Written algorithms in the primary years: Undoing the 'good work'?

1 AUTHOR:

Doug Clarke
Australian Catholic University

Written algorithms in the primary years: Undoing the 'good work'?*

Doug M. Clarke
Australian Catholic University

The teaching of conventional written algorithms in primary schools dominates the curriculum with concerning effects on both student understanding and self-confidence. In this paper, I summarise research findings and the opinions of key writers, with particular emphasis on the potential dangers of introducing conventional algorithms too early, and share research data from a follow-up study to the Victorian Early Numeracy Research Project. I make the argument that there is far more important work to be done in these years in developing concepts and strategies for mental computation, and offer some practical suggestions.

Advantages and disadvantages of teaching algorithms in the early years of primary school

Given that the teaching and regular practice of written algorithms is widespread in primary classrooms, there must be a number of reasons for their prominent role in everyday mathematics activity. Plunkett (1979), Thompson (1997), Usiskin (1998) and other writers offered several reasons for this. These included:

- algorithms have been traditional primary mathematics content around the world for many years;
- algorithms are powerful in solving classes of problems, particularly where the computation involves many numbers, where memory may be overloaded;
- algorithms are contracted, summarising several lines of equations involving distributivity and associativity;
- algorithms are automatic, being able to be taught to, and carried out by, someone without having to analyse the underlying basis of the algorithm;
- algorithms are fast, with a direct route to the answer;
- algorithms provide a written record of computation, enabling teachers and students to locate any errors in the algorithm;
- algorithms can be instructive;
- for the teacher, algorithms are easy to manage and assess.

These, at first glance, appear to be a powerful list of reasons why the teaching of conventional written algorithms has been traditional and should continue in the primary years. However, a number of writers have identified potential dangers to teaching conventional written algorithms to primary children (Kamii & Dominick, 1998; McIntosh, 1998;

* This paper has been accepted by peer review.

Northcote & McIntosh, 1999; Thompson, 1997; Usiskin, 1998). These can be summarised as follows.

- They do not correspond to the ways in which people tend to think about numbers; for example, in the context of use of most conventional algorithms, the '4' in the number 547 is treated as '4' and not '40'.
- They encourage children to give up their own thinking, leading to a loss of 'ownership of ideas'. The original purpose of algorithms in previous centuries was for clerks to be able to carry out a large number of calculations in a short period of time. Thinking was not the focus, but rather quick and reliable answers. Technology changes the relative importance of algorithms — some become more important, some less important. Most clerks today, given a large number of calculations, would use either a calculator or pre-prepared spreadsheet to carry out these calculations.
- The traditionally-taught algorithms may no longer be the most efficient and easily learned. There is evidence (e.g., Groves & Stacey, 1998; Shuard, 1990) that children in classrooms where the regular use of calculators for conceptual development is encouraged may develop an algorithm for subtraction that blends place value understanding and negative numbers (e.g., for 354 – 278, a child might use an algorithm that involves the following steps: 300 – 200 = 100; 50 take 70 is -20; 4 take 8 is -4; so the answer is 100 take 20 take 4, giving 76).
- They tend to lead to blind acceptance of results and over-zealous applications. Given the focus on procedures that require little thinking, children often use an algorithm when it is not at all necessary. Hope's (1986) example of finding $100 – $99.95 using the conventional algorithm is a classic case of this.

There is also the issue of relevance. Adults use formal written computation for only a small proportion of their calculations. Northcote and McIntosh (1999) found, in a survey conducted with two hundred adults over a twenty-four-hour period, that only 11.1% of all calculations involved a written component. It has become increasingly unusual for standard written algorithms to be used anywhere except in the mathematics classroom. Most calculations required only an estimate. They also found that for 60% of all calculations, only an estimate of the correct answer was needed. The ways in which conventional algorithms are traditionally taught discourage the application of number sense by estimating first or assessing the reasonableness of the answer afterwards.

An example from research of the possible detrimental effect of teaching written algorithms in the early years on children's mental strategies and number sense

Narode, Board and Davenport (1993) conducted a year-long study of nineteen first, second and third-grade students, involving videotaped interviews. All students were asked to solve two-digit addition and subtraction computations embedded in simple story problems and in familiar contexts, such as stones or marbles. The students were asked to solve each problem, first using base 10 blocks and then mentally or with paper and pencil as they chose. The students were also asked whether they knew of any alternative ways to solve the problem.

Interestingly, almost all children interviewed before instruction in addition and subtraction algorithms used invented strategies which used traditional, front-end approaches (not the usual right-to-left order).

The researchers discussed the case of Jamie (a second grade girl), who was interviewed on several occasions during the school year. It is important to note that Jamie had not met conventional algorithms prior to second grade, but was introduced to them during second grade.

> Early in the school year, Jamie successfully added 19 and 26 mentally: 'I know I have 30 because I have a group of ten and two more tens. Then if I take 1 from the 6 and give it to the 9, I'll have another group of 10. That leaves five left, so the answer is 45.'
> After five months of school and work with conventional algorithms, Jamie attempted to add 34 and 99 by beginning to group the 9 tens and 3 tens, then stopped and said, 'Oh, I have to add the ones first.' She then grouped the units, and traded for a ten to solve the problem.
> In the last month of the school year, asked about the possibility of solving the problem by adding the tens first, Jamie emphatically stated, 'No, you never add the tens first.' She was invited to suggest another way that the problem could be solved. Jamie suggested that another way to solve the problem might be to know the answer from memory. Finally, she was confronted with her own invented strategy, from earlier in the year, as a strategy 'someone used' to add 49 + 19 ('I think of 50 + 19 and then subtract one to get 68'). When asked if she thought this method might work, she replied 'If you know that way, it's okay, but it's much, much better to just add the ones first.' (p. 259)

This one example is a powerful demonstration of the way in which a child can move from trusting their understanding of numbers and flexible strategies to following a single procedure without much hesitation. Narode, Board and Davenport (1993) summarised their findings:

> We believe that by encouraging students to use only one method (algorithmic) to solve problems, they lose some of their capacity for flexible and creative thought. They become less willing to attempt problems in alternative ways, and they become afraid to take risks. Furthermore, there is a high probability that the students will lose conceptual knowledge in the process of gaining procedural knowledge (p. 260).

The *Early Numeracy Research Project* (ENRP) was established in 1999 by the (then) Victorian Department of Education, with a Prep to Grade 2 mathematics focus.

The ENRP became a collaborative venture between Australian Catholic University, Monash University, the Victorian Department of Employment, Education and Training, the Catholic Education Office (Melbourne), and the Association of Independent Schools Victoria. The project was funded to early 2002 in thirty-five project ('trial') schools and thirty-five control ('reference') schools.

There were two main features of the ENRP that made it different in important ways from previous projects: the ENRP growth points and task-based interview. The framework of growth points provides a means for *understanding* young students' mathematical thinking in general, and the interview provides a tool for *assessing* this thinking for particular individuals and groups. The main project concluded in early 2002 (for further information, see Clarke, 2001), but the focus here is on a small follow-up project and the data which emerged from it.

It was agreed that in November 2002, a sample of Grade 3 students would be interviewed by the research team at Australian Catholic University, to be followed by another interview period in November 2003, with Grade 4 students. These students would be chosen from those who had participated in six previous interviews, at the beginning and end of the Prep, Grade 1, and Grade 2 years. Six hundred and thirty Grade 3 children were interviewed in 2002, and five hundred and seventy-two of the same children (now at the end of Grade 4) were interviewed in 2003.

The mathematical domain of *addition and subtraction strategies* provided data that was of some concern, in relation to the traditional emphasis on teaching written algorithms in Grades 3 and 4.

The six growth points for the domain of addition and subtraction strategies are:

1. Count-all (two collections)
   Counts all to find the total of two collections.
2. Count-on
   Counts on from one number to find the total of two collections.
3. Count-back/count-down-to/count-up-from
   Given a subtraction situation, chooses appropriately from strategies including count-back, count-down-to and count-up-from.
4. Basic strategies (doubles, commutativity, adding 10, tens facts, other known facts)
   Given an addition or subtraction problem, strategies such as doubles, commutativity, adding 10, tens facts, and other known facts are evident.
5. Derived strategies (near doubles, adding 9, build to next ten, fact families, intuitive strategies)
   Given an addition or subtraction problem, strategies such as near doubles, adding 9, build to next ten, fact families and intuitive strategies are evident.
6. Extending and applying addition and subtraction using basic, derived and intuitive strategies
   Given a range of tasks (including multi-digit numbers), can solve them mentally, using the appropriate strategies and a clear understanding of key concepts.

The data for the relevant growth points for Grade 4 students at the end of the school year are given in Table 1.

Table 1. Percentage of Grade 4 students achieving each ENRP growth point
for domain of addition & subtraction strategies (November 2003).

Growth points	Reference schools % n = 174	Trial schools % n = 398
0		
1	0.6	1.0
2	12.6	8.0
3	7.5	4.8
4	20.1	25.4
5	46.6	41.2
6	12.6	19.6

If the increasingly-common argument that students should not be taught conventional written algorithms until they are able to add and subtract two-digit numbers in their head (which underpins the latest UK mathematics curriculum is accepted, then this means that around 40% of trial school and reference school students were not ready for this content by the end of Grade 4. These are the students who have not yet grasped both basic and derived strategies, with many counting by ones for all such problems. This is not necessarily a criticism of the teaching or the students, but may say something about readiness for these ideas.

It is also worth noting the statement in the most recent New South Wales Mathematics syllabus (Board of Studies New South Wales, 2002) that 'formal written algorithms are introduced after students have gained a firm understanding of basic concepts including place value, and have developed mental strategies for computing with two-digit and three-digit numbers' (p. 9). I strongly agree with this position.

Of course, students can be increasingly encouraged to record the various steps in their calculations, in ways that make sense to them. The danger is not so much with the written form, but the imposition of the teacher's method for recording, which, as was shown earlier, can have unfortunate consequences. In this way, students are developing and gradually refining their own invented algorithms, in conversations with their peers and the teacher.

Are any student-invented algorithms okay?

It is a natural process for children to record their thinking on paper, as the numbers become too large for everything to be retained in their head. As students start to develop their own algorithms, a question arises: are any student-invented algorithms acceptable? How should children's invented algorithms be treated in the classroom?

Early in school, given that the algorithm leads to a correct answer, the answer is probably, 'Yes: they are okay'; but over time, we want to encourage children to consider whether the procedures are:

- efficient enough to be used regularly without considerable loss of time;
- mathematically valid;
- generalisable (can the algorithm be applied to the full range of problems of the type being solved?) (Campbell, Rowan & Suarez, 1998).

Occasionally, teachers claim that, 'only the brighter children can create their own algorithms'. Those involved in projects that encourage children to create their own algorithms dispute this, but even if it were true, the encouragement for children to do so will likely yield a range of algorithms. These can be shared publicly and discussed, and children who are unable to create a written method of their own will at least have a range of options from which to choose for their own use.

When should conventional algorithms be presented to students?

I believe that there is no place for formally introducing conventional algorithms to children in the first five years of school. If they arise during classroom problem solving (and they almost certainly will, given the input of parents and siblings into the process), they can be considered and discussed.

By giving arithmetic a problem solving focus, and by providing a whole range of problems for children to solve (preferably in story contexts of interest to children), we redefine the role of students, in the words of Lampert (1989), from the task of 'remembering what to do and in what order to do it, to a problem of figuring out why arithmetic rules make sense in the first place' (p. 34).

The cognitively guided instruction (CGI) problem types (e.g., Fennema, Carpenter, Franke, Levi, Jacobs & Empson, 1996) provide one basis for creating such story problems. My suggestion would be to use a variety of such problem types, with increasingly large numbers, challenging children to solve them, by any method that makes sense to them. Through sharing their methods, children can make a start on the process of evaluating various methods for their mathematical validity, their efficiency, and their generalisability, though not in these terms! In time, when they meet conventional algorithms (in upper primary, if at all), they will be in a strong position to compare the various possibilities on a fair basis, without feeling pressured to discard all that they have learned.

Concluding thought (from 1830)

> The learner should never be told directly how to perform any operation in arithmetic… Nothing gives scholars so much confidence in their own powers and stimulates them so much to use their own efforts as to allow them to pursue their own methods and to encourage them in them (Colburn, 1912, p. 463).

References

Board of Studies New South Wales (2002). *Mathematics K–6 Syllabus*. Sydney: Author.

Campbell, P. F., Rowan, T. E. & Suarez, A. (1998). What criteria for student-invented algorithms? In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics* (Yearbook of the National Council of Teachers of Mathematics, pp. 49–55). Reston, VA: NCTM.

Clarke, D. M. (2001). Understanding, assessing and developing young children's mathematical thinking: Research as a powerful tool for professional growth. In J. Bobis, B. Perry & M. Mitchelmore (Eds), *Numeracy and Beyond* (Proceedings of the 24th Annual Conference of the Mathematics Education Research Group of Australasia Vol. 1, pp. 9–26). Sydney: Mathematics Education Research Group of Australasia.

Colburn, W. (1970). Teaching of arithmetic. *Elementary School Journal, 12,* 463–480.

Fennema, E., Carpenter, T. P., Franke, M. L., Levi, L., Jacobs, V. R. & Empson, S. B. (1996). A longitudinal study of learning to use children's thinking in mathematics instruction. *Journal for Research in Mathematics Education, 27*(4), 403–434.

Groves, S. & Stacey, K. (1998). Calculators in primary mathematics: Exploring number before teaching algorithms. In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics* (Yearbook of the National Council of Teachers of Mathematics, pp. 120–129). Reston, VA: NCTM.

Hope, J. A. (1986). Mental calculation: Anachronism or basic skill? In H. Schoen & M. J. Zweng (Eds), *Estimation and Mental Computation* (Yearbook of the National Council of Teachers of Mathematics, pp. 45–54). Reston, VA: NCTM.

Kamii, C. & Dominick, A. (1998). The harmful effects of algorithms in grades 1–4. In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics* (Yearbook of the National Council of Teachers of Mathematics, pp. 130–140). Reston, VA: NCTM.

Lampert, M. (1989). Arithmetic as problem solving (Research into practice series). *Arithmetic Teacher, 36*(7), 34–36.

McIntosh, A. (1998). Teaching mental algorithms constructively. In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics* (Yearbook of the National Council of Teachers of Mathematics, pp. 44–48). Reston, VA: NCTM.

Narode, R., Board, J. & Davenport, L. (1993). Algorithms supplant understanding: Case studies of primary students' strategies for double-digit addition and subtraction. In J. R. Becker & B. J. Preece (Eds), *Proceedings of the Fifteenth Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (Vol. 1, pp. 254–260). San Jose, CA: Center for Mathematics and Computer Science Education, San Jose State University.

Northcote, M. & McIntosh, M. (1999). What mathematics do adults really do in everyday life? *Australian Primary Mathematics Classroom, 4*(1), 19–21.

Plunkett, S. (1979). Decomposition and all that rot. *Mathematics in School, 8,* 2-5.

Shuard, H. (1990). *Developing a Calculator-Aware Number Curriculum in Britain.* Unpublished manuscript, Homerton College, Cambridge, England.

Thompson, I. (1997). Mental and written algorithms: Can the gap be bridged? In I. Thompson (Ed.), *Teaching and Learning Early Number* (pp. 97–109). Buckingham, UK: Open University Press.

Usiskin, Z. (1998). Paper-and-pencil algorithms in a calculator and computer age. In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics* (Yearbook of the National Council of Teachers of Mathematics, pp. 7–20). Reston, VA: NCTM.