A demonstration of Numbas, an assessment system for mathematical disciplines

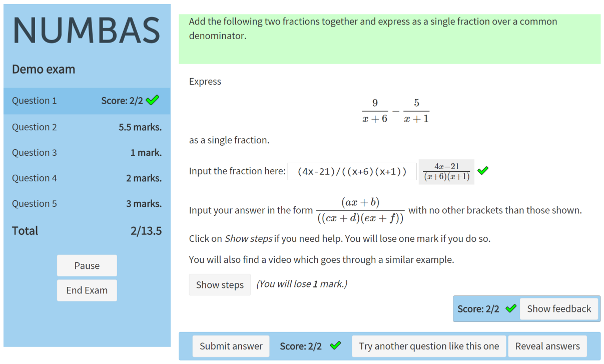
Christian Perfect

Newcastle University, Newcastle upon Tyne, UK
christian.perfect@ncl.ac.uk

Abstract. *Numbas* is a free and open-source e-assessment system developed at Newcastle University. This demonstration highlights the key features of *Numbas* and the design considerations for an e-assessment tool focused on mathematics.

Keywords: e-assessment, mathematics, e-learning, computer assisted assessment, computer algebra

1 About Numbas



NUMBAS

Demo exam

Question 1	Score: 2/2 ✓
Question 2	5.5 marks.
Question 3	1 mark.
Question 4	2 marks.
Question 5	3 marks.
Total	2/13.5

Add the following two fractions together and express as a single fraction over a common denominator.

Express

$$\frac{9}{x+6} - \frac{5}{x+1}$$

as a single fraction.

Input the fraction here: (4x-21)/((x+6)(x+1)) $\frac{4x-21}{(x+6)(x+1)}$ ✓

Input your answer in the form $\frac{(ax+b)}{((cx+d)(ex+f))}$ with no other brackets than those shown.

Click on *Show steps* if you need help. You will lose one mark if you do so.

You will also find a video which goes through a similar example.

(You will lose 1 mark.)

Score: 2/2 ✓

Score: 2/2 ✓

Fig. 1. A typical Numbas question.

Numbas[9] is a free and open-source e-assessment system developed at Newcastle University's School of Mathematics and Statistics. It follows the CALM[1] model: a test consists of several questions, each of which consisting of one or more parts which assess an individual answer entered by the student. Development of *Numbas* began with the aim of replacing the commercial systems already in use at Newcastle with a more modern and open-source alternative. Today, *Numbas* is also used at over 30 institutions across the UK and around the world.

Questions are randomised according to a set of variables, defined by arbitrary mathematical expressions. A question consists of a statement, one or more *parts*, and an advice section. The advice section typically contains a fully worked solution to the question and is revealed once the student has finished the question.

A part can provide *steps*, revealed if the student clicks a button, which break the part into more parts to guide the student through the question, and allow the student to receive partial marks for an incomplete answer.

Numbas supports several question types, including most of the types you'd expect - number entry, multiple choice/multiple response, and text entry. Marking is performed on the client, so feedback is immediate.

A *Numbas* test is a standalone package containing an HTML page, with all the exam logic implemented in JavaScript and running on the client side. Because a *Numbas* test is just a web page, questions can include any of the rich content found on the web, including videos, interactive graphics, and other elements.

Questions are created using a web-based graphical editor.

2 Formative vs summative assessment

We are strongly of the opinion that computer-based assessment of degree-level mathematics is most suitable in a formative mode rather than summative. The kinds of questions which can be marked automatically by a computer are mainly restricted to mechanical, turn-the-handle techniques.

More creative, problem-solving tasks are much harder to automate and repeat.

Numbas tests are ideally suited for use as an inexhaustible bank of practice questions to practise basic techniques. Students can receive immediate, high-quality feedback on their answers, and each question has a "Try another question like this one" button which generates a new randomised question of the same type.

3 Suitable topics for assessment with Numbas

Numbas is used at Newcastle to assess undergraduate-level mathematics, both in the mathematics degree and in other service courses. We believe computer-assisted assessment of mathematics is most useful around the transition to university, when we need to ensure that students are proficient in the basic skills of algebra and calculus. Statistical modelling tasks of the sort undertaken in first-year courses in STEM and other numerate disciplines are also particularly adaptable to randomisation and automated marking.

While a few final-year modules use *Numbas* to provide an opportunity for practice at new but mainly algorithmic tasks, such as cryptography and coding theory, there are very few topics at that stage of the degree that are suitable for purely computer-based assessment.

Conversely, *Numbas* as it exists today has limited value in primary education, which requires a simpler interface and presentation is much more important than sophistication in generation and marking.

4 Current use

Every module in Stage 1 and Stage 2 of the mathematics degree at Newcastle University includes at least two *Numbas* tests. Students can access each test in "practice mode" for one week, during which they can make as many attempts as they like and get full feedback, followed by "assessed mode" the next week, for which they can make only one attempt[4].

A publicly-accessible instance of the *Numbas* editor is available at **math**centre[14]. Its question bank contains over 4,000 questions written by lecturers and teachers all over the world, including over 700 questions created at Newcastle and released under the Creative Commons Attribution licence. *Numbas* has been translated into Norwegian Bokmål, Spanish and Dutch.

5 Client-side design

Numbas runs entirely on the client. This means it's very fast – no calls to external computer algebra systems to generate questions or mark answers – and can be distributed to students as a standalone webpage, or integrated into a VLE without any setup.

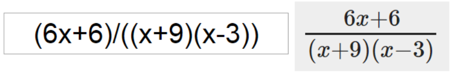
While many other mathematics assessment systems have a dependency on server-side computer algebra systems, *Numbas* contains a computer algebra system written entirely in JavaScript, so expressions can be interpreted and evaluated entirely on the client, with no delay while a network request is made. This system is quite fast – a question can generate random data sets of several thousand elements and compute associated statistics in a fraction of a second.

By their very nature client-side designs are not as secure as server-side implementations and an enterprising student could find ways to cheat without suitable safeguards. This is only an issue for summative high-stakes examinations and can be easily circumvented – we recommend that all testing takes place in an invigilated environment. For less-important summative assessment, such as the fortnightly in-course assessment required in all maths modules at Newcastle, we trade some convenience for security and provide the students with a standalone app which is the only means of accessing the site storing their *Numbas* tests. This way, students are prevented from breaking the software to cheat, but can still copy off each other, look up answers on the internet, or do any of the other things badly-behaved students also do for paper-based homework.

6 Input of mathematical expressions

The syntax used by the student to enter a mathematical expression as their answer must be simple, memorable and unambiguous (discussed by Sangwin and Ramsden in [13]). *Numbas* uses a syntax similar to that which a student would type on a calculator or computer algebra system such as Maxima[8]. Importantly, the student's expression is rendered in mathematical notation beside the input field and updated immediately as the student types, so they can be sure that the system has interpreted their expression as they intended.

Students by and large have no problem with this syntax[10]. One error which occurs quite often is the omission of the multiplication symbol between variable names, or a variable name and an opening bracket. To avoid this, the question author can provide a list of expected variable names in the student's answer; the student is shown a warning if they use any variable name not in the list.



(6x+6)/((x+9)(x-3))

$\frac{6x+6}{(x+9)(x-3)}$

Fig. 2. The student's answer is rendered immediately next to the input field.

7 Usability

Usability is an important aspect of any software, and often neglected in mathematical assessment systems. Good user experience is an important goal of *Numbas*, both for the student and question author.

7.1 Student experience

The interface is clean, and accessible by e.g. screen readers. Because everything runs on the client, feedback is immediate.

Over several years of use and based on student feedback, the interface has been made more intuitive. Because *Numbas* runs through the web, it can be accessed from any device, including mobile phones and tablets.

7.2 Author experience

The barrier to entry for question authors is low - using the graphical editor, one can create a simple question with no knowledge of programming or LaTeX.

As the author gains experience, they can gradually add mathematical notation written in LaTeX (we believe this is the most straightforward way of entering mathematical notation on a computer) and more complex variable definitions.

The rich-text content editor provides instant renderings of LaTeX notation, cutting out the write-compile-preview loop.

Questions can be run immediately inside the editor, so the author can see the question as a student would, and find errors in rendering or marking.

Numbas allows the creation of *template questions* which can easily be adapted to new subjects without changing the underlying maths. For example, a question on linear regression can provide several variables representing parts of the statement text and parameters for the underlying statistical distribution from which random values are generated. An author can edit these variables as simple text strings or numbers, respectively, without modifying the more complex parts of the question.

8 Complexity of questions

Standard questions encountered in a high school or early undergraduate maths course are easily implemented in *Numbas*. However, questions covering more advanced topics or with more sophisticated designs are of course more complex, particularly when the question is randomised: first of all, defining a set of variables which generate a question with sensible setup and solution (or even any solution at all) is not always trivial, and often effectively boils down to a programming exercise; secondly, marking a mathematical expression might not involve just deciding if the student's answer matches a given model answer exactly, instead asking if the student's answer has certain mathematical properties, or is given in a particular form.

Numbas has been designed with the aim of minimising this complexity and providing abstractions of common patterns to avoid repetition.

8.1 Variable generation

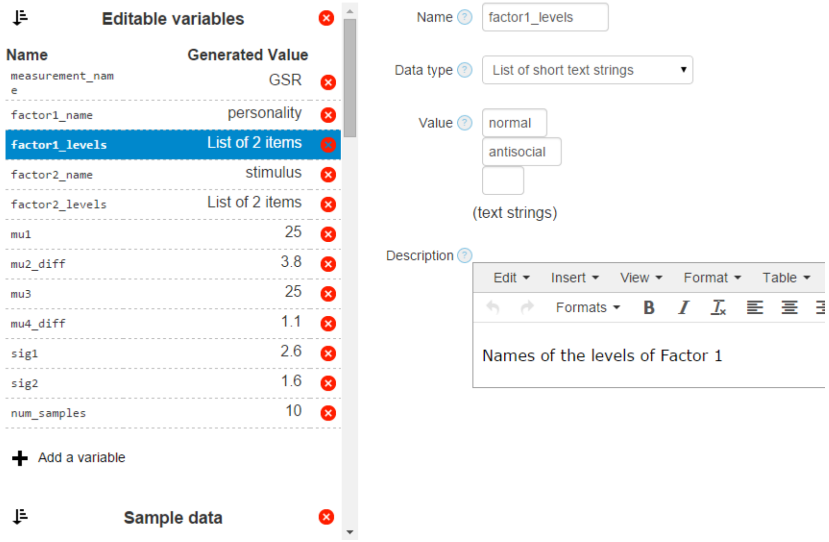
Question variables in *Numbas* are generated declaratively. Generating random numbers, text, or other data is very simple. Other variables can then do further computations with those values.

The variable definition interface allows you to work interactively, with variable values generated immediately with each change to the definitions. The variable testing tool allows you to generate large numbers of sets of variables and compute statistics about their values. This helps the author get an idea of how often particular cases occur. You can also specify a condition on the variables which must be satisfied, to avoid pathological cases.

8.2 Marking

The algorithm used by *Numbas* to mark mathematical expressions is conceptually simple, but good enough to cover a wide range of question types - the student's answer and the model answer are evaluated over a randomly selected set of variable values, and compared. If the student's answer matches the model at every point, it's marked as correct.

Often, questions assess a mathematical expression not just for the value it represents, but for its form – for example, when asking a student to factorise a quadratic, it's important that the answer is of the form $(x + a)(x + b)$ and not $x^2 + ax + b$. *Numbas* has a sophisticated pattern-matching system which can be used to make assertions about the form of the student's answer and tailor feedback accordingly.



Editable variables

Name	Generated Value
measurement_name	GSR
factor1_name	personality
factor1_levels	List of 2 items
factor2_name	stimulus
factor2_levels	List of 2 items
mu1	25
mu2_diff	3.8
mu3	25
mu4_diff	1.1
sig1	2.6
sig2	1.6
num_samples	10

Add a variable

Sample data

Name: factor1_levels

Data type: List of short text strings

Value: normal
antisocial

(text strings)

Description: Names of the levels of Factor 1

Fig. 3. The variable definition interface.

An abstraction for marking schemes that doesn't involve programming would be very useful. STACK[12] uses an acyclic directed graph with tests at each node, which we find both cumbersome and limited. For now, *Numbas* allows authors to write custom JavaScript marking algorithms which can manipulate the student's answer to decide how it should be marked.

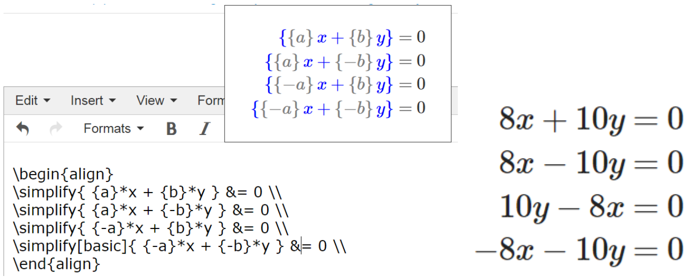
9 Rendering mathematical notation

Numbas uses the MathJax[7] library to render mathematical notation. Question authors can write mathematics using LaTeX syntax, with question variables substituted in.

The substitution of random variables into mathematical expression raises a significant problem – how to present the resulting expressions in a natural way. *Numbas* has a sophisticated and customisable system of simplification rules based on a unique pattern-matching system, which cleans up mathematical expressions before they are rendered.

10 Customisation

A large driver towards the creation of *Numbas* was the lack of customisability in previous systems. *Numbas* has separated the display of exams from the underlying logic. This means authors can create new themes to change the look of their tests. This can be cosmetic changes, such as a different logo or changes to



$$\{\{a\}\,x + \{b\}\,y\} = 0$$
$$\{\{a\}\,x + \{-b\}\,y\} = 0$$
$$\{\{-a\}\,x + \{b\}\,y\} = 0$$
$$\{\{-a\}\,x + \{-b\}\,y\} = 0$$

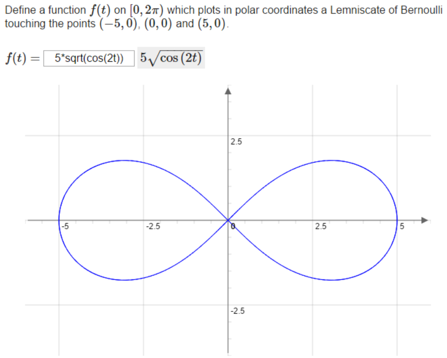
```
\begin{align}
\simplify{ {a}*x + {b}*y } &= 0 \\
\simplify{ {a}*x + {-b}*y } &= 0 \\
\simplify{ {-a}*x + {b}*y } &= 0 \\
\simplify[basic]{ {-a}*x + {-b}*y } &= 0 \\
\end{align}
```

$$\begin{aligned} 8x + 10y &= 0 \\ 8x - 10y &= 0 \\ 10y - 8x &= 0 \\ -8x - 10y &= 0 \end{aligned}$$

Fig. 4. Preview of LaTeX in the question editor, and the result as shown to the student.

the stylesheet, up to total reimaginings of how the test is delivered, as seen in Peter Rowlett's printed worksheet theme [11].

Numbas also has an extensions interface. Extensions can provide new data types, functions and resources to questions. Examples include a library of statistical functions, data types representing permutations and linear codes, and functions to draw interactive diagrams with JSXGraph[5] [6].


Define a function $f(t)$ on $[0, 2\pi)$ which plots in polar coordinates a Lemniscate of Bernoulli touching the points $(-5, 0)$, $(0, 0)$ and $(5, 0)$.

$f(t) =$ 5*sqrt(cos(2t)) $5\sqrt{\cos(2t)}$

Fig. 5. Dynamic rendering of the graph of a student's answer using JSXGraph.

11 Integration with a VLE

Numbas uses the SCORM 2004[2] standard to integrate with compliant VLEs, such as Moodle[3] and Blackboard. This way, the VLE handles session management and data storage, leaving *Numbas* as a standalone assessment tool. *Numbas* tests can be accessed just like any other web page outside a VLE, but without score tracking.

References

1. Beevers, C., Cherry, B., Clark, D., Foster, M., McGuire, G., Renshaw, J.: The {CALM} before the storm! {CAL} in university mathematics. Computers & Education 12(1), 43 – 47 (1988), http://www.sciencedirect.com/science/article/pii/036013158890053X
2. Dodds, P., Thropp, S.: Scorm® 2004 3rd edition (2006), http://www.adlnet.gov/scorm/scorm-2004-3rd/
3. Dougiamas, M., Taylor, P.: Moodle: Using learning communities to create an open source course management system. In: World conference on educational multimedia, hypermedia and telecommunications. vol. 2003, pp. 171–178 (2003)
4. Foster, B., Perfect, C., Youd, A.: A completely client-side approach to e-assessment and e-learning of mathematics and statistics. International Journal of e-Assessment 2(2) (2012)
5. Gerhäuser, M., Miller, C., Valentin, B., Wassermann, A., Wilfahrt, P.: Jsxgraph: Dynamic mathematics running on (nearly) every device. Electronic Journal of Mathematics and Technology 5 (2011)
6. JSXGraph: (2015), http://jsxgraph.uni-bayreuth.de/wp/
7. MathJax: (2015), http://www.mathjax.org
8. Maxima: (2014), http://maxima.sourceforge.net/
9. Numbas: (2015), http://www.ncl.ac.uk/maths/numbas
10. Parker, N.G.: An analysis of computer-based assessment in the school of mathematics and statistics (2013), http://blogs.mas.ncl.ac.uk/numbas/2013/06/an-analysis-of-computer-based-assessment-in-the-school-of-mathematics-and-statistics/, [accessed on 2015/03/05]
11. Rowlett, P.: Development and evaluation of a partially-automated approach to the assessment of undergraduate mathematics. In: Pope, S. (ed.) Proceedings of the 8th British Congress of Mathematics Education. pp. 295–302 (2014), http://bsrlm.org.uk/BCME8/BCME8-38.pdf
12. Sangwin, C.J.: Assessing elementary algebra with stack. International Journal of Mathematical Education in Science and Technology 38(8), 987–1002 (2007), http://dx.doi.org/10.1080/00207390601002906
13. Sangwin, C., Ramsden, P.: Linear syntax for communicating elementary mathematics. Journal of Symbolic Computation 42(9), 920 – 934 (2007), http://www.sciencedirect.com/science/article/pii/S0747717107000752
14. **math**centre: Publicly accessible numbas editor (2015), http://numbas.mathcentre.ac.uk