

e-FRAN > PLATEFORME

e-FRAN > DES TERRITOIRES ÉDUCATIFS
D'INNOVATION NUMÉRIQUE

Mission Monteil > POUR LE NUMÉRIQUE
DANS L'ÉDUCATION

ProFAN > DES COMPÉTENCES
POUR LES EMPLOIS DU FUTUR



Évaluation d'une expérimentation randomisée de la pensée informatique, vecteur d'apprentissage des mathématiques au cycle 3 de l'école élémentaire, en classe de CM1 et CM2

Manon CHEVALIER-LAURENT

Mots-clés – Niveaux et Public concernés

Mots-clés : pensée informatique, mathématiques, programmation, école primaire, efficacité enseignement, anxiété mathématique, motivation autodéterminée, sentiment de compétence

Niveaux : CM1 et CM2

Public : enseignant-e-s de cycle 3

À quelles questions cette étude tente-t-elle de répondre ?

Cette recherche porte sur l'effet de la pensée informatique via la programmation en tant que facteur d'efficacité sur les acquisitions et les variables motivationnelles, en mathématiques, au cycle 3 de l'école primaire. Dans ce cadre, nous posons deux questions : (1) L'utilisation de la programmation visuelle pour résoudre des problèmes mathématiques permet-elle de favoriser les acquisitions en mathématiques ? (2) L'environnement de programmation, parce qu'il est différent de l'environnement habituel et permet de délivrer aux élèves des rétroactions directes et neutres affectivement, peut-il lever certains freins sociocognitifs vis-à-vis des mathématiques tels que l'anxiété et favoriser des comportements d'approche vis-à-vis de cette activité (motivation, sentiment d'efficacité personnelle) ?

Pourquoi ces questions sont-elles pertinentes ?

Question 1 : Programmer pour apprendre les mathématiques a d'abord été mis en œuvre dans les années 70 sous l'impulsion de Papert (1980) du MIT avec le langage Logo (et d'autres tels que Pascal ou Basic), spécialement développé pour les jeunes enfants. Dans une perspective constructionniste, la programmation permettait une forme de manipulation favorisant l'accès aux concepts abstraits. Néanmoins, la difficile maîtrise de ces langages a freiné le développement de cette pratique qui s'est progressivement éteinte. Dans les années 2000, le développement de langages visuels, tels que *Scratch*,

simplifie l'accès à la compréhension des concepts de programmation (Lye et Koh, 2014) et rend à nouveau possible les projets d'apprentissages des mathématiques avec la programmation. Concomitamment, les compétences liées à la pensée informatique sont considérées comme indispensables à la formation du citoyen du 21^e siècle et se voient intégrées dans les programmes scolaires dès le primaire, souvent à travers la discipline des mathématiques (Bocconi *et al.*, 2016; Tang *et al.*, 2019; Wing et Stanzione, 2016).

Les apprentissages mathématiques sont cruciaux pour la réussite académique des élèves, mais les enquêtes nationales et internationales indiquent que les résultats des élèves français dans cette discipline sont en baisse depuis plus de 30 ans (Chabanon et Pastor, 2019). La mise en œuvre de méthodes efficaces est donc nécessaire pour améliorer ces apprentissages et la pratique de la programmation pourrait être une piste pertinente pour cela. Outre qu'elle permette une manipulation concrète des objets mathématiques et offre des rétroactions directes non affectives, elle permet de traiter les notions mathématiques à travers le registre de la programmation et celui de l'écriture mathématique traditionnelle. Le passage d'un registre à l'autre s'établit par une conversion de registres (Duval, 1999; Duval, 2000), forme de transfert d'apprentissage entre les deux types de représentations mathématiques. Pour Duval (1999) et Duval (2000), la compréhension profonde des notions mathématiques dépend de la maîtrise de la conversion de registres.

Pour autant, il convient d'évaluer les effets de cette pratique d'ores et déjà à l'œuvre dans le primaire (B.O. spécial n°11 du 26/11/2015) car elle présente également des points qui pourraient freiner les apprentissages. Ainsi, la conversion de registres demeure difficile à réaliser et doit être soigneusement guidée et soutenue pour aboutir (Duval, 1999; Duval, 2000). De plus, l'utilisation des langages visuels peut engendrer une charge cognitive trop lourde gênant le traitement des données nécessaires aux apprentissages. Les études qui portent sur les effets de cette pratique sont rares et relèvent peu souvent d'une méthodologie fiable (Hickmott *et al.*, 2018; Moreno-León et Robles, 2016). Notre étude contrôlée randomisée permet d'apporter des résultats répondant aux exigences de qualité fixées par la recherche en éducation, à savoir, des résultats issus de données probantes sur les effets des activités de programmation en mathématiques sur les acquisitions de cette discipline.

Question 2 : Les usages numériques sont fréquemment mis en avant notamment pour leurs effets positifs sur la motivation des élèves, mais peu d'études évaluent leur relation (Galand, 2020; Tricot, 2020). Pour l'étudier, nous nous sommes intéressés à trois construits motivationnels, prédicteurs de réussite scolaire : la motivation autodéterminée, le sentiment de compétence et l'anxiété.

Ces trois construits fonctionnent en constellation : dans un schéma adaptatif, plus la motivation est autodéterminée, plus le sentiment de compétence est fort et l'anxiété faible. La formation de comportements adaptatifs est favorisée par des pratiques de classes qui soutiennent trois besoins psychologiques fondamentaux : l'autonomie, la compétence et la proximité sociale (Sarrazin *et al.*, 2011). Parmi ces pratiques, la structuration des apprentissages et la qualité des interactions sont centrales. L'accueil des erreurs comme indices d'apprentissage, et non comme indices d'une intelligence moindre favorise la motivation, le sentiment de compétence et la baisse de l'anxiété (Dweck, 2015). À l'inverse, le manque de soutien ou encore la gêne occasionnée par l'échec face aux pairs sont les ingrédients de l'anxiété en mathématiques (Ashcraft et Krause, 2007).

À ce titre, certaines fonctionnalités des outils de programmation peuvent favoriser la motivation parce qu'elles offrent des espaces d'autonomie qui renforcent le sentiment de contrôle des élèves et sont sources de motivation (Viau, 2009). Par ailleurs, elles permettent aux élèves d'obtenir une rétroaction directe, sans jugement car délivrée par la machine, évitant aux élèves la peur de se tromper en public. Ceci pourrait réduire l'anxiété des élèves anxieux (Ashcraft et Krause, 2007). Les élèves peuvent ainsi rester engagés dans la tâche en réagissant aux rétroactions directes par l'ajustement, si nécessaire, de leur raisonnement et le test de nouvelles solutions algorithmiques.

Hypothèses : Notre travail teste quatre hypothèses, la première sur les acquisitions mathématiques et les trois suivantes sur la conation (concernant les variables motivationnelles) en mathématiques. Nous supposons que l'utilisation du dispositif de programmation visuelle améliorera les performances

des élèves sur trois notions étudiées : la division euclidienne, la décomposition additive et les fractions, comparativement aux élèves du groupe contrôle (H1). Nous faisons l'hypothèse que l'utilisation de la programmation visuelle, de par son caractère innovant et ludique, agira favorablement sur les différents processus motivationnels que nous avons mesurés, en soutenant la formation de régulations plus autodéterminées de la motivation en mathématiques¹ (H2.1), en affectant positivement le sentiment de compétence en mathématiques (H2.2), et en réduisant l'anxiété en mathématique des élèves comparativement aux élèves du groupe contrôle (H2.3).

Quelle méthodologie de recherche a-t-on utilisée ?

Notre étude a porté sur 2 591 élèves de CM1 et CM2 issus de 109 classes de 47 écoles de la circonscription de Grenoble recrutées en fin d'année scolaire 2016-2017 (tous les enseignants qui se sont portés volontaires suite à un appel à enseignants auprès de la circonscription de Grenoble ont été acceptés). Les écoles ont été choisies comme unités de randomisation pour l'assignement aux deux conditions : programmation ($n_{écoles} = 28$, $n_{classes} = 68$) et contrôle ($n_{écoles} = 18$, $n_{classes} = 41$). Parmi les 109 classes recrutées, 36 se trouvaient en zones d'éducation prioritaire.

L'intervention dans les classes s'est déroulée sur une durée d'environ cinq mois. Les enseignants ont été formés (T0) et ont reçu la documentation nécessaire à la mise en œuvre des enseignements spécifiques à leur groupe. Chaque notion (division euclidienne, décomposition additive, fractions) comptait trois séances de 50 min maximum. Les élèves ont travaillé en binômes dans les deux conditions de travail.

Avant l'intervention (T1), les questionnaires élèves ont permis de recueillir les données conatives : anxiété en mathématiques, mesurée avec l'échelle adaptée de Pouille (2016), le sentiment de compétence en mathématiques, mesuré avec l'échelle de Harter (1985) sur la dimension des « compétences scolaires » (*Scholastic Competence*) et la motivation autodéterminée en mathématiques, mesurée avec l'échelle de Guay *et al.*, (2010). À cette même période, les élèves ont réalisé un test de performance générale en mathématiques, et les enseignants ont rempli des questionnaires en ligne pour recueillir des informations sur leurs pratiques et avis sur la formation Expire (projet dans lequel s'inscrit ce travail de recherche).

À partir d'octobre 2017, les séquences de mathématiques ont débuté selon la progression prescrite. Chacune des notions a donné lieu à un prétest (T2, T4, T6) et un post-test (T3, T5, T7), similaires pour chaque notion, mesurant les acquisitions des élèves sur chacune des notions traitées, administrés par les enseignants en classe entière.

À la fin de la mise en œuvre du dispositif dans les classes, les élèves ont à nouveau été soumis à des questionnaires afin de recueillir les données conatives post-intervention, identiques à celles recueillies en T1. Les enseignants ont à nouveau répondu à des questionnaires en ligne, pour recueillir notamment leur avis sur le déroulement du dispositif dans leur classe (T8).

Deux principes ont guidé la conception des activités de nos groupes :

1. Recourir aux activités les plus courantes pour le développement de la notion visée pour les activités du groupe contrôle.
2. Transposer les activités du groupe contrôle en activités de programmation, au plus près de celles du groupe contrôle : les situations d'apprentissage « traditionnelles » du groupe contrôle, réalisées avec le matériel habituel de la classe, devaient être transposables en situation de résolution de problèmes par la programmation. Cela impliquait qu'il soit possible de proposer, par le biais de *Scratch* en l'occurrence, un matériel visuel — affiché dans l'espace d'exécution du programme — suffisamment simple pour faciliter la compréhension de la tâche et permettre aux élèves de valider leur programme lors de son exécution (l'exécution du programme produit une action comme

1. L'hypothèse H2.1 se décline en trois sous-hypothèses, la motivation auto-déterminée étant ici étudiée au regard de trois dimensions. Nous nous attendons à observer, une baisse de la régulation contrôlée (H2.1.1) et une augmentation des régulations identifiée (H2.1.2) et intrinsèque (H2.1.3) chez les élèves du groupe expérimental, comparativement au groupe contrôle.

le déplacement d'un curseur par exemple, visible sur ce visuel). L'idée clé de la conception de cette expérimentation réside dans la possibilité de construire un programme pour résoudre un problème qui soit une traduction de l'expression symbolique mathématique à apprendre pour la notion visée.

Quels résultats a-t-on obtenus ?

Nous avons opté pour une approche analytique multiniveau permettant de saisir l'effet que produit chaque « niveau » (école, classe, élève) sur nos variables dépendantes. Nous avons d'abord construit des modèles à 3 niveaux incluant ces niveaux. Cependant la variance entre les écoles n'était pas significative et les valeurs des coefficients de corrélation intraclasse étaient trop faibles pour justifier la prise en compte du niveau « école » dans les analyses. Nos modèles ont donc porté sur deux niveaux : classes et élèves. Les écoles ont été choisies comme unités de randomisation (pour éviter que plusieurs classes d'une même école ne fassent partie de groupes expérimentaux différents).

Question 1 : Nos résultats montrent que les élèves du groupe de programmation font moins de progrès que ceux du groupe de contrôle. Les effets moyens sur les acquisitions des habiletés spécifiques finales du groupe de programmation sont de -0,16 ET (pour la première séquence portant sur la division euclidienne), -0,19 ET (pour la deuxième séquence portant sur les décompositions additives) et -0,21 ET (pour la dernière séquence portant sur les fractions). Ces effets sont nettement plus négatifs que ceux de l'étude *ScratchMath* (Boylan *et al.*, 2018), qui n'a montré aucun effet de la programmation sur l'apprentissage général des mathématiques (0,03 ET). Ces éléments montrent que, conformément à la littérature, le transfert d'apprentissages escompté dans le cadre de l'apprentissage des mathématiques en passant par des activités de programmation n'est pas automatique.

L'utilisation de la pensée informatique est pourtant particulièrement pertinente dans le domaine des mathématiques car la conversion d'un registre (programmation) à un autre (de l'écriture symbolique mathématique) est censée favoriser la compréhension profonde des notions mathématiques (Duval, 2006). Cependant, la conversion entre les deux registres ne s'est pas produite de la manière attendue, malgré le fait que les enseignants aient reçu la consigne de se montrer le plus explicite possible sur les correspondances et similarités entre ces deux registres lors de la phase d'explication. Il est possible que cette phase n'ait pas été réalisée de façon aussi soutenue qu'attendu. Les enseignants n'ont peut-être pas saisi l'importance de cette phase pour la réussite des élèves du groupe de programmation et cela aura affecté leur apprentissage en entravant le transfert d'apprentissage. Ainsi, nous concluons que le transfert n'a pas été favorisé comme attendu.

Il est possible que les élèves du groupe de programmation aient dû faire face à un trop-plein d'informations liées à la compréhension de l'environnement de programmation, au détriment de l'attention à porter sur la notion à apprendre. Bien qu'ayant bénéficié du même temps d'enseignement des mathématiques que le groupe contrôle, les élèves du groupe de programmation ont pu avoir moins de temps pour s'engager activement dans des activités mathématiques, consacrant une partie de leur temps à la compréhension de leur nouvel environnement de travail. Or, le temps consacré à la tâche est connu pour être un fort prédicteur de réussite (Brophy, 1986). Ce prédicteur de réussite est d'autant plus puissant qu'il est associé à la réussite dans la tâche.

Il est possible que les élèves du groupe programmation aient eu plus de difficulté à réussir les tâches qui leur étaient proposées en programmation, ce qui aura affecté négativement leur temps d'apprentissage académique (Fisher *et al.*, 2015). De même, il est possible que les enseignants du groupe de programmation aient réduit leur temps d'instruction directement ciblé sur les mathématiques pour en consacrer une partie à la gestion du matériel informatique ou à l'utilisation de *Scratch*.

Il existe un consensus sur le fait que trop d'informations dépasse les limites de la mémoire de travail (Baddeley, 1992; Chandler et Sweller, 1991; Sweller *et al.*, 2019). La théorie de la charge cognitive est basée sur une conception de l'architecture cognitive selon laquelle l'apprentissage nécessite une réorganisation des connaissances impliquant la mémoire de travail. Cette mémoire est limitée et dépend

de la charge cognitive impliquée dans la situation d'apprentissage. Si cette charge dépasse la capacité de la mémoire de travail, l'apprentissage est compromis.

Question 2 : Nos résultats ne montrent aucun effet de la condition programmation sur nos variables conatives relatives aux mathématiques. Ainsi, comparativement à la condition contrôle, aucun effet n'apparaît sur les régulations de la motivation auto-déterminée (H2.1), sur le sentiment de compétence (H2.2) ou sur l'anxiété en mathématique (H2.3), nous contraignant à rejeter ces trois hypothèses conatives. L'homogénéité de ces résultats, qui portent sur des variables qui fonctionnent en constellations (Moore *et al.*, 2015; Viau, 1995), donne une bonne assurance de leur validité générale.

De nombreuses études évoquent l'effet de la programmation sur la motivation. Les arguments en faveur d'une motivation accrue par l'usage des langages de programmation s'appuient sur le constat du plaisir qu'ont les enfants à utiliser ces logiciels, ce qui favoriserait les apprentissages. Cependant, le plaisir à utiliser une technologie ne correspond pas systématiquement au plaisir de réaliser la tâche. C'est le plaisir à réaliser la tâche qui entretient la motivation intrinsèque et favorise les apprentissages. Le plaisir doit donc être centré sur l'apprentissage et non lié exclusivement à l'utilisation de l'outil informatique (Tricot, 2020; Viau, 2009). Des éléments de notre recherche viennent étayer cet argument. Ainsi, les réponses aux questionnaires passés auprès des enseignants en fin d'expérimentation (T8) pour recueillir leurs perceptions quant à la motivation de leurs élèves indiquent que les enseignants ont perçu une motivation plus forte chez leurs élèves dans la condition programmation. Les enseignants donnaient leur degré d'accord sur une échelle de Likert en 5 points à la question : « les élèves ont été motivés ». Cette question « naïve » a pour intérêt de souligner que les perceptions des enseignants à propos de leurs élèves doivent être considérées avec prudence. En effet, s'ils ont perçu un accroissement de la motivation chez leurs élèves, cela ne se retrouve pas dans les analyses des modèles portant sur la motivation autodéterminée que nous avons présentées ici. Nos conclusions confirment les propos de Tricot (2020), Scherer (2016) et Livingstone (2012) qui indiquent que le plaisir des enfants n'est pas une preuve de motivation à réaliser la tâche. C'est le paradoxe préférence/performance (Tricot, 2020).

Que dois-je retenir de cette étude pour ma pratique ?

- Notre étude a montré que l'utilisation de la programmation comme vecteur d'apprentissage des mathématiques en CM1 et CM2 n'est pas aussi efficace que les leçons ordinaires où les mathématiques sont enseignées pour elles-mêmes. Le transfert de l'apprentissage est crucial, mais difficile à réaliser. Ainsi, les langages de programmation visuelle doivent être introduits avec prudence. Les résultats présentés ici indiquent que cela pourrait être délétère pour l'acquisition des mathématiques.
- L'utilisation de la programmation pourrait être exploitée pour des activités mathématiques complémentaires, qui ne réduiraient pas le temps consacré aux leçons régulières, mais pourraient être placées à d'autres moments sous différentes formes : travail complémentaire pour gérer des groupes de travail différenciés ou approfondissement d'une notion, par exemple.
- Si la pensée informatique doit être considérée comme une nouvelle forme d'alphabétisation, comme l'annonce la Commission Européenne (Bocconi *et al.*, 2016), les éléments précédents accèdent l'idée qu'il faut l'intégrer comme une discipline propre, avec des temps dédiés dans les programmes, au moins pour les classes de CM1 et CM2 et non comme un outil qui pourrait s'intégrer aux disciplines, sans formation spécifique des enseignants. Dans l'optique d'un apprentissage de la pensée informatique, un logiciel tel que *Scratch* pourrait se révéler utile. En effet, bien que nous ne l'ayons pas évalué dans notre travail, l'étude *ScratchMath* (Boylan *et al.*, 2018) indique un effet significatif sur les résultats des tests de pensée informatique (0,10 ET) pour le groupe d'intervention.
- Nos résultats ont montré une absence d'effet des activités de programmation sur nos variables conatives : la motivation autodéterminée, le sentiment de compétence et l'anxiété, en mathématiques. Cette absence d'effet de la programmation sur nos variables conatives est intéressante car

elle remet en question le caractère motivant que produirait systématiquement l'introduction des technologies en classe.

Références

- Ashcraft, M. H., et Krause, J. A. (2007). Working memory, math performance, and math anxiety. *Psychonomic Bulletin & Review*, 14, 243-248. <https://doi.org/10.3758/BF03194059>
- Baddeley, A. (1992). Working memory. *Science*, 255(5044), 556-559. <https://doi.org/10.1126/science.1736359>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., et al., (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (rapp. tech.). European Commission, JRC Science for Policy Report (Seville site). <https://doi.org/10.2791/715431>
- Boylan, M., Demack, S., Wolstenholme, C., Reidy, J., et Reaney, S. (2018). *ScratchMaths : evaluation report and executive summary*. Project Report. Education Endowment.
- Brophy, J. (1986). Teacher influences on student achievement. *American Psychologist*, 41(10), 1069. <https://doi.org/10.1037/0003-066X.41.10.1069>
- Chabanon, L., et Pastor, J. (2019). L'évolution des performances en calcul des élèves de CM2 à trente ans d'intervalle (1987-2017). Note d'information 19.08 de la DEPP (p. 1-4).
- Chandler, P., et Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8(4), 293-332. https://doi.org/10.1207/s1532690xci0804_2
- Duval, R. (1999). *Representation, Vision and Visualization : Cognitive Functions in Mathematical Thinking. Basic Issues for Learning* (ED466379) [rapport de recherche]. ERIC. <https://eric.ed.gov/?id=ED466379>
- Duval, R. (2000). *Basic Issues for Research in Mathematics Education* (ED466737) [rapport de recherche]. ERIC. <https://eric.ed.gov/?id=ED466737>
- Duval, R. (2006). A cognitive analysis of problems of comprehension in a learning of mathematics. *Educational Studies in Mathematics*, 61(1-2), 103-131. <https://doi.org/10.1007/s10649-006-0400-z>
- Dweck, C. (2015). Carol Dweck revisits the growth mindset. *Education Week*, 35(5), 20-24.
- Fisher, C., Berliner, D., Filby, N., Marliave, R., Cahen, L., et Dishaw, M. (2015). Teaching behaviors, academic learning time, and student achievement : An overview. *The Journal of Classroom Interaction*, 50(1), 6-24.
- Galand, B. (2020). Le numérique va-t-il révolutionner l'éducation. *Les cahiers de recherche du GIRSEF*, 120.
- Guay, F., Chanal, J., Ratelle, C. F., Marsh, H. W., Larose, S., et Boivin, M. (2010). Intrinsic, identified, and controlled types of motivation for school subjects in young elementary school children. *British Journal of Educational Psychology*, 80(4), 711-735. <https://doi.org/10.1348/000709910X499084>
- Harter, S. (1985). Competence as a dimension of self-evaluation : Toward a comprehensive model of self-worth. Dans J. Nicholls (dir.), *The Development of the Self* (p. 55-121). JAI Press, New York.
- Hickmott, D., Prieto-Rodriguez, E., et Holmes, K. (2018). A scoping review of studies on computational thinking in K-12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4, 48-69. <https://doi.org/10.1007/s40751-017-0038-8>
- Livingstone, S. (2012). Critical reflections on the benefits of ICT in education. *Oxford Review of Education*, 38(1), 9-24. <https://doi.org/10.1080/03054985.2011.577938>
- Lye, S. Y., et Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming : What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Moore, A., Rudy, N., et Schrat, M. (2015). Affect, motivation, working memory, and mathematics. Dans R. C. Kadosh et A. Dowker (dir.), *The Oxford handbook of mathematical cognition* (p. 918-936). Oxford : Medicine UK.

- Moreno-León, J., et Robles, G. (2016). Code to learn with Scratch? A systematic literature review. *IEEE Global Engineering Education Conference (EDUCON)*, 150-156. <https://doi.org/10.1109/EDUCON.2016.7474546>
- Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas* Basic Books. Inc. New York, NY.
- Pouille, J. (2016). *Perceptions de soi, anxiété et réussite scolaire : l'apprentissage du langage écrit* [thèse de doctorat, Université Grenoble Alpes, Grenoble, France].
- Sarrazin, P., Pelletier, L., Deci, E., et Ryan, R. (2011). Nourrir une motivation autonome et des conséquences positives dans différents milieux de vie : les apports de la théorie de l'autodétermination. Dans C. Martin-Krumm et C. Tarquinio (dir.), *Traité de psychologie positive* (p. 273-312). Bruxelles : De Boeck.
- Scherer, R. (2016). Learning from the past—the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, 7, 1390. <https://doi.org/10.3389/fpsyg.2016.01390>
- Sweller, J., van Merriënboer, J. J., et Paas, F. (2019). Cognitive architecture and instructional design : 20 years later. *Educational Psychology Review*, 31, 261-292. <https://doi.org/10.1007/s10648-019-09465-5>
- Tang, K.-Y., Chou, T.-L., et Tsai, C.-C. (2019). A content analysis of computational thinking research : An international publication trends and research typology. *The Asia-Pacific Education Researcher*, 29(1), 9-19. <https://doi.org/10.1007/s40299-019-00442-8>
- Tricot, A. (2020). *Numérique et apprentissages scolaires : quelles fonctions pédagogiques bénéficient des apports du numérique?* [rapport de recherche]. Centre national d'étude des systèmes scolaires (Cnesco); Conservatoire national des arts et métiers (Cnam).
- Viau, R. (1995). L'état des recherches sur l'anxiété en contexte scolaire. *Cahiers de la recherche en éducation*, 2(2), 375-398. <https://doi.org/10.7202/1018209ar>
- Viau, R. (2009). *La motivation en contexte scolaire* (2^e éd.). Bruxelles : De Boeck.
- Wing, J. M., et Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of the ACM*, 59(7), 10-11. <https://doi.org/10.1145/2933410>