
Snelle oudjes gaan MATLAB

Kees Maas
Theo Olsthoorn

Samenvatting

De komst van de computer, eind jaren '60, ging gepaard met een opmerkelijke opbloei van numerieke methoden die tot dan toe in de marge van de wiskunde een kwijnend bestaan geleden hadden. De sleutel tot hun succes lag feitelijk in de beperkingen van de vroege computers: die waren voornamelijk heel goed in snel optellen en aftrekken. Nu, vijftientig jaar later, wordt het door de snelle opkomst van geavanceerde wiskunde-software plotseling weer aantrekkelijk om terug te grijpen op oude vertrouwde analytische methoden. In dit artikel behandelen we met name de formules van Mazure, De Glee en Bosch. We maken van de gelegenheid gebruik om te laten zien hoe zogenaamde matrixfuncties in combinatie met MATLAB ongekend effectief ingezet kunnen worden om deze formules te generaliseren tot formules voor gelaagde systemen. Oplossingen die voorheen weken programmeerwerk vroegen, kosten in MATLAB letterlijk enkele regels. Met een regel extra krijgt u ook nog een plaatje, kant en klaar voor in uw rapport.

Inleiding

“Snelle oudjes” was de pakkende titel waaronder Sake van der Schaaf een tijdje terug het gebruik van de formule van Mazure voor ons opfriste (Van der Schaaf, 1995). Mazures formule stamt uit 1932; het is er maar één in een hele reeks van analytische modelletjes die sinds de komst van de computer geleidelijk in onbruik zijn geraakt. Zo'n 20 jaar geleden was het artikel van Van der Schaaf nog ondenkbaar geweest: Mazure was basiskennis, en een beetje hydroloog wist prima hoe hij varianten of geheel nieuwe formules moest afleiden als hij zich geconfronteerd zag met situaties waarin de handboeken niet hadden voorzien. Hoe komt het dat deze handige modelletjes zo snel vergeten zijn? Misschien omdat er maar een paar echt handig in het gebruik waren. De meeste analytische oplossingen waren gecompliceerde uitdrukkingen, die het uiterste vroegen van de destijds beschikbare rekenhulpmiddelen. Numerieke methoden bestonden al wel, maar die waren alleen een reële optie voor hydrologen met organisatietalent: een MODFLOW-achtig probleem vergde een zaaltje vol menselijke rekenaars en het doorrekenen van alternatieven vereiste een beslis-

Kees Maas is werkzaam bij KIWA Onderzoek en Advies, Postbus 1072, 3430 BB Nieuwegein, telefoon (030) 606 95 11, telefax (030) 606 11 65, e-mail: kmaas@kiwaoa.nl en bij Sectie Hydrologie en Ecologie, Technische Universiteit Delft.

Theo Olsthoorn is werkzaam bij Gemeentewaterleidingen Amsterdam, Vogelenzangseweg 21, 2114 BA Vogelenzang, telefoon (023) 523 35 69, telefax (023) 528 14 60, e-mail: t.olsthoorn@gw.amsterdam.nl

sing die niet zomaar op de werkvloer genomen werd. Omdat de eerste computers niet veel meer konden dan bliksemsnel optellen en aftrekken, waren het juist deze numerieke methoden die bij de komst van de computer plotseling en onverwacht een hoge vlucht namen. Ook de analytische methoden profiteerden van de computer, maar veel minder opzienbarend, doordat er nog nauwelijks wiskundige software beschikbaar was.

Die situatie verandert snel. De groei van de reken- en geheugencapaciteit van computers is nog steeds indrukwekkend, maar de spectaculaire ontwikkelingen zien we momenteel toch gebeuren op het gebied van de software. Geladen met moderne wiskundeprogramma's zijn computers allang geen domme nummerkrakers meer, en de toekomst ligt weer helemaal open voor methoden die een groter appel doen op de menselijke geest. In dit artikel willen we aan een paar voorbeelden laten zien hoe analytische formules voor stroming in gelaagde poreuze media eenvoudig hanteerbaar worden, door gebruik te maken van – het klinkt tegenstrijdig – het numerieke wiskundepakket MATLAB.

Matrixfuncties

Iedere hydroloog is wel vertrouwd met algemene wiskundige functies $f(x)$, waarin x een willekeurig reëel getal voorstelt. Tegenwoordig wat minder bekend zijn complexe functies $f(z)$, waarin z een complex variabele is. Alle wiskunde is boeiend, maar complexe functies zijn bovendien van een ontroerende schoonheid. Vroeger vonden ze veel toepassing in de geohydrologie. Wie gevoel heeft voor historie en op zoek is naar een spannende hobby, zou eens de eerste (!) druk van Verruijts 'Theory of Groundwater Flow' moeten lezen. Complexe functies zijn nog steeds een fort van analytisch ingestelde hydrologen. Ze vinden uitgebreid toepassing in de analytische-elementenmethode.

Nog minder bekend dan complexe functies zijn zogenaamde matrixfuncties $f(A)$, waarin A een vierkante matrix is. De theorie van de matrixfuncties is betrekkelijk onontwikkeld, maar wat er beschikbaar is, is wonderwel bruikbaar voor het formuleren van grondwaterstroming in gelaagde media (Maas, 1986). Toch was het nut van matrixfuncties tot voor kort beperkt, omdat er geen software bestond die ermee overweg kon. Dat is nu verleden tijd: MATLAB begrijpt matrixfuncties even goed als functies van reële of complexe variabelen.

Matrixfuncties zijn dus functies die in plaats van een enkel getal een matrix als argument hebben. Ze zijn net zo gedefinieerd als gewone functies, namelijk door hun machtsreeks. Zo is de exponentiële matrixfunctie $\exp A$ te berekenen door in de welbekende machtsreeks van $\exp x$ het argument x te vervangen door een willekeurige vierkante matrix A :

$$\exp A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \dots \quad (1)$$

Hoofdletter I is de eenheidsmatrix, dat is een vierkante matrix vol nullen, behalve op de hoofddiagonaal, waar uitsluitend enen staan. In technische literatuur is het gebruikelijk om matrices dik te drukken, maar daarin schuilt geen groot voordeel, dus we kiezen hier voor een gewoon slank lettertype. Omdat de vermenigvuldiging van een matrix met zichzelf een eenduidig gedefinieerde bewerking is, is het een koud kunstje om met de bovenstaande

formule de functie $\exp A$ te berekenen. Daaruit blijkt meteen dat $\exp A$ zelf een matrix is, met dezelfde afmetingen als A .

Elke wiskundige functie heeft een matrix-evenknie. Weliswaar kan niet elke wiskundige functie in een machtsreeks ontwikkeld worden, maar daarvoor zijn oplossingen bedacht die we hier gevoeglijk buiten beschouwing kunnen laten. Het evalueren van matrixfuncties is namelijk een wiskundige taak die de ontwikkelaars van MATLAB van ons hebben overgenomen. Probeer u eens in MATLAB :

```
> A = [1,2,3;4,5,6;7,8,9]
A =
1     2     3
4     5     6
7     8     9
> expm(A)
ans =
1.0e+006 *
1.1189    1.3748    1.6307
2.5339    3.1134    3.6929
3.9489    4.8520    5.7552
```

De regels na de prompt » zijn ingetypt; de rest is door MATLAB zelf gegenereerd. Met de eerste opdracht maakt u een matrix aan, die door MATLAB geëchood wordt. (U kunt de echo desgewenst onderdrukken door de opdracht af te sluiten met een puntkomma). Met de tweede opdracht berekent u de exponentiële matrixfunctie. Matrixfuncties hebben in MATLAB een m achter hun naam, dus: expm in plaats van exp. (De functie exp werkt ook op een matrix, maar heeft een heel andere uitkomst! Verwar ze dus niet).

We veronderstellen dat de moderne lezer de basale regels van het rekenen met matrices wel kent. Voor de zekerheid herinneren we eraan dat het matrixproduct AB ongelijk is aan het matrixproduct BA , behoudens bijzondere gevallen. In zulke gevallen zegt men dat de matrices A en B commuteren. Een belangrijk voorbeeld is de functie $f(A)$; die commuteert met A zelf, dus $A f(A) = f(A) A$. Deze eigenschap wordt overigens in dit artikel niet gebruikt, maar hij kan van pas komen als u zelf aan het experimenteren slaat.

In het navolgende zullen we regelmatig producten tegenkomen van de vorm $x A$, waarin A een vierkante matrix voorstelt die alleen afhangt van bodemconstanten, terwijl x de plaatsvariabele is, dus een 'scalair' getal. In het bijzonder zullen we werken met functies $f(xA)$. Zulke functies kunnen gewoon naar x gedifferentieerd worden, bijvoorbeeld:

$$\frac{d e^{xA}}{dx} = A e^{xA} \quad (2)$$

Op dezelfde manier is

$$\frac{d K_0(rA)}{dr} = -A K_1(rA) \quad (3)$$

waarin nu r de rol van plaatsvariabele vervult. Dit is allemaal overeenkomstig de normale, scalaire theorie. Ook het limiet-gedrag van de functies die we straks nodig hebben lijkt erg op wat we gewend zijn:

$$\begin{aligned}
 \lim_{x \rightarrow \infty} \exp xA &= \infty \\
 \lim_{x \rightarrow \infty} \exp -xA &= 0 \\
 \lim_{x \rightarrow \infty} I_0(rA) &= \infty \\
 \lim_{x \rightarrow \infty} K_0(rA) &= 0 \\
 \lim_{x \rightarrow \infty} K_1(rA) &= \frac{1}{r}
 \end{aligned}
 \tag{4}$$

De systeemmatrix voor meerlagenstroming

We beschouwen nu het algemene rekenschema voor de stroming van grondwater in gelaagde systemen (figuur 1). De differentiaalvergelijking voor een willekeurige aquifer (met rangnummer i) luidt:

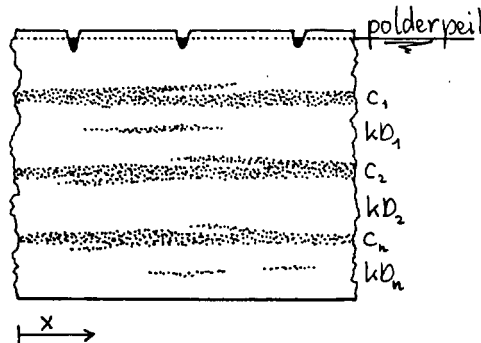
$$\nabla^2 \varphi_i + \frac{\varphi_{i-1} - \varphi_i}{kD_i c_i} + \frac{\varphi_{i+1} - \varphi_i}{kD_i c_{i+1}} = 0
 \tag{5}$$

Hierin is kD_i de kD -waarde van laag i , c_i is de c -waarde, en φ_i is de stijghoogte. De stijghoogte aan de bovenzijde van het systeem kiezen we als referentieniveau, dus $\varphi_0 = 0$. In figuur 1 is de basis ondoorlatend, zodat $c_{n+1} = \infty$, maar daarop kunt u natuurlijk zelf varianten bedenken.

Net als in de lineaire algebra gebruikelijk is, kan dit stelsel van vergelijkingen in matrixvorm gegoten worden:

$$\nabla^2 \varphi - A \varphi = 0
 \tag{6}$$

Hier is φ (zonder index!) een kolomvector die als elementen de stijghoogten φ_i in de verschillende lagen bevat, terwijl A een driediagonaalmatrix is, die we *systeemmatrix* noemen. Voor een systeem dat drie aquifers bevat, neemt A de volgende gedaante aan:



Figuur 1: Geohydrologisch schema

$$A = \begin{bmatrix} \frac{1}{kD_1c_1} + \frac{1}{kD_1c_2} & -\frac{1}{kD_1c_2} & 0 \\ -\frac{1}{kD_2c_2} & \frac{1}{kD_2c_2} + \frac{1}{kD_2c_3} & -\frac{1}{kD_2c_3} \\ 0 & -\frac{1}{kD_3c_3} & \frac{1}{kD_3c_3} \end{bmatrix} \quad (7)$$

Als u de moeite neemt om uitdrukking (6) te controleren, ziet u eenvoudig in hoe A eruit zou zien voor een willekeurig aantal aquifers.

De systeemmatrix speelt een belangrijke rol in dit artikel. Omdat we hem vaak nodig hebben, schrijven we daarvoor een M-file, die de matrix samenstelt als kD en c opgegeven worden:

```
function A = sysmat(kD,c)
% De functie sysmat(kD,c) stelt de systeem-matrix samen.
% kD en c zijn kolomvectoren!
n = length(kD);
a = 1./(kD.*c);
b = 1./(kD.*[c(2:n);inf]);
A = diag(a+b)-diag(a(2:n),-1)-diag(b(1:n-1),1);
```

(Zegt het u op dit moment nog niets? Een M-file is een reeks instructies, die opgeslagen wordt voor later gebruik. Klik op *File, New, M-file*, dan verschijnt er een Windows-kladblok waarin u het bovenstaande overtypt. Daarna klikt u op *Bestand en Opslaan als*, om dit programmaatje een naam te geven. Bij ons heet hij *sysmat.m*).

Probeer u eens:

```
>> kD = [100;200;300;400];
>> c = [500;600;700;800];
>> A = sysmat(kD,c)
A =
  1.0e-004 *
  0.3667   -0.1667    0         0
 -0.0833    0.1548   -0.0714    0
  0         -0.0476    0.0893   -0.0417
  0         0         -0.0312    0.0312
```

Als alles goed gegaan is vindt u hetzelfde antwoord. U zult waarschijnlijk de naam *sysmat* wel kunnen onthouden, maar over een poosje weet u vast niet meer met welke parameters deze functie moest worden aangeroepen. Was het eerst kD en dan c , of juist andersom? Type dan

```
>> help sysmat
```

dan verschijnen op uw scherm de regels uit de M-file *sysmat.m* die met *%* gemerkt zijn:

```
De functie sysmat(kD,c) stelt de systeem-matrix samen.
kD en c zijn kolomvectoren!
```

Mafzure: een generalisatie van de formule van Mazure

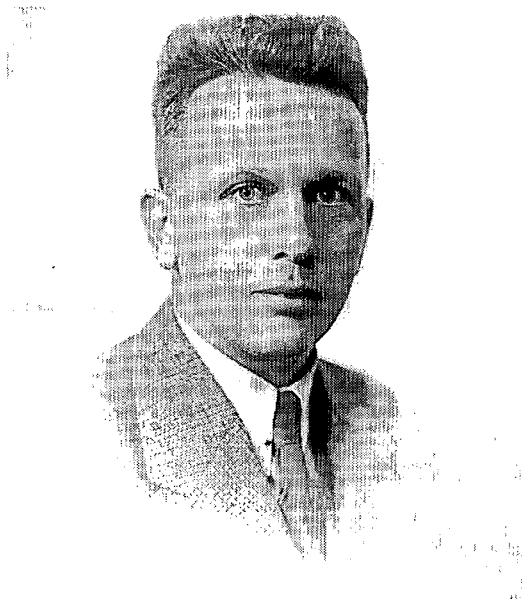
We zijn nu klaar voor een meerslagen-variant op de formule van Mazure, die we Mafzure zullen noemen ('maf' staat voor 'multiple aquifer flow'). Figuur 2 geeft het geohydrologische schema weer. Laten we aannemen dat het buitenwater h meter hoger staat dan het polderpeil. De heersende differentiaalvergelijking is vergelijking (6). De randvoorwaarden die bij dit stromingsgeval passen zijn direct in vectorvorm op te schrijven:

$$\begin{aligned}\varphi(0) &= h \\ \varphi(\infty) &= 0.\end{aligned}\tag{8}$$

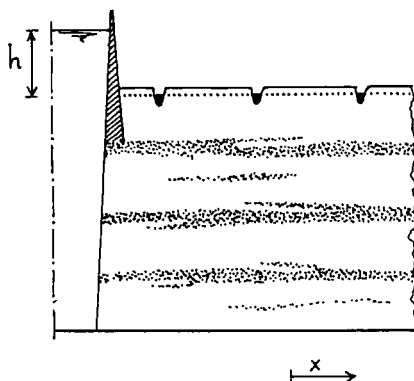
In de bovenste uitdrukking moet h geïnterpreteerd worden als een kolomvector: hij bevat de stijghoogten in de afzonderlijke aquifers ter plaatse van $x = 0$. Geheel overeenkomstig het scalaire geval is de algemene oplossing van vergelijking (6) te schrijven als

$$\varphi(x) = \exp(-x\sqrt{A})c_1 + \exp(x\sqrt{A})c_2\tag{9}$$

Als u wilt kunt u de juistheid van deze uitdrukking controleren, door hem in vergelijking (6) in te vullen. c_1 en c_2 zijn integratieconstanten, of liever -vectoren, die opgelost moeten worden uit randvoorwaarden. (In de scalaire wiskunde is het gebruikelijk om c_1 en c_2 vóór de exponentiële functies te schrijven, maar de conventies van de matrixrekening vergen dat ze daarachter staan). Het is direct in te zien dat c_2 nul moet zijn, anders zou de stijghoogte landinwaarts onbeperkt blijven toenemen. Voor c_1 blijft dan de waarde h als enige mogelijkheid over.



Prof.dr.ir. J.P. Mazure (hier op een foto uit 1937) begon zijn loopbaan bij de voormalige Dienst Zuiderzeewerken. De formule die onder Nederlandse geohydrologen zijn naam draagt, leidde hij af tijdens een onderzoek naar de ontzilting van de Wieringermeerpolder. Van 1956 tot 1968 was Mazure als hoogleraar verbonden aan de Technische Hogeschool te Delft, van welke instelling hij ook conrector geweest is. Aanvankelijk doceerde hij toegepaste mechanica, later bouwconstructies.



Figuur 2: Geohydrologisch schema Mazure

De *gegeneraliseerde formule van Mazure* luidt dus:

$$\varphi(x) = \exp(-x\sqrt{A})h \quad (10)$$

Als voor de matrix A een enkele getal ingevuld wordt staat hier gewoon de oorspronkelijke formule van Mazure. Dit extreem compacte resultaat werd al eerder afgeleid (Maas, 1984). Het is geldig voor een willekeurig aantal aquifers. Wie niet onder de indruk is van de efficiëntie van deze methode zou het eens op de conventionele manier moeten proberen, al is het maar voor twee aquifers.

Hoewel met deze uitdrukking het vraagstuk wiskundig gesproken volledig opgelost is, ging in het verleden het voordeel van de matrixnotatie grotendeels weer teniet bij het numeriek uitwerken van de formule. Door de komst van MATLAB komt de charme van matrixfuncties pas volledig tot haar recht. MATLAB begrijpt formules zoals vergelijking (10) vrijwel letterlijk. Probeer u maar eens:

```

» h = [1;1;1;1];
» x = 500;
» phi = expm(-x*sqrtm(A))*h
phi =
0.2779
0.5273
0.6726
0.7355

```

(We zijn er vanuit gegaan dat u de systeemmatrix A nog had. Zo niet, maak hem dan even opnieuw aan, zoals in de vorige paragraaf is voorgedaan).

U ziet dat dit soort geavanceerde matrix-berekeningen in MATLAB nauwelijks moeilijker is dan scalaire berekeningen op een calculator. Met maar een paar regels typewerk vinden we de grondwaterstand in vier aquifers op 500 meter vanaf de oever, en dat terwijl de wiskundige structuur van de oplossing volledig transparant blijft.

Mazure met een onvolkomen voedende grens

De gegeneraliseerde formule van Mazure is direct bruikbaar als ter plaatse van de rivier de stijghoogten in alle aquifers gegeven zijn. Echter: meestal snijdt de zandige rivierbedding alleen de bovenste aquifer(s) aan (figuur 3). In de diepere aquifers zijn de stijghoogten dan niet zonder meer bekend, maar op grond van symmetrie-overwegingen weten we wel direct dat de gradiënt van de stijghoogten daar nul moet zijn.

Die kennis stelt ons in staat om de stijghoogte in de niet-aangesneden aquifers te berekenen.

Uit vergelijking (10) volgt dat

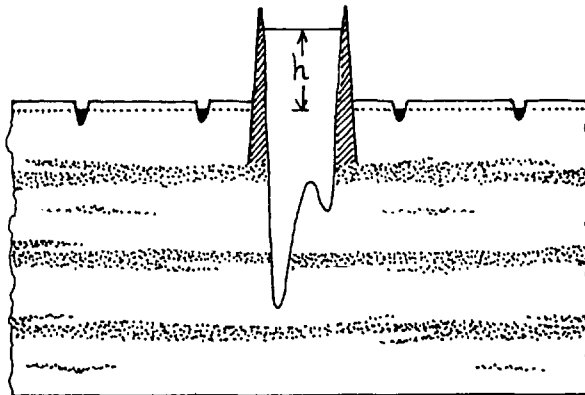
$$\frac{d\varphi}{dx} = -\sqrt{A}h \quad (11)$$

als $x = 0$. Hier staat een stelsel van n vergelijkingen met n onbekenden, maar de onbekenden staan jammer genoeg niet allemaal aan dezelfde kant. Van de vector $d\varphi/dx$ kennen we immers alleen de onderste elementen (die nul zijn) en van de vector h alleen de bovenste (die zijn gelijk aan het rivierpeil). Niettemin is dit een elementair vraagstuk, dat we graag als oefening aan de lezer overlaten. In de appendix treft u de M-file 'stijgh' aan, waarmee de stijghoogten in de niet-aangesneden lagen berekend kunnen worden. Wilt u weten hoe hij werkt? Type

```
>> help stijgh
```

dan zegt MATLAB

```
stijgh(A,p,n) is een hulpprogramma voor Mazure.  
Deze functie berekent de stijghoogten t.p.v. de rivier in de  
niet-aangesneden aquifers.  
A = systeemmatrix; zie functie sysmat  
p = rivierpeil, gerekend t.o.v. polderpeil  
n = aantal aquifers dat wel aangesneden wordt.
```



Figuur 3: Geohydrologisch schema Mazure met onvolkomen voedende grens

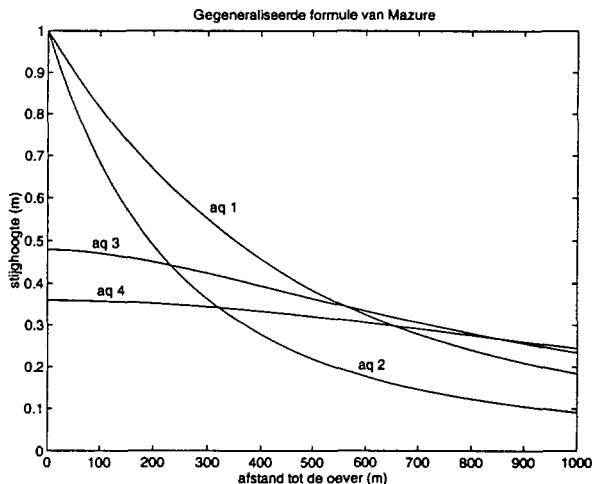
mits u de M-file geladen hebt, natuurlijk. Hier is een testvoorbeeld:

```
» h = stijgh(A,1,2);
» x = 500;
» phi = expm(-x*sqrtm(A))*h
phi =
0.2191
0.3833
0.3634
0.3202
```

We kunnen ons voorstellen dat u liever een grafiekje ziet dan een rijtje getallen. U hoeft MATLAB niet te verlaten om een grafisch pakket aan te roepen, want de standaard beschikbare grafische faciliteiten van MATLAB zijn heel acceptabel. (De meer geavanceerde grafische mogelijkheden van MATLAB zijn magnifiek, maar het kost wat tijd om ze volledig te leren benutten). De volgende instructies, bijvoorbeeld,

```
» x = 0:10:1000;
» phi = zeros(length(kD),length(x));
» for j = 1:length(x)
    phi(:,j) = expm(-x(j).*sqrtm(A))*h;
end;
» plot(x,phi)
```

leveren een kale versie op van de grafiek die als figuur 4 is weergegeven. U moet overigens wel even opletten bij het vermenigvuldigen in MATLAB. Een sterretje staat voor een normale matrixvermenigvuldiging (waarbij een matrix desnoods de afmetingen 1x1 mag hebben); daarentegen betekent `.*` dat de betrokken matrices *element voor element* vermenigvuldigd moeten worden. Zo geeft $(a_1 \ a_2 \ a_3) \cdot (b_1 \ b_2 \ b_3)$ als resultaat een vector $(a_1 b_1 \ a_2 b_2 \ a_3 b_3)$. Dat is heel praktisch, maar MATLAB kan natuurlijk geen waarschuwing afgeven als u deze twee bewerkingen per ongeluk door elkaar haalt.



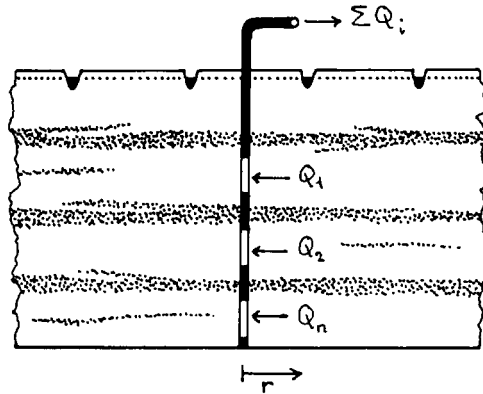
Figuur 4: Stijgheighten in vier aquifers, die gevoed worden door een rivier die alleen de bovenste twee aquifers aansnijdt.

Mafglee: een generalisatie van de formule van De Glee

Tot zover de formule van Mazure. Een andere vaderlandse klassieker is de formule van De Glee (1930). In feite is de formule van De Glee een vereenvoudigde versie van een formule die eerder werd afgeleid door Kooper (1914), zijn chef op het thans legendarische Rijksbureau voor Drinkwatervoorziening. Voor zover wij weten was Kooper de eerste om te beseffen dat de stroming in slecht waterdoorlatende lagen meestal nagenoeg verticaal gericht is. Voor de ontwikkeling van de geohydrologie heeft deze constatering een betekenis die vergelijkbaar is met de aanname van Dupuit, volgens welke de stroming van grondwater in goed doorlatende lagen vrijwel horizontaal verloopt. De Glee meende dat Koopers formule een onnauwkeurigheid bevatte—in zijn proefschrift wijdde hij daar zelfs een stelling aan—maar die claim berust op een verkeerde interpretatie: in werkelijkheid is Koopers formule juist nauwkeuriger, al is er in de praktijk weinig verschil. Hoe het ook zij, het gaat om de stationaire verlaging ten gevolge van een put in semi-spanningswater. Veel later, in 1951, heeft Huisman de formule van De Glee uitgebreid voor een systeem dat twee aquifers bevat. Hij kreeg daarbij hulp van de wiskundige Kemperman, die bij het toenmalige Mathematisch Centrum werkte. Uiteindelijk vonden Hemker en Maas onafhankelijk van elkaar in 1984 de oplossing voor een willekeurig aantal watervoerende lagen, maar de eerlijkheid gebied te zeggen dat die oplossing in Oost-Europa al eerder bekend was.



Dr.ir. G.J. de Glee was in de jaren twintig—toen hij zijn proefschrift schreef—in dienst van het Rijksbureau voor Drinkwatervoorziening, dat tot taak had om Nederland op het waterleidingnet aan te sluiten. Het bureau (later: instituut) was gedurende driekwart eeuw hét Nederlandse kenniscentrum voor grondwater, en een kweekschool voor geohydrologen. In de loop van de jaren tachtig ging het roemloos op en onder in het grotere Rijksinstituut voor de Volksgezondheid, een bolwerk van artsen dat met dit kostbare stukje kennisinfrastructuur kennelijk geen raad wist. Lang daarvoor, in 1962, sloot De Glee een mooie loopbaan af als directeur van de WAPROG, de waterleidingmaatschappij voor de provincie Groningen.



Figuur 5: Geohydrologisch schema Majgley

Figuur 5 geeft het rekenschema weer. De heersende matrix-differentiaalvergelijking is nog steeds vergelijking (6), met dien verstande dat we nu spreken over radiale stroming. De vector-randvoorwaarden zijn:

$$\begin{aligned}
 -\lim_{r \rightarrow 0} 2\pi r k D \cdot * \frac{d\varphi}{dr} &= Q \\
 \varphi(\infty) &= 0
 \end{aligned}
 \tag{12}$$

In de bovenste vergelijking is Q een kolomvector die de onttrekkingen per laag bevat. We hebben hier de handige MATLAB-operator $\cdot *$ gebruikt, ook al is dat geen algemene standaard. In het radiale geval kan de algemene oplossing van vergelijking (6) geschreven worden als

$$\varphi(r) = K_0(r\sqrt{A})c_1 + I_0(r\sqrt{A})c_2
 \tag{13}$$

U kunt dit antwoord desgewenst checken door het in te vullen in vergelijking (6). Het is weer direct in te zien dat c_2 nul moet zijn, zodat

$$\varphi(r) = K_0(r\sqrt{A})c_1
 \tag{14}$$

Om c_1 op te lossen merken we op dat ingevolge vergelijking (3)

$$\varphi(r) = K_0(r\sqrt{A})c_1
 \tag{15}$$

Gebruik makend van het eerder gepresenteerde lijstje limieten (vergelijking (4)) vinden we

$$\lim_{r \rightarrow 0} \frac{d\varphi}{dr} = -\lim_{r \rightarrow 0} \frac{1}{r} c_1
 \tag{16}$$

Substitutie in de bovenste vergelijking van (12) geeft tenslotte

$$c_1 = \frac{1}{2\pi} Q \cdot / k D c
 \tag{17}$$

De operator $/$ is de inverse bewerking van \cdot , dus $Q./kD$ is een kolomvector met elementen Q_i/kD_i . Ingevuld in vergelijking (14) vinden we nu

$$\varphi(r) = \frac{1}{2\pi} K_0(r\sqrt{A})(Q./kD) \quad (18)$$

Dit is de *gegeneraliseerde formule van De Glee*, in wederom een uiterst compacte notatie die niettemin door MATLAB goed begrepen wordt. In tegenstelling tot `expm` is de Besselfunctie K_0 niet standaard als matrixfunctie beschikbaar in MATLAB, maar MATLAB biedt ons de mogelijkheid om zelf matrixfuncties te maken. In zijn algemeenheid gaat dat als volgt:

```
funm(A, 'functienaam')
```

In plaats van `expm(A)` hadden we dus ook kunnen zeggen: `funm(A,'exp')`. MATLAB kent wel de scalaire Besselfuncties. De functie $K_0(x)$, bijvoorbeeld, heet in MATLAB `Besselk(0,x)`. Maar nu lopen we tegen een schoonheidsfoutje van MATLAB aan: `funm` accepteert geen functies met meer dan één argument, dus hij werkt niet op `Besselk(0,x)`. Daar hebben we het volgende op gevonden: we maken twee M-filetjes.

```
function f = besselk0(x)
f = besselk(0,x)
```

```
function f = K0(A)
f = funm(A,'besselk0')
```

De eerste M-file verandert de scalaire functie `besselk(0,x)` in een andere scalaire functie die nog maar één parameter heeft, maar die verder hetzelfde doet. De tweede M-file verandert deze scalaire functie in een matrixfunctie met de naam K_0 . De volgende evaluatie van vergelijking (18) zal nu waarschijnlijk geen uitleg meer behoeven:

```
>> Q = [0;0;0;1200];
>> r = 100;
>> phi = 1/(2*pi)*K0(r*sqrtm(A))*(Q./kD)
phi =
0.0557
0.1323
0.2913
1.0570
```

Ruim tien jaar terug kostte het één van de auteurs weken programmeerwerk om hetzelfde resultaat te bereiken!

Mafglee met kortsluiting tussen de aquifers

Formule (18) is direct bruikbaar als per afgetapte aquifer het debiet gegeven is. In de praktijk komt het geregeld voor dat een putfilter meerdere aquifers tegelijk aftapt, waardoor hydraulische kortsluiting ontstaat. Het is dan niet meer bij voorbaat te zeggen hoeveel water de afzonderlijke aquifers leveren, maar volgens een praktijkregel zijn in dat geval de afzonderlijke debieten evenredig met de overeenkomstige kD -waarden. Meestal (niet altijd!) is dat een heel behoorlijke schatting, maar de debieten zijn ook exact uit te rekenen. Laten we gemakshalve aannemen dat over de hele hoogte van het putfilter dezelfde afpomping optreedt. Aan de rand van de omstorting geldt

$$\varphi(r_0) = \frac{1}{2\pi} K_0(r_0 \sqrt{A})(Q./kD) \quad (19)$$

Net zoals vergelijking (11) is dit een stelsel van n vergelijkingen met n onbekenden, die niet allemaal netjes aan één kant staan: rechts kennen we alleen de debieten van de lagen die niet afgetapt worden (die zijn nul), en links kennen we alleen de verlagingen van de stijghoogte in de lagen die wel worden afgetapt. En zelfs dat niet precies: we weten alleen dat ze even groot zijn. Maar om te beginnen kunnen we ze gelijk aan 1 stellen; dan vinden we alvast de juiste verhouding tussen de debieten, die daarna eenvoudig opgeschaald kunnen worden om samen het totale debiet van de put op te leveren. In wezen is dit elementaire lineaire algebra, die we hier niet hoeven te demonstreren. U treft in de appendix een M-file aan met de naam 'debieten'. Wat doet hij?

» help debieten

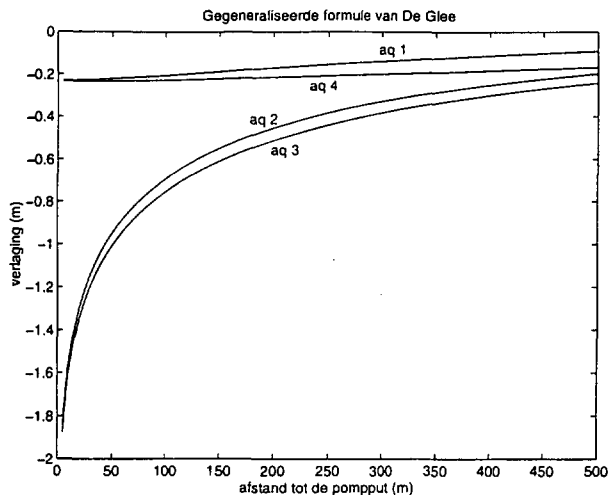
De functie `debieten(kD,c,r_p,Q_tot,laagnrs)` is een hulpprogramma voor de gegeneraliseerde formule van De Glee. Hij berekent de debietvector Q van Mafglee, voor het geval dat het putfilter over meerdere aquifers doorloopt.

`kD` = kolomvector die de kD -waarden bevat;
`c` = kolomvector die de c -waarden bevat;
`r_p` = straal van de omstorting (scalair);
`Q_tot` = totale debiet van de put (scalair);
`laagnrs` = vector met de rangnummers van de afgetapte lagen.

Hier volgt een toepassing:

```
» Q = debieten(kD,c,.2,1200,[2,3]);  
» phi = zeros(length(kD),length(r));  
» for i = 1:length(r)  
    phi(:,i) = 1/(2*pi)*K0(r(i)*sqrtm(A))*(Q./kD);  
    end  
» plot(r,-phi)
```

Figuur 6 geeft het resultaat, maar dan voorzien van bijschriften.



Figuur 6: Mafglee met kortsluiting tussen de aquifers

Misschien is het u opgevallen dat we r gewoon vanaf nul hebben laten oplopen, hoewel we weten dat de verlaging in het punt $r = 0$ oneindig groot is? Eén van de vele charmante trekjes van MATLAB is dat het programma daarmee totaal geen moeite heeft. Geen flauwekul met overflow of error, niets van die ergernissen: ergens in zijn geheugen maakt het programma de aantekening 'NaN' (not a number) of misschien 'inf', maar het rekent gewoon door, en we hoeven dus nooit meer lastige voorzieningen te treffen om singuliere punten af te vangen.

Grondwatergetijden

Het is niet helemaal verwonderlijk dat in Nederland behalve de stroming naar putten ook de voortplanting van grondwatergetijden al vroeg de aandacht trok: Steggewentz (1933), Bosch (1951), Edelman (1953), Wesseling (1959), Ernst (1962) en Van der Kamp (1973) droegen allemaal een steentje bij. In de praktijk wordt hier te lande vooral de formule van Bosch toegepast. Van Bosch' afleiding resteert ons alleen maar een stencil, dat herinnert aan een voordracht voor het Hydrologisch Colloquium. (Eertijds was het Hydrologisch Colloquium een illustre gezelschap, dat onder meer het boekje "Steady Flow of Groundwater towards Wells" uitbracht, onder de schuilnaam Anonymus. Dit werkje stond vrij lang vrij hoog op de internationale hitparade. Als jonge ingenieurs koesterden wij de wens om bij deze club te mogen horen, maar tot onze teleurstelling zijn we nooit door de ballotage heen gekomen.)

Het hydrologische schema dat ten grondslag ligt aan de formule van Bosch is hetzelfde als dat van Mazure, en de generalisatie naar een meerlagen-systeem (figuur 2) verloopt in hoofdlijnen analoog. Een verschil is dat bij Bosch de stijghoogte periodiek varieert, zodat de differentiaalvergelijking een tijdfhankelijke term krijgt. In plaats van vergelijking (6) krijgen we nu dus het stelsel

$$\nabla^2 \varphi - A\varphi = \text{diag}(\varepsilon./kD) \frac{\partial \varphi}{\partial t} \quad (20)$$

Hierin is ε een vector die de elastische bergingscoëfficiënten van de aquifers bevat. $\text{diag}(\varepsilon./kD)$ is een diagonaalmatrix; de diagonaalelementen hebben de waarden ε_i/kD_i . De samendrukbaarheid van de slecht waterdoorlatende lagen wordt door Bosch verwaarloosd. (Daar is ook wel wat voor te zeggen, maar de ruimte laat niet toe dat we daarop ingaan.)

Als we de getijbeweging van het open water schematiseren tot een zuivere sinus, dan zal de stijghoogte van het grondwater overal eveneens een zuiver sinusvormige beweging uitvoeren. De amplitudes en de fasen verschillen echter van plaats tot plaats. In zijn algemeenheid geldt dus dat

$$\varphi_i = h_i \sin \omega(t - t_i) \quad (21)$$

waarin h_i de amplitude is, en ω de hoeksnelheid. Met t_i geven we de vertraging van het getij aan. h_i en t_i zijn functies van x ! Het is bekend dat de amplitude en de fase van de functie

$$\varphi_i = h_i \exp i \omega(t - t_i) \quad (22)$$

zich op precies dezelfde manier gedragen als de amplitude en de fase van (21). Omdat (22) wiskundige voordelen heeft, geven we aan die uitdrukking de voorkeur. We gaan zelfs nog een stapje verder, en schrijven

$$\varphi_i = \bar{\varphi}_i \exp i \omega t \quad (23)$$



Ir. H. Bosch trad direct na zijn studie, in 1947, in dienst van de Haagse Duinwaterleiding DWL (thans DZH). Tijdens de wederopbouw, in de vroege jaren vijftig, legde dit bedrijf een buisleiding aan van Bergambacht naar Scheveningen, om de slinkende voorraad duinwater kunstmatig aan te vullen met rivierwater. De getijdenformule werd afgeleid als voorbereiding op de bouw van het innamewerk aan de Lek. Van 1961 tot 1984 was de heer Bosch directeur van de DWL.

Uit een vergelijking van (22) en (23) blijkt dat

$$\begin{aligned} h_i &= \text{abs} \bar{\varphi}_i \\ t_i &= \frac{1}{\omega} \text{angle} \bar{\varphi}_i \end{aligned} \quad (24)$$

Als we de gestreepte φ kennen, kennen we dus ook de amplitude h_i en de vertraging t_i , waarom het ons te doen is. We zullen daarom eerst de gestreepte φ oplossen. Daartoe substitueren we uitdrukking (23) in het stelsel (20). Na delen door $\exp(i\omega t)$ vinden we

$$\nabla^2 \bar{\varphi} - A \bar{\varphi} - i\omega \text{diag}(\varepsilon./kD) \bar{\varphi} \quad (25)$$

of

$$\nabla^2 \bar{\varphi} - [A\varphi + i\omega \text{diag}(\varepsilon./kD)] \bar{\varphi} = 0 \quad (26)$$

of

$$\nabla^2 \bar{\varphi} - B \bar{\varphi} = 0 \quad (27)$$

Wiskundig gesproken is deze uitdrukking identiek aan vergelijking (6). De randvoorwaarden zijn

$$\begin{aligned} \bar{\varphi}(0) &= h_0 \\ \bar{\varphi}(\infty) &= 0 \end{aligned} \quad (28)$$

waarin h_0 een vector is die de amplitudes op $x = 0$ bevat. (De elementen van h_0 mogen desgewenst complexe getallen zijn, wat zou betekenen dat op $x = 0$ de getijden niet allemaal in fase lopen). De voorwaarden (28) zijn wiskundig identiek aan de voorwaarden (8). Het gehele probleem is dus wiskundig identiek aan het gegeneraliseerde probleem van Mazure, en we zien direct in dat

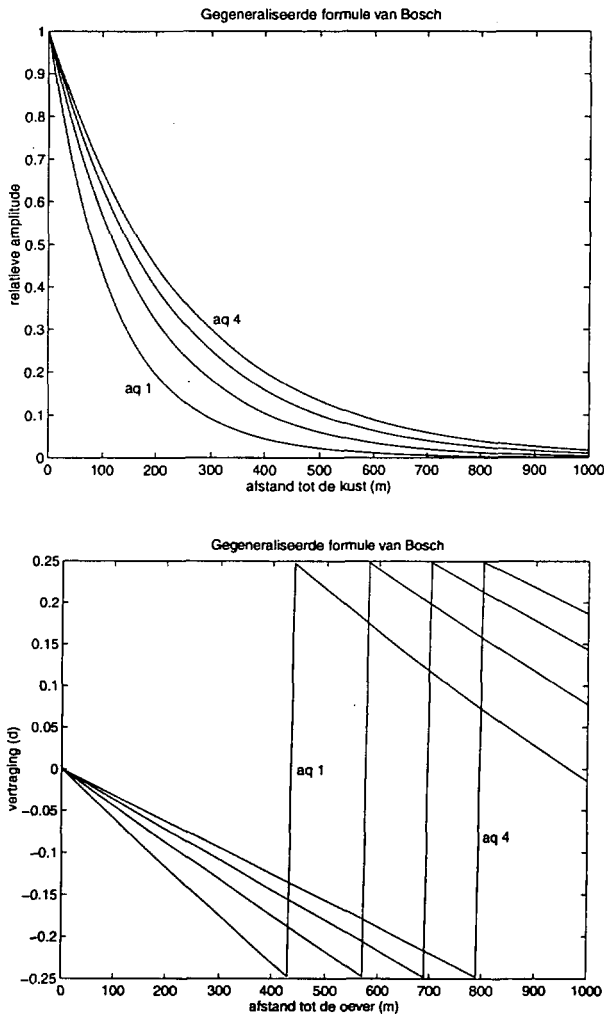
$\bar{\varphi}(x) = \exp(-x\sqrt{B})h_0 \quad (29)$

Hierin is

$$B = A + i\omega \text{diag}(\varepsilon./kD) \quad (30)$$

en A is de systeemmatrix. We aarzelen een beetje om uitdrukking (29) aan te prijzen als de gegeneraliseerde formule van Bosch (Mafbosch), want alle getijformules hebben na generalisatie deze vorm. De verschillende auteurs verschillen alleen van opvatting over de vraag hoe de matrix B gevuld moet worden.

Het wordt misschien eentonig, maar vergelijking (29) vergt alweer slechts een paar regels in MATLAB. Met twee regels extra zijn ook $h(x)$ en $t(x)$ te berekenen, door toepassing van vergelijking (24). (MATLAB kent de functies `abs` en `angle`). We presenteren hier alleen een plaatje (figuur 7), dat berekend is voor het tweemaal daagse getij. De bijbehorende MATLAB-instructies laten we over aan de fantasie van de lezer, om zelf door te gaan met een leuker en realistischer geval, namelijk het geval waarin het getijdewater ondiep is (figuur 8).



Figuur 7: Damping en vertraging van het grondwatergetij volgens de gegeneraliseerde formule van Bosch. De sprong in de vertraging is het gevolg van het feit dat de functie 'angle' meerwaardig is

Grondwatergetijden nabij een ondiepe zee

Dit geval leidt tot een opmerkelijk resultaat. In het binnenland is vergelijking (20) onverminderd van toepassing:

$$\nabla^2 \varphi - A\varphi = \text{diag}(\varepsilon./kD) \frac{\partial \varphi}{\partial t} \quad (31)$$

maar buitendijks geldt

$$\nabla^2 \varphi - A(\varphi - h) = \text{diag}(\varepsilon./kD) \frac{\partial \varphi}{\partial t} - \text{diag}(\varepsilon./kD) \frac{\partial h}{\partial t} \quad (32)$$

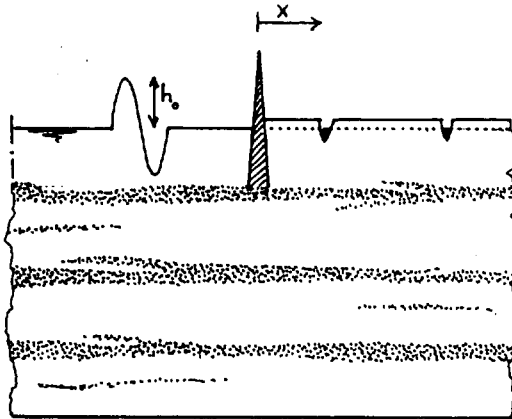
waarin h een vector is waarvan de elementen gelijk zijn aan het op en neer bewegende zeeniveau. Deze vector verschijnt in het linkerlid, omdat in het buitendijkse gebied het zeeniveau het referentiepeil vormt. Hij verschijnt ook in de tweede term van het rechterlid. We hebben het gevoel dat deze term een verklaring behoeft, want voor zover wij weten komt hij in geen enkel grondwatermodel voor. Toch hoort hij erin! De term zegt dat de elastische uitlevering niet alleen afhangt van variaties in de stijghoogte, maar ook van de samendrukking van de pakketten door een in de tijd variërende bovenbelasting. (Voor een uitgebreidere discussie verwijzen we naar Maas en De Lange (1986, 1987).)

Omdat in het buitendijkse gebied de tweede afgeleide van h naar x nul is, kan vergelijking (32) alternatief geschreven worden als

$$\nabla^2(\varphi - h) - A(\varphi - h) = \text{diag}(\varepsilon./kD) \frac{\partial(\varphi - h)}{\partial t} \quad (33)$$

Deze vergelijking is in feite toepasbaar op het gehele gebied (binnen- en buitendijks). Hij kan in de plaats van vergelijkingen (31) en (32) gebruikt worden, mits we afspreken dat

$$h = a \exp i \omega t \quad (x < 0) \quad (34)$$



Figuur 8: Ondiepe zee

met

$$\begin{aligned} a &= h_0 & (x < 0) \\ a &= 0 & (x > 0) \end{aligned} \quad (35)$$

Hierin is h_0 een vector die als elementen de amplitude van het getij in de zee bevat. We kunnen nu a opgebouwd denken uit een symmetrisch deel a_1

$$a_1 = \frac{1}{2} h_0 \quad (-\infty < x < \infty) \quad (36)$$

en een antisymmetrisch deel a_2

$$\begin{aligned} a_2 &= \frac{1}{2} h_0 & (-\infty < x < 0) \\ a_2 &= -\frac{1}{2} h_0 & (0 < x < \infty) \end{aligned} \quad (37)$$

(zie ook figuur 9). Dienovereenkomstig is φ te splitsen in φ_1 (de reactie van de grondwaterstijghoogte op a_1) en φ_2 (de reactie op a_2). Het is direct in te zien dat

$$\varphi_1(x) = \frac{1}{2} h_0 \exp i \omega t \quad (38)$$

want a_1 geeft geen aanleiding tot horizontale stroming. Ter plaatse van de kust geldt dus ook

$$\varphi_1(0) = \frac{1}{2} h_0 \exp i \omega t \quad (39)$$

Wegens antisymmetrie is ter plaatse van de kust

$$\varphi_2(0) = 0 \quad (40)$$

zodat

$$\varphi(0) = \varphi_1(0) + \varphi_2(0) = \frac{1}{2} h_0 \exp i \omega t \quad (41)$$

Het grondwatergetij ter plaatse van de kust loopt dus in fase met het open water, terwijl daar in alle aquifers de amplitude juist de helft is van de amplitude in de zee, ongeacht de bodemconstanten! Als we deze $\varphi(0)$ opvatten als randvoorwaarde voor het domein $x > 0$, vinden we dus dezelfde uitkomst als in de vorige paragraaf (vergelijking 29), maar met de helft van de amplitude:

$$\bar{\varphi}(x) = \frac{1}{2} \exp(-x\sqrt{B}) h_0 \quad (42)$$

Had u dat gedacht?

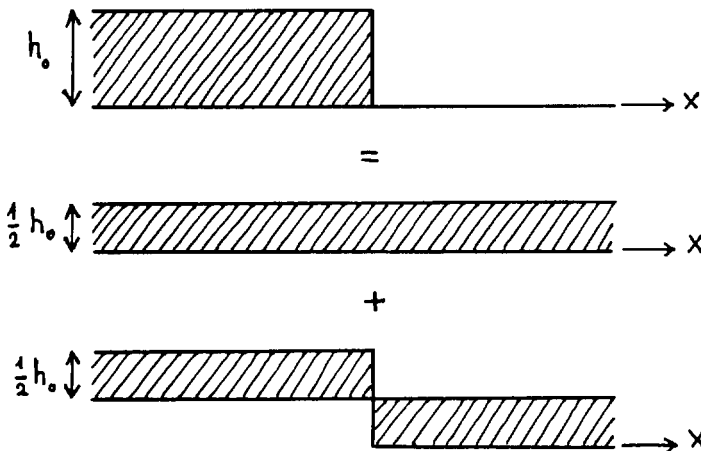
Slot

We hopen dat we met dit artikel het nut van matrixfuncties – vooral in combinatie met het gebruik van MATLAB – duidelijk hebben kunnen maken. Hoewel we maar een fractie van de mogelijkheden van MATLAB hebben aangesproken, denken we dat het genoeg was om ons enthousiasme voor dit schitterende gereedschap op u over te dragen. Wat leven we toch in een heerlijke tijd! Naar goed Hollandse maatstaven is MATLAB wel een duur programma (versie 5 kost ca. f 5000,-) maar u kunt eerst de kat uit de boom kijken met de studentenversie. Bij Broese Kemink aan de Oude Gracht in Utrecht kost die zowel voor Windows 95 als voor de Macintosh maar f 120,30. (U moet even aandringen: hij staat er nog niet op de computer, maar al wel op de plank.) De studentenversie staat op CD-ROM, hij vraagt minimaal 8 Mbyte RAM en 25 Mbyte schijfruimte (inclusief help-files 50 Mbyte). Deze versie heeft de volledige kracht en functionaliteit van MATLAB 5 en bevat zelfs een aantal toolboxes die niet standaard meegeleverd worden met de professionele versie. Alleen is de omvang van de bewerkbare matrices beperkt tot 16.384 elementen, wat neerkomt op een array van 128x128. Voor heel wat toepassingen is dat ruim voldoende.

Als alles goed gaat kunt u binnenkort de M-files uit dit artikel omlaagladen vanaf de nieuwe weblocatie van de NHV.

Dankwoord

De foto van professor Mazure werd ons ter beschikking gesteld door één van zijn zoons, de heer R. Mazure te Bergen. De afbeelding van de heer De Glee komt uit het foto-archief van de heer K.D. Venhuizen te Rolde, die net als De Glee directeur is geweest van de WAPROG. Wij kwamen met hem in contact door tussenkomst van de bibliothecaris van dit bedrijf. De heer Venhuizen introduceerde ons op zijn beurt bij de heer H. Bosch, die in Voorburg woont. Vanaf deze plaats danken wij hen allen voor hun medewerking aan dit artikel.



Figuur 9: Splitsing van de amplitude van het getij in een symmetrisch en een antisymmetrisch deel.

Literatuur

- Bosch, H. (1951)** Geohydrologisch onderzoek te Bergambacht; Voordracht voor het Hydrologisch Colloquium.
- Edelman, T. (1953)** Voortplanting van eb en vloed in het grondwater; Voordracht voor het Hydrologisch Colloquium.
- Ernst, L.F. (1962)** Grondwaterstroming in de verzadigde zone en hun berekening bij aanwezigheid van horizontale evenwijdige open leidingen; proefschrift, PUDOC, Wageningen, 189 pag.
- Glee, G.J. de (1930)** Over grondwaterstromingen bij wateronttrekking door middel van putten; proefschrift, J. Waltman jr. Delft, 175 pag.
- Hemker, C.J. (1984)** Steady groundwater flow in leaky multiple-aquifer systems; in: *Journal of Hydrology*, nr 72, pag 355–374.
- Huisman, L. en J. Kemperman (1951)** Bemaling van Spanningsgrondwater; in: *De Ingenieur*, jrg 62, nr 13, pag 29–35.
- Kamp, S.J.P. van der (1973)** Periodic Flow of Groundwater; proefschrift Vrije Universiteit Amsterdam, Rodopi N.V., Amsterdam, 121 pag.
- Kooper, J. (1914)** Verslag omtrent de werkzaamheden van de Centrale Commissie voor de Drinkwatervoorziening en van het Rijksbureau voor Drinkwatervoorziening in het jaar 1913; 's-Gravenhage, ter Algemeene Landsdrukkerij.
- Maas, C. (1984)** De toepassing van matrixfuncties in de geohydrologie; voordracht ter gelegenheid van de studiedag van de Hydrologische Kring op 4 oktober 1984, Provincie Zeeland, Provinciale Waterstaat, Onderafdeling Waterbeheer, rapport 84-02.
- Maas, C. (1986)** The use of matrix differential calculus in problems of multiple aquifer flow; in: *Journal of Hydrology*, nr 88, pag 43–67.
- Maas, C. en W.J. de Lange (1986)** Over het voorlopen van het grondwatergetij op de getijbeweging van de Hollandse IJssel; in: *H₂O* nr 2/86, pag 24–28.
- Maas, C. en W.J. de Lange (1987)** On the negative phase shift of groundwater tides near shallow tidal rivers – the Gouderak anomaly; in: *Journal of Hydrology*, nr 92, pag 333–349.
- Mazure, J.P. (1932)** Invloed van een weinig doorlatende afdekkende laag op de kwel onder een dijk; in: *De Ingenieur*, nr 47, B41–B43.
- Olsthoorn, T.N. (1997)** Generalized analytical solutions to steady and transient semi-confined multi-aquifers; Amsterdam Water Supply, R&D Department Hydrology, 34 pag.
- Schaaf, S. van der (1995)** Snelle oudjes: Toepassing van Mazure's oplossingen voor eerste effectschattingen van waterhuishoudkundige veranderingen; in: *H₂O*, jrg 28, nr 25, pag 750–753.
- Stegewentz, J.H. (1933)** De invloed van de getijbeweging van zeeën en getijrivieren op de stijghoogte van grondwater; proefschrift, Delft, W.D. Meinema.
- Verruijt, A. (1970)** Theory of Groundwater Flow; Macmillan, 190 pag.

Appendix

```
function h = stijgh(A,p,n)
```

```
% stijgh(A,p,n) is een hulpfunctie voor Mafzure. Deze functie berekent de  
% stijghoogten t.p.v. de rivier in de diepere aquifers die niet door de  
% zandige rivierbedding worden aangesneden.
```

```
% A      = systeemmatrix, zie de functie sysmat;  
% p      = rivierpeil, gerekend t.o.v. het polderpeil;  
% n      = het aantal aquifers dat door de rivier direct wordt gevoed.
```

```
A      = sqrtm(A);           % systeemmatrix  
m      = max(size(A));      % aantal aquifers  
B3     = A(n+1:m,1:n);     % submatrix van A  
B4     = A(n+1:m,n+1:m);   % submatrix van A  
h_bekend = p*ones(n,1);    % stijghoogten in de aangesneden aquifers  
h_onbekend = -inv(B4)*(B3*h_bekend); % stijghoogten in de overige aquifers  
h      = [h_bekend;h_onbekend]; % stijghoogtevector t.p.v. de rivier
```

```
function Q = debieten(kD,c,r_p,Q_tot,laagnrs)
```

```
% De functie debieten(kD,c,r_p,Q_tot,laagnrs) is een hulpprogramma voor de  
% gegeneraliseerde formule van De Glee. Hij berekent de debietvector Q van  
% Mafglee, voor het geval dat het putfilter over meerdere aquifers  
% doorloopt.
```

```
% kD     = kolomvector die de kD-waarden bevat;  
% c      = kolomvector die de c-waarden bevat;  
% r_p    = straal van de omstorting (scalair);  
% Q_tot  = totale debiet van de put (scalair);  
% laagnrs = vector met de rangnummers van de afgetapte lagen.
```

```
A = sysmat(kD,c);           % systeemmatrix  
B = 1/(2*pi)*K0(r_p*sqrtm(A)); % zie rechter lid van vlg (18)  
for i = 1:length(laagnrs)  
    for j = 1:length(laagnrs)  
        C(i,j) = B(laagnrs(i),laagnrs(j)); % submatrix van B  
    end  
end  
enen = ones(length(laagnrs),1); % stel verlagingen in putfilter gelijk aan 1  
QoverkD = inv(C)*enen; % ongeschaalde debieten./kD  
Q = zeros(length(kD),i);  
for i = 1:length(laagnrs)  
    Q(laagnrs(i)) = QoverkD(i)*kD(laagnrs(i)); % ongeschaalde debieten  
end  
Q = Q/sum(Q)*Q_tot; % geschaalde debieten
```