Computational Fluency at What Price?

Jane Keiser

We are currently experiencing an era of change in education and in the topics we teach at each grade level. Although many of these changes are healthy and necessary, I feel that one particular change in the Common Core State Standards for Mathematics (CCSSM) will set back our mathematics learning. Specifically, CCSSM seems to perpetuate the belief that the end goal in computational learning should be students who are fluent in the use of the “standard algorithm” for all four operations (CCSSI 2010). (See CCSSM, p. 29, no. 4; p. 35, no. 5; and p. 42, no. 2.)

For many years, mathematics education researchers have focused on computational fluency and how it can best be developed in students (i.e., Campbell, Rowan, and Suarez 1998; Kamii and Dominick 1998; Russell 1999; Schifter 1999). The NCTM Standards (2000) suggest that all students who finish fifth grade should be fluent in all whole-number computations:

Computational fluency refers to having efficient and accurate methods for computing. Students exhibit computational fluency when they demonstrate flexibility in the computational methods they choose, understand and can explain these methods, and produce accurate answers efficiently. (p. 152)

My view is that students can be efficient, accurate, and flexible without using “standard” algorithms. I also believe that too much time is spent teaching and practicing standard algorithms, with negligible returns in actual learning and proficiency.

I began my career in the early 1980s as a seventh-grade high-level mathematics teacher. It was in the era of the spiral curriculum (Bruner 1960). The textbooks for fifth grade, sixth grade, and seventh grade contained the same content in each chapter; for the higher grades, another layer of depth was added to the learning. When I asked my class to work on long-division problems, it was difficult to motivate students because they had seen and practiced the algorithm for several years, felt that they were fluent (although they were far from it), and wanted to move on. In my work with high-level seventh-grade students, roughly one-half of them were fluent with the standard algorithms. This leads me to generalize that fluency with these algorithms was rare for the average student. Certainly, if used well, these algorithms are efficient. However, my students struggled to be accurate, and they were rarely flexible.

If I asked them to multiply or divide by a power of ten, they used the standard algorithm rather than simply moving the decimal point. They lacked the ability to determine if an answer was reasonable, and estimation skills were also lacking. Computing was a mindless procedure of steps, and either you could do it or you couldn’t. The students who could correctly perform computations were sent on to algebra; those who couldn’t were sent to general math. Do we really want to return to this scenario?

Today, I am a math educator in a mid-size state university that is known for having high standards for
Shouldn't our goal be to develop students who are computationally fluent?

Evaluate $1056 \div 7$, and round the answer to the nearest tenth.

Last year, all but 1 of my 61 students recalled being taught the long-division algorithm and over 75 percent remembered learning how to find a solution past the decimal point. It seems that even with several years of reform behind us, the majority of teachers still believe that the standard algorithm is necessary and worth teaching. If the result is that just over 25 percent of college students can perform it correctly, is it worth the time that was spent trying to get the other 75 percent of college students and perhaps all noncollege students to learn it? Think of all the other pertinent mathematical topics that could be explored if we would stop expecting our students to achieve fluency with the standard algorithms. Think of the change in attitudes!

My eleven-year-old son, John, has had the benefit of being taught by good teachers who effectively used a National Science Foundation–developed K–grade 5 math curriculum. As a result, he has seen and practiced the standard algorithm for each of the operations except division and has learned a variety of strategies for all operations. However, he never quite caught on to the “borrowing” method for subtraction, so he never uses it. With subtraction, his method depends on the situation. He likes to add up, and he likes a method that uses negative numbers and goes left to right. The method that he uses depends on the size of the numbers and the context of the problem he faces. With multiplication, he usually uses a partial-products strategy. However, it, too, depends on the situation. With division, he is very efficient using repeated subtraction but sometimes uses multiplication only and adds up to the dividend. Isn’t this the flexibility that we care about?

One common criticism about using nonstandard strategies is that students take too long to get the answer when using these other methods. They cannot possibly be efficient using them. I do not believe that this is the case. As a result of learning these strategies, John can often do them in his head. His mental-math ability far surpasses my abilities at his age, and they certainly surpass most of the high-level seventh graders I taught in the 1980s. He uses judgment in determining his approach, and because he has amassed a bag of tricks that all work and are supported with sound mathematical properties, he can quickly choose which strategy would best accomplish his specific problem.

Aren’t we tired of seeing students who fail to think clearly about their work? I am thrilled that John and his classmates think before they act on a computation. Pushing students too quickly to the formal algorithms causes them to use such algorithms before they are fully understood. Therefore, students simply learn them as procedures, which are accomplished without the brain being fully engaged.

CCSSM now requires that students be fluent with the standard algorithm for addition and subtraction by fourth grade, the standard algorithm for multiplication by fifth grade, and the standard algorithm for division by sixth grade. I do not mind if students learn about and try these algorithms by the time they exit these grade levels, but to require fluency means that students use and practice these algorithms a lot, so that accuracy and efficiency are achieved. Forget about flexibility; these algorithms can be done in only one way. It is my view that standard algorithms should not be taught until students have a sound conceptual understanding of each operation.

The alternative strategies that are currently being taught in some elementary schools make this understanding very natural, since the strategy often fits the model of the operation being used (i.e., the array or area model for multiplication and repeated subtraction for division). If students experience each operation in various ways, they should be more likely to use judgment when they choose a strategy for a new situation. Only then, I feel, will they be ready to move forward to learning a standard algorithm. But even if we get to this point, do we really need to spend so much time teaching our students algorithms that any machine can do for us? What I am afraid will result is a return to mindless practice and a new population of students who lack discernment when it comes to reasonable answers and good problem-solving behaviors.

Shouldn’t our goal be to develop students who are computation-
ally fluent rather than to develop students who can mindlessly use a procedure? Let’s work for efficiency, accuracy, and flexibility using appropriate strategies for a given task. Let’s forget about designating exactly which algorithm we want to be called “standard.”

REFERENCES

Jane Keiser, keiserjm@muohio.edu, teaches preservice math teachers at Miami University in Oxford, Ohio. She is interested in students’ conceptual understandings of the four mathematical operations.