Adding computing and interactivity to traditional, online, and blended courses using Wolfram Language tools

Peter Barendse, Alexandra Berg, Thiago Bergamaschi, Lauren Berk

We will show how to add computing and interactivity into traditional, online, and blended courses using a suite of tools our group has developed in Wolfram Language. We will also discuss innovative education techniques related to bringing computational thinking into traditional topics within massive online courses, strategies for modern curriculum development, interactive learning, and data analysis that can be used to iteratively improve curriculum and eventually lead to automated tutors in classroom.

We will first describe the roots and initial phases of our project, starting with W. Craig Carter and how computing became an integral part of his subject "mathematical methods for materials science and engineering" (MIT course number 3.016). Then, we'll discuss the tools that were developed in the Wolfram Language to deliver electronic scaffolded lessons that were exchanged via email and graded by hand ("CoSets"). We'll also talk about some tools for creating instant, anonymous in-class interaction with code ("CoLabs"), you can see a short explanation video at https://www.dropbox.com/s/i1750b1wmscxtcq/CoLab.mp4?dl=0

In the Fall of 2017, we led two small first-year seminars called "Coding your GIRs". Our goal was to teach students how to use code to help understand and answer questions in core subjects like physics, math, chemistry, and biology; and to feel empowered to independently dive deeper into those subjects. During this pilot, we wanted to learn if it was feasible to teach an educationally-diverse group of students how to code in the context of subjects they were learning concurrently. There was diversity in both prior knowledge of the science and coding, but there was more diversity in coding ability since this skill is still not taught to most high school students.

This spring, we introduced a series of graded coding assignments using the CodeSeal system into 8.02, (a 700-student GIR covering electromagnetism for first year students at MIT). We also added these assignments to the MITx version of this course, though they are optional assignments for now.

In each seminar and class, we are collecting data on how students are approaching problems, including standard interaction data about time-on-task, keystrokes, and button presses.

We also record each piece of code a student writes while they prepare their final solution. In addition to a student's final solution, this data provides insight into how much effort went into developing it and how they arrived at it.

We will present some preliminary analysis of this data and share our plans for further analysis that we will use to inform future iterations of Code Seal. This data will eventually allow us to create an automated teaching and grading assistant that can help flag students who need immediate intervention, classify students programming style, classify different learning paths, and identify important learning moments within those learning paths.

We will then look ahead to integrations we are developing between Code Seal and MITx. We will demonstrate how LTI components developed in the Wolfram Language can be embedded within MITx course material to allow for complex, interactive problems both on computers and mobile devices. The integration allows for completely customizable grading schemes, extending far beyond the options of MITx (multiple choice, quantitative answer with a tolerance) to include symbolic grading, grading code and written text, and assigning partial credit scores.