

Open source grondwatermodellering met MODFLOW 6

RUBEN CALJÉ, FRANS SCHAARS, DAVID BRAKENHOFF EN ONNO EBBENS

Grondwatermodellen worden steeds vaker vanuit een script opgebouwd. We presenteren hier een MODFLOW 6 grondwatermodel dat geheel via Python-scripts wordt opgebouwd. Tevens presenteren we een aanpak voor het gebruik en delen van scripts binnen de Nederlandse geohydrologische wereld. We hopen hiermee bij te dragen aan de inhoudelijke beantwoording van hydrologische vraagstukken. Door het delen van scripts en tools hoeven we minder aandacht te besteden aan de programmeer-technische details, en blijft er meer tijd over om gekalibreerde modellen op te stellen die passen bij de hydrologische vraagstukken.

Essay

Inleiding

In Nederland worden grondwatermodellen gebruikt voor verschillende doeleinden. Zo worden er intrekgebieden van drinkwaterwinningen mee vastgesteld, effectberekeningen van natuurherstelprojecten mee uitgevoerd of klimaatsscenario's mee doorgerekend. Grondwatermodellen zijn verschillend in detailniveau, afhankelijk van het type vraag dat ermee beantwoord moet worden. De modelleur kiest welke processen in het model worden opgenomen. Deze keuze is afhankelijk van de te beantwoorden vraag, de beschikbare data, de mogelijkheden van de gebruikte software en de beschikbare tijd.

De meeste in Nederland gebruikte grondwatermodellen hebben MODFLOW als rekencode, ontwikkeld door de United States Geological Survey (USGS). Deze modellen worden van oudsher via een gebruikersinterface aangestuurd, zoals bijvoorbeeld Visual Modflow of iMOD. De laatste jaren is de aansturing via scripts, geprogrammeerd in MATLAB, R of Python steeds populairder geworden, vanwege de voordelen van transparantie en reproduceerbaarheid. Hierbij worden modelbestanden met een computerscript aangemaakt, en zorgen de scripts voor de pre- en post-processing van de modeldata. Voor Python is er de FloPy package (Bakker e.a., 2016), actief onderhouden door de USGS, waarmee MODFLOW-modellen kunnen worden aangestuurd.

Binnen de Nederlandse hydrologische wereld zijn de laatste jaren verschillende ontwikkelingen op het vlak van gescript modelleren. Zo heeft Deltares de iMOD-Python package ontwikkeld (<https://gitlab.com/deltares/imod/imod-python>), om MODFLOW- en iMODFLOW-modellen aan te sturen. iMod-Python is daarmee een alternatief voor FloPy.

Daarnaast vinden diverse trajecten plaats om het Nederlands Hydrologisch Instrumentarium (NHI) opnieuw op te bouwen (Programmateam NHI, 2019). Deze trajecten worden door verschillende marktpartijen uitgevoerd, wat zal resulteren in een modelgenerator voor iMod-modellen via batch-files.

MODFLOW 6

In 2017 is MODFLOW 6 uitgekomen (Langevin e.a., 2017), met een groot aantal verbeteringen ten opzichte van eerdere versies van MODFLOW. Het rekengrid kan lokaal verfijnd worden, modellagen hoeven niet vlakdekkend aanwezig te zijn en het droogvallen van cellen wordt robuust berekend. Het uitbreiden van MODFLOW 6 is voor de USGS ook eenvoudiger geworden, omdat het koppelen van verschillende typen modellen is ingebouwd. Zo is het sinds 2020 mogelijk om stoftransport in MODFLOW 6 te simuleren, inclusief dichtheidsafhankelijke stroming (Langevin e.a., 2022). De genoemde verbeteringen waren individueel in eerdere MODFLOW-versies (2005, NWT, USG en SEAWAT) opgenomen, maar konden vaak niet worden gecombineerd. Daarnaast heeft de USGS samen met Deltares een MODFLOW-*Application Programming Interface* (API) ontwikkeld, waarmee gebruikers tijdens een modelrun invloed kunnen uitoefenen op de berekening (Hughes e.a., 2022). Elke twee jaar vindt de conferentie MODFLOW & More plaats, waar nieuwe ontwikkelingen worden besproken, en gebruikers ervaringen uitwisselen. Door gebruik te maken van FloPy kunnen alle recente ontwikkelingen aan MODFLOW 6 worden gebruikt, met volledige vrijheid voor de gebruiker. De kwaliteit van de door de USGS geproduceerde en onderhouden software is hoog, met uitgebreide documentatie. Dit maakt het voor programmeurs mogelijk deze software direct te gebruiken. Dit biedt een groot voordeel ten opzichte van hoogwaardige software (user interfaces) van andere partijen, waar de gebruiker gelimiteerd is door de packages en mogelijkheden die de bouwers van deze software ondersteunen.

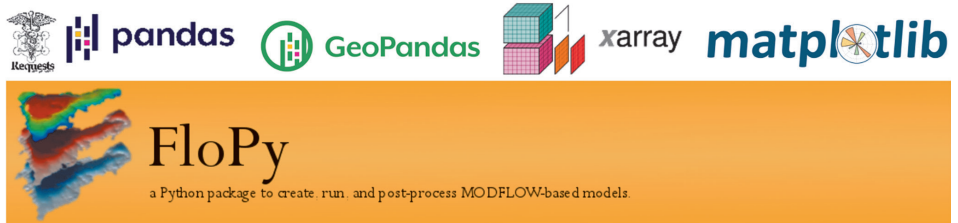
Openbare data

Naast open source software is er ook steeds meer openbaar beschikbare data van het Nederlandse geohydrologische systeem. Zo zijn de ondergrondmodellen van **REGIS** en **GeoTOP** online beschikbaar als NetCDF-bestanden (<https://www.dinoloket.nl/modelbestanden-aanvragen/netcdf>), waarmee eenvoudig ruimtelijke uitsnede uit deze modellen te maken zijn. Het Actueel Hoogtebestand Nederland (**AHN**) en **vaklodgingen** (hoogte zeebodem en IJsselmeer) kunnen worden geraadpleegd voor een nauwkeuriger maaiveldhoogte, op land en op zee. De locatie van afwatering kan worden afgeleid uit de waterdeel-laag van de Basisregistratie Grootschalige Topografie (**BGT**), aangevuld met informatie over het oppervlaktewater van de waterschappen, die vaak ook online beschikbaar is. Invoer van neerslag en verdamping kan gescript worden gedownload van het **KNMI**. In het kader van het NHI of de BRO zou het in de toekomst makkelijker moeten worden om informatie over **grondwateronttrekkingen** (inclusief berekening) te verkrijgen. Als dat eenmaal gelukt is, kan een model volledig op basis van online data gemaakt worden, via een script. Dit maakt het ook eenvoudig om een model te actualiseren met nieuwe data.

Python-packages

Vele beschikbare Python-packages (naast FloPy en iMod-Python) helpen om de data van verschillende bronhouders te downloaden, te interpreteren en om te zetten in modelinvoer. Veelgebruikte packages zijn (zie ook Afbeelding 1):

- Requests: downloaden van verschillende typen data,
- Pandas: beheer van tijdreeksen of tabeldata,
- GeoPandas: beheer van geometrische data, zoals de vlakken van de BGT,
- Xarray: beheer van multidimensionale modeldata, zoals lagenmodellen,
- Rasterio: rasterdata aggregeren naar modelcellen of oppervlaktewatervlakken,
- Matplotlib: visualiseren van modelinvoer en resultaten.



Afbeelding 1: De logo's van enkele genoemde Python packages

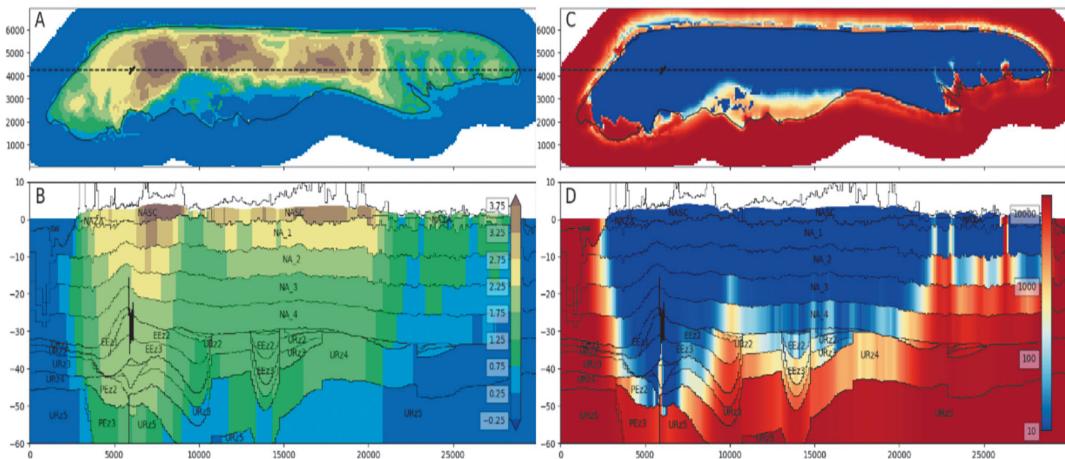
Naast packages ten behoeve van de modelbouw zijn nog vele andere packages beschikbaar voor de modelleur, ter ondersteuning van zijn werkzaamheden, zoals:

- SciPy: optimaliseren van een model via kleinste kwadraten of andere methoden,
- Pastas: uitvoeren van tijdreeksanalyse op grondwaterstand-reeksen,
- TimML en TTim: maken van analytische grondwatermodellen.

Voorbeeld

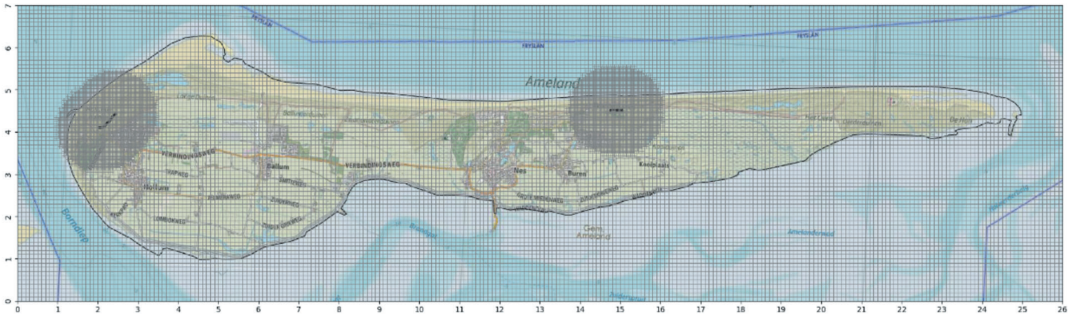
Ter illustratie van de mogelijkheden van MODFLOW 6 en de beschikbare data in Nederland presenteren we een modelgenerator voor de Waddeneilanden. Op de onttrekkingen van Vitens na, is het volledige model opgebouwd op basis van openbare data die via open source software is verwerkt. De schematisatie van de ondergrond is opgesteld op basis van REGIS, waarbij de Holocene deklaag is onderverdeeld met behulp van de laagindeling en grondsoorten uit GeoTOP. Hierbij wordt aan elke grondsoort een doorlatendheid toegekend, waarna deze wordt opgeschaald tot de gekozen laagindeling. De grondwateraanvulling wordt bepaald uit data van het KNMI, waarbij de verdamping ook afhankelijk is van het uit de BGT afgeleide landgebruik. De locatie van het oppervlaktewater is tevens van de BGT afgeleid, waarna informatie van het waterschap (de peilen) aan deze polygonen zijn toegevoegd.

Om de dichtheidseffecten bij de zoetwaterbel op de eilanden te berekenen, wordt stoftransport in de berekening meegenomen, waarbij het zoutgehalte invloed heeft op de dichtheid van het grondwater. Dikke watervoerende pakketten zijn hierbij opgedeeld in dünnere rekenlagen, voor voldoende verticale resolutie (zie Afbeelding 2).



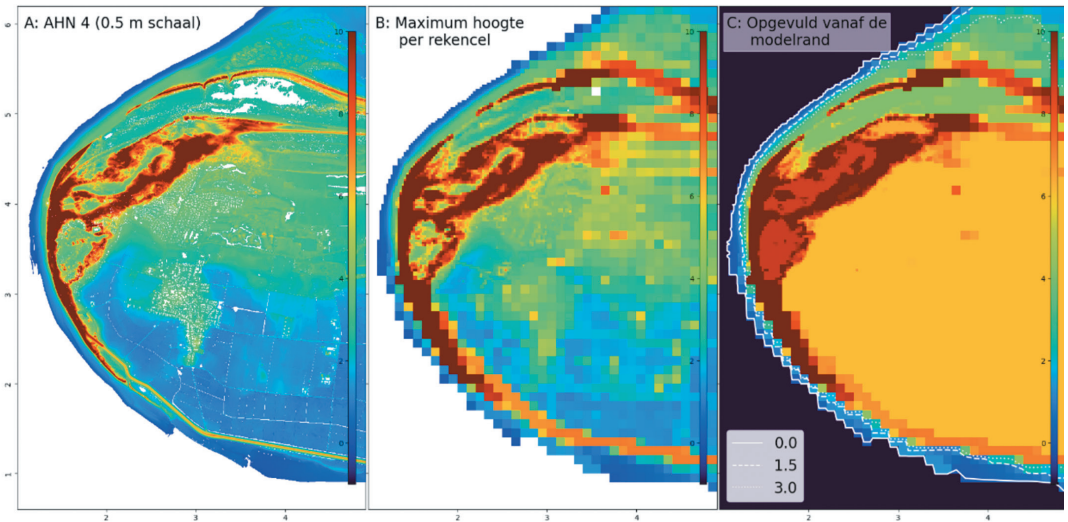
Afbeelding 2: De berekende grondwaterstand (A) en de stijghoogte in een dwarsdoorsnede door de ondergrond (B), in m NAP, en de bijbehorende chloride-concentratie (C en D), in mg/l, op Terschelling. De onttrekkingsfilters zijn weergegeven als zwarte punten (A en C) en lijnen (B en D).

Het doel van de berekeningen bij de Waddeneilanden was om de intrekgebieden van de winlocaties van Vitens vast te stellen. Daarom wordt het grid in een zone van 1000 meter rond de putten verfijnd met behulp van het programma GRIDGEN van de USGS (zie afbeelding 3). De intrekgebieden worden bepaald door middel van stroombaananalyse met behulp van MODPATH. Al deze programma's van de USGS zijn ondersteund in FloPy, met overzichtelijke broncode en voorbeelden.



Afbeelding 3: Het modelgrid voor de berekeningen op Ameland, met verfijning rond de putten.

Een voorbeeld van het gebruik van de mogelijkheden van de verschillende Python-packages bij de Waddeneilanden is de bepaling van de locatie van de zee-randvoorwaarde. Allereerst is, vanuit het AHN 4 op 0,5 meter schaal (afbeelding 4A), met behulp van rasterio het maximum maaiveld in elke cel (afbeelding 4B) bepaald. Vervolgens wordt een opvulalgoritme (uit de Python-package scikit-image) toegepast vanuit de modelranden, zodat voor elke cel bekend is bij welk zeepeil de cel overstroomt (afbeelding 4C). Uit deze kaart kan de locatie van de zee-randvoorwaarde bij een bepaald zeepeil worden afgeleid, te gebruiken bij analyses van het getijde, of van zeespiegelstijging. De algoritmen voor bovenstaande analyse zijn in de gebruikte Python-packages geoptimaliseerd, en werken snel zodat het niet nodig is om tussenresultaten op te slaan.



Afbeelding 4: De bepaling van de zee-randvoorwaarde, op basis van het AHN 4, in het model van Ameland.

Gescripte modelbouw

De voordelen van gescripte modelbouw zijn legio. De berekeningen zijn reproduceerbaar. Het is mogelijk kleine aanpassingen in het script aan te brengen, zoals het vergroten van het modeldomein of het verlengen van de modelperiode, die grote invloed hebben op de resultaten. Het is mogelijk om de modelresultaten in het script meteen te vergelijken met metingen, of om een optimalisatie uit te voeren van modelparameters. Een ander voordeel is eenvoudiger gebruik van versiebeheer. De scripts kunnen op GitHub/GitLab worden geplaatst, zodat bijgehouden wordt wie wat wijzigt en zodat voorbeeldscripts automatisch worden getest. Deze platforms bieden tevens een goede infrastructuur om bugs te vermelden, wensen uit te spreken of discussies te kunnen voeren, optioneel direct gelinkt aan code in de scripts.

Nadelen van een volledig geautomatiseerde workflow zijn er echter ook. De modelleur moet kunnen programmeren, in dit geval in Python. Het uitzoeken van de werking van de verschillende Python-packages vergt tijd, wat ook geldt voor het leren kennen van de mogelijkheden van de verschillende packages in MODFLOW 6. Net als bij het maken van een model via een GUI kunnen verschillende combinaties van modelinvoer ervoor zorgen dat een grondwatermodel niet meer convergeert.

Bij deze problemen geldt dat een oplossing sneller is gevonden als de Nederlandse hydrologen elkaar helpen en elkaars scripts hergebruiken, zodat niet iedereen tegen dezelfde problemen aanloopt. Met dit idee hebben we een eerste aanzet gedaan voor een Python-package, genaamd *nlmod* (<https://github.com/ArtesiaWater/nlmod>). Met deze package is het mogelijk om:

- 1 de discretisatie van een model in ruimte en tijd op te geven,
- 2 data uit externe bronnen (AHN, REGIS, KNMI, enz.) te downloaden,
- 3 modeldata af te leiden en een modelberekening uit te voeren,
- 4 de resultaten te visualiseren en te analyseren.

Met *nlmod* kan hiermee op basis van alleen een modelextent een modelberekening worden uitgevoerd. *nlmod* maakt gebruik van *FloPy* voor het aanmaken van de modelbestanden, zodat maximaal gebruik kan worden gemaakt van de mogelijkheden van MODFLOW 6.

Het is nog een uitdaging om ook hydrologen zonder programmeerwens en/of -ervaring mee te krijgen. De scripts dienen zo overzichtelijk mogelijk te zijn (zie afbeelding 5 voor een beoogd toekomstbeeld), waarbij technische zaken zo veel mogelijk naar de achtergrond worden geplaatst. Het moet eenvoudiger worden om de modellen inclusief invoer in te zien in bepaalde geografische software, zoals QGIS of iMOD, zodat de gebruiker de invoer in voor hem bekende software verder kan analyseren. In QGIS is het via de zogenaamde mesh-layer mogelijk om ook modeldata van onregelmatige grids in te laden.

Naast het gebruik kan ook het installeren van de benodigde Python-packages lastig zijn. Een oplossing hiervoor kan het rekenen op een server zijn, waar alle benodigde packages reeds op zijn geïnstalleerd. Het beperken van de afhanke-


```

1  """
2  This example shows a basic model of 6 by 6 km, using cells of 50 by 50 m,
3  and a border of 1 km around this area with cells of 100 by 100 m.
4  """
5
6  # %% Imports of Python packages
7  import nlm
8  import imod
9
10 # %% Define model properties
11 extent = [153000.0, 161000.0, 410700.0, 418700.0]
12 tmin = '2010-1-1'
13 tmax = '2020-1-1'
14 freq = '14d'
15
16 # %% Get data for the specified extent
17 layers = nlm.get_layers(extent, sources=['regis', 'geotop'])
18 wells = nlm.get_wells(extent)
19 sw = nlm.get_surface_water(extent)
20
21 # %% Generate grid with a refinement of 1 (resulting in cells of 50 m)
22 refinement = {1: [154000.0, 160000.0, 411700.0, 417700.0]}
23 grid = nlm.generate_grid(extent, dx=100., refinement=refinement)
24
25 # %% Generate a model
26 ml = nlm.get_base_model(grid, layers, tmin=tmin, tmax=tmax, freq=freq)
27
28 # %% Download data from knmi and add recharge to the model
29 nlm.add_recharge(ml, method='linear')
30
31 # %% Add surface-water without aggregating surface water bodies
32 nlm.add_surface_water(ml, sw, method='individual')
33
34 # %% Add wells using the Multi-Aquifer Well (MAW) Package
35 nlm.add_wells(ml, wells, method='maw')
36
37 # %% Run model
38 ml.run_model()
39
40 # %% Post-processing, for example plot the average head in REGIS-layer PZWAz3
41 head = ml.get_head()
42 head.sel(layer='PZWAz3').mean('time').plot()
43
44 # %% Or plot the GHG
45 gxg = imod.evaluate.calculate_gxg(head.sel(layer='BXz3'))
46 gxg['ghg'].plot()
47

```

Afbeelding 5: De beoogde eenvoud van een script voor een grondwatermodel.

liikheden van andere Python-packages kan ook een deel van de oplossing zijn, samen met een goede installatiehandleiding.

Het automatisch downloaden van data kan daarnaast voor problemen zorgen als de online beschikbare data veranderen. Hierdoor verandert ook de modelberekening, hoewel er niets aan het script zelf is aangepast. Door middel van duidelijk versiebeheer, niet alleen voor de scripts, maar ook voor de data, dient dit probleem te worden ondervangen.

Veel Python-packages bevatten efficiënte methoden om parameters voor de modelcellen van het grondwatermodel af te leiden. Voor grote gebieden kan het

desondanks enige tijd duren om data te downloaden of te verwerken. Wanneer veel modelruns uitgevoerd worden, bijvoorbeeld in een kalibratie, heeft het de voorkeur om alleen de gewijzigde modelonderdelen opnieuw te genereren. Hiervoor zijn verschillende technieken beschikbaar, zoals caching van data of het gebruik van workflow-managers als Snakemake.

Tot slot

Ondanks de toename van de rekenkracht van computers, zijn grondwatermodellen meestal nog log en traag. Daarom dient de modelleur concessies te doen, en te bepalen waar de werkelijkheid vereenvoudigd kan worden. De berekening moet dus aangepast worden aan de vraag die je wilt beantwoorden. Ben je geïnteresseerd in het effect van eb en vloed op de grondwaterstanden in de duinen? Maak een model met tijdstappen van een uur, download de data van Waterinfo, en verander de zee-randvoorwaarde in elke tijdstap. Of wil je weten wat de invloed is van het afdammen van duinvalleien op de GLG en de GHG? Maak een model met grotere tijdstappen, verfijn het rekengrid rond de duinvalleien en gebruik de Lake en Stream Flow Routing packages van MODFLOW, zodat ook de waterbalans van het oppervlaktewater in de duinen wordt bijgehouden.

Het grote voordeel van deze aanpak is dat elke vraag een passend model krijgt, waarbij de opschaling vanuit de basisdata pas plaatsvindt tijdens het doorrekenen van het model. We denken dat dit veel nauwkeurigere en realistischere antwoorden geeft dan het rekenen met een voor een ander (of zonder) doel ontwikkeld regionaal model, waarbij de vooraf gedefinieerde schaal zelden past bij de vraag.

Daarnaast denken we dat het omarmen van moderne ICT-technieken het modelleren eenvoudiger maakt. Voorbeelden hiervan zijn het gebruik van Python-scripts, voorbeelden in Jupyter Notebooks, versiebeheer via GitHub/GitLab, automatische tests van scripts en caching van tussenresultaten. Hiervoor hebben we een aanzet gedaan in nlmod, wat we graag samen met de Nederlandse geohydrologische gemeenschap verder ontwikkelen.

We denken dat het automatisch genereren van modellen en het delen van code de toekomst van het modelleren is. Nu de scripts beter worden, en er meer gedeeld wordt in de hydrologische wereld, worden standaardbewerkingen eenvoudiger en sneller. Hiermee hoeft minder tijd besteed te worden aan technische zaken, en meer aan het opstellen van het juiste model voor de te beantwoorden vragen en het kalibreren van deze modellen.

Dankwoord

De berekeningen voor de Waddeneilanden zijn uitgevoerd in opdracht van de provincie Fryslân. We zijn dankbaar dat we enkele resultaten van deze studie mochten delen. De ontwikkeling van nlmod is mede mogelijk gemaakt door PWN, Hoogheemraadschap Hollands Noorderkwartier, Vitens en Waternet.

Literatuur

Bakker, M., V. Post, C.D. Langevin, J.D. Hughes, J.T. White, J.J. Starn, J. J. en M.N. Fielen (2017) Scripting MODFLOW Model Development Using Python and FloPy; in: *Groundwater*, vol 54, pag 733-739.

Hughes, J.D. M.J. Russcher, C.D. Langevin, E.D. Morway, en R.R. McDonald (2022), The MODFLOW Application Programming Interface for simulation control and software interoperability; in: *Environmental Modelling & Software*, volume 148.

Langevin, C.D., J.D. Hughes, E.R. Banta, R.G. Niswonger, S. Panday en A.M. Provost (2017) Documentation for the MODFLOW 6 Groundwater Flow Model; in: *U.S. Geological Survey Techniques and Methods*, book 6, chap A55.

Langevin, C.D., A.M. Provost, S. Panday en J.D. Hughes (2022) Documentation for the MODFLOW 6 Groundwater Transport Model; in: *U.S. Geological Survey Techniques and Methods*, book 6, chap A61.

Programmateam NHI (2019) Uitwerking investeringsplan 2020-2021.

Summary Open source groundwater modeling using MODFLOW 6

Groundwater models are increasingly built from a script. We present a MODFLOW 6 groundwater model that is built from scratch by Python scripts. Also, we present an approach for the use and sharing of scripts in the Dutch geohydrological community. We hope this will contribute to answering hydrological research questions. By sharing scripts and tools, we have to pay less attention to the programming-technical details, and more time is left to draw up models that fit the hydrological issues.

Auteurs

RUBEN CALJÉ

Artesia

r.calje@artesia-water.nl

FRANS SCHAARS

Artesia

f.schaars@artesia-water.nl

DAVÍD BRAKENHOFF

Artesia

d.brakenhoff@artesia-water.nl

ONNO EBBENS

Artesia

o.ebbens@artesia-water.nl

