Some Opportunities for Computational Thinking in the Mathematics Classroom

Jonaki B Ghosh
jonakibghosh@gmail.com

Department of Elementary Education, Lady Shri Ram College for Women

University of Delhi

INDIA

Abstract: *Computational thinking has been identified as an important skill for students who wish to pursue mathematics and mathematics related disciplines as a career. This has called for a special focus on CT activities in the K – 12 curriculum. While many definitions of CT may be found in the literature, all of them seem to focus on specific skills, such as, the ability to deal with challenging problems, representing ideas in computationally meaningful ways, creating abstractions for the problem at hand, breaking down problems into simpler ones, assessing the strengths and weaknesses of a representation system and engaging in multiple paths of inquiry. These skills are also critical for mathematics learning and there is a common consensus on the understanding that CT skills have to be developed in mathematics classrooms right from the school years. However finding appropriate tasks which help to develop and elicit such thinking remains to be a key pedagogical challenge. Further teachers need to be empowered to create as well as integrate CT tasks in their lessons. This article hopes to suggest a guide map for preparing and integrating CT based activities in the mathematics classroom. It also attempts to highlight the pivotal role of in engaging students and in steering them towards computational thinking. Examples of tasks ranging from iteration and recursion in fractal constructions to exploration of chaos and simulation of queues and games which were conducted with secondary school students in the Indian context will be discussed.*

1. Introduction

In recent years computational thinking has been identified as one of the key analytical abilities required for mathematics and science learning. The rapidly changing nature of scientific and mathematical disciplines and the need to prepare students for careers in these disciplines have been the primary motivation for bringing computational thinking into classroom practices. Papert [2,4] was the first to emphaisse the importance of computational thinking by referring to the affordances of computational representations for highlighting powerful ideas. Over the decades many researchers have attempted to define computational thinking. Stephen Wolfram [7], in his book *A new kind of science*, talks about the computational world, its relationship with the physical world and the importance of computational thinking. Wing [6] in her seminal article expressed that computational thinking involves solving problems, designing systems and understanding human behaviour by drawing on the concepts fundamental to computer science. However operationalising computational thinking for the K – 12 classroom is a key pedagogical challenge. Experts believe that computational thinking should not be restricted to mathematics and science only, and that it should be integrated in the social sciences, art and language curricula. It would require a monumental effort to design a coherent curriculum which integrates computational thinking in the various school subjects.

In this paper we shall make a case for computational thinking in the mathematics classroom. We argue that it can be developed by engaging students in meaningfully designed tasks which not only highlight the practical relevance of mathematics but also illustrate the benefits of exploring real world phenomena through computer based explorations. The ability to deal with challenging problems, represent ideas in computationally meaningful ways, create abstractions for the problem at hand, break down problems into simpler ones and assess the strengths and weaknesses of a representation system are some important aspects of computational thinking for mathematics learning. In section 2 of the article we shall briefly discuss some of the theoretical underpinnngs of computational thinking. In section 3 we shall describe how students of grades 11 and 12 worked on certain investigatory tasks and engaged in various aspects of computational thinking. These include fractal based explorations, understanding of chaos and simulation of queues and games. Technology in the form of spreadsheets (MS Excel), dynamic geometry software (GeoGebra) and graphics calculators (Casio cg-20) played a pivotal role in enabling these explorations and helping students to obtain a deeper insight into the problems. In section 4 we present a discussion based on students' responses related to the tasks described in section 3.

2. Theoretical Framework

Back in the 1980s Seymour Papert [2] had expounded the role of computers in mathematics learning. In his book, *Mindstorms* he wrote

> Before computers there were very few good points of contact between what is most fundamental and engaging in mathematics and anything firmly planted in everyday life. But the computer — a "mathematics - speaking being" in the midst of the everyday life of the home, school, and workplace—is able to provide such links. The challenge to education is to find ways to exploit them.

His pioneering work paved the way for addressing some key questions related to computational thinking - How is computational thinking related to mathematical thinking or problem solving? How does it relate to the field of computer science?

While the term computational thinking was introduced by Papert, it was Jeanette Wing who popularised it in her short but seminal article [6].

According to Wing [6]

> Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking.

In the article, Wing [6] further explains what computational thinking entails.

> Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

She further elaborates that computational thinking entails "formulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation", thinking recursively, “using abstraction and decomposition when attaching large problems” and using heuristic reasoning.

She also describes some characteristics of computational thinking. For instance, she differentiates between computer science and computer programming

> Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction;

and describes computational thinking as “a way humans solve problems”.

Wing [6] elucidates that computational thinking is not

> …trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations.

Expanding on the relationship between Computer science and Mathematics, which is in a way a thrust of this article, Wing [6] writes

> Computer science inherently draws on mathematical thinking, given that, like all sciences, its formal foundations rest on mathematics. Computer science inherently draws on engineering thinking, given that we build systems that interact with the real world. The constraints of the underlying computing device force computer scientists to think computationally, not just mathematically.

Weintrop et. al [5] attempt to define computational thinking in the context of school mathematics and science education and also suggest a theoretical grounding for the same. They propose a taxonomy comprising four categories: data practices, modeling and simulation practices, computational problem solving practices and systems thinking practices. Their article illustrates a series of computational thinking enhanced lesson plans for high school biology, chemistry, and physics classrooms. Despite the attempts by researchers, there appears to be a lack of precision in documenting how computational thinking can be brought into the mathematics classroom. Hence the focus of this article is to explore the possibilities of integrating CT tasks in the mathematics classroom.

3. Computational Thinking Tasks in the Mathematics Classroom

In this section we shall describe a study in which CT tasks were attempted by senior secondary school students. The students who participated in this study were selected from four schools across New Delhi which follow the curriculum prescribed by the Central Board of Secndary Education (CBSE) [1], a national board of education in India. The tasks were exploratory in nature and required the knowledge of functions, geometric sequences and probability. These topics are a part to the senior secondary mathematics curriculum. Prior to the tasks students were familiarized with the basic features of the computational tool they would require for exploration. In general they were familiarized with basic features of spreadsheets, dynamic geometry software and graphics calculators. The sessions were conducted after school hours and on Saturdays so that they would not hinder the school schedule.

3.1. Fractals constructions and recursive thinking.

The study of Fractals is generally not included in school curricula. However the pedagogical benefits of this topic is immense as it provides a rich source of exploratory tasks which lead to notions of iteration and recursion. Recursive reasoning is an important skill for computational thinking. The concepts of self-simiarity and fractal dimension, if introduced through appropriate tasks, can be well within the reach of school students. In this module students of grade 11 explored various geometric fractals such as the Sierpinski triangle, Sierpinski Carpet, Koch Snowflake and Pythagorean tree and also created their own fractal patterns. This module was introduced after students had learnt the basic properties and formulae related to geometric sequences. In this subsection we shall describe students' explorations of two fractals, namely the Sierpinski carpet and the Pythagorean tree.

3.1.1. Sierpinski Carpet

In the very first session of the module, students were introduced to the Sierspinski Carpet construction where stage 0 comprises a square cut out from a sheet of paper. The four sides are trisected and points of trisection are marked on all the sides. When points of trisection of opposite sides are joined, 9 smaller squares are created. To obtain stage 1, the center square is removed and 8 shaded squares are obtained. This stage appears to be a square with a square 'hole' in the center. The same process of trisection and removing the center square is repeated on the 8 smaller shaded squares to obtain stage 2. This process is further continued to obtain higher stages of the fractal as shown in figure 1. Students did the construction process on a dotted grid paper to ease out the trisection process.

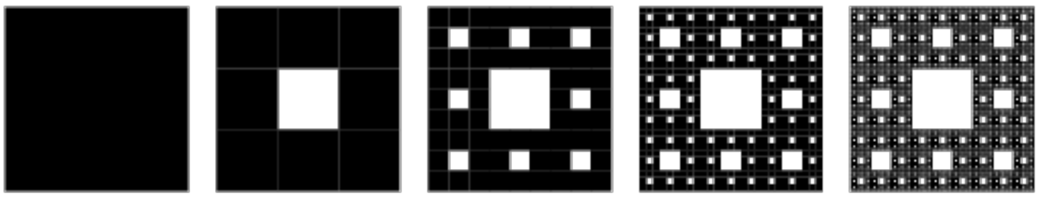
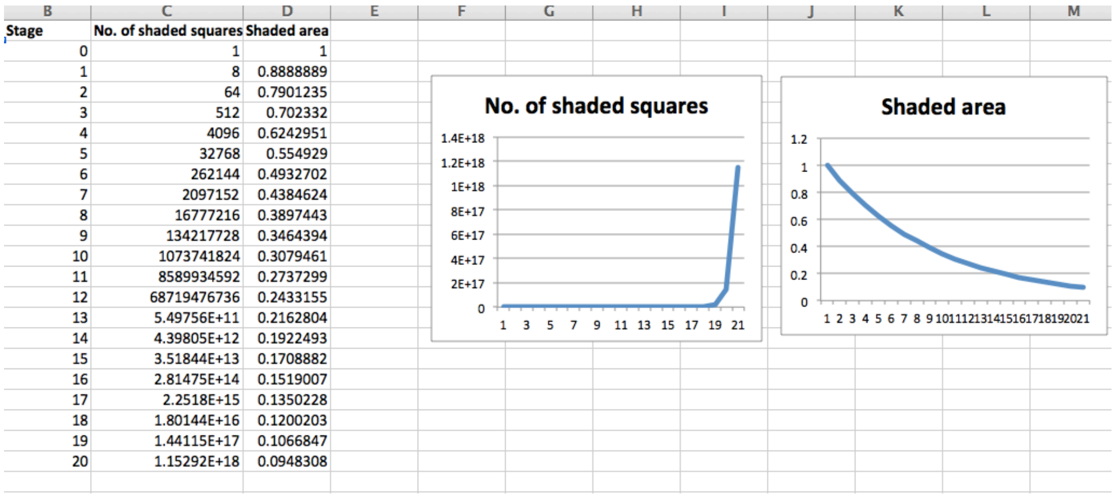


Figure 1. Stages 0 to 4 of the Sierpinski square carpet

As the construction process ensued, they began to predict the geometric sequences, which would emerge by counting the number of shaded squares and the shaded area at each stage. They concluded that the number of shaded squares would lead to the geometric sequence $1, 8, 8^2, 8^3, \ldots\ldots.$ (with a multiplying factor of 8) whereas the shaded area through the stages leads to the sequence 1, 8/9, $(8/9)^2\ldots$ with the multiplying factor of 8/9. They also obtained the recursive and explicit formulae for the number of shaded squares as $S_n = 8S_{n-1}$ and $S_n = 8^n$. The corresponding formula for shaded area at the *n*th stage were obtained as $A_n = (8/9)\ A_{n-1}$ and $A_n = (8/9)^n$ respectively. This was followed by a spreadsheet exploration where students generated these sequences numerically and represented them graphically as shown in figure 2. Both numerical and graphical outputs helped to visualize the fractal at higher stages, which would not have been possible using pictorial representations. They were also able to visualize that the number of shaded squares grow exponentially as the stages increase whereas the shaded area approaches 0. This was followed by students identifying self-similarity within the Sierpinski carpet construction, looking for copies of lowers stages within higher stages of the fractal.



Stage	No. of shaded squares	Shaded area
0	1	1
1	8	0.8888889
2	64	0.7901235
3	512	0.702332
4	4096	0.6242951
5	32768	0.554929
6	262144	0.4932702
7	2097152	0.4384624
8	16777216	0.3897443
9	134217728	0.3464394
10	1073741824	0.3079461
11	8589934592	0.2737299
12	68719476736	0.2433155
13	5.49756E+11	0.2162804
14	4.39805E+12	0.1922493
15	3.51844E+13	0.1708882
16	2.81475E+14	0.1519007
17	2.2518E+15	0.1350228
18	1.80144E+16	0.1200203
19	1.44115E+17	0.1066847
20	1.15292E+18	0.0948308

Figure 2. Excel simulation of the Sierpinski Carpet

In subsequent sessions, students created their own fractal patterns, analysed them using geometric sequences and also explored self similarity within them. Figure 3 illustrates one such example. After dividing the original square into 9 smaller squares, four squares (shown in yellow) are removed as shown. This process is repeated recursively and the number of shaded squares at various stages leads to powers of 5 whereas the shaded area leads to powers of 5/9.

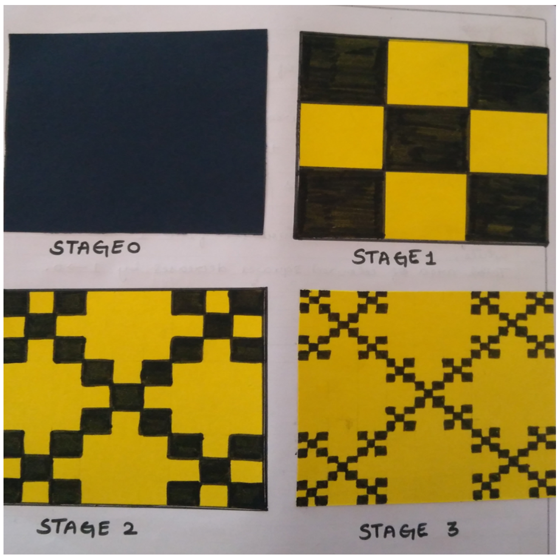


Figure 3. A fractal pattern generated from the square

3.1.2. Pythagorean tree

The Pythagorean tree is constructed iteratively in which squares are erected on all three sides of a right-angled triangle. In subsequent stages, similar right-angled triangles are constructed on the two legs of the original triangle in such a manner that the legs of the original triangle become the hypotenuse of the new triangles. When this process is continued a few times, a beautiful intricate fractal pattern emerges.

The initial steps of the iterative process were demonstrated to students using diagrams on the whiteboard. Subsequently they constructed various stages of the tree on GeoGebra using the ‘create tool’ feature which enables recursive constructions. Figure 4 shows the initial stages of a symmetrical Pythagorean tree and figure 5 shows the asymmetrical version.

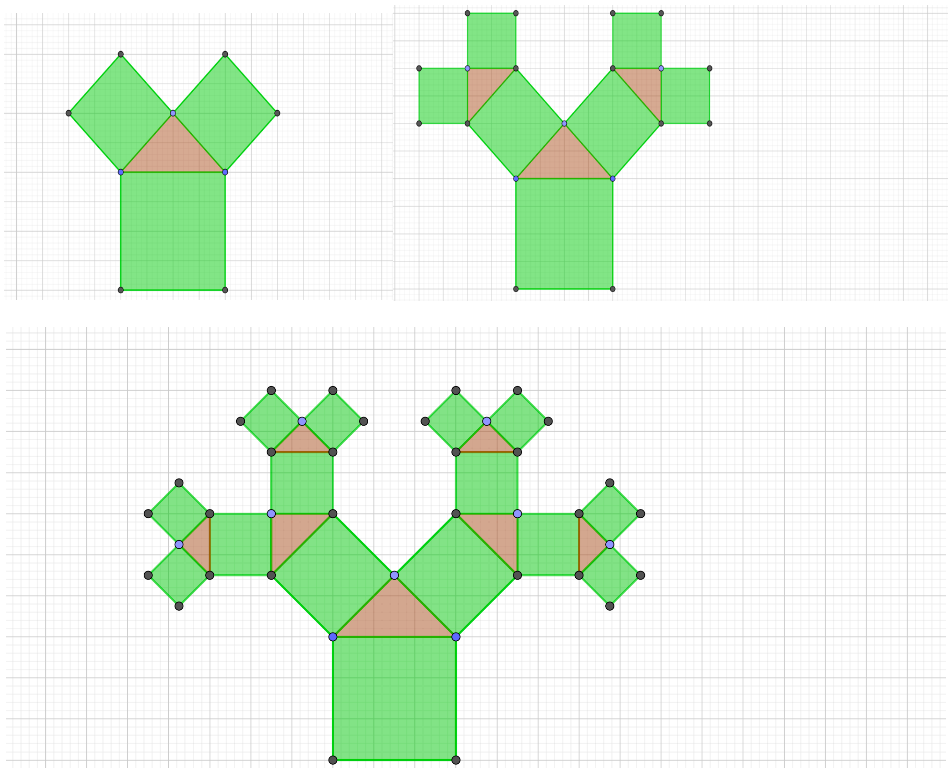


Figure 4. Stages 0, 1 and 2 of the symmetrical Pythagorean tree

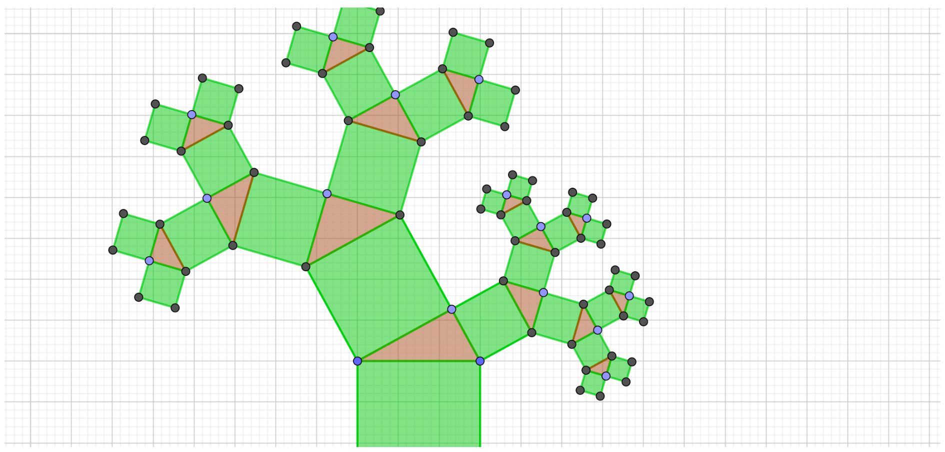


Figure 5. Asymmetrical version of the Pythagorean tree

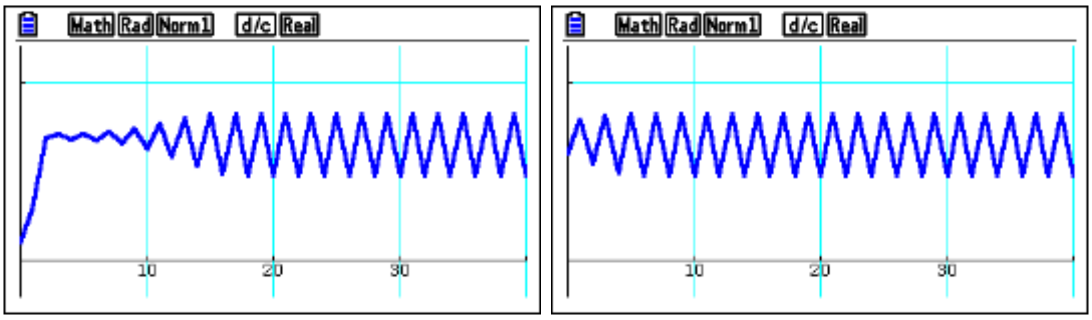
After this students explored various number sequences arising from different attributes of the construction. Firstly they computed the areas of the right triangles appearing along one branch of the tree. Assuming the legs of the initial triangle to be of unit length, they arrived at the geometric sequence $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots\ldots\ldots$, with a multiplying factor of ½. Further they computed the infinite sum of the sequence using the formula $\frac{a}{1-r}$ and discovered that with increasing number of stages, the sum of the areas of the triangles approaches 1! This revelation fascinated them as they had learnt the properties of geometric sequences in their regular classes but had not seen them being applied to any process. Also an infinite process leading to a finite sum was intriguing to them. Similarly, computing the areas of the squares (starting with the square on the hypotenuse of the initial triangle) led to the sequence 2, 1, $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots\ldots\ldots$ The sum of infinite number of terms of this sequence yielded $\frac{2}{1-(\frac{1}{2})} = 4$, once again illustrating a situation in which an infinite process leads to a finite value.

Overall the student engagement in fractal tasks was very satisfying as they actively participated in the explorations, learnt the topic of geometric sequences which is a part of the curriculum, created their own fractal designs and developed a firm grip on the processes of iteration and recursion.

3.2. Understanding Chaos

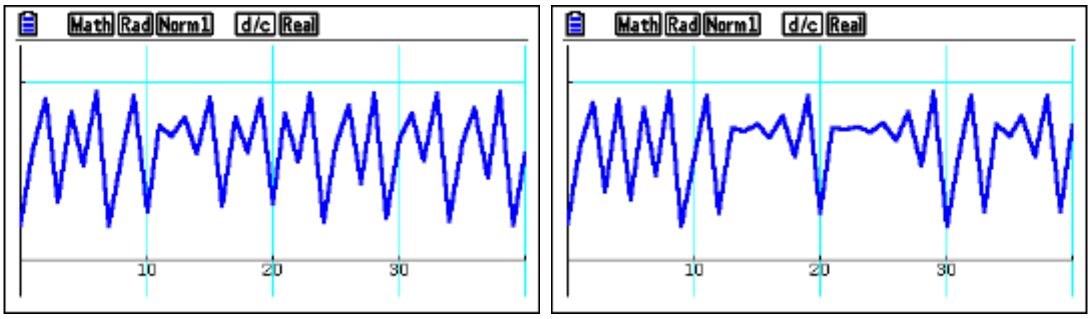
In this module, a group of grade 12 students explored the chaotic behaviour of quadratic functions. These students had studied the topic of functions in their regular classes and were familiar with linear, quadratic, cubic, exponential and logistic functions. In order to explore the idea of chaos, they were prompted to begin with the logistic function $f(x) = ax(1-x)$. The idea was to generate the numerical sequence of iterates of the function for different values of a ranging from 3 to 4 in steps of 0.1. For each value of a different initial seeds were considered to study the behaviour of the the function in the long run.

Graphics calculators were used as the primary vehicles of exploration in this module. In order to explore the given task, students used the recursive mode of the calculator (RECUR) where the quadratic function was entered as a recursive sequence, $a_{n+1} = 3.3a_n(1 - a_n)$. 30 iterations of this sequence were generated in the TABLE mode and were also graphed. The connected graph for a = 3.3 for x_0 = 0.1 and 0.6, shown in figure 6 (i) and (ii), reveals a powerful progression of the terms. Students took different starting points and concluded that "the graphs settle down into the same pattern of oscillating between the values 0.48 and 0.82" even though they may look different in the beginning. For a = 3.5 the iterates form a regular pattern oscillating between 0.87, 0.38, 0.82, and 0.50. However for a = 3.8 a very different scenario emerged. Different initial values x_0 led to different graphical patterns. Figure 7 shows the connected graphs for x_0 = 2.1 and 2.2. Students articulated in their observations that "even though the initial values are very close, the graphs become very different rather quickly" and "They don't appear to settle down at all and behave in an arbitrary manner". This situation was used by the teacher to introduce the idea of *sensitive dependence on initial conditions.*



(i) (ii)

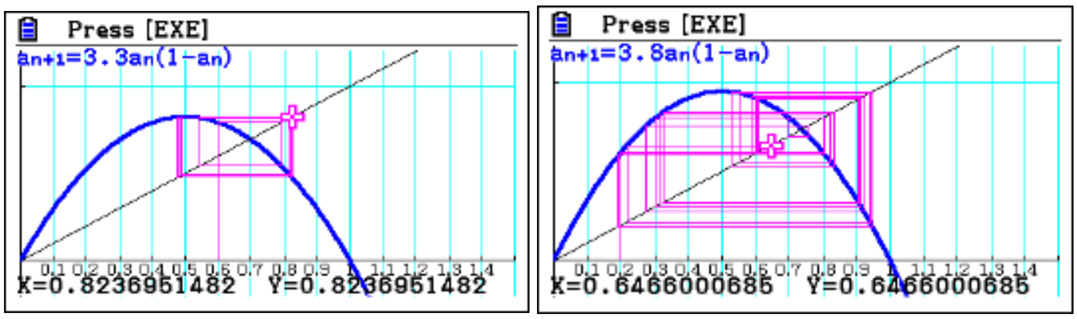
Figure 6. Iterations of the recursive sequence $a_{n+1} = 3.3a_n(1 - a_n)$ for $x_0 = 0.1$ and 0.6



(i) (ii)

Figure 7. Iterations of the recursive sequence $a_{n+1} = 3.8a_n(1 - a_n)$ for $x_0 = 2.1$ and 2.2

The graphics calculator provides another powerful way of visualizing recursive sequences using *cobweb plots*. This feature was exploited by students to see the long term behaviour of various recursive sequences. Figure 8 (i) and (ii) shows the cobweb plots for $a = 3.3$ and $a = 3.8$. For $a = 3.3$, the plot shows a clear convergent rectangular pattern whereas for $a = 3.8$ the pattern is rather chaotic. Sometimes a convergent pattern seems to appear but it disappears very quickly.



(i) (ii)

Figure 8. The cobweb plots for a = 3.3 and a = 3.8

In this module graphics calculators played a crucial role in helping students visualize the notion of a chaotic process. Although numerical and graphical explorations of recursive sequences can be easily done on a spreadsheet, like MS Excel, the additional feature of creating cobweb plots provided by the graphics calculators was particularly helpful in developing a graphical understanding of the iterative process.

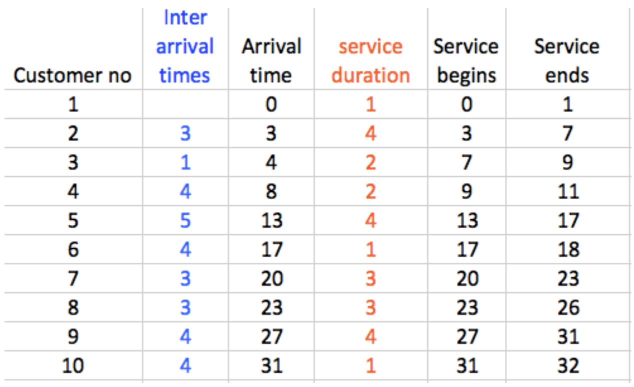
3.3. The power of simulations in developing computational thinking

Simulation is a modeling tool, which is used to imitate real-world problems in order to understand system behavior. In particular *Monte Carlo Simulation* is a problem solving technique used to approximate the probability of certain outcomes by running multiple trial runs, (called simulations), using random variables. Simple simulations of real world problems can be made accessible to school students through spreadsheets such as MS Excel. Through simulations students may be encouraged to explore real problems and also engage in computational thinking.

3.3.1. Simulating queues

In this module grade 11 students observed queues of cars in a vehicle fueling station, which had a single server. The observations were made during peak hours over a week. Later they tried to simulate the problem in Excel to understand the characteristics of the queuing process.

A queuing system is characterised by the nature of arrivals in the queue, the service mechanism, the queue capacity and the manner in which the customers are served. In a single server queue, arrivals or service occur one at a time in a random fashion and once a customer joins the queue, is eventually served. In such a system, there are only two possible events that can affect the state of the system - the arrival event (the entry of a unit into the system) and the departure event (the completion of service). The queuing system includes the server, the unit being served and the units waiting in the queue. A simulation table which keeps a record of the arrivals and service times may be used to track the system. In a queuing system, inter arrival times and service times are randomly distributed and may be simulated using random numbers. Students were familiarised with the concept of random numbers and probability distributions and with how these can be used to model the arrivals and services in a queuing system. At the start of the module they used the RAND and RANDBETWEEN commands in Excel to simulate the throws of a coins and dice. Later they used these to simulate interarrival times and service duration times. Figure 9 shows an Excel simulation for 10 customers (cars) where the times between arrivals range from 1 to 6 minutes and the service times (service duration) randomly varies from 1 to 4 minutes. These time ranges were estimated based on students' observations. Using simple rules, the actual arrival times and the starting service time as well as the ending service time for each customer was computed.



Customer no	Inter arrival times	Arrival time	service duration	Service begins	Service ends
1		0	1	0	1
2	3	3	4	3	7
3	1	4	2	7	9
4	4	8	2	9	11
5	5	13	4	13	17
6	4	17	1	17	18
7	3	20	3	20	23
8	3	23	3	23	26
9	4	27	4	27	31
10	4	31	1	31	32

Figure 9. Simulation of a queue for 10 customers

A natural curiosity which arore among students was to find the average customer waiting time in the queue or the average time that the service counter remains idle. According to them this information would enable them to help the service station owner to improve his service efficiency. Figure 10 shows the simulation where the average waiting time and idle time have been computed.

Customer no	Inter arrival times	Arrival time	service duration	Service begins	Service ends	Waiting time in queue	Total time in system	server Idle time			
1		0	2	0	2	0	2	0			
2	5	5	1	5	6	0	1	3			
3	2	7	2	7	9	0	2	1		Average waiting time	1.62
4	4	11	3	11	14	0	3	2			
5	5	16	4	16	20	0	4	2		Average service time	2.57
6	2	18	3	20	23	2	5	0			
7	3	21	1	23	24	2	3	0		Average server idle time	0.96
8	3	24	4	24	28	0	4	0			
9	2	26	3	28	31	2	5	0			
10	3	29	3	31	34	2	5	0			

Figure 10. Simulation of a queue for 100 customers (all numbers are in minutes)

After simulating for 100 customers the following were calculated

Average waiting time for a customer = $\dfrac{total\ waiting\ time}{total\ number\ of\ customers}$

Average service time = $\dfrac{total\ time\ spent\ in\ service}{total\ number\ of\ customers}$

Average time that the server remains idle = $\dfrac{total\ idle\ time}{total\ number\ of\ customers}$

Students computed these values by for different choices of interarrival times and service durations. Their aim was to reduce waiting time of cars and the idle time of the server.

3.3.2. Simulating games of chance

Another useful way for students to engage with simulations is to analyze games of chance, since such games, by their very nature, are driven by randomness. A very popular game (also played in casinos) is the Game of Craps, which is played with a pair of dice. It is said that if you want to double your money quickly on a game of pure chance, one of your best opportunities is to bet all your money on one game of craps! Your probability of winning is just slightly under 50%. In this module students were familiarized with the rules of the game (as mentioned below) and were asked to analyse the game using rules of probability and by simulation in Excel.

1. If a total of 7 or 11 is obtained on the first roll, you win.

2. If a total of 2, 3 or 12 is obtained on the first roll, you lose.

3. If a total of 4, 5, 6, 8, 9 or 10 is obtained on the first roll, this number becomes your **point**. You continue to roll the dice. If you get your total point before a total of 7 appears, you win. If you roll a total of 7 before your total point appears, you lose.

Grade 12 students who were familiar with the basic concepts of probability attempted this task. They analysed the game by computing the probability of winning in the first roll, that is, getting a total of 7 or 11. The possibilities of the numbers appearing on the dice were enumerated as (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) for 7 and (5,6) and (6,5) for 11. Thus there are 8 favorable outcomes out of a total of 36 possible outcomes leading to a probability of 8/36 = 22.22%

Similarly the probability of losing on the first roll, that is, getting a total of 2, 3 or 12 was calculated to be = 4/36 = 11.11%

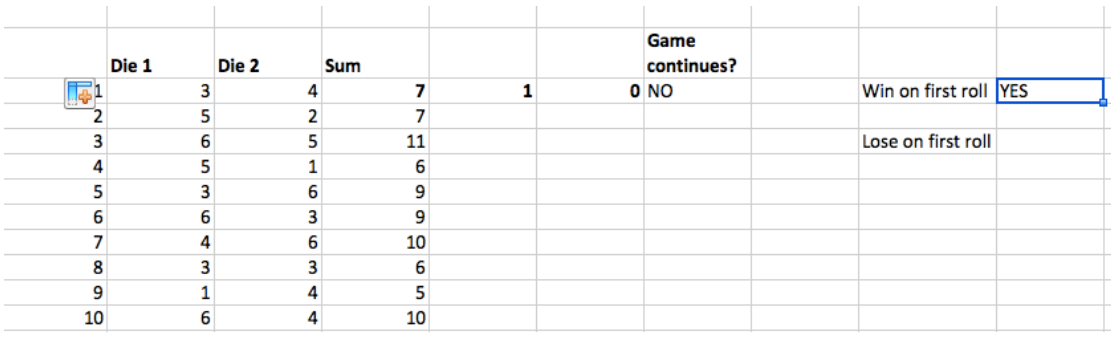
The next task was to calculate the total probability of winning by getting a total of 4, 5, 6, 8, 9 or 10. In order to find the probability of winning by getting a total of, say 4, we would need to find the probability of getting 4 in the first roll and then getting 4 before getting 7 in subsequent throws. This is a case of conditional probability and it can be calculated as follows:

The probability of getting a 4 in the first roll is 3/36 (the possible outcomes being (1,3), (2,2) and (3,1)). In subsequent rolls, only totals of 4 and 7 need to be considered which leads to only 9 possibilities (3 for a total of 4 and 6 for a total of 7). Among these a win is possible only when a total of 4 precedes a total of 7, the probability for which is 3/9. Thus the total probability of winning by getting a 4 in the first roll is $\frac{3}{36} \times \frac{3}{9} = 0.027$. Similarly students computed the probability of winning by getting a total of 5, 6, 8, 9 and 10. The total probablity of winning a single game of craps (as shown in Table 1) works out to approximately 0.49.

Table 1: Probabilities of winning a Game of Craps

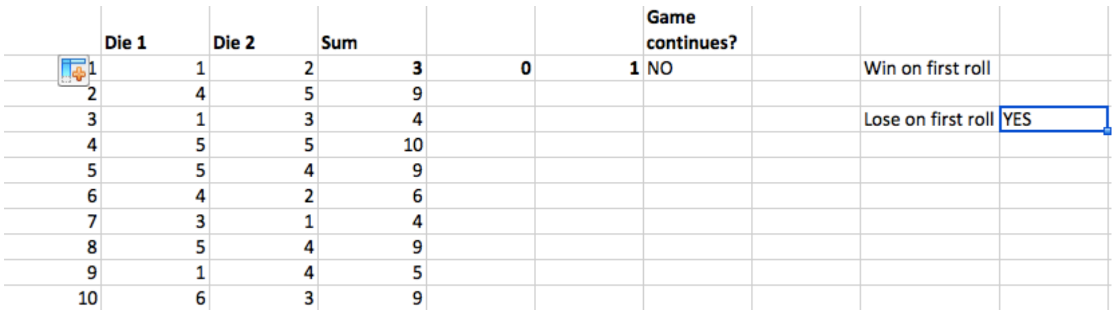
Initial total	Probability
4	0.027
5	0.044
6	0.063
7	0.167
8	0.063
9	0.044
10	0.027
11	0.056
Total probaility of winning	0.493

After the theoretical analysis of the problem, students used the IF, OR, AND, RANDBETWEEN, SUM and COUNT commands in Excel to simulate the game. The aim was to explore the average number of throws required to win the game. Figure 11 illustrates four scenarios. In (a) the sum of the numbers of the two die in the first roll is 7 and the player wins so the game doesn't continue beyond the first roll. In (b) the player loses in the first roll as the sum is 2. Once again the game doesn't continue beyond the first roll. (c) and (d) represent cases when the player neither wins nor loses in the first roll. In (c) the sum in the first roll is 10, which becomes the player's point. The game continues till either 10 or 7 appaers. Since 10 appears in the 6th roll, the player wins the game. (d) illustrates the case where the players' point in the first roll is 5. However 7 appears before 5 and the player loses the game.



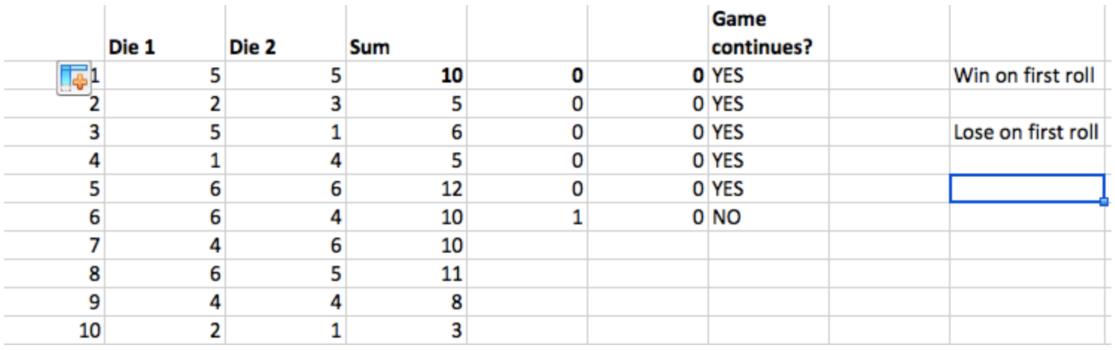
	Die 1	Die 2	Sum			Game continues?			
1	3	4	**7**	**1**	**0**	NO		Win on first roll	YES
2	5	2	7						
3	6	5	11					Lose on first roll	
4	5	1	6						
5	3	6	9						
6	6	3	9						
7	4	6	10						
8	3	3	6						
9	1	4	5						
10	6	4	10						

Figure 11 (a). Player wins on the first roll in the game of Craps



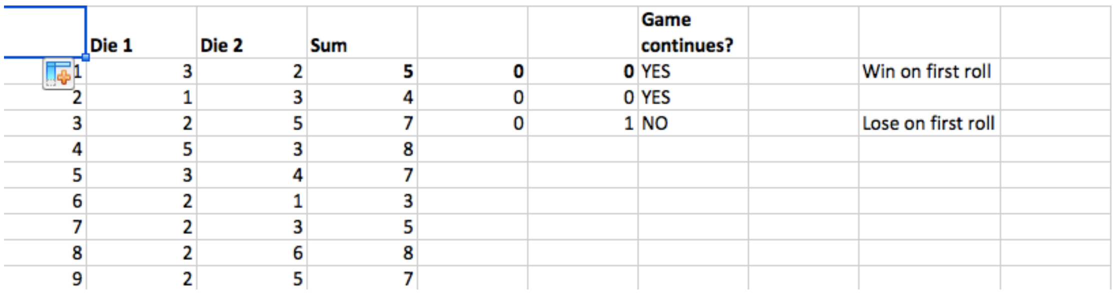
	Die 1	Die 2	Sum			Game continues?			
1	1	2	**3**	**0**	**1**	NO		Win on first roll	
2	4	5	9						
3	1	3	4					Lose on first roll	YES
4	5	5	10						
5	5	4	9						
6	4	2	6						
7	3	1	4						
8	5	4	9						
9	1	4	5						
10	6	3	9						

Figure 11 (b). Player loses on the first roll in the game of Craps



	Die 1	Die 2	Sum			Game continues?		
1	5	5	**10**	**0**	**0**	YES		Win on first roll
2	2	3	5	0	0	YES		
3	5	1	6	0	0	YES		Lose on first roll
4	1	4	5	0	0	YES		
5	6	6	12	0	0	YES		
6	6	4	10	1	0	NO		
7	4	6	10					
8	6	5	11					
9	4	4	8					
10	2	1	3					

Figure 11 (c). Player's point is 10 on the first roll and wins in 6th roll



	Die 1	Die 2	Sum			Game continues?			
1	3	2	**5**	**0**	**0**	YES		Win on first roll	
2	1	3	4	0	0	YES			
3	2	5	7	0	1	NO		Lose on first roll	
4	5	3	8						
5	3	4	7						
6	2	1	3						
7	2	3	5						
8	2	6	8						
9	2	5	7						

Figure 11 (d). Player's point is 5 on the first roll and loses in 3rd roll

Students simulated several games and used the data to compute the average number of throws required to win the game. It created much excitement and some students also attempted to wite a code for the game in Python programming language.

4. Concluding discussion

Despite attempts by various researchers there appears to be ambiguity about computational thinking and how it can be elicited in the mathematics classroom. This article attempts to propose a guide map for preparing and integrating CT based activities in mathematics. While designing the study the author (also the researcher) selected tasks based on students' knowledge and familiarity of mathematical content. Thus the fractal explorations tasks were assigned to grade 11 students as the topic of geometric sequences is taught in grade 11 whereas the game of Craps task was assigned to grade 12 students who study probability as a part of their curriculum. Since recursion has been identified as a key computational thinking skill, an attempt was made to include tasks which involve iteration and recursion. Thus explorations based on fractals and chaos was a natural choice. Further, modeling and simulation is an integral part of the computational thinking in mathematics and science taxonomy as proposed by Weintrop et. al. [5] and it became imperative to include modeling tasks in the study. The simulation of queues was a modeling exercise, which could be attempted by both grade 11 as well as grade 12 students. Further analyzing the game of Craps required simulation as well. Typically each CT task in the study was spread over four weeks. Students were assigned reading materials and exercises, which would help to investigate the problem at hand. They met after school on one weekday and on Saturdays. During these sessions they would engage in discussions and present their ideas to the whole group. Further the exploration of the problem using technology was also done during these sessions. Technology in the form of spreadsheets, graphics calculators and dynamic geometry software played a pivotal role in the exploration of these computational thinking tasks. These tools acted as *ampifiers* as they provided access to higher-level concepts and increased the scope of the tasks. For example, in the fractal explorations, students were able to explore the recursive construction at higher stages as they generated the geometric sequences numerically in Excel. In the chaos exploration, the graphical outputs (including cobweb plots) provided by the graphics calculator helped to develop a graphical insight of the problem. In the queue simulation task, the spreadsheet was critical in helping students understand the nuances of the queuing process. Technology also acted as a *reorganiser*, in the sense that the same sequence did not work for all tasks. Each exploratory task had its unique characteristics and a different sequence of explorations. In general students explored the tasks through multiple representations, that is, graphically, symbolically and numerically. They also developed competence in using the appropriate tool for exploration and appreciated the role of technology in the investigations. All through the explorations, students engaged in the processes of looking for patterns and invariances, selecting between representations and creating new ones, simplifying or generalising problems, making conjectures and even generating new questions for exploration all of which are important skills for computational thinking as well as mathematical learning.

The most encouraging aspect of the study was that the excellent student feedback. They expressed that such tasks should be included in the regular teaching of mathematics. They valued the opportunity of asking 'what if' questions during the interactive sessions and also the opportunity for discussions with their peers. Unlike regular school lessons, the outcomes of each task was not predefined. There was a sense of 'suspense' and there were many 'aha' moments which really appealed to students. They expressed a sense of ownership about their tasks and were responsible for their own learning. The tasks encouraged group work and everyone participated.

Despite the positive findings of the study there were some pitfalls. The explorations were time consuming and integration of the CT tasks needed to be planned carefully. Students needed to be motivated to try out problems which are considered to be 'beyond the school curriculum' and in this regard the teacher played a pivotal role. In general, teachers need to be convinced about the importance of integrating CT based tasks in mathematics learning and also need adequate professional development opportunities to enhance their knowledge and understanding about such tasks. In fact the inclusion of computational thinking in mathematics classrooms must receive the support of all stakeholders - students, teachers, school authorities as well as parents. It may be approptiate to conclude that the study described in this paper points to many benefits of integrating CT based activities in mathematics classrooms. In fact, mathematics provides many rich contexts for integrating computational thinking activities while computational thinking, in turn, enriches mathematics learning.

References

[1] Central Board for Secondary Education. (n.d.). Course structure of mathematics (class XI and XII). Retrieved from http://cbseportal.com/exam/Syllabus/cbse-11th-12th-2011-mathematics.

[2] Papert S (1972) Teaching children to be mathematicians versus teaching about mathematics. International Journal of Mathematical Education in Science and Technology 3(3):249–262.

[3] Papert S. (1980). Mindstorms: children, computers, and powerful ideas. Basic books, New York.

[4] Papert S (1996) An exploration in the space of mathematics educations. International Journal of Computers for Mathematics Learning 1(1):138–142.

[5] Weintrop, D., Behesthi, E, • Horn, M., Orton, k., Jona, K., Trouille, L. & Wilensky, U. (2016). *Defining Computational Thinking for Mathematics and Science Classrooms*, Journal of Science Education and Technology 25:127–147.

[6] Wing J.M. (2006). Computational thinking. Commun ACS. 49(3); 33 – 35.

[7] Wolfram, S. (2002). A new kind of science. 1st edn. Wolfram Media. Tokyo.