Measuring the Power of Learning.®

Research Memorandum
ETS RM–17-02

The *m-rater*™ Engine: Introduction to the Automated Scoring of Mathematics Items

James H. Fife

September 2017

ETS Research Memorandum Series

EIGNOR EXECUTIVE EDITOR

James Carlson
Principal Psychometrician

ASSOCIATE EDITORS

Beata Beigman Klebanov
Senior Research Scientist

Heather Buzick
Research Scientist

Brent Bridgeman
Distinguished Presidential Appointee

Keelan Evanini
Research Director

Marna Golub-Smith
Principal Psychometrician

Shelby Haberman
Distinguished Research Scientist, Edusoft

Anastassia Loukina
Research Scientist

John Mazzeo
Distinguished Presidential Appointee

Donald Powers
Managing Principal Research Scientist

Gautam Puhan
Principal Psychometrician

John Sabatini
Managing Principal Research Scientist

Elizabeth Stone
Research Scientist

Rebecca Zwick
Distinguished Presidential Appointee

PRODUCTION EDITORS

Kim Fryer
Manager, Editing Services

Ayleen Gontz
Senior Editor

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Memorandum series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes. Peer review notwithstanding, the positions expressed in the ETS Research Memorandum series and other published accounts of ETS research are those of the authors and not necessarily those of the Officers and Trustees of Educational Testing Service.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

The *m-rater*™ Engine: Introduction to the Automated Scoring of Mathematics Items

James H. Fife
Educational Testing Service, Princeton, New Jersey

September 2017

Corresponding author: James H. Fife, E-mail: jfife@ets.org

Suggested citation: Fife, J. H. (2017). *The* m-rater™ *engine: Introduction to the automated scoring of mathematics items* (Research Memorandum No. RM-17-02). Princeton, NJ: Educational Testing Service.

Find other ETS-published reports by searching the ETS ReSEARCHER database at http://search.ets.org/researcher/

To obtain a copy of an ETS research report, please visit http://www.ets.org/research/contact.html

Action Editor: Anastassia Loukina

Reviewers: Leslie K. Nabors Olah and Luis E. Saldivia

Copyright © 2017 by Educational Testing Service. All rights reserved.

CBAL, ETS, the ETS logo, GRE, and MEASURING THE POWER OF LEARNING. are registered trademarks of Educational Testing Service (ETS). C-RATER and M-RATER are trademarks of ETS.

All other trademarks are the property of their respective owners.

Abstract

This report provides an introduction to the *m-rater*™ engine, ETS's automated scoring engine for computer-delivered constructed-response items when the response is a number, an equation (or mathematical expression), or a graph. This introduction is intended to acquaint the reader with the types of items that m-rater can score, the requirements for authoring these items on-screen, the methods m-rater uses to score these items, and the features these items must possess to be reliably scored. M-rater can score 3 types of responses—numeric responses, equations, and graphs—and these 3 types of responses are considered separately. Each type of response has certain technical requirements that must be satisfied and certain considerations for scoring. These requirements are discussed in detail in the Numeric Response Items, Equation Items, and Graph Items sections. Also discussed are *generating example* items—items with many correct solutions for which the student must produce 1 correct solution. The final chapter explains how to convert scoring rubrics into concepts and scoring rules, a necessary step for building m-rater scoring models.

Key words: automated scoring, equation editor, graph editor, scoring rubric, scoring tolerance

Table of Contents

What Is *m-rater*™?

Constructed-response test items have a number of documented advantages over traditional multiple-choice items (Bennett, Sebrechts, & Yamamoto, 1991; Bridgeman, 1992; Drasgow & Mattern, 2006; Katz, Friedman, Bennett, & Berger, 1996; Payne & Squibb, 1990; Sebrechts, Bennett, & Rock, 1991). Constructed-response items eliminate the possibility of random guessing, they require recall of the correct response rather than recognition of the correct response from a list of alternatives, they replicate more closely the sorts of tasks test takers face in real-life situations, and they allow more detailed analysis of response errors. Additionally, multiple-choice items are susceptible to back-solving and provide cues to the test taker that an error has been made when the test taker obtains an answer that is not one of the options.

The significant advantage of multiple-choice items is that responses can be scored quickly and inexpensively. But advances in automated scoring technology have made possible the automated scoring of many computer-delivered constructed-response items. This advantage is most apparent with items whose responses are mathematical objects, such as equations, graphs, or simple numbers. This report provides an introduction to the *m-rater*™ engine, an automated scoring engine developed by Educational Testing Service (ETS) for computer-delivered constructed-response items when the response is a number, an equation (or mathematical expression), or a graph. This introduction is intended to acquaint item authors and test developers with the types of items that m-rater can score, the requirements for authoring these items on-screen, the methods m-rater uses to score these items, and the features these items must possess to be reliably scored. Others who can benefit from this introduction include research staff who intend to use m-rater in a research project, IT staff charged with integrating m-rater into other systems, mathematicians interested in the mathematical basis of some of the scoring algorithms, and anyone else with a need to be familiar with this important ETS tool.

When the response to an item is an equation or expression, m-rater is used in conjunction with an equation editor that allows the test taker to enter an equation or other expression in standard mathematical format, with exponents, radical signs, fractions, and so forth. Any third-party equation editor can be used, provided the editor is configurable at the item level and outputs the response in a standard format, such as MathML. When the response is a graph, m-rater is used with the ETS graph editor, which generates a graph as the test taker clicks points on a grid.

M-rater was initially developed at ETS in the mid-1990s (Bennett, Morley, & Quardt, 2000; Bennett, Morley, Quardt, & Rock, 2000; Bennett et al., 1999; Bennett, Steffen, Singley, Morley, & Jacquemin, 1997) and has been used successfully since then to score mathematics tasks in a variety of programs, including the *CBAL*® learning and assessment tool research project and one operational state assessment. Over the years, a number of improvements have been made to m-rater itself, to the graph editor, and to the procedures for writing scoring models (Fife, 2011, 2013). Today, m-rater is a state-of-the-art system that can be used to score responses to numeric, equation, and graph constructed-response items appearing in traditional computer-delivered assessments, K–12 assessments aligned with the Common Core State Standards or the Next Generation Science Standards, and other computer-based mathematics assessments.

Since 2012, m-rater has used an open-source computer algebra system to determine if the student's response to an equation item is mathematically equivalent to the intended response. Additionally, m-rater develops a parse tree for the equation that can be used to determine if the equation is in the correct form, if desired, or to score a response based on mathematical properties of the response rather than the actual value of the response itself. M-rater expects a response consisting of a single equation. M-rater cannot score items for which the response is a sequence of equations or a mixture of text and equations, although current research is investigating the possibility of using m-rater in conjunction with the *c-rater*™ automated scoring engine, ETS's machine learning–based engine for scoring short (one- or two-sentence) text responses for content, to score text responses with embedded equations.

A graph response is scored based on the points that the student plots to generate the graph and the specifications in the rubric. For example, if the student is asked to produce a line with a slope of 3 and the student plots the points (x_1, y_1) and (x_2, y_2), then the response is scored as correct if $(y_2 - y_1)/(x_2 - x_1) = 3$. Other formulas exist for more complicated rubric specifications; the appendix discusses the formulas for rubric specifications involving the graph of a smooth function.

Often an item that was written for paper delivery and human scoring will need to be revised for automated scoring. For example, consider the following typical word problem:

> The treasurer for the homecoming dance sold tickets in advance and tickets at the door. Advance tickets cost \$5 and tickets at the door cost \$6. The treasurer sold a total of 212

> tickets at a total cost of $1,178. Write a system of equations to represent this situation. Then use the system to find the number of advance tickets sold.

From a scoring perspective, the item asks the test taker to provide three pieces of information:

1. An equation that expresses the total number of tickets sold in terms of the number of advance tickets and the number of tickets sold at the door
2. An equation that expresses the total cost of the tickets sold in terms of the number of advance tickets and the number of tickets sold at the door
3. The number of advance tickets sold

To make the item suitable for automated scoring, the three parts must be presented separately so that the test taker can respond to each part separately and each part can be scored separately. For the first two parts, the test taker must be provided with an equation editor window to enter his or her responses; for the third part, the test taker must be provided with a text box that accepts only numeric entries. Additionally, for the first two parts, the item must specify the variables to be used. So here is how the item could be revised to be suitable for automated scoring:

> The treasurer for the homecoming dance sold tickets in advance and tickets at the door. Advance tickets cost $5 and tickets at the door cost $6. The treasurer sold a total of 212 tickets at a total cost of $1,178.
>
> 1. Write an equation that expresses the total number of tickets sold in terms of the number of advance tickets, *x*, and the number of tickets sold at the door, *y*.
> 2. Write an equation that expresses the total cost of the tickets sold in terms of the number of advance tickets, *x*, and the number of tickets sold at the door, *y*.
> 3. Use your two equations to find the number of advance tickets sold. Write the number of advance tickets sold in the following answer box.

M-rater can score a response based on responses to previous items or parts of items. Thus, if a student enters incorrect equations to Parts 1 and 2, m-rater can score the student's response to Part 3 based on the incorrect responses to Parts 1 and 2.

Following this introductory section are sections devoted to each of the three types of responses: numeric responses, equations and mathematical expressions, and graphs. Subsections deal with technical considerations, special considerations for that response type, and scoring that

type of response with a tolerance. For numeric responses and equations, there are also subsections dealing with a type of item known as *generating examples*. For graph items, there are subsections describing the ETS graph editor, the types of graph objects that the editor can capture, how the editor can be configured at the item level, and special considerations regarding the scoring of graphs. After these sections, there is a discussion of concepts and scoring rules as well as necessary steps toward writing scoring models. Finally, the report ends with a summary and conclusion section.

Numeric Response Items

Numeric response items can usually be scored by conventional scoring engines (e.g., ETS's SKM or eSKM scoring engine). But if the scoring rubrics are complicated—for example, if the response must be scored based on previous responses, if the response must be in the form of a fraction, if the response must be scored with a tolerance, or if there are complicated round-off rules—then it may be necessary to score the response with m-rater. As the example in the previous section illustrates, when the response is numeric, the test taker should be provided with a text box in which to enter the response. M-rater can determine if the response is equivalent to the intended response. For example, if the intended response is 1.5, m-rater will recognize that 1.50, 3/2, 6/4, and 1 1/2 are equivalent to 1.5.

Test takers can enter a mixed fraction by entering a space between the whole-number part and the fractional part. Test takers should be cautioned, however, not to enter additional spaces, as m-rater may misinterpret the response. For example, the response "1 1 1/12" will be interpreted by m-rater as $11\frac{1}{12}$; if the student had intended $1\frac{11}{12}$, the student's score would not be based on the student's intended response. To avoid these difficulties, some programs provide separate entry boxes for the numerator and the denominator of a fraction and for the integer part of a mixed fraction. M-rater can score responses captured in this way, also.

Technical Requirements

The answer box for a numeric response should only accept numeric characters and the symbols necessary to create numbers, such as a hyphen (in lieu of a minus sign), a forward slash (to denote a fraction), and a period (to denote a decimal point). In particular, the test taker must not be able to enter alpha characters (letters) in the box; these keys must be disabled. If the alpha keys are not disabled, a student may be tempted to enter dimensions or some other text in the

box; m-rater will not recognize these characters and will return a score of 0. As a general rule, if a numeric response has units (such as feet, minutes, or miles per hour), it is best to write the units outside the text box. If identifying the units is part of what the item is testing, then provide two text boxes: one for the numeric response and one for the units. The units response can be scored using a direct character match, although the scoring rubrics should specify acceptable abbreviations and misspellings. Alternatively, if it is sufficient for the test taker to recognize rather than recall the correct unit, then consider giving the units box a drop-down menu so that the test taker can select the correct response. Either way, the numeric response can be scored conditionally on the response in the units box. For example, if the correct answer to a question is "2 feet," then a response of "24 inches" could be recognized and scored as correct.

Scoring Considerations

If desired, the scoring rubrics can require a response in a certain form. For example, if the rubrics require a decimal response, then "3/2" would not be accepted for 1.5. Or if the rubrics require a fraction response with no common factors in the numerator and denominator, then neither "1.5" nor "6/4" would be accepted for 3/2. The scoring rubrics can also specify a range of correct responses, for example, any decimal number x such that $5 \le x < 7$.

Scoring With a Tolerance

A response can be scored with a tolerance, for example, a correct response could be 1.3 ± 0.1, so that any response between 1.2 and 1.4 would be scored as correct. Note that the tolerance must be specified in the scoring rubrics.

This issue of tolerance is particularly important if the response is a fraction that produces a nonterminating decimal expansion. For example, if the correct response is $13\frac{2}{3}$, then the item writers must specify in the scoring rubrics what sort of responses are to be accepted as correct. If the scoring rubrics do not specify a tolerance, then only responses equivalent to $13\frac{2}{3}$ will be scored as correct (e.g., $\frac{41}{3}$), and a response like 13.667 will be scored as incorrect. On the other hand, if decimal approximations are acceptable, then the rubrics must specify the required degree of precision. For example, is 13.7 acceptable, or must the answer be correct to, say, three decimal places? If so, is 13.666 acceptable? Or is any response within the range $13\frac{2}{3} \pm 0.001$ acceptable?

Item authors can require that a correct response be rounded to a specified number of decimal places, or they can require that a correct response be rounded or truncated to a specified number of decimal places. Or, as stated earlier, they can set a tolerance.

Generating Examples

The term *generating examples* applies to problems that do not have a unique solution but rather present constraints and ask the test taker to provide an example that meets those constraints (Bennett et al., 1999). Here is an example from Bennett et al. (1999):

> Joe is driving cross-country. He travels 3,000 miles in 60 hours, switching cars somewhere along the way. The two cars have different average speeds, each of which does not exceed 70 miles per hour. Give an example of a speed and time for each leg of the trip. (p. 234)

There are, of course, several possible correct responses and several strategies for finding them. One strategy, described in Bennett et al., is first to calculate the overall average speed as 50 miles per hour (3,000 ÷ 60), suppose that the times spent in each car are equal (30 hours each), and then suppose that the average speed of the first car is 45 miles per hour. One can then calculate that the average speed of the second car must be 55 miles per hour. The point is that the test taker must provide examples of four quantities—the speed of the first car, the time spent in the first car, the speed of the second car, and the time spent in the second car. There are many possible answers for these quantities, but if we label these quantities r_1, t_1, r_2, and t_2, then they must satisfy the equations and inequalities

$$\begin{aligned} r_1 t_1 + r_2 t_2 &= 3,000, \\ t_1 + t_2 &= 60, \\ 0 < r_1 &\le 70, \\ 0 < r_2 &\le 70. \end{aligned}$$

Whatever numbers the test taker enters in the response fields (there must be one response field for each quantity), m-rater can determine if that collection of numbers constitutes a correct response.

Equation Items

M-rater can score items for which the response is a single equation or other mathematical expression. The expression can involve one or more variables. The "key" can be a specific equation or expression, or it can be one or more properties that a correct response is required to have. For example, consider the following items:

1. Write an equation of the line containing the points (1, 5) and (2, 7).
2. Write an equation of the line containing the points (1, 5) and (2, 7). Put your answer in the form $ax + by = c$.
3. Write an equation of a line with a slope of 3.

For Item 1, a correct response will be any linear equation mathematically equivalent to $y = 2x + 3$. M-rater can determine if a response is a linear equation and if it is equivalent to $y = 2x + 3$. For Item 2, m-rater can also determine if the response is in the correct form; thus, for this item, $y = 2x + 3$ would be scored as incorrect, but $2x - y = -3$ would be scored as correct. For Item 3, a response will be scored as correct if it is a linear equation and its slope is 3.

Technical Requirements

As stated earlier, the item must provide an equation editor in which test takers can enter their responses. The equation editor will ensure that test takers can enter expressions involving fractions, exponents, radicals, and so on, with proper mathematical notation. (See Figure 1 for an example of an item with an equation editor.) The editor must output the response to m-rater encoded as MathML or in some other standard format that m-rater can read. Most equation editors can be configured for the item so that only those symbols that are relevant to that item are present. In addition, it must be possible to configure the equation editor to restrict the characters that test takers can enter from the keyboard. In particular, alpha characters must be restricted to relevant variables. Otherwise, test takers may attempt to enter text in the response. The editor will interpret the extra characters as variables and try to compose an equation that includes these variables. For example, if the correct response is 6*x*, meaning "6*x* feet," and a test taker attempts to enter "6*x* ft," the editor will interpret the response as the three-variable expression 6*xft*. M-rater, of course, will score such an expression as incorrect.

Note that this does not mean one should never allow students to enter incorrect variables; if the item is designed, in part, to see if students know which letters to use as variables, then test takers should be allowed to enter targeted incorrect variables. For example, the following item was administered as part of an ETS research project (Fife, Graf, Ohls, & Marquez, 2008):

Write a linear equation in the form $ax + b = 0$ with a solution of $x = -3$.

When administered on paper, a common incorrect response was $-3a + b = 0$. Because part of what was being tested was to see if the test taker knew that a correct response would be an equation in *x*, it would be reasonable to allow test takers to enter *a* and *b* in their responses.

Usually, the text of the item should specify the variables to be used. Test takers cannot choose their own variables; m-rater needs to know in advance what variables to expect. As stated earlier, if m-rater encounters variables it does not expect, it will score the response as incorrect.

Special Considerations

Be sure to specify if the response is to be an equation or an expression or to state in the scoring rubrics that either is acceptable. For example, consider the following released *GRE*® general test item:

If $y = 2x - 1$, what is the value of *x* in terms of *y*?

(A) $\frac{y}{2} - 1$ (B) $\frac{y}{2} - \frac{1}{2}$ (C) $\frac{y}{2} + \frac{1}{2}$

(D) $\frac{y}{2} + 1$ (E) $y + \frac{1}{2}$

It is clear that the correct response is an expression and not an equation of the form $x = f(y)$ because all the options are expressions; none is an equation. But if this item were converted into a constructed-response item by simply removing the options, then it would not be clear if the correct response should be the expression

$$\frac{y+1}{2} \tag{1}$$

or the equation

$$x = \frac{y+1}{2} \tag{2}$$

Either the constructed-response version of the item should be revised to make clear that the answer should be an expression ("Let $y = 2x + 1$. Give an expression for the value of y"), or else the scoring rubrics should be written to accept either (1) or (2). The point here is that if the rubrics require the expression and a student enters the equation, the student's response will be scored by m-rater as incorrect. (Actually, even if you are not testing whether the test taker knows the difference between an equation and an expression, and you are willing to accept either, you probably should still specify one in the item so test takers who do appreciate the difference will not worry about which is expected.)

Scoring With a Tolerance

Sometimes an equation must be scored with a tolerance. This could be the case, for example, if the test taker is asked to determine an equation from a graph when the graph does not pass exactly through the grid points. For example, see the item in Figure 1. The student is shown the graph of a line and is asked to write an equation of the line. The actual equation of the line is $h = 2.4a + 31$. But a student most likely would not be expected to write the exact equation, so a tolerance must be established for the coefficients in the response. When this item was scored in CBAL, a response was given full credit if the slope was between 2 and 3 and the y-intercept was between 30 and 32, and part credit was given if the slope was between 1.5 and 3.1 and the y-intercept was between 30 and 33. As with numeric response items, the tolerance must be specified in the scoring rubrics.

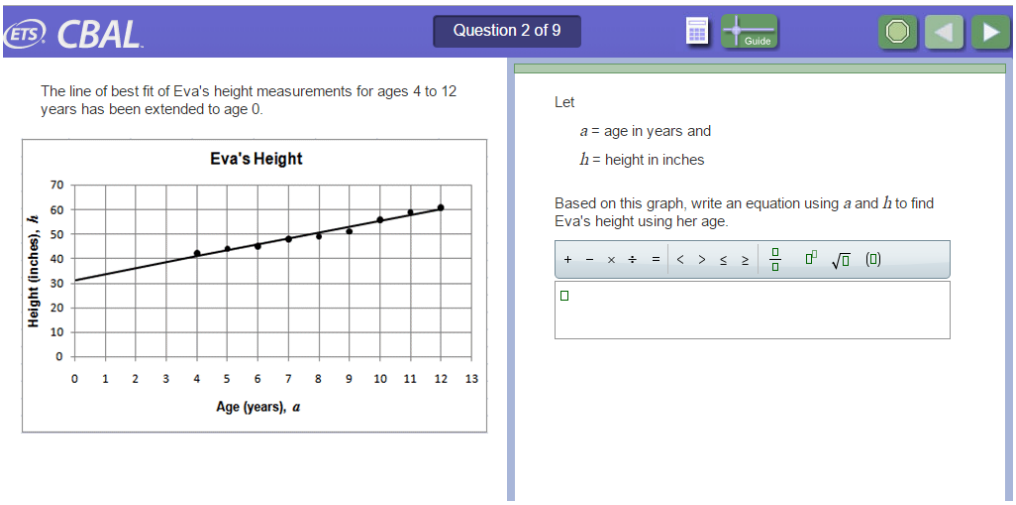


Figure 1. A CBAL equation item scored with tolerances.

Generating Examples

Generating example items can have equation responses as well as numeric responses. Consider, for example, this item from Bennett, Morley, and Quardt (2000):

> Two lines in the *xy*-coordinate plane are perpendicular. If the equation of the first line is $x+5y=17$, what is a possible equation, in the form $y=f(x)$, of the second line?

There are infinitely many correct responses, of course, and m-rater can score the item because it can determine if a response is a linear equation in the requested form with a slope of 5.

Graph Items

As stated previously, when the response to an item consists of one or more graphs, the ETS graph editor must be used to capture the graphs for the responses to be scored by m-rater. The editor can capture six types of graph objects: points, lines, line segments, graphs of smooth functions, graphs of piecewise-linear functions, and connected line segments. A response can consist of any combination of these graph objects. The test taker enters the graph by clicking points on a grid; as the test taker clicks the points, the editor generates the desired graph object. If appropriate, the test taker can be required to first click a button to indicate the type of graph object to be generated. The response is scored based on the points that are plotted and the type of graph object generated (or that the test taker selected to be generated).

Examples of Responses That the Graph Editor Can Capture

Following are examples of each of the types of graph responses.

Points. Figure 2 shows an item for which the test taker must plot a collection of points. To enter a response, the test taker clicks in the coordinate plane where the points are to be plotted. Figure 3 shows the completed response. In this item, the student was asked to plot five points. The editor can be configured to accept only five points (or any other particular number of points), or it can be configured to accept an unlimited number of points. Note that the graph editor has a snap-to-grid feature that has been enabled for this item, so that for this item, only grid points can be plotted. This feature can be enabled or disabled for each item; the details are discussed later.

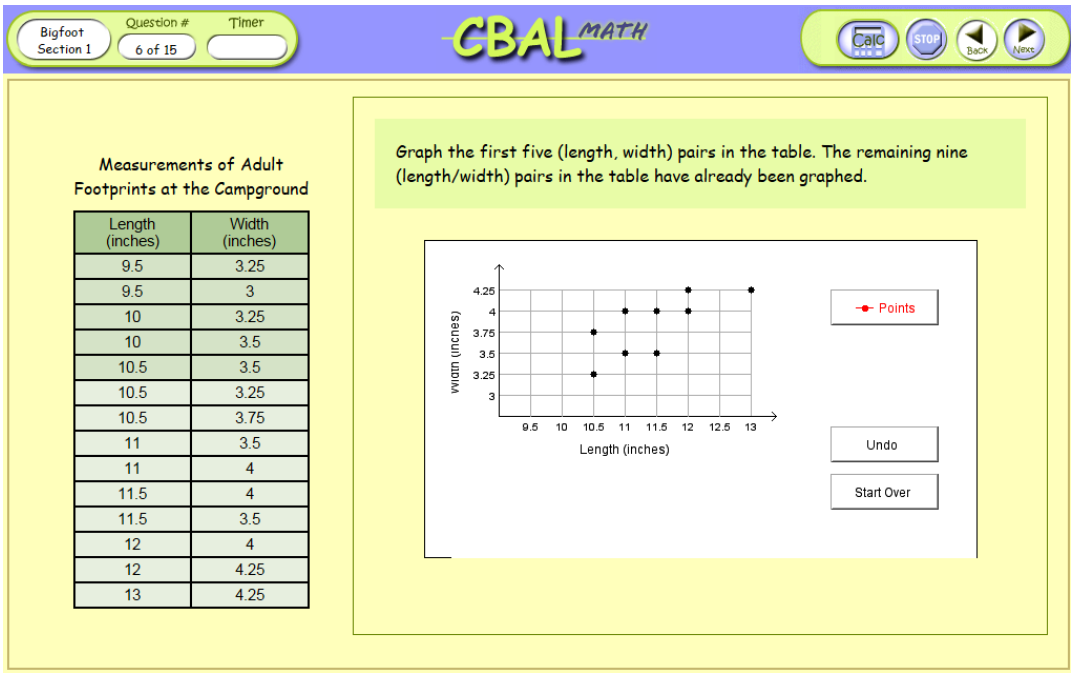


Figure 2. A CBAL item for which the test taker must plot a collection of points.

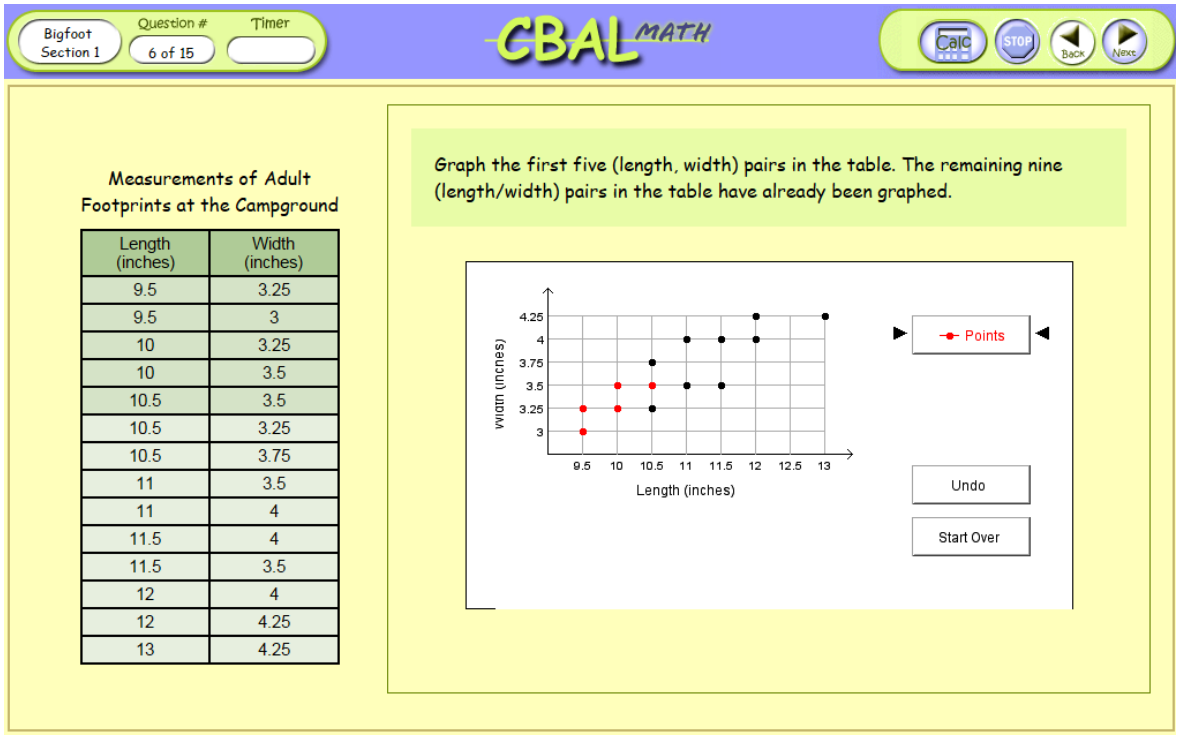


Figure 3. The completed response for the CBAL item in Figure 2.

Lines. Figure 4 shows an item in which the test taker was asked to plot three points and then to plot a line. To plot the line, the test taker must click on two points on the line. The test taker was asked to click on two of the three points already plotted, but in fact any two points that are on the line $y = 3x$ will do. Figure 5 shows a response in which the points (0, 0) and (3, 9) were plotted. The line containing the two points is generated automatically when the second point is plotted. After the line has been generated, the test taker can move the line by clicking and dragging one or both of the two points to other locations.

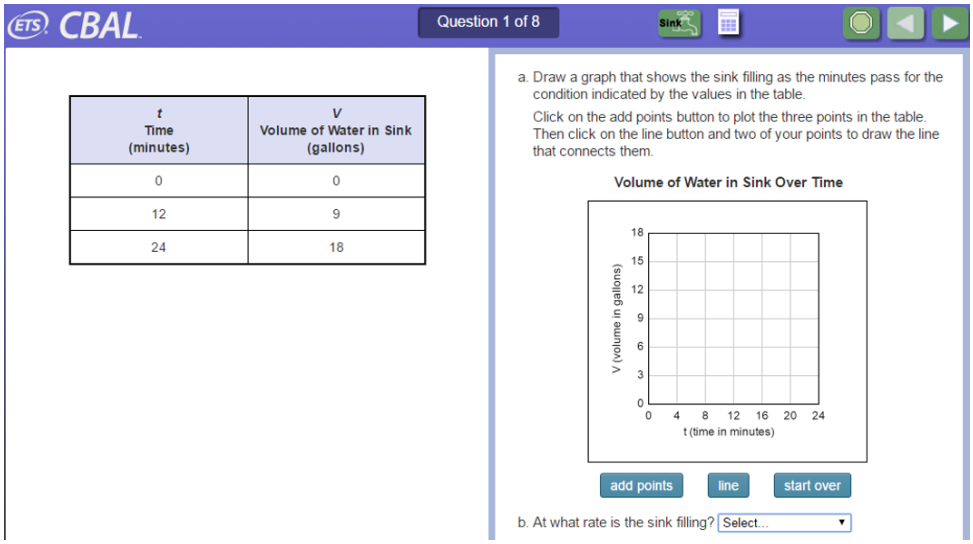


Figure 4. A CBAL item for which the test taker must plot a line.

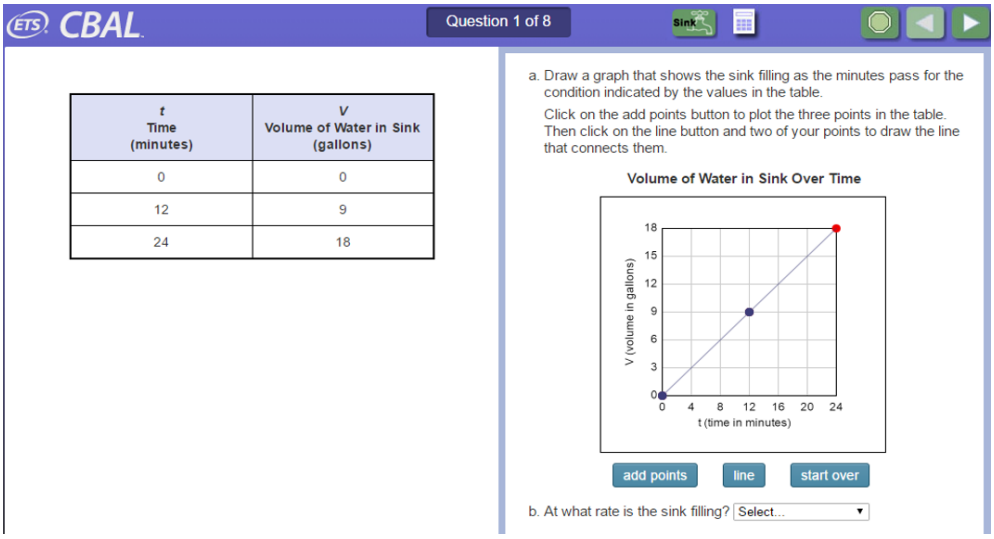


Figure 5. The completed response for the CBAL item in Figure 4.

Line segments. Figure 6 shows an item for which the test taker has been asked to plot a line segment; the correct response is shown. To enter a response, the test taker must click the two endpoints of the segment, in this case, the points (0, 30) and (15, 0). Note that part of the point being tested is to see if the test taker knows that the line segment must extend to (15, 0)—the end of the moving sidewalk—and not terminate at (13, 4), which is the final data point given. If the graph editor were configured to plot a line instead of a line segment, then the entire line from (0, 30) to (15, 0) would be generated, even if the test taker plotted the points (0, 30) and (13, 4), or even (0, 30) and (4, 22). In this case, a correct response would not provide evidence that the test taker understood the point being tested. Care must be taken by item authors to ensure that responses to graphing items (and, more generally, to all computer-delivered constructed-response mathematics items) are able to provide evidence that the test taker understands the aspect of the construct being tested.

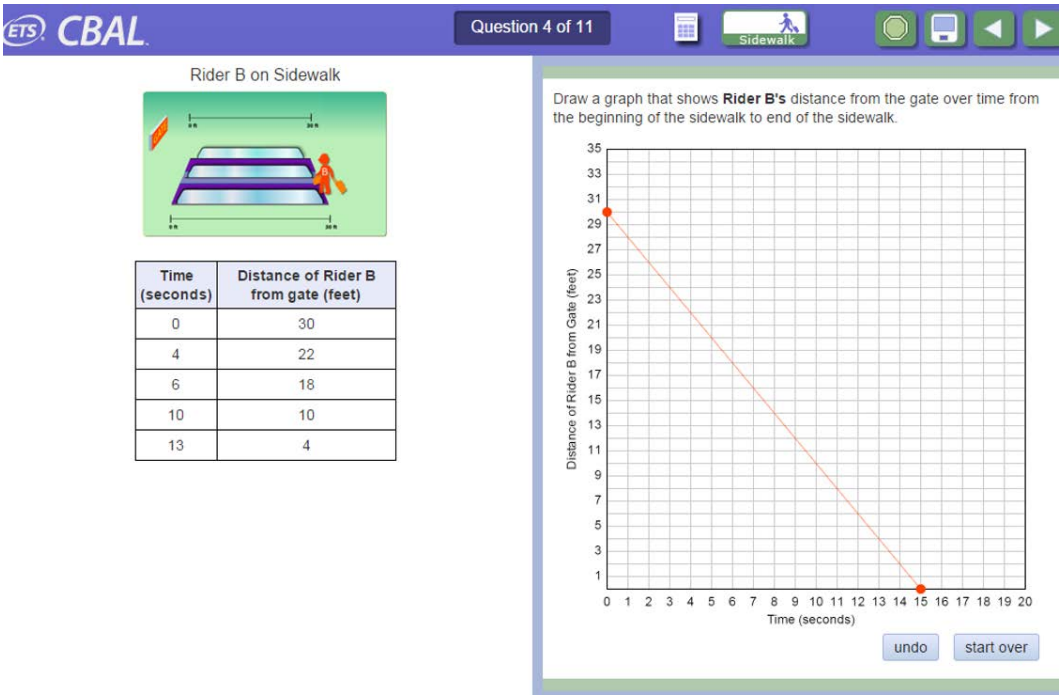


Figure 6. A CBAL item for which the test taker must plot a line segment.

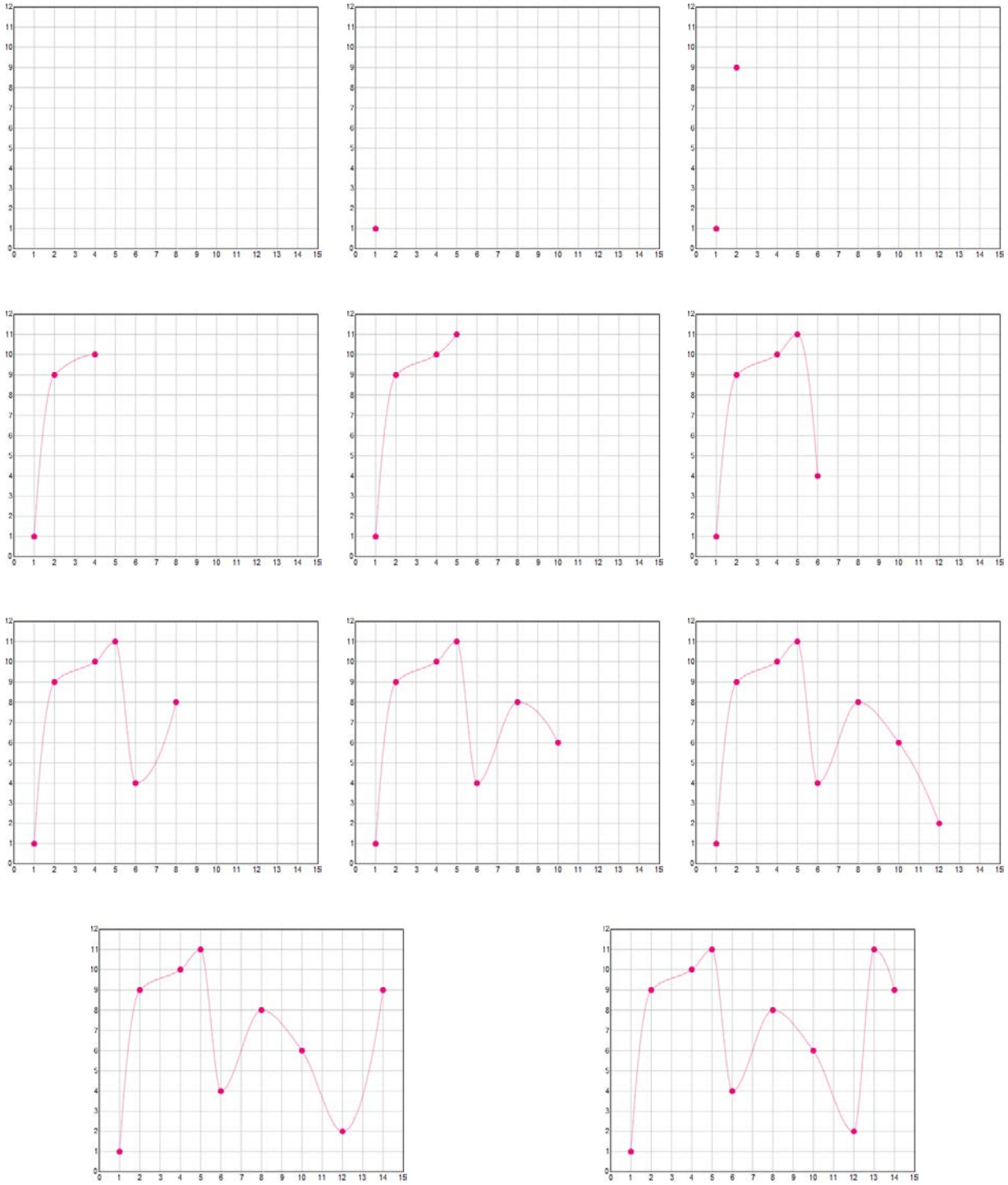


Figure 7. Graph of a smooth function plotted in the graph editor.

Graphs of smooth functions. The graph editor can capture the graph of a smooth function. The test taker plots the graph by clicking points in the coordinate plane; as the test taker

clicks the points, the graph is automatically generated by the editor. The points can be clicked in any order; see the sequence of graphs in Figure 7. Note that generation of the graph begins once three points have been plotted and that as additional points are plotted, the graph adapts to fit the additional points. Note also that the graph preserves local extrema in the sense that if the *y*-coordinate of a plotted point is greater than the *y*-coordinates of its two neighbors, the graph will have a local maximum at the point, and similarly with local minima. As with lines and line segments, once all the points have been plotted, the test taker can edit the graph by clicking and dragging points to other locations, subject to the constraint that two points cannot have the same *x*-coordinate. (Note that the graph as a whole cannot be selected and moved, although each point can be moved separately.) As with points, the editor can be configured to accept only a fixed number of points, or it can be configured to accept an unlimited number of points.

Graphs of piecewise-linear functions. The graph editor can also capture the graph of a piecewise-linear function. The test taker clicks points in the coordinate plane; as the test taker clicks the points, the editor automatically connects the points with a piecewise-linear graph. The points can be plotted in any order (Figure 8). The last two points plotted are (13, 5) and (3, 7), in that order. But when the last point is plotted, the editor does not generate a line segment from the point (13, 5) to the point (3, 7). Instead, the editor "inserts" the point (3, 7) between the points (2, 3) and (6, 6) and replaces the line segment between (2, 3) and (6, 6) with two line segments.

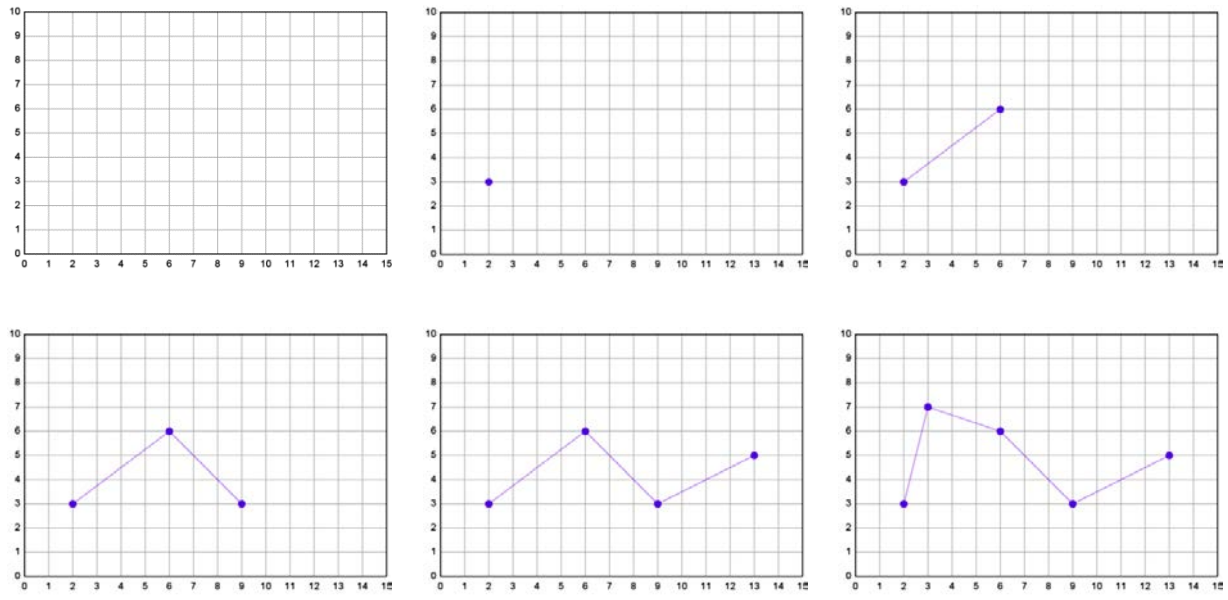


Figure 8. Plotting points to generate the graph of a piecewise-linear function.

Connected line segments. A graph of connected line segments is a graph consisting of a sequence of line segments for which the endpoint of each segment is the beginning point of the next. A graph of connected line segments is similar to the graph of a piecewise-linear function in Figure 8, except that the points are connected by line segments in the order in which they are plotted, and therefore, the resulting graph may not be the graph of a function (see Figure 9). When the point (−2, −2) is plotted, the editor does not insert that point between the points (−4, 3) and (2, 2); instead, it generates a line segment between the previous point, (4, −3), and the point (−2, −2).

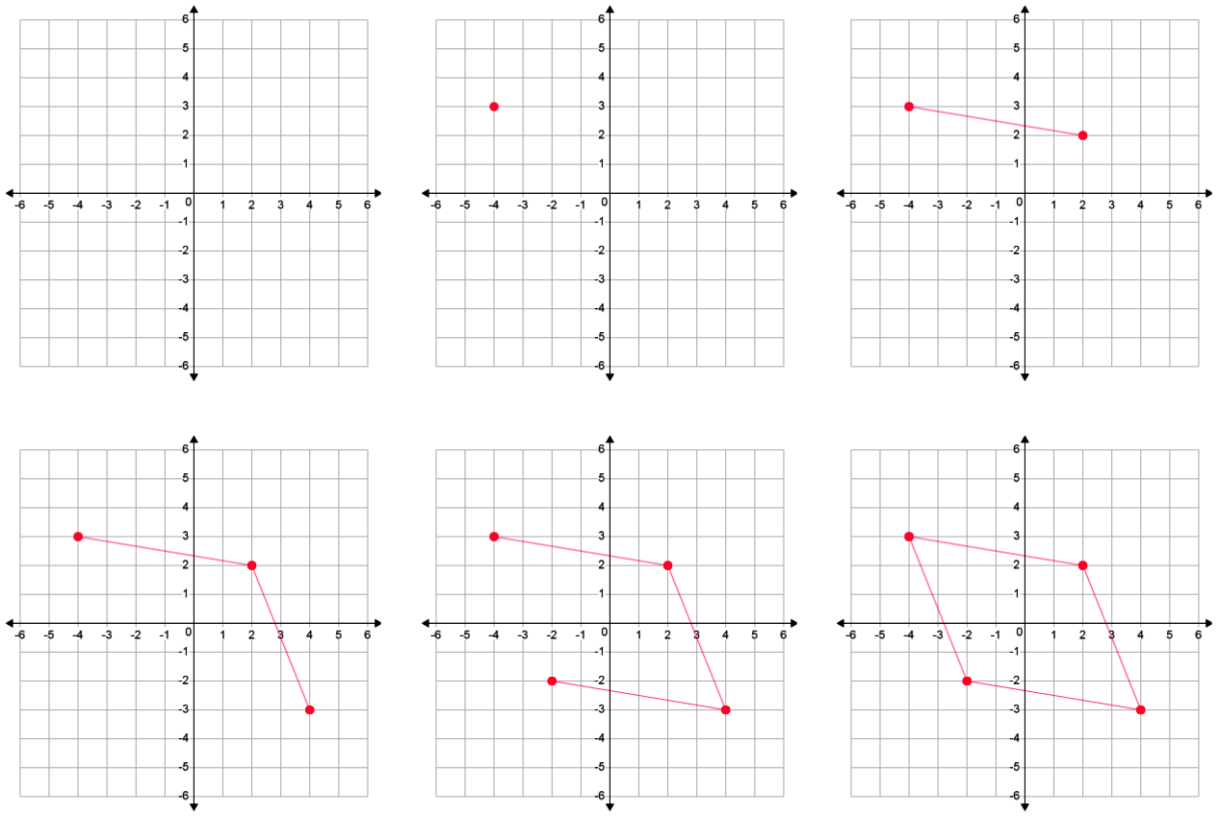


Figure 9. Generating a graph of connected line segments.

Configuring the Graph Editor

As can be seen from the preceding examples, the graph editor can be configured differently for each item in a variety of ways, including the choice of the viewing window, the axis labels, whether snap-to-grid is enabled, the number of graph objects to be plotted, the types of graph objects to be plotted, and the colors and other surface features of the graph objects.

Viewing window. The *viewing window* is that portion of the coordinate plane that is displayed in the graph editor. The viewing window determines (or is determined by) the range of the *x*-axis and the *y*-axis. In Figure 2, for example, *x* ranges from 9 to 13, with a grid point every 0.5, while *y* ranges from 2.75 to 4.25, with a grid point every 0.25; in Figure 6, *x* ranges from 0 to 20 and *y* ranges from 0 to 35, both with unit grid points. The item author has the option of requiring the test taker to select the viewing window if the ability to determine an appropriate viewing window is part of what is being assessed. Otherwise, the item author and content specialists must specify the viewing window parameters $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ as well as the grid widths for the *x*- and *y*-axes. Note that a graph's physical size and aspect ratio (the ratio of the width of the graph to its height) can also be configured for each item.

Snap-to-grid. The graph editor has a snap-to-grid feature that can be enabled or disabled at the decision of the item author and the content reviewers. When snap-to-grid is enabled, the test taker can only plot grid points; when the test taker clicks somewhere inside a grid, the point "snaps" to the nearest grid point. For example, in Figure 1, one of the points the test taker must plot is the point (9.5, 3.25); if the test taker clicks anywhere close to (9.5, 3.25), the point snaps to (9.5, 3.25). As a result, it is easy for the test taker to provide precise responses.

However, care must be taken when snap-to-grid is enabled to ensure that the grid width is selected in such a way that only grid points need to be plotted. As a result, snap-to-grid with items that use real data is often problematic; these items are frequently impossible to configure in a way that only grid points need to be plotted. For example, consider the item in Figure 10. If the graph editor in this item is to be configured so that only grid points need to be plotted, then there must be a grid point every unit along the *y*-axis, and hence, there must be 1,500 horizontal grid lines. Clearly this would make the graph unreadable. So for this item, snap-to-grid must be disabled. As a result, however, test takers cannot be expected to plot points with precision.

To plot the point (12, 168), for example, the test taker must estimate where 168 is along the *y*-axis and may actually plot the point (12, 165) or (12, 171) instead of (12, 168). As a result, when snap-to-grid is disabled, responses must be scored with a tolerance. This point is discussed later.

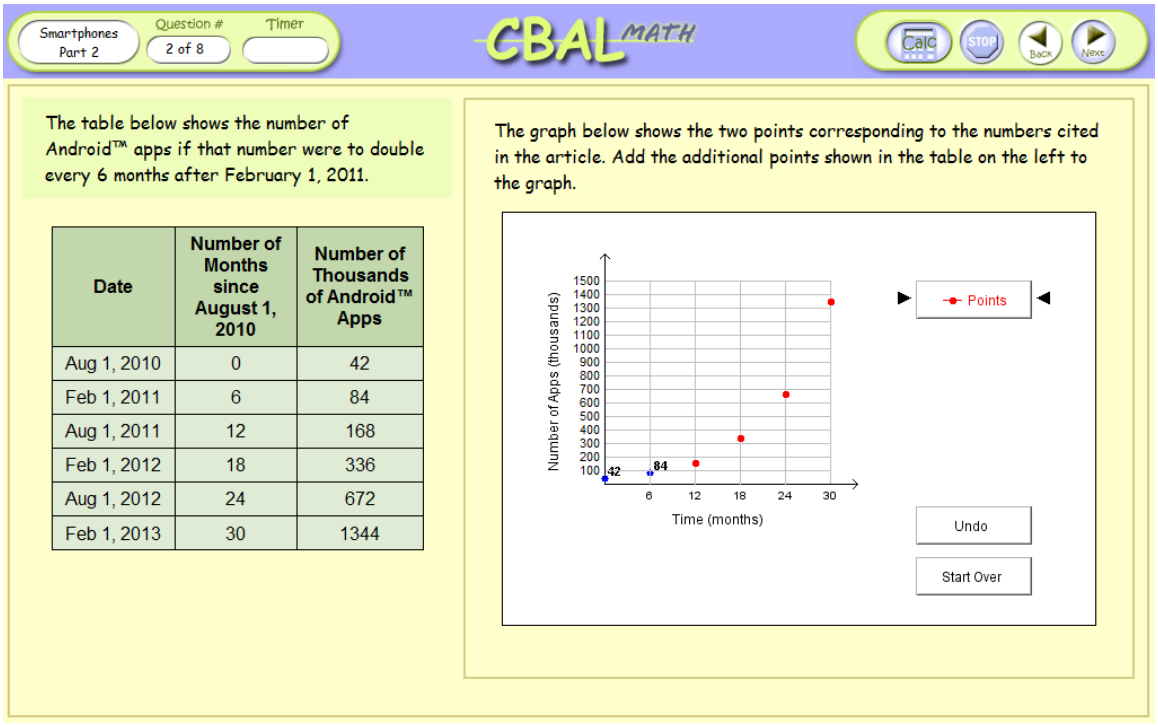


Figure 10. A CBAL item for which snap-to-grid is deactivated.

Test taker–determined viewing window and axis labels. As stated, item authors and content specialists have the option of requiring the test taker to select the viewing window. Test takers can also be required to enter labels for the axes. If this option is chosen for a particular item, then in the space where the graphing window would normally be displayed, there will be fields for the test taker to enter minimum and maximum values for *x* and *y* and the labels for the *x*- and *y*-axes. (There are checks to prevent a test taker from entering a maximum value that is less than or equal to the minimum value.) The test taker then clicks a button and the graphing window appears with the test taker's choice of viewing window and axis labels (see Figure 11).

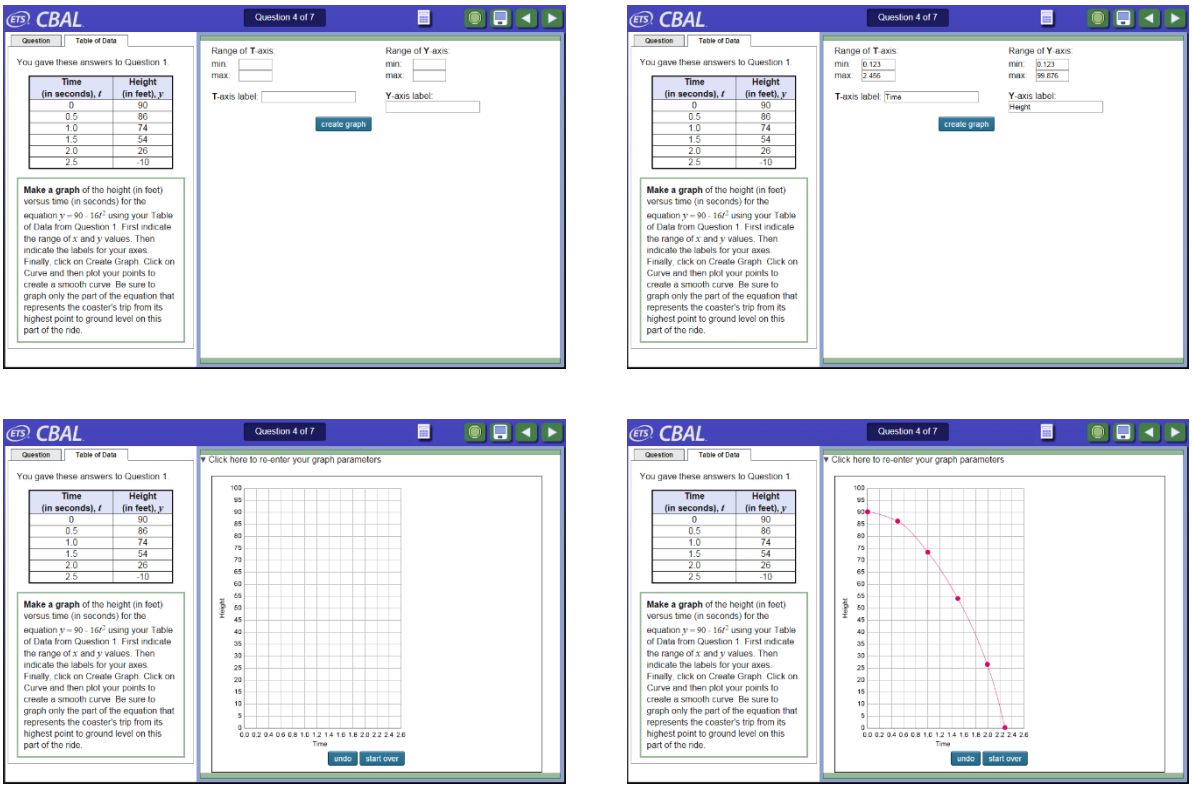


Figure 11. A CBAL item in which the test taker sets the viewing window and axis labels.

As explained in Fife (2013), test takers do not determine the grid width; instead, the item author establishes the approximate number of grid lines for the item, and the graph editor then calculates the grid width based on the number of grid lines set by the item author and the maximum and minimum selected by the test taker. Similarly, if a test taker selects maximum and minimum values that are "weird," the editor will overrule those choices and choose "nice" values instead. This function is illustrated in Figure 11: The test taker has chosen 0.123 as the minimum value for both axes, but this preference is overridden, and 0 is selected instead as the minimum.

Because test takers cannot be expected to select parameters that produce grid points that correspond to correct responses, it is necessary, when test takers set the viewing window, that snap-to-grid be disabled.

In addition, the graph editor can be configured so that the test taker must provide the *x*-axis and *y*-axis labels. Test takers can be provided with a drop-down list from which to select a label, or they can be provided with a text box in which to enter the label as free text. The test

takers' choice of labels can be scored by c-rater or other techniques for scoring text, such as character match.

Number of points to be plotted. When the object to be graphed is a set of points, a smooth curve, a piecewise-linear curve, or a set of connected line segments, a limit can be placed on the number of points that the test taker is allowed to plot; alternatively, this number can be left unlimited. Thus, for example, if the item asks the test taker to plot five points, the editor can be configured to accept only five points, preventing the test taker from making the mistake of plotting too many points, or it can be configured to accept as many points as the test taker tries to plot, allowing the test taker to make the mistake of plotting too many points.

Variety of graph objects to be plotted. In most of the preceding examples, the items involved plotting just one graph object. But an item can involve plotting any number of graph objects. For example, an item could ask the test taker first to plot the line $y = 2x + 1$ and then to plot a line perpendicular to the first line. Or an item could ask a test taker to plot a curve, a point on the curve, and a line tangent to the curve at that point. When an item asks that several graph objects be plotted, the graph editor is configured with buttons, appropriately labeled, for each graph object. The test taker clicks each button before plotting the graph object associated with that button. Figure 4 provides an example of this. In this item, the test taker was asked to plot three points and then to plot a line.

Test taker–determined type of graph object. At the discretion of the item author and content specialists, the graph editor can be configured to plot automatically the type of graph object appropriate to the item (when the test taker clicks points), or the editor can be configured first to require the test taker to indicate the type of object to be plotted. For example, if an item asks the test taker to plot the graph of a function with certain properties, the editor could be configured to have three buttons: one for the graph of a smooth function, one for the graph of a piecewise-linear function, and one for the graph of a straight line. The test taker would need to know which type of graph object was appropriate for the item.

Graph object labels. When the editor displays buttons for various graph objects, the labels on the buttons are configurable per item. Item authors and content specialists can choose descriptive labels, such as "Bart's walk" or "Andy's travels," for these buttons. Additionally, the editor can be configured so that the labels appear on the grid when the objects are graphed. Also,

the colors of lines and curves and the colors and shapes of points are configurable for each item. When multiple graph objects are present, each can be given a different color.

Scoring Rubrics

As stated earlier, a graph response is scored based on the points that the test taker has plotted, but the scoring rubrics must specify how this is to be done. The scoring rubrics can be specified in a variety of ways, depending on the type of object the test taker was asked to graph and the nature of the question the item asks about that object.

For example, in the item in Figure 2, the test taker was asked to plot five points. The scoring rubrics must specify those points. But the rubrics must also specify how the response should be scored if the test taker plotted the five correct points but also plotted additional, incorrect points. Additionally, if partial credit is to be given for a partial response, say, four correct points and one incorrect point, then the rubrics must specify that also. The following is one possible set of rubrics:

2 points: The five correct points and no incorrect points are plotted.

1 point: Five points are plotted, at least three of which are correct.

0 points: Any other response.

For another example, in the item in Figure 4, the test taker was asked to plot the line $y = 3x$. The test taker could plot several possible pairs of points to create the correct line: the points (0, 0) and (1, 3), the points (2, 6) and (4, 12), the points (3, 9) and (6, 18), and so on. The scoring rubrics do not need to specify all pairs of points that generate the correct line, however. The scoring rubrics only need to specify the equation of the correct line; m-rater will calculate the equation of the line plotted by the test taker and score the response accordingly.

Alternatively, a response could be scored based on certain properties of the response instead of the precise response itself. For example, in the item in Figure 12, adapted from a research project at ETS, the test taker is asked to plot the graph of a smooth function; to be correct, the function must pass through the points (0, 0) and (8, 6), must be increasing, and must be concave down. An example of a correct response is shown in Figure 13. M-rater can determine, from the points plotted, if the curve has the required properties. (See the appendix for the formulas m-rater uses to determine the properties of the curve.)

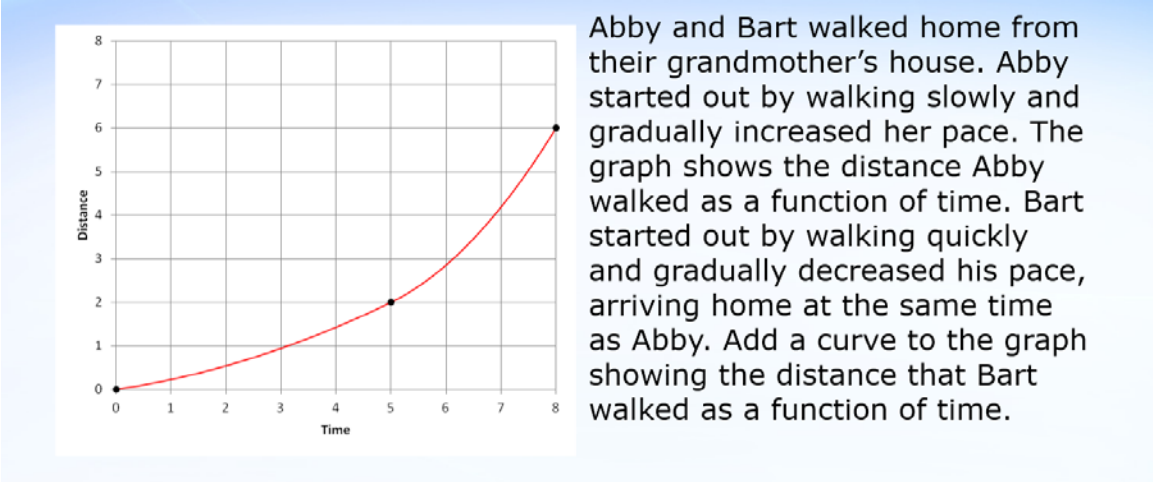


Figure 12. An item whose response is the graph of a smooth function.

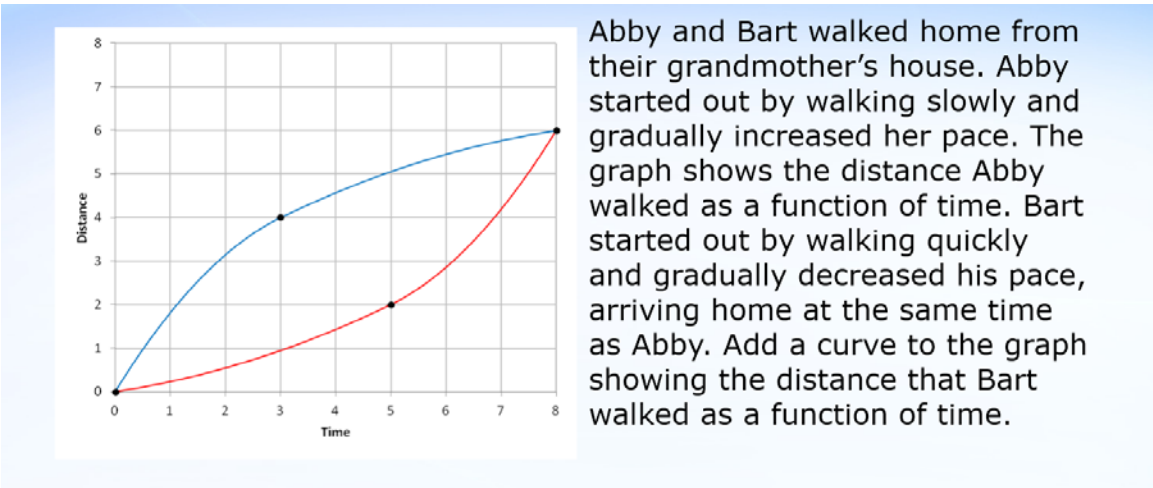


Figure 13. A correct response to the item in Figure 12.

Scoring rubrics can be quite complicated. For example, see the item in Figure 14, adapted from a released operational item used in a state assessment. Note that the item has many possible correct responses; the graph in the figure shows one such response. For example, we are not told how fast Andy bikes from his house to his school or from his school to his friend Jack's house, but we do know that he bikes from the school to Jack's house more slowly than he bikes from his house to his school. So although there are many possibilities for the slope of the first line segment and many possibilities for the slope of the second, we do know that the slope of both line segments must be positive and that the slope of the second line segment must be less than the slope of the first.

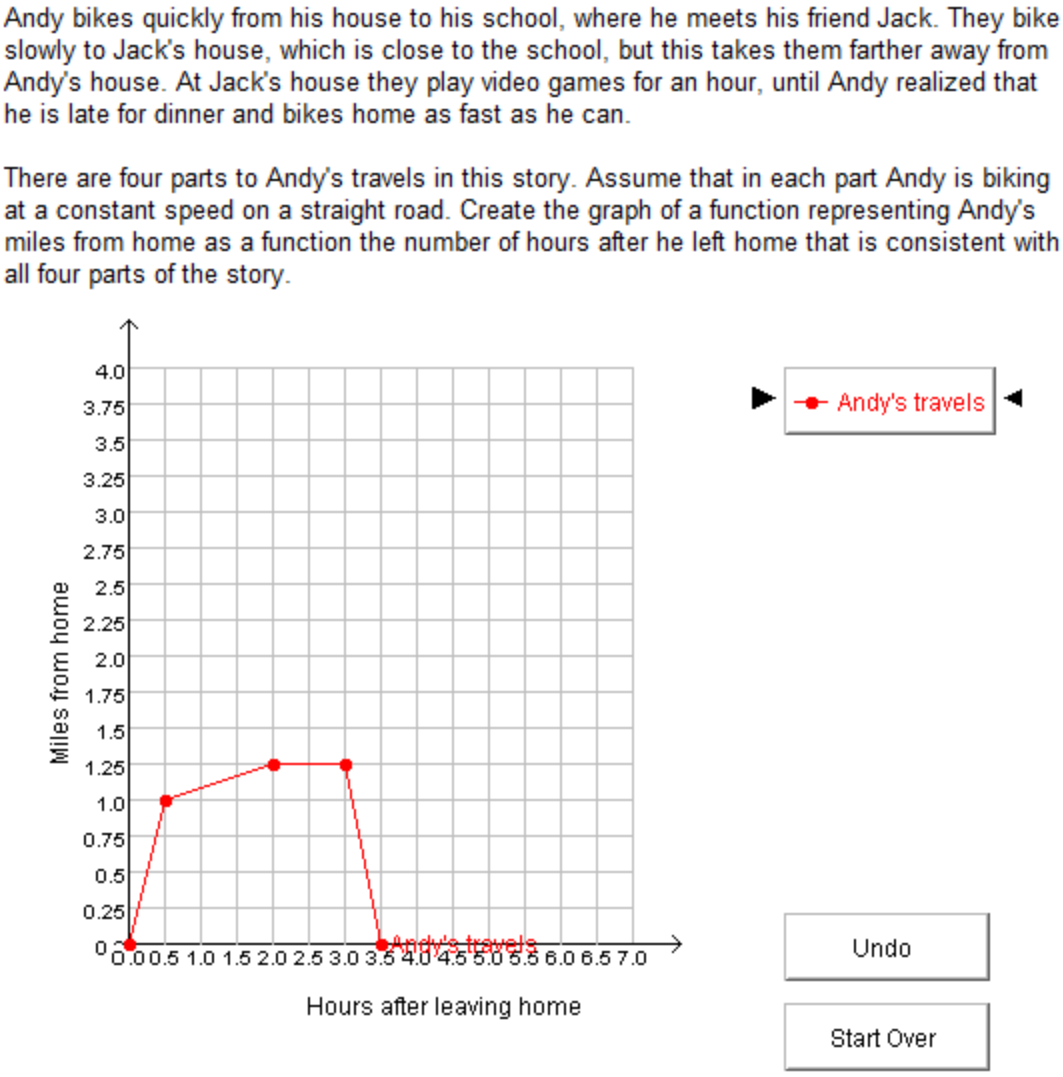


Figure 14. A CBAL item in which the test taker must plot the graph of a piecewise-linear function.

Similarly, the slope of the third line segment must be zero and its length must represent 1 hour, and the slope of the fourth line segment must be negative, with absolute value at least as great as the slope of the first line segment, because Andy biked home as fast as he could and therefore at least as fast as he biked to school. Thus, to receive full credit, a response must have the following properties:

- The response contains four line segments.
- The first segment begins at the point (0, 0) and has positive slope.

- The second segment has a positive slope that is less than the slope of the first segment.
- The third segment has a slope of 0 and a length of 1.
- The fourth segment has a negative slope whose absolute value is greater than or equal to the slope of the first segment.

M-rater can determine if a response has these properties and score it accordingly. M-rater can also give partial credit to a response that has some but not all of these properties; the scoring rubrics must specify how partial credit is to be assigned.

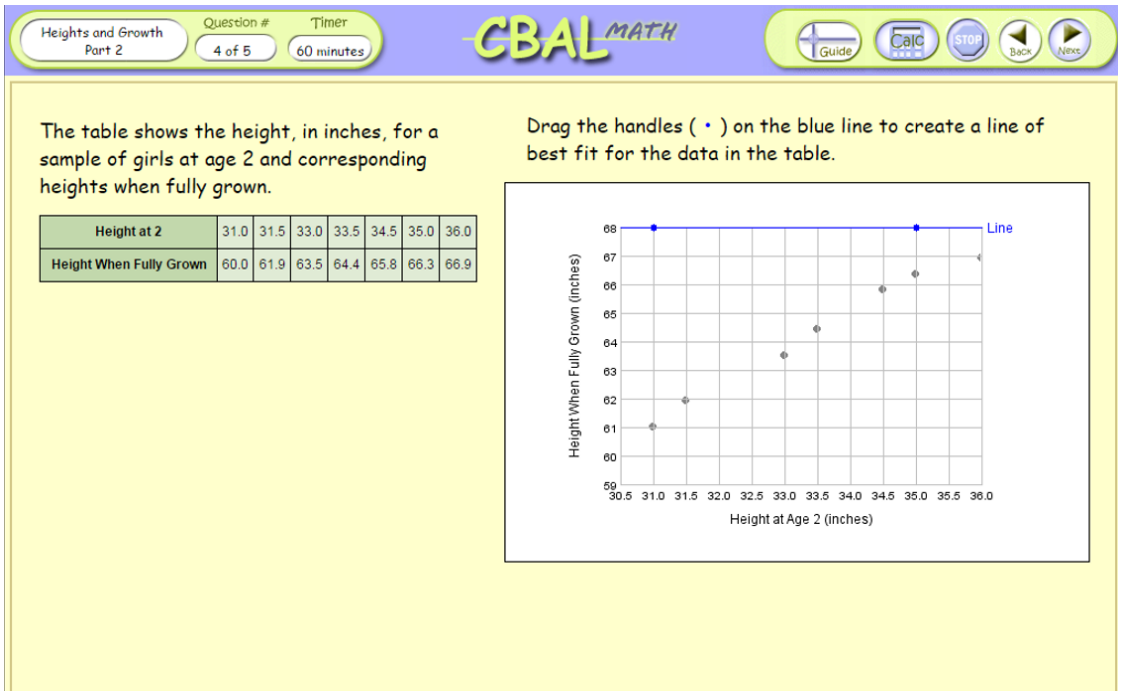
Scoring With a Tolerance

As with numeric responses and equation responses, graph responses must sometimes be scored with a tolerance. Following are some common examples.

Line of best fit items. Consider the item shown in Figure 15. The test taker must plot a line that provides a good fit for the data points. There are many such lines, of course, so the scoring rubrics must set a tolerance. As shown in Fife (2013), it is not sufficient to calculate the actual line of best fit and then to set independent tolerances on the slope and the *y*-intercept of the response. One can set a tolerance on the *y*-intercept that is a function of the slope, but it is easier to set tolerances on the endpoints of the line as drawn in the viewing window. For example, here is a reasonable scoring rubric for the item in Figure 15:

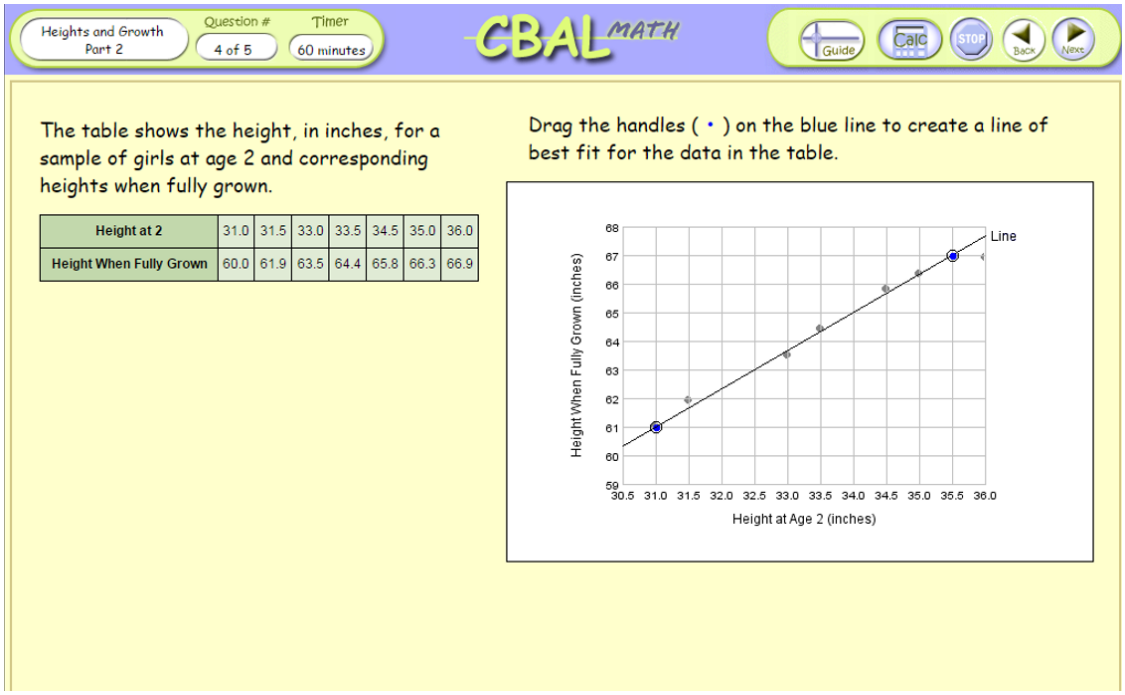
> Let y_L and y_R denote the *y*-coordinates of the left- and right-hand endpoints, respectively. A response is scored as correct if $60 \le y_L \le 61$ and $67 \le y_R \le 68$ OR if $61 \le y_L \le 62$ and $66 \le y_R \le 67$.

The response in Figure 16 satisfies the first of these two conditions and hence would be scored as correct.



Height at 2	31.0	31.5	33.0	33.5	34.5	35.0	36.0
Height When Fully Grown	60.0	61.9	63.5	64.4	65.8	66.3	66.9

Figure 15. A CBAL line of best fit item.



Height at 2	31.0	31.5	33.0	33.5	34.5	35.0	36.0
Height When Fully Grown	60.0	61.9	63.5	64.4	65.8	66.3	66.9

Figure 16. A CBAL response to the line of best fit item.

Tolerance in graphs with snap-to-grid disabled. Recall that if snap-to-grid is disabled in the graph editor, then the responses must be scored with a tolerance. Consider, for example, the item in Figure 10. As was pointed out earlier, to plot the point (12, 168), the test taker must estimate where 168 is along the *y*-axis; the test taker easily could be slightly off and plot instead the point (12, 165) or (12, 171). As a result, a tolerance must be established within which responses will be considered as correct. Any point (*x*, *y*) with $|x-12| \le \delta$ and $|y-168| \le \varepsilon$ for suitable δ and ε will be considered as a correct response for the point (12, 168). The appropriate values of δ and ε must be set by content specialists and, if appropriate, developmental psychologists as part of the scoring rubrics. It is recommended that tolerances be established from actual student response data; based on the response data, histograms can be prepared that show the distribution of student responses. For example, the histogram in Figure 17 shows the distribution of the actual *y*-values that were plotted by a sample of students attempting to plot the point (12, 168) in response to this item. From the histogram, it is clear that the bulk of the responses are between 125 and 200, so a reasonable value of ε might be $168 - 125 = 43$.

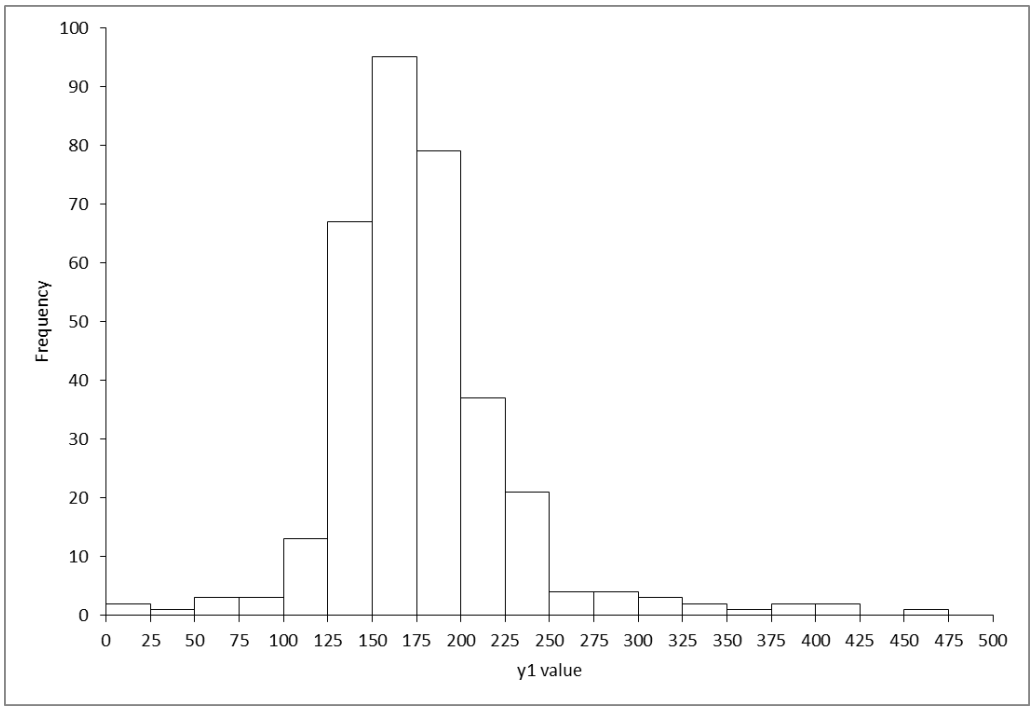


Figure 17. A histogram showing the distribution of plotted *y*-values for the point (12, 168).

Credit for incorrect points plotted. Consider again the item in Figure 2. In this item, the student is asked to plot five points. Recall that it is possible to configure the graph editor so that the student can plot only five points; if this has been done, then a student who plots the five correct points will not have plotted any incorrect points, and, conversely, a student who plots one or more incorrect points will not have been able to plot all five correct points. But if the editor has been configured so that the student can plot more than five points, then the student could plot all five correct points but also plot some incorrect points. Indeed, if the editor has been configured so that there is no maximum on the number of points that the student can plot, then the student could just plot all available grid points. The student would, by default, have plotted all of the correct points but would, of course, have plotted all possible incorrect points as well. In this case, the scoring rubrics must specify how incorrectly plotted points are to be handled. For example, a student could receive full credit if all five correct points are plotted and no incorrect points; a student could receive partial credit if exactly five points are plotted, at least three of which are correct.

The situation is more complicated when snap-to-grid has been disabled. Consider again the item in Figure 10. A test taker cannot click the same point on the graph twice, but because snap-to-grid has been disabled in this item, a test taker can click many points "close" to each correct point, and because responses are being scored with a tolerance, all of those points would be scored as correct points. So it is not enough for this item to specify that a student gets full credit if four correct points are plotted and no incorrect points are plotted, because a student could click the "same" point four times. So the rubrics must state that each of the four points has been plotted (at least) once and no incorrect points have been plotted.

Concepts and Scoring Rules

Before a scoring model can be built for an item, the scoring rubrics must be translated into *concepts* and *scoring rules*. A concept is a feature of the response whose presence justifies awarding the response full or partial credit or, sometimes, justifies withholding credit from the response. For example, one concept could be the correct equation or graph; another could be a response that is to be awarded partial credit. A scoring rule is a rule of the form

any n of Concepts x, …, z is worth m points.

For example, a scoring rule might say "any 2 of Concepts 1, 2, and 3 is worth 1 point."

Example 1: An Equation Scored With Partial Credit

Write the equation of the line through the points (1, 5) and (3, 9).

Rubrics:

2 points: Any equation equivalent to $y = 2x + 3$.

1 point: Any incorrect linear equation with a slope of 2.

0 points: Any other response.

For these rubrics, there could be two concepts and two scoring rules:

Concept 1: The equation $y = 2x + 3$.

Concept 2: Any linear equation with a slope of 2.

Scoring Rule 1: Any 1 of Concept 1 is worth 2 points.

Scoring Rule 2: Any 1 of Concept 2 is worth 1 point.

Note that the order in which the concepts are listed is irrelevant, but the order in which the scoring rules are listed is important. A response is given the score specified by the first rule in the list that the response satisfies. In our example, the correct response of $y = 2x + 3$ satisfies both rules, but it receives 2 points because the 2-point rule is first in the list. If the rules were listed in the reverse order, so that the 1-point rule were first, then the response $y = 2x + 3$ would receive only 1 point.

Example 2: Another Equation Scored With Partial Credit

Write the equation of the line through the points (1, 5) and (3, 9).

Rubrics:

2 points: Any equation equivalent to $y = 2x + 3$.

1 points: Any incorrect linear equation with a slope of 2.

1 point: Any incorrect linear equation with a y-intercept of 3.

0 points: Any other response.

These rubrics can be translated into two concepts and two scoring rules, as follows:

Concept 1: A linear equation with slope 2.

Concept 2: A linear equation with y-intercept 3.

Scoring Rule 1: Any 2 of Concepts 1 and 2 is worth 2 points.

Scoring Rule 2: Any 1 of Concepts 1 and 2 is worth 1 point.

Example 3: Graph of a Line Scored Without Partial Credit

Draw the graph of $y = 2x + 3$.

Rubrics:

1 point: The correct graph.

This item will have one concept and one scoring rule:

Concept 1: The graph $y = 2x + 3$.

Scoring Rule 1: Any 1 of Concept 1 is worth 1 point.

Example 4: A Set of Points Scored With Partial Credit

Consider the item shown in Figure 2, with the following scoring rubrics.

Rubrics:

2 points: The five correct points and no incorrect points are plotted.

1 point: Five points are plotted, at least three of which are correct.

0 points: Any other response.

To translate these rubrics into concepts and scoring rules, begin by letting S be the set of correct points to be plotted. Then, as our first attempt at writing concepts, we have these two concepts:

Concept 1: The number of points plotted from the set S equals 5.

Concept 2: The number of points plotted from the set S is greater than or equal to 3.

Then a 2-point response will satisfy Concept 1 and a 1-point response will satisfy Concept 2. But both of these concepts leave open the possibility that additional, incorrect points have been plotted. Thus we need a third concept:

Concept 3: The number of points plotted equals 5.

We then have the following scoring rules:

Scoring Rule 1: Any 2 of Concepts 1 and 3 is worth 2 points.

Scoring Rule 2: Any 2 of Concepts 2 and 3 is worth 1 point.

Example 5: A Set of Points Scored With Tolerance and Partial Credit—Negative Concepts

Consider the item shown in Figure 10, with the following rubrics:

Rubrics:

2 points: Each of the four points is plotted within its tolerance, with no additional points plotted.

1 point: Three of the four points are plotted within their tolerance, with at most one additional point plotted.

The scoring rubrics for this item are similar to those of the previous item, but the presence of a tolerance in the scoring rubric forces the concepts to take a different form. For example, to satisfy the 2-point rubric, it is not sufficient for a concept to say that four points are plotted correctly. As explained earlier, this is because each of the four points could be a correct plot of the same target point; that is, each of the four points could be within the tolerance for the same target point. Thus a concept that says "four points plotted correctly" would be satisfied by a response that consisted of four correct attempts to plot the point (12, 168) and no attempts to plot any of the other three points.

So the concepts have to specify that each of the four points is plotted within its tolerance. It is easiest to do this with four separate concepts:

Concept 1: Point 1 is plotted within its tolerance.

Concept 2: Point 2 is plotted within its tolerance.

Concept 3: Point 3 is plotted within its tolerance.

Concept 4: Point 4 is plotted within its tolerance.

Now we need a concept that checks that the test taker did not plot more than four points. We could try the following:

Concept 5: No more than four points are plotted.

Then the 2-point scoring rubric would be satisfied by the presence of all five concepts:

Scoring Rule 1: Any 5 of Concepts 1–5 is worth 2 points.

But what about the 1-point rubric? It would require the following scoring rule:

Scoring Rule 2: Any 3 of Concepts 1–4 and any 1 of Concept 5 is worth 1 point.

But scoring rules like this are not possible; this scoring rule does not fit the syntax specified earlier. The way around this problem is to convert Concept 5 into a *negative concept* that targets a property that a response should not have and for which a response can lose points:

Concept 5: More than four points are plotted.

Scoring Rule 1: Any 1 of Concept 5 is worth 0 points.

Scoring Rule 2: Any 4 of Concepts 1–4 are worth 2 points.

Scoring Rule 3: Any 3 of Concepts 1–4 are worth 1 point.

Note that, for a response to receive 2 points, it must satisfy Scoring Rule 2—it must satisfy Concepts 1–4, that is, Points 1, 2, 3, and 4 are plotted—but additionally, it must fail to satisfy Scoring Rule 1, which means that at most four points are plotted. Taken together, this means that Points 1, 2, 3, and 4 are plotted and no additional points are plotted. A similar analysis holds for responses receiving 1 point.

Summary and Conclusion

M-rater automatically scores responses to computer-delivered constructed-response items when the response is a number, an equation, or a graph. M-rater can score a response based on the equivalence of the response to the correct response, or it can score a response based on properties of the response. M-rater can score responses based on responses to previous items.

For m-rater to score responses reliably, certain technical requirements must be satisfied. Numeric responses can be captured in a text box, equation responses must be captured in an equation editor, and graph responses are captured in ETS's graph editor. In each case, the answer box or editor must be properly configured to capture the test taker's intended response. These requirements have been discussed in the Technical Requirements subsections for numeric responses and equations and in the Configuring the Graph Editor subsection.

Care must be taken in writing scoring rubrics for m-rater-scored items, as m-rater will score the items exactly as the rubrics specify. If responses are to be scored with a tolerance, the rubrics must specify the tolerance, although the tolerance may be determined by looking at a sample of test taker response data. (Actually, looking at student responses before finalizing rubrics is generally good practice.) For graphs, a response is scored based on the points that the test taker has plotted, but the scoring rubrics must specify how this is to be done. After the scoring rubrics have been finalized, they must be converted into concepts and scoring rules; scoring models are built from these concepts and rules.

Finally, m-rater can score generating example items—problems that do not have a unique solution but instead ask the test taker to provide a solution that meets certain constraints. There may be many possible correct answers; m-rater can determine the correctness of the test taker's response.

This report has provided an introduction to m-rater and its technical requirements, with information about how to write items that take advantage of m-rater's capabilities while ensuring that m-rater will score responses reliably. With m-rater, item authors and test developers who want to include constructed-response mathematics items on assessments can do so without sacrificing the speed and efficiency of automated scoring.

References

Bennett, R. E., Morley, M., & Quardt, D. (2000). Three response types for broadening the conception of mathematical problem solving in computerized tests. *Applied Psychological Measurement, 24*, 294–309.

Bennett, R. E., Morley, M., Quardt, D., & Rock, D. A. (2000). Graphical modeling: A new response type for measuring the qualitative component of mathematical reasonng. *Applied Measurement in Education, 13*, 303–322.

Bennett, R. E., Morley, M., Quardt, D., Rock, D., Singley, M. K., Katz, I. R., & Nhouyvanisvong, A. (1999). Psychometric and cognitive functioning of an under-determined computer-based response type for quantitative reasoning. *Journal of Educational Measurement, 36*, 233–252.

Bennett, R. E., Sebrechts, M. M., & Yamamoto, K. (1991). *Fitting new measurement models to GRE General Test constructed-response item data.* Princeton, NJ: Educational Testing Service.

Bennett, R. E., Steffen, M., Singley, M. K., Morley, M., & Jacquemin, D. (1997). Evaluating an automatically scorable, open-ended response type for measuring mathematical reasoning in computer-adaptive tests. *Journal of Educational Measurement, 34*, 162–176.

Bridgeman, B. (1992). A comparison of quantitative questions in open-ended and multiple-choice formats. *Journal of Educational Measurement, 29*, 253–271.

Drasgow, F., & Mattern, K. (2006). New tests and new items: Opportunities and issues. In D. Bartram & R. Hambleton (Eds.), *Computer-based testing and the Internet: Issues and advances* (pp. 59–75). West Sussex, UK: John Wiley.

Fife, J. H. (2011). *Automated scoring of CBAL mathematics tasks with m-rater* (Research Memorandum No. RM-11-12). Princeton, NJ: Educational Testing Service.

Fife, J. H. (2013). *Automated scoring of mathematics tasks in the Common Core era: Enhancements to m-rater in support of CBAL mathematics and the Common Core assessments* (Research Report No. RR-13-26). Princeton, NJ: Educational Testing Service. https://doi.org/10.1002/j.2333-8504.2013.tb02333.x

Fife, J. H., Graf, E. A., Ohls, S., & Marquez, E. (2008). *Identifying common misconceptions: An analysis of the Mathematics Intervention Module (MIM) data* (Research Memorandum No. RM-08-16). Princeton, NJ: Educational Testing Service.

Katz, I. R., Friedman, D. E., Bennett, R. E., & Berger, A. E. (1996). *Differences in strategies used to solve stem-equivalent constructed-response and multiple-choice SAT-Mathematics items.* New York, NY: College Entrance Examination Board.

Payne, S. J., & Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of error. *Cognitive Science, 14*, 445–481.

Sebrechts, M. M., Bennett, R. E., & Rock, D. A. (1991). *Machine-scorable complex constructed-response quantitative items: Agreement between expert system and human rater scores.* Princeton, NJ: Educational Testing Service.

Appendix. Scoring Graphs of Smooth Functions

As indicated in the introductory section, a student enters a graph response by plotting points in the graph editor; the editor then connects the points with the appropriate graph. The response is scored as correct if the graph that is drawn has the desired properties. Whether it has these properties will depend on the points that the test taker plots. For each scorable property, there is a formula that expresses that property as a function of the coordinates of the points that have been plotted. In some cases, that formula will be fairly clear. For example, if the points (x_1, y_1) and (x_2, y_2) are plotted to generate a line, then the slope *m* and *y*-intercept *b* of the line are given by the formulas $m = (y_2 - y_1)/(x_2 - x_1)$ and $b = y_1 - mx_1$. For graphs of smooth functions, however, the formulas are not always so transparent and depend on the formulas that are used to generate the graph of the function.

The test taker draws the graph of a smooth function *f* by plotting at least three points (x_0, y_0), (x_1, y_1), …, (x_n, y_n) in the editor. The editor then draws a smooth curve through the points; on each interval $[x_{i-1}, x_i]$, the editor draws the graph of the cubic polynomial

$$f_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3,$$

where the coefficients a_i, b_i, c_i, and d_i are defined as follows:

$$\left.\begin{aligned} h_i &= x_i - x_{i-1} \\ k_i &= y_i - y_{i-1} \\ s_i &= h_i / k_i \end{aligned}\right\} \text{ for } i = 1, \ldots, n,$$

$$m_i = \begin{cases} \dfrac{2s_i s_{i+1}}{s_i + s_{i+1}} & \text{if } s_i s_{i+1} > 0 \text{ and } i \neq 0, n \\ 0 & \text{if } s_i s_{i+1} \leq 0 \text{ and } i \neq 0, n, \\ 2s_1 - m_1 & \text{if } i = 0 \\ 2s_n - s_{n-1} & \text{if } i = n \end{cases}$$

$$\left.\begin{aligned} a_i &= y_{i-1} \\ b_i &= m_{i-1} \\ c_i &= (3s_i - 2m_{i-1} - m_i)/h_i \\ d_i &= (m_{i-1} + m_i - 2s_i)/h_i^2 \end{aligned}\right\} \text{ for } i = 1, \ldots, n.$$

If *f* is defined by $f(x) = f_i(x)$ when $x_{i-1} \le x \le x_i$, the function *f* is called a Hermite cubic spline. Each piece of the spline is a cubic polynomial, and the spline has the properties that $f(x_i) = y_i$ and $f'(x_i) = m_i$ for all $i = 0,\ldots,n$ (Fife, 2013). For this particular spline, if $y_i > y_{i-1}$ and $y_i > y_{i+1}$ for some $i = 1,\ldots,n-1$, the spline has a local maximum at the point x_i (Fife, 2013). A similar statement holds for local minima. In fact, this spline is monotonic on each interval (Fritsch & Carlson, 1980).

Table A1 lists the features of the graph of *f* that we want to score; the table also indicates, in the second column, how those features are to be scored. Some of these scoring rules are self-evident; following the table is an explanation of the ones that are not.

Table A1. How to Score Features of the Graph of a Smooth Function

Feature	How to score
The parabola $y = ax^2 + bx + c$ is plotted.	*n* is even. $y_i = ax_i^2 + bx_i + c$ for all *i*. $h_i = h_{n-i+1}$ for $i = 1,\ldots,n/2$. $x_{n/2} = -b/(2a)$.
A parabola is plotted.	*n* is even. There are real numbers *a*, *b*, and *c* such that $y_i = ax_i^2 + bx_i + c$ for all *i*, $h_i = h_{n-i+1}$ for $i = 1,\ldots,n/2$, $x_{n/2} = -b/(2a)$.
The function *f* is graphed.	$y_i = f(x_i)$ for all *i*.
The point $P = (\hat{x}, \hat{y})$ is plotted.	$x_i = \hat{x}$ and $y_i = \hat{y}$ for some *i*.
Function is increasing on interval $[x_{i-1}, x_i]$.	$y_{i-1} < y_i$.
Function is decreasing on interval $[x_{i-1}, x_i]$.	$y_{i-1} > y_i$.
Function is concave up on interval $[x_{i-1}, x_i]$.	$m_{i-1} < 3s_i - (m_{i-1} + m_i) < m_i$.
Function is concave down on interval $[x_{i-1}, x_i]$.	$m_i < 3s_i - (m_{i-1} + m_i) < m_{i-1}$.
(x_i, y_i) is a local maximum of the function.	$y_i > y_{i-1}$ and $y_i > y_{i+1}$.
(x_i, y_i) is a local minimum of the function.	$y_i < y_{i-1}$ and $y_i < y_{i+1}$.
(x_i, y_i) is a global maximum of the function.	$y_i > y_j$ for all $j \ne i$.
(x_i, y_i) is a global minimum of the function.	$y_i < y_j$ for all $j \ne i$.
Average value of *f*	$\frac{1}{x_n - x_0}\sum_{i=1}^{n}\left(y_{i-1}h_i + \frac{1}{2}m_{i-1}h_i^2 + \frac{1}{3}c_ih_i^3 + \frac{1}{4}d_ih_i^4\right)$.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by translating the given function *h* units to the right.	$x_i = \hat{x}_i - h$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by translating the given function *k* units up.	$y_i = \hat{y}_i + k$ for all *i*.

Feature	How to score
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by rotating the given function 90° clockwise.	$x_i = -\hat{y}_i$ and $y_i = \hat{x}_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by rotating the given function 180°.	$x_i = -\hat{x}_i$ and $y_i = -\hat{y}_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by rotating the given function 90° counterclockwise.	$x_i = \hat{y}_i$ and $y_i = -\hat{x}_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by reflecting the given function about the *x*-axis.	$x_i = \hat{x}_i$ and $y_i = -\hat{y}_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the function obtained by reflecting the given function about the *y*-axis.	$x_i = -\hat{x}_i$ and $y_i = \hat{y}_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the derivative of the function.	$x_i = \hat{x}_i$ and $y_i = m_i$ for all *i*.
Given the graph of the function determined by the points $(\hat{x}_i, \hat{y}_i)$, graph the inverse of the function.	$x_i = \hat{y}_i$ and $y_i = \hat{x}_i$ for all *i*.
An exponential function is plotted.	There is a positive real number *a* such that $y_i = a^{x_i}$ for all *i* (within a suitable tolerance).
Identify the domain *D* of the function.	$D = [x_0, x_n]$
Identify the range *R* of the function.	$R = [m, M]$, where $m = \min(y_0, \ldots, y_n)$ $M = \max(y_0, \ldots, y_n)$
Identify *x*-intercepts.	The real number *c* is an *x*-intercept if $x_i = c$ and $y_i = 0$ for some *i*.
Identify the *y*-intercept.	The real number *d* is the *y*-intercept if $x_i = 0$ and $y_i = d$ for some *i*.
The function is even.	For each *i*, there is a *j* such that $x_j = -x_i$ and $y_j = y_i$.
The function is odd.	For each *i*, there is a *j* such that $x_j = -x_i$ and $y_j = -y_i$.

The Parabola $y = ax^2 + bx + c$ Is Plotted

For the curve to look like a parabola, the vertex must be plotted and, additionally, points on either side of the vertex must be plotted in pairs. Thus, if $(\overline{x}, f(\overline{x}))$ is the vertex of the parabola and the point $(\overline{x} + h, f(\overline{x} + h))$ is plotted, then the point $(\overline{x} - h, f(\overline{x} - h))$ must also be plotted. It follows that the number of points plotted must be odd. Because $n+1$ points are plotted, it follows that *n* must be even. The pairing of the points plotted can be checked by specifying that $h_1 = h_n$, $h_2 = h_{n-1}$, and in general, $h_i = h_{n-i+1}$ for $i = 1, \ldots, n/2$.

A Parabola Is Plotted

This is similar, except that the coefficients *a*, *b*, and *c* can be any real numbers.

Function Is Increasing on the Interval $[x_{i-1}, x_i]$

Because the spline is monotonic on each interval, it will be increasing if $y_{i-1} < y_i$.

Function Is Decreasing on the Interval $[x_{i-1}, x_i]$

This is similar.

Function Is Concave Up on the Interval $[x_{i-1}, x_i]$

The function is concave up on the interval $[x_{i-1}, x_i]$ if and only if the second derivative $f_i''(x) > 0$ for all $x \in (x_{i-1}, x_i)$, that is, if and only if

$$c_i + 3d_i\left(x - x_{i-1}\right) > 0 \tag{A1}$$

for all $x \in (x_{i-1}, x_i)$. Equation (A1) will hold for all $x \in (x_{i-1}, x_i)$ if and only if

$$\text{(A1) holds for } x = x_{i-1} \tag{A2}$$

and

$$c_i + 3d_i\left(x - x_{i-1}\right) \neq 0 \text{ for any } x \in \left(x_{i-1}, x_i\right). \tag{A3}$$

Equation (A2) is true if and only if $c_i > 0$. On the other hand, the solution to the linear equation $c_i + 3d_i(x - x_{i-1}) = 0$ is $x = x_{i-1} - c_i / (3d_i)$, so (A3) is true if and only if

$$x_{i-1} - c_i \big/ \left(3d_i\right) < x_{i-1} \tag{A4}$$

or

$$x_{i-1} - c_i \big/ \left(3d_i\right) > x_i. \tag{A5}$$

Because $c_i > 0$, (A4) holds if and only if $d_i > 0$. On the other hand, if $d_i \leq 0$, then (A5) will hold if and only if $d_i > -c_i / (3h_i)$. Thus, in either case, if $d_i > -c_i / (3h_i)$, then either (A4) or (A5) is true, and hence so is (A3).

Thus the function is concave up on the interval $[x_{i-1}, x_i]$ if and only if $c_i > 0$ and $d_i > -c_i / (3h_i)$. From the definitions of c_i and d_i, these two conditions are equivalent to

$$\frac{3s_i - 2m_{i-1} - m_i}{h_i} > 0 \tag{A6}$$

and

$$\frac{m_{i-1} + m_i - 2s_i}{h_i^2} > -\frac{3s_i - 2m_{i-1} - m_i}{3h_i^2}. \tag{A7}$$

Equation (A6) is equivalent to $3s_i - (m_{i-1} + m_i) > m_{i-1}$ and (A7) is equivalent to $3s_i - (m_{i-1} + m_i) < m_{i-1}$. Thus the function is concave up on the interval $[x_{i-1}, x_i]$ if and only if $m_{i-1} < 3s_i - (m_{i-1} + m_i) < m_i$.

Function Is Concave Down on the Interval $[x_{i-1}, x_i]$

This is similar.

Point (x_i, y_i) Is a Local Maximum of the Function

As explained previously, the spline is defined so that, if $y_i > y_{i-1}$ and $y_i > y_{i+1}$, the spline has a local maximum at the point $x = x_i$.

Point (x_i, y_i) Is a Local Minimum of the Function

This is similar.

Point (x_i, y_i) Is a Global Maximum of the Function

Because the spline is monotonic on each interval, it follows that the global maximum of the function must occur at one of the plotted points. It follows, then, that the point (x_i, y_i) will be the global maximum if and only if $y_i > y_j$ for all $j \neq i$.

Point (x_i, y_i) Is a Global Minimum of the Function

This is similar.

Average Value of *f*

The average value of the function *f* over the interval $[x_0, x_n]$ is

$$\begin{aligned}
\frac{1}{x_n - x_0}\int_{x_0}^{x_n} f(x)\,dx &= \frac{1}{x_n - x_0}\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} f_i(x)\,dx \\
&= \frac{1}{x_n - x_0}\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i}\left(a_i + b_i\left(x - x_{i-1}\right) + c_i\left(x - x_{i-1}\right)^2 + d_i\left(x - x_{i-1}\right)^3\right)dx \\
&= \frac{1}{x_n - x_0}\sum_{i=1}^{n}\int_{0}^{h_i}\left(a_i + b_i u + c_i u^2 + d_i u^3\right)du \\
&= \frac{1}{x_n - x_0}\sum_{i=1}^{n}\left(a_i h_i + \tfrac{1}{2} b_i h_i^2 + \tfrac{1}{3} c_i h_i^3 + \tfrac{1}{4} d_i h_i^4\right) \\
&= \frac{1}{x_n - x_0}\sum_{i=1}^{n}\left(y_{i-1} h_i + \tfrac{1}{2} m_{i-1} h_i^2 + \tfrac{1}{3} c_i h_i^3 + \tfrac{1}{4} d_i h_i^4\right).
\end{aligned}$$

where the last equality uses the definitions of a_i and b_i.