Assessing programming in Numbas

Chris Graham

George Stagg, Aamir Khan, Christian Lawson-Perfect

School of Mathematics, Statistics & Physics E-Learning Unit

NUMBAS

Collaborative Award for
Teaching Excellence
2020

Xrun

In this talk

Motivation for the Numbas programming extension.

How the extension works.

Application of the extension to our programming modules.

Background

Programming in MSP

Programming is embedded in our mathematics and physics programmes, with compulsory modules at stages 1 and 2.

- Stage 1:
    - Problem solving with Python
    - Introduction to probability and R
- Stage 2:
    - Numerical methods (Python)
    - Computational statistics (R)

Module structure

Teaching weeks are typically structured as:

- 1 hour lecture: covering theory and examples

- 2 hour practical: hands-on practical with embedded exercises

I loved the format; one theory lecture and a two hour practical really worked as I learnt the concepts and then was given contact time to make sure that I understood them

Challenges

Student experience

Students have different backgrounds and experiences with programming and computers.

...from A-level computing experience to computer-phobic.
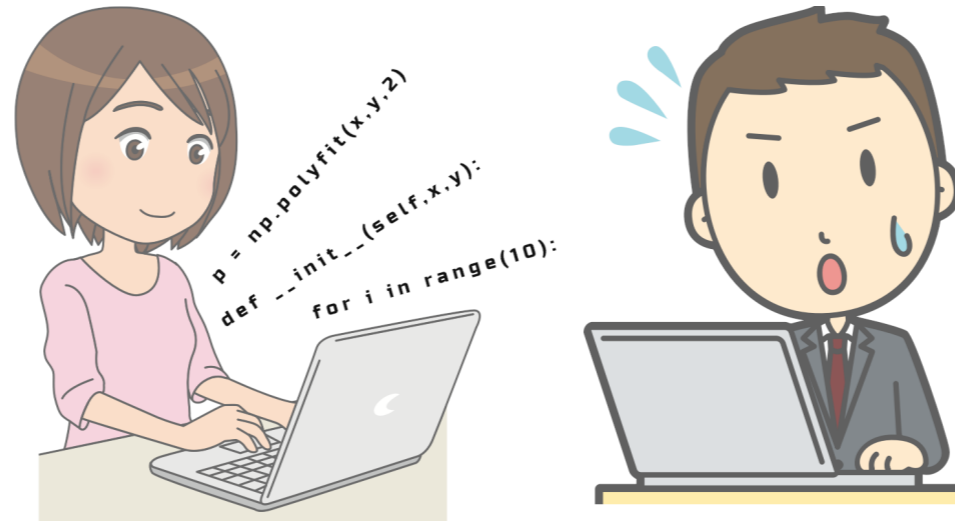
Answering queries

Our experience was that demonstrators often received "low-level" queries about exercises, often requiring no more than signposting.

How do I start this exercise?

I don't understand - what does this mean?!

Or... students gathered around one competent student working on problems in a very non-interactive way

Exercise feedback

Handouts contained written exercises to complete.

Students work through the material at very different speeds, so timing feedback is difficult:

- Solutions released during session, end of session, the next week…?
- Work through on big screens, but when?

Summative Assessment

Programming is difficult to assess.

Little variation between student answers - difficult to detect collusion.

Marking is very time-consuming and repetitive.

Numbas for programming

Why Numbas?

Many of the lucrative features of maths e-assessment also apply to basic programming: automatic marking, instant feedback, randomisation...

Numbas was ideal:

- familiar to our students
- tap into well-developed features: randomisation, steps, adaptive marking, explore mode, etc
- mix coding questions with other part types

Programming in Numbas: first iteration

First attempts used a "follow along" approach:

- Student is presented with a problem in Numbas and completes problem in R/Python.

- Numbas follows along with the analysis in JME, to be able to mark the student's response.

Useful and we still use it for some questions, but limited as it does not directly mark student code.

Programming in Numbas: second iteration

The version used in 2020/21:

- Student enters a code answer

- Student's answer plus unit tests are sent to a server running R / Python

- Server sends back responses which Numbas can use to mark

Very useable, but relies on a server to run code and is fixed to Newcastle.

The Numbas programming extension

Released earlier this year.

Uses WebAssembly/Emscripten to compile Python (Pyodide) and R (WebR) to run code entirely self-contained in a web browser.

No reliance on a server to run code; nothing tied to Newcastle. Therefore shareable!
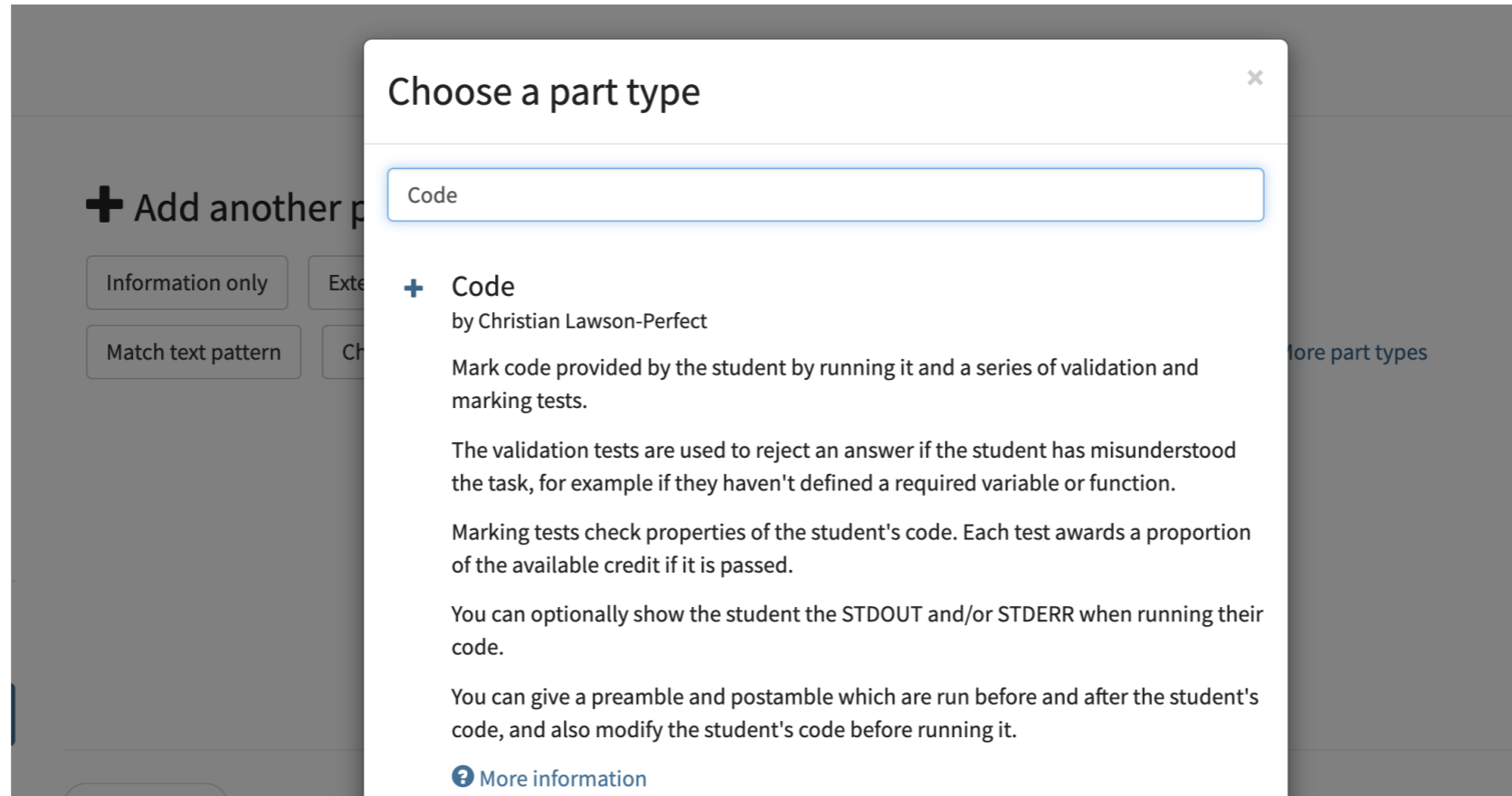
Loading...

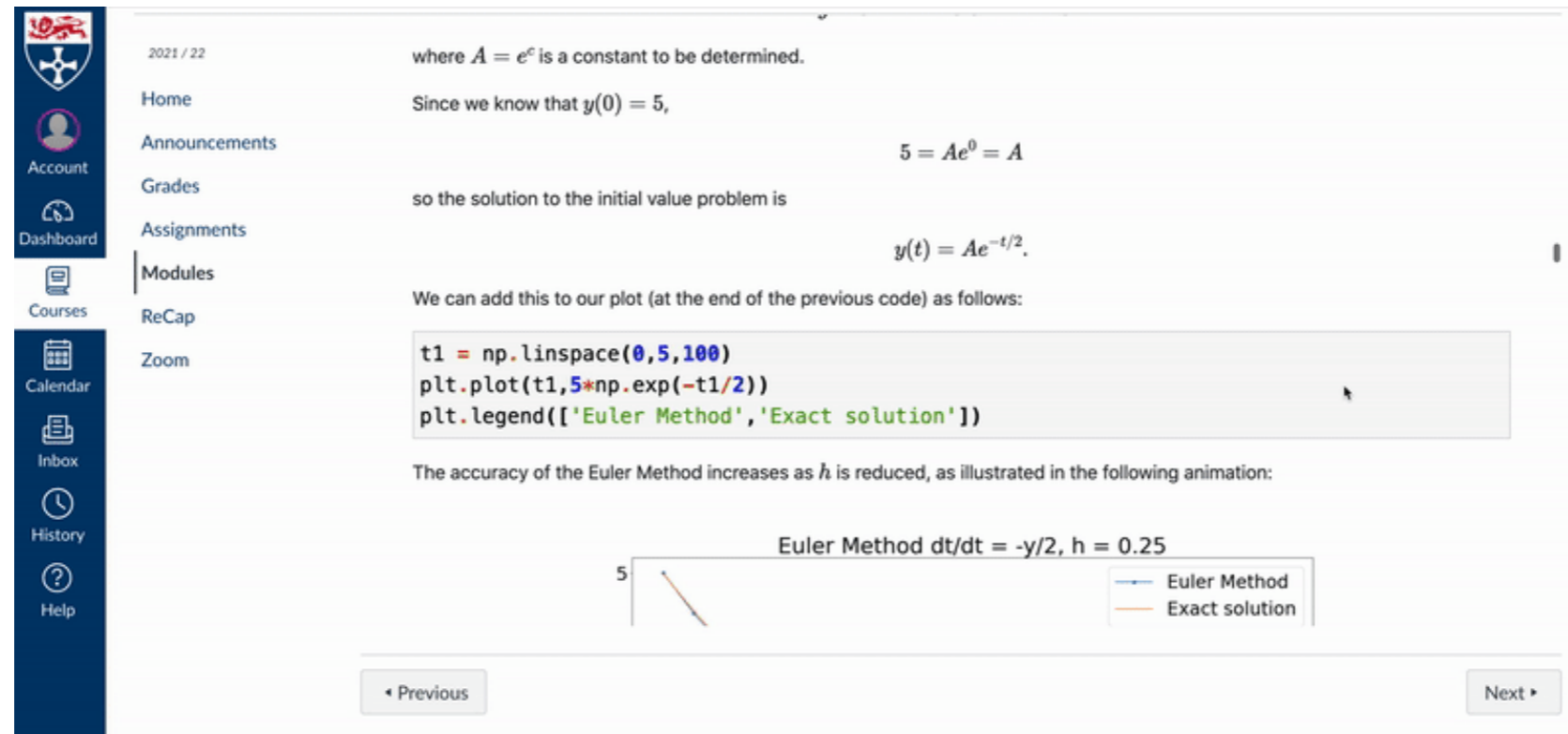# Behind the scenes

Custom part type

Handout exercises

Exercises embedded in the handouts as Numbas questions:

- immediate marking and feedback
- alternative answers to catch common errors
- randomisation to try similar questions over and over
- full worked solutions

And can be scaffolded with steps, or multiple parts to build up solutions.

Embedded exercises

Exercises are embedded directly in handouts:

New experience of handout exercises

Students are able tackle exercises and learn at their own pace.

During Covid, modules ran successfully with asynchronous handout material.

Since returning to campus, the handout exercises have transformed the demonstrator-student dynamic.

Assessment

Formative "Test Yourself" question sets accompanying each week of content.

In-course assessments and final exams were hybrid, ~60% auto-marked: focus manual marking where it is most effective.

Student feedback

I like that I can work through the handout so that I'm learning in the best way for myself, at my own pace

Really efficient way off working, using the handouts and tests!!

The feedback from our assignments was detailed and personal to us and gave us information on what we did well and where we can improve.

More examples

Scaffolding with steps

Loading…

Explore mode

Loading...

Mark a plot (experimental)

Loading...

R example

Loading...

Find out more

- Numbas blog post on the programming extension
- Demo exam
- Extension documentation

Thank you to my colleagues (George, Christian, Aamir) in our E-learning Unit.