

CASE STUDY

Assessment of computing in the mathematics curriculum using Numbas

Chris Graham, School of Mathematics, Statistics & Physics, Newcastle University, Newcastle Upon Tyne, UK. Email: christopher.graham@ncl.ac.uk.

Abstract

This case study discusses the use of the Numbas e-assessment system to assess computing skills across several modules in a mathematics undergraduate degree programme. The modules include basic computing, quantitative analysis of data, and numerical methods. Several approaches are discussed which fit with the teaching of SPSS, R and MATLAB, including randomised data files and questions which can replicate, and therefore mark, calculations made with R data frames and numerical algorithms, such as root finding and curve fitting. In each case, Numbas offers the opportunity to automatically mark and offer immediate feedback to the student. The application of questions inside a computing module is discussed, with a positive response from students to both practice material and hybrid tests, which include some automatic marking alongside submission of the students' code for manual review. There is clear rationale for using an e-assessment system which is already familiar to students, with features such as adaptive marking and the scaffolding of questions, however limitations to the use of Numbas for this purpose are also discussed.

Keywords: computing, e-assessment, Numbas, numerical, quantitative.

1. Background

1.1. Computing in the curriculum

Modules dedicated to computer programming have been a compulsory component of the single-honours mathematics degree programme at Newcastle University since 2015. The addition of computing is in common with many other mathematics departments in the United Kingdom (Sangwin, 2017), motivated by the increasing relevance of computers in mathematical teaching and research, and in the future career prospects of undergraduate students. The curriculum has included the teaching of two programming languages, R and MATLAB (the latter being phased-out, to be replaced entirely with Python by 2022), covering basic programming, statistical analysis and numerical methods.

Most of the computing teaching follows the same format, with three hours of contact time each week. A one-hour lecture is used to introduce the theory behind the week's content and to work through examples. In a two-hour practical that follows, students work through a "handout" of material, which may be either a physical handout or a digital version.

1.2. The Numbas e-assessment tool

Many mathematics departments take advantage of specialist mathematical e-assessment software, such as DEWIS (Gwynllyw and Henderson, 2009), Numbas (Foster, et al, 2012) and STACK (Sangwin, 2015), to automatically mark students' work and offer immediate feedback. At Newcastle, where the software Numbas is developed, e-assessment is embedded into the mathematics degree programme across twenty-four modules.

Following the CALM model (Beevers, et al, 1988), Numbas tests can consist of several questions, each of which may contain one or more parts which assess an answer. A student answer may be a

numerical value, mathematical expression (as in figure 1), or one of several other types supported, such as matrix input.

An example Numbas test

The screenshot shows a Numbas test interface. On the left is a sidebar with the 'NUMBAS' logo and a list of questions. Question 1 is marked as 'Answered' with a score of 3/3. Below the sidebar is a red 'End Exam' button. The main area contains the question: 'Differentiate the following function $f(x)$ using the chain rule.' The function is $f(x) = (2x^7 + 4)^9$. The student's answer is $\frac{df}{dx} = 126 * x^6 * (2 * x^7 + 4)^8$, which is highlighted in green. To the right of the input is a preview of the answer as interpreted by the software: $126x^6(2x^7 + 4)^8$, also with a green checkmark. Below the answer is a 'Show steps' button with the text '(Your score will not be affected.)'. Further down, there is feedback: 'Your answer is numerically correct. You were awarded 3 marks. You scored 3 marks for this part.' At the bottom right, it says 'Score: 3/3' with a green checkmark and 'Answered'. At the bottom of the main area are buttons for 'Submit answer', 'Score: 3/3' with a green checkmark, 'Try another question like this one', and 'Reveal answers'.

Figure 2. Numbas can mark and give feedback on mathematical expressions. To the right of the student input (highlighted green to indicate a correct answer) is a preview of the student's answer as interpreted by the software, which a student can check before submission. The settings of this test are such that the student receives immediate feedback, and has the opportunity to "Try another question like this one", which re-randomises the variables in the question, or to "Reveal answers" to see a full solution. This question has the option to "Show steps" to see a reminder of the chain rule, which is offered without a reduction in the available marks in this case.

Some of the key features of Numbas were inherited from the CALM model, including advice for each question and the facility to scaffold a question into smaller steps. The variables in a Numbas question can be randomised to allow for many versions, each one with a corresponding full solution in the advice. This randomisation, alongside the immediate marking and feedback, makes the tool particularly powerful for formative use. One common complaint by students towards online assessment is its all-or-nothing marking. In response, a feature of adaptive marking was added, whereby the correct answer to a question part could be re-calculated, based on the incorrect student input to a previous part. This digitalises some of the discretion given by a human marker when awarding "error carried forward" marks. These features were important in the choice to use Numbas in computing modules.

Numbas is an open-source tool and is made available to users outside of Newcastle through a public editor hosted by MathCentre (Newcastle University, 2020). Here, teachers can create and share content, with around 8,000 items available for reuse under an open access licence, including many of the examples in this case study.

2. Motivation for using e-assessment in computing teaching

Students working through handouts in computer practicals do so at their own pace, running and investigating the commands given to them. One of the key features of the handouts is formative questions, which help to reinforce the ideas covered, or to investigate a little bit further than is covered in the given material (see figure 2). These questions are typically located at the end of a section, and students work through them as they reach them. Students are encouraged to have a go at the questions before asking for help, including using documentation and online resources. A student may request help from a postgraduate assistant, but many will suffer in silence or skip past questions if they get stuck. The lecturer may often present solutions to the questions at appropriate points during the practical, delivered by display screens and microphone to the entire class.

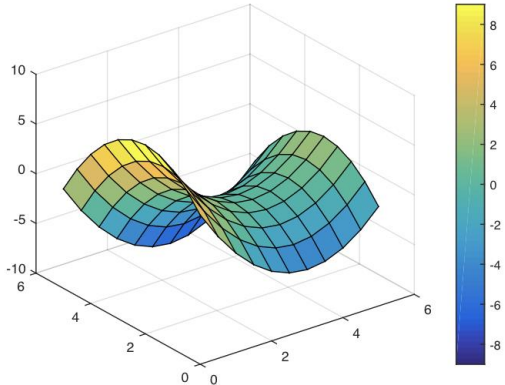
Example content from a computing handout

$$z = f(x,y) = (x-3)^2 - (y-2)^2;$$

We first define z in terms of our mesh grid coordinates X and Y (we'll use big Z for consistency, since we're now using big X and Y), and then plot using the function `surf()`:

```
Z = (X-3).^2 - (Y-2).^2;  
surf(X,Y,Z)  
colorbar
```

The command `colorbar` displays a helpful colour bar as a visual aid to the values of z



Exercise 4.6 Plot the function

$$z = \sqrt{x^2 + y^2}, \quad 0 \leq x \leq 10, \quad 0 \leq y \leq 10$$

Add axis labels, a title and colorbar to your plot.

Figure 2. An excerpt from an early version of the handouts for stage one MATLAB computing. Exercises embedded inside the handout encourage students to consolidate their learning, but when and how to provide solutions poses a problem.

Students work through computing handouts at very different rates, and the question of when to offer solutions to these questions is a difficult one. In a survey of students conducted at the end of the module in the 2019/20 academic year, 48% of 81 respondents agreed that the solutions presented by the lecturer were at the right time to help them with the handout questions, however 21% responded that these interactions were actually distracting. Students were split evenly over whether the solutions to the questions should be released immediately, or at the end of the practical.

E-assessment offers a potential solution: marking and feedback can be offered as and when a student is ready to attempt a practice question. Several tools already exist for marking computer code, including the Moodle plugin Coderunner (Lobb and Harlow, 2016) and the INGIInious assessment platform (Derval, et al, 2015), which run students` code and carry out unit tests to generate marks and feedback. However, we are often interested in the answers that students can interpret from their code, rather than the code itself, and this is exactly the sort of input that Numbas handles so well: numeric values, for example from a statistical analysis, or; mathematical expressions, such as a best fit polynomial for some data. We might also be interested in breaking down questions into smaller steps or using adaptive marking to account for errors in multi-part questions, both of which are accommodated by Numbas. Since the tool is already familiar to our students, it is a logical choice to adapt it for computing questions.

3. Using Numbas to assess computing skills

Each of the examples of using Numbas to assess computing skills uses a similar flow:

- The student is presented with a question generated by Numbas, which may include some randomisation;
- The student tackles the problem on the external software, to obtain an answer;
- Returning to Numbas, the student inputs their answer. Behind the scenes, Numbas can make the same calculations as the external software, and therefore can mark the answer and offer feedback.

There is nothing particularly pioneering about this process; if you were to replace “external software” with “calculator” or “pen and paper” then this is no more than the regular workflow of using an e-assessment system. The key parts are the generation of the data and subsequent parallel analysis that Numbas can make to mark the student’s work. Note that, using these methods, Numbas does not directly mark the student’s code. This does not mean that it cannot offer advice on the code required to obtain the answer, but it relies on these skills being assessed by other means if that is desired. Section 4 describes this hybrid approach to marking for some assessments.

The following sections describe some of the different approaches used to develop questions:

3.1. Scenario-based questions

Assessing a student’s quantitative analysis of data inside external software is made possible by the flexibility of data types available to Numbas to store a copy of the data, including lists and arrays, and the JSON (JavaScript Object Notation) format, which is a lightweight, human-readable format for storing data.

An early example of using Numbas with external software was for a module in quantitative methods with SPSS. The question in figures 3 and 4 assesses the student’s ability to import a file into SPSS, and to perform a two-sample t-test and Levene’s test of equal variances. The student is presented with a random binary SPSS file, of the sort that they are familiar with in other parts of their teaching materials. In this case, Numbas does not directly handle the data. Rather, for each scenario (file

presented to the student), the correct answers are computed in advance using SPSS, and stored in an array of scenarios in the JSON format, alongside the path to the corresponding data file (figure 3).

Numbas Editor interface for a scenario-based randomised SPSS question

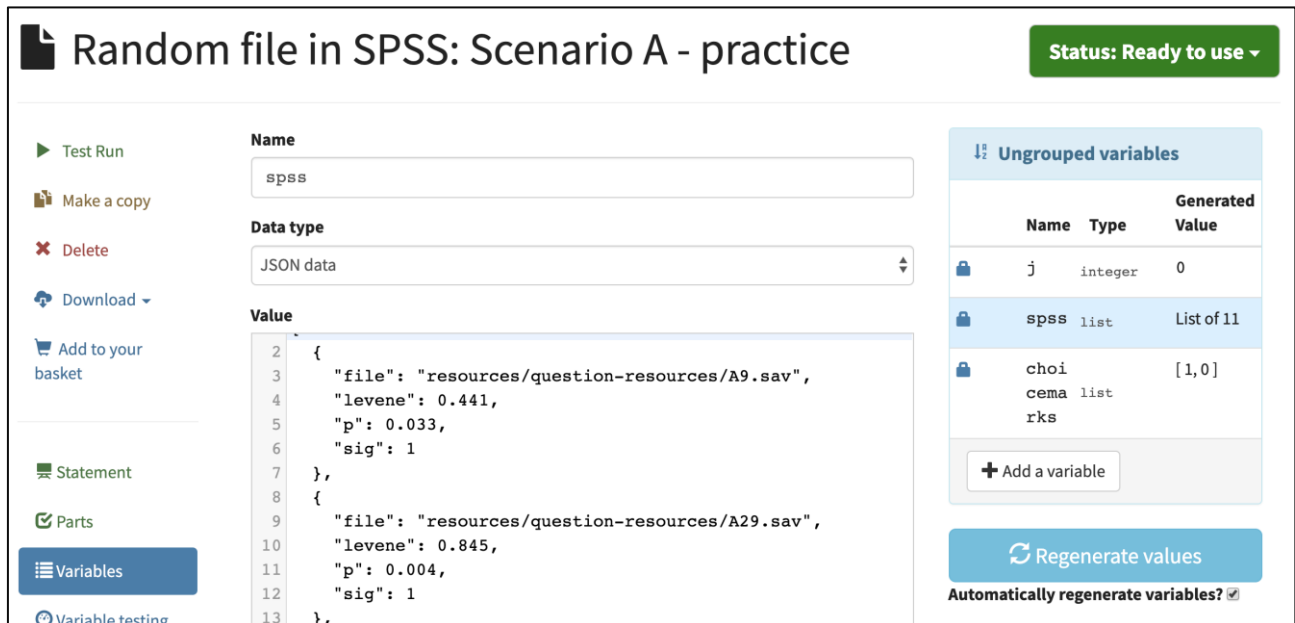


Figure 3. The variable definition in the Numbas Editor for the question illustrated in figure 4. The JSON format allows the structure of scenarios to be laid out, with each scenario containing the path to the relevant file and the correct answers associated with it.

When the question is loaded (figure 4), one of the scenarios is chosen at random and the student receives a file to download. Once they have performed the analysis in SPSS, they return to Numbas to input their answers, and these are compared to the corresponding values for that scenario.

The randomisation of this question is limited by the number of scenarios prepared. It does, however, demonstrate a model for a question that is flexible to any file type or external software, and relatively straightforward to prepare in Numbas, in that it is not necessary to follow along with the calculation of answers.

3.2. Generating data files

An alternative version of the SPSS question provides a data file in CSV (comma-separated values) format, which is text-based and can be generated on-the-fly by Numbas. The Numbas Statistical Functions extension supports many functions for sampling and statistical analysis, and can be used both to generate the data (for example for a particular distribution) and to compute the correct answers. This approach is used in several questions which ask the student to input a file to R and perform queries on the imported data. A similar approach was taken to generate SPSS data and answers by embedding R code inside a DEWIS question by Weir, et al, 2017.

Numbas randomised SPSS question

An environmental scientist who is interested in the water quality of North East rivers collected samples of water from the river Tyne (river 1) and the river Wear (river 2) and measured the turbidity of each of these samples (in Jackson Turbidity units, or JTUs); turbidity is a measure of water quality related to the "cloudiness" of the water.

The data is stored in this [SPSS file](#) (click on the link to download the file and open it in SPSS).

Carry out a two-sample t-test on these data in SPSS to assess whether there is a statistically significant difference in the water quality of the two rivers.

a)

Enter the p-value for Levene's test of equal variances.

Round your answer to 3 decimal places. ✓

Submit part

✓ Your answer is correct. You were awarded 2 marks.

Figure 4. A randomised SPSS question in which the student is presented with a random file to download. Numbas marks the student's input against the corresponding correct answer.

A randomised question using an R data frame

Run the following commands to load the `tv` data frame.

```
library(ncldata)
data(tv)
```

a)

What is the average rating of the TV show *My Family*?

Round your answer to 1 decimal place. ✓

Submit part

✓ Your answer is correct. You were awarded 1 mark.
You scored 1 mark for this part.

Score: 1/1 ✓
Answered

b)

How many of the TV shows both started and ended in the 1990s?

✓

Submit part

✓ Your answer is correct. You were awarded 1 mark.

Figure 5. In this practice question, which accompanies a handout on R data frames, students are instructed to load the TV data frame. Variables are used to randomise parts of the question, in this case the show and decade.

3.3. R data frames

The material for R teaching at stage one is centred on data frames from the IMDB (Internet Movie Database) film information, which is provided through an R package (Stagg, 2019). The data set contains metadata for the films along with aggregated user rating and is, anecdotally at least, engaging for students, who are familiar with the context and interested in the content. Students learn how to query and subset the data frame, different ways to plot and present the data, and later some basic statistical analysis.

To offer practice material to students, we take advantage of the JSON format to store a copy of the data inside Numbas, in order to randomise and automatically mark questions. Figure 5 illustrates one such question.

3.4. Numerical methods

Numerical methods taught in stage two of the mathematics undergraduate degree include root finding, curve-fitting, numerical integration and solutions to ordinary differential equations. Many of the associated algorithms are well-defined and can be replicated in Numbas, which has the facility to include user-defined functions written in JavaScript.

A question might ask a student to use MATLAB's *polyfit* to fit a polynomial to a set of data points. Numbas can randomise the data itself and then replicate the functionality of the *polyfit* function to mark the student's answer. The student could then be asked to calculate the sum of the squared residuals of the fit. Since Numbas has the student's answers to the first part, it can also use adaptive marking to mark this part, as illustrated in figure 6.

4. Assessment Methodology

This section describes several different use cases for the Numbas questions in the numerical methods section of a stage 2 computing module.

4.1. Formative material

A Numbas practice test was provided for each of the four topics of the numerical methods section in the 2019/20 academic year. These tests have settings such that students receive immediate marking and feedback, can reveal answers and regenerate questions, whilst starting as many attempts as they like. Just over half of the 160 enrolled students took advantage of the tests.

Student feedback on the practice material was very positive, with 80% indicating that the Numbas questions helped them to better understand the course material. Comments included, "the practice material was not only useful for the tests but also stimulating and engaging, which made me very interested in the module".

4.2. Written Assignment

A short assignment forms part of the summative assessment on root finding and curve fitting, for which randomised questions offer some mitigation to the potential problem of solutions being shared. The assignment takes a hybrid approach, randomising each question and automatically marking some of the key results, whilst students are also asked to submit their code and a plot of their results, giving the opportunity to assess the students' coding style and their presentation skills.

Adaptive marking applied to a curve-fitting question

Consider the following data

x	1	2	3	4	5
y	0.24	0.15	0.1	0.08	0.09

a)

Fit a function $f(x) = \frac{1}{ax + b}$ to the data using MATLAB's `polyfit` to find a and b

$a =$ Round your answer to 4 decimal places. ✓

$b =$ Round your answer to 4 decimal places. ✗

value for a
✓ Your answer is correct.

value for b
✗ Your answer is incorrect.

You scored 1 mark for this part.

Score: 1/2 ✓
Answered

b)

Calculate the sum of the squared residuals S for the fit $f(x)$ to this data.

$S =$ Round your answer to 4 decimal places. ✓

(Your score will not be affected.)

Figure 6. A MATLAB curve-fitting question. The student is provided with some random data and asked to fit a curve using MATLAB's `polyfit` function. In this case they need to transform the function into a polynomial with a change of variables before using `polyfit`, so a common error is to fit $f(x) = ax + b$ to the data. With this in mind, part b) uses adaptive marking to award the marks to a student who has gone wrong in part a), but subsequently carried out their calculation of the residuals correctly. Part b) also utilises the steps feature to provide advice on how to make the calculation.

4.3. Class test

The main component of assessment for the numerical methods section of the module is a class test. The class test is open book and students are allowed to use any resources with the exception of communication, through any medium. Like the written assignment, a hybrid test uses some subtle randomisation of Numbas questions, whilst students are also asked to upload their code through the virtual learning environment. Tests like these are held in computer clusters, which are not designed for administering examinations and do not offer protection from students reading the screen of their neighbour. Though the test is invigilated, the randomisation again offers some additional protection.

Students are provided with a mock exam in the same format. Because some of the marking is automatic, this gives an opportunity to receive instant feedback on their work. All 160 students who sat the class test in the 2019/20 academic year had attempted the mock, with 343 attempts in total, at an average of 63 minutes spent on each attempt. Fifty-eight percent of 81 surveyed students responded that they prefer the online format to a paper mock test, though many commented that, as well as solutions given inside the advice sections of the Numbas questions, they would like the lecturer to work through the mock test in a lecture.

5. Conclusions

The development of Numbas material for computing modules in the mathematics curriculum has been a success to date. Students particularly appreciate the immediate feedback offered by the formative material that is associated with each module topic. However, there is relatively low uptake of the Numbas material compared to the number of student's working through the handouts. At present, practice tests are separated from the handout on the same topic; Numbas can be embedded directly into web pages, and one possible solution is to replace the static questions in handouts with embedded Numbas questions, so that students can get feedback without disrupting their progress through the handout.

The assessment material currently does not extend Numbas to marking computer code directly. This is a limitation, as a hybrid approach is required to directly assess students' coding techniques and, for example, the presentation of results. Students would benefit from feedback on their code, especially in the introductory material for each programming language. Some early attempts have been made to add this functionality by connecting to an instance of the INGIInious platform from Numbas, and indeed formative questions making use of this method were popular with students.

Resources

Example questions can be found in the Numbas editor in the *Newcastle University Computing for Mathematics* project (Graham, 2020).

6. Acknowledgments

Thank you to Christian Lawson-Perfect, George Stagg and Lee Fawcett for their help with the design and development of questions.

7. References

- Beevers, C.E., Cherry, B.S.G., Clark, D.E.R., Foster, M.G., McGuire, G.R. and Renshaw, J.H., 1988. The CALM before the storm! CAL in university mathematics. *Computers & Education*, 12(1), pp.43-47.
- Derval, G., Gego, A., Reinbold, P., Frantzen, B. and Van Roy, P., 2015. Automatic grading of programming exercises in a MOOC using the INGIInious platform. *European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCs'15)*. pp.86-91.
- Foster, B., Perfect, C. and Youd, A., 2012. A completely client-side approach to e-assessment and e-learning of mathematics and statistics. *International Journal of e-Assessment*, 2(2).
- Graham, C., 2020. *Newcastle University Computing for Mathematics Numbas Project*. Available at: <https://numbas.mathcentre.ac.uk/project/5632/> [Accessed 29 February 2020].

Gwynllyw, R. and Henderson, K., 2009. DEWIS-a computer aided assessment system for mathematics and statistics. *CETL-MSOR Conference 2008*.

Lobb, R. and Harlow, J., 2016. Coderunner: A tool for assessing computer programming skills. *ACM Inroads*, 7(1), pp.47-51.

Newcastle University, 2020. *Numbas Editor*. Available at: <https://numbas.mathcentre.ac.uk> [Accessed 25 February 2020].

Sangwin, C., 2015. Computer aided assessment of mathematics using STACK. In: *Selected regular lectures from the 12th International Congress on Mathematical Education*. Springer, Cham. pp. 695-713.

Sangwin, C.J. and O'Toole, C., 2017. Computer programming in the UK mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), pp.1133-1152. <https://doi.org/10.1080/0020739X.2017.1315186>.

Stagg, G., 2019 *ncldata: NCL example data for use with MAS modules*. Available at: <https://rdr.io/rforge/ncldata/> [Accessed 25 February 2020].

Weir, I., Gwynllyw, R. and Henderson, K., 2017. Creating statistics e-assessments using Dewis with embedded R code. *MSOR Connections*, 15(2), pp.51-59. <https://doi.org/10.21100/msor.v15i2.422>.