

# Computationeel denken in de wiskundeles

## Hoe krijg je dat voor elkaar?

Computationeel denken staat in een groeiende belangstelling. Ook in de wiskundeles zijn er volop mogelijkheden om hier aandacht aan te besteden. Maar hoe doe je dat?

### Inleiding

In de samenleving speelt ICT een steeds grotere rol. Om leerlingen voor te bereiden op deze samenleving en de toekomstige arbeidsmarkt is er in onderwijsbeleidsplannen steeds meer aandacht voor digitale geletterdheid.<sup>[1]</sup>

Onderdeel hiervan is het zogeheten 'computationeel denken', een manier van denken om problemen aan te pakken met digitale middelen, gebruikmakend van de principes van programmeren, zoals algoritmisch denken, patroon herkennen en abstraheren. Computationeel denken geeft handvatten om de digitale wereld op een diepere en rijkere manier te begrijpen. Technologie is dan niet alleen om te consumeren, maar vooral om creatief mee te zijn, aldus Simon Peyton-Jones in zijn TED lezing 'Teaching creative computer science' uit 2014.<sup>[2]</sup> Computationeel denken richt zich op deze creatieve manier van probleem-oplossen, waarbij het stappenplan en de oplossing niet van tevoren duidelijk zijn of vastliggen. Het geeft leerlingen extra gereedschap om voorkomende problemen in allerlei vakken aan te pakken.

Het vak wiskunde leent zich er bij uitstek voor om computationeel denken te ontwikkelen. Er zijn duidelijk raakvlakken, denk aan aspecten zoals generaliseren, abstraheren, probleemoplossen, patroon herkennen en algoritmisch denken. Deze aspecten sluiten nauw aan bij wiskundig denken en dus zouden computationeel denken en wiskundig denken heel goed gecombineerd kunnen worden. Maar hoe besteed je aandacht aan computationeel denken in de wiskundeles en hoe integreer je het in het bestaande curriculum? Een pasklaar antwoord op deze vraag is moeilijk te geven. In dit artikel wil ik ideeën aanreiken door verschillende voorbeelden te geven. Hopelijk inspireert dat om mogelijkheden te verkennen om met computationeel denken aan de slag te gaan!

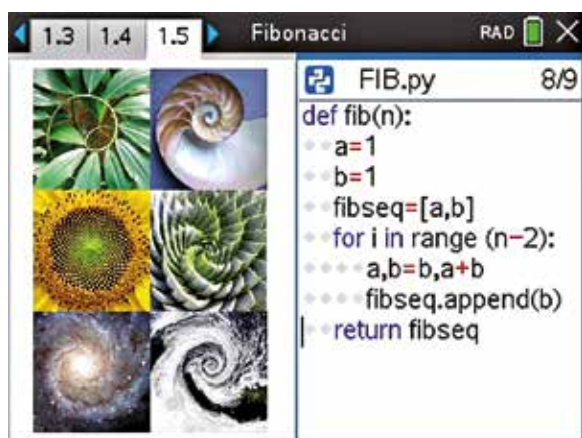
'Moeten we nu gaan programmeren?' is een vraag die

wellicht bij je opkomt als je aan computationeel denken in het onderwijs denkt. Programmeren kan, maar hoeft niet. Voor computationeel denken heb je zelfs niet per se een computer nodig. Het kan ook 'unplugged', zonder computer. Of het kan aan de orde komen bij het gebruik van computer-programma's die je al in de les hanteert, zoals Excel of GeoGebra. Van elk van deze mogelijkheden geef ik hieronder enkele voorbeelden. Eerst bespreek ik toegankelijke manieren om leerlingen te leren programmeren, daarna mogelijkheden om computationeel denken unplugged te beoefenen, en tot slot voorbeelden van computationeel denken waarbij gangbare tools worden gebruikt.

### Computationeel denken door programmeren

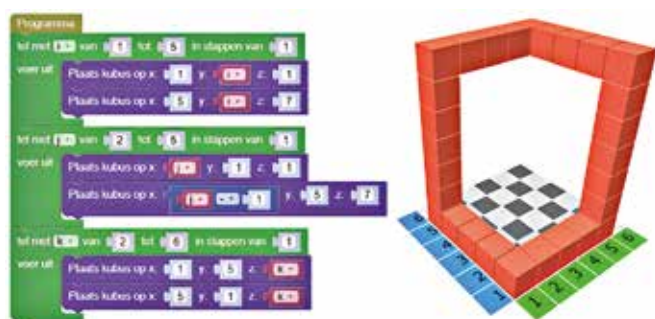
Een vanzelfsprekende manier om computationeel denken toe te passen is door middel van programmeren. Een toegankelijke programmeertaal die veel gebruikt wordt om leerlingen te leren programmeren is Python. Tegenwoordig kun je in Python programmeren op de gangbare grafische rekenmachines zoals die van Texas Instruments, HP, Casio of NumWorks. Maar je kunt natuurlijk ook een computer of laptop gebruiken. Er kunnen ook koppelingen gemaakt worden met andere computers of apparaten. Zo kunnen bepaalde types TI rekenmachines een micro:bit<sup>®</sup> aansturen, een kleine computer waarmee leerlingen de mogelijkheden van elektronica kunnen ontdekken met lampjes en diverse sensoren. Je kunt hierbij denken aan het meten van temperatuur, magneetvelden (kompasfunctie) en acceleratie (trillingen).

De website van TI biedt kant-en-klaar lesmateriaal voor het leren van Python voor de STEM-vakken, waaronder bijvoorbeeld data-analyse, iteraties en functies, fractalen en reële functies.<sup>[3]</sup> In figuur 1 zie je een voorbeeld uit dit materiaal over het programmeren van iteraties en functies, namelijk de reeks van Fibonacci.



figuur 1 Python programmeren op de grafische rekenmachine, voorbeeld over de reeks van Fibonacci uit TI Python BootCamp Deel 2 Iteraties & Functies [3]

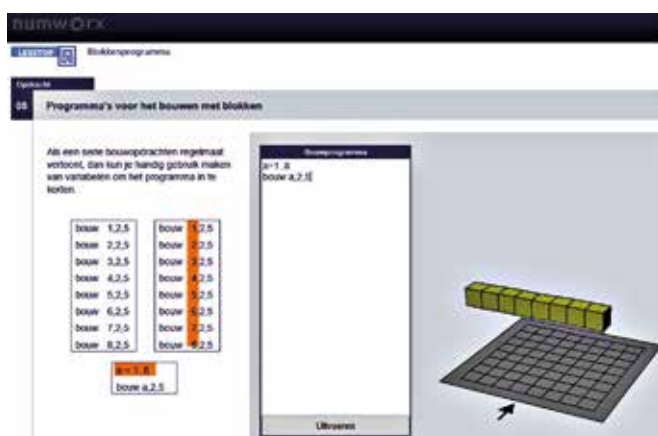
Lesmateriaal om leerlingen stap voor stap te begeleiden bij het leren programmeren in Python kun je vinden op de website van Felienne Hermans van Universiteit Leiden.<sup>[4]</sup> Het materiaal is erop gericht om toegankelijk te zijn voor gebruik in de klas door leerlingen en docenten zonder voorkennis van Python en bevat daarom veel uitleg en oefening. Het doel is vooral ook leerlingen aan te spreken die traditioneel niet voor programmeren zouden kiezen. De modules richten zich daarom juist op andere vakken dan de bètavakken, zoals Nederlands, kunst en geschiedenis. Programmeren kan ook meer visueel worden gedaan met eenvoudige 'programmeerblokken'. Het voordeel van deze aanpak is dat leerlingen minder tijd kwijt zijn met syntax-problemen en typefouten. Voorbeelden van visuele programmeertalen zijn Blockly<sup>[5]</sup>, Scratch<sup>[6]</sup> of Snap!<sup>[7]</sup> Een voorbeeld van Blockly programmeercode zie je in figuur 2.



figuur 2 Visueel programmeren in Blockly met resultaat in augmented reality op smartphone, ontwikkeld in project <colette>

Het voorbeeld komt uit het Europees project '<colette>' (Computational Thinking Learning Environment for Teachers in Europe)<sup>[8]</sup>, waarin een database wordt ontwikkeld van taken voor computationeel denken die

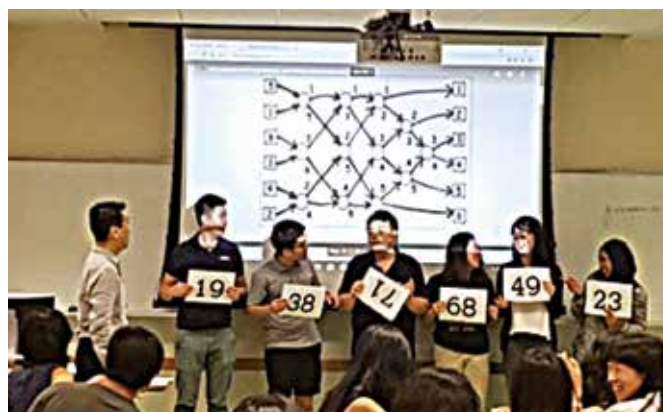
docenten eenvoudig kunnen aanpassen en gebruiken. In het voorbeeld moeten leerlingen een bouwwerk maken door kubussen te plaatsen met programmeercode. Leerlingen gebruiken hiervoor een smartphone en kunnen het geprogrammeerde bouwwerk in augmented reality van alle kanten bekijken. Een andere, toegankelijke manier van programmeren met een beperkte en dus overzichtelijke syntax biedt de Numworx Digitale Wiskunde Omgeving.<sup>[9]</sup> In een serie opgaven kan programmeercode gebruikt worden om gebouwen van blokken te bouwen, zie figuur 3. Hierin komen de basisprincipes van programmeren aan de orde, zoals het gebruik van variabelen.



figuur 3 Blokken bouwen door te programmeren in Numworx Digitale Wiskunde Omgeving<sup>[9]</sup>

## Computationeel denken 'unplugged'

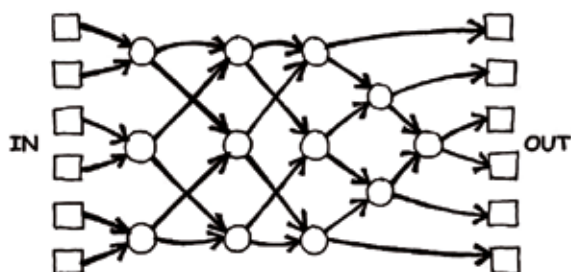
Computationeel denken vereist niet noodzakelijkerwijs een computer. Het kan ook 'unplugged' beoefend worden. Op de website 'CS unplugged'<sup>[10]</sup> staan veel voorbeelden van unplugged activiteiten, bijvoorbeeld over sorteeralgoritmes.



figuur 4 Unplugged computationeel denken activiteit voor docenten<sup>[11]</sup> >

Andere voorbeelden zijn te vinden op KlasCement, dat lesmateriaal aanbiedt voor allerhande niveaus en onderwerpen (van primair tot beroepsonderwijs, van aardrijkskunde tot wiskunde) waarin eenvoudig gezocht en gefilterd kan worden.<sup>[12]</sup> Een voorbeeld van unplugged computationeel denken is een serie video's met doe-het-zelf-materiaal, puzzels, spelletjes en programmeeruitdagingen voor dagelijks gebruik in het gezin en op school. De video's zijn in het Engels en ondertiteld in het Nederlands.

Een website over de link tussen wiskunde en informatica is MATHmaniaCS. Mathmaniacs zijn 'persons exhibiting an excessive passion for MATHematics and Computer Science!'<sup>[13]</sup> De overzichtspagina van lessen biedt een gevarieerde serie onderwerpen voor unplugged computationeel denken, zoals binaire getallen, logica, grafen en sorteren, zie figuur 5.



figuur 5 Sorteernetwerk om te gebruiken op de grond

## Computationeel denken met tools

Een voorbeeld van computationeel denken, waarbij gebruik gemaakt wordt van in het onderwijs gangbare tools, is ontwikkeld in een onderzoek naar computationeel en wiskundig denken van de Universiteit Utrecht in samenwerking met Radboud Universiteit, SLO en vijf scholen.<sup>[14]</sup> In dit project is in de afgelopen twee jaar lesmateriaal ontwikkeld met als uitgangspunt om dicht bij het wiskundecurriculum te blijven. Het materiaal voor 5 vwo wiskunde A sluit aan bij het onderwerp statistiek, gebruikmakend van Excel. Voor wiskunde B is het onderwerp functies en procedures voor het opstellen van raaklijnen met GeoGebra.<sup>[15]</sup> De lessen combineren 'unplugged' werk op papier met het gebruiken van tools op de computer. Het materiaal is inmiddels uitgetest op verschillende scholen.

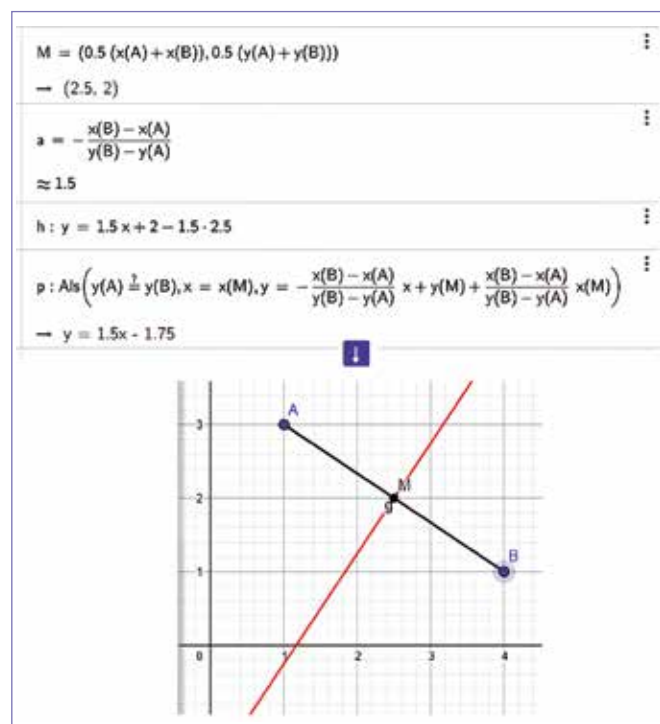
In de wiskunde-A-lessen wordt gewerkt met de dataset met passagiersgegevens van de Titanic en wordt onderzocht of men uit de gegevens kan afleiden of 'vrouwen en kinderen eerst' van boord zijn gegaan. Met behulp van Excel worden gegevens uit kolommen gecombineerd en met formules wordt geoefend in het gebruik van condities

en constructies zoals als-dan-anders, zie figuur 6. Na deze oefening met de gegevens van de Titanic passen leerlingen het geleerde toe in een tweede grote dataset waarbij ze een eigen onderzoeksvraag formuleren en onderzoeken. De voorlopige resultaten van het gebruik van de wiskunde-A-lessen laten zien dat leerlingen na het leren kennen van de functies in Excel beter weten hoe ze met behulp van de functies verbanden kunnen onderzoeken met Excel en dat een probleem hierbij in kleinere deelstappen kan worden opgedeeld.

pass	survived	name	sex	age	allop
1	1	Allen, Miss. Elisabeth Walton	female	29	0
1	1	Allison, Master. Hudson Trevor	male	0,92	1
1	0	Allison, Miss. Helen Loraine	female	2	1
1	0	Allison, Mr. Hudson Joshua Creighton	male	30	1
1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1
1	1	Anderson, Mr. Harry	male	48	0
1	1	Andrews, Miss. Kameelia Theodosia	female	63	1
1	0	Andrews, Mr. Thomas Jr	male	39	0
1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53	2

figuur 6 Dataset Titanic in Excel: Vrouwen en kinderen eerst?

In de wiskunde B-lessen worden op papier procedures uitgewerkt voor het opstellen van bijvoorbeeld raaklijnen of middelloodlijnen, die vervolgens in GeoGebra worden ingevoerd, zie figuur 7. De beschikbare knoppen (bv. voor de



figuur 7 Algemene oplossing voor middelloodlijn met gebruik van condities in GeoGebra

middelloodlijn) worden niet gebruikt, maar de leerling gaat de stappen zelf programmeren. Om die procedures generiek te maken, worden parameters gebruikt, naast iteratie-functies en als-dan-anders condities in GeoGebra-formules. Leerlingen maken zo kennis met principes uit het programmeren binnen een tool die ze al kennen uit de wiskundeles. De resultaten van het gebruik van de wiskunde-B-les- en laten zien dat de combinatie van 'plugged' en 'unplugged' activiteiten een zinvolle strategie is om aspecten van computationeel denken toe te passen.<sup>[16]</sup> De uitdaging zit voor de leerlingen vooral in het leren kennen van de tool, de syntax en in het gebruik van constructies met als-dan-anders. Het lesmateriaal wordt het komend schooljaar verder onderzocht en verbeterd, en na afloop van het project beschikbaar gesteld samen met een docentenhandleiding.

## Tot besluit

Computationeel denken kan op veel manieren tot uiting komen. De kunst is om een passende vorm te kiezen voor de gegeven situatie. Hopelijk heeft dit artikel je ideeën aangereikt over de verschillende mogelijkheden en vormen waarin computationeel denken geleerd kan worden, en nodigt het je uit om zelf te gaan verkennen waar computationeel denken in jouw les een plaats kan krijgen.

## Noten

- [1] Website SLO over digitale geletterdheid: <https://www.slo.nl/thema/meer/digitale-geletterdheid/>, vakportaal digitale geletterdheid: <https://www.slo.nl/vakportalen/vakportaal-digitale-geletterdheid/>
- [2] Simon Peyton-Jones (2014), TED lezing 'Teaching creative computer science': <https://youtu.be/la55clAtDMs>, in het Engels.
- [3] Python programmeren op de rekenmachine, materiaal op de TI website: <https://resources.t3vlaanderen.be/t3vlaanderen-home/>
- [4] Python in de klas: <https://feliennegitbook.io/python-in-de-klas/>
- [5] Blockly opdrachten: <https://www.program-uurtje.org/blockly.html>
- [6] Scratch opdrachten: [https://scratch.mit.edu/projects/editor/?tutorial=getStarted\\_\(kies+Nederlands\)](https://scratch.mit.edu/projects/editor/?tutorial=getStarted_(kies+Nederlands))
- [7] Website 'De Schoonheid en Vreugde van Programmeren', leren programmeren met Snap!: <https://bjoc-nl.github.io/>

- [8] Europees project <colette>: <https://colette-project.eu/>, deels gefinancierd door Erasmus+ van de EU, 20201-DE03-KA201-077363. De ontwikkelde materialen komen in de loop van de komende twee jaar beschikbaar. Voor meer informatie, neem contact op met Sylvia van Borkulo ([s.vanborkulo@uu.nl](mailto:s.vanborkulo@uu.nl))
- [9] Numworx Digitale Wiskunde Omgeving: <https://www.numworx.nl/>
- [10] Website 'CS unplugged', computer science without a computer: <https://csunplugged.org/>
- [11] Uit: Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K. (2019). Educational policy and implementation of computational thinking and programming: Case study of Singapore. In *Computational thinking education* (pp. 345-361). Springer, Singapore.
- [12] Website KlasCement: <https://www.klascement.net/lesmateriaal>
- [13] Website MATHmaniaCS: <http://mathmaniacs.org/>
- [14] Deze lessenseries zijn ontwikkeld in het kader van het project 'Computationeel denken en wiskundig denken: digitale geletterdheid in wiskundecurricula', gefinancierd door NRO (projectnummer 00517751).
- [15] Van Borkulo, S. P., & Drijvers, P. (2020). Het Flzier gericht op... Computationeel denken in de wiskundeles. *Euclides*, 96(3), 20-21.
- [16] Sylvia van Borkulo, Christos Chytas, Paul Drijvers, Erik Barendsen, and Jos Tolboom. 2021. Computational Thinking in the Mathematics Classroom: Fostering Algorithmic Thinking and Generalization Skills Using Dynamic Mathematics Software. In *The 16th Workshop in Primary and Secondary Computing Education (WiPSCE '21), October 18-20, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3481312.3481319>

## Over de auteur

Sylvia van Borkulo is universitair docent bij het Freudenthal Instituut van de Universiteit Utrecht en houdt zich bezig met technologie en wiskundeonderwijs. E-mailadres: [s.vanborkulo@uu.nl](mailto:s.vanborkulo@uu.nl)