
MODFLOW-optimalisatie met OpenDA

Harry Boukes en Nils van Velzen¹

Grondwatermodellen worden gevoed met bodemparameters, maar hoe bepaal je nu die parameterwaarden? In dit artikel worden de drukmetingen in de pompputten van een winveld gebruikt om kD -, c - en S -waarden te bepalen. Eerst wordt het programma MLU gebruikt, waarna de uitwerking geschikt wordt gemaakt voor MODFLOW. De optimalisatie wordt dan uitgevoerd met PEST. Omdat die berekeningen wel erg lang duren, is er een alternatief gezocht en gevonden in OpenDA. Een optimalisatie van 19 parameters leidt in ons geval dan in twee uur tot een resultaat.

Het model als middel

Om te kunnen adviseren over beleid en beheer van het watersysteem, is kennis van dat systeem een eerste randvoorwaarde. Een hydroloog heeft grofweg twee methoden om die kennis te verkrijgen: meten en modelleren. Een meting vertelt de hydroloog hoe de situatie is, een meetreeks hoe de situatie was. Om te voorspellen hoe een systeem gaat worden zijn modellen nodig.

Voor de modellering van grondwaterproblemen is er in de loop der jaren standaard-programmatuur ontwikkeld, zodat niet steeds vanaf het begin een rekenmodel opgetuigd hoeft te worden. MODFLOW (McDonald en Harbaugh, 1984) is zo'n reken-programma dat afgelopen decennia veelvuldig is toegepast. Het faciliteert een reken-omgeving voor tijdsafhankelijke 3-D grondwaterstanden en stroming. Om een goede weergave van de werkelijke stroming te verkrijgen moeten er een groot aantal parameterwaarden worden ingevuld en om te controleren hoe goed de werkelijkheid wordt benaderd, moeten de modeluitkomsten altijd vergeleken worden met gemeten waarden.

(Automatische) kalibratie

Nu zijn die parameterwaarden op voorhand niet altijd even goed bekend en soms ook afhankelijk van de wijze waarop het probleem geschematiseerd is. In dat geval mogen ze zo bijgeregeld worden, dat er een goede overeenkomst ontstaat tussen gemeten en berekende waarden (kalibratie). Sinds de jaren negentig is zo'n kalibratie geen hand-

¹ Adviesburo Harry Boukes, info@harryboukes.nl en VORtech BV, nils.vanvelzen@vortech.nl

matig trial-and-error-proces meer, maar bieden (andere) programma's de mogelijkheid om rekenkundig die parameterwaarden te vinden waarbij het verschil tussen berekende en gemeten waarden minimaal is (automatische kalibratie).

In de jaren negentig was het programma PEST (Parameter ESTimation) één van de eerste programma's waarmee gemeten en berekende waarden op elkaar afgestemd konden worden (Doherty, 1994). PEST vormt een soort schil om het feitelijke rekenprogramma heen. De gebruiker geeft aan welke parameterwaarden geoptimaliseerd moeten worden. PEST maakt dan invoerbestanden aan, en laat het programma lopen. Vervolgens leest het de uitvoerbestanden, en pakt daar de berekende waarden uit die met de metingen vergeleken moeten worden. Dan stelt PEST de parameterwaarden bij, maakt nieuwe invoerbestanden aan en laat het programma opnieuw lopen. Het onderzoekt hoe de verschillen tussen gemeten en berekende waarden veranderen en past opnieuw de parameterwaarden zodanig aan dat deze verschillen steeds kleiner worden. Op een gegeven moment lukt het niet meer om de resultaten te verbeteren, en de dan verkregen parameterwaarden worden als optimaal gepresenteerd.

De omslachtigheid van PEST volgt direct uit bovenstaande omschrijving: de berekening moet steeds van begin tot eind worden doorlopen, invoer en uitvoer moeten steeds volledig worden afgerond voordat PEST de vergelijking tussen gemeten en berekende waarden kan uitvoeren. De makers van MODFLOW hebben daarom besloten om de mogelijkheid parameters te optimaliseren in het programma zelf op te nemen. De invoer hoeft slechts eenmaal te worden ingelezen en het programma berekent intern de verschillen tussen gemeten en berekende waarden. Omdat alle data in interne bestanden in het werkgeheugen van de computer staan, zijn aanpassingen in de data snel gemaakt: er hoeft geen volledige tussenuitvoer en data-invoer te worden gegenereerd. Dit leverde zeker in het tijdperk van traag schrijvende harde schijven tijdswinst op, ervan uitgaande dat de optimalisatiemethoden vergelijkbaar zijn.

Het nadeel van efficiëntie

Hoewel de voordelen van interne parameter-optimalisatie evident zijn, ontnemt het de hydroloog soms ook flexibiliteit: de parameters moeten immers wel als zodanig herkenbaar in MODFLOW opgenomen zijn. Er zijn situaties waarbij de parameterwaarden voor MODFLOW door preprocessing worden gegenereerd. Als die preprocessing buiten MODFLOW om plaats vindt, kunnen de betreffende parameters niet binnen MODFLOW geoptimaliseerd worden. Als een hydroloog zich daar niet bij neer wil leggen, zal er een stap terug in de tijd worden gedaan, en opnieuw gebruik worden gemaakt van een programma als PEST.

De onbedoelde pompproef

Een waterleidingbedrijf onttrekt grondwater doorgaans uit vertikaal geplaatste pompputten. Zodra een pomp aanschakelt, zakt het niveau in de put, waardoor het water door het bodemmateriaal komt toestromen. Afhankelijk van het bodemmateriaal kunnen

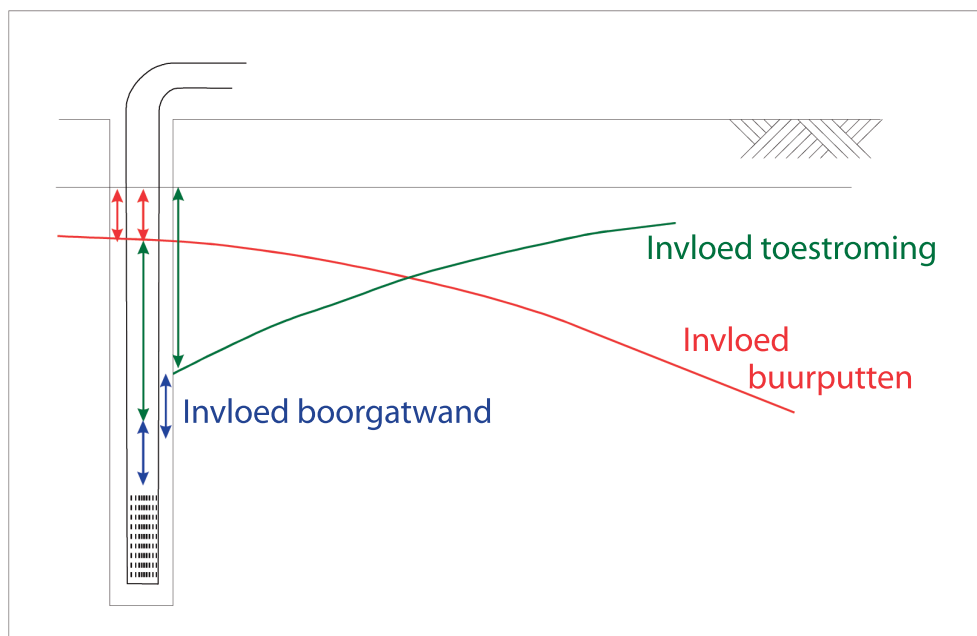
hele fijne bodemdeeltjes met het grondwater meestromen. Deze deeltjes hebben de neiging om zich rond de boorgatwand te verzamelen en daar de bodem te verstoppert. Het gevolg is dat (uitgaand van een vast debiet) de niveaudaling in de put na aanschakelen (afpompings) in de loop van de tijd steeds groter wordt. Als de afpompings zo groot wordt dat de pomp droogvalt, betekent dit schade aan de pomp en verontreiniging van de ruwwaterleidingen (ijzerneerslag) en mogelijk bacteriële besmetting van de zuivering.

Veel pompputten zijn daarom voorzien van drukmeting, zodat bij een te grote afpompings de betreffende pomp uitgeschakeld wordt. Als deze drukmetingen worden geregistreerd, blijkt uit beschouwing van de meetreeksen dat het niveau niet alleen daalt als de betreffende pompput bijschakelt. Ook als een buurput schakelt wordt een reactie waargenomen. Dat is een prettige bijkomstigheid, want op die manier wordt zichtbaar hoe het betreffende pakket reageert. In feite is er sprake van een onbedoeld uitgevoerde pompproef. Dit fenomeen is in 2007 voor het eerst verkend en beschreven (Boukes, 2007). In 2013 is een toepassing in de praktijk beschreven (Boukes, 2013). In beide publicaties gaat het om de uitwerking van een dataset van de waterwinning Drongelen van Brabant Water N.V.

Voor Drongelen is de afpompings in drie componenten uitgesplitst:

1. Drukverval bij het passeren van de boorgatwand.
2. Drukverval als gevolg van de toestroming van het verpompte water door het pakket.
3. Beïnvloeding door omliggende pompputten.

De componenten 2 en 3 worden bepaald door de eigenschappen van het watervoerend pakket. Als deze bekend zijn en de totale afpompings wordt gemeten, volgt uit een



Afbeelding 1: De totale afpompings in een pompput is uitgesplitst in drie componenten.

simpele optel- en aftreksom het drukverval over de boorgatwand. Veranderingen hierin kunnen realtime online worden geregistreerd. Ook is het sommetje erg makkelijk te maken tot hoe ver (uitgaande van een gelijkblijvende weerstand op de boorgatwand) de afpompingsdruk kan oplopen als op een warme zomerdag alle putten tegelijk aan moeten staan. Op die manier is de noodzaak om te regenereren aangetoond voordat de afpompingsdruk kritische waarden bereikt.

Uitwerking van de onbedoelde pompproef

Voor Boukes (2013) was het de uitdaging om de eigenschappen van het watervoerend pakket af te leiden louter uit de metingen in de pompputten. Dat lukte vrij behoorlijk, in ieder geval voldoende goed om de componenten 2 en 3 uit de metingen te filteren. Dit leverde een consistente reeks verstoppingswaarden op. Het hydrologisch systeem was vereenvoudigd tot drie parameters: het doorlaatvermogen van het gepompte pakket (kD-waarde), de bergingscoëfficiënt (S-waarde) en de weerstand van de afdekkende laag (c-waarde, en dat dan nog even los van de vraag wat een c-waarde in het systeem van Drongelen precies voorstelt). Er werd gebruik gemaakt van het programma MLU van Kick Hemker (zie www.microfem.com) omdat dit programma de mogelijkheid biedt om de weerstand rond de boorgatwand in de berekeningen mee te nemen. MLU heeft een eigen ingebouwde optimalisatiemogelijkheid, waarmee elf parameters werden bepaald: de drie bodemparameters en de acht boorgatweerstanden van de acht pompputten.

Van MLU naar MODFLOW

Om de toepassingsmogelijkheden uit te breiden, is aansluitend getest of de methode met MODFLOW dezelfde resultaten geeft. In dat geval wordt het immers ook in heterogene gebieden mogelijk om geautomatiseerde metingen als onbedoelde pompproef uit te voeren. Nadeel van MODFLOW is dat het geen mogelijkheden kent om bij een vaste flux het drukverval bij het passeren van de boorgatwand te verdisconteren. Dit effect moet dan ook via pre- en postprocessing aan de berekende resultaten worden toegevoegd. En dat maakte het onmogelijk om gebruik te maken van de ingebouwde optimalisatiemogelijkheden van MODFLOW.

De eerste vraag in dit verband luidde of met MODFLOW vergelijkbare verlagingen worden berekend in een gegeven situatie. Waar de analytische methode van MLU ook rond de boorgatwand op elk punt een exacte waarde berekent, was het maar de vraag in hoeverre een lijn, geïnterpoleerd door een reeks cellen een zelfde resultaat op zou leveren. Het bleek echter al snel dat bij een netwerk wat rond de pompputten sterk is verdicht, tot op de centimeter identieke resultaten werden verkregen.

Hierna diende de pre- en postprocessing te worden aangebracht, waarbij er afhankelijk van het debiet en een verstoppingsweerstand een extra afpompingsdruk aan de berekende waarden werd toegevoegd, zodat MODFLOW zich verder alleen met de hydrologische invloed van het pakket hoefde te bemoeien. Voor hydrologen die ook maar een beetje kunnen programmeren zijn zulke programma's een fluitje van een cent.

De combinatie MODFLOW - PEST

Ook PEST is een programma met al een lange historie. Het programma blijkt nog steeds te bestaan en is freeware. Op hoofdlijnen blijkt het programma nog net zo aan te sturen en te functioneren als in de jaren negentig: de gebruiker moet niet bang zijn om in een zwart ASCII-scherm commando's in te typen. De handleiding is wollig, maar alles staat er wel in.

Wat al een beetje te verwachten was: de combi MODFLOW en PEST is tientallen procenten trager dan MLU, maar daar was het uiteindelijk ook niet om te doen. Sowieso is elke vergelijking van snelheid tricky, omdat de rekensnelheid erg afhankelijk kan zijn van startwaarden, ingestelde marges en optimalisatie-parameters. Al in de eerste runs berekende PEST vergelijkbare parameterwaarden als MLU, dus dat gaf vertrouwen. Alleen bleek de kwadraten-som (de som van alle gekwadrateerde verschillen tussen berekende en gemeten waarden, dat is de functie waarop PEST het minimum zoekt) bij PEST twee keer zo groot als bij MLU. Nadere beschouwing leverde dat de tijds-discretisatie nog aandacht vroeg. De interpolatie in de tijd tussen de door-gerekende tijdstappen is bij MODFLOW lineair. Bij een meettijdstip kort na aanschakelen van de pompput, en nog voor de eerste rekenstap is uitgevoerd, levert dat veel onnauwkeurigheid. Opnieuw blijkt de analytische benadering zoals van MLU charmanter, omdat de tijd-stijghoogte continu in de tijd wordt berekend. Maar nadat hier rekening mee gehouden was, liep ook de kwadraten-som naar waarden waar MLU op uitgekomen was.

Meer mogelijkheden?

Op dit punt aangekomen, ontstond er een extra optimalisatiewens. Bij gericht uitgevoerde pompproeven wordt er altijd eerst een evenwichtssituatie gecreëerd en vervolgens een pompput aan- of juist uitgezet. De veranderingen van de grondwaterstand zijn dan direct aan de ingreep toe te wijzen. Bij een onbedoelde pompproef schakelen de pompputten naar believen aan en uit. Op $t=0$ van de metingen is er dus geen onbeïnvloede evenwichtssituatie, er is op dat moment een invloed van de bedrijfsvoering voorafgaand aan $t=0$. Sowieso moeten dus de schakelingen voorafgaand aan $t=0$ dus ook aan MLU of MODFLOW aangeboden worden, maar dan is de invloed daarvan op de drukmetingen nog onbekend: deze is immers afhankelijk van de kD -, c - en S -waarde, en die moeten nog bepaald worden. Bij de exercitie met MLU was al duidelijk geworden dat de $t=0$ -verlaging ten opzichte waarvan de metingen fluctueren handmatig ingesteld moest worden en dat de keuze de uitkomsten beïnvloedde, zowel qua parameterwaarden als qua hoogte van de kwadraten-som. Als die $t=0$ -verlaging niet goed ingesteld was, werd dat zichtbaar in de verlagingcurves, bijvoorbeeld als de eindverlaging te ver ging afwijken, of juist als de put uitstond en er toch verlagingen werden berekend. Via trial and error zijn met MLU de beginverlagingen in de acht pompputten op tijdstip $t=0$ bepaald, maar het heeft uiteraard de voorkeur om die acht waarden automatisch te laten bepalen.

Omdat het verdisconteren van de $t=0$ -verlagingen een bewerking van de metingen is, was dit met MLU niet mogelijk: een meting is een meting en die kun je niet verschuiven.

Ook binnen MODFLOW geldt een meting als vaste waarde. Omdat PEST werkt als een schil om één of meer programma's kan er eenvoudig nog extra pre- en postprocessing worden toegevoegd (dus ook van de meetdata) waarna binnen het geheel de gewenste parameters geoptimaliseerd worden.

Het wordt wel wat veel, allemaal...

Op voorhand was het allerminst zeker dat deze opzet zou slagen. De ervaring leert dat optimalisatie slechts op een beperkt aantal parameters tegelijk kan worden uitgevoerd, omdat anders afhankelijkheden ontstaan waarbij de ene parameter een afwijking van de andere kan compenseren. In dat geval kunnen er meerdere 'goede' oplossingen zijn die tot een minimale kwadratenom leiden.

In de berekening Drongelen was het aantal te optimaliseren opgevoerd tot negentien: acht $t=0$ -verlagingen, acht boorgatweerstanden en drie bodemparameters. Normaal gesproken is dit tot mislukken gedoemd, maar zonder een uitgebreide analyse te hebben gemaakt was er toch wel hoop dat het met de afhankelijkheden mee zou vallen. Zo is evident dat de boorgatweerstand van een put vrijwel 100% correleert met de kD -waarde: als de kD -waarde 10% te hoog wordt genomen, kan dezelfde berekende afpompijswaarde weer worden verkregen door ter compensatie de boorgatweerstand te laag te nemen. Alleen als de put uit staat, is de berekende druk uitsluitend afhankelijk van de kD -waarde, want bij een stilstaande put is er geen drukverschil over de boorgatwand. Juist de situatie als de put uit staat levert een bepaling op van de drie bodemparameters, waarna de boorgatwandweerstand hierop worden afgestemd. Daarbij is helder dat de ene boorgatweerstand onafhankelijk is van de andere zeven. Op een zelfde manier viel te verwachten dat de $t=0$ -verlagingen per pompput sowieso onafhankelijk van elkaar waren, en door situaties aan te bieden waar alle putten zowel aan als uit staan, ook onafhankelijk van de bodemparameters.

..., maar het lukt!!

In een eerste grove berekening raakte de optimalisatie volledig de weg kwijt. Door in stapjes de marges van de parameter ruimte in te perken, bleek het uiteindelijk mogelijk toch alle negentien parameters in één run te optimaliseren. Weliswaar duurde de berekening 71,5 uur, werd MODFLOW 452 keer aangeroepen, maar uiteindelijk werkt de methode dus toch!

OpenDA

Zodra iets goed gaat, is de volgende vraag of het ook beter kan. De werkwijze was immers wel erg 20-ste eeuws, terwijl we in de 21-ste eeuw toch al een aardig eindje op weg zijn. Bij de zoektocht naar een alternatief voor PEST, werd ik gewezen op het bestaan van OpenDA (zie www.openda.com en (Ridler, 2014)). Dit is een open source initiatief van de TU-Delft, VORtech en Deltares. Het doel van OpenDA is om een

raamwerk aan te bieden waarin verschillende kalibratietechnieken en data-assimilatie toegepast kunnen worden op willekeurige rekenmodellen. Kalibratie is het proces waarbij modelparameters afgeregeld worden zodat de model voorspellingen overeen komen met gemeten waarden. Data-assimilatie combineert (real time) metingen met modelvoorspellingen zodat het model continu op de werkelijkheid blijft afgeregeld. Alle hiervoor bestaande wiskundige algoritmes hebben hun eigen toepassingen, sterke en zwakke punten. De basisingrediënten van al deze methodes zijn steeds hetzelfde: een model, waarnemingen en een algoritme, maar in de praktijk is het niet eenvoudig om van methode te wisselen omdat deze vaak als individueel programma beschikbaar is. Elk programma kent daarbij een eigen manier om de waarnemingen in te lezen en om het model aan te sturen.

OpenDA is bedoeld als toolbox, waarbinnen verschillende rekenmethoden worden aangeboden. Dit biedt de eindgebruiker de mogelijkheid om volop te experimenteren met verschillende algoritmen om zo uit te vinden wat het beste werkt voor een gegeven probleem. Aan de andere kant biedt het ontwikkelaars van data assimilatie en kalibratie algoritmes de mogelijkheid om nieuwe methodes snel te kunnen uitproberen op een breed scala van modellen.

Koppeling met MODFLOW

Wanneer een model voor de eerste keer geschikt gemaakt wordt voor OpenDA, moet er een koppeling geprogrammeerd worden. In het geval van MODFLOW hebben we ervoor gekozen om een black-box-koppeling te maken. Bij een black-box-koppeling zal OpenDA MODFLOW aansturen door invoerbestanden klaar te zetten en na een MODFLOW-run de relevante uitvoer in te lezen. In feite is dit een zelfde aanpak als bij PEST. Omdat er voor MODFLOW nog geen koppeling met OpenDA beschikbaar was moesten er drie kleine stukjes Java geprogrammeerd worden voor:

- Het lezen en schrijven van parameters in de MODFLOW invoer bestanden;
- Het lezen van de uitvoer reeksen van MODFLOW;
- Het lezen van meetreeksen in PEST formaat.

Het kunnen inlezen van meetreeksen in PEST-formaat is niet noodzakelijk voor de koppeling maar was wel erg handig omdat de reeksen nu eenmaal al in dat formaat beschikbaar waren en we deze nu niet hoefden te converteren naar een ander, al reeds ondersteund formaat. Het schrijven van deze koppeling is in principe een eenmalige actie waarvan het resultaat in de toekomst voor vele soorten van MODFLOW-kalibraties hergebruikt kan worden. De nu beschikbare versie heeft ons geleerd op welke wijze op de MODFLOW-invoer kan worden ingegrepen, maar is verder nog erg basaal. Inmiddels zijn er plannen ontwikkeld om zo'n koppeling flexibeler en gebruiksvriendelijker te maken.

De volgende stap is dan het maken van een model configuratie. Het maakt de black-box-koppeling immers niet uit voor welke programma's invoerbestanden worden gegenereerd. In ons geval moesten we voorafgaand aan de MODFLOW-berekening de $t=0$ -verlagingen worden verdisconteerd en na afloop van de MODFLOW-berekening de verlaging als gevolg van de boorgatwand passage worden toegevoegd. Dit is

gerealiseerd door een treintje van vijf programma's aan te bieden, waarvan MODFLOW de middelste was.

Ten slotte moet de gebruiker nog een laatste configuratiebestand schrijven waarin aangegeven wordt welk algoritme en welke meetreeksen gebruikt worden voor de kalibratie.

Zou het beter gaan?

Door op deze manier met de berekening om te gaan, werden een aantal keuzes nog eens tegen het licht gehouden. Zo bleek in de optimalisatie van MODFLOW niet voor de meest efficiënte methode te zijn gekozen. De eerste winst werd al behaald voordat er één berekening met OpenDA werd uitgevoerd: MODFLOW bleek met een andere instelling drie keer zo snel te rekenen. In plaats van 71,5 uur bleek PEST met 452 modelruns nog 'maar' 24 uur en 18 minuten nodig te hebben.

Binnen OpenDA zijn op dit moment vier optimalisatietechnieken beschikbaar te zijn. Drie ervan bleken niet in staat om een correct optimum te komen, in ieder geval niet bij de begininstellingen zoals die aan de berekening werden aangeboden. De techniek waar het meeste van verwacht werd (DuD, zie (Ralstom en Jennrich, 1978)) leverde wel een resultaat en ook nog eens veel sneller dan PEST: opeens waren er nog maar 67 berekeningen van MODFLOW nodig, en dat kostte nog slechts 3 uur en 48 minuten. De resultaten waren zo goed als identiek aan die van PEST. Kort daarna leidde de aanschaf van een nieuwe computer nog eens tot een halvering van de rekentijd.

Als een berekening binnen een paar weken tijd opeens 36 keer zo snel wordt uitgevoerd, tsja, dat mag toch wel vooruitgang worden genoemd...

Nadere beschouwing

Latere berekeningen gaven aan dat DuD nog wel gevoelig is voor de startwaarden waarmee de berekening wordt begonnen. Eén keer had het pakket slecht 53 berekeningen nodig om de optimale waarden te vinden, maar ook waren er situaties waarbij de berekening oneindig rondjes bleef draaien. Dat is op zich niet ongebruikelijk bij optimalisatieprogramma's, maar in die zin is OpenDA met DuD dus ook geen wonderprogramma.

Dat roept de vraag op of dit resultaat niet een toevalstreffer zou kunnen zijn. Wij denken van niet, omdat we begrijpen waarom DuD een snellere techniek is dan PEST.

Waarom DuD beter werkt dan PEST

In zijn algemeenheid is het doel van parameteroptimalisatie dat het programma die parameterwaarden vindt waarbij het verschil tussen gemeten en berekende waarden minimaal is. Het verschil tussen gemeten en berekend wordt uitgedrukt in de som van

de kwadraten van alle verschillen. De hoogte van die kwadratensom is afhankelijk van welke parameters aan het model worden meegegeven. Voor alle parameters wordt een startwaarde opgegeven, en het optimalisatieprogramma rekent eerst uit welke kwadratensom wordt berekend bij deze startwaarden.

Vervolgens gaat het programma alle te optimaliseren parameters één voor één af, en verschuift ze iets. Voor al die verschoven parameters wordt berekend hoe de kwadratensom verandert. Hiermee creëer je een p -dimensionale vector, waarbij p het aantal te optimaliseren parameters is.

Het programma gaat in de richting van die p -dimensionale vector zoeken, maar weet op voorhand niet precies hoe ver het moet zoeken. De meeste programma's gebruiken een eerste orde Taylor-benadering om dit punt te vinden.

Als het in die richting gezochte punt gevonden is, betekent het niet dat de som klaar is. Hydrologische sommen zijn vaak niet-lineaire systemen, en op het gezochte punt aangekomen heeft de vector van de kwadratensom mogelijk een andere p -dimensionale richting, waardoor de kwadratensom nog lager kan worden berekend. Een programma als PEST gaat vanaf dit gevonden punt opnieuw alle parameters af om te kijken in welke p -dimensionale richting er een verdere daling van de kwadratensom mogelijk is.

Dud benadert dit proces op een iets andere manier. In de eerste slag wordt nog steeds voor alle parameters een zoekrichting en helling berekend, maar daarna bewaart het programma de informatie uit de vorige $p+1$ berekeningen. Het programma benadert de functie met een lineaire functie die precies overeenkomt met de berekende waarden van de laatste $p+1$ berekeningen. Hierdoor hoeft Dud telkens maar één nieuwe berekening uit te voeren om een vernieuwde zoekrichting te berekenen. Zeker bij een groot aantal parameters (zoals bij de grote regionale modellen) scheelt dat heel veel berekeningen. Weliswaar zijn er wat meer iteraties nodig, maar als er ter voorbereiding daarvan maar één berekening uitgevoerd hoeft te worden in plaats van p , dan kan de tijdswinst flink oplopen.

Vervolgstappen

De komende tijd zullen zowel ten aanzien van de onbedoelde pompproef als de optimalisatie met OpenDA vervolgstappen worden gezet. Er is in december 2013 een meetperiode ingesteld bij de winning Seppe. Zonder dat de exploitatie er merkbare hinder van ondervond, is een dataset verkregen die in de loop van 2014 zal worden uitgewerkt naar bodemparameters, $t=0$ -verlagingen en boorgatweerstanden. Omdat het 21 pompputten betreft, loopt het aantal te optimaliseren parameters op tot 45. Bij ons zijn geen hydrologische berekeningen bekend waarbij zoveel parameters tegelijk geoptimaliseerd zijn.

Ten aanzien van OpenDA is een voorstel ingediend om een gebruikersschil te programmeren, zodanig dat toekomstige gebruikers van de combinatie MODFLOW met OpenDA (al dan niet met verdere pre- en postprocessing) geen Java-programmeerkennis hoeven te hebben.

Literatuur

Boukes, H. (2007) Het uitwerken van een pompproef die nooit is uitgevoerd; *Stromingen*, jaargang 13, nummer 3.

Boukes, H. (2013) Nooit meer putverstoppingsmetingen; Vakblad H2O, gepubliceerd via de website, 2013.

Doherty, J. (1994) PEST: Model-Independent Parameter Estimation; Watermark Numerical Computing, Third Edition.

McDonald, M.G. en Harbaugh, A.W. (1984) A modular three-dimensional finite-difference ground-water flow model; U.S. Geological Survey Open-File Report 83-875.

Ralston M.L en Jennrich R.I. (1978) Dud, A derivative-free algorithm for nonlinear least squares; *Technometrics* Vol 20 no 1.

Ridler, M.E., Velzen, N. van, Hummel, S., Sandholt, I., Falk, A.K., Heemink, A. en Madsen, H. (2014) Data assimilation framework: Linking an open data assimilation library (OpenDA) to a widely adopted model interface (OpenMI); *Environmental Modelling and Software* jaargang 2014 nummer 1.