

Spreadsheets: A Technological Nexus between Big Ideas in Mathematics and Computational Thinking in Mathematics Classrooms

Weng Kin HO

wengkin.ho@nie.edu.sg

National Institute of Education
Mathematics and Mathematics Education
Nanyang Technological University
Singapore

Jonaki GHOSH

jonakibghosh@gmail.com

Department of Elementary Education
Lady Shri Ram College for Women
University of Delhi
Delhi, India

Berinderjeet KAUR

berinderjeet.kau@nie.edu.sg

National Institute of Education
Mathematics and Mathematics Education
Nanyang Technological University
Singapore

Chee Kit LOOI

cheekit.looi@nie.edu.sg

National Institute of Education
Mathematics and Mathematics Education
Nanyang Technological University
Singapore

Abstract: *Recent years in Mathematics Education research saw the emergence of ‘Big Ideas in Mathematics’ and ‘Computational Thinking’ as two areas of intense research. This paper highlights the technological and pedagogical affordances of electronic spreadsheets, which make them a convenient and practical technological nexus between ‘Big Ideas in Mathematics’ and ‘Computational Thinking’ in a Mathematics classroom. To demonstrate how such a connection is realized in authentic classroom settings, some instructional exemplars that have been designed and deployed in Asian (senior) high schools are provided here. Our present approach for generating such mathematics lessons may be considered as one of the ways for mathematics teachers to transform their understanding of big ideas of mathematics into tasks and lessons in the classroom.*

1. Introduction: Big Ideas of Mathematics and Computational Thinking

In recent years of Mathematics Education research, two areas of intense research started to emerge, namely, ‘Big Ideas in Mathematics’ and ‘Computational Thinking’. As defined in Charles (2005):

A big idea is a statement of an idea that is central to the learning of mathematics, one that links numerous mathematical understandings into a coherent whole. (p. 10)

Several countries have begun to emphasize not only a need for their teachers “to understand big ideas of mathematics and be able to represent mathematics as a coherent and connected enterprise” (National Council of Teachers of Mathematics 2000, p. 17), but also for students of this age to understand “the connections of mathematical concepts and their application in their world as a motivation to learn” (Australian Curriculum Assessment and Reporting Authority 2013). For instance, in Singapore the big ideas in mathematics are articulated explicitly in the 2020 Mathematics document (Ministry of Education, 2018) as a set of eight big ideas that are very much aligned with Charles’s (2005) definition: Big Ideas about (1) Equivalence, (2) Proportionality, (3) Invariance, (4) Measures, (5) Functions, (6) Notations, (7) Diagrams, and (8) Models. It is envisioned that teachers, by organizing their teaching around these big ideas, may be able to focus their efforts on supporting their students to make connection among these ideas – this very ability

of making connections has earlier been identified as a positive indicator for students' understanding (Hiebert and Carpenter, 1992).

Concurrently, computational thinking has repeatedly been featured in mathematics curricula and teaching standards worldwide. For example, the new PISA 2021 draft spells out computational thinking skills as an essential skill set which is expected to be mastered by the 21st century student. Cognizant of the growing role of technology in students' lives, PISA 2021 also acknowledges that the "long-term trajectory of mathematical literacy should also encompass the synergistic and reciprocal relationship between *mathematical thinking* and *computational thinking*" (OECD, p.7). While Papert's (1980) original use of 'Computational Thinking' stressed on computational representations for highlighting and accessing powerful ideas, Wing's (2006) latter popularization of this term focused on the attributes of this paradigm, i.e., decomposition, pattern recognition, abstraction and algorithm design. Notably, Ho et al (2019) used those Computational Thinking Practices first proposed by Hoyles and Noss (2015) and Tabesh (2017), i.e., Pattern recognition (seeing a new problem as related to previous problems), Abstraction (seeing a problem at diverse d levels of detail), Decomposition (solving a problem comprises solving a set of smaller problems), and Algorithm design (seeing tasks as smaller associated discrete steps), as operational labels in lesson designs that incorporate computational thinking into mathematics teaching and learning. Recently, practitioners have broadened the meaning of 'Computational Thinking'. Sanford (2018), describes it as "a partnership of human mind and digital machinery" (p. 104); thus implying that one of the goals of education would be for students to think of digital aid as "just part of a natural process as common as using a pencil or a reference book" (p. 104). See Chan et al (2022) for a thorough and recent survey on the tools and approaches for integrating computational thinking and mathematics.

Although it is intuitive that there should be some linkages between the applications of 'Big Ideas in Mathematics' and of 'Computational Thinking' to Mathematics Education, a thorough search reveals the paucity of literature lying in their intersection. Encouraged by the theme 'Connecting All Applications to Mathematics and Technology' of the 27th ATCM, the authors propose a nexus between 'Big Ideas in Mathematics' and 'Computational Thinking' by appealing to the technological and pedagogical affordances of electronic spreadsheets (*Microsoft Excel* being a common prototype). Such a connection would enable instructors of mathematics to design and deploy mathematics lessons which exploit electronic spreadsheets as the vehicle to achieve simultaneously the educational goals of 'Big Ideas in Mathematics' and 'Computational Thinking'. The purpose of this opinion paper is to persuade our readers that one possible and practical link between 'Big Ideas in Mathematics' and 'Computational Thinking' – too seemingly disjoint topics in Mathematics Education research – can be made through the technology of electronic spreadsheets. Apart from building this theoretical bridge, we are motivated by practical concerns; one of which is aptly articulated by Choy (2019) – "besides grappling with the definitions of big ideas, teachers also have to deal with the challenges of transforming their understanding of the big ideas into tasks or lessons" (p. 107). Our theoretical consideration gives rise to practical ways of generating spreadsheet-based lessons to tackle such practice-related challenges.

This paper is organized as follows. In Section 1, we motivate our scholarly discussion. Section 2 expounds on those technological affordances of electronic spreadsheets, which connect with certain big ideas in mathematics. Section 3 translates these connections into actual classroom implementation revealing the pedagogical affordances of spreadsheets. We adapt a task design schema developed by Ho et al (2021) that teachers can use to design lessons and tasks aimed at invoking big ideas in mathematics and computational thinking. Using this schema, we construct two instructional exemplars: *Number Patterns Problem* and *Spaghetti Problem* which have been

designed and deployed in authentic school settings of the authors' respective countries. Lastly, we make some concluding remarks in Section 4. In this paper, we shall use *Microsoft Excel* as the illustrating software for electronic spreadsheets due to ease of access by teachers and students in most countries; whence, we waste no time to explain basic MS Excel terminology. However, the instructional exemplars may be deployed using other spreadsheets such as Google Sheets.

2. Technological affordances of spreadsheets and big ideas in mathematics

2.1 Numbers, The Base Ten Numeration System, Equivalence, Relations and Functions

Charles (2005) singles out *Numbers* as the Big Idea #1 which focuses on the infinitude of real numbers, each corresponding to a unique point on the number line (p. 12). We are reminded of the set of counting numbers (1, 2, 3, ...), that of whole numbers (0, 1, 2, 3, ...), and two further extensions, namely, that of integers and rational numbers. Thus, a minimum requirement for any computing device is its ability to store and calculate with these numbers. Other than this basic capability, MS Excel can assign formats (number, currency, date, time, percentage, fraction, scientific, text) to any number located in a cell (see Figure 1 (left) for various formats of 0.14).

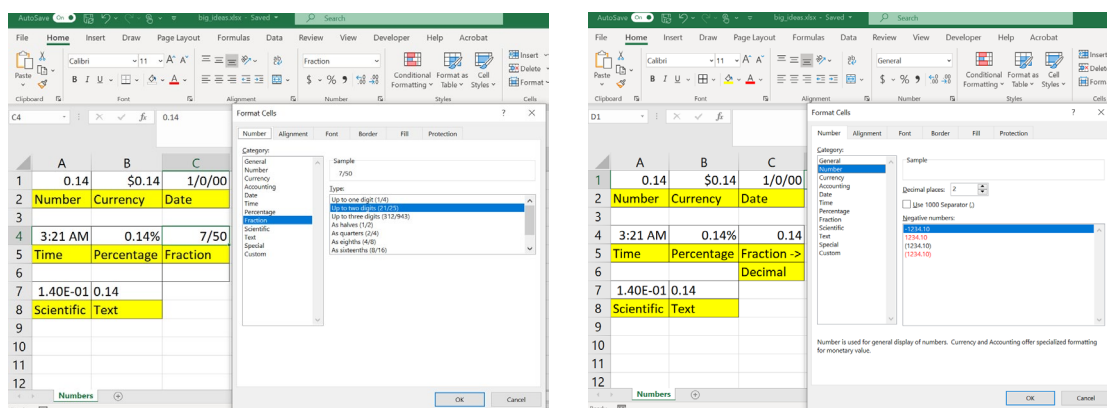


Figure 1. The number '0.14' in various formats

A student's entry of 14/100 (fraction format) into C4 was automatically expressed as 7/50; this automated conversion presents her a chance to encounter *Equivalence* (Big Idea #3) – the idea that “any number ... can be represented in an infinite number of ways that have the same value” (Charles 2005, p. 14). Changing the format of C4 from fraction to number (specified to 2 decimal place) converts the fraction 7/50 back to 0.14, affirming her that 7/50 is equivalent to 0.14. Not all the formats of 0.14 are equivalent: for example, 1.4E-01 (A7), being the standard form notation 1.4×10^{-1} of 0.14 in Excel, is equivalent to 0.14 but 0.14% (B4) is not. Exploring number formats reminds the learner that a real number can be corrected to a specified accuracy of decimal places, thus reinforcing one's understanding of Big Idea # 2 *Base Ten Numeration System* which is summarized by Charles (2005) as the “base-ten numeration system extends infinitely to very large and very small numbers (e.g., millions & millionths)” (p. 14). The opportunistic teacher may also heighten his students' curiosity by asking why A4 reads as “3:21 AM”. Finally, when the student replaced “0.14” by another number in A1, entries in B1, C1, A4, B4, C4, A7, B7 and C7 were automatically updated. This phenomenon teases out *Relations & Functions* (Big Idea #12), which highlights that “mathematical rules (relations) can be used to assign members of one set to members of another set” (p. 18, Charles, 2005), accounting for dependencies among cells.

This simple number-format facility of Excel (when exploited in the lesson) offers several opportunities to engage learners in mathematical thinking, leveraging on certain big ideas in mathematics. Excel has two important features. Firstly, Excel facilitates *computation* as can be seen from its ability to express “0.14” in various formats and to create functional relations between cells. Secondly, Excel helps learners *forge their big ideas in mathematics* through the creation of a meaningful product (yes, the spreadsheet!). Forging big ideas involves ‘acting out’ mathematical thoughts through treatment, transformation, organization, presentation and analysis of relevant data within some given contexts. Reminiscent of the “Act-It-Out” heuristic on Mathematical Problem Solving (Toh et al, 2011), this second aspect opens a door to using “computational thinking to forge ideas” (Papert, 1980) and creates an environment to improve one’s learning through the act of “constructing a meaningful product” – the core of Papert’s constructionism (Papert, 1996). In short, electronic spreadsheets possess simultaneously the technological affordance of computation and the pedagogical affordance to forge big ideas in mathematics that, through *intentional instructional design*, can help students learn mathematics. We now dig deeper by pairing up those specific technological capabilities of the Excel spreadsheets with their corresponding big ideas in mathematics.

2.2 Variable, Equations and Inequalities

Letters are used in mathematics to represent unknown in equations and form algebraic expressions. This spells out Big Idea #10 *Variables* (p. 18, Charles, 2005). Together with the concept of function in Big Idea #12 *Relations & Functions*, variables are used to build mathematical relationships – the value of one quantity depends on the value of the other quantity. In Excel, one can define the content of a cell in terms of others, treating them as variables. Take for example the equation $y = x^2$ that connects the variables x and y , where x ranges over $\{-3, -2, -1, 0, 1, 2, 3\}$. In Excel, a table of values is drawn up as in Figure 2 to demonstrate how the variable y is written as a function of the variable x , simply by allocating range **A2:A8** of cells headed by “ x ” as instances of x and the corresponding range **B2:B8** headed by “ y ” as instances of y . The relationship between the cells in these ranges is typified by that of the first pair of cells, namely **A2** and **B2**, where **B2** is assigned the value obtained by squaring the value found in **A2**, i.e., **B2=A2^2** (Figure 2, left). In fact, we see that Big Idea #13 *Equations (and Inequalities)* (p. 19, Charles, 2005) comes into the picture here as an equation has been set up to define the assignment of values to variables, where variables are denoted by cell references in Excel.

Excel has a powerful tool called the fill handle – appearing as a small square in the bottom-right corner of the selected cells (see Figure 2, right). By clicking, holding and dragging the fill handle until all of the cells one wants to fill are selected, the same function applies to selected series of cells “but transposes the cells references as one fills up, down or across” (p. 7, Marley-Pane and Dituri, 2019).

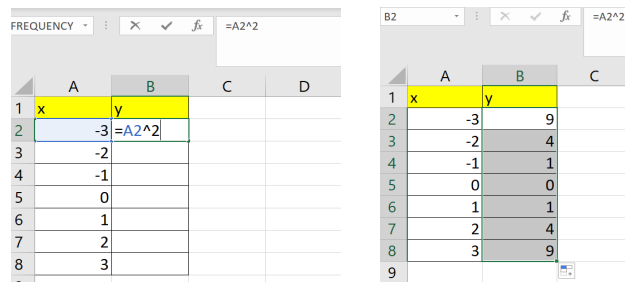


Figure 2. Assignment of values in cells using variables, equations and fill handle

For example, after dragging the fill-handle through the range **B2:B8**, all the cells in this range would be designated the square of the values entered in the corresponding cells in the range **A2:A8**; in particular, **B5=A5^2**. The workings and power of the fill function are discussed in Baker and Baker (2010), and why this feature makes spreadsheets a technology well-suited to mathematics education has been expounded by several authors (Marley-Pane and Dituri, 2019; Abramovich and Brantlinger, 1998). Hvorecky and Trencansky (1998) explained that the fill function essentially characterizes the manipulation with spreadsheet values in three steps: “(1) The *initialization* of (a relatively small number of) cells, (2) the *definitions of relationships* between initialized cells and selected ones, and (3) *spreading* of the relationship through the sheet” – whence, “spreadsheet”.

2.3 Comparison, Inequalities, Estimation

Excel can carry out Boolean-valued comparisons between two numbers using the inequalities “<” (less than), “>” (greater than), “<=” (less than or equal to) and “>=” (greater than or equal to). For example, in Figure 3, various comparisons have been made between the values of “x” and “y” and presented in the corresponding columns; for example, **C2=(A2<B2)**. This capability allows the student to realize Big Idea #4 *Comparison*, where numbers, expressions, and measures can be compared by their relative values (p. 14, Charles, 2005). Estimation by rounding up (respectively, down), to nearest multiple of a given number, in Excel is possible via the floor function (respectively, ceiling function); for example, **G2=FLOOR(A2,1)** rounds down the value in **A2** to the nearest integer and **H2 = CEILING(A2,0.01)** rounds up the value in **A2** to the nearest multiple of 0.01 (see Figure 3). At this juncture, the teacher can challenge his students to devise a simple formula for correcting a number to the nearest integer; e.g., 2.4 is corrected to 2 while 2.6 to 3.

	A	B	C	D	E	F	G	H
1	x	y	x < y	x > y	x <= y	x >= y	FLOOR(x,1)	CEILING(x,0.01)
2	0.521	0.5	= (A2<B2)	TRUE	FALSE	TRUE	0	0.53
3	0.612	1	TRUE	FALSE	TRUE	FALSE	0	0.62
4	1.301	0.8	FALSE	TRUE	FALSE	TRUE	1	1.31

Figure 3. Comparison and estimation in Excel

2.4 Data Representation and Data Distribution

Excel can display visual representations, e.g., charts, box-and-whisker plots, scatter plots, and graphs. Figure 4 (left) shows a histogram obtained by highlighting the frequency table in the range **B2:C7**. Figure 4 (right) shows how a regression line (with its equation) and r^2 (the square of the Pearson’s linear correlation coefficient) can be calculated for a bivariate data set (ranging over **A2:B7**), together with its scatter plot. This technological affordance of the spreadsheet relates to Big Idea #19 *Data Representation* (Charles, 2005), where data can be represented visually using tables (an intrinsic advantage of spreadsheets), charts, and graphs (p. 21).

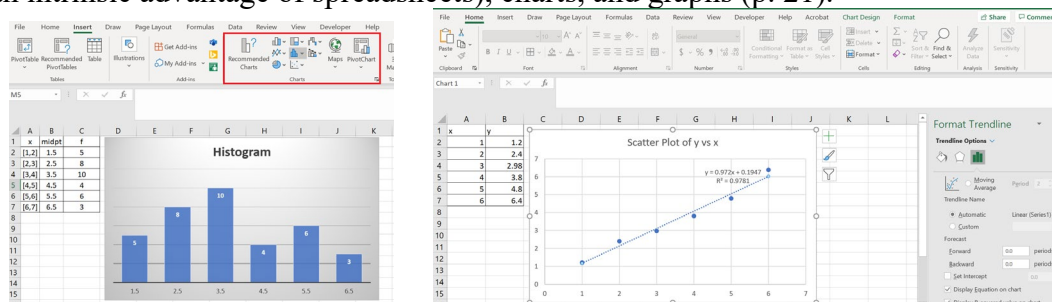


Figure 4. Visual representations (histogram, scatter plot) of data in Excel

Additionally, Excel has inbuilt functions to calculate mean, mode and median. Figure 5 shows how these three measures of central tendency for the data set {1.1, 1.1, 1.5, 1.6, 2.1, 2.1, 2.1, 3.4} can be calculated: **B3 = AVERAGE(B1:B11)**, **B5 = MODE(B1:B11)**, and **B7 = MEDIAN(B1:B11)**. It also can calculate the standard deviation: **E3 = STDEV(B1:B11)**. This feature of Excel can be paired with Big Idea #20 *Data Distribution*, where there are special numerical measures that describe the center and spread of numerical data sets (p. 21, Charles, 2005).

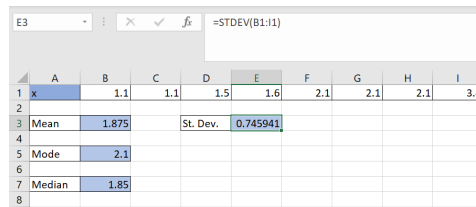


Figure 5. Inbuilt functions to calculate Mean, Mode, Median and Standard deviation

2.5 Spreadsheet features as building blocks that enable computational thinking practices

The simplicity and power of the above Excel features – like Lego© blocks of big ideas in mathematics – enable one to build complex instructional tools, for specific learning purposes, out of these ‘atomic’ technological affordances. In this way, we see that the aforementioned spreadsheet features provide a sandbox for students to practice their computational thinking. Spreadsheets thus become a practical way for both teachers and students to harness the symbiotic power of big ideas in mathematics and computational thinking.

3. Pedagogical affordances of spreadsheets and some instructional exemplars

We now address Choy’s (2019) problem: “teachers ... have to deal with the challenges of transforming their understanding of the big ideas into tasks or lessons” (p. 107).

3.1 Task design schema

Task design schemas can create cognitive shortcut for the task designer so that the specific task design that leverages on computational thinking can be stored and retrieved from the designer’s long term memory efficiently (Ho et al, 2019). Here we propose a modified four-stage schema suited for our current purpose (Figure 6).

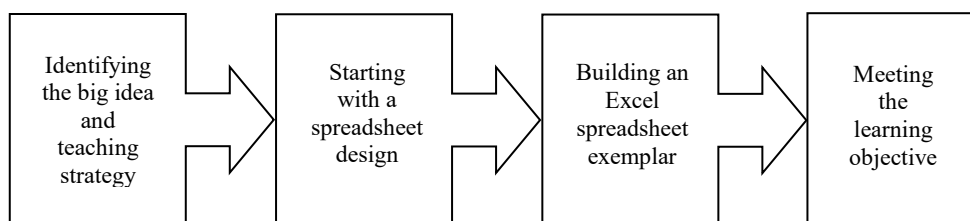


Figure 6. Task design schema: Leveraging big ideas in mathematics and computational thinking

Identifying the big idea and teaching strategy. When planning an instructional package to teach a certain topic, the instructor naturally examines the entire topic to identify the big idea in the topic. The instructor then decides on an appropriate instructional strategy for the lesson or task.

Starting with a spreadsheet design. The lesson or task will involve certain data to be organized in tables, i.e., certain identified fields with appropriate headers. What are the underlying

relationships between cells in this spreadsheet? Which visual representation of the data is appropriate for the lesson or task to bring out the big idea in an impactful manner?

Building an Excel spreadsheet exemplar. Which technological affordances of spreadsheets can be exploited in connection to the identified big idea in mathematics? The Excel spreadsheet exemplar will be the meaningful product to be constructed by the students or used by the teacher. In what way is this a meaningful product? What role does the spreadsheet play in lesson delivery?

Meeting the learning objective. Does the Excel exemplar engage the students in deepening the understanding of the relevant concepts in the topic? At this stage, the instructor reflects on the design of the lesson and task in connection with the manufactured exemplar as well as whether the lesson meets the intended learning objective.

3.2 Two instructional exemplars

In this subsection, we describe two instructional exemplars that made essential use of Excel spreadsheets. Both the lessons have been implemented over two 90-minute sessions in schools from two different countries, namely, Singapore (*Number Pattern Problem*) and India (*Spaghetti Problem*). In our description we use the task design schema for designing the lesson, leaving out issues of efficacy (see Chan et al (2021) for discussion of the efficacy of the first implementation).

3.2.1 Number Pattern Problem: Learning algebra by Grade 8 students

Identifying the big idea and teaching strategy. In this lesson, we harness on the link between Big Idea #9 *Patterns* and Computational Thinking (focusing on Pattern Recognition and Abstraction). One of the subtopics in Number Topics (Grade 8, Singapore) involves writing down an algebraic expression that represents the n th term of a number pattern sequence (Ministry of Education 2018). Typically, a diagram showing the first three patterns (of dots) in a sequence is given (see Figure 7). The students are asked to recognize the progression of number pattern by drawing out the next few patterns, e.g., Pattern 4. The learner is to write down the general formula for the number of dots found in Pattern n , in terms of n . This learning objective is aligned with the big idea statement for *Patterns* that “relationships can be described, and generalization made for mathematical situations that have numbers or objects that repeat in predictable ways” (p. 17, Charles, 2005).

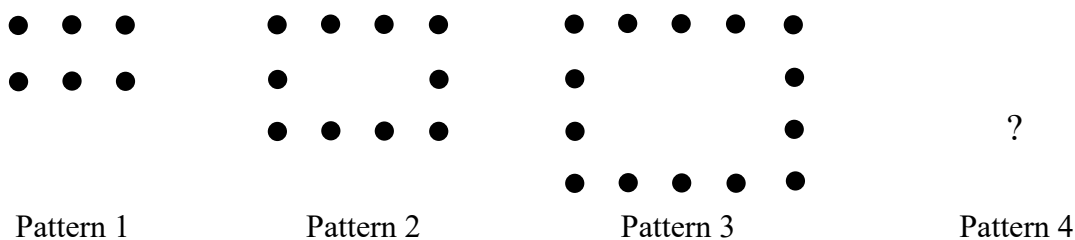


Figure 7. A number pattern sequence.

Starting with a spreadsheet design. In a typical lesson, students will be asked to complete a table as shown in Table 1.

Table 1. Number of dots in the n th number pattern

Pattern number n	Number of dots T_n
1	
2	
3	
4	

This tabular form suggests spreadsheets are an ideal technological nexus between big ideas in mathematics and computational thinking. Here we deviate from the usual lesson task. Our lesson

design is aimed to help an experimental group of seventy Grade 8 students from a Singapore secondary school deepen their understanding of number patterns in two ways. Firstly, recursive relationships provide an algorithmic way to compute accurate numerical solution to this problem, even if the closed formula is not observed by the students. Secondly, the graphical features of the spreadsheet can be employed to analyze the relationship between n and T_n and even predict T_n .

A spreadsheet approach is ideal for the purpose of teaching number patterns via recursion as advised by Hvorecky and Trencansky (1998). Here we adopt Papert's constructionist approach, i.e., the students learn the big idea of Patterns by using the technological affordances of fill form in Excel spreadsheets. In our ensuing description of the instructional design of this spreadsheet exemplar, we make use of the labels for Computational Thinking (CT) Practices, denoted by [Pattern recognition], [Abstraction], [Decomposition], [Algorithm design], to operationalize the definition of Computational Thinking in conveying Big Idea #9 Patterns.

Lesson induction [Pattern recognition] The students worked in pairs and set up a driver-navigator role, i.e., the driver is the one who will work on the spreadsheet, e.g., typing the codes in the cells, while the navigator will give the needed instructions and directions for the driver. Augmenting to the spreadsheet, students are also given physical hands-on worksheet to complete certain tasks on pen-and-paper form. The lesson began with showing a video about patterns and how they are ubiquitous in real world. Students began to recall prior knowledge about number patterns (acquired in their primary school mathematics).

Development (I) [Algorithm Design] Students were explicitly taught the concepts of CT, which is a way of thinking and solving problems through the use of the computers. This was intended to give an experiential emphasis on the partnership between the human and the digital technology in problem solving (Sanford, 2018). Here the lesson objective was made clear to the students, i.e., (a) learn to generate number patterns using spreadsheets, and (b) construct the general term of a given number pattern.

[Pattern recognition & Algorithm design] The teacher demonstrated how Excel spreadsheets work by introducing spreadsheet as a large array (or matrix) of boxes called cells, i.e., in the same spirit as (Hvorecky and Trencansky, 1998). Each cell was a place to store a datum and is referenced using its coordinates – a concept learnt in mathematics and also one which ties in with Big Idea #15 *Orientation and Location* (articulating that objects in space are oriented in an infinite number of ways, and an object's location in space can be described quantitatively).

Building an Excel spreadsheet exemplar. Instructions are then given for the students to key in words in cells, e.g., heading "Pattern number (n)" in **A1**, and the number '1' in **A2**. Crucially, to generate the list of consecutive counting numbers 1 to 10, the teacher demonstrated the use of the fill handle by transposing the formula in **A3 = A2 + 1** to the rest of the cells in range **A2:A11**. The students are then taught the recursion method for number patterns; precisely, in this case, the counting numbers are generated recursively by the rule of successor, i.e., $a_{n+1} = a_n + 1$, $a_1 = 1$.

Development (II) In this part of the lesson development, the students are to focus on the CT Practice of Algorithm design. The end goal is for them to create an algorithm that calculates the n th term of the number pattern sequence based on the questions crafted on a physical worksheet, the first part of which comprises the number patterns shown in Figure 7, and the table to be completed as shown in Table 1. The important difference here is that they complete the table by partnering with the computer, i.e., they do not carry out the menial task of computation by their human effort. In order that the students knew the end goal, i.e., the final useful product – the Excel exemplar, they would be shown what this final product looked like on the screen (Figure 8).

	A	B	C
1	Pattern Number (n)	Number of dots (T_n)	
2	1	6	
3	2	10	
4	3	14	
5	4	18	
6	5	22	
7	6	26	
8	7	30	
9	8	34	
10	9	38	
11	10	42	

Figure 8. Final product of the Excel exemplar

The students are guided to apply the CT practice of Decomposition, addressing sub-tasks: (1) What is the starting number (which cell and what to initialize)? (2) How to use the recursive method to generate the number pattern? What is the algebraic formula for the recursive rule? Using the answers to (1) and (2), students used their spreadsheet to find the 10th and 100th term.

Development (III) When the number n gets larger, the recursive method will be troublesome to use, even though we have a spreadsheet for it. Thus a (closed) formula in terms of n is a direct and faster method. To find such a direct relationship between n and T_n the students drew a scatter plot as a graphical representation to show the underlying pattern (Figure 9).

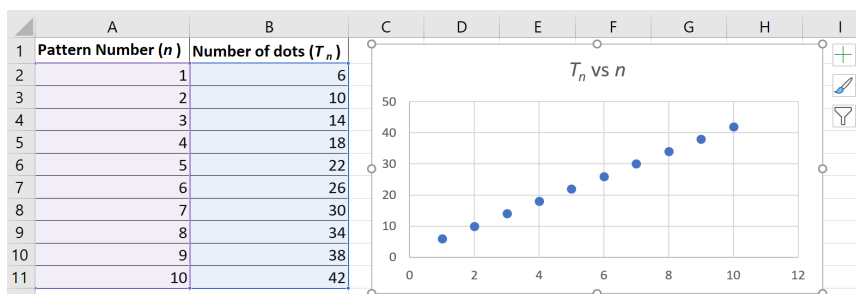


Figure 9: Scatter plot used to reveal the algebraic relationship between T_n and n

[Pattern recognition & Abstraction] The students invoke Big Idea #9 *Patterns* (via multiple representations -pictorial, tabular, graphical) to observe a linear relationship between the two variables (a straight line passes through all the points in the scatter plot). The teacher then helped students transit from the concrete case for $n = 10$ (Figure 10, left) to the abstract one for a general n (Figure 10, right).

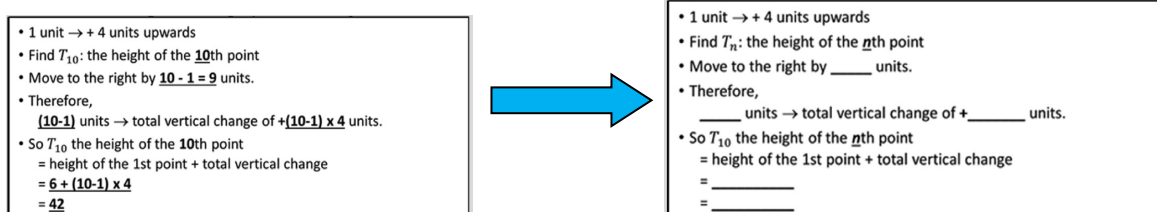


Figure 10: Engaging the CT Practice of Abstraction

After the students managed to derive the equation $T_n = 6 + (n - 1) \times 4 = 4n + 2$ they used the spreadsheet (see Figure 11) to create a new column C for the number of dots T_n that is generated by the closed formula, say **C2** = **4*n+2**. The students then check that the values of T_n derived recursively (**B2:B11**) – in the first part of the lesson – and those directly using the closed formula they obtained (**C2:C11**) are equal (see the “True” values appearing in **D2:D11** as a result of comparing for equality, e.g., **D2** = (**B2** = **C2**), ..., **D11** = (**B11** = **C11**)). Even if students do not

manage to derive T_n initially, they can test different formulae for column C, and check the results. This enables students to experience yet another key practice in computational thinking, namely testing and debugging. The derived formula was then applied to find T_{100} .

	A	B	C	D
1	Pattern Number (n)	Number of dots (T_n) [recursive method]	Number of dots (T_n) [direct method]	B = C?
2	1	6	6	TRUE
3	2	10	10	TRUE
4	3	14	14	TRUE
5	4	18	18	TRUE
6	5	22	22	TRUE
7	6	26	26	TRUE
8	7	30	30	TRUE
9	8	34	34	TRUE
10	9	38	38	TRUE
11	10	42	42	TRUE

Figure 11. Comparing the two different methods of calculating T_n

Meeting the Learning Objective.

Consolidation: The students are guided to realize that a number sequence, such as T_n , may be defined by recursion and hence calculated recursively (say, using the spreadsheet) or by a closed formula directly in terms of n . In the event a closed formula is unavailable, recursion provides an easy and fast numerical solution of the problem by exploiting computational thinking and the big idea of Patterns in the environment of spreadsheets.

3.2.2 Spaghetti Problem: An exploration of Probability by Grade 11 students

Chance (Big Idea #21) is an important idea in mathematics. Charles (2005) states that “the chance of an event occurring can be described numerically by a number between 0 and 1 inclusive and used to make predictions about other events” (p. 21). Such a number is referred to as probability of the occurrence of an event. To demonstrate this big idea, it is of paramount importance that the mediating technology – electronic spreadsheets – has the facility for simulating random events to which we may assign their probabilities. One of the powerful features of Excel is its ability to generate random numbers, which enables the user to simulate problems based on chance.

Identifying the big idea and teaching strategy. This can be very useful for students at the senior secondary school level who study the topic of probability as a part of the curriculum. Most of the problems in the textbooks relate to tossing of coins, rolling of dice or selecting cards from a standard deck. These seem a bit contrived; thus, not providing sufficient motivation for students to learn. Interesting problems centered on probability abound and can be used to enliven the probability class. In this subsection, we describe how twenty-six grade 11 students of a school in New Delhi explored the *Spaghetti Problem*. We shall illustrate the pedagogical approach adopted by the teacher in facilitating students’ explorations by first helping them to mathematize the problem and then simulate it using Excel. The exploration through guided discovery (Bruner, 1961) was completed in two 90-minute sessions. The following problem was posed:

Spaghetti Problem. *A spaghetti stick of length L is randomly broken into three pieces. What is the probability that the three pieces will form a triangle?*

Lesson induction: Prior to the activity each student was asked to bring a few spaghetti sticks to the class. At the beginning of the first session, they tried to break each stick into three pieces and arrange them to form a triangle. A few cases resulted in triangles while others did not. After a few trials many students conjectured that the probability of a triangle being formed is $\frac{1}{2}$.

Development (I) [Decomposition] The teacher asked the students to come up with a condition, which would help determine if the three broken spaghetti pieces form a triangle. To steer their thinking, the teacher asked, “given three lengths, how can you check if a triangle is formed? Can you relate this to something you have studied in your earlier classes?” This cue immediately elicited a response from some students: “The sum of the length of any two pieces must be greater than the third”. Many seconded this observation by relating to the triangle inequality learnt in grade 9. In order to apply the triangle inequality to the Spaghetti problem, the teacher encouraged the students to use algebra to come up with inequations to model the problem. The lengths of the three pieces were taken as x units, y units and $(L - x - y)$ units respectively (since the problem mentions the length of the stick to be L units). With some scaffolding, students used the condition that the *three pieces will form a triangle if the sum of lengths of any two pieces is greater than the third* to arrive at inequations (1) – (3), which can be reduced further to (4) – (5) below:

$$x + y > L - x - y \quad (1) \qquad x + y > L/2 \quad (4)$$

$$x + (L - x - y) > y \quad (2) \qquad y < L/2 \quad (5)$$

$$y + (L - x - y) > x \quad (3) \qquad x < L/2 \quad (6)$$

Students acknowledged that the variables x and y must be greater than 0 and less than L . They argued that neither lengths, x or y , can exceed L units. Students visualized the situation by considering specific values of x , y and L . Finally, they concluded that $0 < x, y < L$.

Development (II) [Abstraction] At this point the teacher encouraged students to think of the lengths of the pieces x and y as being an ordered pair. The realization that each broken stick can be associated with an ordered pair of numbers (x,y) and can be *represented as a point on the coordinate plane* led to much excitement among students. They were familiar with the concept of ordered pairs from their study of coordinate geometry. Further, using ordered pairs to represent broken spaghetti sticks, presented them with a novel way of representing and thinking about the problem opened the door to a new kind of exploration. Students agreed that since x and y are greater than 0, all these points will lie in the first quadrant of the coordinate plane. Also, the inequations (4), (5) and (6) represent those ordered pairs (x,y) which lead to the formation of a triangle. In their curriculum they had learned to graph linear inequations. The first step was to graph the equations $x + y = L/2$, $x + y = L$, $x = L/2$ and $y = L/2$ which led to Figure 12.

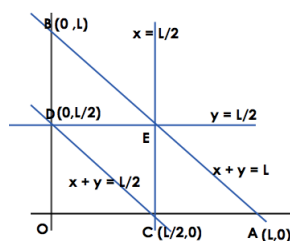


Figure 12. Students graphed the equations emerging from modeling the Spaghetti problem

The next step was to find the common feasible region which satisfied the inequations (4), (5) and (6). A majority of the students were able to conclude that the common region is the triangle DEC. The teacher guided the others who were unable to reach this conclusion on their own. It was reiterated that the common region DEC represented only those ordered pairs (x,y) or broken spaghetti sticks from which triangles could be formed.

The next question was – *what does triangle OAB represent?* This led to plenty of discussion

in the class. Some students suggested that broken sticks would lead to points which lie on the “left side” or “lower side” of the line $x + y = L$ (They were referring to the side containing the origin). This led to the conclusion that every time a spaghetti stick is broken, the corresponding ordered pair (x, y) will lie on or inside the triangle OAB.

The teacher asked, “*what will happen if we break a large number of spaghetti sticks into three pieces each?*” After some discussion students arrived at the conclusion that this would lead to a large number of points inside the triangle OAB. Then the ‘aha’ moment of the class happened when the students realized that only those points that actually lie on or inside triangle DEC represent cases where a triangle can be formed. The teacher then emphasized that *if a sufficiently large number of spaghetti sticks are broken such that their corresponding ordered pairs (x,y) fill up triangle OAB*, then the fraction of those that form triangles will be given by:

$$\frac{\text{Number of ordered pairs } (x,y) \text{ on or inside triangle DEC}}{\text{Total number of ordered pairs } (x,y) \text{ on or inside triangle OAB}}$$

Thus, for a sufficiently large number of broken sticks, the probability of a randomly broken stick (into three pieces) forming a triangle will be given by Area of triangle CDE/Area of triangle OAB.

After this, it was a matter of working out the algebra. Students worked out the areas of the two triangles and arrived at the classical or theoretical probability, i.e., $\frac{1}{4}$ via $\frac{\frac{1}{2} \times \frac{L}{2} \times \frac{L}{2}}{\frac{1}{2} \times L \times L} = \frac{L^2/8}{L^2/2} = \frac{1}{4}$

A few students pointed out that triangle OAB is divided into four congruent triangles which included triangle DEC. Thus, the probability of $\frac{1}{4}$ was an obvious conclusion, which could be reached without even working out the areas of the triangles! The conclusion that the *theoretical probability of a randomly broken spaghetti stick, into three parts, forming a triangle is $\frac{1}{4}$* came as a surprise to those students who had initially conjectured the probability to be $\frac{1}{2}$. The few students who were still not convinced and said that they would actually like to break many sticks to see if the proportion of broken sticks forming triangles is close to 0.25. However, others declared such a feat to be impractical. Questions arose such as: “*How many sticks should we break to verify the answer? What would be an optimum number of sticks?*” For the teacher, this was another high point of the class. This seemed to be the most appropriate time to encourage students to simulate the problem on a spreadsheet.

Starting with the spreadsheet design. What should go into the spreadsheet?

Development (III) [Decomposition] The first thing to do is to decide what data determines the random event of breaking a given piece of spaghetti stick at two points (to yield three pieces). Figure 13 shows possible positions of the breaking points X and Y:

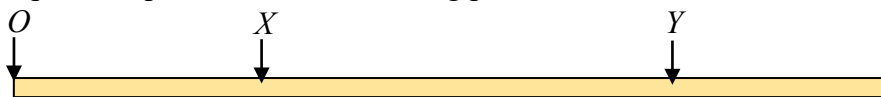


Figure 13. Possible breaking points P and Q on a given spaghetti stick.

The point X may be thought of as the point of distance X from the left end O, and likewise for Y. So these two numbers are randomly generated to give many different scenarios of breaking a given spaghetti sticks at two points. This preliminary thought process yields this simple table.

Table 2. A simple table listing possible breaking points of a given spaghetti stick

Spaghetti Number	X	Y
1		
2		
⋮		

Building an Excel spreadsheet exemplar. [Algorithm design] Prior to this activity, the students had simulated the roll of a pair of dice using the **RANDBETWEEN** command in MS Excel. The Excel syntax **=RANDBETWEEN(1, n)** generates a random integer between 1 and n . Thus, Excel was the natural choice for simulating the breaking of spaghetti sticks described in the Spaghetti Problem. The teacher explained that the lengths of the three pieces represented variables, which were generated randomly via **RANDBETWEEN**. For convenience, the length of each spaghetti stick was assumed to be 100 units and those of the three pieces were a , b and c units respectively. As each stick is broken at two points, these “breaking points” were represented by X and Y respectively (in the sense of Figure 13). Hence, the smaller of these two would be assigned to the first piece a , and the difference $|Y - X|$ be assigned to b . The third piece is of length $(100 - a - b)$. Students were encouraged to visualize the process before proceeding with a 7-step simulation:

Step 1: A column of numbers from 1 to 100 was created in column A to keep track of the simulations. (This was done by entering 1 in cell **A2** followed by **=A2+1** in cell **A3**. Dragging the formula from cell **A3** and till cell **A101** created the required column of numbers.)

Step 2: Since the stick is broken at two points, X and Y respectively, the values of these variables were obtained by entering **=RANDBETWEEN(1,100)** in **B2** and **C2**, where entry in **B2** represented X and that in **C2** Y . The fill handles at **B2** and **C2** then produce 100 pairs of X and Y .

	A	B	C
1	S.No	X	Y
2	1	88	75
3	2	38	37
4	3	94	68
5	4	55	28
6	5	77	58
7	6	20	59
8	7	71	45
9	8	49	30
10	9	43	80
11	10	75	64
12	11	46	45
13	12	31	85
14	13	23	46
15	14	77	90
16	15	41	85

Figure 14: The **RANDBETWEEN** command generates two numbers X and Y between 1 and 100.

Step 3: The smaller of the two numbers X and Y was assigned to a . To simulate this, the following conditional formula was entered in cell **D2=IF(B2<C2,B2,C2)**.

Step 4: After X or Y is assigned to a , the difference $X - Y$ or $Y - X$ was assigned to b . This was achieved by entering the following conditional statement in **E2=IF(B2<C2,C2-B2,B2-C2)**.

Step 5: Finally, simulate the length c of the third piece: **F2=100-(D2+E2)** at cell **F2**.

Once the data was generated (Figure 15), students were encouraged to explain and question different aspects of the data. The first observation was that *each row of the generated data represented one broken spaghetti stick*. Students checked each case for the correct output.

	A	B	C	D	E	F
1	S.No	X	Y	a	b	c
2	1	88	75	75	13	12
3	2	38	37	37	1	62
4	3	94	68	68	26	6
5	4	55	28	28	27	45
6	5	77	58	58	19	23
7	6	20	59	20	39	41
8	7	71	45	45	26	29
9	8	49	30	30	19	51
10	9	43	80	43	37	20
11	10	75	64	64	11	25
12	11	46	45	45	1	54
13	12	31	85	31	54	15
14	13	23	46	23	23	54
15	14	77	90	77	13	10
16	15	41	85	41	44	15

Figure 15: The pieces of lengths a , b and c are simulated in columns D, E and F respectively.

Step 6: The next step was to identify those cases in which a triangle was formed. Some students explained that if one of the three pieces a , b or c is greater than or equal to 50 units then a triangle cannot be formed. They concluded that a triangle will be formed only if the maximum length among a , b and c is less than 50 (half the length of the original stick). Alternatively, the students could apply the inequalities (4) to (6) to reach the same condition. This was then translated into the formula $=\text{IF}(\text{MAX}(\text{D2},\text{E2},\text{F2})<50,\text{“YES”},\text{“NO”})$ which entered in G2. This led to the output in cell G2 - “YES” if a triangle was formed and “NO” otherwise (Figure 16).

	A	B	C	D	E	F	G
1	S.No	X	Y	a	b	c	Forms a triangle?
2	1	88	75	75	13	12	NO
3	2	38	37	37	1	62	NO
4	3	94	68	68	26	6	NO
5	4	55	28	28	27	45	YES
6	5	77	58	58	19	23	NO
7	6	20	59	20	39	41	YES
8	7	71	45	45	26	29	YES
9	8	49	30	30	19	51	NO
10	9	43	80	43	37	20	YES
11	10	75	64	64	11	25	NO
12	11	46	45	45	1	54	NO
13	12	31	85	31	54	15	NO
14	13	23	46	23	23	54	NO
15	14	77	90	77	13	10	NO
16	15	41	85	41	44	15	YES

Figure 16: The conditional statement $=\text{IF}(\text{MAX}(\text{DE},\text{E2},\text{F2})<50,\text{“YES”},\text{“NO”})$ was used to verify if each row of the simulation represents the formation of a triangle.

Step 7: Finally it was required to count the number of cases in which a triangle was formed (indicated by “YES” in column G) to compute the empirical probability. For this the formula $=\text{COUNTIF}(\text{G2}:\text{G101},\text{“YES”})/100$ was entered in J2. The empirical probability was computed as

$$\frac{\text{Number of broken spaghetti sticks which form a triangle}}{\text{Total number of spaghetti sticks}}$$

	A	B	C	D	E	F	G	H	I	J
1	S.No	X	Y	a	b	c	Forms a triangle?			
2	1	88	75	75	13	12	NO			Probability
3	2	38	37	37	1	62	NO			0.28
4	3	94	68	68	26	6	NO			
5	4	55	28	28	27	45	YES			
6	5	77	58	58	19	23	NO			
7	6	20	59	20	39	41	YES			
8	7	71	45	45	26	29	YES			
9	8	49	30	30	19	51	NO			
10	9	43	80	43	37	20	YES			
11	10	75	64	64	11	25	NO			
12	11	46	45	45	1	54	NO			
13	12	31	85	31	54	15	NO			
14	13	23	46	23	23	54	NO			
15	14	77	90	77	13	10	NO			
16	15	41	85	41	44	15	YES			

Figure 17: The empirical probability of a triangle being formed in these 100 simulations is 0.28.

Meeting the Learning Objective. It was emphasized that since each row represents the data of *one* broken stick, the entire data is a simulation of 100 sticks. Each time a cell in the spreadsheet was clicked, a new set of 100 simulations would be obtained. The empirical probability would show up in cell J2 on their sheet. Students were asked to create 10 simulations each (leading to a total of 1000 broken sticks) and to note the empirical probability (Figure 17). A student proposed a shortcut to this. She suggested that column A should accommodate numbers from 1 to 1000 (in the very first step and each formula can be evaluated to arrive at a spreadsheet with 1000 simulations at one go! This suggestion was welcomed by everyone. The teacher asked students to compare the empirical probability obtained from 100 simulations with that obtained from 1000 simulations. The spreadsheet was further modified to accommodate 5000 simulations. A careful observation of the data generated led students to conclude that as the number of simulations increases, the empirical probability gives a better approximation and is closer to the theoretical probability $\frac{1}{4}$.

Consolidation Simulation enabled the students to explore the Spaghetti problem numerically,

which would be impossible without a spreadsheet. As a natural extension of the problem, students inquired: *if a stick is broken into four pieces, what is the probability that they will form a quadrilateral?* Some ventured to find the answer by extending the spreadsheet simulation for three pieces to four pieces to find the empirical probability. Others attempted a theoretical investigation. This instructional exemplar resulted in students' engagement in multiple paths of inquiry.

4. Concluding remarks

This paper proposes the use of spreadsheet as a connection between big ideas in mathematics and computational thinking, and how such a connection can be translated using a simple task design schema to create engaging mathematics lesson activities. Apart from the obvious time-saving benefit of using spreadsheets in preference to programming languages and packages (Morshita et al, 2001; Baker and Sudgen, 2003), the use of spreadsheet forges a new non-algebraic way of learning algebra (Sutherland and Rojano, 1993). We amplify the versatility of spreadsheets in mathematics education by making explicit the link between *big ideas in mathematics* and *computational thinking*. By so doing, we hope to encourage teachers to exploit this connection and to put it to good use in creating effective mathematics lessons that engage students' mathematical and computational thinking. In an early paper presented by D'Sourza and Wood (2001) in the 6th ATCM, the use of spreadsheets was proposed (theoretically) to improve students' learning of mathematics in a collaborative environment. Our present paper can be taken as a realization of D'Sourza and Wood's proposal, perhaps taken a step further by considering not just collaborative but constructionistic environment – where students are computationally-minded forgers of big ideas in mathematics.

Acknowledgements The work reported in this paper is part of the “Big Ideas in School Mathematics” (BISM) project (OER 31/19 BK), where the third author is the PI and the first a collaborator; and also part of the “How to Bring Computational Thinking Into Mathematics Classrooms: Designing for Discipline-specific CT” project (OER 10/18 LCK), where the fourth author is the PI and the first is a Co-PI.

References

- [1] Abramovich, S., and Brantlinger, A. (2003). Linking numeric, graphic, and geometric representations of number theory concepts in a spreadsheet environment. In G. Goodell (Ed.), *Proceedings of the 15th Annual International Conference of Technology in Collegiate Mathematics*, (pp. 1 – 5). Addison-Wesley.
- [2] Arganbright, D. (2006). Enhancing Mathematical Graphical Displays in Excel through Animation. *Spreadsheets in Education*, 2(1), 125 – 147.
- [3] Australian Curriculum Assessment and Reporting Authority. (2013). *Australian curriculum: Mathematics*. Retrieved from <http://www.australiancurriculum.edu.au/>
- [4] Baker, J., and Sudgen, S. (2003). Spreadsheets in Education – The First 25 Years. *Spreadsheets in Education*, 1(1), 18 – 43.
- [5] Bruner, J. (1961). *The act of discovery*. Harvard Educational Review.
- [6] Chan, S.-W., Looi, C.-K., Ho, W. K., Huang, W., Seow, P. and Wu, L. (2021). Learning number patterns through computational thinking activities: A Rasch model analysis. *Heliyon*, 7 (e07922), 1 – 14.
- [7] Chan, S.-W., Looi, C. K., Ho, W.K., and Kim, M. S. (2022). Tools and Approaches for Integrating Computational Thinking and Mathematics: A Scoping Review of Current

- Empirical Studies. *Journal of Educational Computing Research*, Advanced online publication, 1 – 59. Available at <https://doi.org/10.1177/07356331221098793>.
- [8] Charles, R. I. (2005). Big ideas and understandings as foundations for elementary and middle school mathematics. *Journal of Mathematics Education Leadership*, 7(3), 9 – 14.
- [9] Choy, B. H. (2019). Teaching towards Big Ideas: Challenges and Opportunities. In T. L. Toh and J. B. W. Yeo (Eds.), *Big Ideas In Mathematics, Yearbook 2019, Association of Mathematics Educators* (pp. 95 – 112). World Scientific.
- [10] D’Sourza, S. M., and Wood, L. N. (2001). Investigating the Effects of Using Spreadsheet in a Collaborative Learning Environment. In W.-C. Yang, S.-C., Chu, Z. Karian, and G., Fitz-Gerald (Eds.), *Electronic Proceedings of the 6th ATCM*, Australia.
- [11] Hiebert, J., and Carpenter, T. P. (1992). Learning and teaching with understanding. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 65 – 97). New York: Macmillan Publishing Company.
- [12] Ho, W. K., Looi, C. K., Huang, W., Seow, P., Chan, S. W., and Wu, L. (2021). Flowgorithm: A possible passage from algorithmic reasoning to creatively founded mathematical reasoning. In Yang, W.-C., and Meades, D. & Majewski, M. (Eds.), *Electronic Proceedings of the Twenty-Sixth Asian Technology Conference in Mathematics (Scopus Indexed)*, 97 – 111. United States: Mathematics and Technology, LLC.
- [13] Ho, W. K., Looi, C. K., Huang, W., Seow, P., and Wu, L. (2019). Realizing Computational Thinking in Mathematics Classroom: Bridging the Theory-Practice Gap. In W.-C. Yang, & D. Meades (Eds.), *Proceedings of the Twenty-fourth Asian Technology Conference in Mathematics*, Mathematics and Technology, LLC.
- [14] Hoyle, C., and Noss, R. (2015). Revisiting Programming to Enhance Mathematics Learning. *Math + Coding Symposium Western University*, London, Ontario, Canada.
- [15] Hvorecky, J., and Trencansky, I. (1998). Recursive Computations in Spreadsheets. In W.-C. Yang, K. Shirayanagi, S.-C. Chu, and G. Fitz-Gerald (Eds.), *Electronic Proceedings of the 3rd Asian Technological Conference in Mathematics*, 290 – 298. Springer, Tokyo.
- [16] Marley-Pane, J. and Dituri, P. (2019). Spreadsheets as an Effective Use of Technology in Mathematics Education. *Spreadsheets in Education*, 12(1), 1 – 26.
- [17] Ministry of Education. (2018). *2020 secondary mathematics syllabuses (draft)*. Singapore: Curriculum Planning and Development Division.
- [18] Morishita, E., Iwata, Y., Yoshida, K. Y., and Yoshida, H. (2001). Spreadsheet fluid dynamics for aeronautical course problems. *International Journal of Engineering Education*, 17(3), 294 – 311.
- [19] National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: National Council of Teachers of Mathematics.
- [20] OECD. (2021). PISA 2021 Mathematics Framework (Draft). November 2018.
- [21] Papert, S. (1996). An Exploration in the Space of Mathematics Educations, *International Journal of Computers for Mathematical Learning*, 1(1), 95 – 123.
- [22] Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic books, New York.
- [23] Sanford, J. (2018). Introducing Computational Thinking Through Spreadsheets, In M.S. Kline (Ed.), *Computational Thinking in the STEM disciplines*. Springer International Publishing AG, pp. 99 – 124.
- [24] Sutherland, R., and Rojano, T. (1993). A Spreadsheet Approach to Solving Algebra Problems. *Journal of Mathematical Behaviour*, 12(4), 351 – 383.

- [25] Tabesh, Y. (2017). Computational thinking: a 21st century skill [Special issue]. *Olymp. Info. 11*, 65 – 70.
- [26] Toh, T. L., Quek, K. S., Leong, Y. H., Dindyal, J. and Tay, E. G. (2011). *Making Mathematics Practical: An Approach to Problem Solving*. World Scientific.
- [27] Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33 – 35.