Sketching Graphs in a Calculus MOOC: Preliminary Results

Jennifer French, Martin A. Segado, and Phillip Z. Ai

Abstract We developed an online sketch-based input and grading tool, which can be integrated into existing MOOC platforms; this tool allows students to construct graphs using mouse, touch, or pen input and provides immediate targeted feedback. This paper presents some preliminary results from use of this sketching tool in *18.01.1x Calculus 1A: Differentiation*, an online calculus course offered by MIT on the edX platform.

1 Introduction

The ability to assess student graphs of mathematical functions is an important part of introductory calculus curricula. Moreover, the use of qualitative hand drawn sketches is a standard expert problem solving technique in a variety of fields: science, technology, engineering, and math. Pedagogically, tasks such as graph sketching—which require students to construct responses from scratch—offer increased cognitive engagement over passive recognition tasks [3], and have the potential to measure skills that cannot be adequately measured by multiple-choice questions [8].

Mathematical sketching problems are typically completed on paper and graded manually. However, the feedback students receive is delayed due to the logistics of human grading, and may not be detailed. Automatic grading of free-form student sketches can provide immediate, targeted feedback at early stages in the learning process to provide the most beneficial learning gains [6, 11]. Additionally, there has

Jennifer French
Department of Mathematics, MIT, Cambridge, MA, USA, e-mail: jfrench@mit.edu

Martin A. Segado
Department of Mechanical Engineering, MIT, Cambridge, MA, USA, e-mail: msegado@mit.edu

Phillip Z. Ai
Class of 2016, MIT, Cambridge, MA, USA

been an explosion in the popularity of Massively Open Online Courses (MOOCs) amongst both learners and universities. Assessing graphical fluency in such environments is challenging: students submit graphs from the wide range of devices used to access MOOC courseware (e.g. mobile phones, tablets, shared desktops, laptops) and grading must be accomplished at scale.

Based on the perceived learning gains awarded to both on-campus and MOOC students, we created a flexible sketch input and grading tool to support the functionality needed for introductory calculus assessment and deployed this tool in a calculus MOOC [10]. At a high level, the tool comprises a front-end HTML5 web application, which may be embedded in MOOC content (Figure 1), and a back-end grading library, which provides instructors with pre-built functions that use common calculus language to check features of submitted graphs over specified points or intervals (value, slope, domain where increasing or decreasing, concavity over intervals, placement of critical points, locations of asymptotes, and the like). This tool has the following novel characteristics:

- *MOOC compatibility.* The tool integrates with the edX platform (though it may also be used independently) and is designed to support all major browsers without requiring additional plugins. While it may be used with touch or pen devices, it also uses smoothing in order to remain usable with ordinary mouse input to support the diversity of platforms used by learners. To the best of the authors' knowledge, this is the first sketching tool to be used in a MOOC.
- *Extensibility.* The tool has a modular architecture and will be open-sourced by mid-2016. New front-end plugins may be added to support additional drawing functionality, and different back-end grading libraries may be developed to support domains other than calculus. All components share a common data format to maximize interoperability and the potential for widespread analysis. To the best of the authors' knowledge, all related tools are proprietary.
- *Semantic multi-part data entry.* Drawing plugins in the toolbar are given domain-specific labels based on what they are intended to represent (e.g. "Function f(x)" or "Asymptote"), and learners use these plugins to enter multiple types of data on the same graph. The choice of drawing plugins and their associated labels (as well as axis scaling and other variables) are configured by course content creators on a per-problem basis. While semantic labeling is supported at least tangentially in one other application (ETS's proprietary Graph Editor; see below) it was a central feature in both the design of our tool and its use in calculus problems.

More information on the development of the sketch input tool may be found in [10].

2 Related Work

Several others have explored the area of automatic mathematical graph grading. One project with an approach similar to ours is *BeSocratic* [1, 2]—specifically, its *SocraticGraphs* module. This standalone tool also allows students to draw curvedline graphs using either mouse or touch input; the tool checks these sketches against a set of instructor-configured grading rules (specified using a graphical authoring interface) and displays feedback to the student. *BeSocratic* has been deployed in chemistry, molecular biology, and computer science courses at several universities.

Another similar project is the web-based function plot response tool developed at Michigan State University [7] as a way of teaching kinematics graphs. The tool

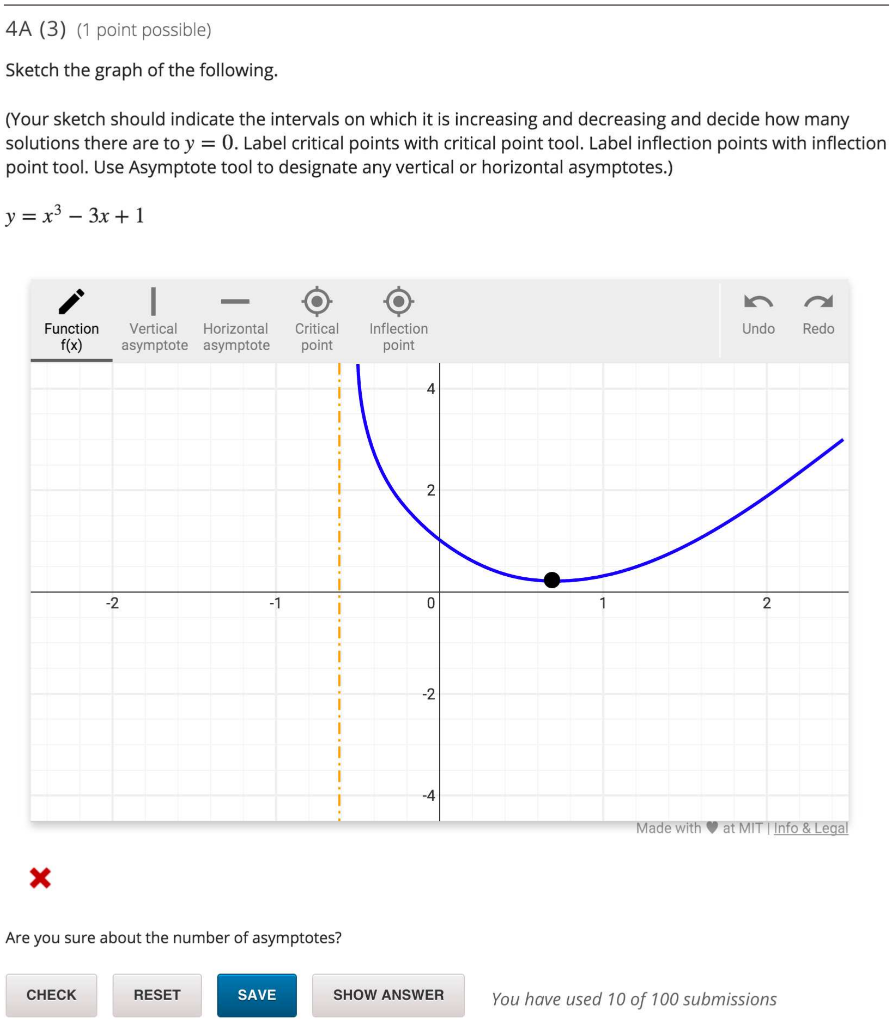


Fig. 1 A graphing problem in MIT's 18.01.1x Calculus MOOC using the sketch input tool (rectangular, center). The tool is embedded into an edX problem block and uses the standard edX platform mechanisms to handle submissions, state persistence, and feedback display. Note the semantically-labeled drawing plugins in the toolbar; the types, order, and labeling of these may be configured on a per-problem basis by course content authors.

is designed to be embedded directly into online course platforms (LON-CAPA in this case) and is also capable of providing immediate targeted feedback. In contrast to both *SocraticGraphs* and our own sketch input tool, this system does not permit direct unconstrained sketching, but instead allows users to construct plots by dragging the control points of cubic Hermite splines. A longitudinal study is underway to assess the formative benefits of this tool and, hopefully, provide more data on the relative merits of graph construction and graph interpretation problems.

A pair of tools which focus more on summative assessment are the *m-rater* scoring engine and accompanying Graph Editor developed by ETS [8, 4]. As with the function plot response tool above, this editor does not support direct sketching, but

Table 1 Summary of problems using the sketch input tool in 18.01.1x. For clarity, inessential details have been omitted from most problem statements

ID	Problem Statement
app1-14-2	Sketch a function f on the interval $-4 < x < 6$ that has the extrema and inflection points determined by the previous problem.[a]
app2-7-1	Sketch the rational function $f(x)$ defined below. $f(x) = \frac{2x^2+1}{x^2-1}, \quad f'(x) = \frac{-6x}{(x^2-1)^2}, \quad f''(x) = \frac{6(3x^2+1)}{(x^2-1)^3}$
app2-17-1	Sketch e^x/x on the interval $-\infty < x < \infty$.
hw4A-4-1	Sketch the graph of the following: $y = x^3 - 3x + 1$
hw4A-5-1	Sketch the graph of the following: $y = x^4 - 4x + 1$
hw4A-6-1	Sketch the graph of the following: $y = x^2/(x-1)$
hw4A-7-1	Sketch the graph of the following: $y = e^{-x^2}$
hw4A-8-1	Sketch the graph of the derivative of the function displayed below.[b] Include zero crossings and label critical points.
hw4B-2-1	Sketch the graph of $f(x) = \frac{1}{x^2-1}$.
hw4B-2-2	Sketch the graph of f' and f'' given $f(x) = \frac{1}{x^2-1}$
hw4B-4-4	In the case where A is critically damped[c] and $a = 1$, sketch the graph of A. (The function sketch can be qualitative. Label any asymptotes, critical points, and inflection points qualitatively.)
final-4-1	Draw the graph of the function (qualitatively accurate). $y = \sqrt{x+1}/(x-b), \qquad 1 < b < 2$ (Your sketch should indicate the intervals on which it is increasing and decreasing and decide how many solutions there are to $y = 0$. In this problem, you do not need to test the second derivative to identify the concavity. There are no inflection points. Label critical point and asymptotes using the appropriate drawing tools.)
final-8-2	Sketch the graph of $f(x) = 1 + \frac{a}{x} + \frac{a}{x^2}$ (same function as above[d]) on $-\infty < x < \infty$ showing the horizontal and vertical asymptotes. While the sketch need not be to scale, please make sure the x–coordinate of all critical point(s) and inflection point(s) are accurate.

[a] The problem preceding this one asked students to find critical points and inflection points of a function given the signs of its first and second derivatives.
[b] The original problem statement included a graph of a rational function with labeled asymptotes, one critical point, and one inflection point.
[c] An earlier problem detailed a mathematical model for a heavy swinging door with angle A.
[d] An earlier problem asked students to find the intervals on which the given function was increasing, decreasing, concave up or concave down.

instead allows users to graph points, straight lines, and smooth curves (splines) by dragging points onto a set of axes. The *m-rater* scores straight lines and quadratic curves analytically, and scores more general curves by checking them at several locations. The ETS Graph Editor shares our support for multi-part data entry and also appears to support semantic labeling.[1]

A completely different approach to grading sketched graphs was evaluated by Lukoff as part of his doctoral dissertation [9]; instead of evaluating the graph according to a fixed rubric, Lukoff trained support-vector-machine classifiers on a set of manually-labeled graph sketches (which were produced and labeled online using Amazon Mechanical Turk). The technique showed promise, but does not yet appear to be in use in any educational environments.

3 Methodology

A set of thirteen graphical input problems[2] were created and included in the 18.01.1x calculus MOOC [5]. The problems are summarized in Table 1. In the hopes of avoiding frustration with the new tool, most problems allowed effectively unlimited tries with no penalty; only the two final exam problems had an imposed limit of five attempts each.

The sketch input tool itself was built and refined in parallel with course content to ensure that its functionality aligned with course goals and assessments. The initial grader tolerances and behavior were based on data from an earlier prototype of the tool, but updated based on student reports of graphs that did not grade correctly (e.g., due to unforeseen drawing styles). Such graphs were examined manually to ensure their correctness and, when possible, the grading library was updated to support them.

4 Preliminary Results

A total of 8187 non-blank final responses were submitted by 871 unique users for the sketch problems described in Table 1. Of these, 7311 (89.3%) were classified as correct by our grader code. A breakdown by problem is provided in Table 2. Overall, the data are encouraging; most students who submitted responses were able to arrive at the correct answer, many in their first attempt, with 2–3 attempts being the norm. The median number of attempts taken for incorrect submissions was similar, except in the two final exam problems, where most students demonstrated enough persistence to use up all five permitted attempts.

[1] See the “Graph Editor" figure on page 3 of ETS’s automatic scoring technologies brochure: https://www.ets.org/Media/Home/pdf/AutomatedScoring.pdf

[2] A tutorial problem and several problems created with an earlier prototype of the sketching tool are not include in these results.

Table 2 Summary of non-blank submissions for graphing problems described in Table 1

ID	No. Final Responses	Percent Correct	No. Attempts (by percentile) Correct 10th	50th	90th	Incorrect 10th	50th	90th
app1-14-2	700	89.3	1	2	5	1	2	6
app2-7-1	669	87.9	1	2	7	1	3	9
app2-17-1	603	87.9	1	2	5	1	2	8
hw4A-4-1	674	94.1	1	2	5	1	3.5	9.2
hw4A-5-1	646	95.2	1	2	4	1	2	6
hw4A-6-1	623	95.2	1	2	4	1	2	6.1
hw4A-7-1	616	94.0	1	3	6	1	3	10
hw4A-8-1	575	87.0	2	3	9	1	3	13.4
hw4B-2-1	654	95.3	1	2	4	1	2	8
hw4B-2-2	617	88.3	1	3	7	1	2	7.9
hw4B-4-4	506	92.9	2	3	7	1	3	9
final-4-1	648	73.5	1	2	4	1	5	5
final-8-2	656	81.2	1	1	4	1	5	5

The graphs of individual submissions may also be examined for insight. Figure 2 shows a subset of 25 such submissions, drawn randomly from the pool of correct responses to the *final-8-2* problem. One notable feature is the degree of variability in these submissions: all indeed appear correct given the problem statement, but details such as the vertical position of critical points and the way the curves approach the vertical asymptote vary considerable between graphs.

Submissions that were graded as incorrect proved more difficult to examine in aggregate. Since each submission consists of multiple points and line segments, it becomes difficult to identify which of these belong together when a large number are overlaid, even when submissions are first grouped by the feedback message that was displayed to students. Nevertheless, some of these grouped visualizations did convey interesting trends; the submissions shown in Figure 3 clearly illustrates some common misconceptions, with the subplot on the left identifying students who

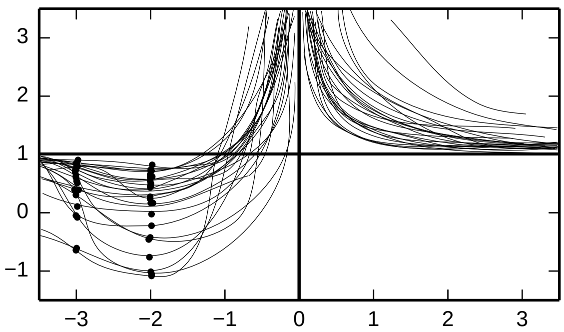


Fig. 2 Subset of 25 correct submissions for problem *final-8-2*. The answers vary considerably within the parameters permitted by the problem statement but all appear qualitatively correct

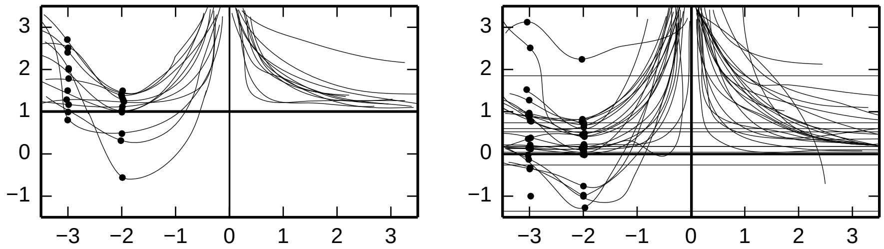
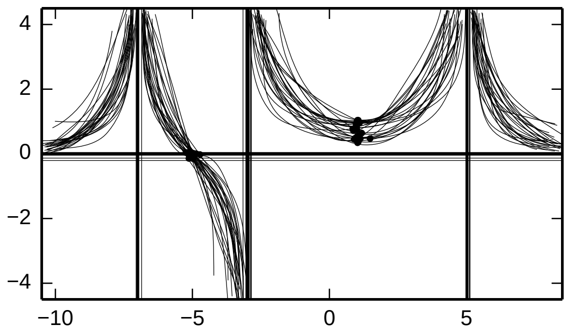


Fig. 3 Subset of incorrect submissions for problem *final-8-2*. Submissions on the left were rejected with feedback "What is the relationship of your function to the horizontal asymptote?" while those on the right were rejected with feedback "Check the location of your horizontal asymptote"

Fig. 4 Subset of 25 correct submissions for problem *hw4A-8-1*, one of the more complex graphs students were asked to sketch in 18.01.1x



didn't realize that $f(x) \to 1^-$ as $x \to -\infty$ and many of those on the right identifying $y = 0$ instead of $y = 1$ as the position of the horizontal asymptote.

Another takeaway from our data is that MOOC students are capable of entering complex plots quite accurately with this tool (e.g. Figure 4), despite our expectation that at least some of these were drawn using mouse input instead of the more natural touch or pen modalities.[3]

5 Future Work

A variety of opportunities exist for future work. These may be loosely divided into four categories:

Data Analysis There is considerable room for analysis of existing results. Manual grading of individual responses could identify grading errors and lead to improvements in the grader library default tolerances. Clustering approaches for identifying common misconceptions in solutions may also be valuable; in particular, it would be interesting to identify features most often missing from nearly-correct graphs

[3] While information on the devices used by students is indeed logged by edX platform, it is not exported by edX's instructor-facing interface. We hope to obtain this information in the future.

and find ways to improve our feedback mechanism to guarantee student success. It would also be interesting to use the built in tolerances in our grading library to determine confidence intervals for correctly- and incorrectly-graded student sketches. These borderline cases could then be flagged for review, which would lead to further improvements of the grading library algorithms.

We are also collecting new data from a second MOOC course (*18.01.2x Calculus 1B: Integration*), as well as two on-campus courses currently running at MIT: (*18.01 Calculus I* and *2.001 Mechanics and Materials I*). The tool is being used in new ways in these courses. In the new MOOC, students sketch areas and regions between curves in order to set up integration problems. This allows us to provide scaffolding and feedback on integration problems that would be impossible without the use of graphing. All of the MOOC calculus sketching problems are being used on-campus, which gives us the opportunity to interview students directly about their experience using the tool. In the mechanics course, the tool is being used to grade axial force diagrams: students are shown an image of a beam with forces along its length and asked to draw the axial force resultant directly on top of the image (as well as label any discontinuities).

Grading Improvements Certain limitations of the grading library were identified through the use of this tool in 18.01.1x. One particular challenge is checking the curvature of functions in a way that is consistent with a human grader. We are implementing more robust grading algorithms and using the student data to determine the appropriate tolerances. Another limitation lies in the typical structure of our grading scripts: to grade a complex graph, it is often easiest to check features of the free-form curve relative to student labeled landmarks such as asymptotes, critical points, and inflection points. If a student fails to provide these landmarks, the grader stops immediately and gives students feedback on the missing items; however, providing feedback in this order may be in conflict with the optimal learning process.

We hope to simplify the instructor interface with the grading library. Currently, an instructor writes simple Python scripts specifying the conditions to check; for problems in a wide range of disciplines, however, this process could be automated (e.g., by having grading conditions for the time response of an electrical circuit or mechanical system auto-generated from the problem specifications). This also opens up the exciting possibility of auto-generating entire randomized *problems*: for example, the number and locations of forces on a mechanical beam could be randomized, thus providing students with infinite practice opportunities for a particular type of mechanics problem.

It may also be interesting to explore peer grading or instructor-assisted grading approaches for graphical problems, though this is not on our immediate roadmap.

Learner Interface Improvements The addition of common editing features (e.g. dragging, deleting, and clipboard functions) would improve overall usability. We hope to add a spline tool similar to that in [1] and [8], making the tool more usable for learners lacking a pen or touch device. Such a spline tool could also feature a keyboard-and-screenreader interface to enable its use by visually- and mobility-impaired students.

Collaboration Finally, once the tool is stable and accessible, we will be releasing it under an open-source license so that others can both use and improve it. Open access is key, and important so that we can begin to forge collaborations both in and out of MIT.

6 Conclusions

A free-form sketch-input and automatic grading tool designed for use with pen, touch, and mouse input was deployed in an calculus MOOC delivered online through edX. In addition to focusing on MOOC compatibility, the key features of the tool include a modular architecture designed for open-source collaboration, and the gradeability of multiple customizable, semantically-labelled plugins in one student sketch based on standard calculus terminology.

An analysis of 8187 responses collected with the graphical input tool showed that the median total attempts for students who do not get a correct answer is 2 or 3, suggesting that if we have not given a student useful feedback quickly, they may not persist to get a correct response. Those who get correct responses tend to do so on average in 2 or 3 attempts, with the 90^{th} percentile using between 4–9 attempts depending on the difficulty of the problem, despite essentially unlimited attempts. Qualitatively, our grader generally performed as expected; however, analysis of the incorrect graphs showed that the feedback was not always correlated with all of the misconceptions in incorrect graphs, and a finer level of feedback based on student errors may helpful to improve learning gains.

Preliminary results are encouraging. While considerable work remains to be done, we expect that the use of such graphical tools in both MOOCs and classroom environments enhances learning gains in student ability to sketch functions and provides a rich source of data for future study.

References

[1] Bryfczynski S, Pargas R, Cooper M, Kylmkowsky M (2012) BeSocratic: Graphically assessing student knowledge. In: Proceedings of the IADIS International Conference Mobile Learning, IADIS, pp 3–10

[2] Bryfczynski S, Pargas RP, Cooper MM, Michael Klymkowsky JH, Grove NP (2015) Classroom uses for BeSocratic. In: The Impact of Pen and Touch Technology on Education, Springer, pp 127–136

[3] Chi MTH, Wylie R (2014) The ICAP framework: Linking cognitive engagement to active learning outcomes. Educ Psychol 49(4):219–243

[4] Fife JH (2013) Automated scoring of mathematics tasks in the Common Core era: Enhancements to m-rater in support of CBAL^{TM} mathematics and the Common Core assessments. ETS Res Rep Ser 2013(2):i–35

[5] French J, Wang S, Jerison D, Staffilani G (2015) Calculus 1A: Differentiation. Online course. https://www.edx.org/course/calculus-1a-differentiation-mitx-18-01-1x. Accessed 30 Jan 2016

[6] Hattie J, Timperley H (2007) The power of feedback. Rev Educ Res 77:81–112

[7] Laverty J, Kortemeyer G (2012) Function plot response: A scalable system for teaching kinematics graphs. Am J Phys 80(8):724–733

[8] Livingston SA (2009) Constructed-response test questions: Why we use them; how we score them. Educational Testing Service R&D Connections 11

[9] Lukoff B (2010) The design and validation of an automatically-scored constructed-response item type of measuring graphical representation skill. PhD thesis, Stanford University

[10] Segado M, French J (2015) Interactive sketch input tool for MOOCs. Paper presented at the Learning with MOOCs II Workshop, Columbia University, 2–3 Oct 2015. http://web.mit.edu/~jfrench/www/pdf/sketchinput_paper.pdf. Accessed 29 Jan 2016

[11] Shute VJ (2008) Focus on formative feedback. Rev Educ Res 78:153–189