

COMPARISON OF SKETCHPAD AND GEOGEBRA



By Scott Steketee

Key Curriculum Press

Dynamic mathematics software can make a significant difference in student learning and understanding, and can enhance a teacher's ability to demonstrate mathematical concepts. To achieve these goals, the software must be mathematically accurate, easy to use, and pedagogically well-designed. This document highlights the most important mathematical, interface, and pedagogical differences between The Geometer's Sketchpad Version 5.01 (released in May 2010) and GeoGebra 3.2.44.0 (released in July 2010).

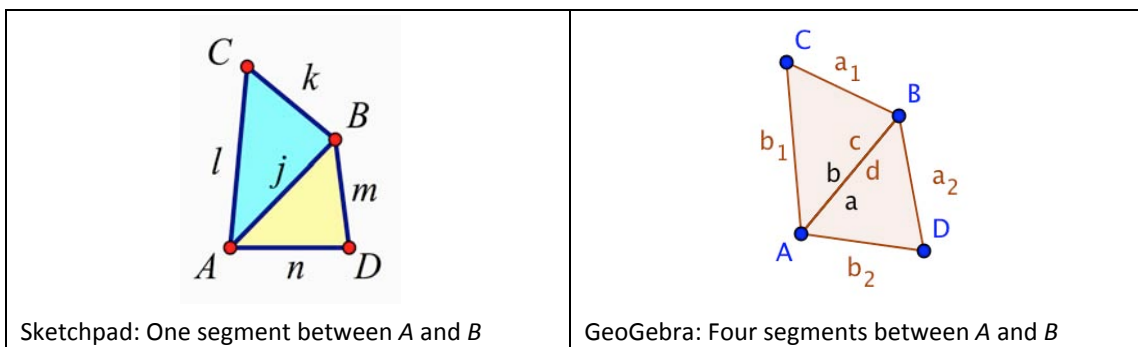
These highlights are based on the current versions of the programs as of August 2010, and are grouped into the following sections:

- Geometry
- Algebra
- Algebra/Geometry Integration
- Mathematical Communication
- Usability
- Support
- Summary

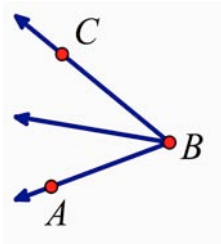
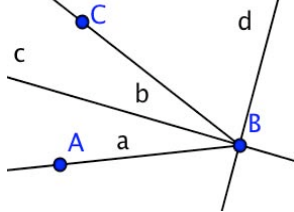
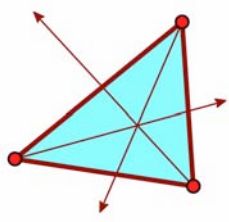
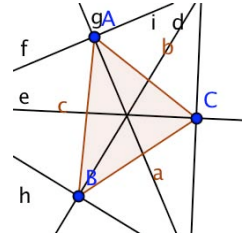
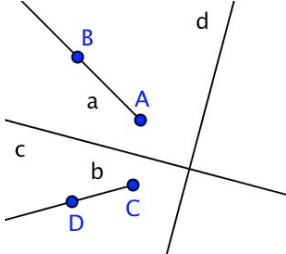
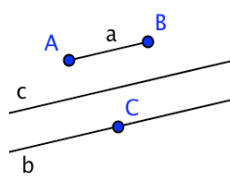
Geometry

Sketchpad's constructions conform to Euclidean norms and to the geometry taught in K–12 classrooms; GeoGebra differs significantly from those norms in several respects.

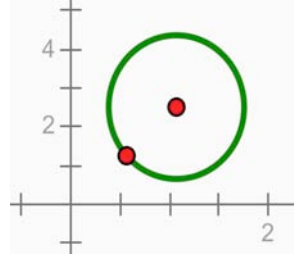
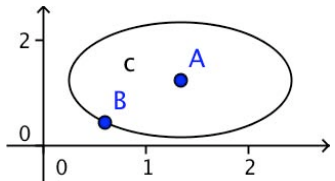
In Sketchpad there's a single line (or segment) defined by two points (for instance, segment j in the illustration below left). In GeoGebra (below right) segments a , b , c , and d are distinct: one is from A to B , one is from B to A , one is an edge of triangle ABC , and one is an edge of triangle ABD .



In Sketchpad an angle bisector is a ray defined by a simple (non-reflex) angle. The angle can be specified by three points, with the vertex second, or by two rays or segments sharing an endpoint. In either case, the vertex must exist. GeoGebra adopts a non-traditional approach to angle bisectors. An angle bisector is defined as a line, not a ray, and GeoGebra constructs angle bisectors even when a vertex does not exist.

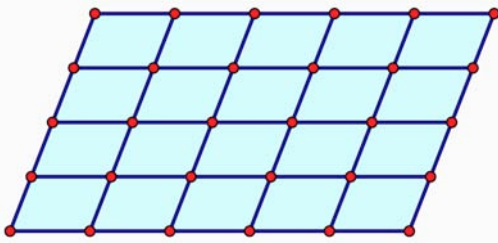
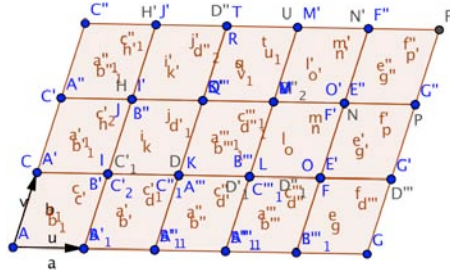
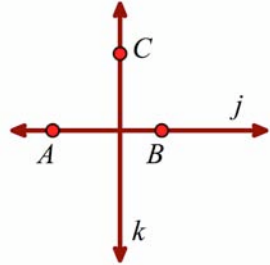
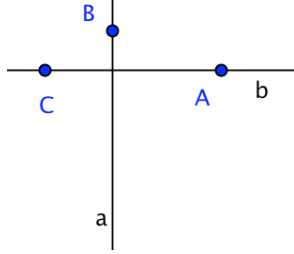
 <p>Sketchpad: An angle bisector of two rays</p>	 <p>GeoGebra: Perpendicular lines c and d are the "angle bisectors" of the two rays a and b.</p>
 <p>Sketchpad: Angle bisectors of a triangle</p>	 <p>GeoGebra: Angle bisectors of a triangle</p>
 <p>GeoGebra: Lines c and d are the "angle bisectors" of rays a and b.</p>	 <p>GeoGebra: Line c is the "angle bisector" of segment a and parallel line b.</p>

Sketchpad's geometric objects don't require a coordinate system; the size and shape of a geometric object is independent of the coordinate axes. GeoGebra's geometric objects are automatically defined in terms of GeoGebra's coordinate system; the size and shape of a geometric object changes if the axes are rescaled.

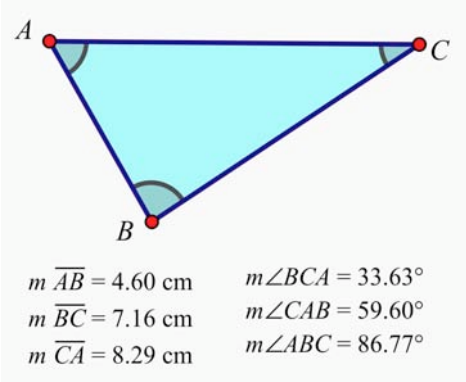
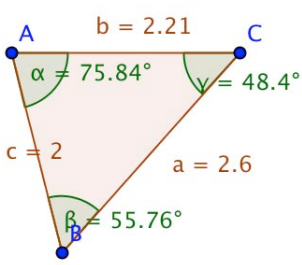
 <p>Sketchpad: The plane remains Euclidean. A circle remains circular even when axes are scaled.</p>	 <p>GeoGebra: Scaling the axes scales the plane, turning a Euclidean circle into an ellipse</p>
---	---

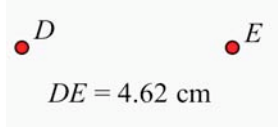
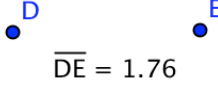
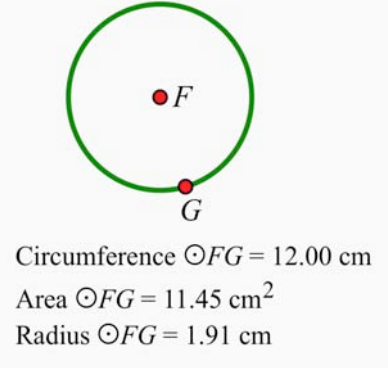
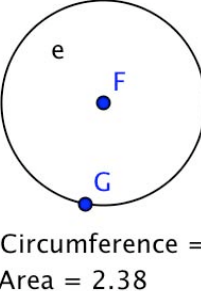
Sketchpad creates labels for geometric objects as labels are needed for measurements or requested by the user; GeoGebra creates labels as objects are constructed, creating visual clutter and rapidly exhausting simple one-letter labels. (Though it's possible to change GeoGebra's preferences to avoid displaying the labels, they are always generated whether displayed or not.) Sketchpad's labels can be dragged to any appropriate location close to

the object being labeled; GeoGebra's labels for non-point objects are confined to circular regions that simultaneously allow an object's label to stray far from the object being labeled while preventing it from being placed near other parts of the object.

 <p>Sketchpad: Parallelogram tessellation produces no labels until they're needed.</p>	 <p>GeoGebra: Parallelogram tessellation produces a proliferation of labels.</p>
 <p>Sketchpad: Labels can be dragged to any desired position near the labeled object.</p>	 <p>GeoGebra: These labels have been dragged. The points are actually A, C, and B from left to right, the horizontal line is a, and the vertical line is b.</p>

Sketchpad's geometric measurements are consistent, clearly labeled, and simple for students to understand; GeoGebra's measurements omit units for lengths and areas, often fail to identify the object being measured, include the names of objects that have not been created, conflate the name of an object with its measurement, and are inconsistent in displaying precision (by dropping trailing zeros).

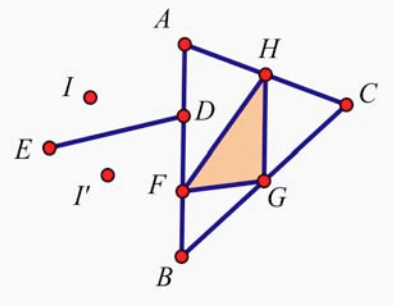
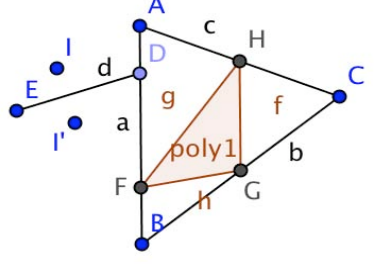
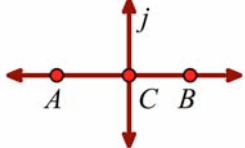
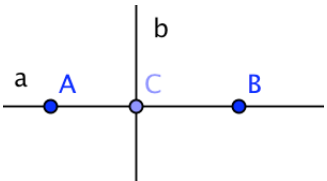

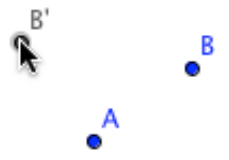
 <p>Sketchpad: Triangle measurements</p> <p> $m \overline{AB} = 4.60 \text{ cm}$ $m \angle BCA = 33.63^\circ$ $m \overline{BC} = 7.16 \text{ cm}$ $m \angle CAB = 59.60^\circ$ $m \overline{CA} = 8.29 \text{ cm}$ $m \angle ABC = 86.77^\circ$ </p>	 <p>GeoGebra: Triangle measurements. Without trailing zeros, measurements fail to indicate precision.</p> <p> $b = 2.21$ $c = 2$ $a = 2.6$ $\alpha = 75.84^\circ$ $\beta = 55.76^\circ$ $\gamma = 48.4^\circ$ </p>
--	---

 <p>Sketchpad: The distance between two points.</p>	 <p>GeoGebra: The distance between two points is formatted as though it were a segment.</p>
 <p>Sketchpad: Circle measurements</p>	 <p>GeoGebra: Circle measurements. (The radius can be measured only from the command line; the result appears in the algebra window.)</p>

Because dragging of objects is central to dynamic mathematics software, Sketchpad allows the user to drag any object or set of objects in a way that's both intuitive and mathematically revealing. In the context of dragging, "intuitive" means that all selected object(s) follow the motion of the pointer, and "revealing" means that the motion reveals as much of the mathematics as possible, by affecting a minimal set of unselected objects. A revealing drag allows the user to see relationships by distinguishing the variant objects from the invariant ones. (For instance, a Sketchpad user can drag a reflected image to see how the pre-image must move to produce any desired movement of the image.)

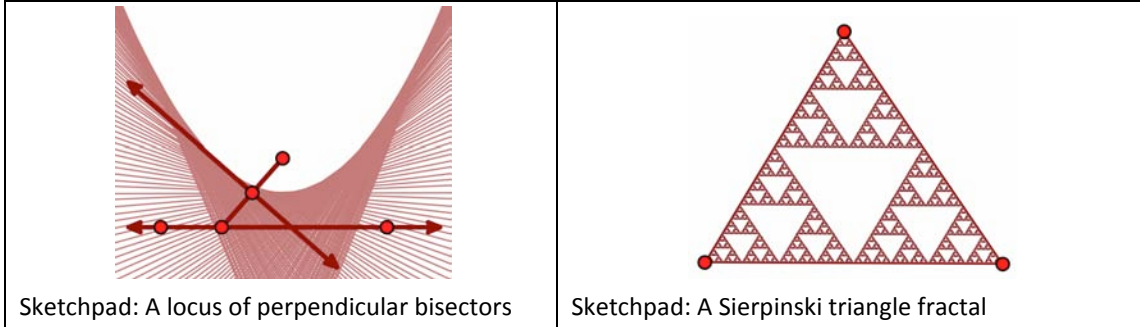
Sketchpad's design encourages students to use the "drag test," investigating a construction and testing its integrity by dragging different parts of it. The drag test encourages students to be self-reliant users. By dragging, they form and immediately test various conjectures, and assess their work as they proceed.

GeoGebra allows some objects to be dragged, but not others. The user can drag independent points, but not intersection points or segment midpoints. The user can drag a limited number of other constructed objects, such as an entire polygon or one or more vertices of a polygon, but not a side of a polygon. The user can drag a segment constructed from independent points, but not a segment with an endpoint attached to another object. Dragging a combination of related objects is likely to produce strange behavior, discouraging students from dragging anything other than independent points and objects constructed from independent points. With dragging allowed for some objects and forbidden for others, students are discouraged from dragging, severely limiting their ability to manipulate and investigate the mathematics embodied in the construction..

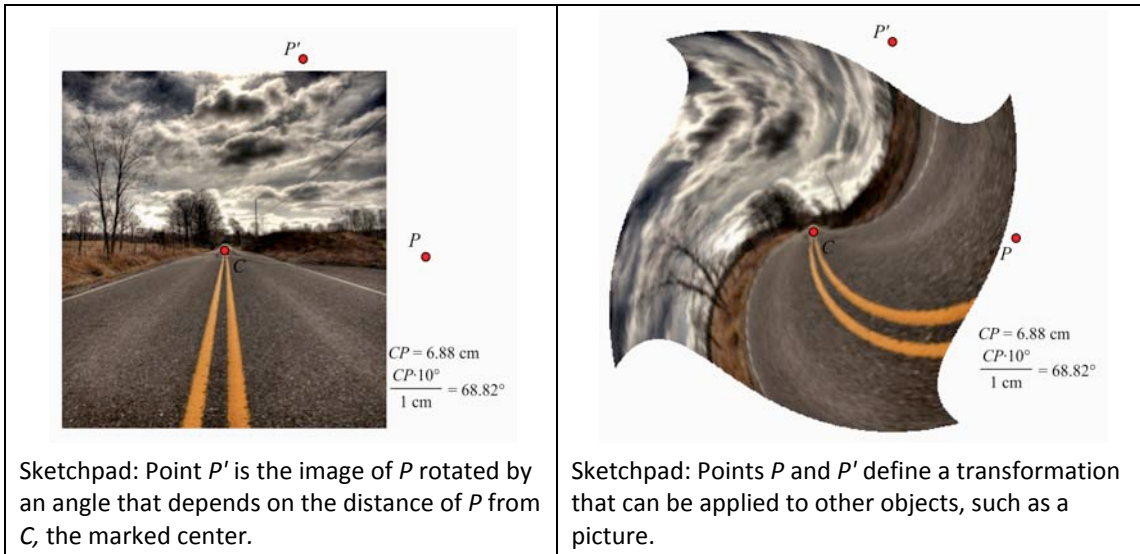
 <p>Sketchpad: Every object, even the reflected image point I', can be dragged.</p>	 <p>GeoGebra: Segments d, f, g, and h, midpoints G and H, the polygon, and reflected image I' cannot be dragged.</p>
 <p>Sketchpad: Dragging line j moves point C along segment AB, revealing that C is moveable and that j is perpendicular to segment AB.</p>	 <p>GeoGebra: Line b cannot be dragged.</p>
 <p>Sketchpad: Dragging rotated image point A' while tracing reveals its relationship to A.</p>	 <p>GeoGebra: Point B' cannot be dragged.</p>

Animation is another important way to investigate the behavior of geometric constructions. Sketchpad treats animation like dragging; it allows the user to animate any geometric object by animating the points on which the object is based. In GeoGebra, animation is limited to sliders.

Sketchpad supports loci for most geometric objects (including pictures); GeoGebra supports loci of points only, provided the point is not constructed on a path. Sketchpad supports a family of loci; GeoGebra does not. Sketchpad supports the iteration of geometric objects (enabling easy construction of fractals like the Sierpinski triangle); GeoGebra does not.



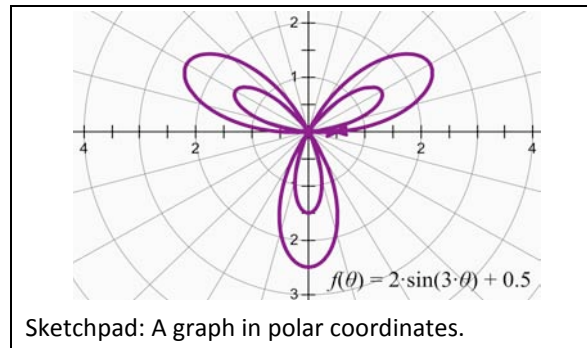
Sketchpad supports an unlimited set of transformations (isometries, similarities, affinities, and custom transformations defined by any construction that defines one point in terms of another), and allows the user to apply transformations to a wide variety of pre-image objects, including pictures. GeoGebra's transformations are limited to the similarity transformations, point reflections, and circle inversions. Thus, transformations can be explored much more widely with Sketchpad.



Algebra

Sketchpad has a robust range of features that support algebraic representations and understanding.

Multiple coordinate systems are available in Sketchpad, and useful when showing two graphs at once, particularly when they require different scales. Sketchpad also supports polar grids and polar functions. GeoGebra supports neither polar coordinates nor multiple coordinate systems.



Sketchpad's trigonometric axis labeling adjusts appropriately as the axis is scaled. GeoGebra's trigonometric axis labeling is very limited.

<p>Sketchpad: Trigonometric axis adjusts numbering appropriately at small scale.</p>	<p>GeoGebra: Trigonometric axis cannot show finer numbering at small scale.</p>
<p>Sketchpad: Trigonometric axis adjusts numbering appropriately at large scale.</p>	<p>GeoGebra: Trigonometric axis numbering is erratic for large scales.</p>

In Sketchpad the scale of an axis can either be determined by a geometric measurement, or can be adjusted freely by dragging one of the axis numbers. In GeoGebra the scale of an axis cannot be determined by other objects in the sketch, and is more difficult to adjust.

Sketchpad allows the user to graph y as a function of x , x as a function of y , r as a function of θ , and θ as a function of r . GeoGebra supports only graphing y as a function of x .

Only Sketchpad allows the display of a dotted grid. Combined with snapping of points, such a grid is useful for geoboard activities.

Sketchpad measures the equations of lines in standard slope-intercept form. GeoGebra uses a modified two-point form that depends upon the particular points used to define the line: $(y_B - y_A)x - (x_B - x_A)y = x_A y_B - y_A x_B$. One consequence is that the equation of the line from A to B appears different from that of the line from B to A . Below are the equations from each program of the lines defined by three collinear points. (The defining points for the six GeoGebra lines are A and B , B and A , A and C , C and A , B and C , and C and B .)

$\begin{aligned} \overleftrightarrow{AB}: y &= 0.39x + 4.84 \\ \overleftrightarrow{AC}: y &= 0.39x + 4.84 \\ \overleftrightarrow{BC}: y &= 0.39x + 4.84 \end{aligned}$	<ul style="list-style-type: none"> ● a: $-0.58x + 1.12y = 6.41$ ● b: $0.58x - 1.12y = -6.41$ ● c: $0.17x - 0.33y = -1.91$ ● d: $-0.17x + 0.33y = 1.91$ ● e: $0.75x - 1.45y = -8.32$ ● f: $-0.75x + 1.45y = 8.32$
Sketchpad: Equations of all lines defined by three collinear points A, B, and C	GeoGebra: Equations of all lines defined by three collinear points A, B, and C

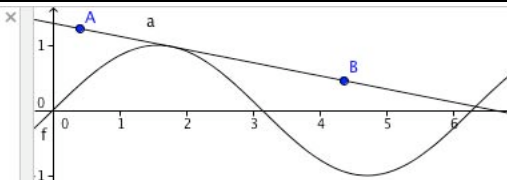
Sketchpad uses a robust method for finding intersections of objects such as function plots, straight objects, circles, and arcs. Intersections, even multiple intersections, are found reliably regardless of the type of function. GeoGebra's Intersection tool is limited to finding a single intersection at a time for functions that are not polynomials, and even then uses a fragile algorithm that sometimes fails to find existing intersections.

Free Objects

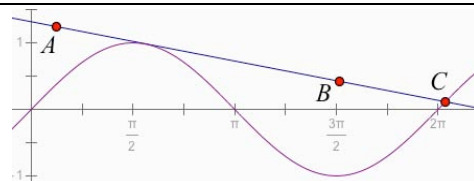
- A = (0.4, 1.26)
- B = (4.36, 0.46)
- f(x) = sin(x)

Dependent Objects

- C undefined
- a: $0.8x + 3.96y = 5.31$

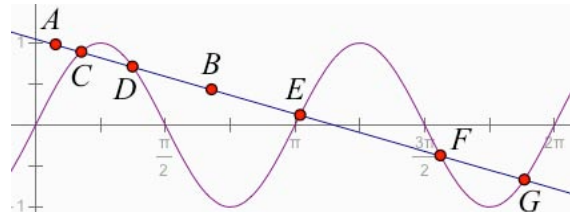


GeoGebra's intersection finder gets stuck when line *a* is adjusted to approach but no longer touch the sine curve. GeoGebra reports that intersection C is undefined, even though an intersection exists.

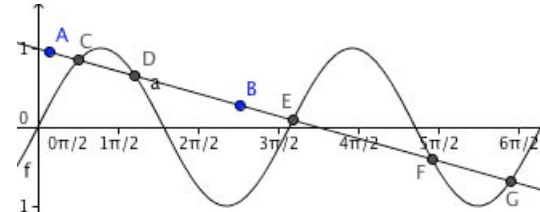


Sketchpad's robust intersection finder finds the first intersection without being misled by close approaches.

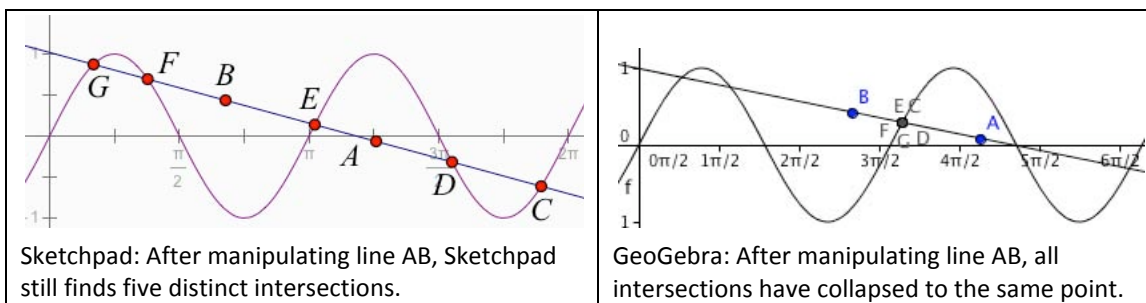
When two objects have multiple intersections. Sketchpad keeps the intersections distinct when possible. GeoGebra fails to track the intersections separately. In the example below, five intersections have been constructed. Provided at least five intersections exist, Sketchpad keeps them distinct; with GeoGebra, manipulation of the line causes the five intersections to collapse to a single point.



Sketchpad: Before manipulating line AB.



GeoGebra: Before manipulating line AB.



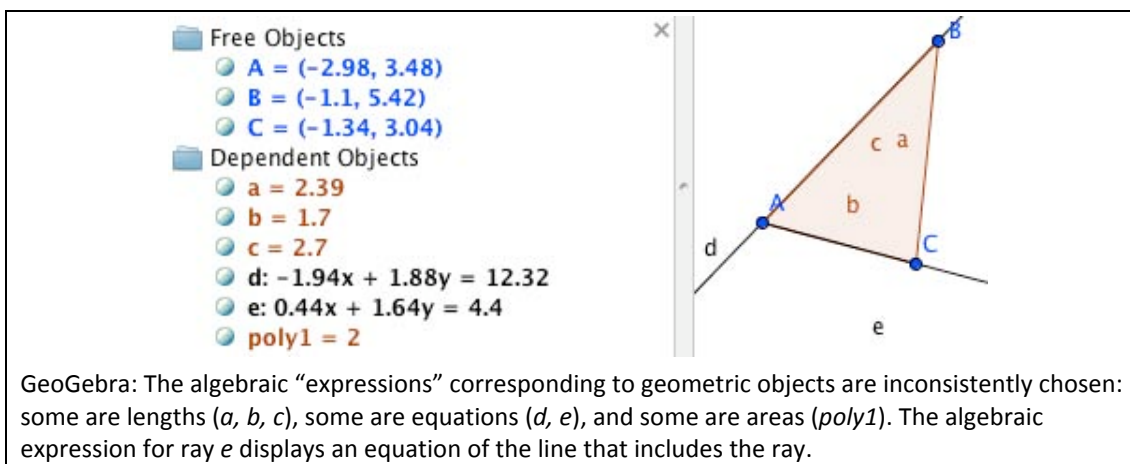
Algebra/Geometry Integration

Sketchpad uses sound mathematical and pedagogical principles for its algebra/geometry integration, allowing full two-way interaction between geometric and algebraic elements, including the coordinate system.

The user can define one or more coordinate systems geometrically, based on a geometric point as the origin and with the horizontal and vertical scales determined by unit points, by one or two lengths or distance measurements, or by a unit circle. The resulting coordinate system can be easily dragged and rescaled, and its axes are geometric lines.

GeoGebra's single coordinate system is created automatically whether needed or not, and has a fixed origin and scales that cannot be altered by geometric objects. The axes are lines in some respects, but they cannot be selected or dragged like other geometric objects, they have no object properties, and they do not appear in the algebra window's list of objects. This inflexibility is fundamental to GeoGebra's design, because GeoGebra uses its coordinate system to define its Euclidean plane.

GeoGebra separates the display into a "geometry window" and an "algebra window," claiming that every "expression in the algebra window corresponds to an object in the geometry window and vice versa." These windows are better described as a "graphics window" and a "list window." The list window includes a measurement of a property of each object. However, these measurements are inconsistent and not necessarily clear—some of the displayed properties are algebraic (equations) and some are more commonly considered geometric (length or area). And again, length and area measurements do not display units.



Free Objects

- A = (-3.76, 5.72)
- B = (-1.76, 4.76)
- C = (-3.68, 3.14)

Dependent Objects

- D = (-1.95, 3.68)
- D' = (-2.85, 4.7)
- c = 5.64
- d = 1.42
- f = 8.22
- $\alpha = 0^\circ$

GeoGebra: Not all objects appear in both windows. Locus *loc1* is missing from the algebra window, and measurements *d*, *f*, and α do not appear in the geometry window. Arc *c*'s expression is the arc's length, and *d* measures its radius. Measurement *f* is defined as the arc's "circumference" and measurement α as the arc's "angle." It's unclear what either of these expressions actually measures.

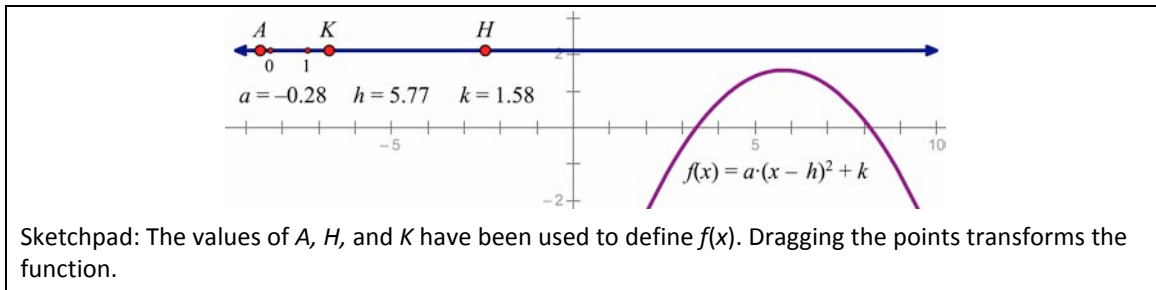
Sketchpad avoids an imposed decision as to what measurement takes precedence for each type of object, and displays all measurements, with units, in a consistent way in the single sketch window. All measurements appear in a similar format, identify the objects they measure, and can be dragged wherever the user desires. (Measurements can also be attached to the objects they measure.)

Iteration and locus constructions provide another example of Sketchpad's flexible and useful integration of the geometric and algebraic realms; these constructions apply equally to geometric and algebraic objects. A locus can be driven geometrically by a point, or algebraically by a parameter. Similarly, an iteration can iterate any combination of points and parameters, to a depth controlled manually or by a parameter. These constructions with Sketchpad enable the exploration of families of functions, fractals, Taylor series visualizations, iterated function systems, and so forth.

Finally, Sketchpad integrates algebra and geometry by allowing any geometric path to be used as an algebraic "axis." For example, the user can draw a segment, construct a point on the segment, and then measure the "value" of the point based on its position along the segment. (Sketchpad defines the values of the segment's endpoints as 0 and 1 and the value of the point as a number that varies from 0 to 1 as it is dragged along the path.) Similarly, Sketchpad allows the user to plot a value on any path.

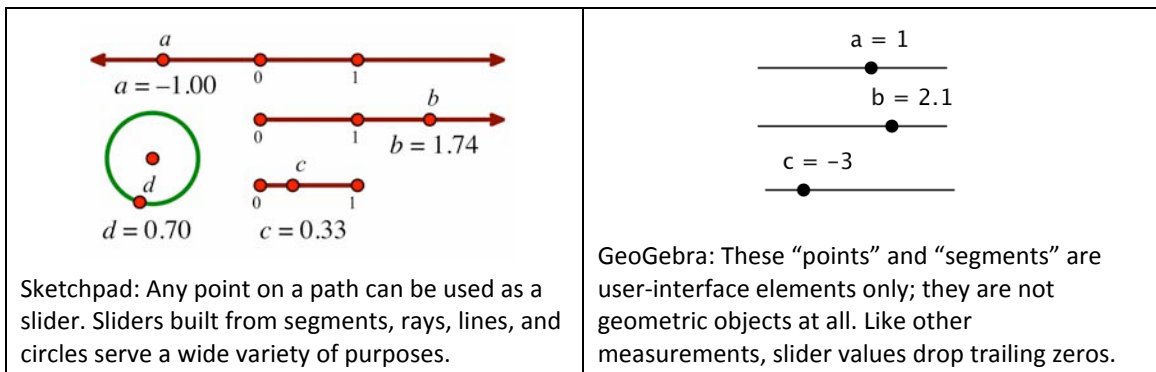
GeoGebra's coordinates are limited to those defined by the underlying axes; it provides no general way to connect geometry and algebra by measuring position on a path or plotting a point on a path. .

<p>$C \text{ on } \overline{AB} = 0.64$</p>	<p>$a = -0.83 \quad h = 5.15 \quad k = 1.65$</p>
<p>Sketchpad: The value of point <i>C</i> is measured on segment <i>AB</i>. (The defining points of the segment have values of 0 and 1.)</p>	<p>Sketchpad: The values of <i>A</i>, <i>H</i>, and <i>K</i> have been measured on the line. (For convenience, the line's defining points have been relabeled as 0 and 1.)</p>



Because every position on a geometric path is associated with an algebraic value, the user can turn any Sketchpad path into a slider by constructing and measuring a point on the path.

This form of integration between geometry and algebra is missing from GeoGebra. Sliders in GeoGebra appear in the geometry window, but neither the point nor the segment can be used in geometric constructions.



All of these features contribute to Sketchpad’s effective integration of the geometric and algebraic realms in a way that’s consistent, powerful, and under full user control.

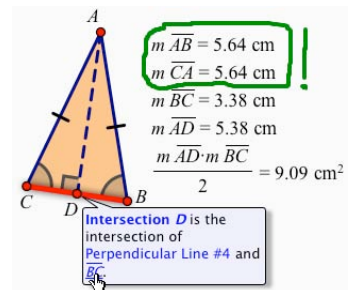
Mathematical Communication

Sketchpad places a strong emphasis on users’ ability to express themselves mathematically and to present their thinking to others. There are three aspects of the way users express themselves: in the mathematical construction itself, in their writing about their mathematical thinking, and in their presentation of mathematical investigations to others. Sketchpad provides a suite of features to support all three forms of expressivity and to facilitate demonstrations using interactive whiteboards.

Expressivity in constructions

Sketchpad provides a number of capabilities to enhance the mathematical expressivity of constructions.

- The **Marker** tool allows freehand drawing, creates crossbars and arrowheads on paths, creates angle markers at vertices of angles, and permits freehand drawing of function graphs



- The **Information** tool can be clicked on any object to see a description of the object and links to related objects. Pointing at any link (as in the illustration above) highlights the linked object, making it easy to understand the structure of a construction and to determine which objects depend on which other objects.
- Measurements, calculations, and functions appear with proper mathematical formatting, including appropriately formatted fractions, italicization, and so forth.
- Angles measured in radians are displayed using multiples and fractions of π when appropriate.

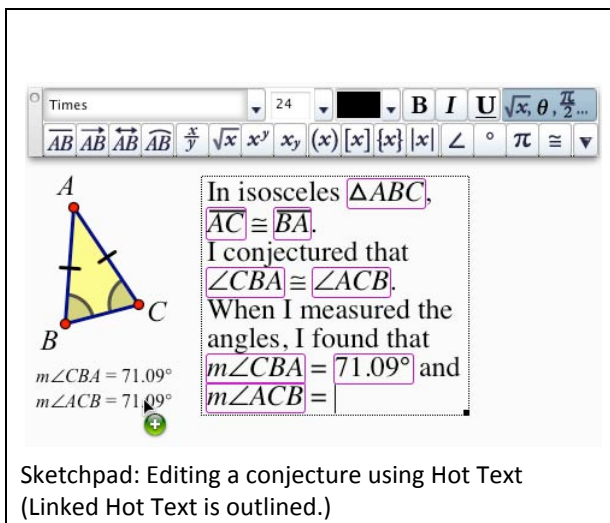
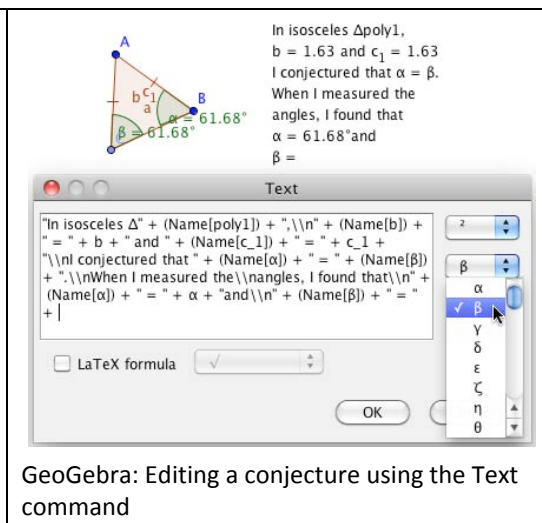
GeoGebra includes none of these features.

$$m\angle CDA = \frac{\pi}{2} \text{ radians}$$

Expressivity in writing

Sketchpad provides exceptionally strong support for mathematical writing. Hot Text™ makes it easy for users to write about their mathematical thinking, their conjectures, and their observations: while typing a caption, the user clicks an object in the sketch to incorporate its label or value into the caption. The resulting reference is “live,” updating as the value or label changes and highlighting the object in the sketch when the mouse passes over the reference. The user can also incorporate mathematical formatting, including overbars, superscripts, subscripts, fractions, and square root signs.

In GeoGebra, users must employ the Text command and use complex syntax to enter similar text. Errors are likely, and students may find correcting them difficult. Italicization, overbars, fractions, square roots, and characters such as the angle and congruence symbols require the use of the document markup language LaTeX.

 <p>Sketchpad: Editing a conjecture using Hot Text (Linked Hot Text is outlined.)</p>	 <p>GeoGebra: Editing a conjecture using the Text command</p>
--	---

Expressivity in presentation

Sketchpad provides users with powerful ways to present their work, including multi-page documents and action buttons. With multiple pages a user can create a sequence of related constructions that tell a mathematical story; they are a perfect vehicle for creating a multi-stage activity or a student project. Six types of action buttons support effective presentations:

- **Hide/Show buttons** hide or show objects or collections of objects, revealing a sequence of steps, showing construction lines, or reducing clutter.

- **Animation buttons** animate any combination of geometric objects and parameters, revealing how changing chosen elements of the sketch affects other elements.
- **Movement buttons** move any combination of moveable points and changeable parameters to user-designated target points and values, showing a progression of the sketch's mathematics from one state to another.
- **Presentation buttons** combine the actions of other buttons, creating more complex presentations that use a series of steps to tell a story.
- **Link buttons** connect related constructions on different pages of a document, or connect to related content available locally or on the internet.
- **Sound buttons** play a sound defined by a mathematical function.

GeoGebra provides only a limited form of Hide/Show button.

Interactive whiteboards

Sketchpad supports presentations using interactive whiteboards in a variety of ways:

- Default points are of medium size, lines are of medium width, and fonts are large. All are clearly visible when projected, even from the back of the room.
- The **Marker** tool allows freehand drawing, creates crossbars and arrowheads on paths, creates angle markers at vertices of angles, and permits freehand drawing of function graphs.
- No special key is required to select multiple objects.
- Clear onscreen feedback immediately indicates when an object will be constructed on a path or at an intersection.
- The status line at the bottom of the screen describes the current status or the effect of clicking at the current pointer location.
- The “magnetism” preference makes the pointer more magnetic, so that it's easier to click and drag objects even if the whiteboard is slightly miscalibrated.

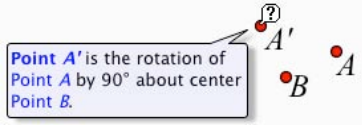
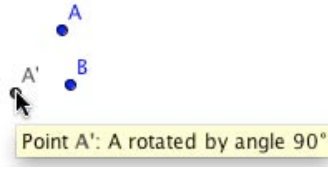
GeoGebra has none of these features.

Usability

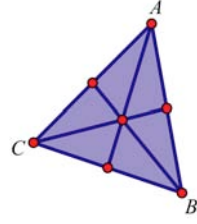
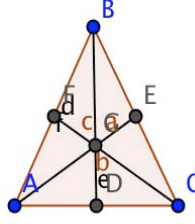
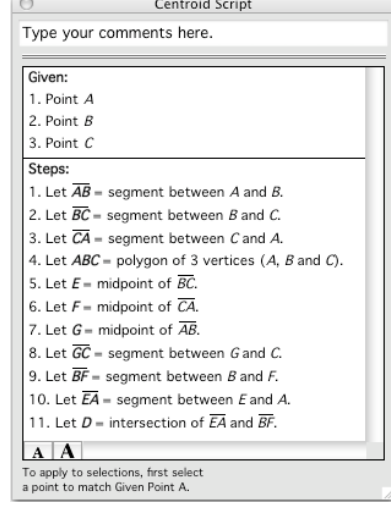
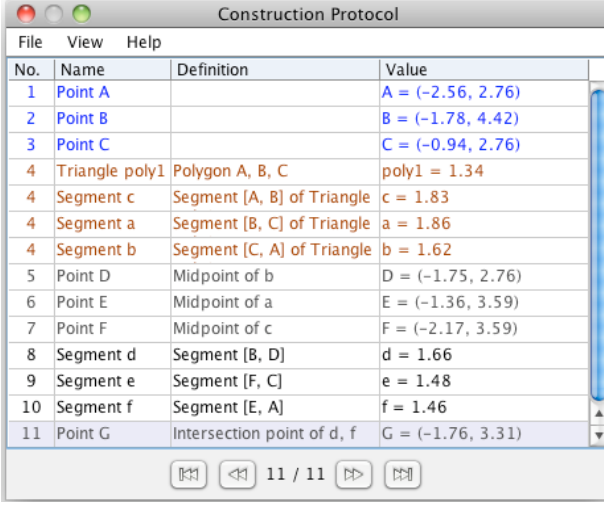
Sketchpad's design places a premium on ease of use, both for students engaged in hands-on explorations and for teachers conducting whole-class demonstrations.

Sketchpad's dragging behavior encourages students to check their work using the drag test, leading to important student behaviors such as investigation, conjecturing and testing conjectures, and self-assessment.

Sketchpad describes the relationships between objects in full sentences, with appropriate formatting (including italicization) and links to related objects. In GeoGebra the descriptions are unformatted sentence fragments without links.

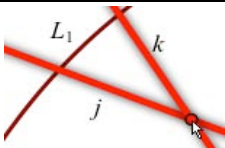
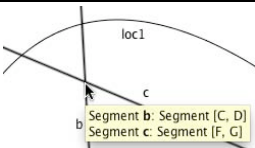
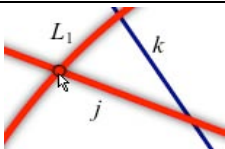
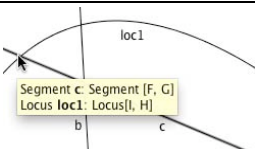
 <p>Sketchpad: Descriptions are full sentences, with links to related objects.</p>	 <p>GeoGebra: Descriptions are fragments, without links. This example fails to identify the center point of the rotation.</p>
---	---

Similarly, Sketchpad’s description of an entire construction is a “script” in full sentences and with mathematical formatting. GeoGebra’s description is a much more limited “Construction Protocol.”

																																																													
 <p>Sketchpad: A script uses sentences to describe a construction or tool.</p>	 <p>GeoGebra: A protocol for the same construction.</p> <table border="1" data-bbox="755 934 1356 1354"> <thead> <tr> <th>No.</th> <th>Name</th> <th>Definition</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Point A</td> <td></td> <td>A = (-2.56, 2.76)</td> </tr> <tr> <td>2</td> <td>Point B</td> <td></td> <td>B = (-1.78, 4.42)</td> </tr> <tr> <td>3</td> <td>Point C</td> <td></td> <td>C = (-0.94, 2.76)</td> </tr> <tr> <td>4</td> <td>Triangle poly1</td> <td>Polygon A, B, C</td> <td>poly1 = 1.34</td> </tr> <tr> <td>4</td> <td>Segment c</td> <td>Segment [A, B] of Triangle</td> <td>c = 1.83</td> </tr> <tr> <td>4</td> <td>Segment a</td> <td>Segment [B, C] of Triangle</td> <td>a = 1.86</td> </tr> <tr> <td>4</td> <td>Segment b</td> <td>Segment [C, A] of Triangle</td> <td>b = 1.62</td> </tr> <tr> <td>5</td> <td>Point D</td> <td>Midpoint of b</td> <td>D = (-1.75, 2.76)</td> </tr> <tr> <td>6</td> <td>Point E</td> <td>Midpoint of a</td> <td>E = (-1.36, 3.59)</td> </tr> <tr> <td>7</td> <td>Point F</td> <td>Midpoint of c</td> <td>F = (-2.17, 3.59)</td> </tr> <tr> <td>8</td> <td>Segment d</td> <td>Segment [B, D]</td> <td>d = 1.66</td> </tr> <tr> <td>9</td> <td>Segment e</td> <td>Segment [F, C]</td> <td>e = 1.48</td> </tr> <tr> <td>10</td> <td>Segment f</td> <td>Segment [E, A]</td> <td>f = 1.46</td> </tr> <tr> <td>11</td> <td>Point G</td> <td>Intersection point of d, f</td> <td>G = (-1.76, 3.31)</td> </tr> </tbody> </table>	No.	Name	Definition	Value	1	Point A		A = (-2.56, 2.76)	2	Point B		B = (-1.78, 4.42)	3	Point C		C = (-0.94, 2.76)	4	Triangle poly1	Polygon A, B, C	poly1 = 1.34	4	Segment c	Segment [A, B] of Triangle	c = 1.83	4	Segment a	Segment [B, C] of Triangle	a = 1.86	4	Segment b	Segment [C, A] of Triangle	b = 1.62	5	Point D	Midpoint of b	D = (-1.75, 2.76)	6	Point E	Midpoint of a	E = (-1.36, 3.59)	7	Point F	Midpoint of c	F = (-2.17, 3.59)	8	Segment d	Segment [B, D]	d = 1.66	9	Segment e	Segment [F, C]	e = 1.48	10	Segment f	Segment [E, A]	f = 1.46	11	Point G	Intersection point of d, f	G = (-1.76, 3.31)
No.	Name	Definition	Value																																																										
1	Point A		A = (-2.56, 2.76)																																																										
2	Point B		B = (-1.78, 4.42)																																																										
3	Point C		C = (-0.94, 2.76)																																																										
4	Triangle poly1	Polygon A, B, C	poly1 = 1.34																																																										
4	Segment c	Segment [A, B] of Triangle	c = 1.83																																																										
4	Segment a	Segment [B, C] of Triangle	a = 1.86																																																										
4	Segment b	Segment [C, A] of Triangle	b = 1.62																																																										
5	Point D	Midpoint of b	D = (-1.75, 2.76)																																																										
6	Point E	Midpoint of a	E = (-1.36, 3.59)																																																										
7	Point F	Midpoint of c	F = (-2.17, 3.59)																																																										
8	Segment d	Segment [B, D]	d = 1.66																																																										
9	Segment e	Segment [F, C]	e = 1.48																																																										
10	Segment f	Segment [E, A]	f = 1.46																																																										
11	Point G	Intersection point of d, f	G = (-1.76, 3.31)																																																										

Sketchpad enables all constructions, even the most powerful ones, through its graphical user interface, and by design avoids giving the user error messages. Instead, it communicates possible constructions and operations to the user by means of tool feedback and enabling/disabling of commands.

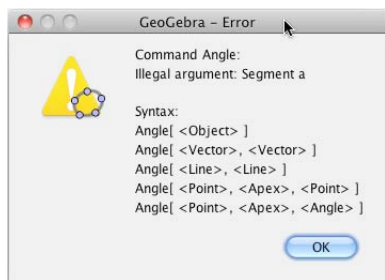
Sketchpad’s tool feedback is clearer and more visual than GeoGebra’s text-based feedback. For example, the following illustrations show the use of Sketchpad’s **Point** tool and GeoGebra’s **Intersection** tool to construct the intersection of two objects. Sketchpad provides prominent visual feedback, including a preview of the intersection point itself. GeoGebra’s feedback is text-based, and potentially misleading in cases in which the intersection construction is unsupported.

 <p>Sketchpad: Feedback when positioning the Point tool at the intersection of two segments. (This construction is supported.)</p>	 <p>GeoGebra: Feedback when positioning the Intersection tool at the intersection of two segments. (This construction is supported.)</p>
 <p>Sketchpad: Feedback when positioning the Point tool at the intersection of a segment and a locus. (This construction is supported.)</p>	 <p>GeoGebra: Feedback when positioning the Intersection tool at the intersection of a segment and a locus. (This construction is not supported.)</p>

GeoGebra’s tools can be used to construct objects that don’t exist, such as the intersection of two parallel lines or the intersection of two segments that don’t intersect. Sketchpad does not allow such impossible constructions.

In addition to its visual feedback when using tools, Sketchpad provides feedback when using menu commands by enabling them and changing their wording according to the selected objects. Depending on what objects are selected, the Trace command might become **Trace Circle**, **Trace Perpendicular Lines**, **Trace Straight Objects** (for a segment and a ray), or even **Trace Objects** (for a point and a circle).

GeoGebra uses a combination of tools and its “Input” line, instead of menu commands. Using the Input line to enter a command can be confusing and error-prone. For instance, there’s no easy way to discover that typing “(2,3)” in the input line will plot a point at (2, 3), that typing “(a,b)” will plot the point determined by the lengths of segments *a* and *b*, or that typing “(Radius[*c*],Area[*c*])” will plot the point determined by the radius and area measurements of circle *c*. GeoGebra provides a list of over 160 available commands, arranged alphabetically, but with no easy way to find documentation or examples.



Command-line interfaces, such as that used in GeoGebra, have largely disappeared from consumer software, in favor of tools and menus, for three reasons: the command-line interface fails to inform the user of possible choices; a long list of commands is hard to remember; and the rigid syntax of commands results in errors. For instance, GeoGebra’s Angle command can be used to measure an angle or to construct a rotated image of an existing point, but if the user makes a mistake (for

instance, by typing “Angle[*a*]” when *a* is a segment), an error dialog box appears as shown in the illustration. This error message may be confusing to users.

Sketchpad’s organization of commands into menus makes them easier to find, the disabling of inappropriate commands makes it easy to determine what commands can be used in conjunction with the selected objects, and the Help menu provides easy access to full information about each command, including selection prerequisites.

Sketchpad has many additional features, not available in GeoGebra, that make a variety of constructions and actions easy to accomplish:

- Easy creation of parameters (determining their displayed precision and keyboard +/- adjustment by the number of decimal places typed), easy editing of values (directly in the sketch), and easy animation (via keyboard shortcut, Display menu, Motion Controller, or Animation action button).
- Easy measurement of the value of a point on any path, and easy plotting of a value on any path. (This coordinatization of every path adds an important element of algebra/geometry integration to Sketchpad.)
- Most commands that apply to a particular object are available through the object's Context menu (by right-clicking or control-clicking the object).
- It's easy to redefine objects using the **Split** and **Merge** commands, without needing to type error-prone definitions.
- It's easy to copy and paste any combination of selected objects for use within Sketchpad or for graphic export; for graphic export, Sketchpad automatically crops white space. GeoGebra does not support copying and pasting within the program, and exports only the entire uncropped contents of the geometry window.
- Sketchpad's interface for creating custom tools is simple: select the completed construction and choose **Create New Tool**. Sketchpad does the work of identifying the given objects and the constructed objects, and displays a script showing the construction steps. In GeoGebra the user must specify given and constructed objects separately, and no script is available.

Support

Sketchpad's Learning Center (available from the Help menu) contains a number of videos that show an introduction to Sketchpad, teachers describing how they use it at various grade levels, and suggestions for getting started using it in the classroom.

Sketchpad's Learning Center also contains a set of 12 self-guided, video-supported tutorials that cover the most important Sketchpad skills in a mathematical context. It also contains more than 70 Sketchpad tips, in both video and comic-strip format, that show how to use the most important Sketchpad tools, menu commands, and techniques. GeoGebra has very limited support materials.

Sketchpad's Reference Center contains approximately 350 topics, illustrated and cross-linked, along with a comprehensive index and full-text search tool. GeoGebra's help system contains few illustrations or examples, and its index is unlinked, including only page numbers, with no mechanism for finding those pages.

Sketchpad's Reference Center and most of its Learning Center are available from the user's own computer, without depending on internet connectivity. GeoGebra's help file (as well as the software itself) requires internet access.

Key Curriculum Press provides six-week online moderated courses for elementary, middle-school, high school algebra, and high school geometry teachers, face-to-face workshops of various lengths at a variety of levels, and provides subscriptions to Sketchpad LessonLink, a library of more than 500 Sketchpad activities (including extensive teacher notes) correlated to major textbooks and state standards at every level.

Available activities for GeoGebra are of uneven quality and are not well organized.

Summary

Dynamic mathematics software is a powerful tool for student learning and understanding. Whether it's used for whole-class presentations or hands-on student activities, the software must be easy to learn, adhere to mathematical standards, integrate geometry and algebra effectively, encourage experimentation, and enhance mathematical communication. Sketchpad excels in all these areas.

Sketchpad encourages beginning users by:

- Providing easily-accessible guided tutorials and Sketchpad Tips
- Displaying an uncluttered window and toolbox, so it's easy to begin with compass and straightedge
- Making tools work intuitively, whether by press-and-drag or by click-release-click-release
- Giving highly-visible onscreen feedback about the results of constructions
- Eliminating error messages, allowing students to evaluate their own work by deciding whether a construction works as expected or desired

Sketchpad follows conventional mathematical usage by:

- Using the Euclidean plane for fundamental geometric constructions
- Constructing only a single segment between two points
- Constructing angle bisectors as rays, and only in properly-defined angles
- Making a distinction between an object and its measure
- Using correct mathematical terminology

Sketchpad integrates geometry and algebra by:

- Giving the user control to define coordinate systems based on geometrical objects
- Putting geometric and algebraic objects in the same window
- Providing consistent ways to measure the various attributes of geometric objects
- Providing easy ways to incorporate measurements and parameters into geometric definitions
- Allowing any geometric path to be used as a number line

Sketchpad encourages student experimentation and conjecturing by:

- Enabling effective use of the “drag test,” by making every object draggable in a way that reveals its mathematical constraints
- Making all constructions, transformations, measurements, and other commands available from the menu bar
- Providing a small group of powerful, flexible tools

Sketchpad enhances mathematical communication by:

- Describing relationships between objects in complete sentences

- Enabling students to easily write about their constructions, conjectures, and conclusions while using Hot Text to link their writing to the mathematical objects
- Providing a comprehensive list of the objects in a sketch or tool, in complete sentences
- Making effective presentations easy, with large points, lines, and fonts and with action buttons
- Creating an attractive uncluttered workspace within which to create and display interesting mathematics

Sketchpad remains the gold standard of dynamic mathematics software. It provides a consistent and intuitive implementation of algebraic and geometric constructions, and puts the user in control of the connections between the algebraic and geometric realms. It enables powerful mathematics, with strong handling of iteration, loci, transformations, and digital images. It enables a more powerful level of mathematical expressivity, encouraging students to explore, write about, and present their mathematical ideas. And its careful attention to attractive appearance and consistent and intuitive interface design make it a satisfying and friendly program for teachers and students alike.

Scott Steketee has been involved in Sketchpad design and development since 1992. This document reflects both insights and biases associated with that involvement.