

---

# HatsikD

---

Eerst een punt van orde: doordat ik deze rubriek al een poosje niet meer onderteken is het sommige lezers niet meer duidelijk dat ze hun inzendingen kunnen richten aan kees.maas@kiwa.nl. Iedereen die een vuistregel kent, of een compact presenteerbare wetenswaardigheid die met ons vak te maken heeft, is welkom. Verder haast ik me om te stellen dat ik deze HatsikD niet zelf gemaakt heb. Het is een inzending van Theo Olsthoorn. Ter aanvulling vooraf merk ik nog op dat de studentenversie van Matlab niet meer te koop is in de vorm die we destijds in een artikel in Stromingen aanprezen. Tegenwoordig is een collegekaart vereist, en de aanschaf omvat tevens Simulink en een symbolisch wiskundepakket.

## Niet-stationaire putonttrekking in n-lagen in Matlab

Sinds hun artikelen in Journal of Hydrology, mogen Kees Maas en Kick Hemker met recht de godfathers worden genoemd van de elegante meerlaagsoplossingen voor de stroming naar putten. Dat deze oplossingen in matrixvorm bijzonder eenvoudig zijn te implementeren in een modern rekenpakket als Matlab is in dit blad reeds getoond (Maas en Olsthoorn, 1997). De Glee, Mazure en een aantal oplossingen met golvende randvoorwaarden in een willekeurig aantal lagen, bleken te kunnen worden geprogrammeerd in slechts enkele regels Matlab en waren gemakkelijk te doen in de goedkope studentenversie van dit pakket (ca 50 Euro). De matrix-aanpak opende een wereld van mogelijkheden. Hemker en Maas (1987) hebben de stroming naar een put in een willekeurig aantal lagen niet alleen stationair, maar

ook niet stationair opgelost. Hierbij wordt eerst via een Laplace transformatie de tijdsafhankelijkheid geëlimineerd, waarmee het probleem wordt teruggebracht tot een "stationair meerlagenprobleem", zij het dat daarin de Laplace-parameter voorkomt en de uitkomst is gegeven in de Laplace-getransformeerde stijghoogte. Verder is er geen verschil, zodat de meerlagen oplossing meteen bekend is. Resterende de terugtransformatie vanuit de Laplace-ruimte naar de gewone stijghoogte. Beide heren hebben uitgezocht dat hiervoor een groot aantal verschillende methoden bestaat, maar dat die van Stehfest (1970) uitstekend voldoet. De aanpak staat precies beschreven in het artikel van Maas en Hemker (1987) en in het proefschrift van Hemker (2001, p57). Ik heb dit nu in Matlab geprogrammeerd en bied dit u bij deze aan, zodat u voortaan zelf gemakkelijk niet stationaire pompproeven in een willekeurig aantal lagen kan uitwerken of via superpositie pompstations kan doorrekenen met pompfilters en onttrekkingen die willekeurig ruimtelijk zijn verdeeld. Het programmaatje is zo krachtig en compact, dat het met recht een "HatsikD" mag worden genoemd. De appendix geeft het programmaatje en vervolgens de test van Hemker en Maas (1987) met de uitkomst daarvan. Deze is exact gelijk aan die in het genoemde artikel.

Opm. 1. Bij gebruik van vectoren voor  $r$  en  $t$  worden de waarden voor alle lagen en combinaties van  $r$  en  $t$  in een keer uitgerekend, wat een 3d matrix oplevert, lagen verticaal,  $t$  horizontaal en  $r$  in de derde dimensie.

Opm 2. Het gemodelleerde lagensysteem heeft altijd een scheidende laag aan top en aan de basis met daarboven resp. daaronder een op nul gefixeerde stijghoogte. Zowel de scheidende als de watervoerende lagen hebben elk een bergingscoëfficiënt. Door instelling van weerstanden en bergingscoëfficiënten zijn alle varianten

mogelijk.

Opm 3. De inwendige lus vangt het geval met slechts 1 watervoerende laag op, zodat het programma voor elk willekeurig aantal lagen werkt.

## Referenties

**Hemker, C.J. (2000)** Groundwater flow in layered aquifer systems. PhD, Vrije Universiteit Amsterdam.

**Hemker, C.J. & C. Maas (1987)** Unsteady flow to wells in layered and fissured aquifer systems. *Journal of Hydrology*, 90, 231-249.

**Hemker, C.J. (1985)** Transient well flow in leaky multiple-aquifer systems. *Journal of Hydrology*, 81, 111-126.

**Hemker, C.J. (1984)** Steady flow in leaky

multiple-aquifer systems. *Journal of Hydrology*, 72, 355-374.

**Maas, C. & T.N.Olsthorn (1997)** Snelle oudjes gaan Matlab. *Stromingen*, 3, 21-42

**Maas, C. (1984)** De toepassing van matrixfuncties in de geohydrologie. Voordracht hydrologische kring 4 Okt. 1984; Rapport Prov. Zeeland, Prov. Waterstaat, Onderafd. Waterbeh. Rap. 84-02.

**Maas, C. (1986)** The use of matrix differential calculus in problems of multiple aquifer flow; in *Journal of Hydrology*, 88, 43-67.

**Stehfest, H. (1970)** Numerical inversion of Laplace transforms. *Commun. ACM*, 13: 47-49.

---

## Appendix

```
function drawdown=hantushn(Q,r,t,St,c,Sf,kD,N)
% drawdown=hantushn(Q,r,t,St,c,Sf,kD,N)
% multi-layer transient drawdown by a well extraction from arbitrary aquifers
% implementation of Maas & Hemker(1987), see PhD of Hemker, 2000, p54
% Q column vector of extraction
% r vector of distances to well centre
% t vector of times [T]
% St Storage of aquitard bed [-]
% c aquitard resistance [T]
% Sf aquifer storage coefficient [-]
% kD aquifer transmissivity [L2/T]
% N Stehfest points [-], optional, default is 10
% TO 10-4-2003

if nargin<8,
    N=10; % number of Stehfest points, use 10 as default if omitted
end

r=r(:); t=t(:)'; St=St(:); c=c(:); Sf=Sf(:); kD=kD(:); % put vectors in column or row

% ===== Compute Stehfest weights v =====
v=zeros(N,1);
for i=1:N
```

```

dum=0;
for k=floor((i+1)/2):min(i,N/2)
    dum=dum+k^(N/2)*factorial(2*k)/(factorial(N/2-k)*factorial(k)...
        *factorial(k-1)*factorial(i-k)*factorial(2*k-i));
end
v(i)=(-1)^(i+N/2)*dum;
end

drawdown=zeros(length(kD),length(t),length(r)); % layers vert., time hor., r 3rd dim
for ir=1:length(r) % for all r
    for it=1:length(t) % for all t
        s=zeros(size(kD));
        for iStehfest=1:N % for all Stehfest points
            p=iStehfest*log(2)/t(it); % p is Laplace operator
            d=p*Sf./kD;
            b=sqrt(p*St.*c);
            if length(kD)==1
                eii=(b(1).*coth(b(1)))./(c(1).*kD);
                eij=(b(2).*coth(b(2)))./(c(2).*kD);
                A=eii+eij+d;
            else
                eii=(b(1:end-1).*coth(b(1:end-1)))./(c(1:end-1).*kD);
                eij=(b(2:end ).*coth(b(2:end )))./(c(2:end ).*kD);
                fii=b(2:end-1)./(c(2:end-1).*kD(2:end ).*sinh(b(2:end-1)));
                fij=b(2:end-1)./(c(2:end-1).*kD(1:end-1).*sinh(b(2:end-1)));
                A=diag(eii+eij+d)-diag(fii,-1)-diag(fij,+1); % sys. Mat., Laplace
            end
        end
        s=s+v(iStehfest)/(2*pi*p)*K0(r(ir)*sqrtm(A))*diag(1./kD)*Q; % back trfrm.
    end
    drawdown(:,it,ir)=s*log(2)/t(it); % finally the value in real space
end
end

function Y=K0(A)
% matrix bessel K0 function
Y=funm(A,'besselk0');

function Y=besselk0(A)
Y=besselk(0,A)

Om de functie te testen genereren we de invoer van Hemker en Maas (1987)

r=10;
t=[1;2;5]*logspace(-5,0,6); t=t(:);

% three aquitards with zero drawdown at top and bottom
St =[0.0016 0.0016 0.0016]';

```

```

c =[ 100 100 100 ]';

% two aquifers
Sf =[ 1e-4 1e-4 ]';
kDn=[ 100 100 ]';
Qn =[ 0 4*pi*100 ]';

out=t;
for N=[8 10 12] %Compute HantushN for three values of stehfest's N
    out=[out hantushn(Qn,r,t,St,c,Sf,kDn,N) '];
end

fprintf('          -----N=8-----          -----N=10-----          -----N=12-----\n');
fprintf('          t          s(1)          s(2)          s(1)          s(2)          s(1)          s(2)\n');
display(out); % display the result table, with t in first column

% The Result below exactly equals those of Hemker & Maas (1987)
          -----N=8-----          -----N=10-----          -----N=12-----
          t          s(1)          s(2)          s(1)          s(2)          s(1)          s(2)
0.0000          0          0.0164          0          0.0165          0          0.0165
0.0000          0          0.0962          0          0.0957          0          0.0956
0.0000          0          0.3591          0          0.3590          0          0.3591
0.0001 -0.0000          0.6584          0.0000          0.6585          -0.0000          0.6586
0.0002 -0.0000          1.0043          0.0000          1.0044          -0.0000          1.0045
0.0005 -0.0000          1.4871          0.0000          1.4872          -0.0000          1.4872
0.0010 -0.0000          1.8540          0.0000          1.8541          0.0000          1.8541
0.0020          0.0000          2.2167          -0.0000          2.2168          -0.0000          2.2168
0.0050          0.0000          2.6890          0.0000          2.6891          0.0000          2.6891
0.0100          0.0027          3.0417          0.0030          3.0419          0.0030          3.0419
0.0200          0.0361          3.3921          0.0358          3.3921          0.0357          3.3921
0.0500          0.1824          3.8383          0.1822          3.8382          0.1822          3.8383
0.1000          0.3403          4.0990          0.3404          4.0997          0.3404          4.0998
0.2000          0.4674          4.2486          0.4679          4.2488          0.4681          4.2487
0.5000          0.5315          4.3131          0.5315          4.3130          0.5314          4.3129
1.0000          0.5365          4.3180          0.5362          4.3178          0.5362          4.3177
2.0000          0.5363          4.3178          0.5362          4.3178          0.5363          4.3178
5.0000          0.5363          4.3178          0.5363          4.3178          0.5363          4.3178

```