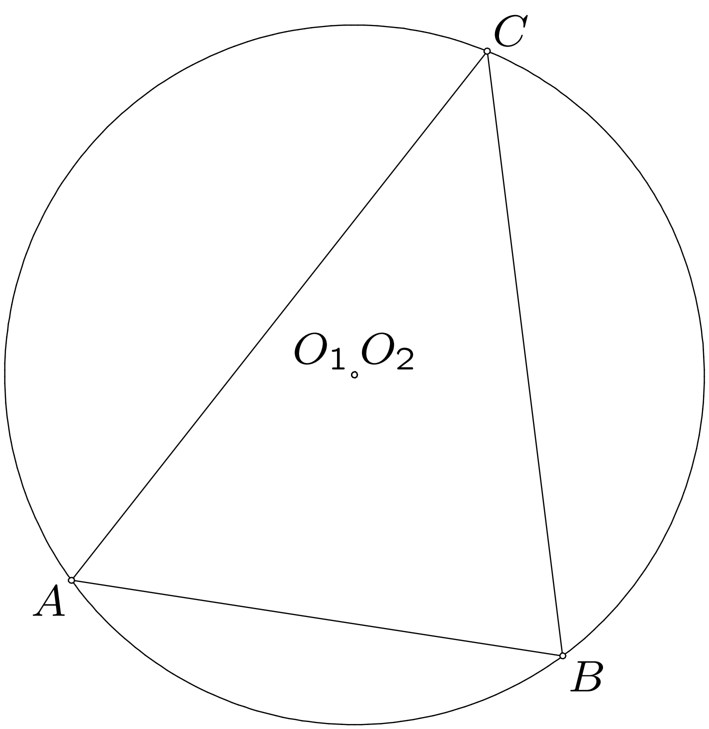
Simple example (part II):

```
% labelling points
cmark_lb A
cmark_rb B
cmark_rt C
cmark_lt O_1
cmark_rt O_2

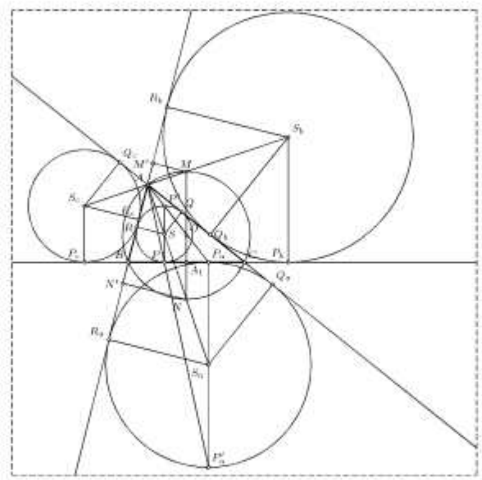
% drawing the sides of the triangle ABC
drawsegment A B
drawsegment A C
drawsegment B C

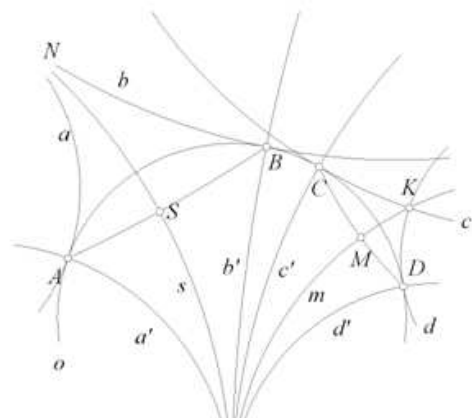
% drawing the circumcircle of the triangle
drawcircle O_1 A
```

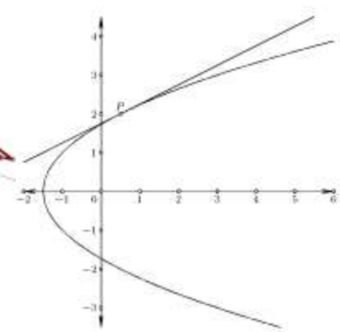
Simple example (part III):

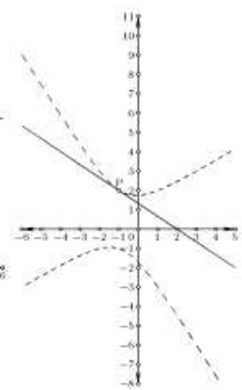


Samples:

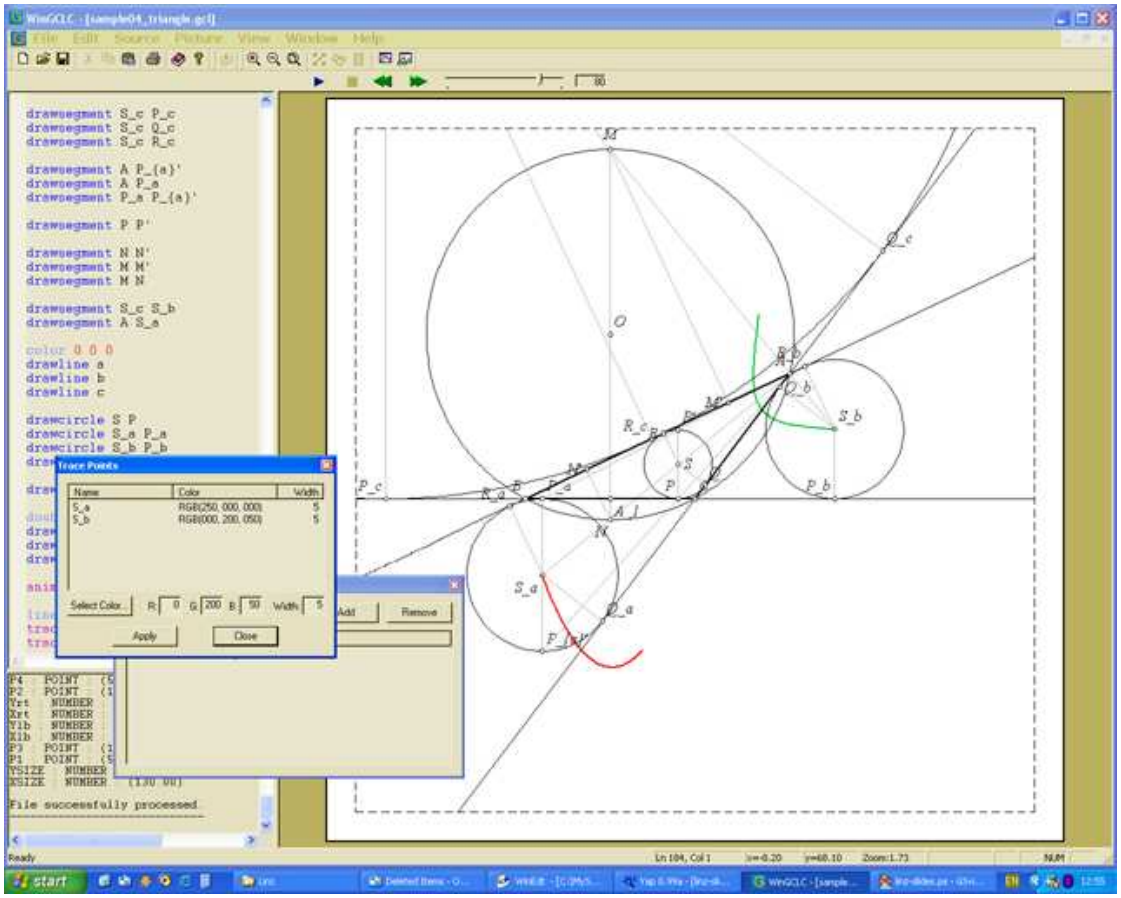








WinGCLC screenshot:



Built-in Theorem prover

- Joint work with Pedro Quaresma, University of Coimbra.
- Based on the area method (Chou et. al., mid 90's).
- Produces synthetic, traditional, human-readable proofs.
- Proofs generated in LaTeX with explanations for each step.

Properties of the area method:

- Wide realm, covers many non-trivial theorems.
- The procedure is a decision procedure for a fragment of geometry.
- The method does not have any branching.
- The method can transform a conjecture given as a geometry quantity of a degree d, involving n constructed points, to a quantity not involving constructed points, and with a degree at most $5d3^{5n}$, while this number is usually much less, and not reached also thanks to the used simplification procedures.

All proof steps made explicit:

- elimination steps (eliminate constructed points in reverse order, by using appropriate lemmas)

- algebraic simplifications (e.g. $x + 0 \to x$, $\frac{x}{y} + \frac{u}{v} \to \frac{x \cdot v + u \cdot y}{y \cdot v}$);

- geometric simplifications (e.g., $P_{AAB} \to 0$, $S_{ABC} \to S_{BCA}$).

- proofs given in layers

Using the theorem prover

- For the given example, points `O_1` and `O_2` are identical. It can be stated as follows (since $P_3(A,B,C) = AB^2 + CB^2 - AC^2$):

- ```
  prove { equal
            { pythagoras_difference3 O_1 O_2 O_1 }
            0
        }
  ```
  ```

Fragment of the Proof

$$(0.062500 \cdot (P_{CBC} \cdot S_{BAC})) = \left(\frac{1}{4} \cdot \left(P_{CBM_a^0} \cdot S_{BAM_a^0}\right)\right) \quad \text{, by algebraic simplifications} \tag{113}$$

$$(0.062500 \cdot (P_{CBC} \cdot S_{BAC})) = \left(\frac{1}{4} \cdot \left(\left(P_{CBB} + \left(\frac{1}{2} \cdot (P_{CBC} + (-1 \cdot P_{CBB}))\right)\right) \cdot S_{BAM_a^0}\right)\right) \quad \text{, by Lemma 29 (point } M_a^0 \text{ eliminated)} \tag{114}$$

$$(0.062500 \cdot (P_{CBC} \cdot S_{BAC})) = \left(\frac{1}{4} \cdot \left(\left(0 + \left(\frac{1}{2} \cdot (P_{CBC} + (-1 \cdot 0))\right)\right) \cdot S_{BAM_a^0}\right)\right) \quad \text{, by geometric simplifications} \tag{115}$$

$$(0.062500 \cdot S_{BAC}) = \left(\frac{1}{8} \cdot S_{BAM_a^0}\right) \quad \text{, by algebraic simplifications} \tag{116}$$

$$(0.062500 \cdot S_{BAC}) = \left(\frac{1}{8} \cdot \left(S_{BAB} + \left(\frac{1}{2} \cdot (S_{BAC} + (-1 \cdot S_{BAB}))\right)\right)\right) \quad \text{, by Lemma 29 (point } M_a^0 \text{ eliminated)} \tag{117}$$

$$(0.062500 \cdot S_{BAC}) = \left(\frac{1}{8} \cdot \left(0 + \left(\frac{1}{2} \cdot (S_{BAC} + (-1 \cdot 0))\right)\right)\right) \quad \text{, by geometric simplifications} \tag{118}$$

$$0 = 0 \quad \text{, by algebraic simplifications} \tag{119}$$

Integration:

- The prover tightly integrated into GCLC
- GCLC and GCLCprover share the parsing module
- GCLC file should be augmented only by a single line with a conjecture

Experimental Results:

Theorem Name	elim.steps	geom.steps	alg.steps	time
Ceva	3	6	23	0.001s
Gauss line	14	51	234	0.029s
Thales	6	18	34	0.001s
Menelaus	5	9	39	0.002s
Midpoint	8	19	45	0.002s
Pappus' Hexagon	24	65	269	0.040s
Ratio of Areas of Parallelograms	62	152	582	0.190s
Triangle Circuncircle	50	104	43	0.028s
Distance of a line containing the centroid to the vertices	274	673	3196	8.364s

Automatic verification of regular constructions:

- The system for automated testing whether a construction is regular or illegal;

- For instance — constructing a line l determined by (identical points) O_1 and O_2 from the above example is not a regular construction step;

- Test is made by the theorem prover and the argument is given as a syntectic proof (the only such geometry tool?).

GeoThms:

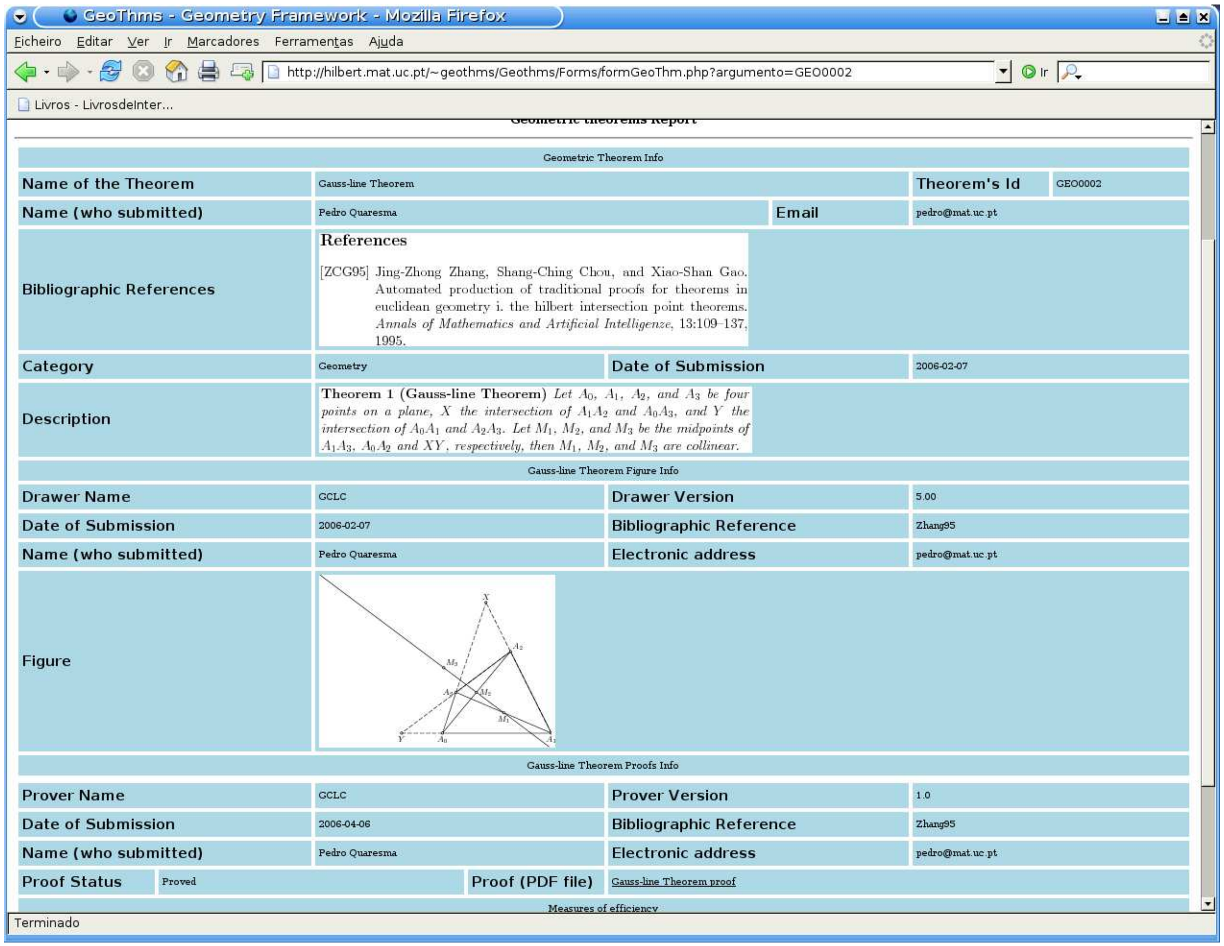
- Main author: Pedro Quaresma (University of Coimbra)

- An Internet framework that links dynamic geometry software (GCLC, Eukleides), geometry theorem provers (GCLCprover), and a repository of geometry problems (geoDB).

- A user can easily browse through the list of geometric problems, their statements, illustrations and proofs.

- `http://hilbert.mat.uc.pt/~geothms`

GeoThms screenshot:



GeoThms - Geometry Framework - Mozilla Firefox

Ficheiro Editar Ver Ir Marcadores Ferramentas Ajuda

http://hilbert.mat.uc.pt/~geothms/Geothms/Forms/formGeoThm.php?argumento=GEO0002

Livros - LivrosdeInter...

Geometric theorems Report

Geometric Theorem Info			
Name of the Theorem	Gauss-line Theorem	Theorem's Id	GEO0002
Name (who submitted)	Pedro Quaresma	Email	pedro@mat.uc.pt
Bibliographic References	References [ZCG95] Jing-Zhong Zhang, Shang-Ching Chou, and Xiao-Shan Gao. Automated production of traditional proofs for theorems in euclidean geometry i. the hilbert intersection point theorems. *Annals of Mathematics and Artificial Intelligenze*, 13:109–137, 1995.		
Category	Geometry	Date of Submission	2006-02-07
Description	**Theorem 1 (Gauss-line Theorem)** *Let A_0, A_1, A_2, and A_3 be four points on a plane, X the intersection of A_1A_2 and A_0A_3, and Y the intersection of A_0A_1 and A_2A_3. Let M_1, M_2, and M_3 be the midpoints of A_1A_3, A_0A_2 and XY, respectively, then M_1, M_2, and M_3 are collinear.*		
Gauss-line Theorem Figure Info			
Drawer Name	GCLC	Drawer Version	5.00
Date of Submission	2006-02-07	Bibliographic Reference	Zhang95
Name (who submitted)	Pedro Quaresma	Electronic address	pedro@mat.uc.pt
Figure			
Gauss-line Theorem Proofs Info			
Prover Name	GCLC	Prover Version	1.0
Date of Submission	2006-04-06	Bibliographic Reference	Zhang95
Name (who submitted)	Pedro Quaresma	Electronic address	pedro@mat.uc.pt
Proof Status	Proved	Proof (PDF file)	Gauss-line Theorem proof

Measures of efficiency

Terminado

XML support:

- format for descriptions of constructions and proofs;
- potentially common interchange format for different tools for geometrical constructions;
- Web presentation, in different forms.

XML construction:

Description of construction:

Let us define the following fixed points:

- Let *P1* be a point with Cartesian coordinates (5.000000, 5.000000).
- Let *P3* be a point with Cartesian coordinates (125.000000, 125.000000).

Let us draw the following objects:

- Visible area: left-bottom corner (5.000000,5.000000), right-top corner (125.000000, 125.000000).

Let us define the following fixed points:

- Let *P2* be a point with Cartesian coordinates (125.000000, 5.000000).
- Let *P4* be a point with Cartesian coordinates (5.000000, 125.000000).

Let us draw (using dashed style) the following objects:

- The segment with endpoints *P1* and *P2*.
- The segment with endpoints *P2* and *P3*.
- The segment with endpoints *P3* and *P4*.
- The segment with endpoints *P4* and *P1*.

Let us define the following fixed points:

- Let *B* be a point with Cartesian coordinates (35.000000, 60.000000).
- Let *C* be a point with Cartesian coordinates (65.000000, 60.000000).
- Let *A* be a point with Cartesian coordinates (40.000000, 80.000000).

Let us construct the following objects:

XML proof:

Step 1

(3.000000) * (s4(A, B, C, D)) = (13.000000) * (s3(A, B, A_2))

the statement

Semantic values: 4500.000000 = 4500.000865

Step 2

(3.000000) * (s4(A, B, C, D)) = (13.000000) * ((((s3(A, B, B_1)) * (s3(A, B, A_1))) + ((-1.000000) * ((s3(A_1, B, B_1)) * (s3(A, B, A))))) / (s4(A, B, A_1, B_1)))

Lemma 30 (point A_2 eliminated)

Semantic values: 4500.000000 = 4500.000865

Step 3

(3.000000) * (s4(A, B, C, D)) = (13.000000) * ((((s3(A, B, B_1)) * (s3(A, B, A_1))) + ((-1.000000) * ((s3(A_1, B, B_1)) * (0.000000)))) / (s4(A, B, A_1, B_1)))

Lemma 2 (equal)

Semantic values: 4500.000000 = 4500.000865

Step 4

(3.000000) * (s4(A, B, C, D)) = (13.000000) * ((((s3(A, B, B_1)) * (s3(A, B, A_1))) + ((-1.000000) * (0.000000))) / (s4(A, B, A_1, B_1)))

multiplication by 0

Semantic values: 4500.000000 = 4500.000865

Step 5

(3.000000) * (s4(A, B, C, D)) = (13.000000) * ((((s3(A, B, B_1)) * (s3(A, B, A_1))) + (0.000000)) / (s4(A, B, A_1, B_1)))

multiplication by 0

Semantic values: 4500.000000 = 4500.000865

Further Work:

- support for additional mathematical objects;
- additional import/export options;
- additional theorem provers and decision procedures (not only geometrical);
- enabling moving along/packing/unpacking parts of a proof;
- additional options for interactive work, especially developing automatic tutors.