COMPUTATIONAL THINKING IN LINEAR ALGEBRA

Spencer Bagley, University of California, San Diego
Jeff Rabin, University of California, San Diego

In this work, we examine students' ways of thinking when presented with a novel linear algebra problem. We have hypothesized that in order to succeed in linear algebra, students must employ and coordinate three modes of thinking, which we call computational, abstract, and geometric. This study examines the solution strategies that undergraduate honors linear algebra students employ to solve the problem, the variety of productive and reflective ways in which the computational mode of thinking is used, and the ways in which they coordinate the computational mode of thinking with other modes.

Key words: Justification, Linear Algebra, Problem Solving, Procedural Understanding

Purpose and Background

The field of linear algebra has recently attracted much attention in the literature; many studies have examined students' difficulties in linear algebra (e.g., Carlson, 1993; Sierpinska, 2000). By contrast, this study examines students' abilities. We focus on the ways in which students are able to think productively, in order to provide a model of successful thinking in linear algebra. In particular, we examine the productive ways students use computational thinking to reason through a novel problem about basis.

We have adopted a three-fold taxonomy of ways of thinking in linear algebra. The ways of thinking we have identified, which we refer to as *abstract*, *computational*, and *geometric* thinking, are roughly parallel to those of Sierpinska (2000) and Hillel (2000). *Abstract thinking* is indicated by working with vectors as formal objects characterized by the vector space axioms, or by statements of definitions and theorems in coordinate-free language. *Computational thinking* is typically indicated by explicit reference to particular algorithms, such as row reduction or the Gram-Schmidt process, and representation of vectors in terms of their components. It includes not only carrying out a computation, but also choosing the appropriate computation to solve a particular problem and understanding what the result of that computation means in context. Finally, *geometric thinking* is indicated by the use of language such as line, plane, ray, angle, length, or intersection, and geometric knowledge such as the Pythagorean Theorem. Usually this language is used in the context of $\mathbf{R}^2$ or $\mathbf{R}^3$, or by analogy with one of these spaces.

We argue that in order to be successful in linear algebra, students must come to be able to use and coordinate each of these three ways of thinking; we follow Hillel (2000) in attributing many student difficulties to their trouble switching between, or relating, these three languages. These ways of thinking are not sharply demarcated; powerful tools for reasoning can be found at their overlaps. This study was designed to address the following research questions: What strategies do students use to solve a novel problem (detailed below)? What are the uses, affordances, and constraints of each mode of thinking? In what ways do students coordinate these modes of thinking?

Design, Settings, and Methods

Eight first-year undergraduate honors linear algebra students from a large public university in the southwestern United States volunteered to participate in individual clinical interviews, conducted near the end of the course. The interview centered on the "Michelle problem":

> Michelle would like to create a basis for $\mathbf{R}^4$. She has already listed two vectors **v** and **w** that she would like to include in her basis, and wants to add more vectors to her list until she obtains a basis. What instructions would you give her on how to accomplish this?

Based on a prior study (Wawro, Sweeney, & Rabin, 2010), this problem was expected to provide rich opportunities for each mode of thinking. The problem was initially presented abstractly, as above, and students were encouraged to describe a general method by which the problem could be solved. After this, specific vectors were given, and the students were asked to try using their method on these given vectors. We asked a number of follow-up questions to probe students' intuition and solution procedures, whether students could formulate their solution procedure in an algorithmic way, and whether they could justify their procedure.

The interviews were videotaped and transcribed, and students' written work produced during the interviews was retained. These recordings, transcripts, and written documents formed the corpus of data analyzed in this study. Using grounded theory (Strauss & Corbin, 1994), we coded students' utterances as instances of abstract, computational, or geometric thinking, referring to written work for confirmatory evidence, and documented the ways in which students used each way of thinking.

As we began analyzing the data, we noted a preponderance of computational thinking. Since we saw a much more balanced use of modes of thinking in the pilot study, this result was unexpected. We were thus led to ask, in addition to the initial research questions, why students used computational thinking so much, and in what productive ways they used it.

Results

One result from our data is the surprising variety of ways in which students were able to productively use computational thinking. Computational thinking is often maligned in the literature and in educators' opinions. The common perception is that computational thinking is exclusively procedural, that students simply wish to feed numbers unreflectively into an algorithm, and that they commonly make significant errors while doing so. This analysis highlights, by contrast, the *productive* and *reflective* ways students can use computational thinking in linear algebra. For example, using computational thinking, students were able to generate strategies for choosing additional vectors, produce proofs of their methods, and get "un-stuck" when they encountered roadblocks.

Generating strategies

Computational thinking can inspire the creation of strategies for choosing vectors. Evan (all names are pseudonyms), for instance, initially proposed a strategy of pure guess and check. The interviewers asked him if there was a strategy he could use to make informed guesses:

> *Evan*: Well, I'd first, I'd reduce the first, these two vectors [i.e., **v** and **w**] and let – so I'll just make a 4x2 matrix. And to reduce – If I can have two pivotal columns, so like, for these two, because they are linearly independent, so sure I can have two pivotal columns. And I can just pick another two that has different pivotal columns [*sic*: he means rows], like in the third column or fourth column, to get one.

When asked for a way to make better guesses, he suggested row-reducing **v** and **w** to see where the pivots were, then choosing standard basis vectors to provide the missing pivots. Thus, by using computational reasoning to think about the test his vectors must pass, he was able to engineer vectors that are guaranteed to pass it. Although he does not formulate his reasoning as a proof or justification, it essentially serves as such.

Computational proofs

We had anticipated that the justification questions would prompt abstract thinking. However, we found that several students were able to produce fully valid proofs using computational thinking alone. Bob, for example, produced a proof by analyzing his algorithm. His strategy was similar to Evan's; he row-reduced the two given vectors, then used standard basis vectors to supply the missing pivots. He argued abstractly that the new vectors will be independent, and then justified this claim computationally (transcript edited for length and clarity):

> *Bob*: Well, by the definition of linear independence, their matrix has to row-reduce to the identity. All the columns will be pivotal. So, by using these facts about linear independence and pivotal columns, this procedure is a way to find two more columns that will be pivotal columns independent of the other ones already found. With the two vectors she already has, she has two pivotal columns here, and they both represent pivotal ones. It's just a way to find the other two columns that won't form the same pivotal row as another one, so that they'll all be independent.

This technique of proof by analysis of an algorithm is a particularly valuable way of constructing formal justifications. Many textbook proofs in linear algebra proceed in a similar fashion. The fact that students are capable of producing such justifications is perhaps an argument for teachers to highlight this method when discussing proof techniques.

Roadblocks and resolutions

Students often encounter difficulties and "get stuck" when solving a problem. We initially conjectured that one way of getting around such roadblocks would be for students to transition to another way of thinking. However, we found that in many situations where students encountered difficulties relating to their computational thinking, they were able to resolve the difficulty by continuing to think computationally.

Greg is an example of a student who encountered such a roadblock. His solution method was to guess a vector to potentially include in the basis, then show that it would work by showing "that it's not a linear combination of these two [i.e., **v** and **w**]." He augmented the two given vectors with a vector he guessed and proceeded to row-reduce the resulting matrix. He hit a roadblock when the third row of his matrix reduced to [0 0 | -15], saying, "I'm actually kinda confused about what this tells me. Did I make a mistake?"

It appeared that Greg is used to rows of the form [0 0 | 1] signaling that something bad has happened. However, he was able to step back and reason (computationally) about this result, coming to the conclusion that it was the appropriate one:

> *Greg*: All right. Well actually, if I continue row-reducing this, then I would get a 1 here... then that would make it unsolvable. So then I suppose ... Yeah. Okay. And then, that would mean that this can't be a linear combination of these two. So then, it's not in the span.
>
> *Int*: And is that good or bad, for purposes of this problem?
>
> *Greg*: That's good. This could be an additional vector for a basis.

The fact that the system he had constructed was "unsolvable" made him second-guess himself for a moment, but by reflecting on the framing of the algorithm, he was able to realize that this is exactly what he had wanted. By *framing* we mean the context in which the algorithm is applied; its goals and the meaning of its inputs and outputs. Reasoning about the framing of an algorithm in this way is a productive overlap of the computational and abstract modes of thinking.

Our talk will explicate a number of other findings, including an elaboration of students' productive uses of the other two ways of thinking. Additionally, we will discuss at greater length the ways in which students coordinated and transitioned between multiple ways of thinking.

Conclusion

Our data contribute to the reconceptualization of procedural knowledge as a useful and productive mathematical resource (see, e.g., Star, 2005). Additionally, they show that computational thinking is more varied, flexible, and sophisticated than the common perception. We have presented evidence of students producing sophisticated strategies for choosing vectors, justifying their approaches, and resolving problems they encounter through the use of computational thinking. This evidence of student ability provides direct recommendations for pedagogical practice in linear algebra.

This study contributes to the literature by documenting student abilities in linear algebra. While it is true that these are advanced students, and thus our results may not generalize to the broader population of undergraduate linear algebra students, it is still useful to know what students are capable of. Although they are honors students, they are still freshmen, so it is reasonable to expect that their abilities are within the range of potential development of other students. Our study provides evidence that the fog need not always roll in (Carlson, 1993).

Discussion Questions

1. In what other ways might computational thinking be useful in linear algebra? in other areas of undergraduate mathematics?
2. How distinct are the three modes of thinking? Should they be thought of as separate but coordinated, or as shading into one another at their boundaries?
3. What other overlaps between the three modes of thinking might be expected?

References

Carlson, D. (1993). Teaching linear algebra: Must the fog always roll in? *College Mathematics Journal, 24,* 29–40.

Hillel, J. (2000). Modes of description and the problem of representation in linear algebra. In J.-L. Dorier (Ed.), *On the teaching of linear algebra* (pp. 191-207). Dordrecht: Kluwer.

Sierpinska, A. (2000). On some aspects of students' thinking in linear algebra. In J.-L. Dorier (Ed.), *On the teaching of linear algebra* (pp. 209-246). Dordrecht: Kluwer.

Star, J. R. (2005). Reconceptualizing procedural knowledge. *Journal for Research in Mathematics Education, 36*(5), 404-411.

Strauss, A., & Corbin, J. (1994). Grounded theory methodology: An overview. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 273 – 285). Thousand Oaks: Sage Publications.

Wawro, M., Sweeney, G., & Rabin, J. (2010). Subspace in linear algebra: investigating students' concept images and interactions with the formal definition. *Educational Studies in Mathematics, 78(1),* 1-19.