



**HAL**  
open science

## An analytical framework for programming tasks in mathematics textbooks

Mari Solberg Jensen, Antoine Julien, Abolfazl Rafiepour, Alexander  
Schmeding

► **To cite this version:**

Mari Solberg Jensen, Antoine Julien, Abolfazl Rafiepour, Alexander Schmeding. An analytical framework for programming tasks in mathematics textbooks. Thirteenth Congress of the European Society for Research in Mathematics Education (CERME13), Alfréd Rényi Institute of Mathematics; Eötvös Loránd University of Budapest, Jul 2023, Budapest, Hungary. hal-04417510

**HAL Id: hal-04417510**

**<https://hal.science/hal-04417510>**

Submitted on 25 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An analytical framework for programming tasks in mathematics textbooks

Mari Solberg Jensen<sup>1</sup>, Antoine Julien<sup>1</sup>, Abolfazl Rafiepour<sup>1</sup> and Alexander Schmeding<sup>1</sup>

<sup>1</sup>Nord University, Faculty of Education and Arts, Levanger, Norway; [antoine.julien@nord.no](mailto:antoine.julien@nord.no)

*The Nordic countries recently introduced programming as a new topic in school mathematics. Due to this reform, mathematics textbooks now include programming tasks. We propose in this article a framework for the analysis of programming tasks in mathematical textbooks. It seeks to classify programming tasks on the intersection of computational thinking and mathematics didactics. For this, we develop established frameworks of textbook analysis based on computational thinking. Beyond that, our framework emphasises the connection to mathematics education. Finally, we showcase the framework in examples from Norwegian mathematics textbooks for schools.*

*Keywords: Mathematics materials, programming, mathematics education, mathematics curriculum.*

In recent years, programming has been introduced as a new mandatory topic in the school curriculum of several of the Nordic countries. One main rationale to do so is the emphasis of 21st century skills, such as computational thinking (CT) (Wing, 2006), connected to programming and programming related activities. Denmark has chosen to integrate programming as part of a new subject (“Teknologifag”, see Elicer & Tamborg, 2022). In contrast both Sweden and Norway chose to integrate programming as a topic in the mathematics curriculum. Albeit there being several subjects in Norwegian schools which incorporate elements of programming, mathematics is the main responsible for education in programming. Thus, mathematics teachers face the challenge to teach programming in relation with mathematics and in a mathematical context (cf. Stigberg & Stigberg 2020, p. 494). These changes in the curriculum are now being implemented by a workforce with, on average, little to no knowledge of programming.

## Theoretical background

When a new curriculum is implemented, the textbooks are updated as well. In mathematics education, textbooks play an important role (Gueudet et al., 2012; Li et al., 2009) for several reasons. First, because of their close relation to classroom instruction (Valverde et al., 2002). If a topic is presented in the textbook, it will very likely be introduced in the classroom (Johansson, 2006). They also often specify how classroom lessons can be structured with suitable exercises and activities (Valverde et al., 2002), and many teachers use the textbook mainly to provide exercises for their students (Pepin & Haggarty, 2001). Second, mathematics textbooks both represent and translate the abstract curriculum into operations that students and teachers can carry out (Valverde et al., 2002). The textbook therefore has a guiding function (Rezat, 2012, p. 231), especially when teachers are in a position where they both need to understand and implement a new curriculum.

Viewed against the background of the curriculum reform in Norway, the paramount importance of textbooks and their tasks leaves us at an interesting crossroads. On one hand the curriculum demands the implementation of programming. It includes for almost every schoolyear at least one learning goal directly connected to programming and CT. On the other hand, programming tasks were not traditionally incorporated in Norwegian mathematics textbooks for schools. Textbook authors and

publishers were thus faced with the challenge of introducing new types of activities and tasks in their books. At the same time, many of the learning goals related with CT are extremely broad. For example, “Explore mathematical properties and connections using programming” (UDIR 2022, our translation) is an aim for the 10<sup>th</sup> grade. Here textbook authors need to decide how to interpret the aim when creating tasks. Textbooks in Norway are not checked by a government agency before being admitted for use in schools. Hence, there is no standard analytical framework to analyse them and it is important to develop independent analytical frameworks for their content.

In the present paper, we propose an analytical framework for programming tasks. Our aim is to create a tool for textbook analysis of programming tasks specifically in mathematics textbooks. The purpose of the analytical tool is threefold: It will allow one to assess the current state of programming tasks in the textbooks for mathematics education. Further, it can be used as a framework for professionals seeking to assess tasks for use in teaching. Finally, it will provide a theoretical basis to develop rich tasks which connect programming and mathematics in a school context. Before we outline the core tenets of our framework, let us comment on previous work available in the literature. Programming was also newly introduced in Sweden and Denmark. Elicer and Tamborg (2022) analyse and classify tasks in the Danish context with respect to connections to mathematics in six categories. Their scheme is too coarse for our purpose. For the Swedish context Bråting and Kilhamn (2022) have investigated elementary mathematics textbooks for grades 1–7. Their coding scheme combines a CT framework and Benton et. al.’s (2016) 5Es to describe both CT aspects and the connection to mathematics. Our approach is inspired by their framework but develops it further as we will outline now.

### **The analytic framework for programming tasks in mathematics textbooks**

We propose to analyse the tasks on several levels. These levels are being developed from a theoretical perspective with a view towards the curricular dimensions put forward by the Norwegian government. The different levels we propose are the following:

1. Formal properties related to programming.
2. Connection to computational thinking (CT).
3. Accessibility and potential of the task (Low floor; High ceiling).
4. Connection between programming and mathematics

The first level of our framework is purely descriptive and classifies formal aspects via the following descriptors: **Type of programming**, i.e., analog programming, block-based (e.g., Scratch) or text based programming (e.g., Python). Is the **programming optional or mandatory** to complete the task? Does the task **involve tangibles** such as Lego Mindstorms, BBC Microbit, BeeBots or similar.

The second level of our framework investigates the connection of the task to the concept of CT. As Shute et. al. (2017) remark, there is no consensus as to the definition or model of CT. However, practical considerations informed our choice of a CT model. The curricular goals involving CT and programming are similar in the Norwegian and Swedish systems. Hence a natural starting point for our categories is the framework of Bråting and Kilhamn (2022) for the Swedish context. They describe CT as a multifaceted concept divided into several categories described each by an action

verb. We choose the same verbs as starting point. However, several adjustments of the model were necessary. As a first change, we chose to add the category “explore” from the model of Benton et al. (2016). This addition helps to accommodate open tasks which encourage pupils to investigate different aspects of mathematics or programming. Tasks involving finding mistakes are now also part of the exploring category in our framework. Norwegian mathematics textbooks feature many tasks where pupils must run a code listed in the textbook. These tasks we categorized as “Follow a procedure”. Here the instruction of running the code should be understood as a stepwise instruction: the pupils are not creating the code by themselves. Summing up, the chosen categories for level two are built to encapsulate aspects of CT. The descriptors we use are defined in Table 1.

**Table 1. Categories for the second level of coding: connection to computational thinking**

Follow a procedure	Follow stepwise instructions, plot and run a predefined program code or code sequence and repeat or to continue a pattern.
Figure out	Work out the procedure, the rule or the pattern that generates an outcome.
Form and create	Give instructions, create a pattern, write a code or a code sequence, make a program or represent with symbols
Explain	Explain/describe a procedure, a rule, a code, a pattern, or a concept using words and natural language.
Envisage	Predict or reflect on what will happen, reflect on possible outcomes when conditions or values change.
Explore	Explore the tool and the ideas in a task, try things, compare, strategies or outcomes and debug errors.

The next level of our framework addresses the accessibility and potential of the programming tasks. Exercises in Norwegian textbooks typically feature a difficulty progression starting easy and raising the difficulty throughout subtasks. Due to the novelty of programming in textbooks, the difficulty level and accessibility of the task is an important indicator for the choices textbook authors have made concerning the placement of programming in their material and with respect to the curricular aims. In this level, we will evaluate whether the tasks have “low-floor and high-ceiling”, a notion attributed to Seymour Papert, and which Boaler et al. (2019) define as tasks which “provide the best conditions for engaging all students ... and students can take the ideas to high levels”. A prerequisite to high ceiling is richness of the task to allow a meaningful engagement with CT and mathematical ideas.

The first three levels describe the programming content of the task: in formal terms, in relationship with actions related to CT, and whether it has a low-floor and high-ceiling. One of our aims, however, is also to investigate tasks in terms of their connection to mathematics. Indeed, the curriculum introduces programming as a part of mathematics, and the textbooks we study are mathematics textbooks. The fourth level therefore focusses on the mathematical aspects of the tasks. We ask how the task connects to big ideas in mathematics. As Charles (2005, p.10) states, a big idea “is central to the learning of mathematics, ... [and] links numerous mathematical understandings into a coherent whole” (cf. also Tout & Spithill, 2015). In Norway, several curricular aims are connected to specific mathematical goals (e.g., in Grade 9, programming should be used to explore statistics and

probability). However, other learning goals are broad and not explicitly connected to a specific topic. Especially for Grade 10, the curriculum states the aim “Explore mathematical properties with the help of programming”. Hence the choices made for the mathematical content in programming tasks are directly influenced by the textbook authors interpretation of the curricular aims. Thus the fourth level asks the questions: Does the task connect to mathematical big ideas (or not)? Is programming a natural tool to explore these big ideas? Does the task use the potential of programming in the investigation, or is the connection surface-level? These questions (especially the second) can be related to the concept of authenticity of the task, see Vos (2018). Another useful perspective is instrument theory (Rabardel, 2002) which investigates an activity and the tools used in it with a view towards constraints, enabling, potentials, and affordances with respect to learning. This theory provides a lens to understand the role of technology in the teaching of mathematics.

The four levels described are the core of our theoretical framework. As outlined, the model is rooted in the CT theory, instrument theory and didactics of mathematics. Tasks are investigated for their connections between programming and mathematics. Special emphasis is placed on the roots of the model in the mathematics didactics which stresses the importance of level 4 for our framework.

### **Application of the framework to examples from Norwegian textbooks**

We illustrate the application of our model in two tasks from 10<sup>th</sup> grade mathematics textbooks for Norwegian schools. Tasks (with translation) are shown respectively in Figure 1 and 2.

#### **Example 1: Task from the textbook Matemagisk 10**

The relevant task is reproduced together with a translation in Figure 1 below.

**Overview:** The code defines the function  $f: \mathbb{R} \rightarrow \mathbb{R}, x \mapsto 2x + 1$  as a Python function, then plots it.

**Level 1:** The task involves mandatory text-based programming in Python without the use of tangibles.

**Level 2: a** The word “Execute” signals that one should copy the code and try it on a computer. This is a purely formal process which is preparatory and a prelude to the directive “Explain”. Hence, we chose to classify the task both as “Follow a procedure” and as “Explain”. **b** The task is of the “Form and Create” and the “Explain” type. While it asks to execute a program, this is secondary to the open directive to modify the program for a new function type. **c** This task fits to the “Explore” category. It asks the user to exchange a number in the code and explore the consequences to the graph. **d** Is of the “Explore” type. **e** Is asking for an expansion of the code and therefore “Form and Create”.

**Level 3:** The difficulty progression is typical for Norwegian textbooks. The first tasks are simple and straightforward, asking only for execution and small changes. The later tasks open up and invite to explore possibilities to change the program. This progression therefore provides a low entry level for all students in the first sub-task, while the later tasks can be classified as having a “high ceiling.”

**Level 4:** The task is set in the context of representation of functions. In **b/d** it asks to plot other functions than the original one, but there is no investigation of the function itself – just focusses on the coding. Here the task illustrates the plotting capabilities of Python (enablement perspective). In **c**


<p><b>OPPGAVE 23.88</b></p> <p>Løsn oppgaven skal vi lage et program i Python som plottet grafen til en funksjon. Da trenger vi en liste med x-verdier og en liste med tilhørende funksjonsverdier.</p> <pre> 1 from pylab import * 2 3 def f(tall): 4     svar = 2*tall + 1 5     return svar 6 7 x = linspace(0, 50, 101) 8 y = f(x) 9 plot(x, y) 10 show() </pre> <p>Kommandoen <code>linspace(0, 50, 101)</code> lager en liste med 101 jevnt fordelte tall, der første tall er 0 og siste tall er 50</p> <p>På linje 8 sender vi en liste inn i funksjonen f. Da returneres en liste med tall, der hvert tall hører til tallet på tilsvarende plass i lista x.</p>  <ul style="list-style-type: none"> <li>a Kjør programmet og forklar hva som skjer på hver linje.</li> <li>b Erklær funksjonsuttrykket på linje 4 til en eksponentalfunksjon. Kjør programmet, og beskriv hva som skjer.</li> <li>c Endre tallet 101 på linje 7 til andre tall. Hvordan påvirker dette grafen?</li> <li>d Flytt med andre funksjonsuttrykk du har arbeidet med i dette kapitlet.</li> <li>e Ihvildd programmet slik at det plottet to grafer i det samme koordinatsystemet.</li> </ul>	<p>In this exercise we will create a program in Python which plots the graph of a function. For this we need a list of x-values and a list with the corresponding function values.</p> <p>a Execute the program and explain what happens in every line of code.</p> <p>b Change the function in line 4 to an exponential function. Execute the program and describe what happens.</p> <p>c Change the number 101 in line 7 to another number. How does this affect the graph?</p>
<p>d Try the code with other types of functions we have been working with in this chapter.</p>	
<p>e Expand the program such that it plots two graphs into the same coordinate system.</p>	

Figure 1: Task from Matemagisk 10 (Kongsnes & Wallace, 2021), instructions translated on the right

<p>Samarbeid to og to. I dette programmet leter vi etter tilnærmede verdier for nullpunktene til funksjonen <math>f(x) = x^2 - 2x - 1</math>.</p> <pre> 1 from pylab import * 2 3 def f(x): 4     return x**2-2*x-1 5 6 x=linspace(-10,10,100) 7 y=f(x) 8 9 ylim(-3,6) 10 xlim(-4,4) 11 akser=.gca() 12 akser.spines["bottom"].set_position("zero") 13 akser.spines["left"].set_position("zero") 14 akser.spines["top"].set_visible(False) 15 akser.spines["right"].set_visible(False) 16 17 plot(x,y) 18 grid() 19 show() 20 21 antall=int(input("Hvor mange nullpunkter skal vi finne?")) 22 23 for i in range(0,antall,1): 24     print("\n02 finner nullpunkt nr.",i+1) 25     under=float(input("Oppgi nedre grense:")) 26     over=float(input("Oppgi øvre grense:")) 27 28     for m in range(0,6,1): 29         x_verdi=(under+over)/2 30         if f(under)*f(x_verdi)&lt;0: 31             over=x_verdi 32         else: under=x_verdi 33 34     print("Nullpunkt",i+1, ":", round(x_verdi,4)) </pre> <ul style="list-style-type: none"> <li>a Kjør programmet og finn ut hvordan det fungerer. Hvorfor er grafplotting tatt med når målet er å finne nullpunktene?</li> <li>b I programmet brukes «halveringsmetoden» for å finne nullpunkter. Forklar hvordan metoden fungerer, og hvorfor dette er et godt navn på metoden.</li> <li>c Skriv inn forklaringstekster til hver del av programmet (#-linjer), slik at funksjonaliteten i programmet kan forstås bedre.</li> <li>d Forbedre programmet, slik at nullpunktene finnes med større nøyaktighet. Forklar hvordan dere velger å forbedre, og hvorfor.</li> <li>e Forbedre programmet slik at det             <ol style="list-style-type: none"> <li>1 blir mer generelt og kan finne nullpunkter for en kvadratisk funksjon som velges av brukeren</li> <li>2 blir mer robust overfor feil innstillinger av brukeren</li> </ol> </li> </ul>	<p>Work in pairs. In the program we are seeking approximative values for zeroes of the function <math>f(x) = x^2 - 2x - 1</math></p> <p>a Execute the program and find out how it works. Why is graph plotting included when our aim is to find zeroes of a function?</p> <p>b The program uses the “bisection method” to find zeroes of a function. Explain this method and why this is a good name for the method.</p> <p>c Write an explanation text for every part of the program (#-lines), such that the functionality in the program is better understandable.</p> <p>d Improve the program such that all zeroes are computed with higher accuracy. Explain how you improved the program and why.</p> <p>e Improve the program such that it...             <ol style="list-style-type: none"> <li>1 becomes more general and can find zeroes of quadratic functions the user can choose</li> <li>2 is more robust against input errors by the user</li> </ol> </p>
---	---

Figure 2: Task from Maximum 10 (Tofteberg et al., 2021), instructions translated on the right

the change of parameter changes the amount of points used in the plot. It connects to a technical aspect of plotting in Python but not to the concept of function. Summing up, this programming task is only loosely rooted in mathematics and does not connect to a big idea (such as the idea of functions and relations). There are good reasons to include plotting of functions in Python (see example 2). However, Python is impractical for plotting compared to computer algebra systems such as Geogebra.

### Example 2: Task from the textbook Maximum 10

The relevant task is reproduced together with a translation in Figure 2 above.

**Overview:** The code defines  $f: \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^2 - 2x - 1$  as a Python function and plots it. Then the user needs to input how many zeroes are to be searched. Repeating (iterating over  $i$ ) as many times as instructed, the user gives bounds for the  $i^{\text{th}}$  zero; then, the program finds an approximation of it (checking for sign changes of function values) through bisecting the interval a fixed number of times.

**Level 1:** The task involves mandatory text-based programming in Python without the use of tangibles.

**Level 2: a** This task starts with the word “Execute” which indicates to copy the code, and according to this, we classify the task as “Follow a procedure”. One is also tasked to find out how the program works. Since we expect pupils to communicate their findings to answer this question, we choose to categorize this task as “Explain”. **b** Task b asks one to explain why this method works, and why the name of the method is suitable. Even though the word explain could indicate the “Explain” category, one needs to work out the procedure first. Therefore, this task fits in the “Figure out” category. **c** This task fits to the “Explain” category. It asks the user to write explanatory texts explaining each part of the program. **d** Tasks where one is asked to improve a program, could belong in two categories. Here improving means to calculate more accurately. Hence understanding is indispensable before one can create new code. Due to this, we categorize this task as both “Figure out” and “Form and create”. In addition, the task asks for an explanation of how one has improved the program and why, which puts the task in the “Explain” category. **e**. Like the previous task, both “Figure out” and “Form and create”.

**Level 3:** This task will most probably be difficult or even impossible for average pupils in Norwegian 10<sup>th</sup> grade. Task **a** has elements which only ask to execute the code, but then asks for an explanation of the program which most pupils will find difficult (we investigated this task with pre-service teachers and most of them failed here without scaffolding). Interestingly, task **b** presents a difficulty spike in the exercise as it asks for an explanation of the numerical algorithm employed. In Norwegian the term *halveringsmetode* (halving method) is more transparent than the English “bisection method”. However, to answer **b** one needs to identify the part of the program executing the algorithm (the for-loop). Then the algorithmic meaning of the loop needs to be extracted (the task asks for explanation, it is not sufficient to answer that something is halved, whence the name of the method). Subtask **c** is easier again, as it asks for grouping the code in blocks and explain their general significance (which would have been useful for part b). Tasks **d** and **e** are in line of the difficulty progression more open.

**Level 4:** The task is exhibiting a standard numerical algorithm to find approximations to zeroes of a function. Hence it connects to the ideas of estimation and numerical algorithms. Numerical algorithms, while not dependent on programming and computer assistance per se, were not part of the usual mathematics curriculum. It is much more convenient to implement numerical methods with

the help of programming. Hence this task enables the inclusion of numerical methods in school mathematics. Moreover, it might be argued that programming provides the potential to explore ideas from numerical mathematics at all (enabling and potentiality in the sense of instrument theory).

## Conclusion and outlook

Computer programming has not traditionally been a school subject in the Nordic countries. There is therefore an acute need to evaluate and develop teaching resources which are appropriate to teach this topic in schools. More importantly maybe, there is a need to develop a content-specific pedagogy in programming (or programming didactics) which is distinct from higher-education didactics. As a matter of analogy, we will point out the distinction between school mathematics and mathematics as a research field: both have distinct goals; the construction of knowledge in the classroom differs very much from the construction of knowledge in the field; learning and teaching these topics require different approaches and conceptual constructs. Though school mathematics and mathematics need to be all but independent, the distinction between mathematics and school mathematics – both the subject matter, and its teaching and learning – is well established. Such a distinction does not exist to such an extent for school programming, though having now programming as a school subject so systematically and to such a scale creates a need for new conceptual frameworks. With this paper, we bring our modest contribution to the continued development a specific didactics of school programming. Our framework for evaluating programming tasks aims at recognizing the merits of a task both as an instrument for learning programming, and as an instrument for learning mathematics. Our short- to mid-term goal is to apply this framework to evaluate existing textbooks and provide a tool for editors and educators to evaluate the potential of the tasks they develop. In this process, we expect that aspects of our framework will need to be refined. In the long run, we will use the resulting updated framework to develop and test programming tasks with high mathematical and CT potential.

## References

- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016). Building mathematical knowledge with programming: Insights from the ScratchMaths project. In A. Sipitakiat & N. Tutiya-phuengprasert (Eds.), *Constructionism in Action 2016, Conference Proceedings* (pp. 25–32). Suksapattana Foundation.
- Boaler, J., Munson, J., & Williams, C. (2019). *Mindset mathematics: Visualizing and investigating big ideas, grade 7*. John Wiley & Sons.
- Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: Investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research*, 66(4), 594–609. <https://doi.org/10.1080/00313831.2021.1897879>
- Charles, R. I. (2005). Big ideas and understandings as the foundation for elementary and middle school mathematics. *Journal of Mathematics Education Leadership*, 7(3), 9–24.
- Elicer, R., & Tamborg, A. L. (2022). Nature of the relations between programming and computational thinking and mathematics in Danish teaching resources. In U. Y. Jankvist, R. Elicer, A. Clark-Wilson, H.-G. Weigand, & M. Thomsen, *Proceedings of the 15th International Conference on Technology in Mathematics Teaching (ICTMT 15)* (pp. 45–52). Aarhus University. <https://doi.org/10.7146/aul.452>



- Gueudet, G., Pepin, B., & Trouche, L. (2012). Introduction. In G. Gueudet, B. Pepin, & L. Trouche (Eds.), *From text to 'lived' resources: Mathematics curriculum materials and teacher development*. (pp. ix–xiii). Springer.
- Johansson, M. (2006). Textbooks as instruments: Three teachers' ways to organize their mathematics lessons. *Nordic Studies in Mathematics Education*, 11(3), 5–30.
- Li, Y., Chen, X., & An, S. (2009). Conceptualizing and organizing content for teaching and learning in selected Chinese, and US mathematics textbooks: The case of fraction division. *ZDM – Mathematics Education*, 41(6), 809–826. <https://doi.org/10.1007/s11858-009-0177-5>
- Kongsnes, A. L., & Wallace, A. K. (2021) *Matemagisk 10*. Aschehough undervisning.
- Pepin, B., & Haggarty, L. (2002). An investigation of mathematics textbooks and their use in English, French and German classrooms: Who gets an opportunity to learn what? *British Educational Research Journal*, 28(4), 567–590. <https://doi.org/10.1080/0141192022000005832>
- Rabardel, P. (2002). *People and technology: a cognitive approach to contemporary instruments*. Université Paris 8. <https://hal.science/hal-01020705>
- Rezat, S. (2012). Interactions of teachers' and students' use of mathematics textbooks. In G. Gueudet, B. Pepin, & L. Trouche (Eds.), *From text to 'lived' resources: Mathematics curriculum materials and teacher development* (pp. 231–246). Springer. [https://doi.org/10.1007/978-94-007-1966-8\\_12](https://doi.org/10.1007/978-94-007-1966-8_12)
- Shute, V. J., Sun, C., Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Stigberg, H., & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, 18(4), 483–496. <https://doi.org/10.1177/1478210319894785>
- Tofteberg, G. N., Tangen, J., Bråthe, L. T., & Stedøy, I. (2021). *Maximum 10*. Gyldendal.
- Tout, D., & Spithill, J. (2015). Big ideas in mathematics teaching. *The Research Digest*, QCT, 2015 (11). <http://www.qct.edu.au>
- Utdanningsdirektoratet (2020). *Læreplan i matematikk 1.-10. trinn* (MAT01-05) [Curriculum in maths grades 1-10]. <https://www.udir.no/lk20/mat01-05>
- Valverde, G., Bianchi, L. J., Wolfe, R., Schmidt, W. H., & Houang, R. T. (2002). *According to the book: Using TIMSS to investigate the translation of policy into practice through the world of textbooks*. Springer.
- Vos, P. (2018). “How real people really need mathematics in the real world”—Authenticity in mathematics education. *Education Sciences*, 8(4), Article 195. <https://doi.org/10.3390/educsci8040195>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>