GeoGebra Manual

The official manual of GeoGebra.

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Thu, 15 Dec 2011 01:15:41 UTC

Contents

Articles

Objects

Introduction

What is GeoGebra

GeoGebra [1] is open source dynamic mathematics software for learning and teaching at all levels. This manual covers the commands and tools of GeoGebra 4.0.

Create dynamic constructions

Constructions in GeoGebra consist of mathematical objects of several types which can be created using tools or commands. The tutorials may guide you through your first constructions.

Objects

- Geometric Objects
- General Objects
- Action Objects
- Object Properties
- Naming Objects
- Labels and Captions
- Selecting objects
- Change Values
- Animation
- Tracing
- Advanced Features
- Scripting

Tools

- About tools
- List of tools

Commands

- About commands
- List of commands

Expressions

- Predefined Functions and Operators

Get to grips with GeoGebra's user interface

The main window is divided to views. By default Algebra View is displayed on the left side and Graphics View on the right. Above these views there is a Menubar and Toolbar, underneath Navigation Bar can be placed. Many features of GeoGebra can be accessed via Keyboard Shortcuts. GeoGebra also includes accessibility features such as Virtual Keyboard.

Main components

- Menubar
- Toolbar
- Context Menu
- Navigation Bar
- Virtual Keyboard
- Input Bar

Menus

- File Menu
- Edit Menu
- View Menu
- Options Menu
- Tools Menu
- Window Menu
- Help Menu

Views

- Algebra View
- CAS View
- Graphics View
- Spreadsheet View

Dialogs

- Properties Dialog
- Construction Protocol
- Tool Creation Dialog
- Tool Manager Dialog
- Redefine Dialog
- Options Dialog
- Export Graphics Dialog
- Export Worksheet Dialog
- Print Preview Dialog

Publish your work

- Share your dynamic worksheets online at GeoGebraTube [2]
- Print your construction, possibly together with the Construction Protocol
- Save image files in various formats

Troubleshooting

- The Installation Guide helps you with installation questions on different platforms
- The Compatibility page explains small differences between GeoGebra versions
- Visit our GeoGebra User Forum [3] if you have any questions or suggestions

References

[1] http://www.geogebra.org
[2] http://www.geogebratube.org
[3] http://www.geogebra.org/forum

Compatibility

GeoGebra is backward compatible in sense that every file created with older version should open flawlessly in the current one. There are however several things which behave differently in 3.2 and 4.0:

- lists of angles, integrals, barcharts, histograms etc. are now visible
- lists {Segment[A,B], Segment[B,C] } are now draggable
- circle with given radius (e.g. Circle[(1,1),2]) draggable
- Distance[Point, Segment] gives distance to the Segment (was to the extrapolated line in 3.2)
- Angle[A,B,C] now resizes if B is too close to A or C
- Integral[function f,function g,a,b] is now transcribed to IntegralBetween[function f,function g,a,b].
- Objects that are a translation by a free vector are now draggable, eg Translate[A, Vector[(1,1)]]

LaTeX issues

The LaTeX rendering is now nicer, but some errors in LaTeX syntax which were ignored in 3.2 will cause missing texts in 4.0.

- Make sure that each `\left\{` has corresponding `\right.`.
- The array environment needs specification of columns (although it may be empty). Please use `$ \begin{array}{} a & b \\ c & d \\ \end{array}$` for left aligned columns or `$ \begin{array}{rr} a & b \\ c & d \\ \end{array}$` for right-aligned ones. Old syntax `$ \begin{array} a & b \\ c & d \\ \end{array}$` wouldn't work any more.

Installation Guide

Webstart

Webstart reinstallation on Windows XP

- Start Menu, Run..., type "javaws -viewer" into the open field and press enter
- Right-click on GeoGebra -> Delete
- Re-run GeoGebra Webstart

Webstart reinstallation on Windows 7

- In the Start Menu type "javaws -viewer" into the search field and press enter
- Right-click on GeoGebra -> Delete
- Re-run GeoGebra Webstart

Webstart reinstallation on Windows Vista

- Disable UAC
- Restart computer
- In the Start Menu type "javaws -viewer" into the search field and press enter
- Right-click on GeoGebra -> Delete
- Re-run GeoGebra Webstart
- Turn UAC back on

Webstart reinstallation on a Mac

- Delete the GeoGebra.app from my Applications folder.
- Go into the Java Preferences -> Network -> View Cache Files and delete the GeoGebra.app file that is there and re-run GeoGebra Webstart

Webstart reinstallation on Linux

- Open a terminal
- javaws -viewer
- Right-click on geogebra.jnlp -> Delete
- Re-run GeoGebra Webstart

Applet Problems

First, check Java is working on your computer: http://www.java.com/en/download/help/testvm.xml

- Then Java Control Panel -> General -> Temporary Internet Files -> Settings -> Delete files...
- To get the Java Control Panel in Windows 7, open Control Panel then type "Java" in the search box (top right).

Associating .ggb files with Webstart (Windows)

- Start Menu -> Run -> javaws -verbose -import -shortcut -association http://www.geogebra.org/webstart/geogebra.jnlp

Problems with the offline installer (Windows) removing an old version of GeoGebra

- Start Menu -> Run -> explorer C:\Program Files\Zero G Registry
- Edit this file in Notepad: .com.zerog.registry.xml and remove the GeoGebra related bits

NB C:\Program Files\Zero G Registry is a hidden folder, so normally won't appear in C:\Program Files

Network install (Windows)

- Install on a standalone machine
- Copy the files from C:\Program Files\GeoGebra to the network
- Associate .ggb and .ggt files with GeoGebra.exe

Other error messages

Error message "Installer User Interface Not Supported" This is a problem when your Windows username contains unusual characters, eg !, # Solution: Create another user eg Test and install using that http://www.hauser-wenz.de/s9y/index.php?/archives/138-Installer-User-Interface-Mode-Not-Supported.html

Objects

Free, Dependent and Auxiliary Objects

There are two types of objects in GeoGebra: free and dependent. Some of them can be defined to be auxiliary.

Free objects

are objects whose position or value doesn't depend on any other objects. They are created by direct input or e.g. New Point Tool. They can be moved, unless they are fixed.

Dependent objects

are objects that depend on some other objects. They are created using tools and commands.

Auxiliary objects

are either objects which are defined to be auxiliary by user, or objects which were created by specific tools, e.g. Regular Polygon Tool. Spreadsheet cells are also considered to be auxiliary. They have their separate place in Algebra View.

Geometric Objects

GeoGebra works with many types of geometric objects

- Points and Vectors
- Lines and Axes
- Conic sections and Arcs
- Functions
- Curves
- Inequalities
- Intervals

Paths

Some of the above mentioned objects (lines, conic sections, arcs, polygons, functions, single variable inequalities, intervals, lists of points) are referred to as *paths*. One can define a point to belong to a path using the Point Command. Each point on a path has a path parameter, which is a number ranging from 0 to 1. To get this parameter, you can use the PathParameter Command.

Note: Lists of other paths are also paths.

Regions

You can also restrict points to a *region* (polygon, conic, arc, two variable inequality) using the PointIn Command or Point on Object Tool.

Note: See also Attach / Detach Point Tool.

Points and Vectors

Points and vectors may be entered via Input Bar in Cartesian or polar coordinates (see Numbers and Angles). Points can also be created using  Point tools,  Vector from Point Tool,  Vector between Two Points Tool and a variety of commands.

Note: Upper case labels denote points, whereas lower case labels refer to vectors. This convention is not mandatory.

Example:

To enter a point P or a vector v in Cartesian coordinates you may use P = (1, 0) or v = (0, 5). In order to use polar coordinates type in P = (1; 0°) or v = (5; 90°).

Note: You need to use a semicolon to separate polar coordinates. If you don't type the degree symbol, GeoGebra will treat the angle as if entered in radians.

Calculations

In GeoGebra, you can also do calculations with points and vectors.

Example:

You can create the midpoint M of two points A and B by entering M = (A + B) / 2 into the Input Bar. You may calculate the length of a vector v using length = sqrt(v * v)If A = (a, b), then A + 1 returns (a + 1, b + 1). If A is a Complex Numberscomplex number a+bί, then A+1 returns a + 1 + bί.

Vector Product

For two points or vectors `(a, b) ⊗ (c, d)` returns the z-coordinate of vector product *(a, b, 0)* □ *(c, d, 0)* as single number. Similar syntax is valid for lists, but the result in such case is a list.

Example:

{1, 2} ⊗ {4, 5} returns {0, 0, -3}{1, 2, 3} ⊗ {4, 5, 6} returns {3, 6, -3}.

Lines and Axes

Lines

You can enter a line as a linear equation in x and y or in parametric form into the Input Bar. In both cases previously defined variables (e. g. numbers, points, vectors) can be used within the equation.

Note: You can enter a line's name at the beginning of the input followed by a colon.

Example:

Type in g: 3x + 4y = 2 to enter line g as a linear equation. You can enter a line in parametric form thus: g: X = (-5, 5) + t (4, -3)Define the parameters m = 2 and b = -1. Then, you can enter the equationh: y = m*x + b to get a line h in y-intercept-form.

Axes

The two coordinate axes are available in commands using the names *xAxis* and *yAxis*.

Example: The command `PerpendicularLine[A, xAxis]` constructs the perpendicular line to the x-axis through a given point A.

Conic sections

You may enter a conic section as a quadratic equation in x and y. Prior defined variables (e. g., numbers, points, vectors) can be used within the conic's equation.

Note: The conic section's name can be entered at the beginning of the input, followed by a colon.

Examples

Conic section	Input
Ellipse ell	ell: 9 x^2 + 16 y^2 = 144
Hyperbola hyp	hyp: 9 x^2 – 16 y^2 = 144
Parabola par	par: y^2 = 4 x
Circle c1	c1: x^2 + y^2 = 25
Circle c2	c2: (x – 5)^2 + (y + 2)^2 = 25

Note: If you define two parameters *a = 4* and *b = 3* in advance, you may enter for example an ellipse as `ell: b^2 x^2 + a^2 y^2 = a^2 b^2`.

Functions

To enter a function you can use previously defined variables (e. g. numbers, points, vectors) as well as other functions.

Example:

- Function f: `f(x) = 3 x^3 – x^2`
- Function g: `g(x) = tan(f(x))`
- Nameless function: `sin(3 x) + tan(x)`

Note: All available predefined functions (e. g. sin, cos, tan) are described in section Predefined Functions and Operators.

In GeoGebra you can also use commands to get for example, the integral and derivative of a function. You can use If Command to get Conditional Functions.

Note: You can also use the commands f'(x) or f''(x), … in order to get the derivatives of a previously defined function f(x).

Example: Define function *f* as `f(x) = 3 x^3 – x^2`. Then, you can type in `g(x) = cos(f' (x + 2))` in order to get function *g*.

Furthermore, functions can be translated by a vector (see Translate Command) and a free function can be moved with the mouse by using the Move Tool. Other Transformation Commands can be also applied to functions, but in most cases the result is not a function but a curve.

Limit Function to Interval

In order to limit a function to an interval [a, b], you can use the Function Command or If Command.

Example: `If[x≥3 ∧ x≤5,x^2]` and `Function[x^2,3,5]` are both definitions of function x^2 restricted to interval [3,5]

Curves

There are two types of curves in GeoGebra.

Parametric curves

Parametric curves of the form a(t)=(f(t),g(t)) where *t* is real parameter within certain range can be created using the Curve Command. They can be used in Tangent Command and Point Command.

Note:

Parametric curves can be used with pre-defined functions and arithmetic operations. For example, input c(3) returns the point at parameter position 3 on curve c. Using the mouse you can also place a point on a curve using tool New Point ToolNew Point or command Point CommandPoint. Since the endpoints a and b are dynamic you can use slider variables as well (see tool Slider ToolSlider).

Creating parametric curve going through given points is not possible. You can however try e.g. FitPoly Command to get a function going through these points.

Implicit curves

Implicit curves are polynomials in variables *x* and *y*. The can be entered directly into Input Bar.

Example: x^4+y^3=2x*y

Inequalities

GeoGebra supports inequalities in one or two variables. There are no limitations for inequalities to appear in Algebra View, but only specific inequalities can be drawn in Graphics View:

- polynomial inequalities in one variable, e.g. *x^3 > x + 1* or *y^2>y*,
- quadratic inequalities in two variables, e.g. *x^2 + y^2 + x*y ≤ 4*,
- inequalities linear in one variable, e.g. *2x > sin(y)* or *y < sqrt(x)*.

For inequality sign you can use <, >, ≤, ≥. The Symbols <= and => also valid.

Inequalities are similar to functions, you can test whether *x* and *y* satisfy inequality *a* by typing *a(x,y)* in the Input Bar, also when *A* is a point, syntax *a(A)* is valid. A point can be restricted to the region given by inequality using PointIn Command. For inequality *b* in one variable, e.g. in *x*, Point[b] yields a point restricted to the part of x-axis which satisfies inequality *b*.

Conjunction and disjunction are also supported for inequalities, e.g. *(x>y)&&(x+y<3)* can be drawn.

Intervals

An interval is a set of numbers between upper and lower bound. To create an interval, type e.g. `2 < x < 3` in Input Bar. Interval in previous example is open. You can also define closed (`2 ≤ x ≤ 3`) and semi-closed (`2 ≤ x < 3`) intervals.

Note: See also Boolean values.

To determine whether number *c* belongs to interval *r* type `r(c)` into the Input Bar, the result will be a Boolean value. Generalization of intervals are Inequalities.

Commands for intervals

- Min, Max, Midpoint for an interval with lower bound *a* and upper bound *b* return numbers *a*, *b* and \(\frac{a+b}2\) respectively. The result doesn't depend on whether the interval is open, closed or semi-closed.
- Point returns a moveable point whose x-coordinate belongs to the interval and y-coordinate is 0.
- PointIn returns a moveable point whose x-coordinate belongs to the interval and y-coordinate may be changed arbitrarily.

General Objects

Besides Geometric Objects GeoGebra can also handle

- Numbers and Angles
- Complex Numbers
- Boolean values
- Lists
- Matrices
- Texts

Numbers and Angles

Numbers

You can create numbers by using the Input Bar. If you only type in a number (e. g., 3), GeoGebra assigns a lower case letter as the name of the number. If you want to give your number a specific name, you can type in the name followed by an equal sign and the number (e. g., create a decimal r by typing in r = 5.32).

Note: In GeoGebra, numbers and angles use a period (.) as a decimal point.

You can also use the constant π and the Euler constant e for expressions and calculations by selecting them from the drop down list next to the Input Bar or by using Keyboard Shortcuts.

Note: If the variable e is not used as a name of an existing object yet, GeoGebra will recognize it as the Euler constant if you use it in new expressions.

Angles

Angles are entered in degree (°) or radians (rad). The constant π is useful for radian values and can also be entered as pi.

Note: You can enter a degree symbol (°) or the pi symbol (π) by using the following keyboard shortcuts:

(Mac OS:) for the degree symbol ° (Mac OS:) for the pi symbol π

Example: You can enter an angle α in degree (e. g., $\alpha = 60°$) or in radians (e. g.,α = pi/3).

Note: GeoGebra does all internal calculations in radians. The degree symbol (°) is nothing but the constant $\pi/180$ used to convert degree into radians.

Example:

If a = 30 is a number, then $\alpha = a°$ converts number a to an angle $\alpha = 30°$, without changing its value. If you type in b = α / °, the angle α is converted back to the number b = 30, without changing its value.

Note: For dependent angles you can specify whether they may become reflex or not on tab Basic of the Properties Dialog.

Free Numbers and Angles

Free numbers and angles can be displayed as sliders in the Graphics View (see Slider Tool). Using the arrow keys, you may change the value of numbers and angles in the Algebra View too (see Manual Animation section).

Limit Value to Interval

Free numbers and angles may be limited to an interval [min, max] by using tab Slider of the Properties Dialog (see also Slider Tool).

Texts

Text objects can be easily created using Text Command or ABC Insert Text Tool, or dragging an object from the Algebra View to the Graphics View. Another way **for advanced users** (described below) is typing into Input Bar directly.

Static text

does not depend on any mathematical objects and is usually not affected by changes of the construction.

Dynamic text

contains values of objects that automatically adapt to changes made to these objects.

Mixed text

is a combination of static and dynamic text. In order to create a mixed text you may enter the static part of the text using the keyboard (e. g., Point A =). Then, click on the object whose value you want to display in the dynamic part of the text.

Note: GeoGebra automatically adds the syntax ("Point A = " + A) necessary to create your mixed text: quotation marks around the static part of the text and a plus (+) symbol to connect the different parts of the text.

Input	Description
This is static text	Static text
A	Dynamic text (if point A exists)
"Point A = " + A	Two-part mixed text using the value of point A
"a = " + a + "cm"	Three-part mixed text using the value of number a

Note: If an object with the name xx already exists and you want to create a static text using the object's name, you need to enter it with quotation marks ("xx"). Otherwise, GeoGebra will automatically create a dynamic text that gives you the value of object xx instead of its name. However, you can type any text that doesn't match any existing object's name without the quotation marks.

Note: Within a mixed text, the static part needs to be in between a pair of quotation marks. Different parts of a text (e. g., static and dynamic parts) can be connected using plus (+) symbols. Since 4.0, the + symbols are not mandatory.

Text objects can also use LaTeX for typesetting math.

Boolean values

You can use the Boolean variables *true* and *false* in GeoGebra. Just type, for example, `a = true` or `b = false` into the Input Bar and press the Enter-key.

Check Box and Arrow Keys

Free Boolean variables can be displayed as check boxes in the Graphics View (see tool Check Box to Show/Hide Objects Tool). By using the arrow keys of your keyboard you may also change Boolean variables in the Algebra View (see section Manual Animation).

Note: You may also use Boolean variables like numbers (value 0 or 1). This allows you to use a checkbox as the dynamic speed of an animated slider allowing you to start and stop the animation. In this case, the animation button is only shown in the Graphics View if there is also an animated slider with static (i. e. non-dynamic) speed.

Operations

You can use the following operations for Boolean variables and conditions in GeoGebra by either selecting them from the list next to the Input Bar or by entering them using the keyboard:

<table>
<tr><th>Operation</th><th>List</th><th>Keyboard</th><th>Example</th><th>Object types</th></tr>
<tr><td>Equal</td><td>▯</td><td>==</td><td>a ▯ b or a == b</td><td>numbers, points, lines, conics a, b</td></tr>
<tr><td>Unequal</td><td>≠</td><td>!=</td><td>a ≠ b or a != b</td><td>numbers, points, lines, conics a, b</td></tr>
<tr><td>Less than</td><td><</td><td><</td><td>a < b</td><td>numbers a, b</td></tr>
<tr><td>Greater than</td><td>></td><td>></td><td>a > b</td><td>numbers a, b</td></tr>
<tr><td>Less or equal than</td><td>≤</td><td><=</td><td>a ≤ b or a <= b</td><td>numbers a, b</td></tr>
<tr><td>Greater or equal than</td><td>≥</td><td>>=</td><td>a ≥ b or a >= b</td><td>numbers a, b</td></tr>
<tr><td>And</td><td>∧</td><td>&&</td><td>a ∧ b or a && b</td><td>Booleans a, b</td></tr>
<tr><td>Or</td><td>∨</td><td>||</td><td>a ∨ b or a || b</td><td>Booleans a, b</td></tr>
<tr><td>Not</td><td>¬</td><td>!</td><td>¬a or !a</td><td>Boolean a</td></tr>
<tr><td>Parallel</td><td>∥</td><td></td><td>a ∥ b</td><td>lines a, b</td></tr>
<tr><td>Perpendicular</td><td>⊥</td><td></td><td>a ⊥ b</td><td>lines a, b</td></tr>
<tr><td>Belongs to</td><td>∈</td><td></td><td>a ∈ list1</td><td>number a, list of numbers list1</td></tr>
</table>

Complex Numbers

GeoGebra does not support complex numbers directly, but you may use points to simulate operations with complex numbers.

Example: If you enter the complex number 3 + 4i into the Input Bar, you get the point (3, 4) in the Graphics View. This point's coordinates are shown as 3 + 4i in the Algebra View.

Note: You can display any point as a complex number in the Algebra View. Open the Properties Dialog for the point and select Complex Number from the list of Coordinates formats on tab Algebra.

If the variable i has not already been defined, it is recognized as the ordered pair i = (0, 1) or the complex number 0 + 1i. This also means, that you can use this variable i in order to type complex numbers into the Input Bar (e. g., q = 3 + 4i).

Example: Addition and subtraction:

(2 + 1i) + (1 − 2i) gives you the complex number 3 − 1i. (2 + 1i) - (1 − 2i) gives you the complex number 1 + 3i.

Example: Multiplication and division:

(2 + 1i) * (1 − 2i) gives you the complex number 4 − 3i. (2 + 1i) / (1 − 2i) gives you the complex number 0 + 1i.

Note: The usual multiplication (2, 1)*(1, -2) gives you the scalar product of the two vectors.

GeoGebra also recognizes expressions involving real and complex numbers.

Example:

3 + (4 + 5i) gives you the complex number 7 + 5i. 3 - (4 + 5i) gives you the complex number -1 - 5i. 3 / (0 + 1i) gives you the complex number 0 - 3i. 3 * (1 + 2i) gives you the complex number 3 + 6i.

Lists

Using curly braces you can create a list of several objects (e. g. points, segments, circles).

Example:

L = {A, B, C} gives you a list consisting of three prior defined points A, B, and C. L = {(0, 0), (1, 1), (2, 2)} produces a list that consists of the entered points, as well as these nameless points.

Note: By default, the elements of this list are not shown in the Graphics View.

To access particular elements of the list you can use Element Command. Lists can be used as arguments in list operations (mentioned further in this article) or List Commands.

Compare Lists of Objects

You can compare two lists of objects by using the following syntax:

- List1 == List2: Checks if the two lists are equal and gives you true or false as a result.
- List1 != List2: Checks if the two lists are not equal and gives you true or false as a result.

List Operations

<Object> ∈ <List> is an element of

<List> ⊆ <List> is subset of

<List> ⊂ <List> is subset of (strict)

<List> \ <List> set difference

Apply Predefined Operations and Functions to Lists

If you apply Predefined Functions and Operators to lists, you will always get a new list as a result.

Addition and subtraction

- List1 + List2: Adds corresponding elements of two lists. **Note:** The two lists need to be of the same length.
- List + Number: Adds the number to every element of the list.
- List1 – List2: Subtracts the elements of the second list from corresponding elements of the first list. **Note:** The lists need to be of the same length.
- List – Number: Subtracts the number from every element of the list.

Multiplication and division

- List1 * List2: Multiplies corresponding elements of two lists. **Note:** The lists need to be of the same length. If the two lists are compatible matrices, matrix multiplication is used.
- List * Number: Multiplies every list element with the number.
- List1 / List2: Divides elements of the first list by corresponding elements of the second list. **Note:** The two lists need to be of the same length.
- List / Number: Divides every list element by the number.
- Number / List: Divides the number by every element of the list.

Note: See also Vector product.

Other examples

- List^2: Squares every element of the list.
- 2^List: Creates list of powers of two with exponents from the list.
- List1^List2: Creates list of *a*^*b*, where *a* and *b* are corresponding elements of List1 and List2.
- sin(List): Applies the sine function to every element of the list. User defined functions can be applied the same way as well.

Matrices

GeoGebra also supports matrices, which are represented as a list of lists that contain the rows of the matrix.

Example: In GeoGebra, {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}} represents a 3x3 matrix.

In order to display nicely a matrix in the Graphic View, using LaTeX formatting, use FormulaText command.

Example: In the input bar type `FormulaText[{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}]` to display the matrix using LaTeX formatting.

Matrix Operations

Addition and subtraction examples

- Matrix1 + Matrix2: Adds the corresponding elements of two compatible matrices.
- Matrix1 – Matrix2: Subtracts the corresponding elements of two compatible matrices.

Multiplication examples

- Matrix * Number: Multiplies every element of the matrix by the given number.
- Matrix1 * Matrix2: Uses matrix multiplication to calculate the resulting matrix.

Note: The rows of the first and columns of the second matrix need to have the same number of elements.

Example: * gives you the matrix .

- 2x2 Matrix * Point (or Vector): Multiplies the matrix with the given point/vector and gives you a point as a result.

Example: * (3, 4) gives you the point A = (11, 25).

- 3x3 Matrix * Point (or Vector): Multiplies the matrix with the given point/vector and gives you a point as a result.

Example: * (1, 2) gives you the point A = (8, 20).

Note: This is a special case for affine transformations where homogeneous coordinates are used: (x, y, 1) for a point and (x, y, 0) for a vector. This example is therefore equivalent to: `{{1, 2, 3}, {4, 5, 6}, {0, 0, 1}} * {1, 2, 1}`.

Other examples

see also section Matrix Commands

- Determinant[Matrix]: Calculates the determinant for the given matrix.
- Invert[Matrix]: Inverts the given matrix
- Transpose[Matrix]: Transposes the given matrix
- ApplyMatrix[Matrix,Object]: Apply affine transform given by matrix on object.
- ReducedRowEchelonForm[Matrix]: Converts the matrix to a reduced row-echelon form

Action Objects

For interactive worksheets with scripting Action Objects may come handy. There are four types of them

Checkboxes

Checkboxes are graphical representations of Boolean values. See Check Box to Show / Hide Objects Tool for details. Checkboxes can be created using the aforementioned tool or Checkbox Command.

Input Boxes

Input Boxes work as text inputs for scripts. The script is triggered by changing text in the Input Box and either pressing enter or leaving the Input Box. The inserted value may be accessed using the %0 variable.

Example: Input Box with `a=a+%0` in script will increase number *a* by the entered value. Works only if *a* is free.

If you want the Input Box to change value of a free object (or redefine dependent object), you may define that object as linked. This way you don't have to insert any script. Input Boxes can be created using Insert Input Box Tool or InputBox Command.

Buttons

Buttons are meant to trigger scripts by being clicked. Although scripts can be triggered by clicking any other object (e.g. an image), using buttons for this makes your worksheet more intuitive. Input Boxes can be created using Insert Button Tool or Button Command.

Comboboxes

Comboboxes are available only in Spreadsheet View. You may enter a list into spreadsheet cell and then turn on option *Use Buttons and Checkboxes* in Spreadsheet Tab of Options Dialog. The list will turn into a combobox; the selected element may be obtained using SelectedIndex Command and SelectedElement Command.

Selecting objects

To select an object means to click on it with the mouse after selecting the Move Tool.

If you want to select several objects at the same time, you could draw a selection rectangle: Select the Move Tool and click on the position of the first corner of your desired selection rectangle. Hold the left mouse key pressed down and move the pointer to the position of the diagonally opposite corner of your desired selection rectangle. After releasing the mouse button, all objects within the selection rectangle are selected.
Note: By holding the Ctrl-key (Mac OS: Cmd-key) while clicking on different objects, you can select several objects at the same time.

Change Values

There are two ways of manipulating a free object’s value:

- Change the value of the object by entering its name and the new value in the Input Bar.

Example: If you want to change the value of an existing number a = 3, type a = 5 into the Input Bar and press the Enter-key.

- Edit the algebraic representation: Activate Move Tool and double click on the object in the Algebra View. This opens a text box where you can edit the object’s value. Press the Enter-key to apply your changes.

Note: While free objects’ values can be changed directly, the values of dependent objects can only be influenced by changing their parent objects or by redefining the dependent object.

Naming Objects

You can assign a certain name to an object when you create it using the Input Bar:

- Points: In GeoGebra, points are always named using upper case letters. Just type in the name (e. g., A, P) and an equal sign in front of the coordinates or commands.

Example: `C = (2, 4)`, `P = (1; 180°)`

- Vectors: In order to distinguish between points and vectors, vectors need to have a lower case name in GeoGebra. Again, type in the name (e. g., v, u) and an equal sign in front of the coordinates or commands.

Example: `v = (1, 3)`, `u = (3; 90°)`

- Lines, circles, and conic sections: These objects can be named by typing in the name and a colon in front of their equations or commands.

Example: `g: y = x + 3`, `c: (x-1)^2 + (y – 2)^2 = 4`, `hyp: x^2 – y^2 = 2`

- Functions: You can name functions by typing, for example, `f(x) =` or `g(x)=` in front of the function's equation or commands.

Example: `h(x) = 2 x + 4`, `q(x) = x^2, trig(x) = sin(x)`

Note:

If you don't manually assign a name to an object, GeoGebra assigns the names of new objects in alphabetical order. You can create indices within the names of objects by using an underscore. For example A1 is entered as A_1 and sAB is entered as s_{AB}.

Reserved labels

These are the labels you can't use for objects: x, y, xAxis, yAxis, zAxis, abs, sgn, sqrt, exp, log, ln, ld, lg, cos, sin, tan, acos, arcos, arccos, asin, arcsin, atan, arctan, cosh, sinh, tanh, acosh, arcosh, arccosh, asinh, arcsinh, atanh, arctanh, atan2, erf, floor, ceil, round, random, conjugate, arg, gamma, gammaRegularized, beta, betaRegularized, sec, csc, cosec, cot, sech, csch, coth

In the symbol list of the input bar, you will find special characters for the following constants:

- π - the circle constant pi, which you can also type with
- *e* - the Euler number, e.g. for the exponential function *e*^x, which you can also type with
- ί - the imaginary unit, e.g. for complex numbers like z = 3 + ί, which you can also type with

When the variable names e and i are not used yet, they are automatically read as *e* and ί respectively for convenience.

See also Labels and Captions.

Animation

Automatic Animation

GeoGebra allows you to animate not only free numbers and/or angles at the same time, but also dependent points that are constrained on an object (segment, line, function, curve, etc.). In order to be automatically animated, free numbers/angles need to be shown as sliders in the Graphics View.

If you want to animate a free number or angle in GeoGebra, as well as a dependent point, you need to right click (Mac OS: Ctrl-click) on the number, angle or point and select Animation On from the appearing Context Menu. In order to stop the animation, you need to un-check Animation On in the same Context Menu.

Note: After animating a free number, a free angle or a dependent point, an animation button appears in the lower left corner of the Graphics View. It allows you to either pause or continue an animation.

In the Properties Dialog on tab Slider you can change the behavior of the animation:

On the one hand, you may control the Speed of the animation.

Note: A speed of 1 means that the animation takes about 10 seconds to run once through the interval of the slider.

On the other hand, you can decide how the animation cycle is repeated:

⇔ Oscillating

The animation cycle alternates between Decreasing and Increasing.

⇒ Increasing

The slider value is always increasing. After reaching the maximum value of the slider, it jumps back to the minimum value and continues the animation.

⇐ Decreasing

The slider value is always decreasing. After reaching the minimum value of the slider, it jumps back to the maximum value and continues the animation.

Note: while an automatic animation is activated, GeoGebra remains fully functional. This allows you to make changes to your construction while the animation is playing.

Manual Animation

To manually change a number, angle or point position continuously, select the Move Tool. Then, click on a free number, angle or a dependent point and press either the + or – key or the arrow keys on you keyboard. Keeping one of these keys pressed allows you to produce manual animations.

Example: If the point coordinates depend on a number *t* like in P = (2 t, t), the point will move along a straight line when *t* is changed continuously.

Note: You can adjust the increment of the slider on tab Slider of the Properties Dialog of this object.

Keyboard Shortcuts

- Shift + arrow key gives you a step width of 0.1 units
- Ctrl + arrow key gives you a step width of 10 units
- Alt + arrow key gives you a step width of 100 units

Note: A point on a line can also be moved along that line using the + or – key.

Tracing

Objects can leave a trace in the Graphics View when they are moved. Use the Context Menu to switch this Trace On. Then, modify the construction so that the object whose trace you turned on changes its position and leaves a trace.

You can turn off the trace of an object by un-checking Trace On in the Context Menu. The menu item Refresh Views in the View Menu clears all trace.

Note: The trace is not permanent, it disappears e.g. on zoom. Instead of permanent trace of a point you can use Locus.

Tracing to Spreadsheet

When the Spreadsheet View is enabled, it is also possible to trace the changing positions of a point in the Graphics View. To do so, right-click on a point in the Graphics View, then click on Trace to spreadsheet option.

Object Properties

Following properties can be set via Properties Dialog.

Visibility

You may show or hide objects in the Graphics View in different ways.

- You may use tool Show/Hide Object Tool to show or hide objects.
- Open the Context Menu and select item Show Object to change the visibility status of the selected object.
- In the Algebra View, the icon to the left of every object shows its current visibility state (shown or hidden). You may directly click on the little marble icon in order to change the visibility status of an object.
- You can also use the Check Box to Show/Hide Objects Tool in order to show or hide one or several objects.

Note: To make an object "invisible" in Algebra View, make it Auxiliary.

Fixed objects

You can define an object to be fixed via Properties Dialog. Fixed objects (both free and dependent) cannot be moved, redefined or deleted.

Filling

For closed Curves and regions you can specify filling using the Style tab of Object properties dialog. There are three types of filling:

Standard

Fills the object by color specified in the Color tab. The same color is used to draw border of that object. Using Style tab you can define opacity -- e.g. Conics have by default opacity 0 which means they are transparent.

Hatch

The object is hatched, angle of hatches and distance between them can be specified. Thickness of hatches equals the thickness of object's border.

Image

You can specify location of image on the local disc. The image is repeated periodically, its size is fixed in pixels and top left corner is aligned to the top left corner of the view.

Advanced properties

Advanced properties such as Dynamic Colors and Conditional Visibility are listed in article Advanced Features.

Labels and Captions

In GeoGebra, each object has its unique **label**. For labeling you can choose one or more letters, possibly with subscript. For details see Naming Objects.

Show and Hide Labels

You can show or hide the objects labels in the Graphics View in different ways:

- Select the A A Show / Hide Label Tool and click on the object whose label you would like to show or hide.
- Open the Context Menu for the desired object and select A A Show Label.
- Open the Properties Dialog for the desired object and check or un-check the checkbox Show Label on tab Basic.

Name and Value

In GeoGebra, every object has a unique name that can be used to label the object in the Graphics View. In addition, an object can also be labeled using its value or its name and value. You can change this label setting in the Properties Dialog on tab Basic by selecting the corresponding option Name, Value, or Name & Value from the drop down menu next to the checkbox Show Label.

Note: The value of a point is its coordinates, while the value of a function is its equation.

Caption

However, sometimes you might want to give several objects the same label, for example, to label the four edges of a square `a`. In this case, GeoGebra offers captions for all objects, in addition to the three labeling options mentioned above. You can set the caption of an object on tab Basic of the Properties Dialog by entering the desired caption into the text field called "Caption". Afterwards, you can select the labeling option "Caption" from the drop down menu next to the checkbox "Show Label".

You can use following placeholders in captions:

Placeholder	Meaning
%v	Value
%n	Name
%x	x coordinate (or x coefficient for the line a x + b y + c = 0)
%y	y coordinate (or x coefficient for the line a x + b y + c = 0)
%z	the 'c' term for the line a x + b y + c = 0 (also: z-coordinate, ready for a 3D View)

Example: Let A be a point and (1,2) be its coordinates. Setting the caption to "Point %n has coordinates %v" results in caption "Point A has coordinates (1,2)"

LaTeX in Captions

You can also use LaTeX in your labels, enclosing the desired LaTeX command in dollar characters (eg **$ x^{2} $**), and choose from a list of most commonly used Greek letters and operators, just clicking on the little box displayed at the end of the Caption field.

Example: If you want to display a nicely formatted math text, use LaTeX in captions, e.g. to display a fraction, in the caption field type "$\frac{a}{b}$"

Note: LaTeX Captions don't work for Textfields, Buttons and Checkboxes

Advanced Features

For geometric objects, following properties can be found on the Advanced tab

- Layers
- Conditional Visibility
- Dynamic Colors
- Tooltips
- Object Position

For texts there is a powerful feature: LaTeX. It allows user to create nice looking mathematical formulas.

Object Position

Position of objects can be specified in the Position tab of Properties Dialog.

- For vectors, the position is specified by the start point.
- For images, the position is specified by one, two or three corners (see Image properties for details.)

Position of images and sliders may be fixed with respect to screen. This feature is by default enabled for sliders and disabled for images. To change it, switch *Absolute Position On Screen* in Basic Tab of Properties Dialog. The Action Objects have always absolute position on screen.

Conditional Visibility

Apart from just showing or hiding certain objects you can also have their visibility status depend on a certain condition. For example, you would like an object to appear on screen if you check a checkbox positioned in the Graphics View or if a slider is changed to a certain value.

Conditionally Show or Hide Existing Objects

You can use the Check Box to Show/Hide Objects Tool in order to create a checkbox that controls the visibility of one or more existing objects on screen.
Alternatively, you could also create a Boolean variable (e. g., `b = true`) using the Input Bar and make it visible as a checkbox in the Graphics View by changing its visibility status (e. g., use Show / Hide Object Tool or the Context Menu). In order to use this Boolean variable as a condition for the visibility of certain objects, you need to follow the steps described below.

Changing the Visibility of Newly Created Objects

In the Properties Dialog, you can enter a condition for the visibility of an object on tab Advanced.

Note: You can select the logic operators (e. g., ≠, ≥, ˆ, ∥) from the drop down list in order to create your conditional statements.

Example:

If a is a slider, then the conditional statement a < 2 means that the corresponding object is only shown in the Graphics View if the slider value is less than 2. If b is a Boolean variable, you can use b as a conditional statement. The corresponding object is shown whenever the value of b is true and is hidden when the value of b is false. If g and h are two lines and you would like a text to be shown whenever these lines are parallel, then you could use g ∥ h as a conditional statement for the text.

Dynamic Colors

In GeoGebra, you can change the color of objects using tab Color of the Properties Dialog. However, you can also have the color of an object change dynamically: Open the Properties Dialog for a certain object whose color you would like to change and click on tab Advanced. There you will find a section called Dynamic Colors with text boxes for the color components Red, Green, and Blue.

Note: In each of these text boxes, you can enter a function with range [0, 1].

Example:

Create three Slider Toolslider a, b, and c with an interval from 0 to 1. Create a polygon whose color should be influenced by the slider values. Open the Properties Dialog for the polygon poly1 and enter the names of the three sliders into the text boxes for the color components. Close the Properties Dialog and change the values of the sliders in order to find out how each color component influences the resulting color of the polygon.

Note: You could also animate the sliders with different speeds in order to see the color of the polygon change automatically.

The Dynamic Colors section also contains a text box which allows you to change the Opacity of the selected object. You can enter a number ranging in [0,1] (where 0 means transparent and 1 means 100% opaque), as well as a slider, in order to obtain a dynamic opacity. Other numbers will be ignored.

RGB / HSV / HSL

For some dynamic color behaviors it may be easier to use a different color model. In that case instead of default RGB select either HSV or HSL from the drop down list at the bottom of the Dynamic Colors section of the Advanced tab of the Properties dialog.

Example: To make a point *A* go through all colors of the rainbow when moved left and right, switch to HSV mode and let saturation and value be 1 and set hue to `x(A)`.

LaTeX

In GeoGebra you can write formulas as well. To do so, check the box LaTeX formula in the dialog window of the Insert Text Tool and enter your formula in LaTeX syntax.

Note: In order to create text that contains a LaTeX formula as well as static text you may enter the static part of the text and then add the LaTeX formula in between a set of dollar symbols ($).

Example: The length of the diagonal is $\sqrt{ 2 }$.

Note: You can simply obtain a LaTeX text containing the value of an object listed in the Algebra View by dragging that object in the Algebra View and dropping it in a selected location of the Graphics View .

You can select the syntax for common formula symbols from the drop-down menu next to the LaTeX checkbox. This inserts the corresponding LaTeX code into the text field and places the cursor in between a set of curly brackets. The Symbols drop-down menu contains a list of common math symbols, Greek letters and operators. If you would like to create dynamic text within the formula, you need to select the relating objects from the Objects drop-down list, causing GeoGebra to insert their names as well as the syntax for mixed text.

Some important LaTeX commands are explained in following table. Please have a look at any LaTeX documentation for further information.

LaTeX input	Result
a \cdot b	\(a \cdot b\)
\frac{a}{b}	\(\frac{a}{b}\)
\sqrt{x}	\(\sqrt{x}\)
\sqrt[n]{x}	\(\sqrt[n]{x}\)
\vec{v}	\(\vec{v}\)
\overline{AB}	\(\overline{AB}\)
x^{2}	\(x^{2}\)
a_{1}	\(a_{1}\)
\sin\alpha + \cos\beta	\(\sin\alpha + \cos\beta\)
\int_{a}^{b} x dx	\(\int_{a}^{b} x dx\)
\sum_{i=1}^{n} i^2	\(\sum_{i=1}^{n} i^2\)

Layers

Note: In GeoGebra, layers are used to determine which object to select or drag when the user clicks on multiple objects at the same time.

By default, all objects are drawn on layer 0, which is basically the background layer of the Graphics View. A total of 10 layers are available (numbers 0 to 9) and higher numbered layers are drawn on top of lower numbered layers.

Using the Advanced tab of the Properties Dialog, you can change the layer for a certain object (layers from 0 to 9 available). Once you change the layer number for at least one object to be different from layer 0 (e. g., to layer 3), all new objects will be drawn on the layer with the highest number that is used for any object.

Note: After selecting any object, you can select all objects in the same layer by selecting item Select Current Layer (keyboard shortcut:) from the Edit menu. This menu item is only available if all selected objects lie on the same layer.

Further use of layers

- For SVG export, objects are grouped by layer.
- Layers can be controlled using the JavaScript Interface for GeoGebra applets.

Scripting

Script is a sequence of commands, that are executed one after each other. GeoGebra supports two scripting languages - GGBScript and Javascript. The execution can be triggered by :

- clicking a particular object
- updating a particular object (when value or properties of the object are changed)
- loading the file (in case of JavaScript)
- Javascript listeners (see Reference:JavaScript)

You can set this script via Scripting panel of Properties Dialog.

GGBScript

You can create scripts consisting of GeoGebra commands, like you can use them in the Input Bar. After triggering the script, every command is executed one after each other.

Example:

- *a* is an integer-valued slider ranging from 1 to 3
- `list1 = {"red", "green", "blue"}`
- in properties of *a*, set "On Update" script to `SetColor[a, Element[list1, a]]`
- by moving the slider you change its color

Explanation: Every time the slider ist moved, there is happening an update. So, for every move the script is called and the value of "a" is used to get one color from the list and change the color of the slider "a".

Hint: There are commands that can be only used for scripting. You can find them in the page Scripting_Commands.

JavaScript

JavaScript is a programming language used by many Internet technologies. Unlike GeoGebra Script, in Javascript the commands don't have to be executed as a simple sequence, but a control flow (`if`, `while`, `for`) can be used. For generic JavaScript you can find a nice tutorial on developer.mozilla.org [1]. In GeoGebra, you can use special JavaScript methods which allow you to change the construction. These methods belong to ggbApplet object, which means that you call them as `ggbApplet.method_name(parameter,..,parameter)`. For complete list of these methods see Reference:JavaScript.

Example: for(var i =0;i<10;i++) ggbApplet.evalCommand("A_"+i+"=(random()*10,random()*10)");

This script creates 10 points A_0 to A_9 at random coordinates.

GeoGebra contains its own JavaScript engine. When exported as Dynamic Worksheet one can choose whether to use this engine or the one contained in browser applets. If you edit JavaScript in a HTML page, the ggbApplet variable will not be initialized, you have to initialize it e.g. using `ggbApplet=document.applets[0];` first.

Global JavaScript

In the Global JavaScript part of Scripting tab in Properties Dialog you may define some functions or do some assignments that will be done before the construction is loaded. You can also define function *ggbOnInit()*, which is called automatically once the construction is loaded. The ggbOnInit function can be used for registering some listeners, as shown below.

Example: function onAdd(name){ alert("Object "+name+" was added."); } function ggbOnInit(){ ggbApplet.registerAddListener("onAdd"); } First we defined function *onAdd* that given a string shows a message depending on that string. After that, using the *ggbOnInit* function, we told GeoGebra to call this function whenever a new object is added. Once we reload our construction, function ggbOnInit will be called and since then, when user adds a point named e.g. *A*, message "Object A was added" will appear.

You can also use listeners for actions like rename, delete and clear construction. Complete list is available in Reference:JavaScript.

Note: Using any ggbApplet methods in Global JavaScript outside of ggbOnInit will not work as intended since they will be called before the construction is loaded.

USB Data Logging (From GeoGebra 4.2)

For logging data from some Vernier USB Data Loggers, eg *Go!Motion* and *Go!Temp* one can define a logger listener using the registerLoggerListener method. Such listener can look like this:

```
function logger(value) {
   var d = value * 1;
   ggbApplet.evalCommand("(CopyFreeObject[a],"+d+")");
   ggbApplet.evalCommand("SetValue[a,a+1]");
}
```

This script assumes that there is a free number *a* in the construction. Each time number *d* is logged, point *(a,d)* is constructed and *a* is increased.

References

[1] https://developer.mozilla.org/en/JavaScript/Guide

Tooltips

Tooltips are texts that appear next to your mouse cursor when you hover the cursor over an object in Graphics View. In the Advanced tab of Properties Dialog you can specify five tooltip modes:

Automatic

Tooltips are shown if Algebra View only is active. Tooltip contains object type and name; in case of dependent objects the tooltip also includes object description.

On

Tooltips are shown whether Algebra View is shown or not. Content of the Tooltip is the same as for *Automatic*.

Off

No tooltip is shown.

Caption

Caption of the object is used as tooltip. You can set Caption in Basic tab of Properties Dialog.

Next Cell

If the object is a Spreadsheet cell, content of the cell to the right is used as tooltip.

In Advanced tab of Options Dialog you may also specify the language and timeout for tooltips.

Tools

Tools

Tools enable you to produce new objects using your pointing device. All of them have their commands equivalents which are suitable for more complicated constructions.

Construction tools or modes can be activated by clicking on the buttons of the Toolbar. You can click on the small arrow in the lower right corner of an icon to open a Toolbox with similar other tools. Other way of selecting tools is using the Tools Menu.

By default, the tools are ordered in twelve toolboxes:

- Movement Tools
- Point Tools
- Line Tools
- Special Line Tools
- Polygon Tools
- Circle and Arc Tools
- Conic Section Tools
- Measurement Tools
- Transformation Tools
- Special Object Tools
- Action Object Tools
- General Tools
- Custom Tools

You can reorder these toolboxes and save the setting in the GeoGebra Worksheet (*.ggb). See Customizing the Toolbar for details.

Note: With most construction tools you can easily create new points by clicking on empty spaces on the drawing pad.

In CAS View and Spreadsheet View you can use CAS Tools and Spreadsheet Tools respectively.

Movement Tools

Movement tools are by default grouped under icon (the first from left) in the toolbar. Currently there are three movement tools:

- Move
- Rotate around Point
- Record to Spreadsheet

Move Tool

Drag and drop free objects with the mouse. If you select an object by clicking on it in Move mode, you may…

- … delete the object by pressing the *Delete*-key
- … move the object by using the arrow keys (see section Manual Animation)

Note:

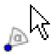
You can quickly activate the Move tool by pressing the Esc-key of your keyboard.To move a Slider Toolslider when Move Tool is selected, you need to drag it with your right mouse button.

Record to Spreadsheet Tool

This tool allows you to move an object and to record a sequence of its values in the Spreadsheet View. This tool works for numbers, points, and vectors.

Note: GeoGebra will use the first two empty columns of the Spreadsheet View to record the values of the selected objects.

Rotate around Point Tool



Select the center point of the rotation first. Then, you may rotate free objects around this point by dragging them with the mouse (see also Rotate command).

Point Tools

Point tools are by default grouped under  icon (the second from the left) in the toolbar. Currently there are five point tools:

- New Point
- Point in Region
- Intersect Two Objects
- Midpoint or Centre
- Attach / Detach Point
- Complex Number

New Point Tool

Click on the drawing pad in the Graphics View in order to create a new point. The coordinates of the point are fixed when the mouse button is released.

Note:

By clicking on a segment (or interval), straight line, polygon, conic section, function, or curve you can create a point on this object (see also Point CommandPoint command). Clicking on the intersection of two objects creates this intersection point (see also Intersect Two Objects ToolIntersect Two Objects tool and Intersect CommandIntersect command).

Attach / Detach Point Tool

To **attach a point** to a path or region click a free point and the path or region. From now on, the point can still be moved via Move Tool, but only within the path or region.
To **detach a point** that is defined as point on path or region simply click the point. The point will become free.

Note: You can also use Point Command and PointIn Command for attaching a point. See also CopyFreeObject Command.

Complex Number Tool

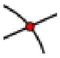
Click in the the Graphics View in order to create a new complex number. The value of the complex number point is fixed when the mouse button is released.

Point on Object Tool

To create a point, which is fixed to an object, click on the tool button first and then on the object. This new point can be moved via Move Tool, but only within the object.

Note: To put a point in the interior of a Circle or Ellipse you will need to increase the Opacity from 0 first. If you click on the perimeter of an object (eg Circle, Ellipse, Polygon), then the point will be fixed to the perimeter rather than the interior.

Intersect Two Objects Tool



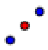
Intersection points of two objects can be created in two ways (see also Intersect command).

- Selecting two objects creates all intersection points (if possible).
- Directly clicking on an intersection of the two objects creates only this *single intersection point*.

Note: For segments, rays, or arcs you may specify whether you want to *Allow outlying intersections* on tab *Basic* of the Properties Dialog. This can be used to get intersection points that lie on the extension of an object. For example, the extension of a segment or a ray is a straight line.

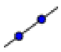
Note: Sometimes it's useful to display only the portions of the intersecating objects near the intersection point. To do so, right click on the intersection point, and check the option *Show trimmed intersection lines* in the *Basic* tab of the *Properties* dialog of the object, then hide the intersecting objects.

Midpoint or Center Tool



You may click on either two points or one segment to get its midpoint. You can also click on a conic section (circle or ellipse) in order to create its center point (see also Center and Midpoint commands).

Line Tools

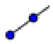
Line tools are by default grouped under  icon (the third from left) in the toolbar. Currently there are six line tools:

- Line through Two Points
- Segment between Two Points
- Segment with Given Length from Point
- Ray through Two Points
- Vector between Two Points
- Vector from Point

Vector from Point Tool

Select a point *A* and a vector *v* to create the new point *B* = *A* + *v* as well as the vector from *A* to *B* (see also Vector command).

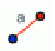
Ray through Two Points Tool



Selecting two points *A* and *B* creates a ray starting at *A* through *B* (see also Ray command).

Note: In the Algebra View the equation of the corresponding line is displayed.

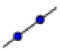
Segment with Given Length from Point Tool



Click on a point *A* that should be the starting point of the segment. Specify the desired length *a* of the segment in the appearing window (see also Segment command).

Note: This tool creates a segment with length *a* and endpoint *B* which may be rotated around the starting point *A* by using tool Move.

Line through Two Points Tool



Selecting two points *A* and *B* creates a straight line through *A* and *B* (see also Line command).

Note: The line's direction vector is *(B - A)*.

Segment between Two Points Tool

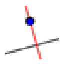
Select two points *A* and *B* in order to create a segment between *A* and *B* (see also Segment command).

Note: In the Algebra View, the segment's length is displayed.

Vector between Two Points Tool

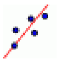
Select the starting point and then the end point of the vector (see also Vector command).

Special Line Tools

Special line tools are by default grouped under  icon (the fourth from the left) in the toolbar. Currently there are eight line tools:

- Perpendicular Line
- Parallel Line
- Perpendicular Bisector
- Angle Bisector
- Tangents
- Polar or Diameter Line
- Best Fit Line
- Locus

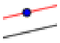
Best Fit Line Tool



Creates the best fit line for a set of points, chosen as follows (see also FitLine command):

- Creating a selection rectangle that contains all points.
- Selecting a list of points .

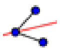
Parallel Line Tool



Selecting a line *g* and a point *A* defines a straight line through *A* parallel to *g* (see also Line command).

Note: The line's direction is the direction of line *g*.

Angle Bisector Tool

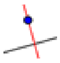


Angle bisectors can be defined in two ways (also see command AngleBisector):

- Selecting three points *A*, *B*, and *C* produces the angle bisector of the enclosed angle, where point *B* is the apex.
- Selecting two lines produces their two angle bisectors.

Note: The direction vectors of all angle bisectors have length 1.

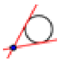
Perpendicular Line Tool



Selecting a line (or a segment) *g* and a point *A* creates a straight line through *A* perpendicular to line (or segment) *g* (see also PerpendicularLine command).

Note: The line's direction is equivalent to the perpendicular vector of *g* (see also PerpendicularVector command).

Tangents Tool

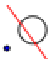


Tangents to a conic section can be produced in several ways (see also Tangent command):

- Selecting a point *A* and a conic *c* produces all tangents through *A* to *c*.
- Selecting a line *g* and a conic *c* produces all tangents to *c* that are parallel to line *g*.
- Selecting a point *A* and a function *f* produces the tangent line to *f* in *x = x(A)*.
- Selecting two circles *c* and *d* produces the common tangents to the two circles (up to 4).

Note: *x(A)* represents the *x*-coordinate of point *A*. If point *A* lies on the function graph, the tangent runs through point *A*.

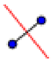
Polar or Diameter Line Tool



This tool creates the polar or diameter line of a conic section (see also Polar command).

- Select a point and a conic section to get the polar line.
- Select a line or a vector and a conic section to get the diameter line.

Perpendicular Bisector Tool



Click on either a segment (or interval) *s* or two points *A* and *B* in order to create a perpendicular bisector (also see command PerpendicularBisector).

Note: The bisector's direction is equivalent to the perpendicular vector of segment (or interval) *s* or *AB* (see also PerpendicularVector command).

Locus Tool

Select a point *B* that depends on another point *A* and whose locus should be drawn. Then, click on point *A* to create the locus of point *B* (see also Locus command).

Note: Point *A* has to be a point on an object (e. g. line, segment/interval, circle).

Example:

Type f(x) = x^2 – 2 x – 1 into the Input Bar and press the Enter-key. Place a new point A on the x-axis (see New Point ToolNew Point tool; see Point CommandPoint command). Create point B = (x(A), f'(x(A))) that depends on point A. Select tool and successively click on point B and point A.Drag point A along the x-axis to see point B moving along its locus line.

Warning: Locus is undefined, if the dependent point depends on Point Command with two parameters or PathParameter Command.

Polygon Tools

Polygon tools are by default grouped under  icon (the fifth from the left) in the toolbar. Currently there are five polygon tools:

- Polygon
- Regular Polygon
- Rigid Polygon
- Vector Polygon
- Polyline

Rigid Polygon Tool

Successively select at least three free points which will be the vertices of the polygon. Then, click the first point again in order to close the polygon (see also Rigid Polygon command). The resulting polygon will keep the shape: you can move it and rotate it by moving two vertices.

Holding down the key when drawing a rigid polygon allows to get angles that are a multiple of 15°.

Note: In the Algebra View, the polygon's area is displayed.

PolyLine Tool

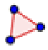
Successively select at least three points which will be the vertices of the polyline. Then, click the first point again in order to finish the construction of the polyline (see also PolyLine command).

Note: The length of the line is displayed in the Algebra View.

Regular Polygon Tool

Select two points *A* and *B* and specify the number *n* of vertices in the text field of the appearing dialog window. This gives you a regular polygon with *n* vertices including points *A* and *B* (see also Polygon command).

Polygon Tool

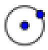


Successively select at least three points which will be the vertices of the polygon. Then, click the first point again in order to close the polygon (see also Polygon command).

Holding down the key when drawing a Polygon allows to get angles that are a multiple of 15°.

Note: The polygon area is displayed in the Algebra View.

Circle & Arc Tools

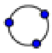
Circle and arc tools are by default grouped under  icon (the sixth from the left) in the toolbar. Currently there are nine circle and arc tools:

- Circle with Centre through Point
- Circle with Centre and Radius
- Compasses
- Circle through Three Points
- Semicircle through Two Points
- Circular Arc with Centre between Two Points
- Circumcircular Arc through Three Points
- Circular Sector with Centre between Two Points
- Circumcircular Sector through Three Points

Circle with Center and Radius Tool

Select the center point *M* and enter the radius in the text field of the appearing dialog window (see also Circle command).

Circle through Three Points Tool



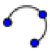
Selecting three points *A*, *B*, and *C* defines a circle through these points (see also Circle command).

Note: If the three points lie on the same line, the circle degenerates to this line.

Circle with Center through Point Tool

Selecting a point *M* and a point *P* defines a circle with center *M* through *P*.

Circumcircular Arc through Three Points Tool

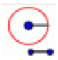


Selecting three points *A*, *B*, and *C* creates a circular arc through these points. Thereby, point *A* is the starting point of the arc, point *B* lies on the arc, and point *C* is the endpoint of the arc (see also CircumcircularArc command).

Circumcircular Sector through Three Points Tool

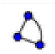
Selecting three points *A*, *B*, and *C* creates a circular sector through these points. Thereby, point *A* is the starting point of the sector's arc, point *B* lies on the arc, and point *C* is the endpoint of the sector's arc (see also CircumcircularSector command).

Compass Tool



Select a segment or two points to specify the radius. Then, click on a point that should be the center of the new circle.

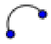
Circular Sector with Center between Two Points Tool



First, select the center point *M* of the circular sector. Then, select the starting point *A* of the sector's arc, before you select a point *B* that specifies the length of the sector's arc (see also CircularSector command).

Note: While point *A* always lies on the sector's arc, point *B* does not have to lie on it.

Semicircle through Two Points Tool



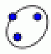
Select two points *A* and *B* to create a semicircle above the segment (or interval) *AB* (see also Semicircle command).

Circular Arc with Center between Two Points Tool

First, select the center point *M* of the circular arc. Then, select the starting point *A* of the arc, before you select a point *B* that specifies the length of the arc (see also CircularArc command).

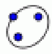
Note: While point *A* always lies on the circular arc, point *B* does not have to lie on it.

Conic Section Tools

Conic section tools are by default grouped under  icon (the sixth from the right) in the toolbar. Currently there are four conic section tools:

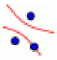
- Ellipse
- Hyperbola
- Parabola
- Conic through Five Points

Ellipse Tool



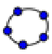
Select the two foci of the ellipse. Then, specify a third point that lies on the ellipse (see also Ellipse command).

Hyperbola Tool



Select the two foci of the hyperbola. Then, specify a third point that lies on the hyperbola (see also Hyperbola command).

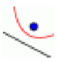
Conic through Five Points Tool



Selecting five points produces a conic section through these points (see also Conic command).

Note: If four of these five points lie on a line, the conic section is not defined.

Parabola Tool



Select a point (focus) and the directrix of the parabola (see also Parabola command).

Measurement Tools

Measurement tools are by default grouped under  icon (the fifth from the right) in the toolbar. Currently there are five measurement tools:

- Angle
- Angle with Given Size
- Distance or Length
- Area
- Slope
- Create List

Distance or Length Tool

This tool gives you the distance between two points, two lines, or a point and a line as a number and shows a dynamic text in the Graphics View. It can also give you the length of a segment (or interval), the circumference of a circle, or the perimeter of a polygon (see also Distance and Length commands).

Angle Tool

With this tool you can create angles in different ways (also see command Angle):

- Click on three points to create an angle between these points. The second point selected is the vertex of the angle.
- Click on two segments to create the angle between them.
- Click on two lines to create the angle between them.
- Click on two vectors to create the angle between them.
- Click on a polygon to create all angles of this polygon.

Note: If the polygon was created by selecting its vertices in counter clockwise orientation, the *Angle* tool gives you the interior angles of the polygon.

Note: Angles are created in *counter clockwise* orientation. Therefore, the order of selecting these objects is relevant for the *Angle* tool. If you want to limit the maximum size of an angle to 180°, un-check *Allow Reflex Angle* on tab *Basic* of the Properties Dialog.

Slope Tool

This tool gives you the slope of a line and shows a slope triangle in the Graphics View (see also Slope command).

Area Tool

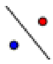
This tool gives you the area of a polygon, circle, or ellipse as a number and shows a dynamic text in the Graphics View (see also Area command).

Angle with Given Size Tool

Select two points *A* and *B* and type the angle's size into the text field of the appearing window (also see command Angle).

Note: This tool creates a point *C* and an angle α, where α is the angle *ABC*.

Transformation Tools

Transformation tools are by default grouped under  icon (the fourth from the right) in the toolbar. Currently there are six transformation tools:

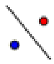
- Reflect Object in Line
- Reflect Object in Point
- Reflect Point in Circle
- Rotate Object around Point by Angle
- Translate Object by Vector
- Enlarge Object from Point by Factor

Translate Object by Vector Tool

Select the object you want to translate. Then, click on the translation vector or click twice to make a vector (see also Translate command).

From version 4.0.15.0 you can also now just drag to clone an object with this tool.

Reflect Object about Line Tool

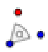


Select the object you want to reflect. Then, click on a line to specify the mirror/line of reflection (see also Reflect command).

Reflect Object about Point Tool

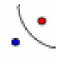
Select the object you want to reflect. Then, click on a point to specify the mirror/point of reflection (see also Reflect command).

Rotate Object around Point by Angle Tool



Select the object you want to rotate. Then, click on a point to specify the center of rotation and enter the rotation angle into the text field of the appearing dialog window (see also Rotate command).

Reflect Object about Circle Tool



This tool allows you to invert a geometric object about a circle. Select the object you want to invert. Then, click on a circle to specify the mirror/circle of inversion (see also Reflect command).

Dilate Object from Point by Factor Tool

Select the object to be dilated. Then, click on a point to specify the dilation center and enter the dilation factor into the text field of the appearing dialog window (see also Dilate command).

Special Object Tools

Special object tools are by default grouped under ABC icon (the third from the right) in the toolbar. Currently there are six special object tools:

- Insert Text
- Insert Image
- Pen Tool
- Relation between Two Objects
- Probability Calculator
- Function Inspector

Insert Image Tool

This tool allows you to insert an image into the Graphics View.
First, specify the location of the image in one of the following two ways:

- Click in the Graphics View to specify the position of the image's lower left corner.
- Click on a point to specify this point as the lower left corner of the image.

Then, a file-open dialog appears that allows you to select the image file from the files saved on your computer.

Note: After selecting the tool Insert Image, you can use the keyboard shortcut *Alt-click* in order to paste an image directly from your computer's clipboard into the Graphics View.

Properties of Images

The position of an image may be absolute on screen or relative to the coordinate system. You can specify this on tab Basic of the Properties Dialog of the image.

You may specify up to three corner points of the image on tab Position of the Properties Dialog. This gives you the flexibility to scale, rotate, and even distort images (also see command Corner Command).

- Corner 1: position of the lower left corner of the image
- Corner 2: position of the lower right corner of the image

Note: This corner may only be set if Corner 1 was set before. It controls the width of the image.

- Corner 4: position of the upper left corner of the image

Note: This corner may only be set if Corner 1 was set before. It controls the height of the image.

Example: Create three points A, B, and C to explore the effects of the corner points.

- Set point A as the first and point B as the second corner of your image. By dragging points A and B in Move mode you can explore their influence.
- Now, remove point B as the second corner of the image. Set point A as the first and point C as the fourth corner and explore how dragging the points now influences the image.
- Finally, you may set all three corner points and see how dragging the points distorts your image.

Example: You already saw how to influence the position and size of your image. If you want to attach your image to a point A and set its width to 3 and its height to 4 units, you could do the following:

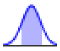
Set Corner 1 to A Set Corner 2 to A + (3, 0) Set Corner 4 to A + (0, 4)

Note: If you now drag point A in Move mode, the size of your image does not change.
You may specify an image as a Background Image on tab Basic of the Properties Dialog. A background image lies behind the coordinate axes and cannot be selected with the mouse any more.

Note: In order to change the background setting of an image, you may open the Properties Dialog by selecting Properties… from the Edit Menu.

The Transparency of an image can be changed in order to see objects or axes that lie behind the image. You can set the transparency of an image by specifying a Filling value between 0 % and 100 % on tab Style of the Properties Dialog.

Probability Calculator Tool

 Clicking on this tool opens a dialog for calculating and graphing probability distributions.

- To change a distribution click on the distribution drop down menu and choose a one from the list. Then adjust the parameters for the distribution in the adjacent text boxes.
- To calculate a probability click on the probability drop down menu and select the interval type. Then adjust the interval in the adjacent text boxes. You can also find probabilities by moving the points on the x-axis in the graph.

Options

The probability calculator has a stylebar with options to change settings and export the graph. To open the style bar, click on the open button in the title bar or double click the title bar. From the style bar you can change the decimal rounding format, use cumulative probability and export the graphics.

Drag and Drop

When the mouse is positioned at the top of the graphics screen the cursor will change to a hand cursor. From here you can drag the plot into GraphicsView 1 or 2 to create a new plot or you can drag an image of the plot into another application that will accept images.

Pen Tool

The Pen Tool allows the user to add freehand notes and drawings to the Graphics View. This makes the Pen Tool particularly useful when using GeoGebra for presentations or with multimedia interactive whiteboards. To add a freehand note onto a selected region of the Graphics View, draw a rectangle with the Move Tool first, or select an existing image first to draw into it. To finish, select another tool.

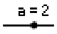
GeoGebra stores the notes you have traced in the Graphic View as an image, so you can do any image operations with it (move, rotate, delete, etc.).

The default color of the pen is black, but you can change the pen properties (color, style, and thickness) using the Styling Bar, selecting the first little icon displayed on the Graphics View bar.

Erasing

To erase a portion of your notes created in the Graphic View with the Pen Tool, press and hold the right mouse button while moving it on the notes you want to delete. Erasing is completed when you release the mouse button.

Slider Tool



Click on any free place in the Graphics View to create a slider for a number or an angle. The appearing dialog window allows you to specify the *Name*, *Interval* [*min*, *max*], and *Increment* of the number or angle, as well as the *Alignment* and *Width* of the slider (in pixels), and its *Speed* and *Animation* modality.
Note: In the Slider dialog window you can enter a degree symbol ° or *pi* (π) for the interval and increment by using the following keyboard shortcuts:

Alt-O (Mac OS: Ctrl-O) for the degree symbol ° Alt-P (Mac OS: Ctrl-P) for the pi symbol π

The position of a slider may be absolute in the Graphics View (this means that the slider is not affected by zooming, but always remains in the visible part of the Graphics View) or relative to the coordinate system (see Properties Dialog of the corresponding number or angle).

Note:

In GeoGebra, a slider is the graphical representation of a Numbers and Angles#Free Numbers and Anglesfree number or free angle. You can easily create a slider for any existing Numbers and Angles#Free Numbers and Anglesfree number or angle by showing this object in the Graphics View (see Context Menu; see tool Show/Hide Object ToolShow/Hide Object). Sliders made with the Slider Tool are fixed by default (from GeoGebra 4.0). To translate a fixed slider when Move Tool is selected, you can drag it with your right mouse button. When Slider Tool is selected, you can use either left or right button.

Relation between Two Objects Tool

Select two objects to get information about their relation in a pop-up window (see also Relation command) .

Function Inspector Tool

 Enter the function you want to analyze. Then choose the tool.

- In the tab *Interval* you can specify the interval, where the tool will find minimum, maximum, root, etc. of the function.
- In the tab *Points* several points of the function are given (step can be changed). Slope etc. can be found at these points.

Insert Text Tool

ABC

With this tool you can create static and dynamic text or LaTeX formulas in the Graphics View.

At first, you need to specify the location of the text in one of the following ways:

- Click in the Graphics View to create a new text at this location.
- Click on a point to create a new text that is attached to this point.

Note: You may specify the position of a text as absolute on screen or relative to the coordinate system on tab Basic of the Properties Dialog.

Then, a dialog appears where you may enter your text, which can be static, dynamic, or mixed.

The text you type directly in the *Edit* field is considered as static, i.e. it's not affected by the objects modifications. If you need to create a dynamic text, which displays the changing values of an object, select the related object from the *Objects* drop-down list. The corresponding name is shown, enclosed in a grey box, in the *Edit* field, and its value is displayed in the *Preview* box. Right-clicking on the grey box allows you to select "Definition" or "Value" for each dynamic object.

It is also possible to perform algebraic operations or apply specific commands to these objects, just clicking in the grey box and typing the algebraic operation or GeoGebra text command desired. The results of these operations will be dynamically shown in the resulting text, in the Graphics View.

Best visual results are obtained when using LaTex formatting for the formulas. Its use is simple and intuitive: just check the *LaTeX Formula* box, and select the desired formula template from the drop-down list. You can also select a variety of mathematical symbols and operators from the *Symbols* drop-down list.

Action Object Tools

These tools allow you to create Action Objects. They are by default grouped under icon (the second from the right) in the toolbar. Currently there are four action object tools:

- Slider
- Check Box to Show / Hide Objects
- Insert Button
- Insert Textfield

Check Box to Show / Hide Objects Tool

Clicking in the Graphics View creates a check box (see section Boolean values) that allows you to show and hide one or more objects. In the appearing dialog window you can specify which objects should be affected by the check box.

Note: You may select these objects from the list provided in the dialog window or select them with the mouse in any view.

Insert Input Box Tool

Click in the Graphics View to insert a textfield. In the appearing dialog you may set its caption and Linked Object.

Insert Button Tool

Click in the Graphics View to insert a button. In the appearing dialog you may set its caption and OnClick script.

General Tools

General tools are by default grouped under icon (the first from the right) in the toolbar. Currently there are seven general tools:

- Move Graphics View
- Zoom In
- Zoom Out
- Show / Hide Object
- Show / Hide Label
- Copy Visual Style
- Delete Object

Custom Tools

GeoGebra allows you to create your own construction tools based on an existing construction. Once created, your custom tool can be used both with the mouse and as a command in the Input Bar. All tools are automatically saved in your GeoGebra file.

Note: Outputs of the tool are not moveable, even if they are defined as `Point[<Path>]`. In case you need moveable output, you can define a list of commands and use it with Execute Command.

Creating custom tools

To create a custom tool, use the option Create new tool from Tools Menu.

Saving custom tools

When you save the construction as GGB file, all custom tools are stored in it. To save the tools in separate file(s) use the Tool Manager Dialog (option Manage Tools from Tools Menu).

Accessing custom tools

If you open a new GeoGebra interface using item New from the File menu, after you created a custom tool, it will still be part of the GeoGebra Toolbar. However, if you open a new GeoGebra window (item New Window from the File Menu), or open GeoGebra on another day, your custom tools won't be part of the Toolbar any more.

There are different ways of making sure that your user defined tools are displayed in the Toolbar of a new GeoGebra window:

After creating a new user defined tool you can save your settings using item Save Settings from the Options Menu. From now on, your customized tool will be part of the GeoGebra Toolbar.

Note: You can remove the custom tool from the Toolbar after opening item Customize Toolbar… from the Tools Menu. Then, select your custom tool from the list of tools on the left hand side of the appearing dialog window and click button Remove. Don’t forget to save your settings after removing the custom tool.

Importing custom tools

After saving your custom tool on your computer (as a GGT file), you can import it into a new GeoGebra window at any time. Just select item Open from the File Menu and open the file of your custom tool.

Note:

Opening a GeoGebra tool file (GGT) in GeoGebra doesn’t affect your current construction. It only makes this tool part of the current GeoGebra Toolbar. You can also load GGT file by dragging it from file manager and droping into GeoGebra window.

Show / Hide Label Tool

Click on an object to show or hide its label.

Zoom Out Tool

Click on any place on the drawing pad to zoom out (see also Customizing the Graphics View section).

Note: The position of your click determines the center of zoom.

Zoom In Tool

Click on any place on the drawing pad to zoom in (also see section Customizing the Graphics View).

Note: The position of your click determines the center of zoom.

Delete Object Tool

Click on any object you want to delete (see also Delete command).
Note: You can use the Undo button if you accidentally delete the wrong object.

Move Graphics View Tool

Drag and drop the drawing pad in the Graphics View to change its visible area.

Note:

You can also move the drawing pad by pressing the Shift-key (MS Windows: also Ctrl-key) and dragging it with the mouse in any mode. In this mode you can also scale each of the axes by dragging it with the mouse.

Show / Hide Object Tool

Select the object you want to show or hide after activating this tool. Then, switch to another tool in order to apply the visibility changes to this object.

Note: When you activate this tool, all objects that should be hidden are displayed in the Graphics View highlighted. In this way, you can easily show hidden objects again by deselecting them before switching to another tool.

Copy Visual Style Tool

This tool allows you to copy visual properties (e. g., color, size, line style) from one object to one or more other objects. To do so, first select the object whose properties you want to copy. Then, click on all other objects that should adopt these properties.

Commands

Commands

Using commands you can produce new and modify existing objects.

Note: A command's result may be named by entering a label followed by an equal sign (=). In the example below, the new point is named S.

Example: To get the intersection point of two lines g and h you can enter `S = Intersect[g, h]` (see Intersect Command).

Note: You can also use indices within the names of objects: A_1 is entered as `A_1` while S_{AB} is created using `S_{AB}`. This is part of LaTeX syntax.

Geometry Commands

- AffineRatio
- Angle
- AngleBisector
- Arc
- Area
- Centroid
- CircularArc
- CircularSector
- CircumcircularArc
- CircumcircularSector
- Circumference
- ClosestPoint
- CrossRatio
- Direction
- Distance
- Incircle
- Intersect
- IntersectRegion
- Length
- Line
- Locus
- Midpoint
- Perimeter
- PerpendicularBisector
- PerpendicularLine
- Point
- PointIn
- Polygon
- PolyLine

- Radius
- Ray
- RigidPolygon
- Sector
- Segment
- Slope
- Tangent
- Vertex

AffineRatio Command

AffineRatio[Point A, Point B, Point C]

Returns the affine ratio λ of three collinear points *A*, *B*, and *C*, where $C = A + \lambda * AB$.

Angle Command

Angle[Vector v1, Vector v2]

Returns the angle between two vectors *v1* and *v2* (between 0 and 360°) .

Angle[Line g, Line h]

Returns the angle between the direction vectors of two lines *g* and *h* (between 0 and 360°) .

Angle[Point A, Point B, Point C]

Returns the angle enclosed by *BA* and *BC* (between 0 and 360°), where point *B* is the apex.

Angle[Point A, Point B, Angle α]

Returns the angle of size *α* drawn from point *A* with apex *B*.

Note: The point *Rotate[A, α, B]* is created as well.

Angle[Conic]

Returns the angle of twist of a conic section’s major axis (see command Axes) .

Angle[Vector]

Returns the angle between the *x*-axis and given vector.

Angle[Point]

Returns the angle between the *x*-axis and the position vector of the given point.

Angle[Number]

Converts the number into an angle (result between 0 and 2pi).

Angle[Polygon]

Creates all angles of a polygon in mathematically positive orientation (i.e., counter clockwise).

Note: If the polygon was created in counter clockwise orientation, you get the interior angles. If the polygon was created in clockwise orientation, you get the exterior angles.

Note: See also Angle and Angle with Given Size tools .

AngleBisector Command

AngleBisector[Point A, Point B, Point C]

Returns the angle bisector of the angle defined by points *A*, *B*, and *C*.

Note: Point *B* is apex of this angle.

AngleBisector[Line g, Line h]

Returns both angle bisectors of the lines.

Note: See also Angle Bisector tool .

Arc Command

Arc[Conic, Point A, Point B]

Returns a conic section arc between two points *A* and *B* on the circle or ellipse *c*. For other conics is the result undefined.

Arc[Conic, Number t1, Number t2]

Returns a conic section arc between two parameter values *t1* and *t2* on the circle or ellipse. For other conics is the result undefined.

Note: Internally the following parametric forms are used:

Circle: (r cos(t), r sin(t)) where r is the circle's radius. Ellipse: (a cos(t), b sin(t)) where a and b are the lengths of the semimajor and semiminor axes.

Area Command

Area[Point A, Point B, Point C, ...]

Calculates the area of the polygon defined by the given points *A*, *B*, *C*,…

Area[Conic c]

Calculates the area of a conic section *c* (circle or ellipse).

Area[Arc or Sector c]

Calculates area of the arc or sector.

Area[Polygon]

Yields the area of the polygon.

Note:

In order to calculate the area between two function graphs, you need to use the command Integral CommandIntegral. Also see tool Area ToolArea.

Centroid Command

Centroid[Polygon]

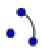
Returns the centroid of the polygon.

CircularArc Command

CircularArc[Point M, Point A, Point B]

Creates a circular arc with midpoint *M* between points *A* and *B*.

Note: Point *B* does not have to lie on the arc.

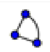
Note: See also  Circular Arc with Center between Two Points tool.

CircularSector Command

CircularSector[Point M, Point A, Point B]

Creates a circular sector with midpoint *M* between two points *A* and *B*.

Note: Point *B* does not have to lie on the arc of the sector.

Note: See also  Circular Sector with Center between Two Points tool.

CircumcircularArc Command

CircumcircularArc[Point A, Point B, Point C]

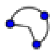
Creates a circular arc through three points *A*, *B*, and *C*, where *A* is the starting point and *C* is the endpoint of the circumcircular arc.

Note: See also  Circumcircular Arc through Three Points tool.

CircumcircularSector Command

CircumcircularSector[Point A, Point B, Point C]

Creates a circular sector whose arc runs through the three points *A*, *B*, and *C*. Point *A* is the starting point and point *C* is the endpoint of the arc.

Note: See also  Circumcircular Sector through Three Points tool.

Circumference Command

Circumference[Polygon]

Returns the circumference of a Polygon.

Circumference[Conic]

Returns the circumference of a circle or ellipse.

ClosestPoint Command

ClosestPoint[Path P, Point A]

Returns a point on path *P* which is the closest to point *A*.

Note: For Functions, this command will return the point vertically in line rather than the nearest point

CrossRatio Command

CrossRatio[Point A, Point B, Point C, Point D]

Calculates the cross ratio λ of four collinear points *A*, *B*, *C*, and *D*, where *λ = AffineRatio[B, C, D] / AffineRatio[A, C, D]*.

Direction Command

Direction[Line]

Yields the direction vector of the line.

Note: A line with equation $ax + by = c$ has the direction vector $(b, - a)$.

Distance Command

Distance[Point A, Object O]

Yields the (shortest) distance between points *A* and the Object. Works for Points, Segments, Lines, Conics and Implicit Curves. When *f* is a function, `Distance[A,f]` returns distance between *A* and *(x(A),f(x(A)))*.

Note: In GeoGebra 3.2 **Distance[Point A, Segment]** gave the distance to the segment extended to an infinite line. From GeoGebra 4.0 it gives the distance to the Segment itself.

Distance[Line g, Line h]

Yields the distance between the parallel lines *g* and *h*.

Note: The distance between intersecting lines is 0. Thus, this command is only interesting for parallel lines.

Note: See also Distance or Length tool .

Intersect Command

Intersect[Line g, Line h]

Yields the intersection point of lines *g* and *h*.

Intersect[Line, Conic]

Yields all intersection points of the line and conic section (max. 2).

Intersect[Line, Conic, Number n]

Yields the n^{th} intersection point of the line and the conic section.

Intersect[Conic c1, Conic c2]

Yields all intersection points of conic sections *c1* and *c2* (max. 4).

Intersect[Conic c1, Conic c2, Number n]

Yields the n^{th} intersection point of conic sections *c1* and *c2*.

Intersect[Polynomial f1, Polynomial f2]

Yields all intersection points of polynomials *f1* and *f2*.

Intersect[Polynomial f1, Polynomial f2, Number n]

Yields the n^{th} intersection point of polynomials *f1* and *f2*.

Intersect[Polynomial, Line]

Yields all intersection points of the polynomial and the line.

Intersect[Polynomial, Line, Number n]

Yields the n^{th} intersection point of the polynomial and the line.

Intersect[Function f, Function g, Point A]

Calculates the intersection point of functions *f* and *g* by using Newton's method with initial point *A*.

Intersect[Function, Line, Point A]

Calculates the intersection point of the function and the line by using Newton's method with initial point *A*.

Intersect[Function f, Function g, left-x, right-x]

Calculates the intersection points for the two functions in the given interval.

Note: Also see tool Intersect Two Objects.

IntersectRegion Command

IntersectRegion[Polygon poly1, Polygon poly2]

Finds the intersection (overlap) of the two polygons. Works only for where the polygons are not self-intersecting, and where the union is a single polygon.

Length Command

Length[Vector]

Yields the length of the vector.

Length[Point A]

Yields the length of the position vector of the given point .

Length[Function, Number x1, Number x2]

Yields the length of the function graph in the interval [*x1, x2*].

Example:

`Length[2x, 0, 1]` yields \(\sqrt{5}\).

Length[Function, Point A, Point B]

Yields the length of the function graph between the two points *A* and *B*.

Note: If the given points do not lie on the function graph, their x-coordinates are used to determine the interval.

Length[Curve, Number t1, Number t2]

Yields the length of the curve between the parameter values *t1* and *t2*.

Length[Curve c, Point A, Point B]

Yields the length of curve *c* between two points *A* and *B* that lie on the curve.

Length[List]

Yields the length of the list, which is the number of elements in the list.

Length[Text]

Yields the number of characters in the text.

Length[Locus]

Returns the number of points that the given locus is made up of. Use Perimeter[Locus] to get the length of the locus itself. For details see the article about First Command.

Note:

See also 

Distance or Length tool.

CAS Syntax

Length[Function, Number t1, Number t2]

Calculates the length of a function graph from point x=*t1* to point x=*t2*.

Example:

`Length[2x, 0, 1]` yields $\sqrt{5}$.

Length[Function, Variable a, Number t1, Number t2]

Calculates the length of a function graph from point a=*t1* to point a=*t2*.

Example:

`Length[2a, a, 0, 1]` yields $\sqrt{5}$.

Length[Segment]

Yields the length of the segment.

Line Command

Line[Point A, Point B]

Creates a line through two points *A* and *B*.

Line[Point, Parallel Line]

Creates a line through the given point parallel to the given line.

Line[Point, Direction Vector v]

Creates a line through the given point with direction vector *v*.

Note: See also Line through Two Points and Parallel Line tools.

PerpendicularBisector Command

PerpendicularBisector[Point A, Point B]

Yields the perpendicular bisector of the line segment *AB*.

PerpendicularBisector[Segment]

Yields the perpendicular bisector of the segment.

Note: See also Perpendicular Bisector tool.

Locus Command

Locus[Point Q, Point P]

Returns the locus curve of the point *Q*, which depends on the point *P*.

Note: Point *P* needs to be a point on an object (e. g. line, segment, circle).

Locus[Point Q, slider t]

Returns the locus curve of the point *Q*, which depends on the values assumed by the slider *t*.

Loci are specific object types, and appear as auxiliary objects. Besides Locus command, they are the result of some Discrete Math Commands and SolveODE Command. Loci are paths and can be used within path-related commands such as Point. Their properties depend on how they were obtained, see e.g. Perimeter Command and First Command.

Note: See also Locus tool.

Warning: A locus is undefined when the dependent point is the result of a Point Command with two parameters, or a PathParameter Command.

Midpoint Command

Midpoint[Point A, Point B]

Returns the midpoint of points *A* and *B*.

Midpoint[Segment]

Returns the midpoint of the segment.

Midpoint[Interval]

Returns the midpoint of the interval (as number).

Midpoint[Conic]

Returns the center of the conic.

Note: See also  Midpoint or Center tool.

PerpendicularLine Command

PerpendicularLine[Point, Line]

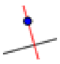
Creates a line through the point perpendicular to the given line.

PerpendicularLine[Point, Segment]

Creates a line through the point perpendicular to the given segment.

PerpendicularLine[Point, Vector]

Creates a line through the point perpendicular to the given vector.

Note: See also  Perpendicular Line tool.

Perimeter Command

Perimeter[Polygon]

Returns the perimeter of the polygon.

Perimeter[Conic]

If the given conic is ellipse, this command returns its perimeter. Otherwise the result is undefined.

Perimeter[Locus]

If the given locus is finite, this command returns its perimeter. Otherwise the result is undefined.

Point Command

Point[Path]

Returns a point on the geometric object. The resulting point can be moved along the path.

Point[Path, Parameter]

Returns a point on the geometric object with given path parameter.

Point[Point, Vector]

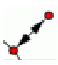
Creates a new point by adding the vector to the given point.

Note: See also New Point tool.

PointIn Command

PointIn[<Region>]

Returns a point restricted to given region.

Note: See also  Attach / Detach Point Tool.

PolyLine Command

PolyLine[<Point>, ..., <Point>]

Creates an open polygonal chain (i.e. a connected series of segments) having the initial vertex in the first entered point, and the final vertex in the last entered point.

Note: The polygonal chain length is displayed in the Algebra View.

PolyLine[<List of Points>]

Creates an open polygonal chain (i.e. a connected series of segments) having the initial vertex in the first point of the list, and the final vertex in the last point of the list.

Note: The polygonal chain length is displayed in the Algebra View.

See also Polygon command.

Polygon Command

Polygon[Point A, Point B, Point C,...]

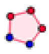
Returns a polygon defined by the given points A, B, C,…

Polygon[Point A, Point B, Number n]

Creates a regular polygon with n vertices (and one side between points A and B).

Polygon[List of points]

Returns a polygon defined by the points in the list.

Note: See also  Polygon and  Regular Polygon tools.

Radius Command

Radius[Circle]

Returns the radius of the circle.

Ray Command

Ray[Point A, Point B]

Creates a ray starting at point *A* through point *B*.

Ray[Point, Vector v]

Creates a ray starting at the given point which has the direction vector *v*.

Note: See also Ray through Two Points tool.

RigidPolygon Command

RigidPolygon[<Free Point>, ..., <Free Point>]

Creates polygon whose shape cannot be changed. This polygon can be translated by dragging its first vertex and rotated by dragging its second vertex.

Sector Command

Sector[Conic, Point A, Point B]

Yields a conic sector between two points *A* and *B* on the conic section.

Note: This works only for a circle or ellipse.

Sector[Conic, Number t1, Number t2]

Yields a conic sector between two parameter values *t1* and *t2* on the conic section.

Note: Internally the following parametric forms are used:

Circle: (r cos(t), r sin(t)) where r is the circle's radius. Ellipse: (a cos(t), b sin(t)) where a and b are the lengths of the semimajor and semiminor axes.

Segment Command

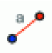
Segment[Point A, Point B]

Creates a segment between two points *A* and *B*.

Segment[Point A, Number a]

Creates a segment with length *a* and starting point *A*.

Note: The endpoint of the segment is created as well.

Note: See also  Segment between Two Points and  Segment with Given Length from Point tools.

Slope Command

Slope[Line]

Returns the slope of the given line.

Note: This command also draws the slope triangle whose size may be changed on tab Style of the Properties Dialog.

Note: See also  Slope tool.

Tangent Command

Tangent[Point, Conic]

Creates (all) tangents through the point to the conic section.

Tangent[Line, Conic]

Creates (all) tangents to the conic section that are parallel to the given line.

Tangent[Segment, Conic]

Creates (all) tangents to the conic section that are parallel to the given segment.

Tangent[Number a, Function]

Creates the tangent to the function at $x = a$.

Tangent[Point A, Function]

Creates the tangent to the function at $x = x(A)$.

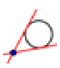
Note: $x(A)$ is the x-coordinate of point *A*.

Tangent[Point, Curve]

Creates the tangent to the curve in the given point.

Tangent[<Circle>, <Circle>]

Creates the common tangents to the two Circles (up to 4).

Note: See also  Tangents tool.

Vertex Command

Vertex[Conic]

Returns (all) vertices of the conic section.

Vertex[Polygon]

Returns (all) vertices of the polygon.

Vertex[Polygon, Number n]

Returns *n*-th vertex of the polygon.

Note: To get vertices of the polygon / conic as list, use `{Vertex[t]}`.

Algebra Commands

- Div
- Expand
- Factor
- GCD
- LCM
- Max
- Min
- Mod
- PrimeFactors
- Product
- Simplify

Div Command

Div[<Dividend Number>, <Divisor Number>]

Returns the quotient (integer part of the result) of the two numbers.

Example:

`Div[16, 3]` yields *5*.

Div[<Dividend Polynomial>, <Divisor Polynomial>]

Returns the quotient of the two polynomials.

Example:

`Div[x^2 + 3 x + 1, x - 1]` yields $f(x) = x + 4$.

CAS Syntax

Div[<Dividend Number>, <Divisor Number>]

Returns the quotient (integer part of the result) of the two numbers.

Example:

`Div[16, 3]` yields *5*.

Div[<Dividend Polynomial>, <Divisor Polynomial>]

Returns the quotient of the two polynomials.

Example:

`Div[x^2 + 3 x + 1, x - 1]` yields $x + 4$.

Expand Command

Expand[<Function>]

Expands the function expression.

Example: `Expand[(2x - 1)² + 2x + 3]` returns the expression $4x^2 - 2x + 4$.

This command works in CAS too.

Factor Command

Factor[<Polynomial>]

Factors the polynomial.

Example:

`Factor[x^2 + x - 6]` yields $f(x) = (x - 2)\,(x + 3)$.

CAS Syntax

Factor[<Polynomial>]

Factors the polynomial.

Example:

`Factor[x^2 - y^2]` yields $(x + y)\,(x - y)$.

Factor[<Expression>, <Variable>]

Factorizes an expression with respect to a given variable.

Example:

- `Factor[x^2 - y^2, x]` yields $(x + y)\,(x - y)$, the factorization of $x^2 - y^2$ with respect to x,
- `Factor[x^2 - y^2, y]` yields $(-x - y)\,(-x + y)$, the factorization of $x^2 - y^2$ with respect to y.

Note:

See also CFactor Command.

GCD Command

GCD[Number a, Number b]

Calculates the greatest common divisor of numbers *a* and *b*.

Example:

`GCD[12, 15]` yields *3*.

GCD[List of Numbers]

Calculates the greatest common divisor of the list of numbers.

Example:

`GCD[{12, 30, 18}]` yields *6*.

CAS Syntax

GCD[<Number , Number>]

Calculates the greatest common divisor of the two numbers .

Example:

`GCD[12, 15]` yields *3*.

GCD[<List of Numbers>]

Calculates the greatest common divisor of the list of numbers.

Example:

`GCD[{12, 30, 18}]` yields *6*.

GCD[<Polynomial>, <Polynomial>]

Calculates the greatest common divisor of the two polynomials.

Example:

`GCD[x^2 + 4 x + 4, x^2 - x - 6]` yields $x + 2$.

GCD[<List of Polynomials>]

Calculates the greatest common divisor of the list of polynomials.

Example:

`GCD[{x^2 + 4 x + 4, x^2 - x - 6, x³ - 4x² - 3x + 18}]` yields $x + 2$.

LCM Command

UK English: LCM = lowest common multiple

LCM[Number a, Number b]

Calculates the least common multiple of two numbers *a* and *b*.

Example:

`LCM[12, 15]` yields *60*.

LCM[List of numbers]

Calculates the least common multiple of the elements of the list.

Example:

`LCM[{12, 30, 18}]` yields *180*.

CAS Syntax

LCM[Number a, Number b]

Calculates the least common multiple of numbers *a* and *b*.

Example:

`LCM[12, 15]` yields *60*.

LCM[List of Numbers]

Calculates the least common multiple of the list of numbers.

Example:

`LCM[{12, 30, 18}]` yields *180*.

LCM[Polynomial, Polynomial]

Calculates the least common multiple of the two polynomials.

Example:

`LCM[x^2 + 4 x + 4, x^2 - x - 6]` yields $x^3 + x^2 - 8x - 12$.

LCM[List of Polynomials]

Calculates the least common multiple of the list of polynomials.

Example:

`LCM[{x^2 + 4 x + 4, x^2 - x - 6, x³ - 4x² - 3x + 18}]` yields $x^4 - 2x^3 - 11x^2 + 12x + 36$.

Max Command

Max[<Number a>, <Number b>]

Yields the maximum of the given numbers *a* and *b*.

Example:

`Max[12, 15]` yields *15*.

Max[<List of Numbers>]

Yields the maximum of the numbers within the list.

Example:

`Max[{-2, 12, -23, 17, 15}]` yields *17*.

Note: If the input consists of non-numeric objects, then Max[] considers the numbers associated with those objects. For example, Max[List of Segments] will yield the maximum segment length.

Max[<Function>, <left-x>, <right-x>]

Calculates the maximum point for function in the given interval. Function should only have on maximum point in the interval.

Note: See also Extremum Command and Function Inspector Tool.

Max[<Interval>]

Returns the upper bound of the interval, e.g. `Max[2 < x < 3]` returns 3. It is the same for open and closed intervals.

CAS Syntax

Max[<Number a>, <Number b>]

Yields the maximum of the given numbers *a* and *b*.

Example:

`Max[12, 15]` yields *15*.

Max[<List of Numbers>]

Yields the maximum of the numbers within the list.

Example:

`Max[{-2, 12, -23, 17, 15}]` yields *17*.

Min Command

Min[<Number a>, <Number b>]

Yields the minimum of the given numbers *a* and *b*.

Example:

`Min[12, 15]` yields *12*.

Min[<List of Numbers>]

Yields the minimum of the numbers within the list.

Example:

`Min[{-2, 12, -23, 17, 15}]` yields -*23*.

Note: If the input consists of non-numeric objects, then Min[] considers the numbers associated with those objects. For example, Min[List of Segments] will yield the minimum segment length.

Min[<Function>, <left-x>, <right-x>]

Calculates (numerically) the minimum point for function in the given interval. Function should have only one minimum point in the interval.

Note: See also Extremum Command and Function Inspector Tool.

Min[<Interval>]

Returns the lower bound of the interval, e.g. `Min[2 < x < 3]` returns *2*. It is the same for open and closed intervals.

CAS Syntax

Min[<Number a>, <Number b>]

Yields the minimum of the given numbers *a* and *b*.

Example:

`Min[12, 15]` yields *12*.

Min[<List of Numbers>]

Yields the minimum of the numbers within the list.

Example:

`Min[{-2, 12, -23, 17, 15}]` yields -*23*.

Mod Command

Mod[<Integer a>, <Integer b>]

Yields the remainder when integer *a* is divided by integer *b*.

Example:

`Mod[9, 4]` yields *1*.

Mod[<Polynomial>, <Polynomial>]

Yields the remainder when the first entered polynomial is divided by the second polynomial.

Example:

`Mod[x^3 + x^2 + x + 6, x^2 - 3]` yields $9x + 4$.

CAS Syntax

Mod[<Integer a>, <Integer b>]

Yields the remainder when integer *a* is divided by integer *b*.

Example:

`Mod[9, 4]` yields *1*.

Mod[<Polynomial>, <Polynomial>]

Yields the remainder when the first entered polynomial is divided by the second polynomial.

Example:

`Mod[x^3 + x^2 + x + 6, x^2 - 3]` yields $9x + 4$.

PrimeFactors Command

PrimeFactors[<Number>]

Returns the list of primes whose product equals to the given number.

Example:

- `PrimeFactors[1024]` yields *{2, 2, 2, 2, 2, 2, 2, 2, 2, 2}*.
- `PrimeFactors[42]` yields *{2, 3, 7}*.

Note: See also Factors Command.

CAS Syntax

PrimeFactors[<Number>]

Returns the list of primes whose product equals to the given number.

Example:

- `PrimeFactors[1024]` yields *{2, 2, 2, 2, 2, 2, 2, 2, 2, 2}*.
- `PrimeFactors[42]` yields *{2, 3, 7}*.

Note: See also Factors Command.

Simplify Command

Simplify[<Function>]

Simplifies the terms of the given function, if possible.

Example:

`Simplify[x + x + x]` yields the function $f(x) = 3x$.

Simplify[<Text>]

Attempts to tidy up text expressions by removing repeated negatives etc.

Example:

For $a=b=c=-1$ `Simplify["f(x) = " + a + "x² + " + b + "x + " + c]` yields *f(x)=-x^2 - x - 1*.

Note: The FormulaText Command normally produces better results and is simpler.

CAS Syntax

Simplify[<Function>]

Simplifies the terms of the given function, if possible. Undefined variables can be included in the terms.

Example:

`Simplify[3 * x + 4 * x + a * x]` yields $x * (a + 7)$.

Text Commands

- FormulaText
- FractionText
- LetterToUnicode
- Ordinal
- RotateText
- TableText
- Text
- TextToUnicode
- UnicodeToLetter
- UnicodeToText
- VerticalText

See also Insert Text Tool.

FractionText Command

FractionText[Number]

Converts the number to a fraction, which is displayed as a (LaTeX) text object in the Graphics View.

Example: If $a{:}\ y = 1.5\,x + 2$ is a line, then `FractionText[Slope[a]]` gives you the fraction *3/2* as a text.

FormulaText Command

FormulaText[Object]

Returns the formula for the object as a LaTeX text.

Note: By default, values are substituted for variables.

Example: If $a = 2$ and $f(x) = a\,x^2$, then `FormulaText[f]` returns $2\,x^2$ (as a LaTeX text).

FormulaText[Object, Boolean]

Returns the formula for the object as LaTeX text. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).

Example: If $a = 2$ and $f(x) = a\,x^2$, then `FormulaText[f, true]` returns $2\,x^2$ (as a LaTeX text) and `FormulaText[f, false]` returns $a\,x^2$ (as a LaTeX text).

FormulaText[Object, Boolean for Substitution of Variables, Boolean Show Name]

Returns the formula for the object as LaTeX text. The first Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*), the second Boolean variable determines if the object name is shown in the text (*true*) or not (*false*).

Example: If $a = 2$ and $f(x) = a\,x^2$, then `FormulaText[f, true, true]` returns $f(x)=2\,x^2$ (as a LaTeX text) and `FormulaText[f, false, false]` returns $a\,x^2$ (as a LaTeX text).

LetterToUnicode Command

LetterToUnicode["Letter"]

Turns a single letter into its Unicode number.

Note: The letter needs to be in between a set of quotation marks.

Example: `LetterToUnicode["a"]` returns the number 97.

Ordinal Command

Ordinal[<Integer>]

Turns a number into an ordinal (as a text).

Example: `Ordinal[5]` returns "5th".

RotateText Command

RotateText[<Text>, <Angle>]

Returns text rotated by given angle. LaTeX is used for rendering of the result.

TableText Command

TableText[List 1, List 2, List 3,...]

Creates a text that contains a table of the list objects.

Note: By default, each list is displayed in its own row of the table.

Example:

TableText[{x^2, 4}, {x^3, 8}, {x^4, 16}] creates a table as a text object with three rows and two columns. All items of the table are left aligned. TableText[Sequence[i^2, i, 1, 10]] creates a table as a text object with one row. All items of the table are left aligned.

TableText[List 1, List 2, List 3,..., "Alignment of text"]

Creates a text that contains a table of the list objects. The optional text “Alignment of text” controls the orientation and alignment of the table text.

Note: Possible values are "vl", "vc", "vr", "v", "h", "hl", "hc", "hr". Default is "hl".

"v" = vertical, i. e. lists are columns "h" = horizontal, i. e. lists are rows "l" = left aligned "r" = right aligned "c" = centered

Example:

TableText[{1,2,3,4},{1,4,9,16},"v"] creates a text with two columns and four rows whose elements are left aligned. TableText[{1,2,3,4},{1,4,9,16},"h"] creates a text with two rows and four columns whose elements are left aligned. TableText[{11.2,123.1,32423.9,"234.0"},"vr"] creates a text with one column whose elements are right aligned.

now supports brackets by specifying ||||, ||, {}, [] or () in the String, horizontal lines by specifying _ and vertical lines by specifying |

Example:

TableText[{{1,2},{3,4}},"c()"]TableText[{{1,2},{3,4}},"cl_"]TableText[{{1,2},{3,4}},"cll"]TableText[{{1,2},{3,4}},"cllll"].

Text Command

Text[Object]

Returns the formula for the object as a text object.

Note: By default, values are substituted for variables.

Example: If $a = 2$ and $c = a^2$, then `Text[c]` returns the text "4".

Text[Object, Boolean]

Returns the formula for the object as a text object. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).

Example:

If $a = 2$ and $c = a^2$, then

- `Text[c, true]` returns the text "4" and
- `Text[c, false]` returns the text "a^2"

Text[Object, Point]

Returns the formula for the object as a text object at the position of the given point.

Example: `Text["hello", (2, 3)]` draws the text at the position *(2, 3)*.

Text[Object, Point, Boolean]

Returns the formula for the object as a text object at the position of the given point. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).

Text[Object, Point, Boolean substitute,Boolean LaTex]

Returns the formula for the object as a text object at the position of the given point. First Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*). If second boolean variable is true, the result is rendered using LaTeX.

TextToUnicode Command

TextToUnicode["Text"]

Turns the text into a list of Unicode numbers, one for each character.

Example:

TextToUnicode["Some text"] gives you the list of Unicode numbers {83, 111, 109, 101, 32, 116, 101, 120, 116}.If text1 is "hello", then TextToUnicode[text1] gives you the list of Unicode numbers {104, 101, 108, 108, 111}.

UnicodeToLetter Command

UnicodeToLetter[Integer]

Converts the integer Unicode number back into a letter which is displayed as a text object in the Graphics View.

Example: `UnicodeToLetter[97]` gives you the text *"a"*.

UnicodeToText Command

UnicodeToText[List of Integers]

Converts the integer Unicode numbers back into text.

Example: `UnicodeToText[{104, 101, 108, 108, 111}]` gives you the text *"hello"*.

VerticalText Command

VerticalText[Text]

Returns text rotated by 90° counter-clockwise. LaTeX is used for rendering of the result.

Logic Commands

- CountIf
- If
- IsDefined
- IsInRegion
- IsInteger
- KeepIf
- Relation

CountIf Command

CountIf[Condition, List]

Counts the number of elements in the list satisfying the condition.

Example:

CountIf[x < 3, {1, 2, 3, 4, 5}] gives you the number 2. CountIf[x < 3, A1:A10], where A1:A10 is a range of cells in the spreadsheet, counts all cells whose values are less than 3.

Note: For list of numbers arbitrary condition may be used. For list of other objects one can use only conditions of the form `x==constant` or `x!=constant`.

IsDefined Command

IsDefined[Object]

Returns *true* or *false* depending on whether the object is defined or not.

If Command

If[Condition, Object]

Yields a copy of the object if the condition evaluates to *true*, and an undefined object if it evaluates to *false*.

If[Condition, Object a, Object b]

Yields a copy of object *a* if the condition evaluates to *true*, and a copy of object *b* if it evaluates to *false*.

Warning: Both objects must be of the same type.

Conditional Functions

The *If* command can be used to create conditional functions. Such conditional functions may be used as arguments in any command that takes a function argument, such as Derivative, Integral, and Intersect.

Example:

- `f(x) = If[x < 3, sin(x), x^2]` yields a function that equals *sin(x)* for $x < 3$ and x^2 for $x \geq 3$
- `f(x) = If[x < 3 ∧ x>0, x^3]` yields a function that equals *sin(x)* for x between 0 and 3 and undefined for $x \geq 3$ or $x \leq 0$.

Note: See section: Boolean values for the symbols used in conditional statements.

Note: Derivative of **If[condition, f(x), g(x)]** gives **If[condition, f'(x), g'(x)]**. It does not do any evaluation of limits at the critical points.

If Command in Scripting

In many programming languages **if** is has the meaning "If condition holds, do something; otherwise do something else". In GeoGebra, arguments of If are not commands, but values, one of which becomes the value of the result. Therefore if you want to e.g. set value of *b* to 2 provided $a > 2$, correct way to do this is `SetValue[b,If[a>2,2,b]]`. The other way of nesting SetValue and If is incorrect.

IsInRegion Command

IsInRegion[<Point>, <Region>]

Returns *true* if the point is in given region and *false* otherwise.

IsInteger Command

IsInteger[Number]

Returns *true* or *false* depending whether the number is an integer or not.

KeepIf Command

KeepIf[Condition, List]

Creates a new list that only contains those elements of the initial list that fulfil the condition.

Example: `KeepIf[x<3, {1, 2, 3, 4, 1, 5, 6}]` returns the new list *{1, 2, 1}*.

Note: For list of numbers arbitrary condition may be used. For list of other objects one can use only conditions of the form `x==constant` or `x!=constant`.

Relation Command

Relation[Object a, Object b]

Shows a message box that gives you information about the relation between object *a* and object *b*.

This command allows you to find out whether

- two lines are perpendicular
- two lines are parallel
- two objects are equal
- a point lies on a line or conic
- a line is tangent or a passing line to a conic.

Note: See also Relation tool.

Functions & Calculus Commands

- Asymptote
- CompleteSquare
- Coefficients
- ComplexRoot
- Curvature
- CurvatureVector
- Curve
- Degree
- Denominator
- Derivative
- Extremum
- Factors
- Function
- ImplicitCurve
- InflectionPoint
- Integral
- IntegralBetween
- Intersect
- Iteration
- IterationList
- LeftSum
- Limit
- LimitAbove
- LimitBelow
- LowerSum
- Numerator
- OsculatingCircle
- PartialFractions
- PathParameter
- Polynomial
- RectangleSum
- Root
- RootList
- Roots
- SolveODE
- TaylorPolynomial
- TrapezoidalSum
- UpperSum

Asymptote Command

Asymptote[<Hyperbola>]

Yields both asymptotes of the hyperbola.

Asymptote[<Function>]

Yields a list containing all the asymptotes of the function.

Asymptote[<Implicit Curve>]

Yields a list containing all the asymptotes of the Implicit Curve.

Coefficients Command

Coefficients[<Polynomial>]

For polynomial $a_kx^k+a_{k-1}x^{k-1}+\cdots+a_1x+a_0$ returns list $\{a_0,a_1,\ldots,a_k\}$.

Coefficients[<Conic>]

For conic $a\cdot x^2+b\cdot y^2+c+d\cdot x\cdot y+e\cdot x+f\cdot y=0$ returns list $\{a,b,c,d,e,f\}$.

CompleteSquare Command

CompleteSquare[<Quadratic Function f>]

Returns quadratic function *f* in form $a(x-h)^2+k$.

ComplexRoot Command

ComplexRoot[<Polynomial>]

Finds the complex roots of a given polynomial in x. Points are created in Graphics View.

CAS Syntax

Note:

Use CSolve Command instead.

Curvature Command

Curvature[Point, Function]

Calculates the curvature of the function in the given point.

Curvature[Point, Curve]

Calculates the curvature of the curve in the given point.

CurvatureVector Command

CurvatureVector[Point, Function]

Yields the curvature vector of the function in the given point.

CurvatureVector[Point, Curve]

Yields the curvature vector of the curve in the given point.

Curve Command

Curve[Expression e1, Expression e2, Parameter t, Number a, Number b]

Yields the Cartesian parametric curve for the given *x*-expression *e1* and *y*-expression *e2* (using parameter *t*) within the given interval [*a*, *b*].

Example: Input of `c = Curve[2 cos(t), 2 sin(t), t, 0, 2 pi]` creates a circle with radius 2 around the origin of the coordinate system.

Note: Number *b* must be greater than or equal to number *a* and both must be finite

Note: *x* is not allowed as a parameter variable

See Curves for details.

Degree Command

Degree[<Polynomial>]

Gives the degree of a polynomial (in the main variable).

Example:

`Degree[x^4 + 2 x^2]` yields *4*.

CAS Syntax

Degree[<Polynomial>]

Gives the degree of a polynomial (in the main variable).

Example:

- `Degree[x^4 + 2 x^2]` yields *4*,
- `Degree[x^6 y^3 + 2 x^2 y^3]` yields *6*.

Degree[<Polynomial>, <Variable>]

Gives the degree of a polynomial in the given variable.

Example:

- `Degree[x^4 y^3 + 2 x^2 y^3, x]` yields *4*,
- `Degree[x^4 y^3 + 2 x^2 y^3, y]` yields *3*.

Denominator Command

Denominator[<Function>]

Returns the denominator of a function.

Example:

`Denominator[5 / (x^2 + 2)]` yields $f(x)=(x^2 + 2)$.

CAS Syntax

The following syntax works only in CAS View.

Denominator[<Expression>]

Returns the denominator of a rational number or expression.

Example:

`Denominator[2 / 3 + 1 / 15]` yields *15*.

Note:

See also Numerator Command.

Derivative Command

Derivative[<Function>]

Returns the derivative of the function with respect to the main variable.

Derivative[<Function>, <Number n>]

Returns the n^{th} derivative of the function with respect to the main variable.

Derivative[<Function>, <Variable>]

Returns the partial derivative of the function with respect to the given variable.

Example:

`Derivative[x³+3x y, x]` yields $3x^2+3y$.

Derivative[<Function>, <Variable>, <Number n>]

Returns the n^{th} partial derivative of the function with respect to the given variable.

Example:

`Derivative[x³+3x y, x, 2]` yields $6x$.

Derivative[<Curve>]

Returns the derivative of the curve.

Note: It only works for parametric curves.

Derivative[<Curve>, <Number n>]

Returns the n^{th} derivative of the curve.

Note: It only works for parametric curves.

Note: You can use `f'(x)` instead of `Derivative[f]`, or `f''(x)` instead of `Derivative[f, 2]`, and so on.

CAS Syntax

In CAS View only following syntax is supported:

Derivative[<Expression f>]

Returns derivative of *f* with respect to the main variable.

Example:

`Derivative[x^2]` yields $2\,x$.

Example:

`Derivative[t^3]` yields $3\,t^2$.

Derivative[<Expression f>, <Variable a>]

Returns derivative of *f* with respect to the given variable *a*.

Example:

`Derivative[a x^3, a]` yields x^3.

Derivative[<Expression f>, <Variable a>, <Number n>]

Returns the n^{th} derivative of *f* with respect to the given variable *a*.

Example:

`Derivative[a x^3, x, 2]` yields $6\,a\,x$.

Extremum Command

Extremum[Polynomial]

Yields all local extrema of the polynomial function as points on the function graph.

Extremum[Function *f*,left-x,right-x]

Calculates (numerically) the extremum of *f* in the open interval <left-x,right-x>. Function *f* should be continuous in [left-x,right-x], otherwise false extremums near discontinuity might be calculated.

Factors Command

Factors[<Polynomial>]

Returns list of lists *{factor,exponent}* such that product of all these factors raised to corresponding exponents equals the given polynomial.

Example:

`Factors[x^8 - 1]` yields **'.**

Note: Not all of the factors are irreducible over the reals.

Factors[<Number>]

Returns list of lists *{prime, exponent}* such that product of all these primes raised to corresponding exponents equals the given number. Primes are sorted in ascending order.

Example:

- `Factors[1024]` yields *{{2, 10}}*, because $1024 = 2^{10}$.
- `Factors[42]` yields **', because $42 = 2^1 3^1 7^1$.**

Note: See also PrimeFactors Command and Factor Command.

CAS Syntax

Factors[<Polynomial>]

Returns list of lists *{factor, exponent}* such that product of all these factors raised to corresponding exponents equals the given polynomial.

Example:

`Factors[x^8 - 1]` yields **', displayed as \(\begin{pmatrix}** x^4+1&1\\ x^2+1&1\\ x+1&1\\ x-1&1 \end{pmatrix}\).

Note: Not all of the factors are irreducible over the reals.

Factors[<Number>]

Returns list of lists *{prime, exponent}* such that product of all these primes raised to corresponding exponents equals the given number. Primes are sorted in ascending order.

Example:

- `Factors[1024]` yields *{{2, 10}}*, displayed as \(\begin{pmatrix}

2&10 \end{pmatrix}\), because $1024 = 2^{10}$.

- `Factors[42]` yields **', displayed as \(\begin{pmatrix}**

2&1\\ 3&1\\ 7&1 \end{pmatrix}\), because $42 = 2^1 3^1 7^1$.

Note: See also PrimeFactors Command and Factor Command.

Function Command

Function[Function f, Number a, Number b]

Yields a function graph, that is equal to *f* on the interval [*a*, *b*] and not defined outside of [*a*, *b*].

Note:

- This command should be used only to restrict the **display** interval of a function. To restrict the function's domain or use it with the Sequence command, create a conditional function with the If command, e.g. `f(x) = If[-1 < x < 1, x²]`.
- Example: `f(x) = Function[x^2, -1, 1]` produces a function equal to x^2 whose graph appears only in the interval [-*1*, *1*]. However, while `g(x) = 2 f(x)` will produce the function $g(x) = 2\, x^2$ as expected, this function is not restricted to the interval [-*1*, *1*].

ImplicitCurve Command

ImplicitCurve[<List of Points>]

Creates implicit curve through given set of points. The length of the list must be \(\frac{n(n+3)}2\) for implicit curve of degree \(n\).

ImplicitCurve[<f(x,y)>]

Creates the implicit curve f(x,y) = 0. Currently f(x,y) must be a polynomial in x and y.

Integral Command

Integral[Function]

Yields the indefinite integral for the given function with respect to the main variable.

Example:

`Integral[x³]` yields $x^4 / 4$.

Integral[<Function>, <Variable>]

Returns the partial integral of the function with respect to the given variable.

Example:

`Integral[x³+3x y, x]` yields $(x^2(x^2 + 6y)) / 4$.

Integral[Function, Number a, Number b]

Returns the definite integral of the function, with respect to the main variable, in the interval [*a* , *b*].

Note: This command also shadows the area between the function graph of *f* and the *x*-axis.

Integral[Function, Number a, Number b, Boolean Evaluate]

Returns the definite integral of the function, with respect to the main variable, in the interval [*a* , *b*] and shadows the related area when *Evaluate* = *true*. In case *Evaluate* = *false* the related area is shaded but the integral value is not calculated.

CAS Syntax

Integral[Function f]

Yields the indefinite integral for the given function with respect to the main variable.

Example: `Integral[cos(x)]` returns sin(x)+c1.

Integral[Function f, Variable t]

Returns the indefinite integral of the function with respect to the given variable *t*.

Example: `Integral[cos(a t), t]` returns sin(a t)/a+c2.

Integral[Function, Number a, Number b]

Returns the definite integral of the function, with respect to the main variable, in the interval [*a* , *b*].

Example: `Integral[cos(x), a, b]` returns sin(b) - sin(a).

Integral[Function f, Variable t, Number a, Number b]

Returns the definite integral in the interval [*a* , *b*] with respect to the given variable *t*.

Example: `Integral[cos(t), t, a, b]` returns sin(b) - sin(a).

IntegralBetween Command

IntegralBetween[Function f, Function g, Number a, Number b]

Returns the definite integral of the difference *f(x) - g(x)* in the interval [*a, b*] with respect to the main variable.

Note: This command also shades the area between the function graphs of *f* and *g*.

IntegralBetween[Function f, Function g, Number a, Number b, Boolean Evaluate]

Returns the definite integral of the difference *f(x) - g(x)* in the interval [*a, b*] with respect to the main variable and shadows the related area when *Evaluate = true*. In case *Evaluate = false* the related area is shaded but the integral value is not calculated.

CAS Syntax

IntegralBetween[Function f, Function g, Number a, Number b]

Returns the definite integral of the difference *f(x) - g(x)* in the interval [*a, b*] with respect to the main variable.

Example:

`IntegralBetween[sin(x), cos(x), π / 4, π * 5 / 4]` yields \(2 \sqrt{2}\).

IntegralBetween[Function f, Function g, Variable t, Number a, Number b]

Returns the definite integral of the difference *f - g* in the interval [*a, b*] with respect to the given variable t.

Example:

`IntegralBetween[a * sin(t), a * cos(t), t, π / 4, π * 5 / 4]` yields \(2 \sqrt{2} a\).

Intersect Command

Intersect[Line g, Line h]

Yields the intersection point of lines *g* and *h*.

Intersect[Line, Conic]

Yields all intersection points of the line and conic section (max. 2).

Intersect[Line, Conic, Number n]

Yields the n^{th} intersection point of the line and the conic section.

Intersect[Conic c1, Conic c2]

Yields all intersection points of conic sections *c1* and *c2* (max. 4).

Intersect[Conic c1, Conic c2, Number n]

Yields the n^{th} intersection point of conic sections *c1* and *c2*.

Intersect[Polynomial f1, Polynomial f2]

Yields all intersection points of polynomials *f1* and *f2*.

Intersect[Polynomial f1, Polynomial f2, Number n]

Yields the n^{th} intersection point of polynomials *f1* and *f2*.

Intersect[Polynomial, Line]

Yields all intersection points of the polynomial and the line.

Intersect[Polynomial, Line, Number n]

Yields the n^{th} intersection point of the polynomial and the line.

Intersect[Function f, Function g, Point A]

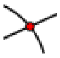
Calculates the intersection point of functions *f* and *g* by using Newton's method with initial point *A*.

Intersect[Function, Line, Point A]

Calculates the intersection point of the function and the line by using Newton's method with initial point *A*.

Intersect[Function f, Function g, left-x, right-x]

Calculates the intersection points for the two functions in the given interval.

Note: Also see tool  Intersect Two Objects.

Iteration Command

Iteration[Function, Number x0, Number n]

Iterates the function *n* times using the given start value *x0*.

Example: After defining `f(x) = x^2` the command `Iteration[f, 3, 2]` gives you the result $(3^2)^2=81$.

IterationList Command

IterationList[Function, Number x0, Number n]

Gives you a list of length *n+1* whose elements are iterations of the function starting with the value *x0*.

Example: After defining `f(x) = x^2` the command `IterationList[f, 3, 2]` gives you the list *L = {3, 9, 81}*.

LeftSum Command

LeftSum[<Function>, <Start x-Value>, <End x-Value>, <Number of Rectangles>]

Calculates the left sum of the function in the interval [*a, b*] using *n* rectangles.

Note:

This command draws the rectangles of the left sum as well. See also RectangleSum Command, LowerSum Command and UpperSum Command.

Limit Command

Limit[<Function f>, <Value t>]

Computes limit of function *f* for given value t of the main function variable.

Note: The limit might be infinity.

CAS Syntax

Limit[<Expression f>, <Value t>]

Computes limit of function *f* for given value t of the main function variable.

Example:

`Limit[a sin(x)/x, 0]` yields *a* .

Limit[<Expression f>, <Variable v>, <Value t>]

Computes limit of function *f* for given value t of the given function variable v.

Example:

`Limit[a sin(v)/v, v, 0]` yields *a* .

Note: See also Asymptote Command, LimitAbove Command and LimitBelow Command.

LimitAbove Command

LimitAbove[<Function f>, <Value t>]

Returns right one-sided limit of the function *f* for given value *t* of the main function variable.

CAS Syntax

LimitAbove[<Expression f>, <Value t>]

Computes the right one-sided limit of the function *f* for the given value *t* of the main function variable.

Example:

`LimitAbove[1 / x, 0]` yields \(\infty\) .

LimitAbove[<Expression f>, <Variable v>, <Value t>]

Computes the right one-sided limit of the multivariate function *f* for the given value *t* of the given function variable *v*.

Example:

`LimitAbove[1 / a, a, 0]` yields \(\infty\) .

Note:

See also Limit Command and LimitBelow Command.

LimitBelow Command

LimitBelow[<Function f>, <Value t>]

Returns left one-sided limit of the function *f* for given value *t* of the main function variable.

CAS Syntax

LimitBelow[<Expression f>, <Value t>]

Computes the left one-sided limit of the function *f* for the given value *t* of the main function variable.

Example:

`LimitBelow[1 / x, 0]` yields \(-\infty\) .

LimitBelow[<Expression f>, <Variable v>, <Value t>]

Computes the left one-sided limit of the multivariate function *f* for the given value *t* of the given function variable *v*.

Example:

`LimitBelow[1 / a, a, 0]` yields \(-\infty\) .

Note:

See also Limit Command and LimitAbove Command.

LowerSum Command

LowerSum[Function, Number a, Number b, Number n]

Yields the lower sum of the given function on the interval [*a, b*] with *n* rectangles.

Note: This command draws the rectangles for the lower sum as well.

Numerator Command

Numerator[<Function>]

Returns the numerator of the function.

Example:

`Numerator[(3x² + 1) / (2x - 1)]` yields $f(x) = 3x^2 + 1$.

CAS Syntax

The following syntax works only in CAS View.

Numerator[<Expression>]

Returns the numerator of a rational number or expression.

Example:

`Numerator[2/3 + 1/15]` yields *11*.

Note:

See also Denominator Command.

OsculatingCircle Command

OsculatingCircle[Point, Function]

Yields the osculating circle of the function in the given point.

OsculatingCircle[Point, Curve]

Yields the osculating circle of the curve in the given point.

PartialFractions Command

PartialFractions[<Function>]

Yields, if possible, the partial fraction of the given function for the main function variable. The graph of the function is plotted in Graphics View.

Example:

`PartialFractions[x^2 / (x^2 - 2x + 1)]` yields $1 + \frac{2}{x - 1} + \frac{1}{x^2 - 2x + 1}$.

CAS Syntax

PartialFractions[<Function>]

Yields, if possible, the partial fraction of the given function for the main function variable.

Example:

`PartialFractions[x^2 / (x^2 - 2x + 1)]` yields $1 + \frac{2}{x - 1} + \frac{1}{x^2 - 2x + 1}$.

PartialFractions[<Function>, <Variable>]

Yields, if possible, the partial fraction of the given function for the given function variable.

Example:

`PartialFractions[a^2 / (a^2 - 2a + 1), a]` yields $1 + \frac{2}{a - 1} + \frac{1}{a^2 - 2a + 1}$.

PathParameter Command

PathParameter[<Point On Path>]

Returns the parameter (i.e. a number ranging from 0 to 1) of the point that belongs to a path.

Polynomial Command

Polynomial[Function]

Yields the expanded polynomial function.

Example: `Polynomial[(x - 3)^2]` yields $x^2 - 6x + 9$.

Polynomial[List of n points]

Creates the interpolation polynomial of degree *n-1* through the given *n* points.

RectangleSum Command

RectangleSum[<Function>, <Start x-Value>, <End x-Value>, <Number of Rectangles>, <Position for rectangle start>]

Calculates the sum of rectangles with left height starting at a fraction *d* of each interval using *n* rectangles. ($0<=d<=1$).

When *d*=0 this is equivalent to the LowerSum Command and when *d*=1 this is equivalent to the UpperSum Command.

Note:

This command draws the rectangles of the left sum as well. See also LeftSum Command.

Root Command

Root[<Polynomial>]

Yields all roots of the polynomial as intersection points of the function graph and the *x*-axis.

Root[<Function>, <Number a>]

Yields one root of the function using the initial value *a* for Newton's method.

Root[<Function>, <Number a>, <Number b>]

Yields one root of the function in the interval [*a, b*] (regula falsi).

CAS Syntax

Root[<Polynomial>]

Yields all roots of the polynomial as intersection points of the function graph and the *x*-axis.

Example:

`Root[x^3 - 3 * x^2 - 4 * x + 12]` yields $\{x = 3, x = 2, x = -2\}$.

Note:

This command is only a special variant of Solve Command.

RootList Command

RootList[<List>]

Given a list of numbers $\{a_1,a_2,...,a_n\}$ creates list of points $\{(a_1,0),(a_2,0),...,(a_n,0)\}$.

Roots Command

Roots[<Function>, <Start x-Value>, <End x-Value>]

Calculates the roots for function in the given interval. The function must be continuous on that interval. Because this algorithm is numeric, it may not find all the roots in some cases.

SolveODE Command

SolveODE[<f'(x,y)>, <Start x>, <Start y>, <End x>, <Step>]

Solves first order ordinary differential equations (ODE) $\frac{dy}{dx}=f'(x,y)$ numerically given start point and end & step for *x*.

For example to solve $\frac{dy}{dx}=-xy$ using *A* as a starting point, enter SolveOde[-x*y, x(A), y(A), 5, 0.1]

Note: Length[<Locus>] allows you to find out how many points are in the computed locus and First[<Locus>, <Number>] allows you to extract the points as a list, for example First[loc1, Length[loc1]].

SolveODE[<f(x,y)>, <g(x,y)>, <Start x>, <Start y>, <End t>, <Step>]

Solves first order ODE $\frac{dy}{dx}=\frac{f(x,y)}{g(x,y)}$ given start point, maximal value of an internal parameter *t* and step for *t*. This version of the command may work where the first one fails eg when the solution curve has vertical points.

For example, to solve $\frac{dy}{dx}=- \frac{x}{y}$ using *A* as a starting point, enter SolveOde[-x, y, x(A), y(A), 5, 0.1].

SolveODE[<b(x)>, <c(x)>, <f(x)>, <Start x>, <Start y>, <Start y'>, <End x>, <Step>]

Solves second order ODE $y+b(x)y'+c(x)y=f(x)$.

Note: Always returns the result as locus. The algorithms are currently based on Runge-Kutta numeric methods.

CAS Syntax

Following two syntaxes work only in CAS View.

SolveODE[<f(x, y)>]

Attempts to find the exact solution of the first order ODE $\frac{dy}{dx}(x)=f(x, y(x))$.

Example:

`SolveODE[y / x]` yields $y = c_1 x$.

SolveODE[<f(v, w)>, <Dependent Variable v>, <Independent Variable w>]

Attempts to find the exact solution of the first order ODE $\frac{dv}{dw}(w)=f(w, v(w))$.

Example:

`SolveODE[y / x, y, x]` yields $y = c_1 x$.

TaylorPolynomial Command

TaylorPolynomial[<Function>, <Number a>, <Number n>]

Creates the power series expansion for the given function about the point $x = a$ to order n.

Example:

`TaylorPolynomial[x^2, 3, 1]` gives $6 x - 9$, the power series expansion of x^2 at $x = 3$ to order 1.

CAS Syntax

TaylorPolynomial[<Function>, <Number a>, <Number n>]

Creates the power series expansion for the given function about the point $x = a$ to order n.

Example:

`TaylorPolynomial[x^2, a, 1]` gives $-a^2 + 2 a x$, the power series expansion of x^2 at $x = a$ to order 1.

TaylorPolynomial[<Function>, <Variable>, <Number a>, <Number n>]

Creates the power series expansion for the given function with respect to the given variable about the point *Variable* = a to order n.

Example:

`TaylorPolynomial[x^3 sin(y), x, 3, 2]` gives $sin(y)\ (9 x^2 - 27 x + 27)$, the power series expansion with respect to x of $x^3 sin(y)$ at $x = 3$ to order 2.

Example:

`TaylorPolynomial[x^3 sin(y), y, 3, 2]` gives \(\frac{cos(3) x^{3} (2 y - 6) + sin(3) x^{3} (-y^{2} + 6 y - 7)}{2}\) , the power series expansion with respect to y of $x^3 sin(y)$ at $y = 3$ to order 2.

Note: The order n has got to be an integer greater or equal to zero.

TrapezoidalSum Command

TrapezoidalSum[Function, Number a, Number b, Number n]

Calculates the trapezoidal sum of the function in the interval [*a, b*] using *n* trapezoids.

Note: This command draws the trapezoids of the trapezoidal sum as well.

InflectionPoint Command

InflectionPoint[Polynomial]

Yields all inflection points of the polynomial as points on the function graph.

UpperSum Command

UpperSum[Function, Number a, Number b, Number n]

Calculates the upper sum of the function on the interval [*a, b*] using *n* rectangles.

Note: This command draws the rectangles of the upper sum as well.

Conic Commands

- Asymptote
- Axes
- Center
- Circle
- Conic
- ConjugateDiameter
- Directrix
- Eccentricity
- Ellipse
- Focus
- Hyperbola
- LinearEccentricity
- MajorAxis
- MinorAxis
- Parabola
- Parameter
- Polar
- Semicircle
- SemiMajorAxisLength
- SemiMinorAxisLength

See also Conic section tools.

Asymptote Command

Asymptote[<Hyperbola>]

Yields both asymptotes of the hyperbola.

Asymptote[<Function>]

Yields a list containing all the asymptotes of the function.

Asymptote[<Implicit Curve>]

Yields a list containing all the asymptotes of the Implicit Curve.

Axes Command

Axes[Conic]

Returns the major and minor axes of a conic section.

Center Command

Center[Conic]

Returns the center of a circle, ellipse, or hyperbola.

Note: See also Midpoint or Center tool .

Circle Command

Circle[Point M, Number r]

Yields a circle with center *M* and radius *r*.

Circle[Point M, Segment]

Yields a circle with center *M* and radius equal to the length of the given segment.

Circle[Point M, Point A]

Yields a circle with center *M* through point *A*.

Circle[Point A, Point B, Point C]

Yields a circle through the given points *A*, *B* and *C*.

Note: See also Compass, Circle with Center through Point, Circle with Center and Radius, and Circle through Three Points tools.

Conic Command

Conic[Point A, Point B, Point C, Point D, Point E]

Returns a conic section through the five given points *A*, *B*, *C*, *D*, and *E*.

Note: If four of the points lie on one line the conic section is not defined.

Conic[<Number a>, <Number b>, <Number c>, <Number d>, <Number e>, <Number f>]

Returns a conic section $ax^2 + bxy + cy^2 + dx + ey + f = 0$.

Note: See also Conic through Five Points tool.

ConjugateDiameter Command

ConjugateDiameter[Line, Conic]

Returns the conjugate diameter of the diameter that is parallel to the line (relative to the conic section).

ConjugateDiameter[Vector, Conic]

Returns the conjugate diameter of the diameter that is parallel to the vector (relative to the conic section).

Directrix Command

Directrix[Parabola]

Yields the directrix of the parabola.

Eccentricity Command

Eccentricity[<Conic>]

Calculates the eccentricity of the conic section.

Ellipse Command

Ellipse[Point F, Point G, Number a]

Creates an ellipse with focal points *F* and *G* and semimajor axis length *a*.

Ellipse[Point F, Point G, Segment]

Creates an ellipse with focal points *F* and *G* where the length of the semimajor axis equals the length of the given segment.

Ellipse[Point F, Point G, Point A]

Creates an ellipse with foci *F* and *G* passing through point *A*.

Note: See also Ellipse tool .

LinearEccentricity Command

LinearEccentricity[Conic]

Calculates the linear eccentricity of the conic section.

Note: The linear eccentricity is the distance between the conic center and its focus (or one of its two foci).

MajorAxis Command

MajorAxis[Conic]

Returns the major axis of the conic section.

SemiMajorAxisLength Command

SemiMajorAxisLength[Conic]

Returns the length of the semimajor axis (half of the major axis) of the conic section.

Focus Command

Focus[Conic]

Yields (all) foci of the conic section.

Hyperbola Command

Hyperbola[Point F, Point G, Number a]

Creates a hyperbola with focal points *F* and *G* and semimajor axis length *a*.

Note: Condition: *0 < 2a < Distance[F, G]*

Hyperbola[Point F, Point G, Segment s]

Creates a hyperbola with focal points *F* and *G* where the length of the semimajor axis equals the length of segment *s*.

Hyperbola[Point F, Point G, Point A]

Creates a hyperbola with foci *F* and *G* passing through point *A*.

Note: See also Hyperbola tool .

Incircle Command

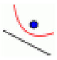
Incircle[<Point>, <Point>, <Point>]

Returns Incircle of the triangle formed by the three Points.

Parabola Command

Parabola[Point F, Line g]

Returns a parabola with focal point *F* and directrix *g*.

Note: See also  Parabola tool .

Parameter Command

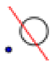
Parameter[Parabola]

Returns the parameter of the parabola, which is the distance between the directrix and the focus.

Polar Command

Polar[Point, Conic]

Creates the polar line of the given point relative to the conic section.

Note: See also  Polar or Diameter Line tool.

MinorAxis Command

MinorAxis[Conic]

Returns the minor axis of the conic section.

SemiMinorAxisLength Command

SemiMinorAxisLength[Conic]

Returns the length of the semiminor axis (half of the minor axis) of the conic section.

Semicircle Command

Semicircle[Point A, Point B]

Creates a semicircle above the segment *AB*.

Note: See also Semicircle tool.

List Commands

- Append
- Classes
- Element
- First
- Frequency
- IndexOf
- Insert
- Intersection
- IterationList
- Join
- Last
- OrdinalRank
- PointList
- Product
- RandomElement
- RemoveUndefined
- Reverse
- RootList
- SelectedElement
- SelectedIndex
- Sequence
- Sort
- Take
- TiedRank
- Union

- Unique
- Zip

Append Command

Append[List, Object]

Appends the object to the list.

Example: `Append[{1, 2, 3}, 4]` gives you *{1, 2, 3, 4}*.

Append[Object, List]

Appends the list to the object.

Example: `Append[4, {1, 2, 3}]` gives you *{4, 1, 2, 3}*.

Classes Command

Classes[<List of Data L>, <Start S>, <Width of Classes w>]

Gives a list of class boundaries. First boundary is equal to *S*, last boundary is at least equal the maximum of *L*, distances between consequent boundaries are equal to *w*.

Classes[<List of Data L>, <Number of Classes>]

Gives a list of class boundaries. First boundary is equal to minimum of *L*, last boundary to maximum of *L*, distances between consequent boundaries are equal.

Element Command

Element[<List>, <Number n>]

Yields the n^{th} element of the list.

Example:

`Element[{1, 3, 2}, 2]` yields *3*, the second element of *{1, 3, 2}*.

Element[<Matrix>, <Row>, <Column>]

Yields the element of the matrix in the given row and column.

Example:

`Element[{{1, 3, 2}, {0, 3, -2}}, 2, 3]` yields *-2*, the third element of the second row of $\begin{pmatrix}1&3&2\\0&3&-2\end{pmatrix}$.

Element[<List L>, <Index1>, <Index2>, ...]

Provided L is *n*-dimensional list, one can specify up to *n* indices to obtain an element (or list of elements) at given coordinates.

Example:

Let *L={{{1, 2}, {3, 4}}, {{5, 6}, {7, 8}}}*. Then `Element[L, 1, 2, 1]` yields *3*, `Element[L, 2, 2]` yields *{7, 8}*.

Note: For this command to work, the list or matrix can contain elements of one object type only (e. g. only numbers or only points).

CAS Syntax

Element[<List>, <Number n>]

Yields the n^{th} element of the list.

Example:

`Element[{a, b, c}, 2]` yields *b*, the second element of *{a, b, c}*.

Element[<Matrix>, <Row>, <Column>]

Yields the element of the matrix in the given row and column.

Example:

`Element[{{a, b, c}, {d, e, f}}, 2, 3]` yields *f*, the third element of the second row of\(\begin{pmatrix}a&b&c\\d&e&f\end{pmatrix}\).

Note:

See also First Command, Last Command and RandomElement Command.

First Command

First[<List L>]

Gives a new list that contains the first element of the list *L*.

Example:

`First[{1, 4, 3}]` yields *{1}*.

Note: To get the first element use `Element[{1, 4, 3}, 1]`.

First[<List L>, <Number n of elements>]

Gives a new list that contains just the first *n* elements of the list *L*.

Example:

`First[{1, 4, 3}, 2]` yields *{1, 4}*.

First[<Text>]

Gives first character of the text.

Example:

`First["Hello"]` yields *"H"*.

First[<Text> , <Number n of elements>]

Gives the first *n* characters of the text.

Example:

`First["Hello",2]` yields *"He"*.

First[<Locus>, <Number n of elements>]

This command is useful for

- loci generated by SolveODE Command - It returns list points that were created in the first *n* steps of the numeric ODE-solving algorithm.
- loci generated using ShortestDistance Command, TravelingSalesman Command, Voronoi Command, MinimumSpanningTree Command, ConvexHull Command and Hull Command Commands - it returns vertices of the graph

CAS Syntax

First[<List L>]

Gives a new list that contains the first element of the list *L*.

Example:

`First[{1, 4, 3}]` yields *{1}*.

Note: To get the first element use `Element[{1, 4, 3}, 1]`.

First[<List L>, <Number n of elements>]

Gives a new list that contains just the first *n* elements of the list *L*.

Example:

`First[{1, 4, 3}, 2]` yields *{1, 4}*.

Note:

See also Last Command.

Frequency Command

Frequency[<List of Raw Data>]

Returns a list with a count of the occurrences of each unique value in the given list of data. This input list can be numbers or text. The list is sorted in ascending order of the unique values. To get a list of the corresponding unique values use the command Unique[<List of Raw Data>]

Example: Enter `list1 = { "a", "a", "x", "x", "x", "b" }`.`Frequency[list1]` returns the list { 2, 1, 3 }. `Unique[list1]` returns the list { "a", "b", "x" }.

Frequency[<Cumulative>, <List of Raw Data>]

If Cumulative = false, returns the same list as Frequency[<List of Raw Data>]

If Cumulative = true, returns a list of cumulative frequencies for Frequency[<List of Raw Data>]

Example: Enter `list1 = { 0, 0, 0, 1 , 1, 2 }`.`Frequency[true, list1]` returns the list { 3, 5, 6 }. `Frequency[false, list1]` returns the list { 3, 2, 1}. `Unique[list1]` returns the list { 0, 1, 2 }

Frequency[<List of Class Boundaries>, <List of Raw Data>]

Returns a list of the counts of values from the given data list that lie within the intervals formed by the given class boundaries. All intervals except the highest interval are of the form [a, b). The highest interval has the form [a, b].

Example: `Frequency[{1,2,3}, {1,1,2,3}]` returns the list { 2, 2 }.

Frequency[<Cumulative>,<List of Class Boundaries>,<List of Raw Data>]

If Cumulative = false, returns the same list as Frequency[<List of Class Boundaries>, <List of Raw Data>]

If Cumulative = true, returns a list of cumulative frequencies for Frequency[<List of Class Boundaries>, <List of Raw Data>]

Frequency[<List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a list of frequencies for the corresponding Histogram Command.

Frequency[<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a list of frequencies for the corresponding Histogram Command.

IndexOf Command

IndexOf[<Object>, <List>]

Returns position of first occurrence of Object in List, e.g. `IndexOf[5,{1,3,5,2,5,4}]` returns 3. When the object is not found, result is *undefined*.

IndexOf[<Object>, <List>, <Start Index>]

Same as above, but the search starts at given index.

Example:

- `IndexOf[5,{1,3,5,2,5,4},3]` returns 3.
- `IndexOf[5,{1,3,5,2,5,4},4]` returns 5.
- `IndexOf[5,{1,3,5,2,5,4},6]` returns *undefined*.

IndexOf[<Text Needle>, <Text Haystack>]

Returns position of first occurrence of Needle in Haystack, e.g. `IndexOf["Ge","GeoGebra"]` returns 1.

IndexOf[<Text Needle>, <Text Haystack>, <Start Index>]

Same as above, but the search starts at given index.

Example: `IndexOf["Ge","GeoGebra",2]` returns 4.

Insert Command

Insert[Object, List, Position]

Inserts the object in the list at the given position.

Example: `Insert[x^2, {1, 2, 3, 4, 5}, 3]` places x^2 at the third position and gives you the list *{1, 2,* x^2*, 3, 4, 5}*.

Note: If the position is a negative number, then the position is counted from the right.

Example: `Insert[x^2, {1, 2, 3, 4, 5}, -1]` places x^2 at the end of the list and gives you the list *{1, 2, 3, 4, 5,* x^2*}*.

Insert[List 1, List 2, Position]

Inserts all elements of *list1* in *list2* at the given position.

Example: `Insert[{11, 12}, {1, 2, 3, 4, 5}, 3]` places the elements of *list1* at the third (and following) position(s) of *list2* and gives you the list *{1, 2, 11, 12, 3, 4, 5}*.

Note: If the position is a negative number, then the position is counted from the right.

Example: `Insert[{11, 12}, {1, 2, 3, 4, 5}, -2]` places the elements of *list1* at the end of *list2* before its last element and gives you *{1, 2, 3, 4, 11, 12, 5}*.

Intersect Command

Intersect[Line g, Line h]

Yields the intersection point of lines *g* and *h*.

Intersect[Line, Conic]

Yields all intersection points of the line and conic section (max. 2).

Intersect[Line, Conic, Number n]

Yields the n^{th} intersection point of the line and the conic section.

Intersect[Conic c1, Conic c2]

Yields all intersection points of conic sections *c1* and *c2* (max. 4).

Intersect[Conic c1, Conic c2, Number n]

Yields the n^{th} intersection point of conic sections *c1* and *c2*.

Intersect[Polynomial f1, Polynomial f2]

Yields all intersection points of polynomials *f1* and *f2*.

Intersect[Polynomial f1, Polynomial f2, Number n]

Yields the n^{th} intersection point of polynomials *f1* and *f2*.

Intersect[Polynomial, Line]

Yields all intersection points of the polynomial and the line.

Intersect[Polynomial, Line, Number n]

Yields the n^{th} intersection point of the polynomial and the line.

Intersect[Function f, Function g, Point A]

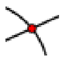
Calculates the intersection point of functions *f* and *g* by using Newton's method with initial point *A*.

Intersect[Function, Line, Point A]

Calculates the intersection point of the function and the line by using Newton's method with initial point *A*.

Intersect[Function f, Function g, left-x, right-x]

Calculates the intersection points for the two functions in the given interval.

Note: Also see tool  Intersect Two Objects.

Intersection Command

Intersection[List 1, List 2]

Gives you a new list containing all elements that are part of both lists.

IterationList Command

IterationList[Function, Number x0, Number n]

Gives you a list of length *n+1* whose elements are iterations of the function starting with the value *x0*.

Example: After defining `f(x) = x^2` the command `IterationList[f, 3, 2]` gives you the list *L = {3, 9, 81}*.

Join Command

Join[List 1, List 2, ...]

Joins the two (or more) lists.

Note: The new list contains all elements of the initial lists even if they are the same. The elements of the new list are not re-ordered.

Example: `Join[{5, 4, 3}, {1, 2, 3}]` creates the list *{5, 4, 3, 1, 2, 3}*.

Join[List of lists]

Joins the sub-lists into one longer list.

Note: The new list contains all elements of the initial lists even if they are the same. The elements of the new list are not re-ordered.

Example:

Join[{{1, 2}}] creates the list {1, 2}.Join[] creates the list {1, 2, 3, 3, 4, 8, 7}.

Last Command

Last[<List L>]

Gives a new list that contains the last element of the list *L*.

Example:

`Last[{1, 4, 3}]` yields *{3}*.

Note: To get the last element use `Element[{1, 4, 3}, 3]`.

Last[<List L>, <Number n of elements>]

Gives a new list that contains just the last *n* elements of the list *L*.

Example:

`Last[{1, 4, 3}, 2]` yields *{4, 3}*.

Last[<Text>]

Gives last character of the text.

Example:

`Last["Hello"]` yields *"o"*.

Last[<Text> , <Number n of elements>]

Gives the last *n* characters of the text.

Example:

`Last["Hello",2]` yields *"lo"*.

CAS Syntax

Last[<List L>]

Gives a new list that contains the last element of the list *L*.

Example:

`Last[{1, 4, 3}]` yields *{3}*.

Note: To get the last element use `Element[{1, 4, 3}, 3]`.

Last[<List L>, <Number n of elements>]

Gives a new list that contains just the last *n* elements of the list *L*.

Example:

`Last[{1, 4, 3}, 2]` yields *{4, 3}*.

Note:

See also First Command.

OrdinalRank Command

OrdinalRank[<List L>]

Returns a list, whose *i*-th element is the rank of *i*-th element of L (rank of element is its position in Sort[L]). If there are more equal elements in L which occupy positions from *k* to *l* in Sort[L], ranks from *k* to *l* are associated with these elements.

Example: `OrdinalRank[{4, 1, 2, 3, 4, 2}]` returns {5, 1, 2, 4, 6, 3}.

Example: `OrdinalRank[{3, 2, 2, 1}]` returns {4, 2, 3, 1}.

Note: Also see command: TiedRank

PointList Command

PointList[<List>]

Creates list of points from a list of two-element lists.

Example: `PointList[]` returns {(1,2),(3,4)}.

Product Command

Product[<List of Numbers>]

Calculates the product of all numbers in the list.

Product[<List of Numbers>, <Number of Elements>]

Calculates the product of the first *n* list elements.

Example:

`Product[{1, 2, 3, 4}, 3]` yields *6*.

CAS Specific Syntax

In CAS View only the following syntax is allowed:

Product[<List of expressions>]

Calculates the product of all elements in the list.

Example:

`Product[{1, 2, x}]` yields *2x*.

Product[<Expression>, <Variable>, <Start>, <End>]

Calculates the product of the expressions that are obtained by replacing the given variable with every integer from *start* to *end*.

Example:

`Product[x + 1, x, 2, 3]` yields *12*.

RandomElement Command

RandomElement[<List>]

Returns randomly chosen element from the list (with uniform probability). All elements in the list must be of the same type.

Example:

`RandomElement[{3, 2, -4, 7}]` yields one of *{-4, 2, 3, 7}*.

Note:

See also Element Command.

CAS Syntax

RandomElement[<List>]

Returns randomly chosen element from the list (with uniform probability). All elements in the list must be of the same type.

Example:

`RandomElement[{3, 2, -4, 7}]` yields one of *{-4, 2, 3, 7}*.

Note:

See also Element Command.

RemoveUndefined Command

RemoveUndefined[List]

Removes undefined objects from a list.

Example: `RemoveUndefined[Sequence[(-1)^i, i, -3, -1, 0.5]]` removes the second and fourth elements of the sequence since expressions \((-1)^{1.5}\) and \((-1)^{2.5}\) are undefined.

Reverse Command

Reverse[List]

Reverses the order of a list.

RootList Command

RootList[<List>]

Given a list of numbers $\{a_1,a_2,...,a_n\}$ creates list of points $\{(a_1,0),(a_2,0),...,(a_n,0)\}$.

SelectedElement Command

SelectedElement[<List>]

Returns the element of a Visible List (i.e. Combobox) that has been selected by the user. Note that these are currently available only in the Spreadsheet View (when the *Use Buttons and Checkboxes* option is enabled).

SelectedIndex Command

SelectedIndex[<List>]

Returns the index of selected element of a Visible List (i.e. Combobox). Note that these are currently available only in the Spreadsheet View (when the *Use Buttons and Checkboxes* option is enabled).

Sequence Command

Sequence[<Expression>, <Variable i>, <Number a>, <Number b>]

Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b*.

Example:

`L = Sequence[(2, i), i, 1, 5]` creates a list of points whose *y*-coordinates range from 1 to 5: `L = {(2, 1), (2, 2), (2, 3), (2, 4), (2, 5)}`.

Sequence[<Expression>, <Variable i>, <Number a>, <Number b>, <Increment>]

Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b* with given increment.

Example:

`L = Sequence[(2, i), i, 1, 3, 0.5]` creates a list of points whose *y*-coordinates range from 1 to 3 with an increment of 0.5: `L = {(2, 1), (2, 1.5), (2, 2), (2, 2.5), (2, 3)}`.

Note: Since the parameters *a* and *b* are dynamic you could use slider variables in both cases above as well.

Sequence[<Number b>]

Creates list of numbers 1 to *b*, e.g. {1, 2, ..., b}.

Example:

`L = 2^Sequence[4]` creates list *{2, 4, 8, 16}*.

Note: See Lists for more information on list operations.

CAS Syntax

Sequence[<Expression>, <Variable i>, <Number a>, <Number b>]

Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b*.

Example:

`Sequence[x^i, i, 1, 10]` generates the sequence $\{x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}\}$.

Sequence[<Expression>, <Variable i>, <Number a>, <Number b>, <Increment>]

Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b* with given increment.

Example:

`Sequence[x^i, i, 1, 10, 2]` generates the sequence $\{x, x^3, x^5, x^7, x^9\}$.

Sequence[<Number b>]

Creates list of numbers 1 to *b*, e.g. {1, 2, ..., b}.

Example:

`Sequence[5]` generates the sequence *{1, 2, 3, 4, 5}*.

Sort Command

Sort[List]

Sorts a list of numbers, text objects, or points.

Note: Lists of points are sorted by *x*-coordinates.

Example:

- `Sort[{3, 2, 1}]` gives you the list *{1, 2, 3}*.
- `Sort[{"pears", "apples", "figs"}]` gives you the list elements in alphabetical order.
- `Sort[{(3, 2), (2, 5), (4, 1)}]` gives you *{(2, 5), (3, 2), (4, 1)}*.

Take Command

Take[<List>, <Start Position m>, <End Position n>]

Returns a list containing the elements from position *m* to *n* of the initial list.

Take[<Text>, <Start Position m>, <End Position n>]

Returns a text containing the elements from position *m* to *n* of the initial text.

CAS Syntax

Take[<List>, <Start Position m>, <End Position n>]

Returns a list containing the elements from position *m* to *n* of the initial list.

Example:

`Take[{1, 2, a, 4, 5}, 2, 4]` yields *{2, a, 4}*.

TiedRank Command

TiedRank[<List L>]

Returns a list, whose *i*-th element is the rank of *i*-th element of L (rank of element is its position in Sort[L]). If there are more equal elements in L which occupy positions from *k* to *l* in Sort[L], the mean of the ranks from *k* to *l* are associated with these elements.

Example: `TiedRank[{4, 1, 2, 3, 4, 2}]` returns {5.5, 1, 2.5, 4, 5.5, 2.5} .

Example: `TiedRank[{3, 2, 2, 1}]` returns {4, 2.5, 2.5, 1}.

Note: Also see OrdinalRank Command

Union Command

Union[List 1, List 2]

Joins the two lists and removes elements that appear multiple times.

Union[Polygon poly1, Polygon poly2]

Finds the union of the two polygons. Works only for where the polygons are not self-intersecting, and where the union is a single polygon.

Unique Command

Unique[<List L>]

Returns list of elements of list L in ascending order, repetitive elements are included only once. Works for both a list of numbers and a list of text. See also the Frequency command.

Example:

- `Unique[{1, 2, 4, 1, 4}]` yields *{1, 2, 4}*.
- `Unique[{"a", "b", "Hello", "Hello"}]` yields *{"Hello", "a", "b"}*.

CAS Syntax

Unique[<List L>]

Returns a list where each element of L occurs only once.

Example:

`Unique[{1, x, x, 1, a}]` yields *{1, x, a}*.

Zip Command

Zip[<Expression>, <Var1>, <List1>, <Var2>, <List2>, ...]

Creates list of objects obtained by substitution of variables in the expression by elements of corresponding lists. Length of the resulting list is minimum of lengths of output lists.

Example: Let P,Q,R,S,T be some points. `Zip[Midpoint[A,B],A,{P,Q},B,{R,S}]` returns a list containing midpoints of segments *PR* and *QS*.

Note: In each list the elements must be of the same type.

Vector & Matrix Commands

- ApplyMatrix
- CurvatureVector
- Determinant
- Identity
- Invert
- PerpendicularVector
- ReducedRowEchelonForm
- Transpose
- UnitPerpendicularVector
- UnitVector
- Vector

ApplyMatrix Command

ApplyMatrix[<Matrix M>, <Geometric Object O>]

Transforms the object so that point *P* of *O* is mapped to

- point *M*P* in case M is a 2x2 matrix or
- point *project(M*(x(P), y(P), 1))* where *project* is a projection mapping point *(x,y,z)* to *(x/z, y/z)* in case of 3x3 matrix.

ApplyMatrix[<Matrix M>, <Image I>]

Applies the same transformation as above to image I.

CurvatureVector Command

CurvatureVector[Point, Function]

Yields the curvature vector of the function in the given point.

CurvatureVector[Point, Curve]

Yields the curvature vector of the curve in the given point.

Determinant Command

Determinant[<Matrix>]

Gives the determinant of the given matrix.

Example:

`Determinant[{{1, 2}, {3, 4}}]` yields *a* = *-2*.

CAS Syntax

Determinant[<Matrix>]

Gives the determinant of the given matrix. If matrix contains undefined variables, it yields a formula for the determinant.

Example:

`Determinant[{{1, a}, {b, 4}}]` yields *-a b + 4*.

Identity Command

Identity[Number n]

Returns the identity matrix with the given order (\(n\times n\)).

Example: `Identity[3]` gives you the matrix *\(\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\)*.

Note: If *A* is a square matrix of order *n*, `A^0` produces the same matrix as `Identity[n]`.
This command is not dynamic. ie `Identity[a]` will not change when **a** is changed.

CAS Syntax

Identity[Number]

Returns the identity matrix with the given order.

Invert Command

Invert[Matrix]

Inverts the given matrix.

Example:

`Invert[{{1, 2}, {3, 4}}]` gives you the inverse matrix \(\begin{pmatrix} -2 & 1\\ 1.5 & -0.5 \end{pmatrix} \).

CAS Syntax

Invert[Matrix]

Inverts the given matrix.

Example:

`Invert[{{a, b}, {c, d}}]` gives you the inverse matrix \(\begin{pmatrix} \frac{d}{a* d- b* c} & \frac{-b}{a* d- b* c}\\ \frac{-c}{a* d- b* c}& \frac{a}{ a* d- b* c} \end{pmatrix} \).

PerpendicularVector Command

PerpendicularVector[Line]

Returns the perpendicular vector of the line.

Note: A line with equation *ax + by = c* has the perpendicular vector *(a, b)*.

PerpendicularVector[Segment]

Returns the perpendicular vector of the segment with the same length.

PerpendicularVector[Vector v]

Returns the perpendicular vector of the given vector.

Note: A vector with coordinates *(a, b)* has the perpendicular vector *(-b, a)*.

CAS Syntax

PerpendicularVector[Vector v]

Returns the perpendicular vector of the given vector.

Example:

- `PerpendicularVector[(3, 2)]` yields the vector *{-2, 3}*.
- `PerpendicularVector[(a, b)]` yields the vector *{-b, a}*.

Note:

See also UnitPerpendicularVector Command.

ReducedRowEchelonForm Command

ReducedRowEchelonForm[<Matrix>]

Returns the reduced echelon form of the matrix.

CAS Syntax

ReducedRowEchelonForm[<Matrix>]

Returns the reduced echelon form of the matrix.

Example:

`ReducedRowEchelonForm[{{1, 6, 4}, {2, 8, 9}, {4, 5, 6}}]` yields the matrix \(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\).

Transpose Command

Transpose[<Matrix>]

Transposes the matrix.

Example:

`Transpose[{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}]` yields the matrix $\begin{pmatrix}1&4&7\\ 2&5&8\\ 3&6&9\end{pmatrix}$.

CAS Syntax

Transpose[<Matrix>]

Transposes the matrix.

Example:

`Transpose[{{a, b}, {c, d}}]` yields the matrix $\begin{pmatrix}a&c\\b&d\end{pmatrix}$.

UnitPerpendicularVector Command

UnitPerpendicularVector[<Line>]

Returns the perpendicular vector with length 1 of the given line.

Example:

`UnitPerpendicularVector[3x + 4y = 5]` yields $\begin{pmatrix}0.6\\0.8\end{pmatrix}$.

UnitPerpendicularVector[<Segment>]

Returns the perpendicular vector with length 1 of the given segment.

UnitPerpendicularVector[<Vector>]

Returns the perpendicular vector with length 1 of the given vector. The vector must be befined first.

Example:

Let *v*=$\begin{pmatrix}3\\4\end{pmatrix}$. `UnitPerpendicularVector[v]` *yields* $\begin{pmatrix}-0.8\\0.6\end{pmatrix}$.

CAS Syntax

In CAS View only one syntax is allowed:

UnitPerpendicularVector[<Vector>]

Yields a perpendicular vector with length 1 of the given vector.

Example:

`UnitPerpendicularVector[{a, b}]` yields {$\frac{-b}{\sqrt{a^{2} + b^{2}}}$, $\frac{a}{\sqrt{a^{2} + b^{2}}}$}.

UnitVector Command

UnitVector[<Vector>]

Yields a vector with length 1, which has the same direction and orientation as the given vector. The vector must be defined first.

Example:

Let *v=\(\begin{pmatrix}3\\4\end{pmatrix}\).* `UnitVector[v]` *yields* \(\begin{pmatrix}0.6\\0.8\end{pmatrix}\).

UnitVector[<Line>]

Yields the direction vector of the given line, with length 1 .

Example:

`UnitVector[3x + 4y = 5]` yields *\(\begin{pmatrix}0.8\\-0.6\end{pmatrix}\).*

UnitVector[<Segment>]

Yields the direction vector of the given segment, with length 1.

CAS Syntax

In CAS View only one syntax is allowed:

UnitVector[<Vector>]

Yields a vector with length 1, which has the same direction and orientation as the given vector.

Example:

`UnitVector[{a, b}]` yields *{\(\frac{a}{\sqrt{a^{2} + b^{2}}}\), \(\frac{b}{\sqrt{a^{2} + b^{2}}}\)}.*

Example:

`UnitVector[{2, 4, 4}]` yields *{\(\frac{1}{3}\), \(\frac{2}{3}\), \(\frac{2}{3}\)}.*

Vector Command

Vector[Point A, Point B]

Creates a vector from point *A* to point *B*.

Vector[Point]

Returns the position vector of the given point.

Note: See also Vector between Two Points tool.

Transformation Commands

- Dilate (Enlarge)
- Reflect
- Rotate
- Shear
- Stretch
- Translate

See also Transformation tools

Dilate Command

Dilate[Geometric Object, Number, Point S]

Dilates the geometric object from point *S* using the given factor.

Note: When dilating polygons, GeoGebra creates also all the transformed vertices and segments.

Dilate[Image, Number, Point S]

Dilates the image from point *S* using the given factor.

Dilate[Geometric Object, Number]

Dilates the geometric object from point of origin using the given factor.

Dilate[Image, Number]

Dilates the image from point of origin using the given factor.

Note: See also Dilate Object from Point by Factor tool.

Reflect Command

Reflect[Geometric Object, Point B]

Reflects the geometric object through point *B*.

Note: When reflecting polygons through a point, the transformed vertices and segments are created as well.

Reflect[Image, Point]

Reflects the image through the given point.

Reflect[Geometric Object, Line]

Reflects the geometric object across the given line.

Note: When reflecting polygons across a line, the transformed vertices and segments are created as well.

Reflect[Image, Line]

Reflects the image across the line.

Reflect[Geometric Object, Circle]

Inverts the geometric object with respect to a circle.

Note: See also Reflect Object about Point, Reflect Object about Line , and Reflect Object about Circle tools.

Rotate Command

Rotate[Geometric Object, Angle]

Rotates the geometric object by the angle around the axis origin.

Rotate[Image, Angle]

Rotates the image by the angle around the axis origin.

Rotate[Geometric Object, Angle, Point P]

Rotates the geometric object by the angle around point P.

Rotate[Image, Angle, Point P]

Rotates the image by the angle around point P.

Note:

Vectors are not rotated around axis origin or point P, but around their initial point. When a polygon, segment or arc are rotated, also images of the vertices / endpoints and sides (in case of polygon) are created. For text rotation use RotateText Command. See also Rotate Object around Point by Angle ToolRotate Object around Point by Angle tool

Shear Command

Shear[<Object>, <Line l>, <Ratio r>]

Shears the object so that points on line *l* stay fixed and points at distance *d* from the line are shifted by $d\cdot r$ in direction of the line (direction of the shift is different for halfplanes with respect to *l*). A sheared plane figure maintains its original area.

Stretch Command

Stretch[<Geometric Object>, <Line>, <Ratio>]

Stretch[<Image>, <Line>, <Ratio>]

The object is stretched **perpendicular** to the line by the given ratio (ie points on the line aren't moved and the distance of other points from the line is multiplied by given ratio.)

Stretch[<Geometric Object>, <Vector>]

Stretch[<Image>, <Vector>]

The object is stretched **parallel** to the given vector by the ratio given by the **magnitude** of the vector (ie points on the line perpendicular to the vector (through its startpoint) stay on their place and distance of other points from the line is multiplied by given ratio.)

Translate Command

Translate[Geometric Object, Vector]

Translates the geometric object by the vector.

Note: When translating a polygon, the transformed new vertices and segments are created as well.

Translate[Image, Vector]

Translates the image by the vector.

Translate[Vector, Point]

Translates the vector *v* to point.

Note: See also  Translate Object by Vector tool.

Chart Commands

- BarChart
- BoxPlot
- DotPlot
- FrequencyPolygon
- Histogram
- HistogramRight
- NormalQuantilePlot
- ResidualPlot
- StemPlot

BarChart Command

BarChart[Start Value, End Value, List of Heights]

Creates a bar chart over the given interval: the number of bars is determined by the length of the list, whose elements are the heights of the bars.

Example: `BarChart[10, 20, {1,2,3,4,5} ]` gives you a bar chart with five bars of specified height in the interval [*10, 20*].

BarChart[Start Value a, End Value b, Expression, Variable k, From Number c, To Number d]

Creates a bar chart over the given interval [*a, b*], that calculates the bars' heights using the expression whose variable *k* varies from number *c* to number *d*.

Example: If $p = 0.1$, $q = 0.9$, and $n = 10$ are numbers, then `BarChart[ -0.5, n + 0.5, BinomialCoefficient[n,k]*p^k*q^(n-k), k, 0, n ]` gives you a bar chart in the interval [*-0.5, n+0.5*]. The heights of the bars depend on the probabilities calculated using the given expression.

BarChart[Start Value a, End Value b, Expression, Variable k, From Number c, To Number d, Step Width s]

Creates a bar chart over the given interval [*a, b*], the bars' heights are calculated using the given expression in which the variable *k* varies from number *c* to number *d* using step width *s*.

BarChart[List of Raw Data, Width of Bars]

Creates a bar chart using the given raw data; the bars have the given width.

Example: `BarChart[ {1,1,1,2,2,2,2,2,3,3,3,5,5,5,5}, 1]`

BarChart[List of Data, List of Frequencies]

Creates a bar chart using the list of data with corresponding frequencies.

Note: The list of data must contain numbers in arithmetic progression.

Example:

- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}]`
- `BarChart[{5, 6, 7, 8, 9}, {1, 0, 12, 43, 3}]`
- `BarChart[{0.3, 0.4, 0.5, 0.6}, {12, 33, 13, 4}]`

BarChart[List of Data , List of Frequencies, Width of Bars w]

Creates a bar chart using the list of data and corresponding frequencies; the bars have width *w*.

Note: The list of data must contain numbers in arithmetic progression.

Example:

- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}, 0.5]` leaves gaps between bars.
- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}, 0]` produces a line graph.

BoxPlot Command

BoxPlot[yOffset, yScale, List of Raw Data]

Creates a box plot using the given raw data and whose vertical position in the coordinate system is controlled by variable *yOffset* and whose height is influenced by factor *yScale*.

Example: `BoxPlot[0, 1, {2,2,3,4,5,5,6,7,7,8,8,8,9}]`

BoxPlot[yOffset, yScale, Start Value a, Q1, Median, Q3, End Value b]

Creates a box plot for the given statistical data in interval [*a, b*].

DotPlot Command

DotPlot[<List of Raw Data>]

Returns dot plot for given list of numbers as a list of points. For each number *a* which occurs in list *k* times, the returned list contains points *(a,1),(a,2),...,(a,k)*.

FrequencyPolygon Command

Note: Frequency polygon is a line graph drawn by joining all the midpoints of the top of the bars of a histogram. Therefore usage of this command is the same as usage of Histogram Command.

FrequencyPolygon[<List of Class Boundaries>, <List of Heights>]

Creates a frequency polygon with vertices in given heights. The class boundaries determine the x-coordinate of each vertex.

Note: For examples see Histogram Command.

FrequencyPolygon[<List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)]

Creates a frequency polygon using the raw data. The class boundaries determine the x-coordinates of vertices and are used to determine how many data elements lie in each class. The y-coordinate of a vertex is determined as follows

- If *Use Density = true*, height = (Density Scale Factor) * (class frequency) / (class width)
- If *Use Density = false*, height = class frequency

By default, Use Density = true and Density Scale Factor = 1.

FrequencyPolygon[<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density> , <Density Scale Factor> (optional)]

If Cumulative is true this creates a frequency polygon where each vertex y-coordinate equals the frequency of the class plus the sum of all previous frequencies.

Histogram Command

Histogram[<List of Class Boundaries>, <List of Heights>]

> Creates a histogram with bars of the given heights. The class boundaries determine the width and position of each bar of the histogram.

Example: `Histogram[{0, 1, 2, 3, 4, 5}, {2, 6, 8, 3, 1}]` creates a histogram with 5 bars of the given heights. The first bar is positioned at the interval [*0, 1*], the second bar is positioned at the interval [*1, 2*], and so on.

Histogram[<List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density>, <Density Scale Factor>(optional)]

> Creates a histogram using the raw data. The class boundaries determine the width and position of each bar of the histogram and are used to determine how many data elements lie in each class. Bar height is determined as follows

- If *Use Density = true*, height = (Density Scale Factor) * (class frequency) / (class width)
- If *Use Density = false*, height = class frequency

> By default, Use Density = true and Density Scale Factor = 1. This creates a histogram with total area = n, the number of data values.

Example: (*Default Histogram*)Histogram[{10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true] creates a histogram with 3 bars, with the heights 0.5 (first bar), 0.2 (second bar), and 0.1 (third bar). This histogram has total area = 0.5*10 + 0.2*10 + 0.1*10 = 8.

Example: (*Count Histogram*)Histogram[{10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, false] creates a histogram with 3 bars, with the heights 5 (first bar), 2 (second bar), and 1 (third bar). This histogram does not use density scaling and gives bar heights that equal the count of values in each class.

Example: (*Relative Frequency Histogram*)Histogram[{10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true, 10/ 8] creates a histogram with 3 bars, with the heights 0.625 (first bar), .25 (second bar), and .125 (third bar). This histogram uses density scaling to give bar heights that equal the proportion of values in each class. If n is the number of data values, and the classes have constant width w, then Density Scale Factor = w/n creates a relative histogram.

Example: (*Normalized Histogram*)Histogram[{10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true, 1/8] creates a histogram with 3 bars, with the heights .0625 (first bar), .025 (second bar), and .0125 (third bar). This histogram has total area = .0625*10 + .025*10 + .0125*10 = 1. If n is the number of data values, then Density Scale Factor = 1/n creates a normalized histogram with total area = 1. This is useful for fitting a histogram with a density curve.

Histogram[<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Boolean Use Density> , <Density Scale Factor> (optional)]

> If Cumulative is true this creates a histogram where each bar height equals the frequency of the class plus the sum of all previous frequencies.

Example: :`Histogram[true, {10, 20, 30, 40}, {10, 11, 11, 12, 18, 20, 25, 40}, true]` creates a histogram with 3 bars, with the heights 0.5 (first bar), 0.7 (second bar), and 0.8 (third bar).

HistogramRight Command

HistogramRight[<List of Class Boundaries B>, <List of Heights H>]

Same as `Histogram[B,H]`

HistogramRight[<List of Class Boundaries B>, <List of Raw Data D>, <Boolean Use Density> , <Density Scale Factor f> (optional)]

Same as `Histogram[B,D,Use Density, f]`, except that if a datum is equal to the right border of a class, it is counted in this class and not in the next one.

HistogramRight[<Boolean Cumulative>, <List of Class Boundaries B>, <List of Raw Data D>, <Boolean Use Density> , <Density Scale Factor f> (optional)]

Same as `Histogram[Cumulative, B,D,Use Density, f]`, except that if a datum is equal to the right border of a class, it is counted in this class and not in the next one.

NormalQuantilePlot Command

NormalQuantilePlot[<List of Raw Data>]

Creates a normal quantile plot from the given list of data and draws a line through the points showing the ideal plot for exactly normal data. Points are formed by plotting data values on the x axis against their expected normal score (Z-score) on the y axis.

ResidualPlot Command

ResidualPlot[<List of Points L>, <Function f>]

Returns a list of points whose x-coordinates are equal to x-coordinates of elements of L and y-coordinates are residuals with respect to *f*. If *i*-th element of L is a point *(a,b)* then *i*-th element of result is *(a,b-f(a))*.

StemPlot Command

StemPlot[<List>]

Returns a stem plot of the given list of numbers. Outliers are removed from the plot and listed separately.

An outlier is defined as a value outside the interval [Q1 - 1.5 (Q3 - Q1) , Q3 + 1.5 (Q3 - Q1)].

StemPlot[<List>, <Adjustment -1|0|1>]

Returns a stem plot of the given list of numbers.

If *Adjustment* = -1 the default stem unit is divided by 10

If *Adjustment* = 0 nothing is changed

If *Adjustment* = 1 the default stem unit is multiplied by 10

Statistics Commands

- ANOVA
- Classes
- CorrelationCoefficient
- Covariance
- Fit
- FitExp
- FitGrowth
- FitLine
- FitLineX
- FitLog
- FitLogistic
- FitPoly
- FitPow
- FitSin
- Frequency
- FrequencyTable
- GeometricMean
- HarmonicMean
- Mean
- MeanX
- MeanY
- Median
- Mode
- Percentile
- Q1
- Q3
- RootMeanSquare
- RSquare
- Sample
- SampleSD
- SampleSDX

- SampleSDY
- SampleVariance
- SD
- SDX
- SDY
- Shuffle
- SigmaXX
- SigmaXY
- SigmaYY
- Spearman
- Sum
- SumSquaredErrors
- Sxx
- Sxy
- Syy
- TMeanEstimate
- TMean2Estimate
- TTest
- TTest2
- TTestPaired
- Variance

See also Probability Calculator Tool.

ANOVA Command

ANOVA[<List>, <List>, ...]

Performs a one-way ANOVA test on the given lists of numbers.

Results are returned in list form as {P value, F test statistic}.

Classes Command

Classes[<List of Data L>, <Start S>, <Width of Classes w>]

Gives a list of class boundaries. First boundary is equal to *S*, last boundary is at least equal the maximum of *L*, distances between consequent boundaries are equal to *w*.

Classes[<List of Data L>, <Number of Classes>]

Gives a list of class boundaries. First boundary is equal to minimum of *L*, last boundary to maximum of *L*, distances between consequent boundaries are equal.

Covariance Command

Covariance[<List 1 of Numbers>, <List 2 of Numbers>]

Calculates the covariance using the elements of both lists.

Covariance[<List of Points>]

Calculates the covariance using the *x*- and *y*-coordinates of the points.

CAS Syntax

Covariance[<List 1 of Numbers>, <List 2 of Numbers>]

Calculates the covariance using the elements of both lists.

Example:

`Covariance[{1, 2, 3}, {1, 3, 7}]` yields *2*, the covariance of *{1, 2, 3}* and *{1, 3, 7}*.

Covariance[<List of Points>]

Calculates the covariance using the *x*- and *y*-coordinates of the points.

Example:

`Covariance[{(1, 1), (2, 3), (3, 7)}]` yields *2*, the covariance of *{1, 2, 3}* and *{1, 3, 7}*.

Fit Command

Fit[<List of Points>,<List of Functions>]

Calculates a linear combination of functions to the points in the list.

Example: With *L={A,B,C,...}, f(x)=1, g(x)=x, h(x)=e^x, F={f,g,h}* the command `Fit[L,F]` calculates a function of the form *a + b x + c e^x* to the points in the list.

Fit[<List of points>, <Function>]

Calculates a minimum squared error function to the points in the list. The Function must depend on one or more sliders, that are taken as start values of parameters to be optimized. The non-linear iteration might not converge, but adjusting the sliders to a better starting point might help.

FitExp Command

FitExp[<List of Points>]

Calculates the exponential regression curve.

Example:

`FitExp[{(0, 1), (2, 4)}]` gives $\square^{0.69x}$.

Note:

- Euler's number *e* can be obtained by pressing .
- See also FitLog Command, FitPoly Command, FitPow Command and FitSin Command.

CAS Syntax

FitExp[<List of Points>]

Calculates the exponential regression curve.

FitGrowth Command

FitGrowth[<List of Points>]

Calculates a function of the form $a*b^x$ to the points in the list. (Just like FitExp[<List of Points>], for users who do not know the meaning of exponential growth).

FitLineX Command

FitLineX[List of Points]

Calculates the *x* on *y* regression line of the points.

Note: See also  Best Fit Line tool and FitLine Command.

FitLine Command

FitLine[List of Points]

Calculates the *y* on *x* regression line of the points.

Note: See also  Best Fit Line tool and FitLineX Command.

FitLog Command

FitLog[<List of Points>]

Calculates the logarithmic regression curve.

Example:

`FitLog[{(e,1), (e^2, 4)}]` gives *3 ln(x) - 2*.

Note:

- Euler's number *e* can be obtained by pressing .
- See also FitExp Command, FitPoly Command, FitPow Command and FitSin Command.

CAS Syntax

FitLog[<List of Points>]

Calculates the logarithmic regression curve.

FitLogistic Command

FitLogistic[List of Points]

Calculates the regression curve in the form a/(1+b e^(-kx)).

Note: The first and last data points should be fairly close to the curve. The list should have at least 3 points, preferably more.

FitPoly Command

FitPoly[<List of Points>, <Degree n of Polynomial>]

Calculates the regression polynomial of degree *n*.

Example:

`FitPoly[{(-1, -1), (0, 1), (1, 1), (2, 5)}, 3]` gives $x^3 - x^2 + 1$.

Note:

See also FitExp Command, FitLog Command, FitPow Command and FitSin Command.

CAS Syntax

FitPoly[<List of Points>, <Degree n of Polynomial>]

Calculates the regression polynomial of degree *n*.

FitPow Command

FitPow[<List of Points>]

Calculates the regression curve in the form $a\ x^b$.

Example:

`FitPow[{(1, 1), (3, 2), (7, 4)}]` creates the regression curve $0.97\ x^{0.71}$.

Note:

- All points used need to be in the first quadrant of the coordinate system.
- See also FitExp Command, FitLog Command, FitPoly Command, and FitSin Command.

CAS Syntax

FitPow[<List of Points>]

Calculates the regression curve in the form $a\ x^b$.

FitSin Command

FitSin[<List of Points>]

Calculates the regression curve in the form *a + b sin(cx + d)*.

Example:

`FitSin[{(1, 1), (2, 2), (3, 1), (4, 0), (5, 1), (6, 2)}]` gives *1 + sin(1.5708x - 1.5708)*.

Note:

- The list should have at least 4 points, preferably more. The list should cover at least two extremal points. The first two local extremal points should not be too different from the absolute extremal points of the curve.
- See also FitExp Command, FitLog Command, FitPoly Command and FitPow Command.

CAS Syntax

FitSin[<List of Points>]

Calculates the regression curve in the form *a + b sin(cx + d)*.

Frequency Command

Frequency[<List of Raw Data>]

Returns a list with a count of the occurrences of each unique value in the given list of data. This input list can be numbers or text. The list is sorted in ascending order of the unique values. To get a list of the corresponding unique values use the command Unique[<List of Raw Data>]

Example: Enter `list1 = { "a", "a", "x", "x", "x", "b" }.Frequency[list1]` returns the list { 2, 1, 3 }. `Unique[list1]` returns the list { "a", "b", "x" }.

Frequency[<Cumulative>, <List of Raw Data>]

If Cumulative = false, returns the same list as Frequency[<List of Raw Data>]

If Cumulative = true, returns a list of cumulative frequencies for Frequency[<List of Raw Data>]

Example: Enter `list1 = { 0, 0, 0, 1 , 1, 2 }.Frequency[true, list1]` returns the list { 3, 5, 6 }. `Frequency[false, list1]` returns the list { 3, 2, 1}. `Unique[list1]` returns the list { 0, 1, 2 }

Frequency[<List of Class Boundaries>, <List of Raw Data>]

Returns a list of the counts of values from the given data list that lie within the intervals formed by the given class boundaries. All intervals except the highest interval are of the form [a, b). The highest interval has the form [a, b].

Example: `Frequency[{1,2,3}, {1,1,2,3}]` returns the list { 2, 2 }.

Frequency[<Cumulative>,<List of Class Boundaries>,<List of Raw Data>]

If Cumulative = false, returns the same list as Frequency[<List of Class Boundaries>, <List of Raw Data>]

If Cumulative = true, returns a list of cumulative frequencies for Frequency[<List of Class Boundaries>, <List of Raw Data>]

Frequency[<List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a list of frequencies for the corresponding Histogram Command.

Frequency[<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a list of frequencies for the corresponding Histogram Command.

FrequencyTable Command

FrequencyTable[<List of Raw Data L>]

Returns a table (as text) whose first column contains sorted list of unique elements of *L* and second column contains the count of the occurrences of value in the first column. List *L* can be numbers or text.

FrequencyTable[<Boolean Cumulative C>, <List of Raw Data L>]

If *Cumulative* = *false*, returns the same table as Frequency[<L>]

If *Cumulative* = *true*, returns a table whose first column is the same as in FrequencyTable[L] and the second contains cumulative frequencies of values in the first column.

FrequencyTable[<List of Class Boundaries C>, <List of Raw Data L>]

Returns a table (as text) whose first column contains intervals (classes) and second column contains the count of numbers in *L* which belong to the interval in the first column. All intervals except the highest interval are of the form [a, b). The highest interval has the form [a, b].

FrequencyTable[<Boolean Cumulative>,<List of Class Boundaries C>,<List of Raw Data L>]

If *Cumulative* = *false*, returns the same table as FrequencyTable[<List of Class Boundaries>, <List of Raw Data>]

If *Cumulative* = *true*, returns a table whose first column is the same as in FrequencyTable[L] and the second contains cumulative frequencies of values in the first column.

FrequencyTable[<List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a table (as text) whose first column contains intervals (classes) and second contains frequencies for the corresponding Histogram Command.

FrequencyTable[<Boolean Cumulative>, <List of Class Boundaries>, <List of Raw Data>, <Use Density> , <Density Scale Factor> (optional)]

Returns a table (as text) whose first column contains intervals (classes) and second contains frequencies for the corresponding Histogram Command.

Note: This command is similar to Frequency Command and Histogram Command. Articles about these commands contain some related examples.

GeometricMean Command

GeometricMean[List of Numbers]

Returns the geometric mean of given list of numbers.

HarmonicMean Command

HarmonicMean[List of Numbers]

Returns the harmonic mean of given list of numbers.

Mean Command

Mean[<List of Numbers>]

Calculates the mean of the list elements.

MeanX[<List of Points>]

Calculates the mean of the *x*-coordinates of the points in the list.

MeanY[<List of Points>]

Calculates the mean of the *y*-coordinates of the points in the list.

CAS Syntax

In CAS View only the following syntax is supported:

Mean[<List of Numbers>]

Calculates the arithmetic mean of the list elements.

Example:

`Mean[{1, 2, 3, 5, 44}]` yields *11*.

Note:

See also SD Command.

MeanX Command

MeanX[List of Points]

Calculates the mean of the *x*-coordinates of the points in the list.

MeanY Command

MeanY[List of Points]

Calculates the mean of the *y*-coordinates of the points in the list.

Median Command

Median[<List of Numbers>]

Determines the median of the list elements.

Example:

- `Median[{1, 2, 3}]` yields *2* and
- `Median[{1, 1, 8, 8}]` yields *4,5*.

Note:

If the length of the given list is even, the arithmetic mean of the two center elements is returned.

CAS Syntax

Median[<List of Numbers>]

Determines the median of the list elements.

Example:

- `Median[{1, 2, 3}]` yields *2* and
- `Median[{1, 1, 8, 8}]` yields *4,5*.

Note:

If the length of the given list is even, the arithmetic mean of the two center elements is returned.

Mode Command

Mode[List of Numbers]

Determines the mode(s) of the list elements.

Example:

Mode[{1,2,3,4}] returns an empty list {}.Mode[{1,1,1,2,3,4}] returns the list {1} .Mode[{1,1,2,2,3,3,4}] returns the list {1, 2, 3}.

CorrelationCoefficient Command

CorrelationCoefficient[List of x-Coordinates, List of y-Coordinates]

Calculates the product moment correlation coefficient using the given *x*- and *y*-coordinates.

CorrelationCoefficient[List of Points]

Calculates the product moment correlation coefficient using the coordinates of the given points.

Percentile Command

Percentile[<List of Numbers>, <Percent>]

Let P equal the given *Percent*.

Returns the value that cuts off the first P percent of the number list when the list is sorted in ascending order. *Percent* must be a number in the interval 0 < P ≤ 1.

Note: The commands Quartile and Percentile use different rules and do not always return matching results. For example, Q1[{1,2,3,4}] = 2 Percentile[{1,2,3,4}, 0.25] = 1

Q1 Command

Q1[List of Numbers]

Determines the lower quartile of the list elements.

Q3 Command

Q3[List of Numbers]

Determines the upper quartile of the list elements.

RSquare Command

RSquare[<List of Points>,<Function>]

Calculates the coefficient of determination $R^2 = 1-SSE/Syy$, between the y-values of the points in the list and the function values of the x-values in the list.

RootMeanSquare Command

RootMeanSquare[<List of Numbers>]

Returns the root mean square of given list of numbers.

SD Command

SD[<List of Numbers>]

Calculates the standard deviation of the numbers in the list.

Example:

`SD[{1, 2, 3, 4, 5}]` yields *1,41*.

CAS Syntax

SD[<List of Numbers>]

Calculates the standard deviation of the numbers in the list.

Example:

`SD[{1, 2, 3, 4, 5}]` yields \(\sqrt{2}\).

Note:

See also Mean Command.

SDX Command

SDX[<List of Points>]

Returns standard deviation of x-coordinates of points from the given list.

SDY Command

SDY[<List of Points>]

Returns standard deviation of y-coordinates of points from the given list.

Sxx Command

Sxx[List of Numbers]

Calculates the statistic $\sum x^2 - \frac{(\sum x)^2}{n}$.

Sxx[List of Points]

Calculates the statistic $\sum x^2 - \frac{(\sum x)^2}{n}$ using the *x*-coordinates of the given points.

Sxy Command

Sxy[List of Numbers, List of Numbers]

Calculates the statistic $\sum xy - \frac{(\sum x) (\sum y)}{n}$.

Sxy[List of Points]

Calculates the statistic $\sum xy - \frac{(\sum x) (\sum y)}{n}$ using the coordinates of the given points.

Syy Command

Syy[List of Points]

Calculates the statistic $\sum y^2 -\frac{ (\sum y)^2}{n}$ using the *y*-coordinates of the given points.

Sample Command

Sample[<List L>, <Size n>]

Returns list of *n* randomly chosen elements of *L*; elements can be chosen several times.

Example:

`Sample[{1, 2, 3, 4, 5}, 5]` yields for example *{1, 2, 1, 5, 4}*.

Sample[<List L>, <Size n>, <Boolean for replacement>]

Returns list of *n* randomly chosen elements of *L*. Elements can be chosen several times if and only if the last parameter is true.

Example:

`Sample[{1, 2, 3, 4, 5}, 5, true]` yields for example *{2, 3, 3, 4, 5}*.

CAS Syntax

Sample[<List L>, <Size n>]

Returns list of *n* randomly chosen elements of *L*; elements can be chosen several times.

Example:

`Sample[{-5, 2, a, 7, c}, 3]` yields for example *{a, 7, -5}*.

Sample[<List L>, <Size n>, <Boolean for replacement>]

Returns list of *n* randomly chosen elements of *L*. Elements can be chosen several times if and only if the last parameter is true.

Example:

The list can include lists as well: Let *List1* be *{1, 2, 3}*.
`Sample[{List1, 4, 5, 6, 7, 8}, 3, false]` yields for example *{6, {1, 2, 3}, 4}*.

SampleSD Command

SampleSD[<List of Numbers>]

Returns sample standard deviation of given list of numbers.

Example:

`SampleSD[{1, 2, 3}]` yields *1*.

CAS Syntax

SampleSD[<List of Numbers>]

Returns sample standard deviation of given list of numbers. If the list contains undefined variables, it yields a formula for the sample standard deviation.

Example:

`SampleSD[{1, 2, a}]` yields $\frac{\sqrt{a^2-3a+3}}{\sqrt{3}}$.

SampleSDX Command

SampleSDX[<List of Points>]

Returns sample standard deviation of x-coordinates of points from the given list.

SampleSDY Command

SampleSDY[<List of Points>]

Returns sample standard deviation of y-coordinates of points from the given list.

SampleVariance Command

SampleVariance[<List of Numbers>]

Returns the sample variance of given list of numbers.

CAS Syntax

SampleVariance[<List of Numbers>]

Returns the sample variance of given list of numbers. If the list contains undefined variables, it yields a formula for the sample variance.

Example:

`SampleVariance[{x, y, z}]` yields $\frac{x^2 - x y - x z + y^2 - y z + z^2}{3}$.

Shuffle Command

Shuffle[<List L>]

Returns list with same elements as *L*, but in random order.

Note: You can recompute the list via *Recompute all objects* in View Menu (or pressing).

See also RandomElement Command and RandomBetween Command.

CAS Syntax

Shuffle[<List L>]

Returns list with same elements as *L*, but in random order.

Example:

`Shuffle[{3, 5, 1, 7, 3}]` yields for example *{5, 1, 3, 3, 7}*.

SigmaXX Command

SigmaXX[List of Numbers]

Calculates the sum of squares of the given numbers.

Example: In order to work out the variance of a list you may use `SigmaXX[list]/Length[list] - Mean[list]^2`.

SigmaXX[List of Points]

Calculates the sum of squares of the *x*-coordinates of the given points.

SigmaXY Command

SigmaXY[List of x-Coordinates, List of y-Coordinates]

Calculates the sum of the products of the *x*- and *y*-coordinates.

SigmaXY[List of Points]

Calculates the sum of the products of the *x*- and *y*-coordinates.

Example: You can work out the covariance of a list of points using `SigmaXY[list]/Length[list] - MeanX[list] * MeanY[list]`.

SigmaYY Command

SigmaYY[List of Points]

Calculates the sum of squares of *y*-coordinates of the given points.

Spearman Command

Spearman[<List of Points L>]

Returns Spearman's rank correlation coefficient of x-coordinates and y-coordinates of points in *L*.

Spearman[<List of Numbers A>, <List of Numbers B>]

Returns Spearman's rank correlation coefficient of lists *A* and *B*.

Sum Command

Sum[<List>]

Calculates the sum of all list elements.

Example:

- `Sum[{1, 2, 3}]` yields the number *a = 6*.
- `Sum[{x^2, x^3}]` yields $f(x) = x^2 + x^3$.
- `Sum[Sequence[i, i, 1, 100]]` yields the number *a = 5050*.
- `Sum[{(1, 2), (2, 3)}]` yields the point *A = (3, 5)*.
- `Sum[{(1, 2), 3}]` yields the point *B = (4, 2)*.
- `Sum[{"a", "b", "c"}]` yields the text *"abc"*.

Note: This command works for numbers, points, vectors, text, and functions.

Sum[<List>, <Number n of Elements>]

Calculates the sum of the first *n* list elements.

Example: `Sum[{1, 2, 3, 4, 5, 6}, 4]` yields the number *a = 10*.

Note: This command works for numbers, points, vectors, text, and functions.

CAS Syntax

Following syntax works only in CAS view:

Sum[<List>]

Calculates the sum of all list elements.

Example:

- `Sum[{1, 2, 3}]` yields *6*.
- `Sum[{a, b, c}]` yields $a + b + c$.

Sum[<Expression f(t)>, <Variable t>, <Start Value s>, <End Value e>]

Computes sum $\sum_{t=s}^{e}f(t)$. End value might be infinity.

Example:

- `Sum[i^2, i, 1, 3]` yields *14*.
- `Sum[r^i, i, 0, n]` yields $\frac{r^{n+1} - 1}{r - 1}$.
- `Sum[(1/3)^i, i, 0, Infinity]` yields $\frac{3}{2}$.

SumSquaredErrors Command

SumSquaredErrors[<List of Points>, <Function>]

Calculates the sum of squared errors, SSE, between the y-values of the points in the list and the function values of the x-values in the list.

Example: If we have a list of points: L={A,B,C,D,E} and have calculated for example: f(x)=RegPoly[L,1] and g(x)=RegPoly[L,2], then it is possible to decide which of the two functions offers the best fit, in the sense of the least sum of squared errors (Gauss), by comparing: sse_f=SumSquaredErrors[L,f] and sse_g=SumSquaredErrors[L,g].

TMean2Estimate Command

TMean2Estimate[<List of Sample Data 1>,<List of Sample Data 2>,<Level>,<Boolean Pooled>]

Calculates a T confidence interval estimate of the difference between two population means using the given sample data sets and confidence *Level*.

If *Pooled* = true, then population variances are assumed equal and sample standard deviations are combined in calculation.

If *Pooled* = false, then population variances are not assumed equal and sample standard deviations are not combined.

Results are returned in list form as {lower confidence limit, upper confidence limit}.

TMean2Estimate[<Sample Mean 1 >,<Sample Standard Deviation 1 >, <Sample Size 1>, <Sample Mean 2 >,<Sample Standard Deviation 2 >, <Sample Size 2>, <Level>,<Boolean Pooled>]

Calculates a T confidence interval estimate of the difference between two population means using the given sample statistics and confidence *Level*. *Pooled* is defined as above. Results are returned in list form as {lower confidence limit, upper confidence limit}.

TMeanEstimate Command

TMeanEstimate[<List of Sample Data>,<Level>]

Calculates a T confidence interval estimate of a population mean using the given sample data and confidence *Level*. Results are returned in list form as {lower confidence limit, upper confidence limit}.

TMeanEstimate[<Sample Mean>,<Sample Standard Deviation>,<Sample Size>,<Level>]

Calculates a T confidence interval estimate of a population mean using the given sample statistics and confidence level. Results are returned in list form as {lower confidence limit, upper confidence limit}.

TTest Command

TTest[<List of Sample Data>,<Hypothesized Mean>,<Tail>]

Performs a one sample T test of a population mean using the given list of sample data. *Hypothesized Mean* is the population mean assumed in the null hypothesis. *Tail* has possible values "<", ">" , "≠". These specify the alternative hypothesis as follows.

"<" = population mean < *Hypothesized Mean*

">" = population mean > *Hypothesized Mean*

"≠" = population mean ≠ *Hypothesized Mean*

Results are returned in list form as {P value, T test statistic}.

TTest[<Sample Mean>,<Sample Standard Deviation>,<Sample Size>,<Hypothesized Mean>,<Tail>]

Performs a one sample T test of a population mean using the given sample statistics. *Hypothesized Mean* and *Tail* are defined as above. Results are returned in list form as {P value, T test statistic}.

TTest2 Command

TTest2[<List of Sample Data 1>,<List of Sample Data 2>,<Tail>,<Boolean Pooled>]

Performs a T test of the difference between two population means using the given list of sample data. Tail has possible values "<", ">" , "≠" that determine the following alternative hypotheses:

"<" = difference in population means < 0

">" = difference in population means > 0

"≠" = difference in population means ≠ 0

If *Pooled* = true, then population variances are assumed equal and sample standard deviations are combined in calculation.

If *Pooled* = false, then population variances are not assumed equal and sample standard deviations are not combined.

Results are returned in list form as {P value, T test statistic}.

TTest2[<Sample Mean 1 >,<Sample Standard Deviation 1>, <Sample Size 1>, <Sample Mean 2 >,<Sample Standard Deviation 2 >, <Sample Size 2>, <Tail>,<Boolean Pooled>]

Performs a T test of the difference between two population means using the given sample statistics. *Tail* and *Pooled* are defined as above.

TTestPaired Command

TTestPaired[<List of Sample Data 1>,<List of Sample Data 2>,<Tail>]

Performs a paired T test using the given lists of paired sample data. *Tail* has possible values "<", ">" , "≠" that determine the following alternative hypotheses:

"<" = $\mu < 0$

">" = $\mu > 0$

"≠" = $\mu \neq 0$

(μ is the mean paired difference of the population)

Results are returned in list form as {P value, T test statistic}.

Variance Command

Variance[<List of Numbers>]

Calculates the variance of list elements.

Example:

`Variance[{1, 2, 3}]` yields *0.67*.

CAS Syntax

Variance[<List of Numbers>]

Calculates the variance of list elements. If the list contains undefined variables, it yields a formula for the variance.

Example:

`Variance[{1, 2, a}]` yields $\frac{2 a^{2} - 6 a + 6}{9}$.

Probability Commands

- Bernoulli
- BinomialDist
- BinomialCoefficient
- Cauchy
- ChiSquared
- Erlang
- Exponential
- FDistribution
- Gamma
- HyperGeometric
- InverseBinomial
- InverseCauchy
- InverseChiSquared
- InverseExponential
- InverseFDistribution
- InverseGamma
- InverseHyperGeometric
- InverseNormal
- InversePascal
- InversePoisson
- InverseTDistribution
- InverseWeibull
- InverseZipf
- Logistic
- LogNormal
- Normal
- Pascal
- Poisson

- RandomBetween
- RandomBinomial
- RandomNormal
- RandomPoisson
- RandomUniform
- TDistribution
- Triangular
- Uniform
- Weibull
- Zipf

Bernoulli Command

Bernoulli[<Probability p>, <Boolean Cumulative>]

For *Cumulative* = *false* returns the bar graph of Bernoulli distribution where probability of success is equal to *p*.

For *Cumulative* = *true* returns the bar graph of cumulative Bernoulli distribution.

BinomialCoefficient Command

BinomialCoefficient[<Number n>, <Number r>]

Calculates the binomial coefficient \(n \choose r \).

Example:

`BinomialCoefficient[5, 3]` yields *10*.

CAS Syntax

BinomialCoefficient[<Number n>, <Number r>]

Calculates the binomial coefficient \(n \choose r \). If you type undefined variables instead of numbers it yields a formula for the binomial coefficient.

Example:

`BinomialCoefficient[n, 3]` yields \(\frac{n^{3} - 3 n^{2} + 2 n}{6}\).

BinomialDist Command

BinomialDist[<Number of Trials>, <Probability of Success>]

Returns a bar graph of a Binomial distribution.

Parameters:

Number of Trials: number of independent Bernoulli trials

Probability of Success: probability of success in one trial

BinomialDist[<Number of Trials>, <Probability of Success>, <Boolean Cumulative>]

Returns a bar graph of a Binomial distribution when *Cumulative* = false.

Returns a bar graph of a cumulative Binomial distribution when *Cumulative* = true.

First two parameters are same as above.

BinomialDist[<Number of Trials>, <Probability of Success>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Binomial random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First two parameters are same as above.

CAS Specific Syntax

In CAS View only one syntax is allowed:

BinomialDist[<Number of Trials>, <Probability of Success>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Binomial random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

Example:

Assume transfering three packets of data over a faulty line. The chance an arbitrary packet transfered over this line becomes corrupted is \(\frac{1}{10}\), hence the propability of transfering an arbitrary packet successfully is \(\frac{9}{10}\).

- `BinomialDist[3, 0.9, 0, false]` yields *\(\frac{1}{1000}\)*, the probability of none of the three packets being transferred successfully,
- `BinomialDist[3, 0.9, 1, false]` yields *\(\frac{27}{1000}\)*, the probability of exactly one of three packets being transferred successfully,
- `BinomialDist[3, 0.9, 2, false]` yields *\(\frac{243}{1000}\)*, the probability of exactly two of three packets being transferred successfully,
- `BinomialDist[3, 0.9, 3, false]` yields *\(\frac{729}{1000}\)*, the probability of all three packets being transferred successfully.
- `BinomialDist[3, 0.9, 0, true]` yields *\(\frac{1}{1000}\)*, the probability of none of the three packets being transferred successfully,
- `BinomialDist[3, 0.9, 1, true]` yields *\(\frac{7}{250}\)*, the probability of at most one of three packets being transferred successfully,
- `BinomialDist[3, 0.9, 2, true]` yields *\(\frac{271}{1000}\)*, the probability of at most two of three packets being transferred successfully,

- `BinomialDist[3, 0.9, 3, true]` yields *1*, the probability of at most three of three packets being transferred successfully.
- `BinomialDist[3, 0.9, 4, false]` yields *0*, the probability of exactly four of three packets being transferred successfully,
- `BinomialDist[3, 0.9, 4, true]` yields *1*, the probability of at most four of three packets being transferred successfully.

Cauchy Command

Cauchy[<Median m>, <Scale s>, x]

Creates probability density function (pdf) of Cauchy distribution.

Cauchy[<Median m>, <Scale s>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of Cauchy distribution, otherwise creates pdf of Cauchy distribution.

Cauchy[<Median m>, <Scale s>, <Variable Value v>]

Calculates the value of cumulative distribution function of Cauchy distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with Cauchy given by parameters *m, s*.

Note: Returns the probability for a given *x*-coordinate's value (or area under the Cauchy distribution curve to the left of the given *x*-coordinate).

CAS Syntax

In CAS View only following syntax is supported:

Cauchy[<Median m>, <Scale s>, <Variable Value v>]

Calculates the value of cumulative distribution function of Cauchy distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with Cauchy given by parameters *m, s*.

Example:

`Cauchy[1, 2, 3]` yields $\frac{3}{4}$.

ChiSquared Command

ChiSquared[<Degrees of Freedom d>, x]

Creates probability density function (pdf) of Chi squared distribution with *d* degrees of freedom.

ChiSquared[<Degrees of Freedom>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of Chi squared distribution, otherwise creates pdf of Chi squared distribution.

ChiSquared[<Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function of Chi squared distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with Chi squared distribution with *d* degrees of freedom.

Note: Returns the probability for a given *x*-coordinate's value (or area under the Chi squared distribution curve to the left of the given *x*-coordinate).

CAS Syntax

In CAS View only following syntax is supported:

ChiSquared[<Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function (cdf) of Chi squared distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with Chi squared distribution with *d* degrees of freedom.

Example:

`ChiSquared[4, 3]` yields $\gamma(2, \frac{3}{2})$, which is approximately *0.44*.

Erlang Command

Erlang[<Shape k>, <Rate λ>, x]

Creates probability density function (pdf) of Erlang distribution with parameters *k, λ*.

Erlang[<Shape k>, <Rate λ>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of Erlang distribution, otherwise creates pdf of Erlang distribution.

Erlang[<Shape k>, <Rate λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of Erlang distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with Erlang distribution given by parameters *k, λ*.

Note: Returns the probability for a given *x*-coordinate's value (or area under the Erlang distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

Erlang[<Shape k>, <Rate λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of Erlang distribution at *v*, i.e. the probability *P(X≤v)* where *X* is a random variable with Erlang distribution given by parameters *k, λ*.

Exponential Command

Exponential[<Rate parameter λ>, x]

Creates probability density function (pdf) of exponential distribution with rate parameter λ.

Exponential[<Rate parameter λ>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function (cdf) of exponential distribution, otherwise creates pdf of Exponential distribution.

Exponential[<Rate parameter λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of Exponential distribution at *v*, i.e. the probability *P(X≤v)* where *X* is a random variable with Exponential distribution with rate parameter λ.

Note: Returns the probability for a given *x*-coordinate's value (or area under the Exponential distribution curve to the left of the given *x*-coordinate).

CAS Syntax

In CAS View only following syntax is supported:

Exponential[<Rate parameter λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of exponential distribution at *v*, i.e. the probability *P(X≤v)* where *X* is a random variable with Exponential distribution with rate parameter λ.

Example:

`Exponential[2, 1]` yields *\(\frac{e^{2}-1}{e^{2} } \)*, which is approximately *0.86*.

FDistribution Command

FDistribution[<Numerator Degrees of Freedom n>, <Denominator Degrees of Freedom d>, x]

Creates probability density function (pdf) of F-distribution with parameters *n, d*.

FDistribution[<Numerator Degrees of Freedom n>, <Denominator Degrees of Freedom d>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of F-distribution, otherwise creates pdf of F-distribution.

FDistribution[<Numerator Degrees of Freedom n>, <Denominator Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function of F-distribution at *v*, i.e. the probability $P(X\leq v)$ where *X* is a random variable with F-distribution given by parameters *n, d*.

Note: Returns the probability for a given *x*-coordinate's value (or area under the F-distribution curve to the left of the given *x*-coordinate).

CAS Syntax

In CAS View only following syntax is supported:

FDistribution[<Numerator Degrees of Freedom n>, <Denominator Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function of F-distribution at *v*, i.e. the probability $P(X\leq v)$ where *X* is a random variable with F-distribution given by parameters *n, d*.

Gamma Command

Gamma[<Number α>, <Number β>, x]

Creates probability density function (pdf) of gamma distribution with parameters α, β.

Gamma[<Number α>, <Number β>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of gamma distribution, otherwise creates pdf of gamma distribution.

Gamma[<Number α>, <Number β>, <Variable Value v>]

Calculates the value of cumulative distribution function of gamma distribution at *v*, i.e. the probability $P(X\leq v)$ where *X* is a random variable with gamma distribution given by parameters α, β.

Note: Returns the probability for a given *x*-coordinate's value (or area under the gamma distribution curve to the left of the given *x*-coordinate).

CAS Syntax

In CAS View only following syntax is supported:

Gamma[<Number α>, <Number β>, <Variable Value v>]

Calculates the value of cumulative distribution function of gamma distribution at *v*, i.e. the probability $P(X\leq v)$ where *X* is a random variable with gamma distribution given by parameters α, β.

HyperGeometric Command

HyperGeometric[<Population Size>, <Number of Successes>, <Sample Size>]

Returns a bar graph of a Hypergeometric distribution.

Parameters:

Population size: number of balls in the urn

Number of Successes: number of white balls in the urn

Sample Size: number of balls drawn from the urn

The bar graph shows the probability function of the number of white balls in the sample.

HyperGeometric[<Population Size>, <Number of Successes>, <Sample Size>, <Boolean Cumulative>]

Returns a bar graph of a Hypergeometric distribution when *Cumulative* = false.

Returns a bar graph of a cumulative Hypergeometric distribution when *Cumulative* = true.

First three parameters are same as above.

HyperGeometric[<Population Size>, <Number of Successes>, <Sample Size>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Hypergeometric random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First three parameters are same as above.

CAS Syntax

In CAS View only one syntax is allowed:

HyperGeometric[<Population Size>, <Number of Successes>, <Sample Size>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Hypergeometric random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First three parameters are same as above.

Example:

Assume you select two balls out of ten balls, two of which are white, without putting any back.

- `HyperGeometric[10, 2, 2, 0, false]` yields $\frac{28}{45}$, the probability of selecting zero white balls,
- `HyperGeometric[10, 2, 2, 1, false]` yields $\frac{16}{45}$, the probability of selecting one white ball,
- `HyperGeometric[10, 2, 2, 2, false]` yields $\frac{1}{45}$, the probability of selecting both white balls,
- `HyperGeometric[10, 2, 2, 3, false]` yields *0*, the probability of selecting three white balls.
- `HyperGeometric[10, 2, 2, 0, true]` yields $\frac{28}{45}$, the probability of selecting zero (or less) white balls,
- `HyperGeometric[10, 2, 2, 1, true]` yields $\frac{44}{45}$, the probability of selecting one or less white balls,

- `HyperGeometric[10, 2, 2, 2, true]` yields *1*, the probability of selecting tow or less white balls and
- `HyperGeometric[10, 2, 2, 3, true]` yields *1*, the probability of selecting three or less white balls.

InverseBinomial Command

InverseBinomial[<Number of Trials>, <Probability of Success>, <Probability p>]

Returns least integer *n* such that $P(X \leq n) \geq p$, where X is binomial random variable given by *Number of Trials* and *Probability of Success*.

Note: See also Binomial Command.

InverseCauchy Command

InverseCauchy[<Median m>, <Scale s>, <Probability p>]

Computes the inverse of cumulative distribution function of Cauchy distribution at *p*, where the Cauchy distribution is given by median *m* and scale *s*. In other words, finds *t* such that $P(X \leq t)=p$, where X is Cauchy random variable. Probability *p* must be from [0,1].

InverseChiSquared Command

InverseChiSquared[<Degrees of Freedom d>, <Probability p>]

Computes the inverse of cumulative distribution function of Chi squared distribution at *p*, where the Chi squared distribution has given *d* degrees of freedom. In other words, finds *t* such that $P(X \leq t)=p$, where X is Chi squared random variable. Probability *p* must be from [0,1].

InverseExponential Command

InverseExponential[<Mean λ>, <Probability p>]

Computes the inverse of cumulative distribution function of exponential distribution at *p*, where the exponential distribution is given by mean *λ*. In other words, finds *t* such that *P(X≤t)=p*, where X is exponential random variable. Probability *p* must be from [0,1].

InverseFDistribution Command

InverseFDistribution[<Numerator Degrees of Freedom n>, <Denominator Degrees of Freedom d>, <Probability p>]

Computes the inverse of cumulative distribution function of F-distribution at *p*, where the exponential distribution is given by parameters *n, d*. In other words, finds *t* such that *P(X≤t)=p*, where X is random variable with F-distribution. Probability *p* must be from [0,1].

InverseGamma Command

InverseGamma[<Number α>, <Number β>, <Probability p>]

Computes the inverse of cumulative distribution function of gamma distribution at *p*, where the gamma distribution is given by parameters *α, β*. In other words, finds *t* such that *P(X≤t)=p*, where X is random variable with gamma distribution. Probability *p* must be from [0,1].

InverseHyperGeometric Command

InverseHyperGeometric[<Population Size>, <Number of Successes>, <Sample Size>, <Probability p>]

Returns least integer n such that $P(X \le n) \ge p$, where X is hypergeometric random variable given by *Population Size*, *Number of Successes* and *Sample Size*.

Note: See also HyperGeometric Command.

InverseNormal Command

InverseNormal[Mean μ, Standard Deviation σ, Probability P]

Calculates the function $\Phi^{-1}(P) * \sigma + \mu$ where Φ^{-1} is the inverse of the cumulative distribution function Φ for $N(0,1)$.

Note: Returns the *x*-coordinate with the given probability to the left under the normal distribution curve.

InversePascal Command

InversePascal[<Number of Successes>, <Probability of Success>, <Probability p>]

Returns least integer n such that $P(X \le n) \ge p$, where X is Pascal random variable [1] given by *Number of Successes* and *Probability of Success*.

Note: See also Pascal Command.

References

[1] http://mathworld.wolfram.com/NegativeBinomialDistribution.html

InversePoisson Command

InversePoisson[<Mean λ>, <Probability p>]

Returns least integer *n* such that $P(X \le n) \ge p$, where X is Poisson random variable with mean λ.

Note: See also Poisson Command.

InverseTDistribution Command

InverseTDistribution[<Degrees of Freedom d>, <Probability p>]

Computes the inverse of cumulative distribution function of t-distribution at *p*, where the t-distribution has *d* degrees of freedom. In other words, finds *r* such that $P(X \le r)=p$, where X is random variable with t-distribution. Probability *p* must be from [0,1].

InverseWeibull Command

InverseWeibull[<Shape k>, <Scale λ>, <Probability p>]

Computes the inverse of cumulative distribution function of Weibull distribution at *p*, where the Weibull distribution is given by shape parameter *k* and scale parameter *λ*. In other words, finds *t* such that $P(X \le t) = p$, where X is random variable with Weibull distribution. Probability *p* must be from [0,1].

InverseZipf Command

InverseZipf[<Number of Elements>, <Exponent>, <Probability p>]

Returns least integer *n* such that $P(X \le n) \ge p$, where X is Zipf random variable given by *Number of Elements* and *Exponent*.

Note: See also Zipf Command.

LogNormal Command

LogNormal[<Mean μ>, <Standard Deviation σ>, x]

Creates probability density function (pdf) of log-normal distribution with parameters μ, σ.

LogNormal[<Mean μ>, <Standard Deviation σ>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative density function of LogNormal distribution, otherwise creates pdf of log-normal distribution.

LogNormal[<Mean μ>, <Standard Deviation σ>, <Variable Value v>]

Calculates the value of cumulative distribution function of log-normal distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with log-normal distribution given by parameters μ, σ.

Note: Returns the probability for a given *x*-coordinate's value (or area under the log-normal distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

LogNormal[<Mean μ>, <Standard Deviation σ>, <Variable Value v>]

Calculates the value of cumulative distribution function of log-normal distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with log-normal distribution given by parameters μ, σ.

Logistic Command

Logistic[<Mean μ>, <Scale s>, x]

Creates probability density function (pdf) of logistic distribution with parameters μ, s.

Logistic[<Mean μ>, <Scale s>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of logistic distribution, otherwise creates pdf of logistic distribution.

Logistic[<Mean μ>, <Scale s>, <Variable Value v>]

Calculates the value of cumulative distribution function of logistic distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with logistic distribution given by parameters μ, s.

Note: Returns the probability for a given *x*-coordinate's value (or area under the logistic distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

Logistic[<Mean μ>, <Scale s>, <Variable Value v>]

Calculates the value of cumulative distribution function of logistic distribution at *v*, i.e. the probability $P(X \le v)$ where *X* is a random variable with log-normal distribution given by parameters μ, s.

Normal Command

Normal[<Mean μ>, <Standard Deviation σ>, <x>]

Creates probability density function (pdf) of normal distribution.

Normal[<Mean μ>, <Standard Deviation σ>, <x>, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of normal distribution, otherwise creates pdf of normal distribution.

Normal[<Mean μ>, <Standard Deviation σ>, <Variable Value v>]

Calculates the function $\Phi((x - \mu) / \sigma)$ at v where Φ is the cumulative distribution function for $N(0,1)$.

Note: Returns the probability for a given *x*-coordinate's value (or area under the normal distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

Normal[<Mean μ>, <Standard Deviation σ>, <Variable Value x>]

Calculates the function $\Phi((x - \mu) / \sigma)$ where Φ is the cumulative distribution function for $N(0,1)$.

Example:

`Normal[2, 0.5, 1]` yields \(0.5 erf(-\sqrt{2}) + 0.5\).

Pascal Command

Pascal[<Number of Successes r>, <Probability of Success p>]

Returns a bar graph of a Pascal distribution [1].

Parameters:

Number of Successes: number of independent Bernoulli trials that must be successful

Probability of Success: probability of success in one trial

Pascal[<Number of Successes>, <Probability of Success>, <Boolean Cumulative>]

Returns a bar graph of a Pascal distribution when *Cumulative* = false.

Returns a bar graph of a cumulative Pascal distribution when *Cumulative* = true.

First two parameters are same as above.

Pascal[<Number of Successes>, <Probability of Success>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Pascal random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First two parameters are same as above.

CAS Syntax

In CAS View only one syntax is allowed:

Pascal[<Number of Successes>, <Probability of Success>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Pascal random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

Poisson Command

Poisson[<Mean λ>]

Returns a bar graph of a Poisson distribution with given mean λ.

Poisson[<Mean λ>, <Boolean Cumulative>]

Returns a bar graph of a Poisson distribution when *Cumulative = false*.

Returns a bar graph of a cumulative Poisson distribution when *Cumulative = true*.

The first parameter is same as above.

Poisson[<Mean λ>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Poisson random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First parameter is same as above.

CAS Specific Syntax

In CAS View only one syntax is supported:

Poisson[<Mean λ>, <Variable Value v>, <Boolean Cumulative>]

Let X be a Poisson random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First parameter is same as above.

RandomBetween Command

RandomBetween[Min Integer, Max Integer]

Generates a random integer between *min* and *max* (inclusive).

RandomBinomial Command

RandomBinomial[<Number n of Trials>, <Probability p>]

Generates a random number from a binomial distribution with *n* trials and probability *p*.

Example:

`RandomBinomial[3, 0.1]` gives $j \in \{0, 1, 2, 3\}$, where the probability of getting *j* is the probability of an event with probability *0.1* occuring *j* times in three tries.

CAS Syntax

RandomBinomial[<Number n of Trials>, <Probability p>]

Generates a random number from a binomial distribution with *n* trials and probability *p*.

Example:

`RandomBinomial[3, 0.1]` gives $j \in \{0, 1, 2, 3\}$, where the probability of getting *j* is the probability of an event with probability *0.1* occuring *j* times in three tries.

Note:

See also RandomNormal Command and RandomPoisson Command.

RandomNormal Command

RandomNormal[<Mean>, <Standard Deviation>]

Generates a random number from a normal distribution with mean and standard deviation.

Example:

`RandomNormal[3, 0.1]` yields a random value from a normal distribution with a mean of *3* and standard deviation of *0.1*.

Note:

See also RandomBinomial Command and RandomPoisson Command.

CAS Syntax

RandomNormal[<Mean>, <Standard Deviation>]

Generates a random number from a normal distribution with mean and standard deviation.

Example:

`RandomNormal[3, 0.1]` yields a random value from a normal distribution with a mean of *3* and standard deviation of *0.1*.

Note:

See also RandomBinomial Command and RandomPoisson Command.

RandomPoisson Command

RandomPoisson[<Mean>]

Generates a random number from a Poisson distribution with given mean.

Example:

`RandomPoisson[3]` yields a random value from a Poisson distribution with a mean of *3*.

Note:

See also RandomBinomial Command and RandomNormal Command.

CAS Syntax

RandomPoisson[<Mean>]

Generates a random number from a Poisson distribution with given mean.

Example:

`RandomPoisson[3]` yields a random value from a Poisson distribution with a mean of *3*.

Note:

See also RandomBinomial Command and RandomNormal Command.

RandomUniform Command

RandomUniform[<Min>, <Max>]

Returns random real number from uniform distribution on interval [min,max].

Note: `RandomUniform[0,1]` is equivalent to `random()` (see Predefined Functions and Operators).

TDistribution Command

TDistribution[<Degrees of Freedom d>, x]

Creates probability density function (pdf) of t-distribution with *d* degrees of freedom.

TDistribution[<Degrees of Freedom d>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of t-distribution, otherwise creates pdf of t-distribution.

TDistribution[<Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function of t-distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with t-distribution with *d* degrees of freedom.

Note: Returns the probability for a given *x*-coordinate's value (or area under the t-distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

TDistribution[<Degrees of Freedom d>, <Variable Value v>]

Calculates the value of cumulative distribution function of T-distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with t-distribution with *d* degrees of freedom.

Triangular Command

Triangular[<Lower Bound min>, <Upper Bound max>, <Mode mod>, x]

Creates probability density function of triangular distribution with parameters *min, max, mod*.

Triangular[<Lower Bound min>, <Upper Bound max>, <Mode mod>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of triangular distribution, otherwise creates probability density function of triangular distribution.

Triangular[<Lower Bound min>, <Upper Bound max>, <Mode mod>, <Variable Value v>]

Calculates the value of cumulative distribution function of triangular distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with triangular distribution given by parameters *min, max, mod*.

Note: Returns the probability for a given *x*-coordinate's value (or area under the triangular distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

Triangular[<Lower Bound min>, <Upper Bound max>, <Mode mod>, <Variable Value v>]

Calculates the value of cumulative distribution function of triangular distribution at *v*, i.e. the probability $P(X \leq v)$ where *X* is a random variable with triangular distribution given by parameters *min, max, mod*.

Uniform Command

Uniform[<Lower Bound min>, <Upper Bound max>, x]

Returns the probability density function of uniform distribution on interval *[min,max]*.

Uniform[<Lower Bound min>, <Upper Bound max>, x, <Boolean Cumulative b>]

For *b=false* returns the probability density function of uniform distribution on interval *[min,max]*.

For *b=true* returns the cumulative distribution function of the same distribution.

Uniform[<Lower Bound min>, <Upper Bound max>, <Variable Value v>]

Returns the value of cumulative distribution function at *v* (i.e. P(X<v)) for uniform distribution on interval *[min,max]*

Weibull Command

Weibull[<Shape k>, <Scale λ>, x]

Creates probability density function (pdf) of Weibull distribution with parameters *k, λ*.

Weibull[<Shape k>, <Scale λ>, x, <Boolean Cumulative>]

If *Cumulative* is true, creates cumulative distribution function of Weibull distribution, otherwise creates pdf of Weibull distribution.

Weibull[<Shape k>, <Scale λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of Weibull distribution at *v*, i.e. the probability *P(X≤v)* where *X* is a random variable with Weibull distribution given by parameters *k, λ*.

Note: Returns the probability for a given *x*-coordinate's value (or area under the Weibull distribution curve to the left of the given *x*-coordinate).

CAS Syntaxes

In CAS View only following syntax is supported:

Weibull[<Shape k>, <Scale λ>, <Variable Value v>]

Calculates the value of cumulative distribution function of Weibull distribution at *v*, i.e. the probability *P(X≤v)* where *X* is a random variable with Weibull distribution given by parameters *k, λ*.

Zipf Command

Zipf[<Number of Elements>, <Exponent>]

Returns a bar graph of a Zipf distribution.

Parameters:

Number of Elements: number of elements whose rank we study

Exponent: exponent characterizing the distribution

Zipf[<Number of Elements>, <Exponent> , <Boolean Cumulative>]

Returns a bar graph of a Zipf distribution when *Cumulative* = false.

Returns a bar graph of a cumulative Pascal distribution when *Cumulative* = true.

First two parameters are same as above.

Zipf[<Number of Elements>, <Exponent> , <Variable Value v>, <Boolean Cumulative>]

Let X be a Zipf random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

First two parameters are same as above.

CAS Syntax

In CAS View only one syntax is allowed:

Zipf[<Number of Elements>, <Exponent> , <Variable Value v>, <Boolean Cumulative>]

Let X be a Zipf random variable.

Returns P(X = *v*) when *Cumulative* = false.

Returns P(X ≤ *v*) when *Cumulative* = true.

Spreadsheet Commands

These commands are designed for referencing data from Spreadsheet View and copying data into it.

- Cell
- CellRange
- Column
- ColumnName
- FillCells
- FillColumn
- FillRow
- Row

Cell Command

Cell[<Column>, <Row>]

Returns copy of spreadsheet cell in given column and row.

Example: `Cell[2,1]` returns copy of B1.

Note: By default the cells in spreadsheet cells are auxiliary and in such case this command returns auxiliary object as well.

Note: You must make sure that the cells you refer to are **earlier** in the Construction_Protocol than this command

CellRange Command

CellRange[Start Cell, End Cell]

Creates a list containing the cell values in this cell range.

Example: Enter the following values into the corresponding spreadsheet cells: `A1 = 1, A2 = 4, A3 = 9`. The command `CellRange[A1, A3]` then gives you the list {1, 4, 9}.

Note: You may also use shorter syntax `A1:A3`.

Column Command

Column[Spreadsheet Cell]

Returns the column of the cell as a number (starting at 1).

Example: `Column[B3]` gives you number *a = 2* since column *B* is the second column of the spreadsheet.

ColumnName Command

ColumnName[Spreadsheet Cell]

Returns the column name of the cell as a text.

Example: `ColumnName[A1]` gives you a text *"A"* in the Graphics View.

FillCells Command

FillCells[<CellRange>, <Object>]

Copies the value/equation etc. of the object to the given cellrange. Resulting cells are free objects (i.e. independent of object).

Note: The parameter *CellRange* has to be entered like this: e.g.: *B2:D5*.

Note: The parameter *Object* can be anything, e.g.: *3*, *RandomBetween[0,10]*, *Circle[A,B]*.

FillCells[<Cell>, <List>]

Copies values from the list to the first cells on the right of the given cell. Resulting cells are free objects (i.e. independent of list L).

FillCells[<Cell>, <Matrix>]

Copies values from the matrix. The upper left corner of the matrix is matched to the given cell. Resulting cells are free objects (i.e. independent of matrix).

Note: Cells are labeled by column and row, e.g.: *B2*.

FillColumn Command

FillColumn[<Column>, <List L>]

Copies values from the list L to the first cells of the column given by number (1 for A, 2 for B, etc.). Resulting cells are free objects (i.e. independent on list L).

FillRow Command

FillRow[<Row>, <List L>]

Copies values from the list L to the first cells of the row given by number. Resulting cells are free objects (i.e. independent on list L).

Row Command

Row[Spreadsheet Cell]

Returns the row number of a spreadsheet cell (starting at 1).

Example: `Row[B3]` gives you number *a = 3*.

Scripting Commands

These commands are substitutes for features accessible e.g. via Properties Dialog and are meant to simplify scripting in GeoGebra.

Note: These commands don't return any object, therefore cannot be nested in other commands.

- Button
- Checkbox
- CopyFreeObject
- Delete
- Execute
- GetTime
- HideLayer
- InputBox
- Pan
- ParseToFunction
- ParseToNumber
- PlaySound
- Rename
- SelectObjects
- SetActiveView
- SetAxesRatio
- SetBackgroundColor
- SetCaption
- SetColor
- SetConditionToShowObject

- SetCoords
- SetDynamicColor
- SetFilling
- SetFixed
- SetLabelMode
- SetLayer
- SetLineStyle
- SetLineThickness
- SetPointSize
- SetPointStyle
- SetTooltipMode
- SetValue
- SetVisibleInView
- ShowLabel
- ShowLayer
- Slider
- StartAnimation
- Translate
- UpdateConstruction
- ZoomIn
- ZoomOut

Button Command

Button[]

Creates new button.

Button[Caption]

Creates new button with given caption.

Checkbox Command

Checkbox[]

Creates a checkbox.

Checkbox[<Caption>]

Creates a checkbox with given caption.

Checkbox[<List>]

Creates a checkbox which, when unchecked, hides listed objects.

Example: Let *A* and *B* be points. `c=Checkbox[{A,B}]` creates checkbox *c*. When *c* is checked, *A* and *B* are visible, otherwise they are hidden.

Checkbox[<Caption>, <List>]

Creates checkbox with given caption which, when unchecked, hides listed objects.

CopyFreeObject Command

CopyFreeObject[<Object>]

Creates a free copy of the object. Preserves all basic Object Properties and copy of Auxiliary Object is auxiliary as well.

Delete Command

Delete[<Object>]

Deletes the object and all its dependent objects.

Note: See also Delete Object tool.

CAS Syntax

Delete[<Object>]

Deletes the object and all its dependent objects in Geogebra and removes any value assigned to the object in the CAS.

Example:

`Delete[a]` clears *a*.

Execute Command

Execute[<List of Texts>]

Executes list of commands entered as texts.

Example:

- `Execute[{"A=(1,1)","B=(3,3)","C=Midpoint[A,B]"}]` creates points *A*, *B* and their midpoint *C*.
- `Execute[Join[{"f_{1}=1","f_{2}=1"},Sequence["f_{"+(i+2)+"}=f_{"+(i+1)+"}+f_{"+i+"}",i,1` creates first 10 elements of Fibonaci sequence.

Execute[<List of Texts>,<Parameter 1>,....,<Parameter 9>]

Replaces %1 for parameter 1, %2 for parameter 2 and so on in each text in list. Up to 9 parameters can be specified. After the replacement, resulting scripts are executed.

Example: `Execute[{"Midpoint[%1,%2]"},A,B]` creates midpoint of segment *AB*.

GetTime Command

GetTime[]

Returns a list such as *{647, 59, 39, 23, 28, 2, 2011, "February", "Monday", 2}*, ie the current time and date in this order:

milliseconds, seconds, minutes, hours (0-23), date, month (1-12), year, month (as text), day (as text), day (1 = Sunday, 2 = Monday, etc)

HideLayer Command

HideLayer[<Number>]

Makes all objects in given layer invisible. Does not override Conditional Visibility.

Pan Command

Pan[<x>, <y>]

Shifts the view by *x* pixels to the left and *y* pixels upwards.

ParseToFunction Command

ParseToFunction[<Function f>, <String>]

Parses the string and stores the result to function *f*. Function *f* must be defined and free before the command is used.

ParseToNumber Command

ParseToNumber[<Number a>, <String>]

Parses the string and stores the result to number *a*. Number *a* must be defined and free before the command is used.

PlaySound Command

PlaySound[<Note>, <Duration>, <Instrument>]

Plays a MIDI note.

Note is an integer from 0 to 127 that represents a musical note given by the table below. When note = 60 a Middle C is played.

Duration is the time to play the note in seconds.

Instrument is an integer that represents the synthesized instrument used to play the note. See technical specifications at midi.org [1] for possible instruments.

Most instruments are supported, but there are differences between computer platforms.

MIDI Notes

PlaySound[<Note Sequence>, <Instrument>]

Plays a sequence of MIDI notes and commands using a JFugue [2] music string.

Note Sequence is a text string that uses JFugue character commands.

Instrument is the default MIDI instrument used when the string is played.

The basic commands are given below. The full command set is described in

The Complete Guide to JFugue [3] (English).

Basic JFugue Commands

Character	**+**	A-G	+	[number]	+	+	+	R	+	w, h, q, i, s	+	/n	+	I[number]	+	V	+	Space	Play the previous note, combination of notes or rest.

Example: `PlaySound["C+E+G Rw Ai Bi Ci A4i B4i C4i ", 0]` Plays a quarter note chord CEG; rests for a whole note; plays the eighth notes A, B, C; plays them again one octave lower. Piano instrument is used.

Example: `PlaySound["I[56] C5q D5q I[71] G5q F5q", 0]` Plays notes with different instruments. Trumpet = 56 and Clarinet = 71.

Example: `PlaySound["V0 A3q B3q C3q B3q V1 A2h C2h", 0]` Plays notes in harmony with different voices.

PlaySound[<File>]

Plays a MIDI file (*.mid) or a text file (* .txt) containing a JFugue string.

"File" is the directory path to this file, e.g. PlaySound["path/to/myFile.mid"]

PlaySound[<Function>, <Min Value>, <Max Value>]

Plays a sound generated by Function, a time-valued function with range [-1,1]. The time units are seconds and the sound is played from time Min Value to Max Value. Sound is generated by 8-bit samples taken at a rate of 8000 samples per second.

Example: `PlaySound[sin(440 2Pi x), 0, 1]`This plays a pure sine wave tone at 440 Hz (musical note A) for one second.

PlaySound[<Function>, <Min Value>, <Max Value>, <Sample Rate>, <Sample Depth>]

Plays a sound generated by Function, a time-valued function with range [-1,1]. The time units are seconds and the sound is played from time Min Value to Max Value. The sampling method is specified by "Sample Depth" and "Sample Rate".

"Sample Rate" is the number of sample function values taken each second. Allowable values are 8000, 11025, 16000, 22050, or 44100

"Sample Depth" is the data size of a sample in bits. Allowable values are 8 and 16.

PlaySound[<Boolean Play>]

Pause or resume play.

PlaySound[true] = play, PlaySound[false] = pause.

References

[1] http://www.midi.org/techspecs/gm1sound.php
[2] http://www.jfugue.org
[3] http://www.jfugue.org/jfugue-chapter1.pdf

Rename Command

Rename[<Object>, <Name N>]

Sets the label of given object to N.

SelectObjects Command

SelectObjects[ ]

Deselects all selected objects.

SelectObjects[<Object>, <Object>, ...]

Deselects all selected objects and selects objects passed as parameters. All parameters must be labeled objects (e.g. `SelectObjects[Midpoint[A,B]]` won't do anything).

SetActiveView Command

SetActiveView[<View Number 1|2>]

Makes given Graphics View active.

SetAxesRatio Command

SetAxesRatio[<Number X>, <Number Y>]

Changes the axes ratio of active Graphics View so that X units on x-axis correspond to the same number of pixels as Y units on y-axis and point (0,0) stays on its coordinates.

SetBackgroundColor Command

SetBackgroundColor[<Object>, <Red>, <Green>, <Blue>]

Changes the background color of given object. This is used for Texts and for objects in the Spreadsheet. The red, green and blue represent amount of corresponding color component, 0 being minimum and 1 maximum. Number t exceeding this interval is mapped to it using function $2\left|\frac{t}2-\mathrm round\left(\frac{t}2\right)\right|$.

SetBackgroundColor[<Object>, "color"]

Changes the background color of given object. This is used for Texts and for objects in the Spreadsheet. The color is entered as text. The command accepts more than a hundred English color names (see Reference:Colors). Some of them can be also used in national languages and are listed below.

- Black
- Dark Gray
- Gray
- Dark Blue
- Blue
- Dark Green
- Green
- Maroon
- Crimson
- Red
- Magenta
- Indigo
- Purple
- Brown
- Orange
- Gold
- Lime
- Cyan
- Turquoise
- Light Blue
- Aqua
- Silver
- Light Gray
- Pink
- Violet
- Yellow
- Light Yellow
- Light Orange
- Light Violet
- Light Purple
- Light Green
-

SetCaption Command

SetCaption[<Object>, <Text>]

Changes caption of given object.

SetColor Command

SetColor[<Object>, <Red>, <Green>, <Blue>]

Changes the color of given object. The red, green and blue represent amount of corresponding color component, 0 being minimum and 1 maximum. Number t exceeding this interval is mapped to it using function $2\left|\frac{t}{2}-\mathrm{round}\left(\frac{t}{2}\right)\right|$.

SetColor[<Object>, "color"]

Changes the color of given object. The color is entered as text. The command accepts more than a hundred English color names (see Reference:Colors). Some of them can be also used in national languages and are listed below.

- Black
- Dark Gray
- Gray
- Dark Blue
- Blue
- Dark Green
- Green
- Maroon
- Crimson
- Red
- Magenta
- Indigo
- Purple
- Brown
- Orange
- Gold
- Lime
- Cyan
- Turquoise
- Light Blue
- Aqua
- Silver
- Light Gray
- Pink
- Violet
- Yellow
- Light Yellow
- Light Orange
- Light Violet
- Light Purple

- Light Green
-

SetConditionToShowObject Command

SetConditionToShowObject[<Object>, <Condition>]

Sets the condition to show given object.

SetCoords Command

SetCoords[<Point>, <x>, <y>]

Changes cartesian coordinates of free point or vector. This command uses values of the coordinates, not their definitions, therefore the point stays free.

SetDynamicColor Command

SetDynamicColor[<Object>, <Red>, <Green>, <Blue>]

Sets the dynamic color of the object.

SetDynamicColor[<Object>, <Red>, <Green>, <Blue>, <Opacity>]

Sets the dynamic color and opacity of the object.

Note: All numbers are on a scale from 0 (off/transparent) to 1 (on/opaque)

SetFilling Command

SetFilling[<Object>, <Number>]

Changes the opacity of given object. Number must be from interval [0,1], where 0 means transparent and 1 means 100% opaque. Other numbers are ignored.

SetFixed Command

SetFixed[<Object>, <true | false>]

Makes the object fixed (for true) or not fixed (for false).

SetLabelMode Command

SetLabelMode[<Object>, <Number>]

Changes the label mode of given object according to the table below. Integers distinct from the ones listed in table are treated as 0.

Number	Mode
0	Name
1	Name + Value
2	Value
3	Caption

SetLayer Command

SetLayer[<Object>, <Layer n>]

Sets the layer for given object to *n*, where *n* must be integer from {0, 1, ..., 9}.

SetLineStyle Command

SetLineStyle[<Object>, <Number>]

Changes the line style of given object according to following table (numbers out of range [0,4] are not valid).

Number	Style
0	Full
1	Dashed long
2	Dashed short
3	Dotted
4	Dash-dot

SetLineThickness Command

SetLineThickness[<Path>, <Number N>]

Sets the line thickness for given path to $\frac{N}{2}$ pixels. Maximal allowed value of the number is 13.

SetPointSize Command

SetPointSize[<Point>, <Number>]

Changes the size of the point.

SetPointStyle Command

SetPointStyle[<Point>, <Number>]

Changes the point style of given point according to following table (numbers out of range [0,9] are not valid).

Number	Style	Symbol
0	Full dot	●
1	Cross	▯
2	Empty dot	○
3	Plus sign	+
4	Full diamond	◆
5	Empty diamond	◇
6	Triangle north	▲
7	Triangle south	▼
8	Triangle east	▶
9	Triangle west	◀

SetTooltipMode Command

SetTooltipMode[<Object>, <Number>]

Changes the tooltip mode for given object according to following table (values out of range [0,4] are treated as 0):

Number	Mode
0	Automatic
1	On
2	Off
3	Caption
4	Next cell

SetValue Command

SetValue[<Object A>, <Object B>]

If A is a free object or a Point restricted to Path or Region, its value is set to current value of *B* (i.e. *A* doesn't change value if *B* is changed afterwards).

SetValue[<Boolean b>, <0|1>]

Sets the state of a boolean / check box : 1 = true, 0 = false

Example: If *b* is a boolean, `SetValue[b,1]` sets the boolean *b* as *true*.

SetValue[<List L>, <Number n>, <Object B>]

Sets n-th element of the free list L to the current value of B. Number *n* can be at most 1 + length of L.

SetVisibleInView Command

SetVisibleInView[<Object>, <View Number 1|2>, <Boolean>]

Makes object visible or hidden in given Graphics View.

ShowLabel Command

ShowLabel[<Object>, <Boolean>]

Shows or hides the label in the Graphics View for the given object.

ShowLayer Command

ShowLayer[<Number>]

Makes all objects in given layer visible. Does not override Conditional Visibility.

Slider Command

Slider[<Min>, <Max>, <Increment>, <Speed>, <Width>,<Is Angle>, <Horizontal>, <Animating>, <Boolean Random>]

Creates a slider with range [Min,Max], given increment, and speed. Width parameter sets its width in pixels, parameters Is Angle, Horizontal, Animating and Random may be true or false. Only the first two parameters are compulsory, default values for the others are 0.1, 1, 100, false, true, false, false respectively.

StartAnimation Command

StartAnimation[ ]

Resumes all animations if they are paused.

StartAnimation[<Boolean b>]

When b is false, pauses all animations, otherwise resumes them.

StartAnimation[<Point or Slider>, <Point or Slider>, ...]

Starts animating given points and sliders, the points must be on paths.

StartAnimation[<Point or Slider>, <Point or Slider>, ..., <Boolean b>]

Starts (for b = true) or permanently stops (for b = false) animating given points and sliders, the points must be on paths.

Note: See also Animation.

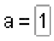
InputBox Command

InputBox[]

Create a new Input Box.

InputBox[<Linked Object>]

Create a new Input Box and associate a Linked Object with it.

Note: See also  Insert Input Box Tool.

UpdateConstruction Command

UpdateConstruction[]

Recomputes all objects (random numbers are regenerated). Same as or .

If you want to refresh the view (e.g to remove traces from Graphics View) you can use `ZoomIn` instead, which is the same as .

ZoomIn Command

ZoomIn[<Scale Factor>]

Zooms the Graphics View in by given factor with respect to current zoom, center of the screen is used as center point for the zoom.

Example: `ZoomIn[1]` doesn't do anything, `ZoomIn[2]` zooms the view in, `ZoomIn[0.5]`is equivalent to `ZoomOut[2]`, i.e. it zooms the view out.

ZoomIn[<Scale Factor>, <Center Point>]

Zooms the Graphics View in by given factor with respect to current zoom, second parameter specifies center point for the zoom.

ZoomIn[<Min-x>, <Min-y>, <Max-x>, <Max-y>]

Zooms the graphics view to the rectangle given by vertices (Min-x, Min-y), (Max-x,Max y). If any of these parameters are dependent or has label set, the bounds of the view become dynamic (e.g. if *a* is a slider, `ZoomIn[-a,-a,a,a]` makes the zoom of the view dependent on slider *a*). To avoid this behaviour, use CopyFreeObject Command.

Note: If multiple Graphics Views are present, the active one is used.

ZoomOut Command

ZoomOut[<Scale Factor>]

Zooms the Graphics View out by given factor with respect to current zoom, center of the screen is used as center point for the zoom.

ZoomOut[<Scale Factor>, <Center Point>]

Zooms the Graphics View out by given factor with respect to current zoom, second parameter specifies center point for the zoom.

Note: `ZoomOut[t]` and `ZoomOut[t,A]` are equivalent to `ZoomIn[1/t]` and `ZoomIn[1/t,A]` respectively.

Discrete Math Commands

- Convex hull
- DelaunayTriangulation
- Hull
- MinimumSpanningTree
- ShortestDistance
- Travelling Salesman
- Voronoi

ConvexHull Command

ConvexHull[<List of Points>]

Creates convex hull of given set of points. Returned object is a locus, so it is auxiliary.

DelaunayTriangulation Command

DelaunayTriangulation[<List of Points>]

Creates Delaunay Triangulation of the list of points. Returned object is a locus, so it is auxiliary.

Hull Command

Hull[<List of Points> , <Percentage p>]

Creates a characteristic hull of the points as described in Efficient generation of simple polygons for characterizing the shape of a set of points in the plane [1]. For *p=1*, result is the same as the result of ConvexHull Command. The lower percentage, the lower area of the hull. For *p=0* the area of resulting shape is not necessarily minimal.

Returned object is a locus, so it is auxiliary.

Note: Values of *p* greater than 1 are treated as 1, values less than 0 are treated as 0.

References

[1] http://www.geosensor.net/papers/duckham08.PR.pdf

MinimumSpanningTree Command

MinimumSpanningTree[<List of Points>]

Returns the minimum spanning tree of a complete graph on given vertices in which weight of edge *(u,v)* is the Euclidian distance between *u* and *v*. The resulting object is a locus.

ShortestDistance Command

ShortestDistance[<List of Segments>, <Start Point>, <End Point>, <Boolean Weighted>]

Finds shortest path between start point and endpoint in a graph given by list of segments. If weighted is false, weight of each edge is supposed to be 1 (i.e. we are looking for the path with least number of edges), otherwise it is the length of given segment (we are looking for the geometrically shortest path).

TravelingSalesman Command

TravelingSalesman[<List of Points>]

Returns the shortest closed path which goes through each point exactly once. Returned object is a locus, so it is auxiliary.

Voronoi Command

Voronoi[<List of Points>]

Draws the Voronoi diagram for given list of points. Returned object is a locus, so it is auxiliary.

GeoGebra Commands

- AxisStepX
- AxisStepY
- ClosestPoint
- ConstructionStep
- Corner
- DynamicCoordinates
- Name
- Object
- SlowPlot
- ToolImage

AxisStepX Command

AxisStepX[]

Returns the current step width for the *x*-axis.

Note: Together with the Corner and Sequence commands, the AxisStepX and AxisStepY commands allow you to create custom axes (also see section Customizing Coordinate Axes and Grid).

AxisStepY Command

AxisStepY[]

Returns the current step width for the *y*-axis.

Note: Together with the Corner and Sequence commands, the AxisStepX and AxisStepY commands allow you to create custom axes (also see section Customizing Coordinate Axes and Grid).

ClosestPoint Command

ClosestPoint[Path P, Point A]

Returns a point on path *P* which is the closest to point *A*.

Note: For Functions, this command will return the point vertically in line rather than the nearest point

ConstructionStep Command

ConstructionStep[]

Returns the current Construction Protocol step as a number.

ConstructionStep[Object]

Returns the Construction Protocol step for the given object as a number.

Corner Command

Corner[Number n of Corner]

For *n=1, 2, 3, 4* creates a point at the corner of the Graphics View, for *n=5* returns point *(w,h)*, where *w* and *h* are width and height of the graphics view in pixels. Always uses first graphics view, even if second is active.

Corner[Graphics view g, Number n of Corner]

Creates a point at the corner of g-th Graphics View (g = 1, 2) which is never visible in that view. Supported values of *n* are 1, 2, 3, 4 and 5 as above.

Corner[Image, Number n of Corner]

Creates a point at the corner of the image (n = 1, 2, 3, 4).

Corner[Text, Number n of Corner]

Creates a point at the corner of the text (n = 1, 2, 3, 4).

Note: The numbering of the corners is counter-clockwise and starts at the lower left corner.

DynamicCoordinates Command

DynamicCoordinates[Point A, Number X, Number Y]

Creates a point with coords (X, Y). This point is dependent, but can be moved. Whenever you try to move the new point to coordinates (x,y), point *A* is moved there and coordinates for the new point are calculated. Works best if point *A* is not visible. At least one of *X* and *Y* should depend on *A*.

Example:

- `B=DynamicCoordinates[A,round(x(A)),round(y(A))]`. When you try to move *B* to (1.3,2.1) using the  Move Tool, point *A* becomes (1.3,2.1) and *B* appears at (1,2).
- `B=DynamicCoordinates[A,x(A),min(y(A),sin(x(A)))]` creates a point under sin(x).

Note: PointIn[y<sin(x)] is the easier solution in this case.

- The following examples show other ways to restrain the positions of a point C: let A=Point[xAxis] and B=Point[xAxis]. Now type in the Input bar: `DynamicCoordinates[B, Min[x(B),x(A)],0]` , and press the Enter key, `SetVisibleInView[B, 1, false]`, and press the Enter key, `SetLayer[C, 1]`, and press the Enter key. Now, C cannot be moved to the right of A.
- Define A=(1,2). Now, type in the Input bar: `SetVisibleInView[A, 1, false]` and press the Enter key, `B=DynamicCoordinates[A, If[x(A) > 3, 3, If[x(A) < -(3), -3, If[x(A)< 0, round(x(A)), x(A)]]], If[x(A) < 0, 0.5, If[y(A) > 2, 2, If[y(A) <0, 0, y(A)]]]]`, and press the Enter key.
- This example makes A a sticky point when a point C is dragged near it. Define A=(1,2) and B=(2,3). Now, type in the Input bar: `SetVisibleInView[B, 1, false]` and press the Enter key, `C=DynamicCoordinates[B, If[Distance[A, B] < 1, x(A), x(B)], If[Distance[A, B] < 1, y(A), y(B)]]`.

Name Command

Name[Object]

Returns the name of an object as a text in the Graphics View.

Note: This command works properly only in dynamic text for objects (so that they work after objects are renamed). The *Name* command is the opposite of the Object command.

Object Command

Object[Name of Object as Text]

Returns the object for a given name. The result is always a dependent object.

Note: The *Object* command is the opposite of the Name command.

Example: If points *A1*, *A2*, ... , *A20* exist and slider *n* = *2*, then `Object["A" + n]` gives you a copy of point *A2*.

Note: You must make sure that the objects you refer to are **earlier** in the Construction_Protocol than this command

Warning: Object command cannot be used in Custom tools

SlowPlot Command

SlowPlot[<Function>]

Creates animated graph of given function: the function is plotted from left to right. The animation is controlled by a slider, which is also created by this command.

ToolImage Command

ToolImage[<Number>]

Creates image of tool icon with given number sized 32x32 pixels. See Reference:Toolbar for the numbering.

Optimization Commands

- Maximize
- Minimize

Maximize Command

Maximize[<dependent number>, <free number>]

Calculates the independent number which gives the maximal value of the dependent number. The independent number must be a slider and the slider interval will be used as the search interval. If the construction is complicated, this command might fail or quit to avoid using too much processor time.

Minimize Command

Minimize[<dependent number>, <free number>]

Calculates the independent number which gives the minimal value of the dependent number. The independent number must be a slider and the slider interval will be used as the search interval. If the construction is complicated, this command might fail or quit to avoid using too much processor time.

CAS Specific Commands

All of the following commands can be used in the CAS View.

- BinomialCoefficient
- BinomialDist
- CFactor
- CSolutions
- CSolve
- Cauchy
- ChiSquared
- Coefficients
- CommonDenominator
- Covariance
- Cross
- Decimal
- Degree
- Delete
- Denominator
- Derivative
- Determinant
- Dimension
- Div
- Division
- Divisors
- DivisorsList
- DivisorsSum
- Dot
- Element
- Expand
- Exponential
- FDistribution
- Factor
- Factors
- First
- FitExp
- FitLog
- FitPoly
- FitPow
- FitSin
- FractionalPart
- GCD
- Gamma
- HyperGeometric
- Identity
- Imaginary
- ImplicitDerivative

- IntegerPart
- Integral
- IntegralBetween
- Intersect
- Invert
- IsPrime
- LCM
- Last
- LeftSide
- Length
- Limit
- LimitAbove
- LimitBelow
- Max
- Mean
- Median
- Min
- MixedNumber
- Mod
- NIntegral
- nPr
- NRoot
- NSolutions
- NSolve
- NextPrime
- Normal
- Numerator
- Numeric
- PartialFractions
- Pascal
- PerpendicularVector
- Poisson
- PreviousPrime
- PrimeFactors
- Product
- RandomBetween
- RandomBinomial
- RandomElement
- RandomNormal
- RandomPoisson
- RandomPolynomial
- Rationalize
- Real
- ReducedRowEchelonForm
- RightSide
- Root
- SD

- Sample
- SampleSD
- SampleVariance
- Sequence
- Shuffle
- Simplify
- Solutions
- Solve
- SolveODE
- Substitute
- Sum
- TDistribution
- Take
- TaylorPolynomial
- ToComplex
- ToExponential
- ToPoint
- ToPolar
- Transpose
- Unique
- UnitPerpendicularVector
- UnitVector
- Variance
- Weibull
- Zipf

CFactor Command

CFactor[<Expression>]

Factorizes a given expression, allowing for complex factors.

Example:

`CFactor[x^2 + 4]` yields $(x + 2 ί) (x - 2 ί)$, the factorization of $x^2 + 4$.

CFactor[<Expression>, <Variable>]

Factorizes an expression with respect to a given variable, allowing for complex factors.

Example:

`CFactor[a^2 + x^2, a]` yields $(a + x ί) (a - x ί)$, the factorization of $a^2 + x^2$ with respect to a.

Example:

`CFactor[a^2 + x^2, x]` yields $(x + a ί) (x - a ί)$, the factorization of $a^2 + x^2$ with respect to x.

Note:

See also Factor Command.

CSolutions Command

CSolutions[<Equation>]

Solves a given equation for the main variable and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolutions[x^2 = -1]` yields $\{ί, -ί\}$, the complex solutions of $x^2 = -1$.

CSolutions[<Equation>, <Variable>]

Solves an equation for a given unknown variable and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolutions[a^2 = -1, a]` yields $\{ί, -ί\}$, the complex solutions of $a^2 = -1$.

CSolutions[<List of Equations>, <List of Variables>]

Solves a set of equations for a given set of unknown variables and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolutions[{y^2 = x - 1, x = 2 * y - 1}, {x, y}]` yields $\begin{pmatrix}1 + 2 ί&1 + ί\\1 - 2 ί&1 - ί\end{pmatrix}$,
the complex solutions of $y^2 = x - 1$ and $x = 2 * y - 1$.

Note:

- The complex ί is obtained by pressing .
- See also CSolve Command and Solutions Command.

CSolve Command

CSolve[<Equation>]

Solves a given equation for the main variable and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolve[x^2 = -1]` yields $\{x = ί, x = -ί\}$, the complex solutions of $x^2 = -1$.

CSolve[<Equation>, <Variable>]

Solves an equation for a given unknown variable and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolve[a^2 = -1, a]` yields $\{a = ί, a = -ί\}$, the complex solutions of $a^2 = -1$.

CSolve[<List of Equations>, <List of Variables>]

Solves a set of equations for a given set of unknown variables and returns a list of all solutions, allowing for complex solutions.

Example:

`CSolve[{y^2 = x - 1, x = 2 * y - 1}, {x, y}]` yields $\{\{x = 1 + 2 ί, y = 1 + ί\}, \{x = 1 - 2 ί, y = 1 - ί\}\}$, the complex solutions of $y^2 = x$ and $x = 2 * y - 1$.

Note:

- The complex ί is obtained by pressing .
- See also CSolutions Command and Solve Command.

CommonDenominator Command

CommonDenominator[<Expression>, <Expression>]

Returns the (lowest) common denominator of the two expressions.

Example:

`CommonDenominator[3 / (2 x + 1), 3 / (4 x^2 + 4 x + 1)]` yields $4 x^2 + 4 x + 1$.

Cross Command

Cross[<Vector u> , <Vector v>]

Calculates the cross product of *u* and *v*.

Example:

`Cross[{1, 3, 2}, {0, 3, -2}]` yields *{-12, 2, 3}*.

If a vector contains undefined variables, it yields a formula for the cross product.

Example:

`Cross[{a, b, c}, {d, e, f}]` yields *{b f - c e, -a f + c d, a e - b d}*.

Note:

- In the Input Bar you can use `u ⊗ v`.
- See also Dot Command.

Decimal Command

Decimal[<Expression>]

Returns the decimal representation of the expression.

Example:

`Decimal[2 / 3 + 1 / 15]` yields *0.73*.

Dimension Command

Dimension[<Vector>]

Returns the dimension of the vector.

Example:

`Dimension[{1, 2, 0, -4, 3}]` yields *5*.

Dimension[<Matrix>]

Returns the dimension of the matrix.

Example:

`Dimension[{{a, b}, {c, d}, {e, f}}]` yields *{3, 2}*.

Division Command

Division[<Dividend Number>, <Divisor Number>]

Returns the quotient (integer part of the result) and the remainder of the division of the two numbers.

Example:

`Division[16, 3]` yields *{5, 1}*.

Division[<Dividend Polynomial>, <Divisor Polynomial>]

Returns the quotient and the remainder of the division of the two polynomials.

Example:

`Division[x^2 + 3 x + 1, x - 1]` yields *{x + 4, 5}*.

Divisors Command

Divisors[<Number>]

Calculates the number of all the positive divisors, including the number itself.

Example:

`Divisors[15]` yields *4*, the number of all positive divisors of *15*, including *15*.

Note:

See also DivisorsList Command and DivisorsSum Command.

DivisorsList Command

DivisorsList[<Number>]

Gives the list of all the positive divisors, including the number itself.

Example:

`DivisorsList[15]` yields *{1, 3, 5, 15}*, the list of all positive divisors of *15*, including *15*.

Note:

See also Divisors Command and DivisorsSum Command.

DivisorsSum Command

DivisorsSum[<Number>]

Calculates the sum of all the positive divisors, including the number itself.

Example:

`DivisorsSum[15]` yields *24*, the sum of *1* + *3* + *5* + *15*.

Note:

See also Divisors Command and DivisorsList Command.

Dot Command

Dot[<Vector>, <Vector>]

Returns the dot product (scalar product) of the two vectors.

Example:

`Dot[{1, 3, 2}, {0, 3, -2}]` yields *5*, the scalar product of *{1, 2, 3}* and *{0, 3, -2}*.

Note:

See also Cross Command.

FractionalPart Command

FractionalPart[<Expression>]

Returns the fractional part of the expression.

Example:

- `FractionalPart[6/5]` yields $\frac{1}{5}$,
- `FractionalPart[1/5 + 3/2 + 2]` yields $\frac{7}{10}$.

Note:

See also IntegerPart Command.

Imaginary Command

Imaginary[<Complex Number>]

Returns the imaginary part of a given complex number.

Example:

`Imaginary[17 + 3 ί]` yields *3*.

Note:

- The complex ί is obtained by pressing .
- See also Real Command.

ImplicitDerivative Command

ImplicitDerivative[<Expression>, <Dependent Variable>, <Independent Variable>]

Yields the implicit derivative of the given expression.

Example:

`ImplicitDerivative[x^2 + y^2, x, y]` yields $-\frac{x}{y}$.

Note:

See also Derivative Command.

IntegerPart Command

IntegerPart[<Expression>]

Returns the integer part of the expression.

Example:

`IntegerPart[6/5]` yields *1*.

Example:

`IntegerPart[1/5 + 3/2 + 2]` yields *3*.

Note:

See also FractionalPart Command.

IsPrime Command

IsPrime[<Number>]

Returns *true* or *false* depending on whether the number is prime or not.

Example:

- `IsPrime[10]` yields *false*,
- `IsPrime[11]` yields *true*.

LeftSide Command

LeftSide[<Equation>]

Returns the left-hand side of the equation.

Example:

`LeftSide[x + 2 = 3x + 1]` yields $x + 2$.

LeftSide[{ <Equation> }]

Returns the left-hand sides of the equations.

Example:

`LeftSide[{a^2 + b^2 = c^2, x + 2 = 3x + 1}]` yields $\{a^2 + b^2, x + 2\}$.

Note:

See also RightSide Command.

MatrixRank Command

MatrixRank[<Matrix>]

Returns the rank of given matrix.

Example:

- `MatrixRank[{{2, 2}, {1, 1}}]` yields *1*,
- `MatrixRank[{{1, 2}, {3, 4}}]` yields *2*.

MixedNumber Command

MixedNumber[<Number>]

Converts the given number to a mixed number.

Example:

- `MixedNumber[3.5]` yields $3 + \frac{1}{2}$.
- `MixedNumber[12 / 3]` yields *4*.
- `MixedNumber[12 / 14]` yields $\frac{6}{7}$.

Note:

See also Rationalize Command.

NIntegral Command

NIntegral[<Function f>, <Start x-Value a>, <End x-Value b>]

Computes the definite integral $\int_a^bf(x)\mathrm{d}x$ numerically.

Example:

`NIntegral[`*e*`^(-x^2), 0, 1]` yields *0.746824132812427*.

NIntegral[<Function f>, <Variable t>, <Start variable-Value a>, <End variable-Value b>]

Computes the definite integral $\int_a^bf(t)\mathrm{d}t$ numerically.

Example:

`NIntegral[`*e*`^(-a^2), a, 0, 1]` yields *0.746824132812427*.

NRoot Command

NRoot[<Expression>, <N>]

Calculates the n^{th} root of a given expression.

Example:

- `NRoot[16, 4]` yields *2*.
- `NRoot[x^8, 2]` yields $(|x|)^4$.

NSolutions Command

NSolutions[<Equation>]

Finds a numeric solution for the given equation for the main variable.

Example:

`NSolutions[cos(x) = x]` yields *{0.7390851332151606}*.

NSolutions[<Equation>, <Variable>]

Finds a numeric solution to the given equation for the given unknown variable.

Example:

`NSolutions[a^4 + 34a^3 - 34, a]` yields *{0.9904738886662206}*.

Note: It is optional to give the starting point like *a=3*.

NSolutions[<List of Equations>, <List of Variables>]

Finds a numeric solution to the given set of equations for the given set of unknown variables.

Example:

`NSolutions[{π / x = cos(x - 2y), 2 y - π = sin(x)}, {x=3, y=1.5}]` yields *{3.141592651686591, 1.570796327746508}*.

Note: It is optional to give the starting point like *{x=3, y=1.5}*.

Note:

- π is obtaind by pressing .
- See also Solutions Command and NSolve Command.

NSolve Command

NSolve[<Equation>]

Finds a numeric solution for a given equation for the main variable.

Example:

`NSolve[cos(x) = x]` yields *{x = 0.7390851332151606}*.

NSolve[<Equation>, <Variable>]

Finds a numeric solution to an equation for the given unknown variable.

Example:

`NSolve[a^4 + 34a^3 - 34, a]` yields *{a = 0.9904738886662206}*.

Note: It is optional to give the starting point like *a=3*.

NSolve[<List of Equations>, <List of Variables>]

Finds a numeric solution to a set of equations for the given set of unknown variables.

Example:

`NSolve[{π / x = cos(x - 2y), 2 y - π = sin(x)}, {x=3, y=1.5}]` yields *{x = 3.141592651686591, y = 1.5707963277746508}*.

Note: It is optional to give the starting point like *{x=3, y=1.5}*.

Note:

- π is obtaind by pressing .
- See also Solve Command and NSolutions Command.

NextPrime Command

NextPrime[<Number>]

Returns the smallest prime greater than the entered number.

Example:

`NextPrime[10000]` yields *10007*.

Note:

See also PreviousPrime Command.

Numeric Command

Numeric[<Expression>]

Tries to determine a numerical approximation of the given expression. The number of decimals depends on the global rounding you choose in the Options Menu.

Example:

`Numeric[3 / 2]` yields *1.5*.

Numeric[<Expression>, <significant figures>]

Tries to determine a numerical approximation of the given expression, using the entered number of significant figures.

Example:

`Numeric[sin(1), 20]` yields *0.84147098480789650665*.

PreviousPrime Command

PreviousPrime[<Number>]

Returns the greatest prime smaller than the entered number.

Example:

`PreviousPrime[10000]` yields *9973*.

Note:

See also NextPrime Command.

RandomPolynomial Command

RandomPolynomial[<Degree d> , <Minimum for Coefficients min>, <Maximum for Coefficients max>]

Returns a randomly generated polynomial in *x* of degree *d*, whose (integer) coefficients are in the range from *min* to *max*, both included.

Example:

- `RandomPolynomial[0, 1, 2]` yields either *1* or *2* and
- `RandomPolynomial[2, 1, 2]` yields a random polynomial with a degree of two and only *1* and *2* as coefficients, for example $2x^2 + x + 1$.

RandomPolynomial[<Variable Var>, <Degree d> , <Minimum for Coefficients min>, <Maximum for Coefficients max>]

Returns a randomly generated polynomial in *Var* of degree *d*, whose (integer) coefficients are in the range from *min* to *max*, both included.

Example:

- `RandomPolynomial[a, 0, 1, 2]` yields either *1* or *2* and
- `RandomPolynomial[a, 2, 1, 2]` yields a random polynomial with a degree of two and only *1* and *2* as coefficients, for example $2a^2 + a + 1$.

Note: In both cases if *min* or *max* are not integers, *round(min)* and *round(max)* are used instead.

Rationalize Command

Rationalize[<Number>]

Creates the fraction of the given *Number*.

Example:

`Rationalize[3.5]` yields \(\frac{7}{2}\).

Note:

See also MixedNumber Command.

Real Command

Real[<Complex Number>]

Returns the real part of a given complex number.

Example:

`Real[17 + 3 ί]` yields *17*, the real part of the complex number *17 + 3 ί*.

Note:

- The complex ί is obtained by pressing .
- See also Imaginary Command.

RightSide Command

RightSide[<Equation>]

Returns the right-hand side of the equation.

Example:

`RightSide[x + 2 = 3x + 1]` yields $3x + 1$.

RightSide[{ <Equation> }]

Returns the right-hand sides of the equations.

Example:

`RightSide[{a^2 + b^2 = c^2, x + 2 = 3x + 1}]` yields $\{c^2, 3x + 1\}$.

Note:

See also LeftSide Command.

Solutions Command

Solutions[<Equation>]

Solves a given equation for the main variable and returns a list of all solutions.

Example:

`Solutions[x^2 = 4x]` yields *{4, 0}*, the solutions of $x^2 = 4x$.

Solutions[<Equation>, <Variable>]

Solves an equation for a given unknown variable and returns a list of all solution.

Example:

`Solutions[x * a^2 = 4a, a]` yields $\{\frac{4}{x},0\}$, the solutions of $x a^2 = 4a$.

Solutions[<List of Equations>, <List of Variables>]

Solves a set of equations for a given set of unknown variables and returns a list of all solutions.

Example:

- `Solutions[{x = 4 x + y , y + x = 2}, {x, y}]` yields *{{-1, 3}}*, the sole solution of $x = 4x + y$ and $y + x = 2$, displayed as $\begin{pmatrix} -1&3 \end{pmatrix}$.
- `Solutions[{2a^2 + 5a + 3 = b, a + b = 3}, {a, b}]` yields *{{0, 3}, {-3, 6}}*, displayed as $\begin{pmatrix} 0&3\\ -3&6 \end{pmatrix}$.

Note: See also Solve Command.

Solve Command

Solve[<Equation>]

Solves a given equation for the main variable and returns a list of all solutions.

Example:

`Solve[x^2 = 4x]` yields *{x = 4, x = 0}*, the solutions of $x^2 = 4x$.

Solve[<Equation>, <Variable>]

Solves an equation for a given unknown variable and returns a list of all solution.

Example:

`Solve[x * a^2 = 4a, a]` yields \(\{a = \frac{4}{x}, a = 0\}\), the solutions of $x a^2 = 4a$.

Solve[<List of Equations>, <List of Variables>]

Solves a set of equations for a given set of unknown variables and returns a list of all solutions.

Example:

- `Solve[{x = 4 x + y , y + x = 2}, {x, y}]` yields *{{x = -1, y = 3}}*, the sole solution of $x = 4x + y$ and $y + x = 2$, and
- `Solve[{2a^2 + 5a + 3 = b, a + b = 3}, {a, b}]` yields **'.**

Note: See also Solutions Command.

Substitute Command

Substitute[<expression>, <from>, <to>]

Substitutes *from* in *expression* with *to*.

Example:

`Substitute[(3 m - 3)^2 - (m + 3)^2, m, a]` yields $8 a^2 - 24 a$.

Substitute[<Expression>, <Substitution List>]

Substitutes in *expression* every variable of the list with the variable or number you choose for it.

Example:

`Substitute[2x + 3y - z, {x=a, y=2, z=b}]` yields $2a - b + 6$.

ToComplex Command

ToComplex[<Vector>]

Transforms a vector or point to a complex number in algebraic form.

Example:

`ToComplex[(3, 2)]` yields *3 + 2ί*.

Note:

- The complex ί is obtained by pressing .
- See also ToExponential Command, ToPoint Command and ToPolar Command.

ToExponential Command

ToExponential[<Complex Number>]

Transforms a complex number into its exponential form.

Example:

`ToExponential[1 + ί]` yields \(\sqrt{2}e^{\frac{i\pi}{4}}\).

Note:

- The complex ί is obtained by pressing .
- See also ToPoint Command, ToComplex Command and ToPolar Command.

ToPoint Command

ToPoint[<Complex Number>]

Transforms a complex number into a point.

Example:

`ToPoint[3 + 2ί]` yields (3, 2).

Note:

- The complex ί is obtained by pressing .
- See also ToComplex Command, ToExponential Command and ToPolar Command.

ToPolar Command

ToPolar[<Vector>]

Transforms a vector into its polar coordinates.

Example:

`ToPolar[{1, sqrt(3)}]` yields*(2;* $\frac{\pi}{3}$*)*, the polar coordinates of *(1,* $\sqrt{3}$*)*.

ToPolar[<Complex Number>]

Transforms a complex number into its polar coordinates.

Example:

`ToPolar[1 + sqrt(3) * ί]` yields*(2;* $\frac{\pi}{3}$*)*, the polar coordinates of *1* + $\sqrt{3}$ * *ί*.

Note:

- The complex ί is obtained by pressing .
- See also ToComplex Command, ToExponential Command and ToPoint Command.

nPr Command

nPr [<Number n>, <Number r>]

Returns the number of possible permutations of *r* elements out of a list of *n* elements.

Example:

`NPr[10, 2]` yields *90*.

Predefined Functions and Operators

To create numbers, coordinates, or equations using the Input Bar you may also use the following pre-defined functions and operations. Logic operators and functions are listed in article about Boolean values.

Note: The predefined functions need to be entered using parentheses. You must not put a space between the function name and the parentheses.

Operation / Function	Input
Addition	+
Subtraction	-
Multiplication	* or Space key
Scalar product	* or Space key
Vector product or determinant (see Points and Vectors)	⊗
Division	/
Exponentiation	^ or superscript (`x^2` or x^2)
Factorial	!
Parentheses	( )
x-coordinate	x( )
y-coordinate	y( )
Argument	arg( )
Conjugate	conjugate( )
Absolute value	abs( )
Sign	sgn( ) or sign()
Square root	sqrt( )
Cubic root	cbrt( )
Random number between 0 and 1	random( )
Exponential function	exp( ) or e^x
Logarithm (natural, to base e)	ln( ) or log( )
Logarithm to base 2	ld( )
Logarithm to base 10	lg( )
Logarithm of *x* to base *b*	log(b, x)
Cosine	cos( )
Sine	sin( )
Tangent	tan( )
Secant	sec()
Cosecant	cosec()
Cotangent	cot()
Arc cosine	acos( ) or arccos( )
Arc sine	asin( ) or arcsin( )
Arc tangent (returns answer between -π/2 and π/2)	atan( ) or arctan( )

Arc tangent (returns answer between -π and π) [1]	atan2(y, x)
Hyperbolic cosine	cosh( )
Hyperbolic sine	sinh( )
Hyperbolic tangent	tanh( )
Hyperbolic secant	sech( )
Hyperbolic cosecant	cosech( )
Hyperbolic cotangent	coth( )
Antihyperbolic cosine	acosh( ) or arccosh( )
Antihyperbolic sine	asinh( ) or arcsinh( )
Antihyperbolic tangent	atanh( ) or arctanh( )
Greatest integer less than or equal	floor( )
Least integer greater than or equal	ceil( )
Round	round( )
Beta function [2] B(a, b)	beta(a, b)
Incomplete beta function [3] B(x;a, b)	beta(a, b, x)
Incomplete regularized beta function [4] I(x; a, b)	betaRegularized(a, b, x)
Gamma function	gamma(x)
(Lower) incomplete gamma function [5] γ(a, x)	gamma(a, x)
(Lower) incomplete regularized gamma function [6]	gammaRegularized(a, x)
Gaussian Error Function	erf(x)

Example:

`Conjugate(17 + 3 * ί)` gives *-3 ί + 17*, the conjugated complex number of *17 + 3 ί*.

See Complex Numbers for details.

References

[1] http://en.wikipedia.org/wiki/Atan2
[2] http://mathworld.wolfram.com/BetaFunction.html
[3] http://mathworld.wolfram.com/IncompleteBetaFunction.html
[4] http://mathworld.wolfram.com/RegularizedBetaFunction.html
[5] http://mathworld.wolfram.com/IncompleteGammaFunction.html
[6] http://mathworld.wolfram.com/RegularizedGammaFunction.html

User interface

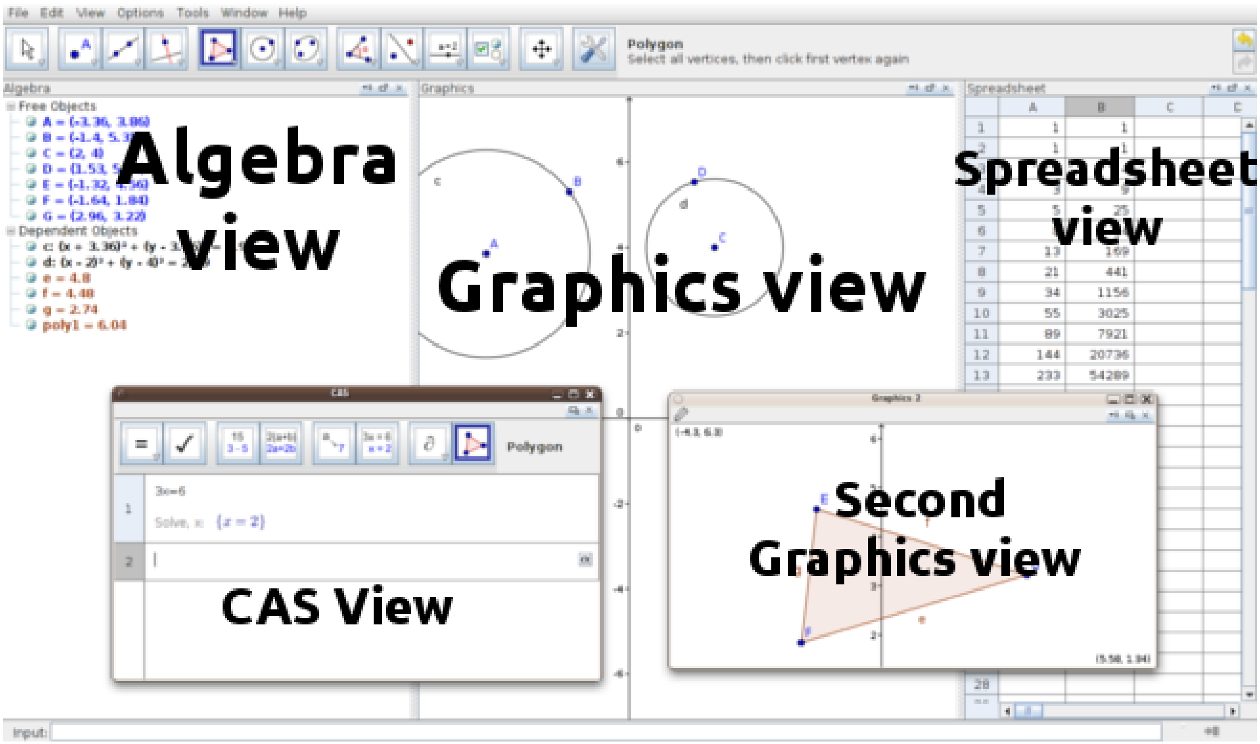
Views

GeoGebra provides three different views of mathematical objects: a Graphics View (available in two different windows), a numeric Algebra View, and a Spreadsheet View.

They allow you to display mathematical objects in three different representations: graphically (e. g., points, function graphs), algebraically (e. g., coordinates of points, equations), and in spreadsheet cells.

Thereby, all representations of the same object are linked dynamically and adapt automatically to changes made to any of the representations, no matter how they were initially created.

Moreover, a CAS View is available, which assists the user to do some calculations, both numerical and symbolic, and supports the main algebraic operations as well as derivation, integration, equations solving, differential equations solving and many others.



Styling Bar

Each view has its own Styling Bar which can be enabled via the Toggle Styling Bar button in the top right part of the view (next to close button). This styling bar offers quick access to formatting features. For more information see sections about styling bars of the particular views:

- Graphics View
- Algebra View
- Spreadsheet View
- CAS View

Graphics View

Using the construction tools available in the Toolbar you can do geometric constructions in the Graphics View with the mouse. Select any construction tool from the Toolbar and read the Toolbar Help (next to the Toolbar) in order to find out how to use the selected tool. Any object you create in the Graphics View also has an algebraic representation in the Algebra View.

Note: After activating the tool Move Tool you are able to move objects in the Graphics View by dragging them with the mouse. At the same time, their algebraic representations are dynamically updated in the Algebra View.
Every icon in the Toolbar represents a toolbox that contains a selection of similar construction tools. In order to open a toolbox, you need to click on the small arrow in the lower right corner of the Toolbar icon.

Hint: Construction tools are organized by the nature of resulting objects or the functionality of the tools. You will find tools that create different types of points in the Point Toolbox (default icon) and tools that allow you to apply geometric transformations in the Transformation Toolbox (default icon).
The Graphics View may include various types of grid and axes – see Customizing the Graphics View for details.

Styling Bar

You can find a button to toggle the styling bar in the upper right corner of the Graphics-View. Depending on the tool you select, there are shown different buttons to enhance your construction. By default you are able to *show/hide the coordinate axes* and the *grid* or change the point capturing. Additionally there are buttons *set color*, *set point style*, *set label style*, *set line style*, *set color and transparency*, *set text color*, *set font style to bold*, *set font style to italic* or *set text size*.

Second Graphics View

A second graphics view may be opened using the View Menu. If two Graphics Views are opened, one of them is always active (either it's being worked with and it has bold caption, or it is the last view that has been worked with). All visible objects created by Commands appear in the active graphics view. For each object you can specify in which Graphic View(s) it will be visible using the *Advanced* tab of the Properties Dialog.

Copy & Paste

Via Keybord Shortcut and (Mac OS: and) you can Copy and Paste selected object (except if they depend on the axes) into either the same window or into another. Copy & Paste will copy every ancestor of the selected objects but makes the non-selected objects invisible. If you copy objects depending on sliders into a new window, it will copy the slider (invisible) into the window, too. The pasted object is fixed when you click on the Graphics View. If the copied object depends at least one Point then it can snap onto existing points when pasted (but only the Point following the mouse pointer will do this).

Customizing the Graphics View

In order to adjust the visible part of the drawing pad in the Graphics View, you can drag the drawing pad by using tool Move Graphics View Tool and use the following ways of zooming:

- You may use the Zoom In Tool and Zoom Out Tool in order to zoom in the Graphics View. **Note:** The position of your click determines the center of zoom.
- You may use the scroll wheel of your mouse in order to zoom in the Graphics View.
- You may use keyboard shortcuts to zoom in (Ctrl +) and to zoom out (Ctrl -).
- After right clicking (Mac OS: Ctrl-click) on an empty spot on the drawing pad a Context Menu appears which allows you to Zoom.
- You may specify a zoom rectangle by right clicking (Mac OS: Cmd - click) on an empty spot in the Graphics View and dragging the mouse to the opposite corner of your desired zoom rectangle. Release the mouse button in order to finish the zoom rectangle, which will then automatically adjust to fill all the space in the Graphics View.

Note: To show or hide the axes and the grid, right click (Mac OS: Ctrl-click) on the drawing pad and select the corresponding items Axes or Grid from the appearing Context Menu.

Showing and hiding objects

In the Algebra View, the icon to the left of every object shows its current visibility state (shown or hidden). You may directly click on the little marble icon in order to change the visibility status of an object. For more information see visibility.

Customizing Coordinate Axes and Grid

The coordinate axes and grid can be customized using the Properties Dialog of the Graphics View. After right clicking (Mac OS: Ctrl-click) on the drawing pad, you can open this dialog window by selecting Graphics... from the appearing Context Menu of the Graphics View.

- On tab Basic, you can, for example, change the line style and color of the coordinate axes, and set the the ratio between the axes .
- Clicking on tabs xAxis and yAxis allows you to customize the axes individually, set the distance of the tickmarks, labeling, axes visibility, units and more. If you want the cross of the axes to be at point *(a,b)*, you can set *Cross at* parameter for xAxis to *b* and for yAxis to *a*. Option *Stick to edge* means that the line remains close to the bottom or left border of the screen. To draw only the part of the axis to the right or to the top of the axes intersection, you can select *Positive direction only*.
- On tab Grid, you can change the color and line style of the coordinate grid, set the distance and ratio for grid lines to a certain value, and the grid visibility. Three types of grid are available: Cartesian, polar and isometric.

Note:

Axes scaling is possible in every mode by pressing and holding the -key (PC: also -key) while dragging the axis. Range of the axes may be given dynamically, e.g. in Basic tab you can set X Min to x(A) and Y Min to y(A) to ensure the lower left corner of the view remains in point A. In such setting, the view cannot be zoomed.

Algebra View

Using the Input Bar you can directly enter algebraic expressions in GeoGebra. After hitting the Enter-key your algebraic input appears in the Algebra View while its graphical representation is automatically displayed in the Graphics View.

Example: The input *f(x) = x^2* gives you the function f in the Algebra View and its function graph in the Graphics View.

In the Algebra View, mathematical objects are organized as free and dependent objects. If you create a new object without using any other existing objects, it is classified as a free object. If your newly created object was created by using other existing objects, it is classified as a dependent object.

Hint: If you want to hide the algebraic representation of an object in the Algebra View, you may specify the object as an auxiliary object: Right click (Mac OS: Ctrl-click) on the corresponding object in the Algebra View and select Properties from the appearing Context Menu. On tab Basic of the Properties Dialog you may specify the object as an Auxiliary Object. By default, auxiliary objects are not shown in the Algebra View, but you can change this setting by selecting the item Auxiliary Objects from the View Menu.

Note that you are able to modify objects in the Algebra View as well: Make sure that you activate the Move Tool before you double click on a free object in the Algebra View. In the appearing text box you can directly edit the algebraic representation of the object. After hitting the Enter-key, the graphical representation of the object will automatically adapt to your changes.
If you double click on a dependent object in the Algebra View, a dialog window appears allowing you to Redefine the object.

GeoGebra also offers a wide range of commands that can be entered into the Input Bar. You can open the list of commands in the right corner of the Input Bar by clicking on the button Command. After selecting a command from this list (or typing its name directly into the Input Bar) you can press the F1-key to get information about the syntax and arguments required to apply the corresponding command.

Styling Bar

This Styling Bar contains two buttons.

Auxiliary objects

toggling this button shows or hides Auxiliary Objects.

Sort Objects by Type

when turned on, objects are sorted by type (e.g. Points, Lines, ...), otherwise they are divided among Free, Dependent and Auxiliary Objects.

Spreadsheet View

In GeoGebra's Spreadsheet View every cell has a specific name that allows you to directly address each cell. For example, the cell in column A and row 1 is named A1.

Note: These cell names can be used in expressions and commands in order to address the content of the corresponding cell.

In the spreadsheet cells you can enter not only numbers, but all types of General and Geometrical Objects that are supported by GeoGebra (e. g., coordinates of points, functions, commands). If possible, GeoGebra immediately displays the graphical representation of the object you entered in a spreadsheet cell in the Graphics View as well. Thereby, the name of the object matches the name of the spreadsheet cell used to initially create it (e. g., A5, C1).

Note: By default, spreadsheet objects are classified as auxiliary objects in the Algebra View. You can show or hide these auxiliary objects by selecting "Auxiliary Objects" from the View Menu.

Relative cell names

By default, if you copy content from one cell to another, all references are changed accordingly to the target position.

Example: Let A1=1,A2=2. In B1 put (A1,A1). By copying B1 to B2 (either via , or by dragging the cell corner) you get (A2,A2) in B2.

To prevent this behaviour, you can insert $ before the column and/or row of the referenced cell.

Note: On Mac the copy & paste shortcuts are and

Getting data into the spreadsheet

Besides manually adding entries to the spreadsheet, you may use FillCoumn, FillRow or FillCells. Also see section tracing to spreadsheet.

Using spreadsheet data in other views

You may process the spreadsheet data by selecting multiple cells, right-clicking and choosing an item from the "Create" submenu of appearing Context Menu.

CAS View

Basic input

- : evaluate input
- : check input but do no evaluate input, e.g. *b* + *b* stays *b* + *b*. Note that assignments are always evaluated, e.g. a := 5
- In an empty row type
  - bar for previous output
  - `)` for previous output in parentheses
  - `=` for previous input
- Suppress output with a semicolon at the end of your input, e.g. `a := 5;`

Toolbar

- Clicking a button in the toolbar applies a command to the currently edited row
- You can select part of the input text to only apply the operation to this selected part

Variables

Assignments & Connection with GeoGebra

- Assignments use the := notation, e.g. `b := 5`, `a(n) := 2n + 3`
- To free up a variable name again, use `Delete[b]` or `b :=`
- Variables and functions are always shared between the CAS view and GeoGebra if possible. If you define `b:=5` in the CAS view, then you can use *b* in all of GeoGebra. If you have a function `f(x)=x^2` in GeoGebra, you can also use this function in the CAS view.

Row References

You can refer to other rows in the CAS view in two ways

- Static row references insert text from another row, so your input is changed.
  - `#` inserts the previous output
  - `#5` inserts the output of row 5
- Dynamic row references use text from another row, but don't change your input.
  - `$` inserts the previous output
  - `$5` inserts the output of row 5

Equations

- Equations are written using the simple Equals sign, e.g. `3x + 5 = 7`
- You can perform arithmetic operations on equations, e.g. `(3x + 5 = 7) - 5` subtracts 5 from both sides of the equation. This is useful for manual equation solving.
- `LeftSide[3x + 5 = 7]` returns $3x + 5$ and `RightSide[3x + 5 = 7]` returns *7*

Commands and Tools

For a complete list of commands and tools see CAS Commands and CAS tools.

Construction Protocol

You can access the interactive Construction Protocol by selecting item Construction Protocol - Show from the View menu. The Construction Protocol is a table that shows all construction steps, allowing you to redo a construction step by step using the Navigation Bar at the bottom of the Construction Protocol dialog.

Navigating and Modifying the Construction Protocol

You may use the keyboard to navigate in the Construction Protocol:

- Use the ↑ up arrow of your keyboard to go to the previous construction step.
- Use the ↓ down arrow of you keyboard to go to the next construction step.
- Use the Home key to go to the beginning of the Construction Protocol.
- Use the End key to go to the end of the Construction Protocol.
- Use the Delete key in order to delete the selected construction step.

  Note: This may also affect other objects that depend on the selected object/construction step.

You may also use the mouse in order to navigate in the Construction Protocol:

- Double click a row to select a construction step.
- Double click the header of any column to go to the beginning of the Construction Protocol.
- Drag and drop a row to move a construction step to another position in the Construction Protocol.

Note: This is not always possible due to the dependencies between different objects.

- Right click a row to open the Context Menu for the object of this construction step.

Note: You can insert construction steps at any position. Select the construction step below you would like to insert a new construction step. Leave the Construction Protocol window open while you create a new object. This new construction step is immediately inserted into the selected position of the Construction Protocol.

Using the column Breakpoint in the View menu of the Construction Protocol window, you can define certain construction steps as Breakpoints. This allows you to group several objects together. When navigating through your construction using the Navigation Bar, groups of objects are shown at the same time.

Note: You may switch the different columns of the Construction Protocol on and off by using the *Styling Bar* of the Construction Protocol window.

Exporting the Construction Protocol as a Webpage

GeoGebra allows you to export the Construction Protocol as a webpage. First, you need to open the Construction Protocol using the View menu. Then open the File menu of the appearing Construction Protocol window and select item Export as Webpage.

In the export window of the Construction Protocol you can enter Title, Author, and a Date for the construction and choose whether or not you want to include a picture of the Graphics View and the Algebra View. In addition, you can also choose to export a Colorful Construction Protocol. This means that objects in the Construction Protocol will match the color of the corresponding objects in the construction.

> **Note:** The exported HTML file can be viewed with any Internet browser (e. g. Firefox, Internet Explorer) and edited with many text processing systems (e. g. OpenOffice Writer).

Input Bar

Input bar is by default located in the bottom of GeoGebra window. You can show it or hide it via View Menu.

It allows you to create and redefine mathematical objects

- directly, using their algebraic representations (e. g., values, coordinates, equations). This representation is shown in the Algebra View. See Geometric Objects and General Objects for details.
- using Commands.

Note: Always press after typing algebraic input into the Input Bar.

Note: Pressing at any time toggles the focus between the Input Bar and the Graphics View. This allows you to enter expressions and commands into the Input Bar without having to click on it with the mouse first.

Example: Typing `A=(1,1)` creates free point A with coordinates (1,1). Typing `A=Midpoint[(2,0),(4,0)]` redefines A: it becomes dependent point.

Display Input Bar History

After placing the cursor in the Input Bar you can use the up and down arrow keys of your keyboard in order to navigate through prior input step by step.

Insert Name, Value, or Definition of an Object into the Input Bar

- **Insert the name of an object**: Activate Move Tool and select the object whose name you want to insert into the Input Bar. Then, press on your keyboard.

Note: The name of the object is appended to any expression you typed into the Input Bar before pressing .

- **Insert the value of an object**: There are two ways of inserting an object's value (e. g., (1, 3), 3x − 5y = 12) into the Input Bar.
  - Right click (Mac OS: Ctrl-click) on the object and select item Copy to Input Bar from the appearing Context Menu.
  - Activate Move Tool and select the object whose value you want to insert into the Input Bar. Then, press on your keyboard.

Note: The value of the object is appended to any expression you typed into the Input Bar before pressing .

- **Insert the definition of an object**: There are two ways of inserting an object's definition (e. g., A = (4, 2), c = Circle[A, B]) into the Input Bar.

- Alt click on the object to insert the object's definition and delete whatever input might have been in the Input Bar before.
- Activate  Move Tool and select the object whose definition you want to insert into the Input Bar. Then, press on your keyboard.

Note: The definition of the object replaces any expression you typed into the Input Bar before pressing .

Menubar

The Menubar is always situated in the top part of GeoGebra window. For applets it can be switched on and off during export. It contains following menus:

- File Menu
- Edit Menu
- View Menu
- Options Menu
- Tools Menu
- Window Menu
- Help Menu

Toolbar

By default Toolbar is located right under Menubar. Each view which was opened as separate window has its own toolbar in the upper part. Each toolbar is divided into toolboxes, each of which can contain one or more Tools. Toolbar of the main window can be moved down by switching off the *Toolbar On Top* checkbox in *Toolbar* submenu of View Menu.

Toolbar Help

Toolbar help is located in the right part of the toolbar and contains information on using the currently selected tool. If you cannot see it, you have to resize the window. When you click it, web page with help for the selected tool opens in a browser.

Toolbars for different views

GeoGebra has three toolbars: one for Graphics View, one for Spreadsheet View and one for CAS View. Once you start using another view within the GeoGebra window, the toolbar changes. If you open a Spreadsheet View or Cas View in separate window, it will have its toolbar attached.

Customizing the Toolbar

All three Toolbars can be customized by selecting *Customize Toolbar…* from the Tools Menu. From the drop-down list select a toolbar you want to edit. To remove a tool or toolbox from toolbar, select it in the list on the left hand side of the appearing dialog window and click button *Remove*. To add a tool, select it in the right list. If you want to add it to a new toolbox, select toolbox left to the desired position and click *Insert*. To add it to existing toolbox, open the toolbox in the left list and select a tool above desired position. To move tool from one toolbox to another you have to remove it and add it.

Note: You can restore the default Toolbar by clicking on the button *Restore Default Toolbar* in the left lower corner of the dialog window.

Appearence of toolbar in Dynamic Worksheets can be set using customToolBar parameter.

Navigation Bar

GeoGebra offers a Navigation Bar that allows you to navigate through the construction steps of a prepared GeoGebra file. Select item Construction Protocol... > Navigation Bar for Construction Steps in the View Menu in order to display the Navigation Bar at the bottom of the Graphics View.

The Navigation Bar provides a set of navigation buttons and displays the number of construction steps (e. g., 2 / 7 means that currently the second step of a total of 7 construction steps is displayed):

- button: go back to step 1
- button: go back step by step
- button: go forward step by step
- button: go to the last step
- Play: automatically play the construction step by step

Note: You may change the speed of this automatic play feature using the text box to the right of the Play button.

- Pause: pause the automatic play feature

Note: This button only appears after you click on the Play button.

button: This button opens the Construction Protocol.

File Menu

New Window

Keyboard shortcut: (MacOS:)

This menu item opens a new GeoGebra window that uses the default settings of the GeoGebra user interface.

Note: If you change and save some of these settings, the new GeoGebra window will open using your customized settings.

New

This menu item opens a new and empty user interface in the same GeoGebra window. You are asked if you would like to save the existing construction before opening the new user interface.

Note: The new user interface adopts the settings used for the prior construction. For example, if the coordinate axes were hidden before selecting the menu item New, the axes will be hidden in the new user interface as well.

Open...

Keyboard shortcut: (MacOS:)

This menu item allows you to open a GeoGebra worksheet (file name extension GGB), GeoGebra tool (file name extension GGT) or dynamic worksheet (HTM or HTML file produced by GeoGebra) that is saved on your computer.

Note: In order to open a GeoGebra file you can also drag it with the mouse to the GeoGebra window and drop it there.

Open Webpage...

This menu item allows you to open a Webpage containing a GeoGebra applet, just entering the Webpage address in the appearing dialog.

Open Recent (submenu)

Lists up to eight recently opened files.

Save

Keyboard shortcut: (MacOS:)

This menu item allows you to save your current construction as a GeoGebra file (file name extension GGB) on your computer.

Note: If the file was saved before, this menu item overwrites the old file by using the same file name.

Save as...

This menu item allows you to save your current construction as a GeoGebra file (file name extension GGB). You will be asked to enter a new name for your GeoGebra file before it is saved on your computer.

Share

Lets you upload your worksheet directly to GeoGebraTube [1], see also Dynamic Worksheet as Webpage (html)... .

Export (submenu)

Offers several export possibilities:

- Dynamic Worksheet as Webpage (html)...
- Graphics View as Picture (png, eps)…
- Graphics View to Clipboard
- ...and others

Print Preview

Keyboard shortcut: (MacOS:)

This menu item opens the Print Preview window for the Graphics View. You may specify Title, Author, Date and the Scale of your printout (in cm).

Note: Press the Enter-key after you made a change in order to update the preview of your printout.

Close

Keyboard shortcut: (MacOS:)

This menu item closes the GeoGebra window. If you didn't save your construction prior to selecting Close, you are asked if you would like to do so.

References

[1] http://www.geogebratube.org/

Edit Menu

Undo

Keyboard shortcut: (MacOS:)

This menu item allows you to undo your activities step by step.

Note: You can also use the Undo button to the right of the Toolbar.

Redo

Keyboard shortcut: (MacOS:)

This menu item allows you to redo your activities step by step.

Note: You can also use the Redo button to the right of the Toolbar.

Object Properties…

Keyboard shortcut: (MacOS:)

This menu item opens the Properties Dialog which allows you to modify the properties of all objects used in the GeoGebra file.

Select All

Keyboard shortcut: (MacOS:)

This menu item allows you to select all objects used in your construction.

Select Current Layer

Keyboard shortcut: (MacOS:)

This menu item allows you to select all objects that are on the same layer as a selected object.

Note: You need to select one object that lies on the desired layer prior to using this menu item.

Select Descendants

Keyboard shortcut: (MacOS:)

This menu item allows you to select all objects that depend on the selected object.

Note: You need to select the parent object prior to using this menu item.

Select Ancestors

Keyboard shortcut:

This menu item allows you to select all objects that are ancestors of the selected object, meaning all objects the selected one depends on.

Note: You need to select the dependent object prior to using this menu item.

Invert Selection

Keyboard shortcut:

Deselects selected objects and vice versa.

Show / Hide Objects

Keyboard shortcut:

Changes visibity of selected objects.

Show / Hide Labels

Keyboard shortcut:

Shows hidden labels for selected objects and hides the shown ones.

Graphics View to Clipboard

Keyboard shortcut: (MacOS:)

This menu item copies the Graphics View to your computer's clipboard. Afterwards, you can easily paste this picture into other documents (e. g., word processing document).

Delete

Keyboard shortcut:

This menu items allows you to delete selected objects and their dependent objects.

Note: You need to select the objects you want to delete first (e. g., use a selection rectangle).

View Menu

Axes

This menu item allows you to show or hide the coordinate axes in the Graphics View.

Grid

This menu item allows you to show or hide the grid in the Graphic View.

Graphics

Keyboard shortcut: (MacOS:)

This menu item allows you to show or hide the Graphics View.

Algebra

Keyboard shortcut: (MacOS:)

This menu item allows you to show or hide the Algebra View.

Spreadsheet

Keyboard shortcut: (MacOS:)

This menu item allows you to show or hide the Spreadsheet View.

CAS

Keyboard shortcut: (MacOS:)

This menu item allows you to show or hide the CAS View.

Graphics 2

Keyboard shortcut: (MacOS:)

This menu item allows you to show or hide a second Graphic View.

Construction Protocol…

This menu item opens the Construction Protocol dialog.

Keyboard

This menu item allows you to show or hide the Virtual Keyboard, that you can use with a mouse, and contains the standard keyboard characters, as well as the most used mathematical symbols and operators.

Input Bar

This menu item allows you to show or hide the Input Bar and the Command List at the bottom of the GeoGebra window.

Toolbar

This menu item allows you to show or hide the Toolbar and decide its position at the top or the bottom of the GeoGebra window.

Navigation Bar for Construction Steps

This menu item allows you to show or hide a bar, designed for an easy navigation through the steps of a GeoGebra construction.

Refresh Views

Keyboard shortcut: (MacOS:)

This menu item allows you to repaint all views on screen.

Note: You can use this menu item to delete any traces of points or lines in the Graphics View.

Recompute All Objects

Keyboard shortcut:

This menu item recomputes all objects used in your GeoGebra file.

Note: You can use this menu item to create new random numbers if you used any in your GeoGebra file.

Perspectives

In the *Perspectives-Menu* you can easily switch between different views, without selecting each individually. You can choose between 5 different standard perspectives:

- *Algebra & Graphics*: The Algebra View and the Graphics View with axes are shown.
- *Basic Geometry*: Only the Graphics View without axes or grid is displayed.
- *Geometry*: Only the Graphics View with grid is shown.
- *Spreadsheet & Graphics*: The Spreadsheet View and the Graphics View are displayed.
- *CAS & Graphics*: The CAS View and the Graphics View are displayed.

It is also possible to create your personal *perspective*. If you want to save the current perspective go to *Perspectives - Save Current Perspective*. Then you have to type a name and click *OK*. You can delete your perspective by clicking *Perspectives - Manage Perspectives*.

Options Menu

Global options may be changed in the menu Options.

Note: To change object settings, please use the Context Menu and Properties Dialog.

Algebra Descriptions

You can set how will objects be represented in Algebra View with this item. There are three possibilities:

Value

show current value of the object.

Definition

show user-friendly description of the object, e.g. "Intersection of *a* and *b*."

Command

show the command that was used to create the object, e.g. "Intersect[a,b]".

Point Capturing

This menu item determines if the point capturing is Off or if points are captured by the grid (item Snap to Grid) or constrained on the grid (item Fixed to Grid)

Note: Option Automatic turns the point capturing On when the grid or the coordinate system are shown and turns it Off if they are hidden.

Rounding

This menu item allows you to set the number of decimal places or significant figures displayed on screen.

Labeling

You can specify whether the label of a newly created object should be shown or not. You can choose between the settings All New Objects, No New Objects, New Points Only, and Automatic.

Note: The setting Automatic shows the labels of newly created objects if the Algebra View is shown.

Font Size

This menu item determines the font size for labels and text in points (pt).

Note: If you are using GeoGebra as a presentation tool, increasing the font size makes it easier for your audience to read text, labels, and algebraic input you are using.

Language

GeoGebra is multilingual and allows you to change the current language setting. This affects all input including command names and all output.

Note: No matter which language was selected, the globe icon will lead you back to the language menu. All language names are always displayed in English.

Settings ...

This menu item opens the Settings Dialog.

Note: You can also open this dialog window by right clicking (Mac OS: Ctrl-click) on the Graphics View or Spreadsheet View and selecting *Graphics* ... and *Spreadsheet Options* respectively.

Save Settings

GeoGebra remembers your favorite settings (e. g., settings in the Options menu, current Toolbar and Graphics View settings) if you select Save settings in the Options menu.

Restore Default Settings

You can restore the default settings of GeoGebra using this menu item.

Tools Menu

Create New Tool…

Based on an existing construction you can create your own tools in GeoGebra. After preparing the construction of your tool, choose Create new tool in the Tools Menu. In the appearing dialog you can specify the output and input objects of your tool and choose names for the Toolbar icon and corresponding command.

Note: Your tool can be used both with the mouse and as a command in the Input Bar. All tools are automatically saved in your GGB construction file.

Manage Tools…

Using the Manage tools dialog you can delete a tool or modify its name and icon. You can also save selected tools to a GeoGebra Tools File (GGT). This file can be used later on (File menu, Open) to load the tools into another construction.

Note: Opening a GGT file doesn’t change your current construction, but opening a GGB file does.

Customize Toolbar…

Opens Customize Toolbar Dialog.

Window Menu

New Window

Keyboard shortcut: (MacOS:)

See File Menu > New Window.

List of GeoGebra windows

If you have more than one GeoGebra window open, this menu item allows you to switch between these different windows.

Note: This might be helpful when you are using GeoGebra as a presentation tool and want to have several GeoGebra files open at the same time as well as to toggle between them.

Help Menu

Note: Following four menu items work only provided you have access to the internet. If you want to access help on a computer that is not connected, please download the PDF version [1] . Instead of reading tutorials you might download the GeoGebra Introductory Book [2].

Help

This menu item opens the HTML-version of the GeoGebra help (the Manual part of GeoGebraWiki) in your browser.

Tutorials

This menu item opens the tutorial part of GeoGebraWiki in your browser.

GeoGebra Forum

This menu item opens the GeoGebra User Forum [3] in your default web browser. You can post and answer GeoGebra-related questions and problems in the GeoGebra User Forum.

www.geogebra.org

This menu item opens the GeoGebra webpage [1] in your default web browser.

About / License

This menu item opens a dialog window that gives you information about the license of GeoGebra and gives credit to people who support the GeoGebra project by contributing in many different ways (e. g., programming, translations).

References

[1] http://www.geogebra.org/help/docuen.pdf
[2] http://www.geogebra.org/cms/en/help

Context Menu

The Context Menu provides a quick way to change the behavior or advanced properties of an object. Right click (Mac OS: Ctrl-click) on an object in order to open its Context Menu. For example, it allows you to change the object's algebraic notation (e. g., polar or Cartesian coordinates, implicit or explicit equation) and to directly access features like rename, Delete, Trace On, Animation On, or Copy to Input Bar.

Note: If you open the Context Menu for a point in the Graphics View, it gives you the option Trace to Spreadsheet (only if the Spreadsheet View is active). Once selected, this feature allows you to record the coordinates of the point in the Spreadsheet View if it is moved.
Note: Selecting Properties… in the Context Menu opens the Properties Dialog, where you can change the properties of all objects used.

Customize the Settings

GeoGebra allows you to change and save settings using the Options Menu. For example, you may change the Angle Unit from Degree to Radians, or change the Point Style, Checkbox Size, and Right Angle Style. In addition, you may change how Coordinates are displayed on screen and which objects are labeled (Labeling).

Please see the section about the Options menu for more information.

You can save your customized settings by selecting item Save Settings from the Options menu. After doing so, GeoGebra will remember your customized settings and use them for every new GeoGebra file you create.

Note: You may restore the default settings by selecting Restore Default Settings from the Options menu.

Note: If you use GeoGebra as a presentation tool, you might want to increase the Font Size (Options menu) so your audience can easily read text and labels of objects.

Export Graphics Dialog

This dialog is accessible via Export submenu of File Menu (item Graphics View as Picture (png, eps)….)

Keyboard shortcut: (Mac OS:)

This dialog allows you to save GeoGebra Graphics View as a picture file on your computer. In the appearing dialog window, you can select the picture file Format, change the Scale (in cm) and Resolution (in dpi) of the picture, and set the image as Transparent.

Note: If you create Points called Export_1 and Export_2 then these will define the rectangle that is exported, otherwise just the visible Graphics View is exported

When exporting the Graphics View as a picture you can choose out of the following formats:

PNG – Portable Network Graphics

> This is a pixel graphics format. The higher the resolution (dpi), the better the quality (300dpi will usually suffice). PNG graphics should not be scaled subsequently to avoid a loss of quality.
>
> PNG graphic files are well suited for the use on web pages (HTML) and in word processing documents.
>
> **Note:** Whenever you insert a PNG graphic file into a word processing document (menu Insert, Image from file) make sure that the size is set to 100 %. Otherwise the given scale (in cm) would be changed.

EPS – Encapsulated Postscript

> This is a vector graphics format. EPS pictures may be scaled without loss of quality. EPS graphic files are well suited for the use with vector graphics programs (e. g., Corel Draw) and professional text processing systems (e. g., LaTeX).
>
> The resolution of an EPS graphic is always 72dpi. This value is only used to calculate the true size of an image in centimeters and has no effect on the image's quality.
>
> **Note:** The transparency effect with filled polygons or conic sections is not possible with EPS.

PDF – Portable Document Format

> (see EPS format above)
>
> **Note:** In SVG and PDF export you have the option to export text as editable text or shapes. This stores the text either as text (this lets you edit the text in e. g., InkScape) or as Bézier curves (this guarantees that the text looks the same even if the correct font is not installed).

SVG – Scalable Vector Graphic

> (see EPS format above)

EMF – Enhanced Metafile

> (see EPS format above)

Export Worksheet Dialog

GeoGebra allows you to create interactive webpages, so called Dynamic Worksheets, from your files. In the File Menu, you need to select item *Export*, then click on item *Dynamic Worksheet as Webpage (html)*. This opens the export dialog window for Dynamic Worksheets.

Upload to GeoGebraTube

Under this tab you can enter a title for your construction, a text above and below the construction (e. g. a description of the construction and some tasks), and then it to GeoGebraTube [1].

Note: When you upload a file to GeoGebraTube, you will be asked to create an account and/or login first.

Export as (html) Webpage

If you want to create an html file on your computer, see Export as html Webpage for details.

Properties Dialog

The Properties Dialog allows you to modify properties of objects (e. g., size, color, filling, line style, line thickness, visibility) as well as automate some object actions using Javascript or GeoGebra Script.

You can open the Properties Dialog in several ways:

- Right click (Mac OS: Ctrl-click) on an object and select Properties… from the appearing Context Menu.
- Select item Properties from the Edit Menu.
- Select the Move Tool and double click on an object in the Graphics View. In the appearing Redefine Dialog window, click on the button Properties….

In the Properties Dialog objects are organized by types (e. g., points, lines, circles) in the list on the left hand side, which makes it easier to handle large numbers of objects. You need to select one or more objects from this list in order to change its/their properties.

Note: By clicking on a heading in the list of objects (e. g., Point) you can select all objects of this type and therefore, quickly change the properties for all these objects.

You can modify the properties of selected objects using the tabs on the right hand side (e. g., Basic, Color, Style, Algebra, Advanced, Scripting).

Note: Depending on the selection of objects in the list, a different set of tabs may be available.

Close the Properties Dialog when you are done with changing properties of objects.

Redefine Dialog

Redefining objects is a very versatile tool to change a construction. Please note that this may also change the order of the construction steps in the Construction Protocol.

Note: The redefined element can only depend on elements defined before -- you may need to change order of the elements in Construction Protocol.

In GeoGebra, an object may be redefined in different ways:

- Select Move Tool and double click on any object in the Algebra View.
  - For free objects an editing field is opened allowing you to directly change the algebraic representation of the object. Hit the Enter-key in order to apply these changes.
  - For dependent objects the Redefine dialog is opened allowing you to redefine the object.
- Select Move Tool and double click on any object in the Graphics View. This opens the Redefine dialog and allows you to redefine the object.
  - Change any object by entering its name and the new definition into the Input Bar.
  - Open the Properties Dialog and change the definition of an object on tab Basic.

Note: Fixed objects cannot be redefined. In order to redefine a fixed object, you need to free it first using tab Basic of the Properties Dialog.

Examples

Example: In order to place a free point A on an existing line h, you first need to double click on the point A to open the Redefine dialog window. Then, enter the command `Point[h]` in the appearing text field and press the Enter-key. To remove point A from this line and make it free again, you need to redefine it to some free coordinates like (1, 2).

Example: Another example is the conversion of a line h through two points A and B into a segment. Open the Redefine dialog for line h and enter the command `Segment[A, B]` in the appearing text field.

Tool Creation Dialog

First, create the construction your tool should be able to create later on. In the Tools menu, click on Create New Tool in order to open the corresponding dialog box. Now you need to fill in the three tabs Output Objects, Input Objects, and Name and Icon in order to create your custom tool.

Example: Create a Square-tool that creates a square whenever you click on two existing points or on two empty spots in the Graphics View.

- Construct a square starting with two points A and B. Construct the other vertices and connect them with the tool Polygon to get the square poly1.
- Select Create New Tool in the Tools menu.
- Specify the Output Objects: Click on the square or select it from the drop down menu. Also, specify the edges of the square as Output Objects.
- Specify the Input Objects: GeoGebra automatically specifies the Input Objects for you (here: points A and B). You can also modify the selection of input objects using the drop down menu or by clicking on them in your construction.
- Specify the Tool Name and Command Name for your new tool.

Note: The Tool Name will appear in GeoGebra Toolbar, while the Command Name can be used in GeoGebra Input Bar.

- You may also enter text to be shown in the Toolbar Help.
- You can also choose an image from you computer for the Toolbar icon. GeoGebra resizes your image automatically to fit on a Toolbar button.

Note: Outputs of the tool are not moveable, even if they are defined as `Point[<Path>]`. In case you need moveable output, you can define a list of commands and use it with Execute Command.

Keyboard Shortcuts

Key	Shortcut	Action
A		Select All
A		View Algebra Window
A		alpha α
B		beta β
C		Copy (spreadsheet) Ctrl-Alt-C copies values (spreadsheet)
C		Graphics View to clipboard
D		Toggle value/definition/command
D		delta δ
E		Open properties
E		Euler *e*
F		Refresh Views
F		phi φ
G		gamma γ
I		imaginary unit í
J		Select descendants
J		Select ancestors
L		Select current layer
L		lambda λ
M		Clipboard Export to Moodle/LMS/VLE etc
M		mu μ
N		New Window
O		Open
O		degree symbol ° (also in slider dialog for min, max, increment)
P		Print Preview
P		Export as picture (png, eps, etc.)
P		pi π (also in slider dialog for min, max, increment)
Q		Select descendants (deprecated)
Q		Select ancestors (deprecated)
R		Recompute all objects (including random numbers)
S		Save
S		View spreadsheet
S		sigma σ
T		Export as PSTricks
T		theta θ
V		Paste (spreadsheet)
W		Close (MacOS)

W		Export Dynamic Worksheet
W		omega ω
Y		Redo
Z		Undo
0		to the power of 0
1		Standard font size, line thickness, and point size
1		to the power of 1
2		Increase font size, line thickness, and point size
2		to the power of 2
3		Black/white mode
3		to the power of 3
4		to the power of 4
5		to the power of 5
6		to the power of 6
7		to the power of 7
8		to the power of 8
9		to the power of 9
-		Decrease selected slider/number Move selected point along curve
-		Zoom out(hold Alt as well for accelerated zoom)
-		minus-or-plus ∓
+		Increase selected slider/number Move selected point along curve
+		Zoom in (hold Alt as well for accelerated zoom)
+		plus-or-minus ±
=		Increase selected slider/number Move selected point along curve
=		Zoom in (hold Alt as well for accelerated zoom)
=		not-equal-to ≠
<		less-than-or-equal-to ≤
, (comma)		less-than-or-equal-to ≤
>		greater-than-or-equal-to ≥
. (period)		greater-than-or-equal-to ≥
F1		Help
F2		Start editing selected object
F3		copy definition of selected object to the Input Bar
F4		copy value of selected object to the Input Bar
F5		copy name of selected object to the Input Bar
F9		Recompute all objects (including random numbers)(MacOS: Cmd-R)
Enter		Toggle input between Graphics View and Input Bar

Left Click	Left Click	(current mode)
Left Click	+Left Click	copy definition to input bar
Left Click	+Left Drag	create list of selected objects in input bar
Right Click	Right click in Graphics View	Fast drag mode (drag on object) Zoom (drag not on object) Open menu (click on object) Open Axes and Grid menu (click not on object)
Right Click	Right click in Graphics View	+Right click in Graphics View
Right Click	+ Right Drag	Zooms without preserving the aspect ratio
Scroll Wheel	Scroll Wheel	Zoom in / out (Application)
Scroll Wheel	+Scroll Wheel	Zoom in / out (Applet)
Scroll Wheel	+Scroll Wheel	Accelerated zoom in / out
Delete		Delete current selection
Backspace		Delete current selection
Up arrow		Increase selected slider/number Move selected point up Go to older entry in Input Bar history Go up in construction protocol
Up arrow		x10 speed multiplier Spreadsheet: go to top of current block of cells (or go up to next defined cell)
Up arrow		x0.1 speed multiplier
Up arrow		x100 multiplier
Right arrow		Increase selected slider/number Move selected point right Go up in construction protocol
Right arrow		x10 speed multiplier Spreadsheet: go to right of current block of cells (or go right to next defined cell)
Right arrow		x0.1 speed multiplier
Right arrow		x100 multiplier
Left arrow		Decrease selected slider/number Move selected point left Go down in construction protocol
Left arrow		x10 speed multiplier Spreadsheet: go to left of current block of cells (or go left to next defined cell)
Left arrow		x0.1 speed multiplier
Left arrow		x100 multiplier
Down arrow		Decrease selected slider/number Move selected point down Go to newer entry in Input Bar history Go down in construction protocol
Down arrow		x10 speed multiplier Spreadsheet: go to bottom of current block of cells (or go down to next defined cell)

Down arrow		x0.1 speed multiplier
Down arrow		x100 multiplier
Home		Go to first item in construction protocol Spreadsheet: go to top left
PgUp		Go to first item in construction protocol
End		Go to last item in construction protocol Spreadsheet: go to bottom right
PgDn		Go to last item in construction protocol

In addition, use Alt-Shift (Mac OSX Ctrl-Shift) to get upper-case Greek characters.

Note that on Mac OSX, instead of pressing Alt to get the Greek and mathematical characters, you must use Ctrl.

Options Dialog

This dialog is available via the *Settings...* item in Options Menu. It is divided into five parts: Defaults, Graphics, Spreadsheet, CAS and Advanced.

Defaults

This part of the dialog lets you define properties of newly created objects. You can set properties for each object type separately, for points there are five subtypes. The way properties are set is similar to Properties Dialog

Graphics

Allows you to set the zoom of graphics view, its axes and grid. See Customizing Coordinate Axes and Grid for details.

Spreadsheet

Allows you to show or hide the inputbar, gridlines, column/row header and scrollbars. You can also enable using buttons and checkboxes.

CAS

Allows you to define a timeout for the CAS in seconds.

Advanced

- *Virtual Keyboard*: You can set the *virtual keyboard language* and the width/height of the virtual keyboard.
- *Tooltips*: You can set the *tooltip language* and a timeout for tooltips.
- *Language*: You can use digits and point names specified for your language.
- *Perspectives*: Here you can manage the perspectives of GeoGebra.
- *Angle Unit*: Switch between *Degree* and *Radians*
- *Continuity*: If Continuity is *On*, GeoGebra tries to set new calculated points near the original ones.
- *Default Point Style*: Set the point style, that is showed in the Graphics-View.
- *Checkbox Size*: Switch between *regular* and *large* checkboxes.
- *Right Angle Style*: Choose the symbol for a right angle.
- *Coordinates*: Define how coordinates are displayed.

- *Miscellaneous*: Here you can enable scripting, use Java fonts, etc.

Virtual Keyboard

The Virtual Keyboard is a semi-transparent keyboard that is displayed on the screen when the corresponding menu item is selected.

It contains the standard keyboard characters, as well as the most used mathematical symbols and operators, and can be used with a mouse or other pointing devices.

This makes the Virtual Keyboard particularly useful when using GeoGebra for presentations or with multimedia interactive whiteboards.

Tool Manager Dialog

You can save your custom tools so you can reuse them in other GeoGebra constructions. In the Tools Menu, select Manage Tools to open this dialog. Then, select the custom tool you want to save from the appearing list. Click on button Save As… in order to save your custom tool on your computer.

Note: User defined tools are saved as files with the file name extension GGT so you can distinguish custom tool files from usual GeoGebra files (GGB).

This dialog also allows you to remove or modify tools. If you decide to modify a tool, new GeoGebra window appears. The input objects are listed as free objects in it. If you have done finishing your changes, you can save the tool via option Create new tool in Tools Menu. Keep the old name to overwrite the tool. To overwrite a tool which was already used, the types of input and output objects must stay the same.

Accessibility

Mouse control

To work with GeoGebra using mouse control only, it is possible to show a virtual keyboard, embedded in GeoGebra 4. Select *View* and *Keyboard* to open it. You can use this keyboard by clicking the letters with the mouse.

Note: There are more symbols available than on a normal keyboard (like *pi*, *integral*, *alpha*, etc.). If you click on the button, you can use further characters.

Keyboard control

To open menus using keyboard only, press and arrows (on Windows). On Mac you have to enable *full keyboard access* first. Press to activate it. Now you are able to select menus by using or on some keyboards . For more keyboard options see the section Keyboard Shortcuts. Moreover, all features of the Properties Dialog are accessible via Scripting Commands.

GeoGebra Primary

To make GeoGebra easier to use for young students, we released GeoGebra primary which has bigger fonts and less GUI features.

GeoGebraPrim

GeoGebraPrim is a version of GeoGebra for primary school pupils. You can find it as webstart application here: http://www.geogebra.org/webstart/4.0/GeoGebraPrim.jnlp

Following features are intended to simplify the use of GeoGebra for the pupils:

- Restricted set of Tools available
- All available Tools visible at once
- Larger Font size
- Larger / Thicker Objects
- Just the Graphics View showing
- Easier to select objects
- Labeling of objects disabled
- Angles always between 0° and 180° by default
- Rounding to nearest whole number

Note: GeoGebra and GeoGebraPrim are actually the same application -- you can switch between these two interfaces via Perspectives submenu in Options Menu. All commands of GeoGebra are available in GeoGebraPrim.

Publishing

Creating Pictures of the Graphics View

You can either save a picture of the Graphics View in a file or copy it to clipboard.

Saving as File

Note: The full Graphics View will be saved as a picture, unless points Export_1 and Export_2 are defined (see below).

If your construction does not use all the available space in the Graphics View, you might want to…

- …use tools Move Graphics View Tool, Zoom In Tool and/or Zoom Out Tool in order to place your construction in the upper left corner of the Graphics View. Afterwards, you may reduce the size of the GeoGebra window by dragging one of its corners with the mouse.
- … use the selection rectangle in order to specify which part of the Graphics View should be exported and saved as a picture.

You create points called Export_1 and Export_2, which will be used to define diagonally opposite corners of the export rectangle.

Note: Points Export1 and Export2 must be within the visible area of the Graphics View.

In the File Menu, select item Export before clicking on item Graphics View as Picture. In the appearing dialog window you may specify the Format, Scale (in cm), and the Resolution (in dpi) of the output picture file.

Note: The true size of the exported image is shown at the bottom of the export window just above the buttons, both in centimeters and pixel.

Please find more information about the different picture files available in section Export Graphics Dialog.

Copying the Graphics View to Clipboard

There are two ways to copy the Graphics View to the clipboard of your computer :

- In the File menu, select Export, then click on Graphics View to Clipboard.
- In the Export Graphics View as Picture dialog window (menu File – Export – Graphics View as Picture (png, eps)…) click on the button Clipboard.

This feature copies a screenshot of the Graphics View to your system's clipboard as a PNG (see PNG format) picture. This picture can be pasted into other documents (e. g. a word processing document).

Note: In order to export your construction at a certain scale (in cm) please use the menu item Graphics View as Picture in the File menu, Export.

Upload to GeoGebraTube

There are two ways to upload a file to GeoGebraTube [2] directly from GeoGebra. First one is using the Export Worksheet Dialog, second one is using the *Share...* option in File Menu.

Note: This feature requires an active internet connection to work correctly.

In the first step GeoGebra is going to prepare your worksheet for upload to GeoGebraTube, afterwards your browser should open up and load a website which leads you through the process of publishing your worksheet on GeoGebraTube. More information about GeoGebraTube and its usage can be found in the wiki section GeoGebraTube.

Note: You can cancel the uploading process at any time by closing the browser window.

Export as html Webpage

To create an html file on your computer, you have to choose the *Export as Worksheet* tab.

- At the top of the export window you can enter the Title, Author and Date for your Dynamic Worksheet.
- Tab General allows you to add some text above and below the dynamic construction (e. g., a description of the construction and some tasks).
- Tab Advanced allows you to change the functionality of the dynamic construction (e. g., show a reset icon, double click to open the GeoGebra application window and browser features) as well as to modify the user interface shown in the interactive applet (e. g., show the Toolbar, modify height and width, enabling saving and printing, and others).

Note: If the size of your applet is too big to fit on a computer screen with standard resolution (1024 x 768), you may want to resize it before the actual export as a Dynamic Worksheet.

The exported HTML file (e. g. circle.html) can be viewed with any Internet browser (e. g. Mozilla, Internet Explorer, Safari). In order to let the dynamic construction work, Java has to be installed on the computer. If you want to use your Dynamic Worksheet in your school's computer network, ask your local network administrator to install Java on the computers. Make sure that your Java install includes the plugin for browsers.

Note:

You can edit the Dynamic Worksheet text with many word processing systems (e. g. FrontPage, OpenOffice Writer) by opening the exported HTML file. You may also edit the Dynamic Worksheet applet by opening the GGB file in GeoGebra and saving it with the same name afterwards. See Embedding to CMS, VLE (Moodle) and Wiki for details about exporting GeoGebra applets to these online systems.

Advanced settings

Functionality:

- *Enable right click, zooming and keyboard editing features*: By selecting this feature you will be able to right click objects or the drawing pad in order to access the features of the context menu (e.g. show / hide object or label, trace on / off, Properties dialog). It is also possible to use the common keyboard shortcuts.
- *Enable dragging of labels*: By selecting this feature you are able to drag labels of points or objects.
- *Show icon to reset construction*: A reset icon is displayed in the upper right corner of the interactive applet allowing your students to reset the interactive figure to its initial state.
- *Double click opens application window*: You will be able to open a full GeoGebra window by double clicking the interactive applet.
- *Button to open application window*: A button is displayed, to open the applet in a full GeoGebra window.
- *Use Browser for JavaScript Scripts*: JavaScipt Scripts are enabled. This allows your browser to show your construction properly.

User interface:

- *Show menubar*: The menubar is displayed within the interactive applet.
- *Enable save, print & undo*: It is possible to provide features for saving, printing and undoing the construction. Since this allows the applet to access your hard drive and printer, signed applets are used and every user of your applet is asked to confirm that he trusts it.
- *Show toolbar*: The toolbar is displayed within the interactive applet allowing to use the geometry tools.
- *Show toolbar help*: In combination with the toolbar you can also display the toolbar help within the interactive applet. If you want to provide geometry tools users of your worksheet can check the toolbar help in order to find out how to operate the different tools on their own.
- *Show inputbar*: The input field is displayed at the bottom of the interactive applet allowing to use algebraic input and commands for explorations.
- *Allow rescaling*: When this option is checked, the applet will try to rescale the construction accordingly to the zoom. This option is ignored if there are multiple views shown in the applet.
- *Width and height of the interactive applet*: You can modify the width and height of the interactive applet.

Note: If you reduce the size of the applet important parts of the dynamic worksheets might be invisible for users.

Hint: If you include the menubar, toolbar, or input field you might want to adjust the height of the interactive applet.

Files:

- *Include *.jar files*: Creates not only the html file, but also the *.jar files.

Note: You have to use this option if you want your applet to be available without connection to http://geogebra.org.

- *Remove Line Breaks*: removes line breaks from the resulting code. This is needed e.g. when including your applets in WordPress [1] but makes the resulting code less friendly for editing.
- *File-Dropdown*: You can choose if the export-file is html, MediaWiki, GoogleGadget or Moodle.
- *Single File*: A dynamic webpage will be created using a single file.
- *Single File (Tabs)*: More than one worksheet can be displayed in the browser, to navigate between the different tasks by using tabs.
- *Linked Files*: Creates linked dynamic worksheets and provides *Next* and *Back* buttons to work on more than one exercise.

Hint: For creating several interactive applets it is necessary to create more than one construction using *File* - *New Window* or .

Hint: The amount of data needed for the applet to load is **significantly** reduced (400kB rather than 1300kB when using geogebra.org as codebase) when following conditions are met:
just one Graphics View used (no Spreadsheet, Algebra View etc) there's no menubar, input bar etc use Browser for JavaScript is checked you don't use any commands which call the CAS eg f'(x), Integral,

Tangent, Expand http://www.geogebra.org/trac/browser/trunk/geogebra/geogebra/kernel/cas

you don't use any of these commands:

http://www.geogebra.org/trac/browser/trunk/geogebra/geogebra/kernel/discrete

References

[1] http://wordpress.com

Embedding to CMS, VLE (Moodle) and Wiki

If your Content Mangagement System or Virtual Learning Environment is capable of inserting raw HTML, the easiest way is to insert an interactive GeoGebra construction is to use the keyboard shortcut. The HTML code will appear in your clipboard and you can just switch your CMS or VLE to the mode which allows you HTML editing and paste it there. If you want to change some parameters of the text being copied, use the Export Worksheet Dialog and on the Advanced tab choose *Clipboard:html*

GeoGebraTube

Another option is to upload your worksheet to GeoGebraTube, which you can now easily do from within GeoGebra with the File > Share... option. GeoGebraTube will then give you an embed code which you can use.

MediaWiki and Google Sites

For MediaWiki [1] or Google Sites [2] you can use *File > Export > Dynamic Worksheet as Webpage* and in *Advanced* tab of the appearing Export Worksheet Dialog you choose *Clipboard:MediaWiki* or *Clipboard:Google Gadget* respectively. For MediaWiki it's sufficient to paste the code from clipboard to the edited page, for Google Sites you first have to store the code as a gadget and then use the Insert Gadget feature.

Note: For MediaWiki, DokuWiki and some other online systems a plugin is required for inserting GeoGebra applets. Please see Tutorial:Main Page for details.

References

[1] http://www.mediawiki.org
[2] http://sites.google.com

Export to LaTeX (PGF, PSTricks) and Asymptote

Export - Graphics View as Animated GIF...

This menu item allows you to save the Graphics View as an Animated GIF.

Export - Graphics View as PSTricks…

Keyboard shortcut: (MacOS:)

This menu item allows you to save the Graphics View as a PSTricks [1] picture file, which is a LaTeX picture format.

Export - Graphics View as PGF/TiKZ…

This menu item allows you to save the Graphics View as a PGF [2] picture file, which is a LaTeX picture format.

Export - Graphics View as Asymptote…

This menu item allows you to save the Graphics View as a Asymptote [3] picture file.

Limits of these export functions

Currently following objects cannot be exported to these formats:

- implicit curves
- loci

References

[1] http://tug.org/PSTricks/main.cgi/
[2] http://sourceforge.net/projects/pgf/
[3] http://asymptote.sourceforge.net/

Printing Options

Printing the Graphics View

GeoGebra allows you to print the Graphics View of your constructions. You can find the corresponding item Print Preview in the File Menu. In the appearing *Print Preview Dialog* window, you can specify the Title, Author, and a Date for the construction. In addition, you can set the Scale of your printout (in cm) and change the Orientation of the paper used (portrait or landscape).

Note: In order to update the Print Preview after you made changes to the text or layout of the printout, you need to press .

Printing the Construction Protocol

If you want to print the Construction Protocol, you first need to open the Construction Protocol dialog window by using the View menu. Then, select Print... from the File menu of this new window.

The Print dialog window allows you to enter the Author or change the paper Margins and Orientation before printing the Construction Protocol.

Note: You may switch the different columns Name, Definition, Command, Algebra, and Breakpoint of the Construction Protocol on and off by using the View menu of the Construction Protocol dialog window.

Article Sources and Contributors

Introduction *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23266 *Contributors*: Christina.biermair, Florian Sonner, Kondr, Markus, Murkle, Noel Lambert

Compatibility *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23455 *Contributors*: Kondr, Murkle

Installation Guide *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=2293 *Contributors*: Kondr

Free, Dependent and Auxiliary Objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10972 *Contributors*: Andrea.duringer, Kondr, Mathmum

Geometric Objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8540 *Contributors*: Kondr, Mathmum

Points and Vectors *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12694 *Contributors*: Andrea.duringer, Kondr, Mathmum

Lines and Axes *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12685 *Contributors*: Andrea.duringer, Kondr, Mathmum, Murkle

Conic sections *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7179 *Contributors*: Kondr, Mathmum

Functions *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8737 *Contributors*: Kondr, Mathmum

Curves *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11140 *Contributors*: Andrea.duringer, Kondr, Mathmum, Murkle

Inequalities *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8413 *Contributors*: Birgit Lachner, Kondr, Mathmum

Intervals *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12105 *Contributors*: Kondr, Mathmum

General Objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5814 *Contributors*: Kondr, Mathmum

Numbers and Angles *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7029 *Contributors*: Kondr, Mathmum

Texts *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23198 *Contributors*: Andrea.duringer, Kondr, Mathmum, Murkle

Boolean values *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23019 *Contributors*: Kondr, Mathmum

Complex Numbers *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10991 *Contributors*: Andrea.duringer, Kondr, Mathmum

Lists *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23232 *Contributors*: Andrea.duringer, Kondr, Mathmum, Murkle

Matrices *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7000 *Contributors*: Kondr, Mathmum

Action Objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11233 *Contributors*: Kondr, Mathmum

Selecting objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5684 *Contributors*: Kondr, Mathmum

Change Values *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5725 *Contributors*: Kondr, Mathmum

Naming Objects *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12693 *Contributors*: Andrea.duringer, Kondr, Markus, Mathmum, Murkle

Animation *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6990 *Contributors*: Florian Sonner, Kondr, Mathmum

Tracing *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5811 *Contributors*: Kondr, Mathmum

Object Properties *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11966 *Contributors*: Andrea.duringer, Kondr, Mathmum

Labels and Captions *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14492 *Contributors*: Birgit Lachner, Kondr, Mathmum, Murkle

Advanced Features *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6645 *Contributors*: Kondr, Mathmum

Object Position *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10975 *Contributors*: Andrea.duringer, Kondr

Conditional Visibility *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5729 *Contributors*: Kondr, Mathmum

Dynamic Colors *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8551 *Contributors*: Kondr, Mathmum

LaTeX *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23193 *Contributors*: Birgit Lachner, Kondr, Mathmum

Layers *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10970 *Contributors*: Andrea.duringer, Kondr, Mathmum

Scripting *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23079 *Contributors*: Birgit Lachner, Corinna, Kondr, Mathmum, Murkle

Tooltips *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8685 *Contributors*: Kondr

Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7155 *Contributors*: Kondr, Mathmum

Movement Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7072 *Contributors*: Kondr, Mathmum

Move Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8585 *Contributors*: Florian Sonner, K Voss, Kondr, Mathmum

Record to Spreadsheet Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5644 *Contributors*: K Voss, Kondr, Mathmum

Rotate around Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5796 *Contributors*: K Voss, Kondr, Mathmum

Point Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11152 *Contributors*: Kondr, Mathmum

New Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5735 *Contributors*: K Voss, Kondr, Mathmum

Attach / Detach Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6852 *Contributors*: Kondr

Complex Number Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11149 *Contributors*: Murkle

Point on Object Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12219 *Contributors*: Andrea.duringer, Kondr, Murkle

Intersect Two Objects Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23199 *Contributors*: K Voss, Kondr, Mathmum

Midpoint or Center Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5564 *Contributors*: K Voss, Kondr, Mathmum

Line Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7080 *Contributors*: Kondr, Mathmum

Vector from Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5667 *Contributors*: K Voss, Kondr, Mathmum

Ray through Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5641 *Contributors*: K Voss, Kondr, Mathmum

Segment with Given Length from Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5690 *Contributors*: K Voss, Kondr, Mathmum

Line through Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5554 *Contributors*: K Voss, Kondr, Mathmum

Segment between Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5642 *Contributors*: K Voss, Kondr, Mathmum

Vector between Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5666 *Contributors*: K Voss, Kondr, Mathmum

Special Line Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7084 *Contributors*: Kondr, Mathmum

Best Fit Line Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5533 *Contributors*: K Voss, Kondr, Mathmum

Parallel Line Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5496 *Contributors*: K Voss, Kondr, Mathmum

Angle Bisector Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5446 *Contributors*: K Voss, Kondr, Mathmum

Perpendicular Line Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5625 *Contributors*: K Voss, Kondr, Mathmum

Tangents Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8918 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Polar or Diameter Line Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5627 *Contributors*: K Voss, Kondr, Mathmum

Perpendicular Bisector Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5624 *Contributors*: K Voss, Kondr, Mathmum

Locus Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22964 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Polygon Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9121 *Contributors*: Kondr, Mathmum

Rigid Polygon Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23298 *Contributors*: Kondr, Mathmum

PolyLine Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11634 *Contributors*: Andrea.duringer, Christina.biermair

Regular Polygon Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5646 *Contributors*: K Voss, Kondr, Mathmum

Polygon Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23295 *Contributors*: K Voss, Kondr, Mathmum

Circle & Arc Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7096 *Contributors*: Kondr, Mathmum

Circle with Center and Radius Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7878 *Contributors*: K Voss, Kondr, Mathmum

Circle through Three Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9002 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Circle with Center through Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5502 *Contributors*: Administrator, K Voss, Kondr, Mathmum

Circumcircular Arc through Three Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5516 *Contributors*: K Voss, Kondr, Mathmum

Circumcircular Sector through Three Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5517 *Contributors*: K Voss, Kondr, Mathmum

Compass Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5522 *Contributors*: Administrator, K Voss, Kondr, Mathmum

Circular Sector with Center between Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5512 *Contributors*: K Voss, Kondr, Mathmum

Semicircle through Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5658 *Contributors*: Kondr, Mathmum

Circular Arc with Center between Two Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8398 *Contributors*: K Voss, Kondr, Mathmum

Conic Section Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9075 *Contributors*: Christina.biermair, Kondr, Mathmum

Ellipse Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5472 *Contributors*: Administrator, Florian Sonner, K Voss, Kondr, Mathmum

Hyperbola Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5493 *Contributors*: K Voss, Kondr, Mathmum

Conic through Five Points Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5528 *Contributors*: K Voss, Kondr, Mathmum

Parabola Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5495 *Contributors*: K Voss, Kondr, Mathmum

Measurement Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11701 *Contributors*: Christina.biermair, Kondr, Mathmum

Distance or Length Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5469 *Contributors*: K Voss, Kondr, Mathmum

Angle Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5454 *Contributors*: K Voss, Kondr, Mathmum

Slope Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5647 *Contributors*: K Voss, Kondr, Mathmum

Area Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5465 *Contributors*: K Voss, Kondr, Mathmum

Angle with Given Size Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5455 *Contributors*: K Voss, Kondr, Mathmum

Transformation Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9073 *Contributors*: Christina.biermair, Kondr, Mathmum

Translate Object by Vector Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23458 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Reflect Object about Line Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5802 *Contributors*: K Voss, Kondr, Mathmum

Reflect Object about Point Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5803 *Contributors*: K Voss, Kondr, Mathmum

Rotate Object around Point by Angle Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5795 *Contributors*: K Voss, Kondr, Mathmum

Reflect Object about Circle Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23372 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Dilate Object from Point by Factor Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5799 *Contributors*: K Voss, Kondr, Mathmum

Special Object Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11861 *Contributors*: Andrea.duringer, Christina.biermair, Kondr, Mathmum

Insert Image Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8414 *Contributors*: Kondr, Mathmum

Probability Calculator Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11806 *Contributors*: Gsturr, Kondr, Mathmum

Pen Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6902 *Contributors*: Kondr, Mathmum

Slider Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=13950 *Contributors*: Andrea.duringer, Florian Sonner, K Voss, Kondr, Mathmum, Murkle

Relation between Two Objects Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5686 *Contributors*: Kondr, Mathmum

Function Inspector Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11764 *Contributors*: Andrea.duringer, Kondr, Mathmum

Insert Text Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23281 *Contributors*: Kondr, Mathmum

Action Object Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11862 *Contributors*: Andrea.duringer, Christina.biermair, Kondr, Mathmum

Check Box to Show / Hide Objects Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8399 *Contributors*: K Voss, Kondr, Mathmum

Insert Input Box Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11221 *Contributors*: Kondr

Insert Button Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8656 *Contributors*: Kondr

General Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9071 *Contributors*: Christina.biermair, Kondr, Mathmum

Custom Tools *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11170 *Contributors*: Kondr, Mathmum

Show / Hide Label Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=1271 *Contributors*: K Voss, Kondr, Mathmum

Zoom Out Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5670 *Contributors*: K Voss, Kondr, Mathmum

Zoom In Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5669 *Contributors*: K Voss, Kondr, Mathmum

Delete Object Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9003 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Move Graphics View Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5733 *Contributors*: Kondr, Mathmum

Show / Hide Object Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=1277 *Contributors*: K Voss, Kondr, Mathmum

Copy Visual Style Tool *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7174 *Contributors*: Kondr, Mathmum

Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9056 *Contributors*: Florian Sonner, Kondr, Mathmum, Noel Lambert

Geometry Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11871 *Contributors*: Andrea.duringer, Kondr, Mathmum

AffineRatio Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5880 *Contributors*: K Voss, Kondr

Angle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14319 *Contributors*: Jackhu, K Voss, Kondr, Mathmum

AngleBisector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5466 *Contributors*: K Voss, Kondr, Mathmum

Arc Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6853 *Contributors*: K Voss, Kondr

Area Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14579 *Contributors*: Administrator, K Voss, Kondr, Mathieu

Centroid Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5476 *Contributors*: K Voss, Kondr, Mathmum

CircularArc Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9011 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

CircularSector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9012 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

CircumcircularArc Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8992 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

CircumcircularSector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8998 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Circumference Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9005 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

ClosestPoint Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14305 *Contributors*: Andrea.duringer, Kondr, Murkle

CrossRatio Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5531 *Contributors*: K Voss, Kondr, Mathmum

Direction Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5602 *Contributors*: K Voss, Kondr, Mathmum

Distance Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10888 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

Intersect Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12617 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

IntersectRegion Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10857 *Contributors*: Kondr

Length Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14602 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathieu, Mathmum, Murkle, UnTom

Line Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5553 *Contributors*: K Voss, Kondr, Mathmum

PerpendicularBisector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10878 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Locus Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23054 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Midpoint Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23065 *Contributors*: K Voss, Kondr, Mathmum

PerpendicularLine Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9015 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Perimeter Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6847 *Contributors*: K Voss, Kondr, Mathmum

Point Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6848 *Contributors*: K Voss, Kondr, Mathmum

PointIn Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10908 *Contributors*: Andrea.duringer, Kondr

PolyLine Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6915 *Contributors*: Kondr, Mathmum

Polygon Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9004 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Radius Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5635 *Contributors*: K Voss, Kondr, Mathmum

Ray Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8996 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

RigidPolygon Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8624 *Contributors*: Kondr

Sector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5577 *Contributors*: K Voss, Kondr, Mathmum

Segment Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9010 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Slope Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10947 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Tangent Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23038 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

Vertex Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8411 *Contributors*: K Voss, Kondr, Mathmum

Algebra Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11864 *Contributors*: Andrea.duringer, Kondr, Mathmum

Div Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12636 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum, UnTom

Expand Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12606 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Noel Lambert

Factor Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12607 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

GCD Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12594 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

LCM Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22997 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Max Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12592 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

Min Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14325 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Murkle

Mod Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12635 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

PrimeFactors Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12655 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

Simplify Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12619 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, Murkle

Text Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6787 *Contributors*: Kondr, Mathmum

FractionText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6326 *Contributors*: K Voss, Kondr, Mathmum

FormulaText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23324 *Contributors*: K Voss, Kondr, Mathmum

LetterToUnicode Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7023 *Contributors*: K Voss, Kondr, Mathmum

Ordinal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8778 *Contributors*: Kondr, Mathmum, Murkle

RotateText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8639 *Contributors*: Kondr

TableText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11789 *Contributors*: K Voss, Kondr, Mathmum, Murkle, Noel Lambert

Text Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8535 *Contributors*: K Voss, Kondr, Mathmum

TextToUnicode Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7019 *Contributors*: K Voss, Kondr, Mathmum

UnicodeToLetter Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6984 *Contributors*: K Voss, Kondr, Mathmum

UnicodeToText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6994 *Contributors*: K Voss, Kondr, Mathmum

VerticalText Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8997 *Contributors*: Andrea.duringer, Kondr

Logic Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7128 *Contributors*: Kondr, Mathmum

CountIf Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14312 *Contributors*: K Voss, Kondr, Mathmum

IsDefined Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5541 *Contributors*: K Voss, Kondr, Mathmum

If Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14299 *Contributors*: Christina.biermair, K Voss, Kondr, Mathmum, Murkle

IsInRegion Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10869 *Contributors*: Andrea.duringer, Kondr

IsInteger Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5542 *Contributors*: K Voss, Kondr, Mathmum

KeepIf Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14311 *Contributors*: K Voss, Kondr, Mathmum

Relation Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5681 *Contributors*: K Voss, Kondr, Mathmum

Functions & Calculus Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11921 *Contributors*: Andrea.duringer, Kondr, Mathmum

Asymptote Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8916 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Coefficients Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12638 *Contributors*: Kondr, Mathmum

CompleteSquare Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8921 *Contributors*: Kondr

ComplexRoot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12514 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum

Curvature Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5448 *Contributors*: K Voss, Kondr, Mathmum

CurvatureVector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5449 *Contributors*: K Voss, Kondr, Mathmum

Curve Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23020 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Degree Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=18178 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum

Denominator Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12644 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

Derivative Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22972 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, Murkle, UnTom

Extremum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14326 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Factors Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12634 *Contributors*: Alexander Hartl, Kondr

Function Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23021 *Contributors*: Florian Sonner, K Voss, Kondr, Mathmum, Murkle

ImplicitCurve Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10025 *Contributors*: Kondr, Murkle

Integral Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22973 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum, Murkle

IntegralBetween Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22974 *Contributors*: Alexander Hartl, Christina.biermair, Kondr

Intersect Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12617 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

Iteration Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6387 *Contributors*: K Voss, Kondr, Mathmum

IterationList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6346 *Contributors*: K Voss, Kondr, Mathmum

LeftSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10871 *Contributors*: Andrea.duringer, Kondr

Limit Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22975 *Contributors*: Alexander Hartl, Christina.biermair, Kondr

LimitAbove Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22976 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum

LimitBelow Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22977 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum, Spanish1

LowerSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5558 *Contributors*: K Voss, Kondr, Mathmum

Numerator Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12643 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

OsculatingCircle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5498 *Contributors*: K Voss, Kondr, Mathmum

PartialFractions Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22980 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

PathParameter Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8541 *Contributors*: Kondr, Mathmum

Polynomial Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7018 *Contributors*: K Voss, Kondr, Mathmum

RectangleSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23326 *Contributors*: Kondr, Mathmum

Root Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12586 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

RootList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6830 *Contributors*: Kondr

Roots Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8582 *Contributors*: Kondr

SolveODE Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23175 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Markus, Mathmum, Murkle

TaylorPolynomial Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12587 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

TrapezoidalSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5654 *Contributors*: K Voss, Kondr, Mathmum

InflectionPoint Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5614 *Contributors*: K Voss, Kondr, Mathmum

UpperSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5662 *Contributors*: K Voss, Kondr, Mathmum

Conic Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10968 *Contributors*: Kondr, Mathmum, Spanish1

Asymptote Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8916 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Axes Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5460 *Contributors*: K Voss, Kondr, Mathmum

Center Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5505 *Contributors*: K Voss, Kondr, Mathmum

Circle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5477 *Contributors*: Administrator, K Voss, Kondr, Mathmum

Conic Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=13877 *Contributors*: Christina.biermair, K Voss, Kondr, Mathmum

ConjugateDiameter Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5529 *Contributors*: K Voss, Kondr, Mathmum

Directrix Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5462 *Contributors*: K Voss, Kondr, Mathmum

Eccentricity Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6891 *Contributors*: Kondr, Mathmum

Ellipse Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5471 *Contributors*: Administrator, K Voss, Kondr, Mathmum

LinearEccentricity Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5552 *Contributors*: K Voss, Kondr, Mathmum

MajorAxis Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5559 *Contributors*: K Voss, Kondr, Mathmum

SemiMajorAxisLength Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5579 *Contributors*: K Voss, Kondr, Mathmum

Focus Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5475 *Contributors*: K Voss, Kondr, Mathmum

Hyperbola Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5492 *Contributors*: K Voss, Kondr, Mathmum

Incircle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9078 *Contributors*: Murkle

Parabola Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11869 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Parameter Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5572 *Contributors*: K Voss, Kondr, Mathmum

Polar Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9014 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

MinorAxis Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5566 *Contributors*: K Voss, Kondr, Mathmum

SemiMinorAxisLength Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5580 *Contributors*: K Voss, Kondr, Mathmum

Semicircle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14484 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

List Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12695 *Contributors*: Andrea.duringer, Kondr, Mathmum

Append Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6992 *Contributors*: K Voss, Kondr, Mathmum

Classes Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8630 *Contributors*: Kondr

Element Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12633 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, Murkle, UnTom

First Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12650 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, Murkle

Frequency Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11843 *Contributors*: Gsturr, Kondr, Mathmum

IndexOf Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8759 *Contributors*: Kondr, Mathmum

Insert Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6964 *Contributors*: K Voss, Kondr, Mathmum

Intersect Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12617 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

Intersection Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5613 *Contributors*: K Voss, Kondr, Mathmum

IterationList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6346 *Contributors*: K Voss, Kondr, Mathmum

Join Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7017 *Contributors*: K Voss, Kondr, Mathmum

Last Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12651 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

OrdinalRank Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14304 *Contributors*: Mathmum, Murkle

PointList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6824 *Contributors*: Kondr

Product Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12608 *Contributors*: Alexander Hartl, Andrea.duringer, K Voss, Kondr, Mathmum, UnTom

RandomElement Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12648 *Contributors*: Christina.biermair, Kondr, UnTom

RemoveUndefined Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8476 *Contributors*: K Voss, Kondr, Mathmum

Reverse Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5710 *Contributors*: K Voss, Kondr, Mathmum

RootList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6830 *Contributors*: Kondr

SelectedElement Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8628 *Contributors*: Kondr

SelectedIndex Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8629 *Contributors*: Kondr

Sequence Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12590 *Contributors*: Alexander Hartl, K Voss, Kondr, Mathmum, UnTom

Sort Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6988 *Contributors*: K Voss, Kondr, Mathmum

Take Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12652 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

TiedRank Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11912 *Contributors*: Andrea.duringer, Kondr, Murkle, Spanish1

Union Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8724 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Unique Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12654 *Contributors*: Alexander Hartl, Andrea.duringer, Gsturr, Kondr, Mathmum, UnTom

Zip Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11000 *Contributors*: Andrea.duringer, Kondr

Vector & Matrix Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11009 *Contributors*: Kondr, Mathmum

ApplyMatrix Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10860 *Contributors*: Kondr

CurvatureVector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5449 *Contributors*: K Voss, Kondr, Mathmum

Determinant Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12615 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

Identity Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12658 *Contributors*: Andrea.duringer, Christina.biermair, Kondr, Murkle

Invert Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14297 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum, Murkle, UnTom

PerpendicularVector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12585 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

ReducedRowEchelonForm Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12656 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Noel Lambert

Transpose Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12614 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

UnitPerpendicularVector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12588 *Contributors*: Alexander Hartl, Andrea.duringer, K Voss, Kondr, Mathmum

UnitVector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12589 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, UnTom

Vector Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8993 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Transformation Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11876 *Contributors*: Andrea.duringer, Kondr, Mathmum

Dilate Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11478 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Reflect Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5801 *Contributors*: K Voss, Kondr, Mathmum

Rotate Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23229 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum, Murkle

Shear Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8757 *Contributors*: Kondr, Mathmum

Stretch Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9093 *Contributors*: Kondr, Murkle

Translate Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8995 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

Chart Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11885 *Contributors*: Christina.biermair, Kondr, Mathmum

BarChart Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7187 *Contributors*: K Voss, Kondr, Mathmum

BoxPlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6997 *Contributors*: K Voss, Kondr, Mathmum

DotPlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8678 *Contributors*: Kondr

FrequencyPolygon Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11793 *Contributors*: Kondr, Mathmum

Histogram Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11845 *Contributors*: Gsturr, K Voss, Kondr, Mathmum, Murkle

HistogramRight Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11891 *Contributors*: Christina.biermair, Kondr

NormalQuantilePlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11787 *Contributors*: Gsturr, Kondr

ResidualPlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8650 *Contributors*: Kondr

StemPlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11068 *Contributors*: Gsturr, Kondr

Statistics Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11931 *Contributors*: Andrea.duringer, Kondr, Mathmum

ANOVA Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10941 *Contributors*: Gsturr, Kondr, Mathmum

Classes Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8630 *Contributors*: Kondr

Covariance Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12599 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

Fit Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6760 *Contributors*: Kondr

FitExp Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12601 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

FitGrowth Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23332 *Contributors*: Kondr, Mathmum

FitLineX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10951 *Contributors*: Andrea.duringer, Kondr

FitLine Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10950 *Contributors*: Andrea.duringer, K Voss, Kondr, Mathmum

FitLog Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12602 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

FitLogistic Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5484 *Contributors*: K Voss, Kondr, Mathmum, Murkle

FitPoly Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12600 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

FitPow Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12603 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

FitSin Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12616 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum, Murkle

Frequency Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11843 *Contributors*: Gsturr, Kondr, Mathmum

FrequencyTable Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11972 *Contributors*: Andrea.duringer, Kondr

GeometricMean Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6835 *Contributors*: Kondr

HarmonicMean Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6836 *Contributors*: Kondr

Mean Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12595 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

MeanX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5432 *Contributors*: Kondr

MeanY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5433 *Contributors*: Kondr

Median Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12598 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

Mode Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7003 *Contributors*: K Voss, Kondr, Mathmum

CorrelationCoefficient Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5538 *Contributors*: K Voss, Kondr, Mathmum

Percentile Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11107 *Contributors*: Andrea.duringer, Gsturr, Kondr

Q1 Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5633 *Contributors*: K Voss, Kondr, Mathmum

Q3 Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5634 *Contributors*: K Voss, Kondr, Mathmum

RSquare Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11894 *Contributors*: Andrea.duringer, Gsturr, Kondr

RootMeanSquare Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6829 *Contributors*: Kondr

SD Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12597 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

SDX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8924 *Contributors*: Kondr

SDY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8925 *Contributors*: Kondr

Sxx Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10808 *Contributors*: K Voss, Kondr, Mathmum

Sxy Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10814 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Syy Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10815 *Contributors*: K Voss, Kondr, Murkle

Sample Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12653 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum

SampleSD Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12646 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

SampleSDX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8671 *Contributors*: Kondr

SampleSDY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8672 *Contributors*: Kondr

SampleVariance Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12645 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum, UnTom

Shuffle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12647 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

SigmaXX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7004 *Contributors*: K Voss, Kondr, Mathmum

SigmaXY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7011 *Contributors*: K Voss, Kondr, Mathmum

SigmaYY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5851 *Contributors*: K Voss, Kondr

Spearman Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8913 *Contributors*: Gsturr, Kondr

Sum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12609 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

SumSquaredErrors Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23333 *Contributors*: Kondr, Mathmum

TMean2Estimate Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12028 *Contributors*: Christina.biermair, Gsturr, Kondr

TMeanEstimate Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10903 *Contributors*: Gsturr, Kondr

TTest Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10892 *Contributors*: Gsturr, Kondr

TTest2 Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10946 *Contributors*: Gsturr, Kondr, Mathmum

TTestPaired Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10900 *Contributors*: Gsturr, Kondr

Variance Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12596 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum

Probability Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11922 *Contributors*: Andrea.duringer, Kondr, Mathmum

Bernoulli Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10103 *Contributors*: Kondr

BinomialCoefficient Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12605 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, K Voss, Kondr, Mathmum, Murkle, Noel Lambert

BinomialDist Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12630 *Contributors*: Alexander Hartl, Murkle

Cauchy Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12623 *Contributors*: Alexander Hartl, Kondr, Mathmum

ChiSquared Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12624 *Contributors*: Alexander Hartl, Christina.biermair, Kondr

Erlang Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11005 *Contributors*: Corinna, Kondr

Exponential Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12625 *Contributors*: Alexander Hartl, Cmiic, Kondr, Noel Lambert

FDistribution Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12621 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Murkle

Gamma Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12622 *Contributors*: Christina.biermair, Kondr

HyperGeometric Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12631 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

InverseBinomial Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10308 *Contributors*: Kondr

InverseCauchy Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10844 *Contributors*: Kondr

InverseChiSquared Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10845 *Contributors*: Kondr

InverseExponential Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10846 *Contributors*: Kondr

InverseFDistribution Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10847 *Contributors*: Kondr

InverseGamma Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10848 *Contributors*: Kondr

InverseHyperGeometric Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10305 *Contributors*: Kondr

InverseNormal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12659 *Contributors*: K Voss, Kondr, Mathmum, Murkle

InversePascal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10301 *Contributors*: Kondr

InversePoisson Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10287 *Contributors*: Kondr

InverseTDistribution Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10851 *Contributors*: Kondr

InverseWeibull Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11264 *Contributors*: Andrea.duringer, Kondr, Murkle

InverseZipf Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10303 *Contributors*: Kondr

LogNormal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10864 *Contributors*: Kondr

Logistic Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11632 *Contributors*: Kondr

Normal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12627 *Contributors*: Alexander Hartl, Andrea.duringer, K Voss, Kondr, Mathmum

Pascal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12628 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

Poisson Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12629 *Contributors*: Alexander Hartl, Christina.biermair, Kondr

RandomBetween Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12610 *Contributors*: K Voss, Kondr, Mathmum

RandomBinomial Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12611 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum, UnTom

RandomNormal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12604 *Contributors*: Christina.biermair, K Voss, Kondr, Mathmum, UnTom

RandomPoisson Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12612 *Contributors*: Christina.biermair, K Voss, Kondr, Mathmum, UnTom

RandomUniform Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8736 *Contributors*: Kondr

TDistribution Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12620 *Contributors*: Kondr, Noel Lambert

Triangular Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11004 *Contributors*: Andrea.duringer, Corinna, Kondr

Uniform Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9109 *Contributors*: Kondr

Weibull Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12626 *Contributors*: Alexander Hartl, Kondr

Zipf Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12632 *Contributors*: Alexander Hartl, Kondr

Spreadsheet Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6792 *Contributors*: Kondr, Mathmum

Cell Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23237 *Contributors*: Kondr, Murkle

CellRange Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7191 *Contributors*: K Voss, Kondr, Mathmum, Spanish1

Column Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6366 *Contributors*: K Voss, Kondr, Mathmum

ColumnName Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6368 *Contributors*: K Voss, Kondr, Mathmum

FillCells Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11753 *Contributors*: Andrea.duringer, Kondr, Spanish1

FillColumn Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6921 *Contributors*: Kondr

FillRow Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6920 *Contributors*: Kondr

Row Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6370 *Contributors*: K Voss, Kondr, Mathmum

Scripting Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11923 *Contributors*: Andrea.duringer, Kondr, Mathmum, Spanish1

Button Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6580 *Contributors*: Kondr

Checkbox Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22999 *Contributors*: Kondr, Murkle

CopyFreeObject Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7158 *Contributors*: Kondr

Delete Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12582 *Contributors*: Alexander Hartl, Christina.biermair, K Voss, Kondr, Mathmum

Execute Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8713 *Contributors*: Gsturr, Kondr, Murkle

GetTime Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23325 *Contributors*: Kondr, Mathmum

HideLayer Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6845 *Contributors*: Kondr

Pan Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6823 *Contributors*: Kondr

ParseToFunction Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8524 *Contributors*: Kondr

ParseToNumber Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8523 *Contributors*: Kondr

PlaySound Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10889 *Contributors*: Gsturr, Kondr, Mathmum

Rename Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6869 *Contributors*: Kondr

SelectObjects Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8654 *Contributors*: Kondr

SetActiveView Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7280 *Contributors*: Kondr

SetAxesRatio Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8702 *Contributors*: Kondr

SetBackgroundColor Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11052 *Contributors*: Kondr, Murkle

SetCaption Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7234 *Contributors*: Kondr

SetColor Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11057 *Contributors*: Kondr, Murkle

SetConditionToShowObject Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6843 *Contributors*: Kondr

SetCoords Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7159 *Contributors*: Kondr

SetDynamicColor Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11733 *Contributors*: Kondr, Murkle, Spanish1

SetFilling Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8515 *Contributors*: Kondr, Mathmum

SetFixed Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7230 *Contributors*: Kondr

SetLabelMode Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7241 *Contributors*: Kondr

SetLayer Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7257 *Contributors*: Kondr

SetLineStyle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8643 *Contributors*: Kondr

SetLineThickness Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6885 *Contributors*: Kondr

SetPointSize Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7231 *Contributors*: Kondr

SetPointStyle Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8646 *Contributors*: Kondr

SetTooltipMode Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8683 *Contributors*: Kondr

SetValue Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23329 *Contributors*: Kondr, Mathmum

SetVisibleInView Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7261 *Contributors*: Kondr

ShowLabel Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8617 *Contributors*: Murkle

ShowLayer Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6846 *Contributors*: Kondr

Slider Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22981 *Contributors*: Kondr, Murkle

StartAnimation Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8584 *Contributors*: Kondr

InputBox Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11229 *Contributors*: Andrea.duringer, Birgit Lachner, Kondr, Spanish1

UpdateConstruction Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23330 *Contributors*: Kondr, Mathmum

ZoomIn Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12107 *Contributors*: Kondr

ZoomOut Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8412 *Contributors*: Kondr, Mathmum

Discrete Math Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6819 *Contributors*: Kondr, Mathmum

ConvexHull Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8637 *Contributors*: Kondr

DelaunayTriangulation Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8638 *Contributors*: Kondr

Hull Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8701 *Contributors*: Kondr

MinimumSpanningTree Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7225 *Contributors*: Kondr

ShortestDistance Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8543 *Contributors*: Kondr

TravelingSalesman Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11065 *Contributors*: Kondr

Voronoi Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8636 *Contributors*: Kondr

GeoGebra Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6790 *Contributors*: Kondr

AxisStepX Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6452 *Contributors*: Kondr

AxisStepY Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6453 *Contributors*: Kondr

ClosestPoint Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14305 *Contributors*: Andrea.duringer, Kondr, Murkle

ConstructionStep Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6457 *Contributors*: K Voss, Kondr, Mathmum

Corner Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7162 *Contributors*: K Voss, Kondr, Mathmum

DynamicCoordinates Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=9092 *Contributors*: Kondr, Mathmum, Noel Lambert, Spanish1

Name Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22970 *Contributors*: K Voss, Kondr, Mathmum, Murkle

Object Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23240 *Contributors*: K Voss, Kondr, Mathmum, Murkle

SlowPlot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8595 *Contributors*: Kondr

ToolImage Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6466 *Contributors*: Kondr

Optimization Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6778 *Contributors*: Kondr, Mathmum

Maximize Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11063 *Contributors*: Kondr

Minimize Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11061 *Contributors*: Kondr

CAS Specific Commands *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14580 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Florian Sonner, Kondr, Mathmum, Spanish1

CFactor Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12502 *Contributors*: Alexander Hartl, Kondr, UnTom

CSolutions Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14595 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum, Noel Lambert

CSolve Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14596 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, UnTom

CommonDenominator Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12513 *Contributors*: Alexander Hartl, Kondr, Mathmum, Murkle

Cross Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12520 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum, Spanish1

Decimal Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12522 *Contributors*: Alexander Hartl, Kondr, Mathmum

Dimension Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12536 *Contributors*: Alexander Hartl, Kondr, Mathmum

Division Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12550 *Contributors*: Alexander Hartl, Kondr, Mathmum

Divisors Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12549 *Contributors*: Alexander Hartl, Kondr, Mathmum

DivisorsList Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12552 *Contributors*: Alexander Hartl, Kondr

DivisorsSum Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12551 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr

Dot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12553 *Contributors*: Alexander Hartl, Kondr, Mathmum

FractionalPart Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11936 *Contributors*: Alexander Hartl, Kondr, Mathmum

Imaginary Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12249 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

ImplicitDerivative Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12253 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr, Mathmum

IntegerPart Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11940 *Contributors*: Alexander Hartl, Kondr, Mathmum

IsPrime Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12572 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

LeftSide Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12254 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr, Mathmum

MatrixRank Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12573 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Murkle

MixedNumber Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12240 *Contributors*: Alexander Hartl, Corinna, Kondr

NIntegral Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12114 *Contributors*: Alexander Hartl, Kondr

NRoot Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12122 *Contributors*: Alexander Hartl, Kondr, UnTom

NSolutions Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22979 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

NSolve Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=22978 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum, UnTom

NextPrime Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12153 *Contributors*: Alexander Hartl, Kondr, Mathmum

Numeric Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12537 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum

PreviousPrime Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12154 *Contributors*: Alexander Hartl, Kondr, Mathmum

RandomPolynomial Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12233 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr, Simon, UnTom

Rationalize Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12235 *Contributors*: Corinna, Kondr, UnTom

Real Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12243 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr, Mathmum

RightSide Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12264 *Contributors*: Alexander Hartl, Andrea.duringer, Kondr, Mathmum

Solutions Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14593 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum, Noel Lambert, UnTom

Solve Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14592 *Contributors*: Alexander Hartl, Christina.biermair, Kondr, Mathmum, Murkle, UnTom

Substitute Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12301 *Contributors*: Alexander Hartl, Christina.biermair, Kondr

ToComplex Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12358 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

ToExponential Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12359 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Spanish1

ToPoint Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12544 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr

ToPolar Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12363 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Kondr, Mathmum, UnTom

nPr Command *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12115 *Contributors*: Alexander Hartl, Kondr, Mathmum

Predefined Functions and Operators *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14597 *Contributors*: Kondr, Murkle, Noel Lambert

Views *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6237 *Contributors*: Kondr, Mathmum

Graphics View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23156 *Contributors*: Christina.biermair, Corinna, Florian Sonner, Kondr, Markus, Mathmum, Murkle, Noel Lambert

Customizing the Graphics View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12078 *Contributors*: Kondr, Mathmum

Algebra View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8529 *Contributors*: Kondr, Mathmum

Spreadsheet View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7244 *Contributors*: Kondr, Mathmum

CAS View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12097 *Contributors*: Alexander Hartl, Andrea.duringer, Christina.biermair, Florian Sonner, Kondr, Mathmum

Construction Protocol *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10989 *Contributors*: Andrea.duringer, Kondr, Mathmum

Input Bar *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7228 *Contributors*: Kondr, Mathmum

Menubar *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5700 *Contributors*: Kondr, Mathmum

Toolbar *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8679 *Contributors*: Kondr

Navigation Bar *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11962 *Contributors*: Kondr, Mathmum

File Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14555 *Contributors*: Christina.biermair, Florian Sonner, Kondr, Mathmum, Murkle

Edit Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7243 *Contributors*: Florian Sonner, Kondr, Mathmum

View Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12353 *Contributors*: Christina.biermair, Kondr, Mathmum

Perspectives *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11879 *Contributors*: Corinna, Kondr

Options Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8715 *Contributors*: Kondr, Mathmum

Tools Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5650 *Contributors*: Kondr, Mathmum

Window Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5687 *Contributors*: Kondr, Mathmum

Help Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=7248 *Contributors*: Kondr

Context Menu *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5721 *Contributors*: Kondr, Mathmum

Customize the Settings *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5453 *Contributors*: Kondr, Mathmum

Export Graphics Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=23013 *Contributors*: Kondr, Mathmum, Murkle

Export Worksheet Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=14495 *Contributors*: Andrea.duringer, Christina.biermair, Kimeswenger, Kondr, Mathmum

Properties Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10969 *Contributors*: Andrea.duringer, Kondr, Mathmum

Redefine Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8396 *Contributors*: Kondr, Mathmum

Tool Creation Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11169 *Contributors*: Kondr, Mathmum

Keyboard Shortcuts *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=16122 *Contributors*: Andrea.duringer, Kondr, Mathmum, Murkle

Options Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=8717 *Contributors*: Corinna, Kondr, Mathmum

Virtual Keyboard *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5866 *Contributors*: Kondr, Mathmum

Tool Manager Dialog *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=10981 *Contributors*: Andrea.duringer, Kondr

Accessibility *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11660 *Contributors*: Corinna, Kondr

GeoGebraPrim *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11305 *Contributors*: Corinna, Kondr

Creating Pictures of the Graphics View *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=6229 *Contributors*: Kondr, Mathmum

Upload to GeoGebraTube *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12671 *Contributors*: Florian Sonner, Kondr

Export as html Webpage *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12382 *Contributors*: Christina.biermair

Embedding to CMS, VLE (Moodle) and Wiki *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=11998 *Contributors*: Kondr, Mathmum, Murkle

Export to LaTeX (PGF, PSTricks) and Asymptote *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=12364 *Contributors*: Christina.biermair, Kondr, Mathmum

Printing Options *Source*: http://wiki.geogebra.org/s/en/index.php?oldid=5854 *Contributors*: Kondr, Mathmum

Image Sources, Licenses and Contributors

Image:Tool New Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_New_Point.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Regular Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Regular_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Vector from Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Vector_from_Point.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Vector between Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Vector_between_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Move.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Slider.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Slider.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Insert Text.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Text.gif *License*: unknown *Contributors*: Kondr, Noel Lambert

Image:Tool Check Box to Show Hide Objects.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Check_Box_to_Show_Hide_Objects.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Move.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Animate Pause.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Animate_Pause.png *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Animate Play.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Animate_Play.png *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Trace_On.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Trace_On.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Refresh.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Refresh.png *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Show Hide Object.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Show_Hide_Object.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Show Hide Label.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Show_Hide_Label.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Show_Hide_Object.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Show_Hide_Object.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Bulbgraph.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Bulbgraph.png *License*: Public Domain *Contributors*: Administrator, Pegasusroe

File:Tool Move.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool_Record_to_Spreadsheet.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Record_to_Spreadsheet.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Rotate around Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Rotate_around_Point.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_New_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_New_Point.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool_New_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_New_Point.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Attach Detach Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Attach_Detach_Point.gif *License*: unknown *Contributors*: Noel Lambert

File:Tool_Complex_Number.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Complex_Number.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Point in Region.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Point_in_Region.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool_Intersect_Two_Objects.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Intersect_Two_Objects.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Midpoint or Center.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Midpoint_or_Center.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Line_through_Two_Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Line_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Vector from Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Vector_from_Point.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Ray through Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Ray_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Segment with Given Length from Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Segment_with_Given_Length_from_Point.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Line through Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Line_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Segment between Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Segment_between_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Vector between Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Vector_between_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Perpendicular_Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Perpendicular_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Fit Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Fit_Line.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool_Parallel_Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Parallel_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Angular Bisector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Angular_Bisector.gif *License*: GNU General Public License *Contributors*: Administrator, Kondr

File:Tool Perpendicular Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Perpendicular_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool_Tangents.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Tangents.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Polar or Diameter Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Polar_or_Diameter_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Perpendicular Bisector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Perpendicular_Bisector.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Administrator, Florian Sonner, Spam cleanup script

Image:Tool_Locus.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Locus.gif *License*: GNU General Public License *Contributors*: Kondr

File:Attention.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Attention.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Florian Sonner, Pegasusroe

Image:Tool_Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Rigid Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Rigid_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr, Noel Lambert

File:Tool Regular Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Regular_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Circle_Center_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Center_Point.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool Circle Center Radius.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Center_Radius.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Circle 3Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_3Points.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool_Circle_Center_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Center_Point.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool Circumcircular Arc 3Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circumcircular_Arc_3Points.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Circumcircular Sector 3Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circumcircular_Sector_3Points.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool_Compasses.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Compasses.gif *License*: GNU General Public License *Contributors*: Administrator, Noel Lambert

File:Tool Circle Sector Center 2Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Sector_Center_2Points.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Semicircle through Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Semicircle_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Circle Arc Center 2Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Arc_Center_2Points.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool_Ellipse.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Ellipse.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool_Ellipse.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Ellipse.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Hyperbola.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Hyperbola.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Conic 5Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Conic_5Points.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool_Parabola.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Parabola.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Angle.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Angle.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool Distance.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Distance.gif *License*: GNU General Public License *Contributors*: K Voss

File:Tool Angle.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Angle.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool Slope.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Slope.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Area.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Area.gif *License*: GNU General Public License *Contributors*: Administrator

File:Tool Angle Fixed.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Angle_Fixed.gif *License*: GNU General Public License *Contributors*: Administrator

Image:Tool_Reflect_Object_in_Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Reflect_Object_in_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Translate Object by Vector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Translate_Object_by_Vector.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Reflect Object in Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Reflect_Object_in_Line.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Reflect Object in Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Reflect_Object_in_Point.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Rotate Object around Point by Angle.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Rotate_Object_around_Point_by_Angle.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Reflect Object in Circle.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Reflect_Object_in_Circle.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Dilate from Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Dilate_from_Point.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool_Insert Text.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Text.gif *License*: unknown *Contributors*: Kondr, Noel Lambert

File:Tool Insert Image.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Image.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Insert Image.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Image.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Menu Properties.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Properties.png *License*: GNU General Public License *Contributors*: Kondr

File:Tool Probability Calculator.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Probability_Calculator.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Pen.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Pen.gif *License*: unknown *Contributors*: Kondr, Spanish1

File:Tool Slider.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Slider.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Slider.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Slider.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Relation between Two Objects.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Relation_between_Two_Objects.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Function Inspector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Function_Inspector.gif *License*: GNU General Public License *Contributors*: Kondr, Noel Lambert

File:Tool Insert Text.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Text.gif *License*: unknown *Contributors*: Kondr, Noel Lambert

File:Tool Check Box to Show Hide Objects.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Check_Box_to_Show_Hide_Objects.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Insert Textfield.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Textfield.gif *License*: GNU General Public License *Contributors*: Kondr, Noel Lambert

File:Tool Insert Button.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Button.gif *License*: GNU General Public License *Contributors*: Kondr, Noel Lambert

Image:Tool_Move_Graphics_View.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move_Graphics_View.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Menu New.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_New.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Save.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Save.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Open.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Open.png *License*: GNU General Public License *Contributors*: Kondr

File:Tool Show Hide Label.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Show_Hide_Label.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Zoom Out.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Zoom_Out.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Zoom In.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Zoom_In.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Delete.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Delete.gif *License*: GNU General Public License *Contributors*: K Voss

Image: Menu_Undo.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Undo.png *License*: GNU General Public License *Contributors*: Kondr

File:Tool Move Graphics View.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move_Graphics_View.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Show Hide Object.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Show_Hide_Object.gif *License*: GNU General Public License *Contributors*: Kondr

File:Tool Copy Visual Style.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Copy_Visual_Style.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Circle_Arc_Center_2Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Arc_Center_2Points.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Circle_Sector_Center_2Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circle_Sector_Center_2Points.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Circumcircular_Arc_3Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circumcircular_Arc_3Points.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Circumcircular_Sector_3Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Circumcircular_Sector_3Points.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Distance.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Distance.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Intersect Two Objects.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Intersect_Two_Objects.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Perpendicular_Bisector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Perpendicular_Bisector.gif *License*: GNU General Public License *Contributors*: Kondr

image : Tool_Midpoint_or_Center.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Midpoint_or_Center.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Attach_Detach_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Attach_Detach_Point.gif *License*: unknown *Contributors*: Noel Lambert

Image:Tool Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Regular_Polygon.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Regular_Polygon.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Ray through Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Ray_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Segment_between_Two_Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Segment_between_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Segment_with_Given_Length_from_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Segment_with_Given_Length_from_Point.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Slope.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Slope.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Tangents.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Tangents.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Parabola.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Parabola.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Polar_or_Diameter_Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Polar_or_Diameter_Line.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Semicircle through Two Points.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Semicircle_through_Two_Points.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool_Dilate_from_Point.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Dilate_from_Point.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool_Rotate_Object_around_Point_by_Angle.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Rotate_Object_around_Point_by_Angle.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Translate Object by Vector.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Translate_Object_by_Vector.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Fit Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Fit_Line.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Tool Insert Textfield.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Insert_Textfield.gif *License*: GNU General Public License *Contributors*: Kondr, Noel Lambert

Image:Views.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Views.png *License*: Public Domain *Contributors*: Kondr, Pegasusroe

Image:Tool Reflect Object in Line.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Reflect_Object_in_Line.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Move Graphics View.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Move_Graphics_View.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Zoom In.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Zoom_In.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Zoom Out.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Zoom_Out.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Axes.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Axes.gif *License*: GNU General Public License *Contributors*: Kondr

Image:grid.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Grid.gif *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Auxiliary.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Auxiliary.png *License*: GNU General Public License *Contributors*: Kimeswenger, Kondr, Pegasusroe

Image:Tree.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tree.png *License*: GNU General Public License *Contributors*: Kimeswenger, Kondr

Image:Copy to Input Bar.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Copy_to_Input_Bar.png *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Navigation_Skip_Back.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Navigation_Skip_Back.png *License*: GNU General Public License *Contributors*: Kondr

Image:Navigation_Rewind.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Navigation_Rewind.png *License*: GNU General Public License *Contributors*: Kondr

Image:Navigation_Fast_Forward.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Navigation_Fast_Forward.png *License*: GNU General Public License *Contributors*: Kondr

Image:Navigation_Skip_Forward.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Navigation_Skip_Forward.png *License*: GNU General Public License *Contributors*: Kondr

Image:Animate_Play.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Animate_Play.png *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Table.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Table.gif *License*: GNU General Public License *Contributors*: Kondr

Image:export_small.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Export_small.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

File:text-html.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Text-html.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

File:image-x-generic.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Image-x-generic.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair, Kondr

File:edit-copy.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Edit-copy.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

Image:Menu Print Preview.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Print_Preview.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Close.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Close.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Undo.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Undo.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Redo.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Redo.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Copy.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Copy.png *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Delete.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Delete.gif *License*: GNU General Public License *Contributors*: K Voss

Image:Menu_Axes.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Axes.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Grid.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Grid.gif *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Menu Point Capturing.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Point_Capturing.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Font.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Font.png *License*: unknown *Contributors*: Kondr

Image:Menu Language.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Language.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Create Tool.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Create_Tool.png *License*: GNU General Public License *Contributors*: Kondr

Image:Pdf.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Pdf.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Help.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Help.png *License*: GNU General Public License *Contributors*: Kondr

Image:Menu Forum.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Forum.png *License*: GNU General Public License *Contributors*: Kondr

Image:GeoGebra 48.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:GeoGebra_48.png *License*: GNU General Public License *Contributors*: Kondr, Pegasusroe

Image:Menu About.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_About.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Rename.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Rename.png *License*: GNU General Public License *Contributors*: Kondr

Image:Trace On.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Trace_On.gif *License*: GNU General Public License *Contributors*: Kondr

Image:Tool Record to Spreadsheet.gif *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Tool_Record_to_Spreadsheet.gif *License*: GNU General Public License *Contributors*: Kondr

Image:image-x-generic.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Image-x-generic.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair, Kondr

File:export.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Export.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

File:Export-html.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Export-html.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

Image:Menu_Create_Tool.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Menu_Create_Tool.png *License*: GNU General Public License *Contributors*: Kondr

Image:Export.png *Source*: http://wiki.geogebra.org/s/en/index.php?title=File:Export.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Christina.biermair

License