In teaching, I have used MATLAB in a lab tied to a Geomorphology course where no prior knowledge or experience with MATLAB was assumed. It was introduced to develop the concept of hillslope diffusion and erosion rates, basic concepts in Geomorphology, and expose students to modeling and a common analytical technique in the field. My experience in learning MATLAB was as a novice in programming but with a need to use it for dissertation work. Being familiar with the challenges on both sides, I have developed an interest and fascination with finding ways to unravel the logical framework of relatively basic computational problems and with the translation process into MATLAB syntax for producing code. This challenge, as a teacher and a learner of MATLAB, is really the same for both and revolves around developing the ability of thinking through modeling problems. What I have come to find important is the need to see beyond the particular skills for composing the syntax to solve a particular problem or answer a question to understanding the logical steps and structure needed to solve a computational problem or model a concept. Recognizing the latter can improve the chances of skill transference to other problems and fill educational gaps in quantitative cognitive development that go far beyond building isolated skills.

The lab in the Geomorphology course only covered a couple of weeks and I taught it as a Teaching Assistant. However, due to my own experiences of learning MATLAB for particular research questions and problems and teaching a limited range of MATLAB code, I am interested in designing a beginner's MATLAB course for Geoscience where some basic and common problems could be generalized on the basis of seeing logical structures needed to approach them. A full semester course would permit time for activities that could better ground conceptual foundations, which I am interested in doing. Two main ways of addressing the challenges of 1) understanding the logic of a computational problem and translating it into MATLAB syntax and 2) being able to transfer it and adapt it to other problems are to have students communicate verbally and in writing about what they did, why they did it, what hurdles were encountered, and whether and how they were overcome. They would need to disclose what they understood and what they did not, critical reflective behavior needed for doing science, but something most find uncomfortable. Students would see multiple ways of devising code to solve the same or similar problems, they would be exposed to common errors and have a chance to help out fellow students work through a problem. Weekly discussion would encourage dialogue about the syntax, the scientific concepts, and further questions about what more could be done with a code, and therefore the analysis of data. In addition, the weekly dialogue would serve as a springboard for forming an in-house community of MATLAB users. The students would be asked to submit written portions about the code they generated and answer questions about what the parts of the script were doing in the program and how the results were reflected in a plot or a visualization.

Addressing the challenges of strengthening quantitative thinking in Geoscience requires planning in curriculum and pedagogy and in expectations of what students should come out of such a curriculum with. This is especially so in teaching quantitative methods because multiple aspects need to come together: the scientific concepts, the logic of problems, and the computational syntax and structures. Students at all levels and backgrounds in a Geoscience program could benefit. Those who go on to using other kinds of software will still benefit from the ability to conceptualize a computational problem and its underlying logic. Those who continue on in research areas that make use of MATLAB will have built a solid foundation.