Inequality: multi-modal equation entry on the web

Andrea Franceschini
University of Cambridge, UK
andrea.franceschini@cl.cam.ac.uk

James P. Sharkey
University of Cambridge, UK
james.sharkey@cl.cam.ac.uk

Alastair R. Beresford
University of Cambridge, UK
alastair.beresford@cl.cam.ac.uk

ABSTRACT

Online learning in STEM subjects requires an easy way to enter and automatically mark mathematical equations. Existing solutions did not meet our requirements, and therefore we developed *Inequality*, a new open-source system which works across all major browsers, supports both mouse and touch-based entry, and is usable by high school children and teachers. Inequality has been in use for over 2 years by about 20 000 students and nearly 900 teachers as part of the Isaac online learning platform. In this paper we evaluate Inequality as an entry method, assess the flexibility of our approach, and the effect the system has on student behaviour. We prepared 343 questions which could be answered using either Inequality or a traditional method. Looking across over 472 000 question attempts, we found that students were equally proficient at answering questions correctly with both entry methods. Moreover, students using Inequality required fewer attempts to arrive at the correct answer 73% of the time. In a detailed analysis of equation construction, we found that Inequality provides significant flexibility in the construction of mathematical expressions, accommodating different working styles. We expected students who first worked on paper before entering their answers would require fewer attempts than those who did not, however this was not the case ($p = 0.0109$). While our system is clearly usable, a user survey highlighted a number of issues which we have addressed in a subsequent update.

ACM Classification Keywords

K.3.m. Computers and Education: Miscellaneous

Author Keywords

Equation entry; symbolic manipulation; Computed Aided Assessment; automated marking; teaching physics; teaching mathematics

INTRODUCTION

Symbolic, or algebraic, manipulation is a fundamental skill in all Science, Technology, Engineering, Mathematics (STEM) subjects and therefore support for symbolic entry is a key feature in many online STEM learning platforms. Isaac Physics[1] is no exception. It provides a web-based platform to help students improve problem-solving skills in physics. Students aged 15 to 19 use Isaac to solve problems from our question bank of over 4 000 problems. Isaac automatically marks answers and provides tailored feedback in case of incorrect answers. The platform includes problems of varied difficulty, from questions designed to help students prepare for their GCSE[2] and A Level[3] qualifications in the UK – or equivalent in other countries – through to admission to university, and first year undergraduate physics.

Problems on Isaac include one or more parts, each requiring an answer in one of three formats: multiple choice, numeric or symbolic. Symbolic entry is particularly challenging as it needs to work on mobile and desktop in a way that is easy to use without requiring specialist knowledge. Pre-university students are used to the traditional mathematical notation taught in schools, using pen and paper; we avoid specialist notations, such as TEX, which require extra effort to learn and use.

In this paper we present Inequality, a graphical, drag-and-drop, symbolic editor for entering mathematical formulæ on the web. Inequality works across all the major web browsers, and across mouse- and touch-based devices. We designed Inequality to present students with the representation of symbolic maths they use in school. We were also keen to provide considerable flexibility in the way students construct and manipulate expressions in order to reduce blind guessing and detect copying.

In summary, we make the following contributions: i) we present Inequality, a novel, web-based, symbolic editor which works across both mouse- and touch-based interaction methods; ii) by looking across a corpus of 472 000 question attempts, we demonstrate that students are equally proficient at answering questions with Inequality and a traditional approach; iii) students exhibit significant variability in the construction of their answers, which contrasts with traditional methods such as multiple choice (which are easily guessed through multiple attempts), or numeric responses (for which it is hard to detect copying since there is little variety in how the question is answered); and iv) when compared to traditional methods, students using Inequality required fewer attempts to arrive at the correct answer 73% of the time.

RELATED WORK

The typical approach to computer-based symbolic editing is text based, where users type mathematical expressions using a keyboard. Within this approach, two styles dominate: one explicitly incorporates mathematical layout, notably used

[1] https://isaacphysics.org/

[2] General Certificate of Secondary Education

[3] General Certificate of Education (GCE) Advanced Level

L@S'19, June 24–25, 2019, Chicago, IL, USA

ISBN 978-1-4503-6804-9/19/06. . . $15.00

DOI: https://doi.org/10.1145/3330430.3333625

by Wolfram Mathematica and Microsoft Word Equation Editor; and the other uses linear expressions, of which TEX and SymPy[6] are examples. In the first style, users select templates of mathematical substructures which appear as a set of empty boxes that are then filled in by typing. While this allows editing in a way that is familiar to students in our target group, it also requires them to plan the expression ahead of entering it, and makes it somewhat cumbersome to re-arrange its parts once they are in place. Conversely, in the second style, building an expression in different parts and re-arranging them afterwards can be easier, but the linear form is strikingly unlike the one normally used in schools, and the syntax may not be natural to students in our target group. For example,

$$\texttt{sqrt(x/(y+z))} \qquad \text{vs} \qquad \sqrt{\frac{x}{y+z}}$$

Students normally deal with symbolic maths by hand writing standard two-dimensional notation, and plenty of evidence suggests that, for non-specialist users, handwriting recognition is the best way of entering mathematical formulæ into computers [4, 5, 8, 9]. However, although handwriting could be a viable input technique on touch-based devices, it is unusable on a desktop or laptop computer without specialist hardware. The requirement for specialist hardware is incompatible with our aim of reaching a wide range of schools with minimal additional overhead. In our case, analytics for 2017 showed that 66% of site visits use a desktop or laptop computer, as opposed to a mobile device, therefore we decided to develop a system that could serve both types of devices. Nevertheless, the use of two-dimensional mathematical layout for symbolic manipulation is a useful tool that allows learners to build mental models of moving pieces of expressions by moving them on the page. [3, 1]. Being able to mentally move pieces around and reason about them is also an invaluable part of problem solving skills in physics, where diagrams are used to explore a problem and plan a solution.

INEQUALITY, A SYSTEM OVERVIEW

Our system is composed of a drag-and-drop GUI which allows students to build and manipulate mathematical expressions, and a web-service that determines whether the students provided the correct answer to a given problem on Isaac Physics.

A graphical, drag-and-drop, symbolic editor

The approach builds on ideas found in our previous system, Equality [2], which allowed users to freely place mathematical symbols on a canvas via drag-and-drop, parsing these symbols into a mathematical expression using their relative positions. A significant drawback of this approach was the frailty of the canvas parser – i.e. it was possible to generate two very different expressions by moving symbols by only one or two pixels – and the potentially long processing time that could delay feedback significantly. Since we do not need to teach mathematical layout, we decided that a more robust interface would only allow users to place symbols in specific, meaningful, and clearly marked spots – visually represented by the blue circles in Figure 2 that we call *docking points*. This has two benefits: i) it allowed us to greatly simplify the parser by creating a linked data structure that could be traversed and translated into a desired output format such as SymPy (which we use in the back-end to evaluate the answers), MathML, and our own Abstract Syntax Tree format; and ii) it removed ambiguity and uncertainty for users in terms of symbol placement.

The workflow for a symbolic question with Inequality is shown in Figure 1. If the answer is incorrect, custom feedback is provided if the wrong answer is a mistake that we can identify automatically through methods including common misconceptions, lack of simplification, or based on our own analysis of common wrong answers found in past attempts by other students.

A service that evaluates symbolic expressions

Verifying the equivalence of symbolic expressions is relatively easy with Computer Algebra Systems (CAS). However, the typical CAS would generally deem two expressions such as $(x-1)(x+1)$ and x^2-1 as equivalent, and this is a problem when questions require students to factorise a polynomial. Similarly, we needed a way of checking equations and inequalities that preserved the two sides of the expression and checked them individually, in addition to checking overall correctness.

For these reasons, we developed a component written in Python based on the SymPy library for Computer Algebra [6]. When students build their answers using Inequality, the application automatically generates a SymPy-compliant expression that is then sent to the symbolic checker running on our server. This service examines the answer submitted against possible correct and incorrect answers, and returns appropriate feedback to students. The ability to check structural as well as mathematical equality allows us to ensure answers comply with our pedagogical goals, and also enables targeted feedback on stylistic mistakes if desired, e.g. $\cos(x)$ vs $\sin(x+\frac{\pi}{2})$.

EVALUATION

In the remainder of this paper we evaluate the effectiveness of our system and compare our approach against more traditional methods. In particular, we considered the following research questions.

1. Do students require fewer attempts to answer questions correctly in symbolic format compared to more traditional formats?
2. How do students use Inequality to answer symbolic questions?
3. What is the experience of students using Inequality to answer symbolic questions?

We use data collected directly through our on-line platform (click streams and answer attempts), and through a questionnaire offered to students on the platform. The next two sections describe our data collection methodology and metrics.

Application logs

Isaac Physics collects a variety of usage data in the form of click streams. These include actions such as visiting a page, answering a question, and so on. Inequality also logs click

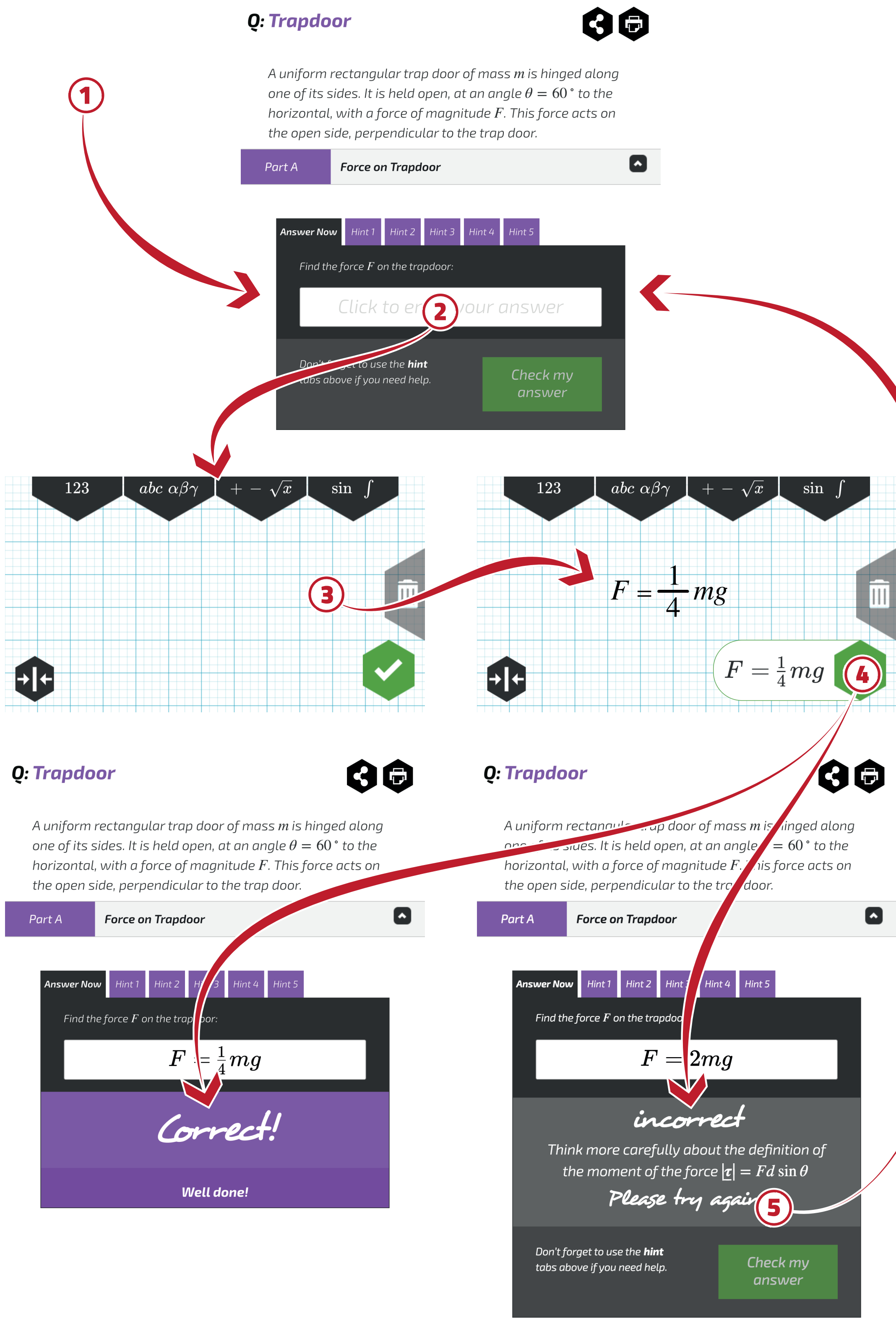


Figure 1: Workflow for answering symbolic questions. A user starts at step ① with a symbolic question, and clicks on the preview box ② to open Inequality. After entering and editing the desired formula ③, the user can either exit the editor discarding the changes, or click on the green tick ④ to confirm their entry and go back to the question screen. After going back to the question screen the user can submit their answer, go back into the editor and change their answer, or leave the page altogether. After submitting their answer by clicking on "Check my answer", if the answer is correct, the user can then move on to another question. In case of an incorrect answer, the user has the option to restart ⑤ or move on.

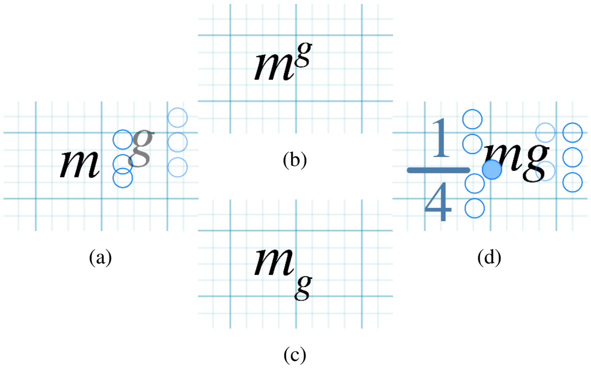


Figure 2: In vignette (a), the letter g is dragged near the symbol m, and can be docked as an exponent (b), a subscript (c), or a multiplier, as vignette (d) shows. Vignette (d) also shows the product of m and g being attached as a multiplier to $\frac{1}{4}$.

streams. These contain a detailed lists of all the actions that a user performs while interacting with the editor, from the moment they open it, to the moment they close it to submit their answer. The list of actions that we log include dragging and dropping symbols, building and manipulating formulæ, and so on. Table 1 summarises the actions we used.

We launched Inequality in August 2016, converting 420 existing questions from either multiple choice or numeric format into our new symbolic format. In this way, we can compare equivalent questions in different formats to understand the effect of the format on the student performance and behaviour.

In our data set of editor sessions, there are approximately 224 thousand unique sessions, comprising between 3 and 721 actions each, with an average of 23.5 actions and a median of 17. These correspond to a resultant expression size between 1 and 79 symbols, with a median of 6. We define the expression size by the number of symbols of which it is composed, including repetitions. For example, $I = V/R$ is composed of 5 symbols, and $x^2 + x + 1$ is composed of 6 symbols.

Questionnaire

Analysing click streams from Inequality allowed us to investigate user behaviour. In particular, we could explore the correct and incorrect answers users submitted, how they built them, and how many of these each user submitted. However, this data only allows us to gain limited insight on the experience of the users while entering and manipulating formulæ. For example, we cannot tell whether users work out their solutions on paper and then simply use the editor to build the expression and submit their answers, or whether they use the editor to manipulate expressions as they work on the problem. In addition, we cannot capture data regarding user experience – for example, whether they find Inequality easy or difficult to use, whether they find it helpful or distracting, and so on.

For these reasons, we asked a random selection of registered students to complete a short questionnaire. To do so, we presented our selected cohort with a pop-up message asking them to answer our questionnaire. Students could ignore our request without penalty if they wished.

The questionnaire covered the students' reasons for using the equation editor, how often they used it, and how they selected questions. To do so, the students were presented with a series of statements on their experience using the equation editor, and asked to rate them on a 5-point Likert agreement scale with the categories Strongly Disagree, Disagree, Neither Agree Nor Disagree, Agree, Strongly Agree. Strongly worded statements were used to elicit a strong response and mitigate central tendency bias. We also repeated statements with different formulations as a way to mitigate acquiescence. We chose a traditional 5-point Likert scale in order to provide a neutral option for those who genuinely feel that way, and we clearly marked it as a middle option, rather than an opt-out [7]. All the groups of questions had an optional comment box for additional insights not captured by closed questions. The last section of the questionnaire collected optional demographic information such as gender and school year. This section also provided a final opportunity to provide general comments on the equation editor and symbolic questions.

We received 685 responses, 122 of which we regarded as invalid, bringing the total number of respondents down to 563. We filtered out invalid responses by taking into account the presence of clearly contradictory answers, and of answers that followed meaningless patterns to categorical questions – such as all identical answers, or deliberately alternating answers.

Metrics

To evaluate the student performance, we considered two metrics: the time spent working on questions, and the number of attempts necessary to answer questions correctly.

A quick survey of the application logs reveals that time spent is not necessarily a good metric, as it can vary widely and depend on individual ability, attitude and behaviour. For example, we found that the time between successive answer attempts ranged from a fraction of a second to several days. These indicate behaviours ranging from rapidly clicking through all the possible options of a multiple choice question, to leaving a browser tab open for several days between attempts. Furthermore, students are known to work at very different rates, depending on their ability and work habits, therefore, even considering only diligent students that complete their work in reasonable and comparable time-frames, we would be likely to encounter significant noise.

Therefore we instead decided to focus on the number of attempts students make before submitting a correct answer. This has the clear advantage of providing a non-ambiguous unit of measure. A survey of the data revealed that, although variation is expected, it is typically of a manageable magnitude – i.e., very rarely do users exceed 20-30 attempts on any single question, while most of them need far fewer than 10 to arrive at the correct solution.

In addition, we also consider ancillary information about incorrect attempts, such as whether the attempt is a common wrong answer that we already know of, or, for numeric questions, errors due to significant figures.

Symbol	Description
OPEN	The editor is opened and a new session starts
CLOSE	Click on the green tick to close the editor
NAVIGATE_AWAY	The editor is closed in some other way (e.g., by closing the browser)
DRAG_POTENTIAL_SYMBOL	A symbol is dragged from the menu
DROP_POTENTIAL_SYMBOL	A symbol that was dragged from the menu is dropped on the canvas
DOCK_POTENTIAL_SYMBOL	A symbol that was dragged from the menu is attached to a symbol on the canvas
ABORT_POTENTIAL_SYMBOL	A symbol that was dragged from the menu is dropped back the menu
TRASH_POTENTIAL_SYMBOL	A symbol that was dragged from the menu is dropped on the trash bin
DRAG_START	A symbol is picked up from the canvas for dragging
UNDOCK_SYMBOL	A symbol that was attached to another symbol is picked up for dragging
DOCK_SYMBOL	A symbol that was on the canvas is docked to another symbol on the canvas
DROP_SYMBOL	A symbol that was on the canvas is dropped back on the canvas, unattached
TRASH_SYMBOL	A symbol that was on the canvas is dropped on the trash bin

Table 1: An explanation of the equation editor actions that we analysed.

FINDINGS FROM THE APPLICATION LOGS

Since August 2016, 73 370 registered students were active users of the platform and 20 009 attempted symbolic questions. Of these attempts, 14 810 were assigned symbolic questions by their teachers and 5 199 students chose to attempt symbolic questions independently. We consider students to be active based on whether they attempted at least one question a month.

In this paper, we analysed logs that were collected between the 1st of August 2016 and the 15th of March 2018.

We examined 420 questions, of which 343 had responses in both symbolic format as well as either multiple choice (155) or numeric (188) formats. We computed the average number of attempts users submitted before providing a correct answer for each individual question, and then compared the averages between each question type.

We found that, on average, 72.9% of the questions required fewer attempts when answered symbolically. There are at least two factors that could generate an inflated number of attempts in the multiple choice and numeric formats. Firstly, since we do not punish incorrect attempts, students can submit as many incorrect answers as they like. Therefore multiple choice questions may encourage students to try all the available options before finding the correct one. Secondly, a considerable number of students struggle with significant figures, and our data shows that these account for about 31.2% of the incorrect attempts on numeric questions. When we excluded incorrect attempts due to significant figures from the comparison above, we found that, on average, 60.9% of the questions (including numeric and multiple choice) required fewer attempts after conversion to the symbolic format. Figure 3 shows the distribution of pairs of average number of attempts.

Based on question type before conversion, we found that

- 60.1% of 188 questions that were originally numeric required fewer attempts after conversion (excluding incorrect attempts due to significant figures), and
- 61.9% of the 155 questions that were originally multiple choice required fewer attempts after conversion.

On Isaac Physics, the vast majority of questions have an accompanying series of hints to guide the students towards the solution, without providing the final answer. We found that, for the questions that were originally multiple choice, the conversion to a symbolic format resulted in a slight increase in the use of hints. This may be explained by the behaviour of some students to try all multiple choice options blindly. Conversely, we did not find any significant difference in hints usage when the questions were originally numeric.

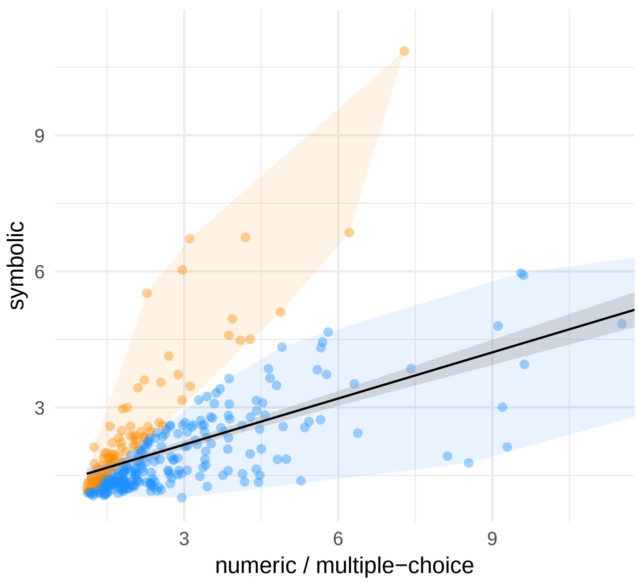
We applied Welch's (unequal variance) t-test to the format comparisons above and found these results to be statistically significant with $p < 0.001$, except for the comparison to questions previously in multiple choice format, where $p = 0.0102$.

Of the 420 questions that were converted to the symbolic format, some stand out with respect to the ratio between the average number of attempts in symbolic format and the average in their old numeric or multiple-choice formats. The questions in the orange cluster in Figure 4 have a ratio higher than $5/4$ – in other words, a question that required 4 attempts on average in either numeric or multiple-choice format, requires at least 5 attempts on average in symbolic format. We therefore decided to take a closer look at these 37 questions in order to understand what, if anything, could be making them harder to answer in symbolic format. We compared these with questions from the blue cluster, particularly with those 96 questions for which the ratio of average attempts after/before conversion is lower than $4/5$ – i.e., a question that used to require 5 attempts is now requiring 4 or fewer, on average.

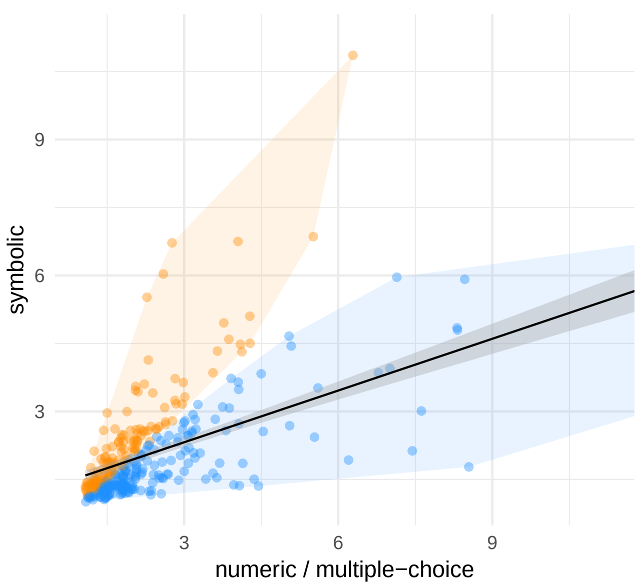
We analysed various aspects of the questions and of the answers, including the size of the expressions required to answer in symbolic form, and the level of difficulty set by content creators. We could not find any significant effect given by any of these factors, nor by the difference in the questions' previous format. A survey of the question contents and problem-solving requirements suggests that the questions in the orange cluster of Figure 4 may be inherently hard to solve, but it is not clear whether the different formulation of the questions before and after conversion has any significant effect.

We analysed the sequences of actions performed in constructing the expressions in the equation editor. An example of the sequences that we encountered is shown in Figure 5. We found that, for the first two or three actions performed on any particular question cluster, the majority of the students fell into three or four groups. However, by the time students perform their third or fourth action, the paths they take diverge massively.

A diagram showing all the actions on a particular question forms a tree where the first few branches closest to the root are very frequently chosen, as in Figure 5, but quickly fans out



(a) Including all question attempts.



(b) Excluding incorrect attempts due to significant figures.

Figure 3: Comparison of the average number of attempts per question on the old versions (x axis) against the new versions (y axis). Each dot represents a question pair, clustered depending on whether, after conversion to symbolic, they required fewer (blue) or more (orange) attempts, on average. The black line and confidence band represent the linear model fitted to the data. One extreme outlier in the blue cluster is excluded from the plot but accounted for in the analysis.

into chains of actions followed by only a handful of students. In Figure 5 the highlighted rectangle shows a student dragging the letter α from the menu to the canvas, then from the canvas to a docking point. The same effect could have been achieved by dragging the letter from the menu and directly docking to a symbol that was on the canvas.

In conclusion, the data contains evidence that some students do build the same expression in the same way. This is the case when expressions are built strictly left-to-right, indicating that the students know the expression they want to build in advance – i.e., they may have worked it out on paper – and they have learned the most efficient way of building expressions with Inequality. However, the data also reveals that this is not the typical way in which most students use the editor, preferring instead to manipulate the formulæ in the editor itself. To confirm this behaviour, we included statements 4.2 and 4.5 in the questionnaire (Table 2), and the responses suggest that our interpretation of the application logs may be correct.

Along with the variety of ways of building expressions, we also considered the number of distinct expressions that were submitted as correct answers to any given question, in order to see how much flexibility our system affords students. We took the SymPy expressions as a proxy for this measure. We have observed from the logs that submitted expressions can vary as little as "`ma`" vs "`F == am`", or as much as

- "`(((2*E_k)/(m)))**(((1)/(2))) == v`" vs

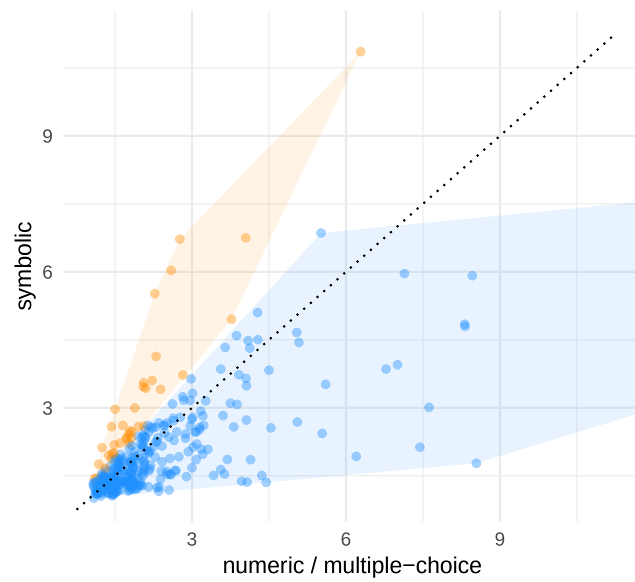


Figure 4: Comparison of the average number of attempts per question on the old versions (x axis) against the new versions (y axis). The smaller, orange cluster comprises questions that required more than $1.25\times$ as many attempts as they did in their old formats, with those at the top of the cluster requiring as many as $2.4\times$. The dotted line is the identity line, for comparison. One very extreme outlier in the blue cluster is excluded from the figure.

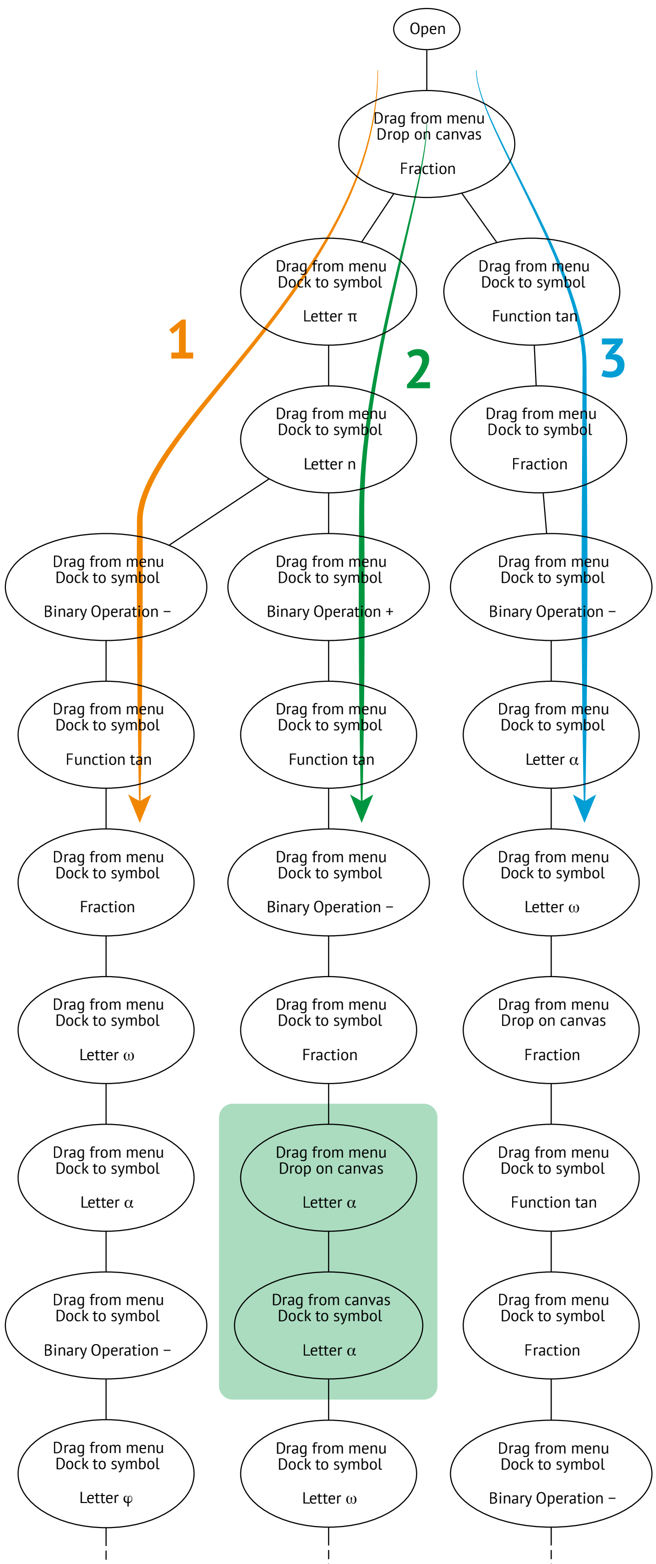


Figure 5: Three ways of building the expression $\frac{\tan^{-1}(-\frac{\alpha}{\omega})+n\pi-\phi}{\omega}$

- "`v == sqrt(2*(E_k)/(m))`".

Figure 6 summarises this by placing the number of distinct expressions that a question has received on the horizontal axis, and counting how many questions have received that number of distinct expressions. A large proportion of the questions we examined have received between 1 and 10 distinct expressions, and only a handful have received more than 50. A closer look at some of the answers that present higher variability reveals that such high variability is often due to commutation, parenthesising, lack of simplification, and so on. The following are three examples from a question with 36 unique answers.

- `(2)/(sqrt(3)) * m*g` $= \frac{2}{\sqrt{3}}mg$
- `(2*m*g*sqrt(3))/(3)` $= \frac{2mg\sqrt{3}}{3}$
- `(m*g)/((sqrt(3))/(2))` $= \frac{mg}{\frac{\sqrt{3}}{2}}$

In general, an expression with many terms offers ample opportunity for equivalent variants and our system copes well with such variability.

FINDINGS FROM THE QUESTIONNAIRE

We analysed 563 valid responses, of whom 311 self-declared as male and 179 female; 73 did not specify a gender. These students were largely in Year 12 (typically aged 16-17) and started using Isaac Physics in the same year (Figure 7). We had a few responses from GCSE students (Year 10), but these are currently a small fraction of our registered students, as the GCSE material has only been published recently. About two thirds of the respondents stated they worked on Isaac Physics in response to assignments set by their teachers. On average, one third of their work involved symbolic questions. Comments reveal that, while students recognise the importance of symbolic work, the perceived difficulty of many questions, combined with some of the quirks of the editor, may negatively influence the uptake of symbolic questions among both teachers and students. In fact, 69.2% of the students declared working with the editor less than once every two weeks, while only 12.9% use it at least two or three times per week.

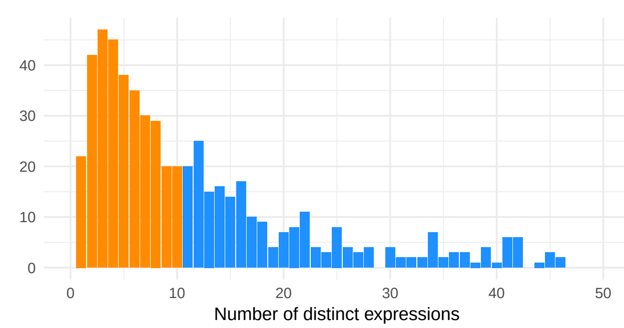


Figure 6: Number of questions (y axis) that elicited a certain number of distinct expressions (x axis) as a correct answer. Most questions elicited between 1 and 10 distinct expressions.

	Statement	SD	D	N	A	SA
Q 4.1	I often avoid symbolic questions	50	47	37	196	**233**
Q 4.10	I enjoy using the equation editor to answer questions on Isaac Physics	101	110	**217**	98	37
Q 4.2	I always work out the answer on paper and then use the equation editor to enter it	74	117	**186**	114	72
Q 4.5	I always use the equation editor to work out the answer	37	66	150	**176**	134
Q 4.8	I often submit symbolic answers even if I think they wrong in order to get feedback	132	144	**155**	89	43
Q 4.13	Using the equation editor encourages me to get the right answer on the first attempt	90	109	**219**	111	34
Q 4.3	I find that working with the equation editor distracts me from working on the answer	94	109	**227**	108	25
Q 4.4	The equation editor helps me reason about symbolic answers	**247**	167	112	29	8
Q 4.6	I think the equation editor slows me down	42	80	**165**	158	118
Q 4.7	The equation editor makes it hard to re-arrange formulæ	56	92	143	**208**	64
Q 4.9	The equation editor limits my freedom when building formulæ	64	122	**196**	107	74
Q 4.11	I find it easy to create a formula in the equation editor	102	126	**169**	122	44
Q 4.12	I avoid questions that I think will require entering complex formulæ	124	**161**	130	95	53

Table 2: Questions on the students' experience working with the equation editor. The numbers in the first column indicate the order in which they were presented to the students. The table shows how we grouped the statements for analysis. The columns on the right summarise the responses.

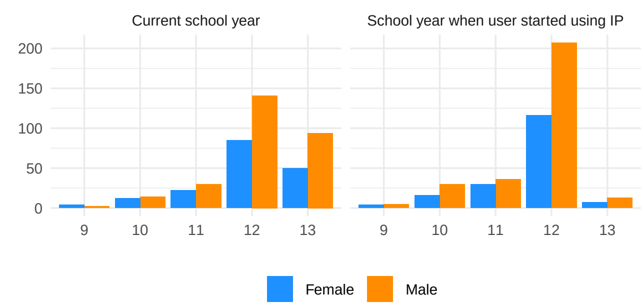


Figure 7: School years in which the students are currently (left) and started using Isaac Physics (right).

Students' experience with the equation editor

Of the 563 valid responses we analysed, only about one in five included comments that we deemed relevant to our analysis. Many of these express negative opinions on the equation editor and on symbolic questions in general. The positive outcome is that most of these negative comments seem to come from frustration with the system rather than prejudice against symbolic work. Respondents usually commented on practical difficulties with the equation editor, in particular when using smaller screens, such as phones and small tablets. The equation editor was not designed to work on screens smaller than a screen size of 1024×768 points.[4] In fact, the application logs reveal that the most common screen size for using the equation editor is around 1366×768 points, which corresponds to a large tablet device, or to a small laptop.

[4]In high resolution displays, point is used as a visual equivalent of pixel in standard resolution displays. In displays using the traditional convention of 72 or 96 pixels-per-inch, one point was equivalent to one physical display pixel. In high resolution displays, one point can comprise several physical pixels. This is useful so that we can refer to the same (nominal) screen size in points while allowing the resolution, and therefore the pixel count, to increase. For example, a first generation Retina iPad has a screen size of 1024×768 points, equivalent to a non-Retina iPad with 1024×768 pixels, despite in fact having 2048×1536 physical pixels.

Table 2 summarises the responses to the thirteen statements in the questionnaire. These reveal that 76.1% of the respondents tend to avoid symbolic questions (Q 4.1), and 76.0% do not particularly enjoy using the equation editor (Q 4.10). During the period under assessment, Inequality was difficult to use with larger, or more complex formulæ, like the example shown in Figure 9a. Students do not seem to avoid questions that they think will require complex formulæ, as shown by statement Q 4.12, although one could argue that the complexity of the formula required may not always be foreseeable.

The second group of statements in Table 2 – Q 4.2, 4.5, 4.8, and 4.13 – suggests that students are slightly more likely to prefer working out symbolic solutions using the equation editor rather than using pen and paper, although the effect is small. Most importantly, students prefer to avoid submitting a large number of wrong answers, despite the potential reward of obtaining custom feedback. However, it does not appear that having to use the equation editor encourages them to submit fewer attempts either. The fact that, according to application logs, they submit fewer attempts in the symbolic format is perhaps due to the fact that the format conversion ends up imposing a lighter workload on the students. In fact, many of the converted questions were originally intended to be symbolic but, since the on-line platform did not support symbolic answers at the time, they were modified to include several additional steps after working out a symbolic solution in order to require a numeric answer that could be checked.

Cross-referencing the answers to statements Q 4.2 and Q 4.5 with the application logs related to the 343 questions converted to symbolic format suggests students are more likely to require fewer attempts where they disagree with statement Q 4.2 – i.e. they would rather not work on paper first and enter answers online later – and agree with Q 4.5 – i.e., they prefer to use the equation editor as part of their problem-solving workflow

(Figure 8). In both cases, the proportion of questions that required fewer attempts after conversion was about 2:1. Overall, one could speculate that, by incorporating the equation editor in their workflow, the students are encouraged to submit fewer attempts before getting the correct answer. Even when discarding errors due to significant figures from the attempts on numeric questions, the ratio of questions that required fewer attempts after conversion was still around 2:1 in both the case of disagreeing with Q 4.2 and agreeing with Q 4.5.

Interestingly, 70.5% of the respondents do not find the equation editor distracting from working out the answers they need, and whilst more than half of the respondents do not think that this system helps them to reason about symbolic formulæ they do not think that it slows them down.

In general, our students regard the editor could be quirky and awkward. Some respondents suggested they would like a choice between using the graphical editor and some form of text-based entry.

DISCUSSION

The evaluation presented in this paper shows that, although Inequality has some usability issues, it provides a functional and flexible way for working with symbolic maths on the web.

With respect to **student performance (RQ 1)**, we considered the average number of attempts required to correctly answer a selection of questions that we converted from their old numeric and multiple-choice formats to the new symbolic format. We have seen that at least 60% of these questions required fewer attempts in symbolic format, but we cannot determine whether this is due to the change in format, a reduction in the number of steps to arrive to an answer, or whether the students change their solution strategies, for example, by working more or less on paper. A reduction of the average number of attempts is encouraging, but we have also seen that a small but considerable number of questions required many more attempts after conversion. We could not determine a definitive reason for this other than the fact that the algebra involved may be inherently hard, and students may have found shortcuts to answer these questions in the old format. We plan to investigate these questions further in a follow-up study.

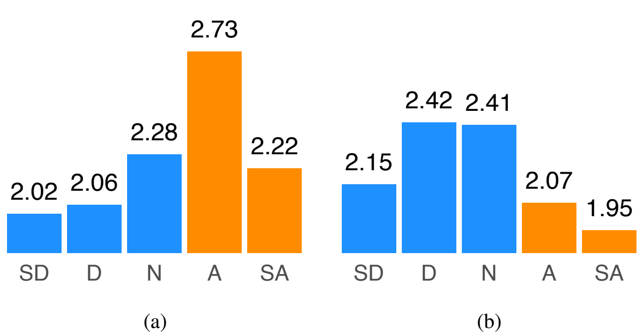


Figure 8: Average number of attempts to provide a correct answer compared with statements Q 4.2 and Q 4.5.

With respect to **how students use Inequality (RQ 2)**, we found that they enjoy considerable freedom in the way they construct their answers and in the way their answers are formulated. This is expected since we designed the system to avoid enforcing one particular working style. We also analysed whether students preferred working out their answers on paper first or using Inequality, and we found a relatively even split, with a very slight preference towards using Inequality. We thought that students who worked on paper first would require fewer attempts to answer questions correctly, but the data rejected this hypothesis ($p = 0.0109$). We cannot currently explain this behaviour and further research, including an in-person study, is required.

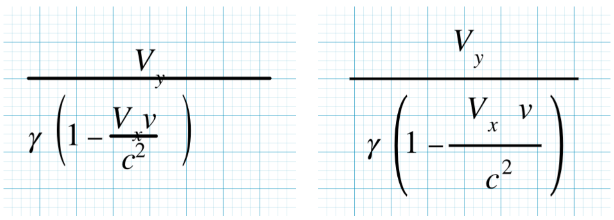
With respect to **student experience (RQ 3)**, we found the overall attitude towards the equation editor to be negative, mostly due to technical difficulties in using it. We have taken the student feedback into consideration for the next iteration. However, although some students and teachers are avoiding symbolic questions, the application logs we collect suggest that many of our users are undeterred, and consistently engage with symbolic questions. Furthermore, a sizeable proportion of respondents stated that they use the equation editor as part of their work flow when answering questions, suggesting that they manipulate formula on the platform rather than on paper. This is encouraging, but has to be put into perspective with the usability issues reported. From the data we collected in this study, we could not determine whether working on paper or on the online editor makes a difference in mathematical reasoning, and this should be the subject of an in-person study.

Improvements to the system

In response to the criticism emerging from the survey, we worked on two fronts: improvements to the graphical editor, and development of a text-based entry system that is tightly coupled with the graphical editor.

Improvements to the graphical editor focused on two aspects: the relative positioning of the symbols, and the docking logic. The docking logic of the version studied was overly complicated and made it difficult to dock a symbol to the intended docking point. We re-implemented it from scratch and preliminary tests indicate that it is now much easier to correctly select the desired docking point while moving a symbol around. The relative positioning of the symbols when docked to each other, as shown in Figure 9a, was problematic. In response, we redesigned the layout model taking into account the feedback and constraints, and the quality of the mathematical layout has now much improved compared to the version described in this paper. Figure 9b provides one example to demonstrate the layout improvements. The new version was released in June 2018 and we are collecting data for future analysis.

In response to feedback we have developed a text-entry system that can be used to type simple expressions that may otherwise take longer to build in the graphical editor, as well as by more advanced users that are used to a linear mathematical syntax. We developed a JavaScript-based parser with a grammar that closely resembles the one used by our checking system (which is built on SymPy). The parser turns mathematical expressions such as $I = V/R$ into the abstract syntax tree representation used internally by the graphical editor. This allows us to render any text entered by the user into the answer box (bottom of Figure 1) as the user types, thus providing immediate feedback. In addition it allows users to move seamlessly between text-based and graphical entry as well as providing a translation path to the SymPy code suitable for evaluation by the back-end checker used to evaluate submitted answers.



(a) (b)

Figure 9: Comparison of the same formula built in the old (a) and new (b) versions of the equation editor.

CONCLUSION

One of the goals of Isaac Physics is to help students develop their problem solving skills in physics and mathematics. Symbolic manipulation is a fundamental skill in these and other subjects. We built a graphical, web-based, symbolic entry system for students to use in our on-line platform. We designed it to have a gentle learning curve, leveraging the students' pre-existing experience of working with mathematical layout on paper, thus enabling students lacking knowledge of more specialist tools to work on symbolic questions on our platform.

Inequality is usable on larger touch-based devices as well as all mouse-based devices; usability remains challenging on small screens. Inequality reduces the effectiveness of guessing and affords considerable variety in both the construction and formulation of the answers. Fixed docking points offer a more robust input method when compared to a previous solution which supported free placement on a canvas.

Inequality now reaches about two thousand users per month, or about 10% of our active user base. Uptake is growing each year as new schools and new students use the platform.

A fully featured demonstration of our system is available at https://isaacphysics.org/equality.

REFERENCES

1. Abraham Arcavi. 1994. Symbol Sense: Informal Sense-Making in Formal Mathematics. *For the Learning of Mathematics* 14, 3 (1994), 24–35. http://www.jstor.org/stable/40248121
2. S. Cummins, I. Davies, A. Rice, and A. R. Beresford. 2015. Equality: A Tool for Free-Form Equation Editing. In *2015 IEEE 15th International Conference on Advanced Learning Technologies*. 270–274. DOI: http://dx.doi.org/10.1109/ICALT.2015.34
3. Robert Goldstone and David Landy. 2009. How much of symbolic manipulation is just symbol pushing?. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 31.
4. Joseph J. LaViola Jr. 2007. Advances in Mathematical Sketching: Moving Toward the Paradigm's Full Potential. *IEEE Computer Graphics and Applications* 27, 1 (Jan. 2007), 38–48. DOI: http://dx.doi.org/10.1109/MCG.2007.2
5. Jakob Leitner, Christian Rendl, Florian Perteneder, Adam Gokcezade, Thomas Seifried, Michael Haller, Robert Zeleznik, and Andrew Bragdon. 2010. NiCE Formula Editor. In *ACM SIGGRAPH 2010 Talks (SIGGRAPH '10)*. ACM, New York, NY, USA, 55:1–55:1. DOI: http://dx.doi.org/10.1145/1837026.1837098
6. Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondrej Certik, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Stepan Roucka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. 2017. SymPy: Symbolic Computing in Python. *PeerJ Computer Science* 3 (Jan. 2017), e103. DOI: http://dx.doi.org/10.7717/peerj-cs.103
7. Joel T. Nadler, Rebecca Weston, and Elora C. Voyles. 2015. Stuck in the Middle: The Use and Interpretation of Mid-Points in Items on Questionnaires. *The Journal of General Psychology* 142, 2 (April 2015), 71–89. DOI: http://dx.doi.org/10.1080/00221309.2014.994590
8. Steve Smithies, Kevin Novins, and James Arvo. 2001. Equation Entry and Editing via Handwriting and Gesture Recognition. *Behaviour & Information Technology* 20, 1 (Jan. 2001), 53–67. DOI: http://dx.doi.org/10.1080/01449290010020657
9. Ba-Quy Vuong, Yulan He, and Siu Cheung Hui. 2010. Towards a Web-Based Progressive Handwriting Recognition Environment for Mathematical Problem Solving. *Expert Systems with Applications* 37, 1 (2010), 886–893.