Mathematical thinking of undergraduate students when using three types of software

Conference or Workshop Item

MATHEMATICAL THINKING OF UNDERGRADUATE STUDENTS WHEN USING THREE TYPES OF SOFTWARE

Anesa Hosein, James Aczel, Doug Clow and John T.E. Richardson

The Open University

The research investigates how conceptual understanding of mathematics is promoted when using three types of software: black-box (no mathematical intermediate steps shown), glass-box (intermediate steps shown) and open-box (interaction at each intermediate step). Thirty-eight students were asked to think-aloud and give detailed explanations whilst answering three types of tasks: mechanical (mostly procedural), interpretive (mostly conceptual) and constructive (mixture of conceptual and procedural). The software types had no impact on how students answered the mechanical tasks; however students using the black-box did better on the constructive tasks because of their increased explorations. Students with low maths confidence resorted to using real-life explanations when answering tasks that were application related.

BACKGROUND AND THEORY

Mathematical software types such as in computer algebra systems/ graphical calculators are used as ubiquitous tools in the teaching of mathematics at the university level. Although, there are pedagogical advantages of using the software (see Heid & Edwards, 2001), Dana-Picard & Steiner (2004) explains that there are some drawbacks namely that students use trial-and-error strategies when solving problems and that the students are unable to see the mathematical intermediate steps because the software is a black-box (BB). Buchberger (1990) recommended that the black-box software should be replaced with glass-box (GB) software in which the steps are included as this should promote students understanding.

There are limited studies into determining whether students learning are better promoted by either the glass-box or black-box software. These studies tend to investigate procedural or mechanical knowledge, or student learning being mediated by a teacher (e.g. Horton, Storm, & Leonard, 2004). The current research looks at how three software types, black-box, glass-box and open-box (OB) help promote learning without the mediation of a teacher. The open-box software requires a decision/ interaction by the student for each step before the problem can be solved (see Figure 1).

The study hopes to delve further into which of the boxes can better promote conceptual knowledge. Horton et al. (2004) have previously found that the open-box helps in students' procedural knowledge when working problems by hand.

According Hiebert & Lefevre (1986) conceptual knowledge is considered to be knowledge that is "*rich in relationships*". Therefore, conceptual knowledge "*cannot* *be an isolated piece of knowledge*" but rather only begins to be "*part of conceptual knowledge only if the holder recognises its relationship to other pieces of information*". Further, the formation of conceptual knowledge occurs "*between two pieces of information that already have been sorted in memory or between an existing piece of knowledge and one that is newly learned*".

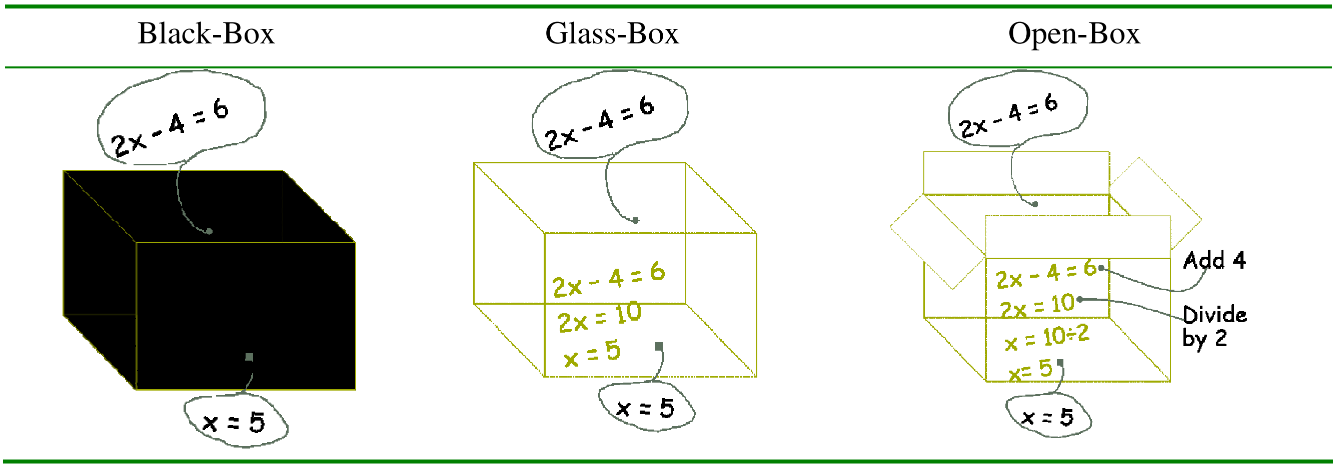

Figure 1: Comparison of an algebra solution by black-box, glass-box and open-box

On the other hand, procedural knowledge is discrete knowledge and there are two parts of procedural knowledge. The first part is "*composed of the formal language, or symbol representation system, of mathematics*". The second part of procedural knowledge is concerned with the algorithms or rules in particular the step-by-step instructions for completing the problem. One thing to infer from these definitions is that not necessarily all information acquired can be sorted into procedural and conceptual knowledge, instead this information is acquired through learning which may or may not consist of procedural or conceptual knowledge.

Hiebert & Lefevre (1986) also describes meaningful learning which is rich with relationships and thus, all conceptual knowledge "*must be learned meaningfully*". In rote learning relationships are absent but are "*tied closely to the context in which it is learned*" and as such knowledge acquired can only be "*accessed and applied only in those contexts that look very much like the origina*l".

To elicit how students connect their pieces of knowledge, the explanations that students make to themselves, that is, self-explanations can provide some insight. In terms of looking at the learning of mathematics and understanding how self-explanations can aid, Renkl (1997) has done an extensive study using 36 students. Renkl asked his students whilst studying problems to use the think-aloud strategy (Ericsson & Simon, 1984). The data was analysed to determine whether there were a) principle-based explanations: that is the mathematical principles of probability b) goal-operator combinations, c) anticipative reasoning, d) elaboration of the situation e) noticing coherence, f) monitoring negative and g) monitoring positive. He found for near-transfer problems (problems that required mostly procedural knowledge), that students' principle-based explanations and pre-test scores were significant predictors which implied that the students were not connecting with any knowledge beyond mathematical principles of which they previously knew. He further confirmed Chi, Bassok, Lewis, Reimann, & Glaser (1989)'s notion that the self-explanations were of major importance in determining whether there is transferable knowledge since in the medium transfer problem, only the self-explanations could explain the variance and not the pre-test values.

Further in other studies (e.g. Schworm & Renkl, 2006), self-explanations were not only applied to think-aloud strategies but also using the written explanations.

METHODOLOGY

For this study, the mathematical domain chosen was linear programming as this was a suitable topic that a wide cross-section of undergraduate students could attempt. Each software box was programmed within MS Excel using Visual Basic Applications (VBA) to perform the simplex algorithm. Students were required to work through three problems which each had three tasks that were based on the taxonomy by Galbraith & Haines (2000). These tasks were mechanical (requiring mostly procedural knowledge), interpretive (requiring mostly conceptual knowledge) or constructive (requiring a mixture of conceptual and procedural knowledge). Only the mechanical tasks absolutely required using the software although the constructive tasks could have been solved either by hand or software. These problems formed the post-test.

Each software box was stored as a separate MS Excel file which had four worksheets; one for a practice question and the remaining three worksheets related to each problem. An answer form was provided on each worksheet in which students were required to type in their answers.

Undergraduate students from the UK and Trinidad and Tobago were observed via web-conferencing remote observation (use of webcams, application sharing and voice/video conversation). A Latin-square experiment design was utilised where 12 students used for each software box type (36 students in all) whilst learning linear programming. Data for two additional students (one each for the glass and black box) were also collected when audio data was lost due to poor internet connection. Audio and screen-capture video data were recorded for all sessions.

The order in which the three problems were given to students was rotated to minimize any carry-over effects. Students were observed for one 2-hour session that followed the experimental procedure of Große & Renkl (2006) in which students were given a background questionnaire, a pretest, instructional materials related to linear programming, a practice question and a post-test. Additionally, the students had to complete an ‘approaches to study’ inventory. The background questionnaire required students to state their gender, degree, mathematics level and self-assess their confidence level for mathematics, and when using MS Excel and computers. The pre-test questionnaire was used to determine whether the mathematical-level skills varied between participants and 7 questions were answered based on algebra, inequalities and simultaneous equations as these were adjudged to be the closest to the linear programming.

During the post-test for approximately one hour the students were asked to think-aloud (Ericsson & Simon, 1984) as they worked through the three problems. Students were prompted to think-aloud by using phrases such as “Keep talking”, “What are you thinking?” and “What are you doing now?”. Whilst the first phrase pre-dominated the session and is recommended, it was necessary to use the latter two phrases when students were not forthcoming or were engrossed in working with the software or watching the screen. Students were not prompted to talk whilst inputting data or typing in their answers. For the constructive and interpretive tasks, students were asked to give detailed explanations. This was also used to ascertain the self-explanations they were making.

The data was coded into whether students were using mathematical principles (using any type of maths) or relating to real-life (using their own practical experience) to solve the problems. Further, the data was also coded according to Roy & Chi (2005) into deep (meaningful) and surface self-explanations. Deep explanations were considered where students made attempts to meaningful learning such as linking their previous knowledge. Surface explanations were considered where students were paraphrasing or repeating things that were already explicitly found. This paper presents the qualitative aspects of the research. In particular, the paper is concerned about how students used the software and answered the problems rather than using pen/paper as in previous research. Also, the paper looks at students’ written explanations based on the problems and the software types they used.

RESULTS AND DISCUSSION

Problems 1 and 2 were both application problems, that is they had some reflection on real-life. Problem 1 dealt with the manufacturing of toys, whilst problem 2 dealt with the manufacturing of furniture. Problem 3 was an abstract problem. Students were divided into groups (based on what appeared to be a bimodal distribution) determined by their assessment of having a high mathematics confidence (7-10 on a scale) or a low mathematics confidence (1-6). Students who had high maths confidence did better than students with low maths confidence in Problem 1 (2.3 vs 1.7) and Problem 3 (2.6 vs 1.9), but did equally well in Problem 2 (3.1 vs 3.1). Generally, students did significantly better in Problem 2 than in Problem 1 and 3.

Mechanical Tasks

All 38 students answered the mechanical part of the problems correctly, as this was procedural and required the clicking buttons. For the open-box, this was slightly more complicated as students had to decide on which pivot variable they had to choose (see Figure 2). Although, in most cases their initial conjectures, that is, choosing the variable that would yield the highest profit was true (timestamp: 52:06), they often got muddled when presented with the second iteration and were uncertain on how to proceed (lines 52:57 to 54:49). Further, these students using the open-box software where they had to interact with it were often concerned about whether they were doing the problem right and if they were getting the right answer (lines 55:27 to 56:51). The students using the black-box and glass-box software did not have these qualms. However, only a couple of students using the glass-box were interested in understanding the steps but their conjectures and explanations of the iterations did not correspond to the theory similarly to what was found for students using the open-box.

- 49:51: “Ok, let’s see what this one is talking about” read the instructional materials, “First of all, let’s solve it, ok, so we can do that”
- 50:18: Looks at the papers and the screen [….]
- 51:20: “Let’s see if I can pick which one to solve.” [….]
- 52:06: “Ok, I’m going with x having the biggest influence, so, I’m going to choose that as my pivot variable” chooses x and gets his first iteration
- 52:57: Hovers over the column x in the new iteration, hover over y and then t.
- 53:12: “The next one I’m going to do is t … which I think is … appears less often, which you want more of” Hovers over y
- 54:00: “Or I could just try them randomly until I get one”
- 54:05: “Try y” Chooses y. And gets that he cannot uses that variable
- 54:17: “Yeah, if I try to do x, y”
- 54:23: Chooses t, and gets a new iteration.
- 54:49: Hovers over the s1 column and then s2
- 55:00: “Let’s see if we got to do another one” Clicks iteration and gets the problem has been solved
- 55:27: “I’m just curious to see what would happen if I had chosen different ways around. So, I’m just going to see if there is a difference, to get rid of these two variables here” Hovers over the y column in the canonical form and t column in the first iteration
- 55:47: He clears all and chooses t as his first pivot variable
- 56:07: He chooses x as his second pivot variable
- 56:33: Find the best solution.
- 56:49: “I can’t really remember what I got the first time around”
- 56:51: “I remember y was zero and therefore the same again … maybe it doesn’t matter too much then but I already did it”

Figure 2: Think-aloud transcript of Participant 33 (M, Low Confidence) solving a mechanical task for Problem 2 using the open-box

Interpretive Tasks

Only two students used the software when answering any of the interpretive tasks. This was expected as these tasks required students to link their conceptual knowledge. The first student (Participant 32, F, OB, Low Confidence) was changing the number values without any idea how changing these values would elicit an answer. However, the second student (Participant 13, M, GB, High Confidence) used the software to test a conjecture/ explanation he was making to see if it was true. The interpretive task was related to Problem 2, where chairs, one of the manufactured furniture, were not produced. Students were asked to make conjectures why this occurred. Most students thought it was because the demand for chairs was low, as a limit was placed on the number of chairs that could be produced (the demand). Participant 13 although initially thought the same, tested the model where there was unlimited demand for chairs and found that this was not true and then made the correct conjecture. The following is his answer:

"*Chairs were not produced. This may be due to the fact that stools required fewer resources and the increased profit on chairs was not enough to outweigh the increased cost in resources. Removing the constraint on the demand did not affect the result*."

Constructive Tasks

For the constructive tasks, students using the black box were more likely to do exploration by testing out numbers. Whilst students explored when using the glass-box and open-box, it occurred more frequently with the black-box. For example, when exploring the constructive task in Problem 3, four students used the black-box to explore values for this task. Participant 9 tried 200 and 1000, Participant 7 used three numbers (100, 200, 300) and Participant 1 tested numbers 200, 1000, -100, -200, -1000 after first exploring what occurs through the removal of a constraint (see Figure 3). The remaining student, Participant 3, only tried one number (125) as he was confirming a calculation he had made.

- 48:34: She changes the u coefficient in the last constraint row to 0 and the RHS from 90 to 0 and performs the iteration.
- 49:23: After being prompted to talk: "Yeah I just had tried putting back the equations without the constraint without the u and gave the answer as … the best value for it to be 200 but I'm not sure if that is the maximum value, I'm still thinking about it". Proceeds to look at the papers she has and correspond with the screen.
- 51:24: Clicks ok to get rid of the answer sheet and changes the u coefficient in the last row to 1 and the RHS to 200 and do the iteration.
- 52:21: She changes the RHS of the last row to 1000 and do the iteration
- 52:31: She clicks input problem highlights the coefficient of u in the last row but then looks back at her papers
- 53:34: After being prompted to talk: "I'm trying something with …um … the input problem to see what different values the last constraint will give me."
- 53:49: Changes the RHS of the last row as -100 and the coefficient of the u as -1 […]
- 54:27: She re-changes the RHS of the last row to -200 and clicks iteration.
- 54:59: She changes the RHS of the last row to -1000 and clicks iteration […]
- 57:35: After being prompted: "Well I'm looking at the constraints to see if 200 is the highest value it can get and looking at Constraints A and B."

Figure 3: Think-aloud transcript for Participant 1 (F, BB, High Confidence) whilst doing a constructive task for Problem 3

For the open-box, only two students explored and they tested one number each (Participant 38: ‘100’ and Participant 32: ‘91’). For the two students using the glass-box, Participant 22 tested two numbers (95 and 100) and Participant 15 tried four numbers (100, 105, 120, 170). All four students using the glass-box and open-box made the wrong conjecture in the end. Participant 15 who tested the most numbers eventually gave his answer as being “*I don’t know*”. Students needed to test numbers above 200 to make the correct conjecture.

What is interesting here is that only students in the black-box were compelled to test a large range of numbers and hence find the correct answer. The students using the other software seemed to limit their exploration and this may be a factor of the high cognitive load required to process the intermediate information that is being presented to them.

Explanations

Students who explained their answers based mostly on real-life explanations often resulted in the wrong answer. For example in the constructive task in Problem 1, students were asked “If the profit increased for toy trains by £1, how this will change what the toy company manufacture?”. Although, the demand on toy trains had already been filled, students still suggested that the toy company would manufacture more toy trains and reduce the production of toy soldiers (the other product). Not surprisingly, students who either explored with the software or used mathematical principles were significantly more likely to get the answer correct (see Figure 4).

Participant 4 (F, BB, High Confidence): *Want to produce more trains if the profit is increased.* (Real-Life Explanation)
Participant 9 (F, BB, High Confidence): *profit would increase to 140 but the numbers of toys made stays the same because constraints is that x =40 maximum so even though they get more profit they cant make any more trains.* (Mathematical Explanation resulting from an exploration)
Participant 13 (M, GB, High Confidence): *If the profit per train is increased, it would likely be more profitable to produce more trains and fewer soldiers. However Constraint C puts an upper bound on the number of trains that can be produced -- a bound which has already been achieved. Hence it is not possible to produce more trains and the number of toy trains and toy soldiers produced would remain the same. This was confirmed by solving the modified problem.* (A Mathematical Explanation confirmed with the testing of the software)

Figure 4: Types of explanations provided by students for a constructive problem

In some cases, students who explored using the software, although initially may have had a real-life explanation, they soon sought to find a mathematical explanation when their real-life conjecture was not confirmed. Further, some students when reading and trying to understand Problem 3 (abstract problem) wanted to relate the abstract problem to something real-life in order to explain it. For example:

> *"Which variable will I not want to have the highest value for … cause I don't know the variable, so I am thinking that the variable could be anything and it will be related to specific problems and depending on what application you want to use it in so I guessing that these variables they could differ according to the problem"* (Participant 15, M, GB, Low Confidence).

Further, it was noted for the application problems (Problems 1 and 2), students with the low maths' confidence were more likely to make real-life applications than students with the high maths' confidence. Further, students with higher maths' confidence tended to rely slightly more on mathematical explanations. Frequently, the mathematical explanations were linked to a more deep approach to learning whilst the real-life approach tended to be linked to surface learning approach.

CONCLUDING REMARKS

Students who were not certain of answers resorted to real-life explanations which often were surface explanations, that is, it was conceptually linked to their experiences. Mathematical explanations were more likely to be made by more mathematically confident students because they were more likely to use explorations and make conceptual linkages.

Further, if a problem had a real-life application, some students would try and understand the problem from the real-life perspective whilst ignoring the underlying mathematics. However, if software is used for exploration, the students can think and determine how the mathematical explanations may fit in or use it to confirm or reject their conjectures. The black-box software seemed the best choice for this in regards to promoting conceptual knowledge. Whilst the open and glass-box, to a lesser extent, helped the student to make these connections, these software types are probably best in promoting the procedural knowledge and understanding of how things can be calculated.

REFERENCES

Buchberger, B. (1990). Should students learn integration rules? *ACM SIGSAM Bulletin, 24*(1), 10-17.

Chi, M. T. H. C., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: how students study and use examples in learning to solve problems. *Cognitive Science, 13*(2), 145-182.

Dana-Picard, T., & Steiner, J. (2004). The importance of 'low-level' CAS commands in teaching engineering mathematics. *European Journal of Engineering Education, 29*(1), 139-146.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: MIT Press.

Galbraith, P., & Haines, C. (2000). Conceptual mis(understandings) of beginning undergraduates. *International Journal of Mathematical Education in Science and Technology, 31*(5), 651-678.

Große, C. S., & Renkl, A. (2006). Effects of multiple solution methods in mathematics learning. *Learning and Instruction, 16*(2), 122-138.

Heid, M. K., & Edwards, M. T. (2001). Computer algebra systems: revolution or retrofit for today's mathematics classrooms? *Theory Into Practice, 40*(2), 128-136.

Hiebert, J., & Lefevre, P. (1986). Conceptual and procedural knowledge in mathematics: an introductory analysis. In J. Hiebert (Ed.), *Conceptual and Procedural Knowledge: The Case of Mathematics* (pp. 1-27). Hillside, NJ: Lawrence Erlbaum Associates Inc.

Horton, R. M., Storm, J., & Leonard, W. H. (2004). The graphing calculator as an aid to teaching algebra. *Contemporary Issues in Technology and Teacher Education, 4*(2), 152-162.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science, 21*(1), 1-29.

Roy, M., & Chi, M. T. H. C. (2005). The self-explanation principle in multimedia learning. In R. E. Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning* (pp. 271-286). New York: Cambridge University Press.

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: when instructional explanations reduce self-explanations. *Computers and Education, 46*(4), 426-445.