Programming as a distinct knowledge domain in mathematics education

An empirical reinvestigation of TPACK

HELGE JEPPESEN AND NILS HENRY RASMUSSEN

Several teachers experience difficulties teaching programming in school mathematics. While the Technological Pedagogical Content framework (TPACK) has previously described links between pedagogical, content and technological knowledge for incorporating technology in competencies for teaching mathematics, these links must be reevaluated after new programming elements have been introduced in the Norwegian national curriculum. Using teachers' self-reported knowledge, 127 answers were analysed through confirmatory and exploratory factor analysis. Results show strong loadings for technological knowledge, but weak associations to pedagogical knowledge, indicating a separation with programming constructs. Our findings challenge the notion of programming as merely a technological component, suggesting programming should be considered a partially separate domain in TPACK.

The integration of technology into educational practices has gained significant attention globally, driven by the evolving digital landscape and the increasing emphasis on digital competencies in national curricula (Elicer et al., 2023; Forsström & Kaufmann, 2018; Geraniou et al., 2024; Ye et al., 2023). In the Norwegian context, the Knowledge Promotion Reform 2020 (Ministry of Education and Research, 2020b) has reinforced the need for teachers to develop proficiencies in both content and digital skills, particularly with respect to teaching programming within mathematics education (Ministry of Education and Research, 2020a). Programming is defined as the process of solving a problem by analysis, design and implementation (Bocconi et al., 2022). Despite these advancements, research suggests that newly qualified teachers struggle with fully integrating programming and computational thinking into their mathematics teaching, often due to gaps in their professional digital competencies (PDC), which include the intersection of technological, pedagogical, and

Helge Jeppesen, *Western Norway University of Applied Sciences*
Nils Henry Rasmussen, *Western Norway University of Applied Sciences*

Jeppesen, H. & Rasmussen, N. H. (2026). Programming as a distinct knowledge domain in mathematics education: an empirical reinvestigation of TPACK. *Nordic Studies in Mathematics Education*, 31 (1), 49–69.

content knowledge (Gudmundsdottir & Hatlevik, 2018). Computational thinking refers to understanding and solving problems in accordance with digital technologies (Bocconi et al., 2022).

The technological, pedagogical, and content knowledge (TPACK) framework emphasises the importance of a synergistic understanding of content, pedagogy, and technology, recognising that teachers must develop a deep, interconnected knowledge base to use technology in ways that enhance student learning (Koehler & Mishra, 2009). The TPACK framework was chosen as recent programming-specific extensions explicitly build on TPACK rather than replace it as the baseline for programming (Karlsen et al., 2025). Moreover, TPACK is the most widely used framework for knowledge needed to integrate technology in mathematics teaching (Gonscherowski & Rott, 2025). Studies have shown that teachers' self-perceived competencies in integrating technology into mathematics is often lacking (Gudmundsdottir & Hatlevik, 2018), particularly in the domain of programming, where traditional pedagogical approaches may not adequately support the use of modern digital tools. This study will employ a quantitative approach to analyse the relationships between components of this framework and teachers' perceived ability to incorporate programming into their teaching.

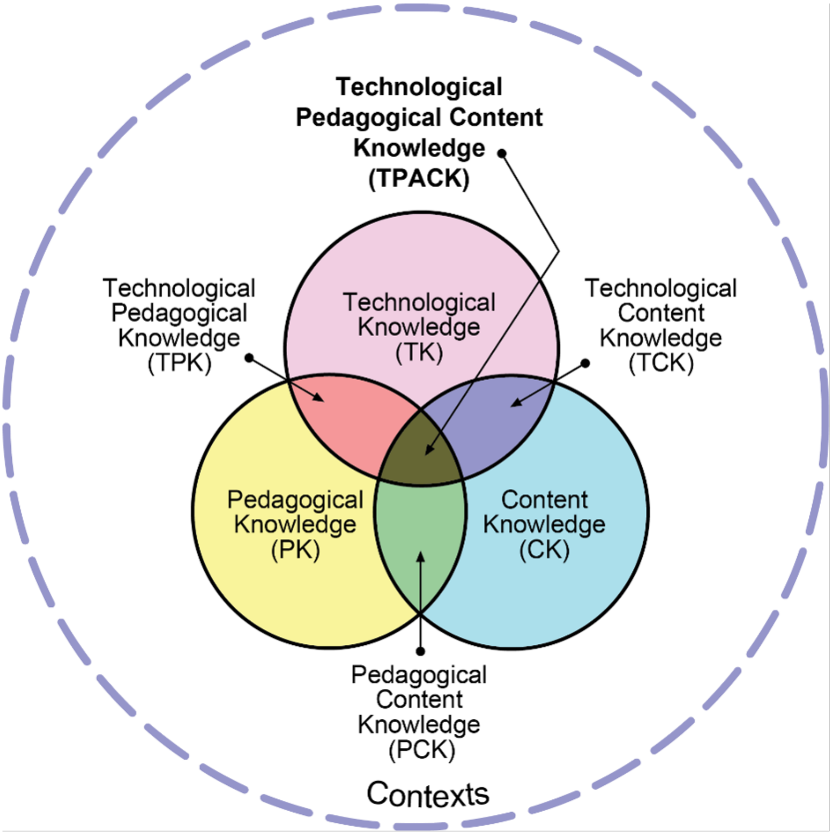
As the Knowledge Reform 2020 places strong emphasis on digital competency (Ministry of Education and Research, 2020b), teachers are required not only to master the use of technology, but also to integrate programming into subjects like mathematics in ways that promote conceptual understanding and critical thinking. The knowledge reform in mathematics incorporates programming as a natural part, while earlier research shows low programming knowledge (Gudmundsdottir & Hatlevik, 2018). There is thus a need to examine how programming is linked to pedagogy and content knowledge among teachers.

In this paper, we ask the following research question: What is the relationship between self-perceived knowledge in pedagogy, programming and mathematics among teachers in lower secondary school?

Theoretical framework

Technological pedagogical content knowledge (TPACK) builds on Shulman's (1986) construct of pedagogical content knowledge by incorporating the technological dimension into teacher knowledge (Koehler & Mishra, 2009). This framework conceptualises the intersection of technological knowledge (TK), pedagogical knowledge (PK) and content knowledge (CK) as a distinct and integrated domain that drives effective technology-enhanced teaching. The value of TPACK lies not in these domains in isolation, but in their synergy to complement each other towards a teaching design where TK, PK and CK mutually reinforce each other dynamically, as shown in figure 1.

Figure 1. *The TPACK framework (reproduced with permission of the publisher, © 2012 by tpack.org)*



Despite its prominence, TPACK has been problematised for its theoretical complexity, lack of conceptual precision, and challenges in reliable operationalisation (Fabian et al., 2024; Li et al., 2025; Schmid et al., 2024). These limitations are further exposed in contexts where technology constitutes a core element of disciplinary content, such as programming, where technological and content knowledge are intertwined. Karlsen et al. (2024) propose a programming-specific framework that extends beyond TPACK to capture specialised competencies required for codingbased instruction in science and technology. Research within informatics and computer science education reveals that TPACK lacks the level of detail to account for technology to serve as both the technological tool and disciplinary content (Giannakos et al., 2015; Ma et al., 2025). Giannakos et al. (2015) confirmed the existence of technological-pedagogical content knowledge as a separate competency dependent on the other aspects of the model, but found less distinctions between TK and CK, while Ma et al. (2025) distinguishes between computer-science specific knowledge domains to identify the insufficiency of generic TPACK to capture programming-specific competencies. It seems to be consensus that the core domains of TPACK are essential, while also revealing uncertainties when the original model is introduced when programming is substituted for technology. From these studies emerges a need for investigating whether programming in mathematics can be coherently situated within the TPACK framework.

Core domains of TPACK

Technological knowledge (TK) refers to teachers' understanding of digital tools and systems, and their capacity to adapt these into their pedagogical practices (Koehler & Mishra, 2009). In a programming and informatics context, TK must also encompass computational thinking (CT) such as abstraction, decomposition, algorithmic design and debugging (Grover & Pea, 2013). TK does not operate as merely a supportive tool, but becomes the content teachers are expected to teach. This shift reveals a limitation of the generic definition of TK since programming requires both technology and content knowledge; hence the strict boundaries between what can be defined as knowledge of technology and content cannot be applied (Fabian et al., 2024; Schmid et al., 2024). Prior research suggests that TK in programming contexts is a specialised form of disciplinary knowledge, requiring an understanding of both technology and content in order to be explained (Giannakos et al., 2015).

Pedagogical knowledge (PK) refers to general strategies for teaching and learning, classroom management and assessment (Koehler & Mishra, 2009). While PK remains essential for creating learning environments, Li et al. (2015) show that PK's unique contribution to TPACK is less pronounced than TK and CK in programming-rich environments, making PK dependent on strong integration of TK and CK. From this, PK can function as a carrier for applying this integrated knowledge, rather than as an independent driver of technology-enhanced teaching.

Content knowledge (CK) encompasses the concepts, structures and epistemologies of a discipline (Koehler & Mishra, 2009). For programming used in mathematics education, CT functions as a bridge between mathematics and programming, enabling teaching through code (Grover & Pea, 2013). Karlsen et al. (2024) argue for a programming-specific knowledge framework because CK and TK converge when technology is the content. For CK in a programming context, TK and CK are mutually reinforcing to the point where they may operate as an integrated knowledge domain (Li et al., 2025; Ma et al., 2025).

Computational Thinking (CT) has been recognised as a key competence in education (Kelentri et al., 2017; Ministry of Education and Research, 2020b). A core understanding from Grover and Pea (2013) involves abstraction, algorithmic design and debugging, elements that could be argued to be core skills within mathematics. Within mathematics and informatics, CT functions as a cognitive skillset positioning programming as both technological and disciplinary content. This dual role of TK and CK exposes limitations in TPACK where the framework more clearly separates technology, pedagogy and content (Koehler & Mishra, 2009), leading to a lack of conceptual precision where technology itself constitutes disciplinary knowledge. Previous research finds TPACK dimensions to be underspecified (Fabian et al., 2024; Schmid et al., 2024), difficult to operationalise (Li et al., 2025) and problematic in programming contexts (Giannakos et al., 2015; Ma et al., 2025), leading to a need for programming-specific knowledge frameworks (Karlsen et al., 2025).

Programming seems to extend to mathematics, relying on abstraction, logic and formal reasoning (Grover & Pea, 2013), yet programming also involves domain-specific practices like syntax, debugging and algorithmic design that mathematics alone cannot provide. Prat et al. (2020) found aptitude and linguistics to be stronger predictors of programming than numeracy, while Graafsma et al. (2023) found that both algebraic reasoning along with grammar and vocabulary skills predicted programming competence. An important finding towards mathematics education is the positive outcome of programming to learn mathematics when tasks were designed to highlight mathematical structures in code (Benton et al., 2018; Graafsma et al., 2023). In contrast, mathematics outcomes were negative when this clear alignment did not clearly coincide (Laurent et al., 2022). This transfer between mathematics knowledge and programming is neither automatic nor symmetrical, but contingent on task design, representation clarity and teacher mediation (Benton et al., 2018; Forsström & Kaufmann, 2018; Nyman et al., 2025). Thus, teachers need both CK and TK in order to implement CT in their teaching.

Methodology

To assess teachers' professional digital competencies, specifically focusing on their ability to integrate programming into mathematics instruction, an adapted version of the survey instrument by Schmidt et al. (2019) was chosen. The original framework was designed to measure the seven domains of TPACK on a five-point Likert scale (Likert, 1932), to assess teachers' perceived confidence in related practices. Its simplistic and concise assessment made it possible to adapt TPACK into a focus on programming. Our survey questions, ranging from 1 (completely disagree) to 5 (completely agree) allowed for nuanced self-assessment. All items were critically assessed by both authors, and the survey was subjected to pilot testing with a small group of in-service teachers. The final questionnaire was adapted based on feedback to remove ambiguity and rewording for clarity.

The survey was distributed to a national sample consisting of 1102 Norwegian schools that offered mathematics lower secondary level education, resulting in 127 responses from Norwegian mathematics teachers. The survey was distributed via SurveyXact, an online survey platform, allowing for efficient data collection, participant accessibility and anonymous participation. Participants were given clear instructions on how to complete the survey, including an explanation of the purpose of the study and the voluntary participation in accordance with ethical guidelines.

Internal consistency was evaluated using both Cronbach's alpha and McDonald's omega, coefficients measuring how well items within the same scale correlate with each other, where both values being over 0.70 are commonly accepted as consistent measurements (Cheung et al., 2024). To examine construct validity, whether the survey measures the intended theoretical model, we conducted a confirmatory factor analysis (CFA). To evaluate model fit we used values by Hu and Bentler (1999). The specified model was compared with a null model with no relationships with Comparative Fit Index (CFI) and Tucker-Lewis Index (TLI), where values above 0.90 indicate acceptable fit. Root Mean Square Error of Approximation (RMSEA) estimates how well our model fits the data, with a cutoff value close to 0.06 indicating good fit. To examine the average difference between observed and predicted correlations, we employed Standardised Root Mean Square Residual (SRMR), where values below 0.08 are desirable. To estimate explained variance of the model, Goodness-of-Fit Index (GFI) should be above 0.90. Chi-Square tests difference between observed and expected covariance and should be small to indicate model fit using the Maximum Likelihood (ML) method. The Chi-Square test indicates a non-significant model fit, but due to its sensitivity to sample size and complexity, we chose to base findings on a combination of fit indices (Kenny et al., 2015). Using ML estimation, model fit was evaluated with multiple indices (Hu & Bentler, 1999; Marsh et al., 2004). CFA demonstrated acceptable fit across all indices (CFI = 0.917, RMSEA = 0.087, SRMR = 0.059, TLI = 0.897, 188 degrees of freedom and Chi-Square < 0.001). While not all results are within cutoff points, this is expected given the model complexity and small sample size. Due to a modest sample size, Chi-Square tends to reject acceptable models with small N (Kenny et al., 2015). RMSEA can overestimate misfit in models with small N and high degrees of freedom, making slight deviations above 0.08 less concerning with our objective to seek a trend in the data (MacCallum et al., 1996). To account for a modest sample size, we relied on multiple fit indices and prioritised factor loadings above 0.60, following situation-based recommendations where strong loadings and a parsimonious model can show stable estimates (Wolf et al., 2013).

Based on results from CFA with factor weightings being either strong or weak, Exploratory Factor Analysis (EFA) was conducted in an attempt to verify three distinct factors (technology, pedagogy and content knowledge), without imposing the predefined TPACK structure. To confirm that the dataset was suitable for factor analysis, we used the Kaiser-Meyer-Olkin test (KMO) to measure whether items share enough common variance to extract factors. Values above 0.80 indicate suitability for factor analysis (Kaiser, 1974). TPACK assumes conceptually related domains, indicating correlations between factors, favouring oblique matrix rotation. This rotation with Promax indicated minimal inter-factor correlations, leading to results being presented with Orthogonal Varimax Rotation (Costello & Osborne, 2005), a rotation that makes factors easier to interpret by maximising separation among loadings. A visual way to determine factors is the Scree test, which plots eigenvalues to examine where the graph levels out (Ledesma et al., 2015). Parallel analysis retains factors with eigenvalues exceeding randomly generated data to rule out that results have come from chance (Dinno, 2009). Minimum Average Partial (MAP) selects the solution maximising Average Squared Partial correlations, favouring the simplest structure that explains common variance (Garrido et al., 2011). The Empirical Kaiser Criterion (EKC) retains factors with eigenvalues greater than 1 as a criterion of variance explanation (Braeken & Van Assen, 2017). The conclusion was inconclusive as to whether two or three factors are needed to explain the data.

Analysis and results

To evaluate reliability of the questionnaire, Cronbach's alpha and McDonald's omega with 95% confidence intervals were both used. With commonly accepted thresholds of 0.70 indicating internal consistency (Dunn et al., 2014; Tavakol & Dennick, 2011), findings demonstrate acceptable reliability compared to Schmidt et al. (2009). While Cronbach's alpha is a measure of how well the items measure the same construct, McDonald's omega outperforms alpha in the case of violations of tau-equivalence. Table 1 compares question reliability from Schmidt et al. (2009) to ours to validate that all questions measure their respective items.

Table 1. *Cronbach's alpha and McDonald's omega for the questionnaire. "Schmidt" refers to results from the study by Schmidt et al. (2009).*

Variable	Schmidt et al.	Current study	Difference	α Schmidt	α current study	ω
Technological knowledge				0.82	0.89	0.90
TK1	0.75	0.76	0.01			
TK2	0.76	0.92	0.16			
TK3	0.70	0.89	0.19			
Content knowledge				0.85	0.78	0.79
CK1	0.89	0.82	-0.07			
CK2	0.89	0.61	-0.28			
CK3	0.86	0.82	-0.04			
Pedagogical knowledge				0.84	0.78	0.81
PK1	0.79	0.79	0.00			
PK2	0.78	0.77	-0.01			
PK3	0.77	0.67	-0.10			
Pedagogical content knowledge				0.85	0.84	0.85
PCK1	0.87	0.83	-0.04			
PCK2	0.84	0.80	-0.04			
PCK3	0.81	0.77	-0.04			
Technological content knowledge				0.80	0.69	0.73
TCK1	0.87	0.91	0.04			
TCK2	0.80	0.30	-0.50			
TCK3	0.69	0.54	-0.15			
Pedagogical technological knowledge				0.86	0.78	0.84
TPK1	0.91	0.85	-0.06			
TPK2	0.89	0.74	-0.15			
TPK3	0.78	0.77	-0.01			
Technological pedagogical content knowledge				0.92	0.92	0.96
TPACK1	0.85	0.85	0.00			
TPACK2	0.87	0.95	0.08			
TPACK3	0.85	0.87	0.02			
TPACK4	0.82	0.81	-0.01			

Confirmatory factor analysis results

To validate the questionnaire against the TPACK model (Koehler & Mishra, 2009), a Confirmatory Factor Analysis was conducted to identify the amount of explanation each question contributes to the overall item. Table 2 indicates high factor loadings on most questions, with Technological Content Knowledge being the construct with lowest loadings.

Table 2. *Factor loadings from questions to each construct.*

Question	TK	PK	CK	TPK	TCK	PCK	TPACK
1	0.76	0.67	0.82	0.85	0.91	0.83	0.85
2	0.92	0.79	0.61	0.74	0.30	0.80	0.95
3	0.89	0.77	0.82	0.77	0.54	0.77	0.87
4							0.81

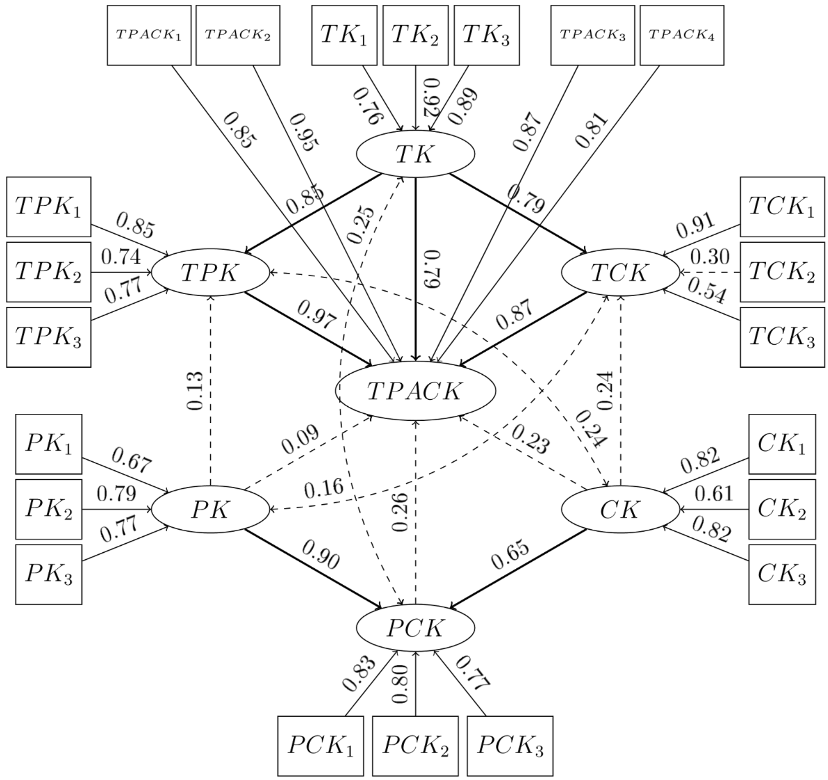
Table 3 presents correlations between latent variables, indicating interconnected variable explanations. An interesting finding is the high contribution towards TPACK for all factors involving technology, and Pedagogical Knowledge contributing overall low on the TPACK-loading.

Table 3. *Factor loadings between latent variables.*

	TPK	TCK	PCK	TPACK
TK	0.85	0.79	0.25	0.79
PK	0.13	0.16	0.90	0.09
CK	0.24	0.24	0.65	0.23
TPK				0.97
TCK				0.87
PCK				0.26

Validation of the questionnaire in table 1 and table 2 indicates high loadings overall, and it is clear from table 3 that all factors with pedagogy have low loadings on TPACK. Technology stands out as the main predictor of TPACK with content contributing mostly when combined with technology. This pattern suggests that Pedagogical Knowledge contributes marginally to the model, while technology is the main contributor. The overall findings from CFA are presented in figure 2, highlighting strong loadings.

Figure 2. *Complete CFA-model.*

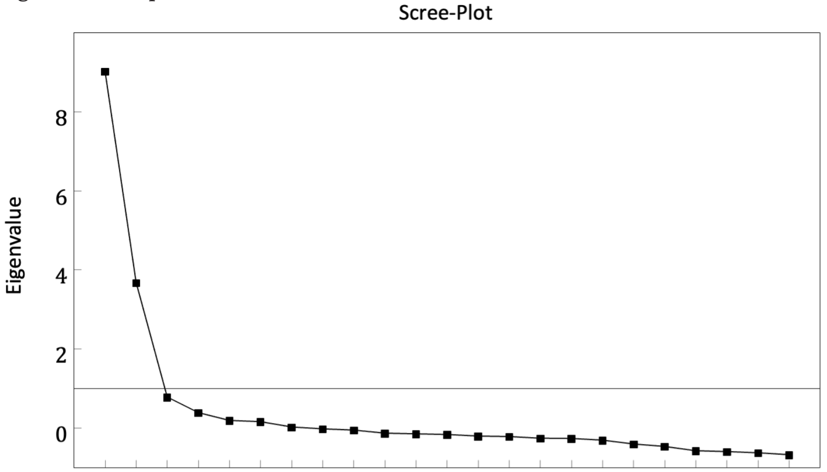


From figure 2, it becomes evident that TK, TPK, TCK and TPACK are connected, and PK, CK and PCK are connected. There is a disconnect between those two domains. It is expected according to Koehler and Mishra (2009) that those domains should be connected. According to our results, variables not involving technology have small contributions to TPACK.

Exploratory factor analysis results

Sampling adequacy was confirmed with Kaiser-Meyer-Olkin's test, with all items scoring above 0.80, where scores above 0.70 indicate suitability for factor analysis. Using the Scree plot for a graphic understanding in figure 3, there are two factors with eigenvalue above one, explaining most of the variation. This can be seen clearly where the graph levels out from the third factor.

Figure 3. *Scree plot*



According to Kaiser's criterion where factors with eigenvalues above 1 are retained, this suggests that two factors are needed to explain 60% of the variance, with the third factor raising the explanation to 66%. The Minimum Average Partial test indicates that adding more than two factors will not substantially improve explained variance, which thus supports retaining two factors. The Empirical Kaiser Criterion results in two factors having eigenvalues above reference value, concluding that only two factors should be retained. Parallel analysis is inconclusive. The two first factors having higher eigenvalues than both the average and the 95th percentile reference values, suggest retaining only two factors. The third factor is 1.065, exceeding both the mean and the 95th percentile value, giving a marginal indication of retaining three factors. EFA indicates that the dataset can be explained with two or three factors, where the third factor is a borderline case. A model with three factors was chosen based on the theoretical foundations of this research.

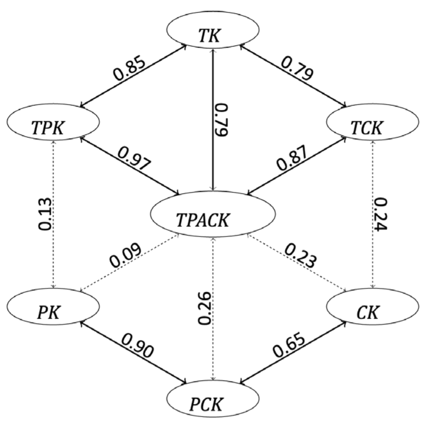
Discussion

As seen in figure 2, TPK, TCK and TK show strong loadings on TPACK, indicating the centrality of technology in explaining the model when programming is represented as the technological dimension. In contrast, PCK, CK and PCK are interconnected but exhibit low loadings on TPACK. A specific finding is that TPK loads highly on both TK and TPACK, yet shows weak association with PK. From previous literature, a link between PK and TPK would normally be expected (Schmidt et al., 2009), as TPK is theorised to capture how pedagogy interacts with technology. In absence of this connection, we expected that inter-factor correlations would occur (Karlsen et al., 2025; Scherer et al., 2017), something our analysis contradicts. To investigate this further, both orthogonal (Varimax) and oblique (Oblimin) rotations were conducted. The Oblimin rotation failed to converge, indicating that factors were not sufficiently correlated to justify oblique rotation. This is a divergence from findings by Karlsen et al. (2024), who reported stronger inter-factor correlations within the TPACK framework. We interpret this as a consequence of the questionnaire's domain-specific focus, shown in Appendix 1. Since our focus is on mathematics education, the non-correlation is consistent with earlier research reporting limited or non-transferable connections between mathematical and programming knowledge (Graafsma et al., 2023; Prat et al., 2020). While our findings show a structural imbalance in how TPACK manifests when programming is considered the technological component in mathematics education, this also indicates that technology is the main factor that binds this model together. These results raise questions about whether TPACK, in its current form, can adequately capture programming as the technology aspect in teacher knowledge. This is further backed up with the ambiguity of whether two or three factors should be retained from EFA, pointing to reduced dimensionality when programming is introduced into mathematics education.

This ambiguity in factor retention, where EFA suggests either two or three factors, points to a reduced dimensionality when technology is represented solely as programming in TPACK. Our explanation relies on the discussion of computational thinking (CT) as defined by Grover and Pea (2013), defining a set of interdisciplinary problem-solving practices that rely on both mathematical and technological processes. CT emphasises decomposition, abstraction and algorithmic design, bridging cognitive practices common in mathematics with practices inherent in programming, hence providing a bridge between content and technology, but does not by itself guarantee pedagogical integration. In our model, the dominance of technological knowledge and the marginal role of pedagogy suggests that teachers perceive programming as a technical tool rather than as a pedagogically integrated practice. This interpretation suggests that there is little linkage to general pedagogy. Similar observations have been shown in previous studies (Fabian et al., 2024; Scherer et al., 2017; Schmid et al., 2024), even though others (Karlsen et al., 2025) found when technology is represented solely as programming, TPACK dimensions underspecify the knowledge used. Similarly, programming draws on language skills in addition to mathematics, where teachers may use CT-aligned technical routines without a corresponding pedagogical connection (Prat et al., 2020). In sum, the pattern that TPK has strong loadings on TK and TPACK but weak loadings on PK is consistent with prior research where also a lack of pedagogical anchoring was found (Karlsen et al., 2025; Scherer et al., 2017).

The pedagogical connection to TPK suggests programming knowledge does not transfer from mathematics, challenging Grover and Pea's (2013) description of CT as a bridge between programming and mathematics. While Grover and Pea (2013) show abstraction, decomposition, and algorithmic reasoning as shared practices, Prat et al. (2020) found that programming success also depends on vocabulary skills, in addition to algebraic and logical skills, indicating a broader cognitive base than mathematics alone (Graafsma et al., 2023). This is reinforced by Prat et al. (2020) in that language skills are as predictive of programming performance as mathematical ability. This resonates with patterns observed in our model where technological and content-related knowledge provides greater explanatory power than pedagogy. Teachers may therefore approach programming primarily as CT-aligned technical routines, which overlap with mathematics at the level of abstraction and algorithmic reasoning, but diverge when programming requires linguistic skills. These findings caution against treating programming as a direct extension of mathematical knowledge within TPACK. Instead, programming appears to be a hybrid domain, situated at the intersection between mathematics, computational thinking and linguistic reasoning. With programming requiring the abstraction and algorithmic thinking from mathematics and CT and, at the same time, practices like syntax and semantics from linguistics, it emerges as a combination that complicates the traditional TPACK categories.

Figure 4. *Visualisation of constructs' explanatory power.*



Our model shows negligible correlations between several factors, figure 4, indicating programming as not a blend of existing constructs, but could be conceptualised as its own distinct domain of teacher knowledge that expands the skills required by today's teachers.

Our findings suggest an imbalance when programming is introduced into TPACK in mathematics education. This asymmetry challenges the assumption that programming can be seamlessly integrated as a generic technological dimension within TPACK. Instead, programming might function as a distinct knowledge domain, drawing on mathematical knowledge, computational thinking and linguistic knowledge. If programming does constitute its own domain, this raises implications for teacher education; rather than relying on computational thinking, more research is needed on what teachers need to know in order to be able to teach programming in school mathematics.

Conclusion

This study examined the applicability of the TPACK model when programming is introduced into mathematics education based on teachers' self-reported knowledge. Our results demonstrate that technology-related knowledge provides the strongest explanatory power when programming constitutes the theoretical domain, while pedagogy provides a marginal explanatory power. This asymmetry indicates that programming does not function as a generic technology domain in TPACK, but instead as its own hybrid domain, drawing on mathematical knowledge, computational thinking and linguistic skills. These findings challenge the assumption that technology can be represented solely as programming. Furthermore, this suggests that programming-specific knowledge, beyond computational thinking, is needed to integrate programming into mathematics teaching. We argue that programming is its own domain, not comprised of technology, mathematics and computational thinking. This might have implications for teacher-education programmes and calls for more research on knowledge for teaching programming in school mathematics.

References

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International journal of child-computer interaction*, *16*, 68–76. https://doi.org/10.1016/j.ijcci.2017.12.004

Bocconi, S., Chioccariello, A., Kampylis, P., Dagien, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasut?, E., & Malagoli, C. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. https://doi.org/10.2760/126955

Braeken, J., & Van Assen, M. A. (2017). An empirical Kaiser criterion. *Psychological methods*, *22*(3), 450–466. https://doi.org/10.1037/met0000074

Cheung, G. W., Cooper-Thomas, H. D., Lau, R. S., & Wang, L. C. (2024). Reporting reliability, convergent and discriminant validity with structural equation modeling: A review and best-practice recommendations. *Asia Pacific Journal of Management*, *41*(2), 745–783. https://doi.org/10.1007/s10490-023-09871-y

Ching Sing Chai, Hwee Ling Koh, J., & Chin-Chung Tsai. (2013). A Review of Technological Pedagogical Content Knowledge. *Journal of Educational Technology & Society*, *16*(2), 31–51.

Costello, A. B., & Osborne, J. (2005). Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical assessment, research, and evaluation*, *10*(1). https://doi.org/10.4135/9781412995627.d8

Dinno, A. (2009). Exploring the Sensitivity of Horn's Parallel Analysis to the Distributional Form of Random Data. *Multivariate Behavioral Research*, *44*(3), 362–388. https://doi.org/10.1080/00273170902938969

Dunn, T. J., Baguley, T., & Brunsden, V. (2014). From alpha to omega: A practical solution to the pervasive problem of internal consistency estimation. *British Journal of Psychology*, *105*(3), 399–412. https://doi.org/10.1111/bjop.12046

Elicer, R., Tamborg, A. L., Bråting, K., & Kilhamn, C. (2023). Comparing the Integration of Programming and Computational Thinking into Danish and Swedish Elementary Mathematics Curriculum Resources. *LUMAT: International Journal on Math, Science and Technology Education*, *11*(3), 77–102. https://doi.org/10.31129/lumat.11.3.1940

Fabian, A., Fütterer, T., Backfisch, I., Lunowa, E., Paravicini, W., Hübner, N., & Lachner, A. (2024). Unraveling TPACK: Investigating the inherent structure of TPACK from a subject-specific angle using test-based instruments. *Computers & Education*, *217*, 105040. https://doi.org/10.1016/j.compedu.2024.105040

Forsström, S. E., & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, *17*(12), 18–32. https://doi.org/10.26803/ijlter.17.12.2

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011). Performance of Velicer's Minimum Average Partial Factor Retention Method With Categorical Variables. *Educational and Psychological Measurement*, *71*(3), 551–570. https://doi.org/10.1177/0013164410389489

Geraniou, E., Jankvist, U. T., Elicer, R., Tamborg, A. L., & Misfeldt, M. (2024). Towards a definition of "mathematical digital competency for teaching". *ZDM – Mathematics Education*, *56*(4), 625–637. https://doi.org/10.1007/s11858-024-01585-9

Giannakos, M. N., Doukakis, S., Pappas, I. O., Adamopoulos, N., & Giannopoulou, P. (2015). Investigating teachers' confidence on technological pedagogical and content knowledge: An initial validation of TPACK scales in K-12 computing education context. *Journal of Computers in Education*, *2*(1), 43–59. https://doi.org/10.1007/s40692-014-0024-8

Gonscherowski, P., & Rott, B. (2025). A systematic review of the literature on TPACK instruments used with pre-service teachers from 2017 to 2023 focused on selecting digital resources. *Journal of Computers in Education*. https://doi.org/10.1007/s40692-025-00379-6

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Graafsma, I. L., Robidoux, S., Nickels, L., Roberts, M., Polito, V., Zhu, J. D., & Marinus, E. (2023). The cognition of programming: Logical reasoning, algebra and vocabulary skills predict programming performance following an introductory computing course. *Journal of Cognitive Psychology*, *35*(3), 364–381. https://doi.org/10.1080/20445911.2023.2166054

Gudmundsdottir, G. B., & Hatlevik, O. E. (2018). Newly qualified teachers' professional digital competence: Implications for teacher education. *European Journal of Teacher Education*, *41*(2), 214–231. https://doi.org/10.1080/02619768.2017.1416085

Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 1–55. https://doi.org/10.1080/10705519909540118

Kaiser, H. F. (1974). An Index of Factorial Simplicity. *Psychometrika*, *39*(1), 31–36. https://doi.org/10.1007/BF02291575

Karlsen, N., Henriksen, E. K., & Pajchel, K. (2025). Assessing Teachers' Knowledge of How to Use Computer Programming in Science and Technology Education. *Journal of Science Education and Technology*, *34*(1), 45–57. https://doi.org/10.1007/s10956-024-10145-5

Kelentrić, M., Helland, K., & Arstorp, A.-T. (2017). Rammeverk for lærerens profesjonsfaglige digitale kompetanse. *Senter for IKT i utdanningen*, 1–21.

Kenny, D. A., Kaniskan, B., & McCoach, D. B. (2015). The Performance of RMSEA in Models With Small Degrees of Freedom. *Sociological Methods & Research*, *44*(3), 486–507. https://doi.org/10.1177/0049124114543236

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)? *Contemporary issues in technology and teacher education*, *9*(1), 60–70.

Laurent, M., Crisci, R., Bressoux, P., Chaachoua, H., Nurra, C., de Vries, E., & Tchounikine, P. (2022). Impact of programming on primary mathematics learning. *Learning and Instruction*, *82*, 101667. https://doi.org/10.1016/j.learninstruc.2022.101667

Ledesma, R. D., Valero-Mora, P., & Macbeth, G. (2015). The scree test and the number of factors: A dynamic graphics approach. *The Spanish journal of psychology*, *18*, E11. https://doi.org/10.1017/sjp.2015.13

Li, M., Vale, C., Tan, H., & Blannin, J. (2025). A systematic review of TPACK research in primary mathematics education. *Mathematics Education Research Journal*, *37*(2), 281–311. https://doi.org/10.1007/s13394-024-00491-3

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology.* https://psycnet.apa.org/record/1933-01885-001/

Ma, H., Dong, Y., Jing, B., Zeng, Y., & Sun, J. (2025). Effectiveness of in-service computer science teachers' professional development in K-12 education: A systematic review and meta-analysis. *International Journal of STEM Education*, *12*(1), 29. https://doi.org/10.1186/s40594-025-00548-0

MacCallum, R. C., Browne, M. W., & Sugawara, H. M. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological methods*, *1*(2), 130–149. https://doi.org/10.1037//1082-989x.1.2.130

Marsh, H. W., Hau, K.-T., & Wen, Z. (2004). In Search of Golden Rules: Comment on Hypothesis-Testing Approaches to Setting Cutoff Values for Fit Indexes and Dangers in Overgeneralizing Hu and Bentler's (1999) Findings. *Structural Equation Modeling: A Multidisciplinary Journal*, *11*(3), 320–341. https://doi.org/10.1207/s15328007sem1103_2

Ministry of Education and Research. (2020a). Overordnet del grunnleggende ferdigheter i matematikk. Læreplanverket for Kunnskapsløftet 2020. https://www.udir.no/lk20/mat01-05/om-faget/grunnleggende-ferdigheter

Ministry of Education and Research. (2020b). *Overordnet del—Grunnleggende ferdigheter.* Læreplanverket for Kunnskapsløftet 2020. https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/grunnleggende-ferdigheter/

Nyman, R., Bråting, K., & Kilhamn, C. (2025). Can programming support mathematics learning? An analysis of Swedish lower secondary textbooks. *International Journal of Mathematical Education in Science and Technology*, *56*(7), 1261–1279. https://doi.org/10.1080/0020739X.2024.2329345

Prat, C. S., Madhyastha, T. M., Mottarella, M. J., & Kuo, C.-H. (2020). Relating natural language aptitude to individual differences in learning programming languages. *Scientific reports*, *10*(1), 3817. https://doi.org/10.1038/s41598-020-60661-8

Scherer, R., Rohatgi, A., & Hatlevik, O. E. (2017). Students' profiles of ICT use: Identification, determinants, and relations to achievement in a computer and information literacy test. *Computers in Human Behavior*, *70*, 486–499. https://doi.org/10.1016/j.chb.2017.01.034

Schmid, M., Brianza, E., Mok, S. Y., & Petko, D. (2024). Running in circles: A systematic review of reviews on technological pedagogical content knowledge (TPACK). *Computers & education*, *214*, 105024. https://doi.org/10.1016/j.compedu.2024.105024

Schmidt, D. A., Baran, E., Thompson, A. D., Mishra, P., Koehler, M. J., & Shin, T. S. (2009). Technological Pedagogical Content Knowledge (TPACK): The Development and Validation of an Assessment Instrument for Preservice Teachers. *Journal of Research on Technology in Education*, *42*(2), 123–149. https://doi.org/10.1080/15391523.2009.10782544

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International journal of medical education*, *2*, 53–55. https://doi.org/10.5116/ijme.4dfb.8dfd

Wolf, E. J., Harrington, K. M., Clark, S. L., & Miller, M. W. (2013). Sample Size Requirements for Structural Equation Models: An Evaluation of Power, Bias, and Solution Propriety. *Educational and Psychological Measurement*, *73*(6), 913–934. https://doi.org/10.1177/0013164413495237

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, *10*(1), 3. https://doi.org/10.1186/s40594-023-00396-w

Appendix 1, Questionnaire

Jeg lærer programmering lett

Jeg er personlig selvgående i programmering
Jeg har nødvendige programmeringsferdigheter til å undervise kompetansemålene i matematikk
Jeg har nødvendig kunnskap i matematikk for læreryrket
Jeg kan bruke matematisk og algoritmisk tenkning
Jeg har flere ulike strategier for å øke min matematikkforståelse
Jeg kan vurdere kompetansen til elever mens jeg er i klasserommet
Jeg kan tilpasse undervisningen til elevenes forkunnskaper
Jeg kan tilpasse egen undervisning til ulike læringsformer
Jeg kan tilpasse undervisningen til å fremme matematisk tenkning og læring på flere måter
Jeg vet om flere ulike didaktiske tilnærmingsmåter til matematikk
Jeg kan bruke teori i min matematikkundervisning på en god måte
Jeg kan bruke programmering for å øke egen matematisk innsikt
Jeg kjenner til flere teknologiske verktøy som fremmer egen matematikklæring
Jeg ser nødvendigheten av programmering i matematikkfaget
Jeg kan bruke programmering til å fremme elevers matematikkunnskap
Jeg kan øke elevers matematiske læringsutbytte gjennom programmering
Jeg vet om flere didaktiske tilnærmingsmåter tilknyttet programmering
Jeg kan lage ulike undervisningsopplegg som inkluderer både matematikk og programmering
Jeg kan kombinere matematikk og programmering for å gi eleven økt læringsutbytte
Jeg kan lage didaktiske programmeringsoppgaver i matematikk tilpasset elevenes nivå
Jeg har en klar didaktisk tilnærming til programmering i matematikk

Helge Jeppesen

Helge Jeppesen is an assistant professor in mathematics education at Western Norway University of Applied Sciences. His background is in both mathematics and programming, with current research interests focusing on algebra and the integration of programming in mathematics education.

Helge.Jeppesen@hvl.no

Nils Henry Rasmussen

Nils Henry Rasmussen is an associate professor in mathematics education at Western Norway University of Applied Sciences. His background is in both mathematics and programming. His current research interests are in culturally relevant mathematics education.

Nils.Henry.Williams.Rasmussen@hvl.no