

A BRIEF TOUR TO DYNAMIC GEOMETRY SOFTWARE

BOYKO B. BANTCHEV

February, 2010

What is dynamic geometry software?

A dynamic geometry (DG) program is a computer program for interactive creation and manipulation of geometric constructions. A characteristic feature of such programs is that they build a geometric model of objects, such as points, lines, circles, etc., together with the dependencies that may relate the objects to each other. The user can manipulate the model by moving some of its parts, and the program accordingly – and instantly – changes the other parts, so that the constraints are preserved.

By contrast with programs that just create images, a drawing in a DG program is a visualisation of an *abstract model* (of geometric nature) and, in particular, provides a visual interface for its manipulation. DG programs vary significantly in their drawing capabilities, but they are all centred around geometric modelling. A geometric model may be used for visualising complex geometric data, for doing calculations – including symbolic, for building and testing geometric hypotheses, or for creating geometrically precise illustrations to be used in printed documents or on the Web.

Model building in most DG systems starts with creating a set of independent, freely existing objects – usually points, and proceeds by constructing ones that are dependent on the former through being geometrically related to them.

Most geometry programs currently in use are designed for planar geometry, but a few allow for spatial constructions as well.

Some of the systems automate proving geometry theorems, or provide assistance in finding such proofs.

Uses & features of DG programs

Typical uses of DG programs are:

- graphical presentation of geometry on the screen;
- exploring geometric properties, testing hypotheses;
- visualising complex data;
- geometric reasoning;
- illustrations in document preparation;
- illustrations for the Web;
- libraries for geometric programming.

There are a number of features that a DG program may possess or lack. Choosing a particular program is best done by listing and considering as many of them as possible, in order to reduce the risk of subsequent dissatisfaction of the choice. Experimental using of different such systems before making a choice is also highly advisable.

Here is a list of the most important features that one should be interested in with respect to a dynamic geometry system:

- 2D or 3D;
- constructive richness;
- easy to use interface and other convenience-related issues;
- kinds and degrees of dynamicism (incl. animation);

- adaptability to specific domains and needs;
- accepting text commands;
- extensibility (through programming);
- observability of the constructive dependencies (what, *how*);
- reusability (of parts or techniques);
- portability to foreign environments: Web, general text processing tools, document preparation systems, automatic provers;
- independence of operating system and other elements of the operating environment;
- representations, file formats:
 - raster (PNG, GIF, ...),
 - vector (PostScript (incl. EPS), SVG, ...),
 - animation, interaction (GIF, SMIL, JAVA applets, SVG + JAVASCRIPT),
 - specialized (e.g. applets, scripts);
- ability to talk to the user in different languages;
- uses and users (many is always better).

Convenience of use deserves special attention because it is very important in two ways. First, it is something that manifests itself in every single act of using a system, and thus immensely affects that system's usability. Moreover, it consists of many factors, some of which non-obvious, and that makes it difficult to estimate it, unless one spends some time using the system and gaining experience of his own.

As an example of inconvenience, in one of the systems listed in the next section, a figure depending on some independent points, say, a circle constructed by a centre and a point on it, cannot be dragged along with the points (i.e. the points cannot be dragged together), which hinders constructing new objects to the left of the given figure, if it itself is near the left edge of the drawing field.

It should be stressed that, as with other kinds of complex software, it is difficult to foresee the ways in which the features of a DGS may need to be combined in order for the user to achieve certain goals. Nor is it possible to anticipate all the different uses that a system may have. Therefore, in general, systems of great flexibility should be preferred over the more rigid ones, even when the latter attract by offering specific functionality that may be missing in the former.

A particularly attractive form of flexibility is the system accepting text commands (in addition to the 'visual' mode of communication) – that makes that system programmable, in a sense. Taking into account the possibility of generating the input programmatically, perhaps by another program, the system is even meta-programmable!

A textual command language not only opens possibilities for programming and meta-programming; it is a means of formal representation of the geometric model, enabling and fostering reuse, among other benefits. Note that reusability may be direct (parts of the model) as well as indirect (techniques, ideas).

A major lever of control and flexibility is provided by making a system extensible by means of programming. This comes in two varieties. One is writing procedures in a scripting language interpreted within the system (such a language accommodates and extends the command language of the system). The other is writing extension modules in the language in which the system itself is implemented, or in an independent scripting language communicating with the implementation language through an established program interface. In each case the added functionality becomes available to the user as a harmonious part of the DG system, through attaching it to menus, key buttons etc.

Among the formats for representing drawings to the outside world, those oriented to Web should be specially mentioned. JAVA applets retain the interactivity of the original drawing and, often, a part of the functionality of the system that produced it. Interactive Web-based drawings can also be obtained by scripting SVG in JAVASCRIPT.

Popular DG programs

Here I list some widely popular dynamic geometry programs. Readers who read this through a program such as Acrobat Reader are invited to follow the links to the corresponding Web sites.

- THE GEOMETER'S SKETCHPAD

Created in the 1990s. Commercially distributed.

JAVASKETCHPAD is a part of THE GEOMETER'S SKETCHPAD that lets one publish SKETCHPAD sketches on the Internet and interact with them independently of SKETCHPAD. The program can be obtained for free, separately from SKETCHPAD.

There are many SKETCHPAD files (sketches) available on the Internet.

The program is widely used but lacks some of the features of its more modern rivals, and is not actively developed.

- CABRI

Created in the 1990s. Together with SKETCHPAD, one of the very first DG programs ever.

The current releases offer 3D construction capabilities.

Commercially distributed.

- CINDERELLA

A JAVA-based program. Has a scripting language equipped with data and control structures for general programming.

Commercially distributed but has a fully functional version that can be used for free.

- ARCHIMEDES GEO3D

An advanced 3D DG program.

Commercially distributed.

- GEOGEBRA

Freely distributed, JAVA-based (thus o.s.-independent).

See the next section for more details on this program.

- C.A.R.

The name is for *Compass and Ruler* or *Construct and Rule*; in German called Z.U.L., for *Zirkel und Lineal* or *Zeichnen und Lernen*.

A JAVA-based program simulating school geometry in the plane.

Offers a construction language. Constructions can be entered immediately through a command line or loaded from an external file.

CARMETAL is basically the same program but with a different, more easy to use interface, and also features 3D capabilities.

Freely distributed.

- GEOMETRIA

Another JAVA-based program.

Works both in 2D and 3D.

Freely distributed.

- POLYHEDRON

A collection of 250 problems built into a computer program and dealing with polyhedra. Simulates ruler, protractor, compass, bisector, and other construction and measuring tools.

An MS DOS program.

Freely distributed.

- WINGEOM
 Constructions in both plane and space.
 A part of **Peanut Software** – an educational program set.
 Freely distributed.
- POLY
 A program for construction and exploration of polyhedra.
 Distributed as shareware.
- GEONEXT
 A modest JAVA program whose distinguishing property is that it is executed as an applet in a Web browser, and can be run directly from a website, without downloading and installation.
 GEONEXT is known to have been used successfully in the elementary and high schools of some countries. Compared to most other DG programs of today, however, this one lags behind in functionality and other aspects of usability.
 Freely distributed.
- DR. GEO
 For young puples.
 Scriptable in the SCHEME programming language (using GUILLE).
 DR. GEO II is a rewrite of DR. GEO for the SQUEAK/SMALLTALK environment.
 Freely distributed.
- KSEG
 A small but nevertheless rather useful program.
 Freely distributed.
- KIG
 Not among the most advanced DG programs but can be extended through scripting in the PYTHON programming language.
 Freely distributed.
- VRR!
 Advantages of this program are that in creating drawings it can cooperate with T_EX, and that it can be scripted in the SCHEME programming language.
 The development seems to have stalled.
 Freely distributed.
- MATHKIT
 The original title is, in latin-transliterated Russian, *Matematicheskii konstruktor*, or *Mathematical Construction Kit* in English.
 Commercially distributed, but there is a freely available version available.
- GEOMVIEW
 Not really a DG program but can be used similarly.
 It is a 3D viewer for Unix/Linux (and Windows with the *Cygwin* package). Can be used as a standalone viewer for static objects or as a display engine for programs which produce dynamically changing geometry.
 Freely distributed.
- GAVIEWER
 Interactive geometric algebra program with OPENGL visualisation.
 Freely distributed.

- CLUCALC/CLUVIZ

A program for 3D visualisation and scientific calculation, originally for geometric algebra.

CLUCALC interprets a script language, CLUSCRIPT, and CLUVIZ is a visualisation engine.

Freely distributed.

- DPGRAPH

A 3D graphing software for visualisation in mathematics and physics.

Commercially distributed but there is a freely available viewer for DPGRAPH files.

- ΕΥΚΛΕΙΔΗΣ (EUKLEIDES)

An Euclidean geometry drawing language: a compiler to \LaTeX , and a GUI front-end.

Freely distributed.

- GEOMETRY EXPERT (GEX)

According to its author, GEX is ‘a software for dynamic diagram drawing and automated geometry theorem proving and discovering’.

JAVA GEOMETRY EXPERT (JGEX) is the JAVA version of GEX, and seems to be the only one that is actively developed.

Both GEX and JGEX are freely distributed.

- GEOMETRY EXPRESSIONS

Rather than being construction-based – proceeding from free to dependent object constructions, and ones that depend on the dependent – the geometry models created by this program are constraint-based.

A model is specified by drawing a sketch – an inaccurate representation – and then attaching symbolic constraints and adding constructions (dependent objects) to it. In order to rebuild the drawing according to the constraints, the system internally creates a construction sequence for the model (such as ‘create a point A , then create B such that ...’, etc.) and executes it, supplying sample numerical values for the free variables remaining in the definition.

Once a model is created, the program can answer – again, symbolically – requests for computing lengths, angles, and areas. It can also construct midpoints, bisectors, tangents, perpendiculars, etc. dependent objects.

For example, a sketch may contain a triangle ABC and a point M on AB ; a possible set of constraints to be attached is $CA = a$, $CB = b$, $\angle ACB = 90^\circ$, $AM = MB$. Then, e. g., the system would be able to deduce that $AM = \sqrt{a^2 + b^2}/2$.

Commercially distributed.

- GCLC

The name is for *Geometry Constructions* \rightarrow \LaTeX converter – the program was originally created as a tool for producing \LaTeX illustrations.

Not truly an interactive geometry system but, given a description of a geometric scene, GCLC can draw it, find locuses, perform animations, and prove or refute propositions about the participating objects. A drawing can be exported to \LaTeX and other formats.

Freely distributed.

- **GEOPROOF**

A program for proving theorems in geometry.

See also *about GeoProof and references*, *slides on GeoProof*, and *more of the author of the program*.

Freely distributed.

Symbolic mathematics systems

The so called symbolic mathematics systems (a.k.a. computer algebra systems), such as MATHEMATICA, MATLAB, MATHCAD, and MAPLE, although not geometry systems per se, provide means for drawing graphs of functions and doing simple geometry constructions. Most of these programs are commercial, and rather expensive products.

Freely available software of this kind includes OCTAVE, SAGE, and MAXIMA. OCTAVE is essentially (an interpreter of) a programming language similar to the language of MATLAB and better than it, that generates a script in the GNUPLOT plotting language. SAGE is scripted in the widely available and popular PYTHON programming language, which makes it extensible to many kinds of applications.

GeoGebra: a modern, free, dynamic geometry system

GEOGEBRA is a typical modern DG system for planar geometry. It is also perhaps the most recommendable among the freely available ones, being also well ahead some of its commercial rivals, e. g. SKETCHPAD.

Like several other DG systems, GEOGEBRA is written in JAVA. That makes it work and look the same way on any operating system with a JAVA virtual machine available.

The following features characterize GEOGEBRA, making it stand out against many of the other planar DG programs.

- rich, well thought-out set of constructive and other commands;
- easy to learn and use, ergonomic interface w. r. t. command access, object and attribute manipulation, etc;
- lack of arbitrary limitations, e. g. all parts of the drawing are draggable and always accessible for adding, changing or deletion of attributes;
- textual constructive language, with elements for programming, such as conditional construction and data structures;
- observability of all aspects of the model (entities, dependencies, construction protocol, values) and of the representation (visibility, colours, widths, ...);
- extensibility through programming in JAVA and JAVASCRIPT;
- output formats for different uses, such as e. g. printable documents and Web.

The GEOGEBRA system is actively used and developed by a large and committed user community. Updates appear regularly.

A number of educational and other applications (extensions) have been developed and are also freely available.