Partial differential equations: Groundwater flow

3 Two-dimensional discretisation and groundwater flow

3.1 Groundwater flow

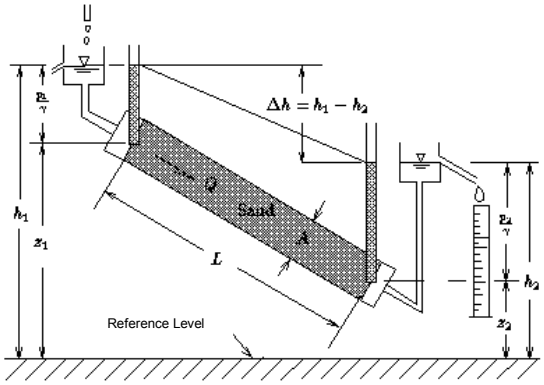
In 1856, Henry Darcy, the Engineer of the town of Dijon, in France, investigated the flow of water in a vertical, saturated, homogeneous sand filter (= column), in connection with the city's fountains. From his experiments, varying the length and diameter of the column, the porous material in it, and the water levels in inlet and outlet reservoirs, he concluded that the rate of flow (the volume of water passing per unit of time), Q, through a sand column of length L and constant cross-sectional area, A, is:

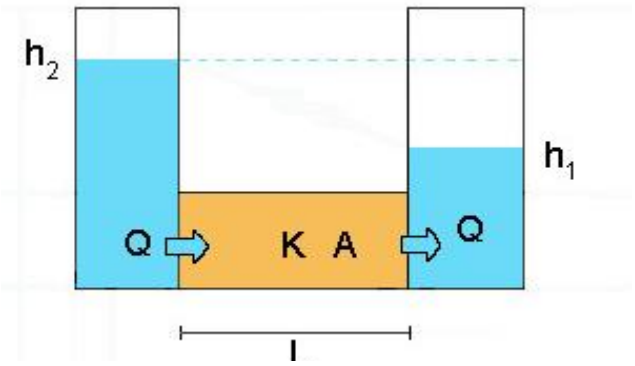
- proportional to the cross-sectional area, A, of the column,
- proportional to the difference in water level elevations, h1 and h2, in the inflow and outflow reservoirs of the column, respectively, and
- inversely proportional to the column length, L.

When combined, these conclusions give the famous Darcy's formula, or Darcy's law:

$$Q = -KA\frac{h_1 - h_2}{L} \qquad (3.1)$$

where Q is volume flux of water (m^3 day^{-1}), A is the cross-sectional area (*doorstromend oppervlak*, m^2), K is a coefficient of proportionality called hydraulic conductivity (m day^{-1}). The elevations h1 and h2 are the hydraulic head with respect to some common reference datum level.





Darcy's law is nowadays the basis of many hydrological models. Because water can flow in three dimensions, Darcy's law should also be solved in three dimensions. Fortunately, Hooghoudt (1940) has shown that when ditches are far apart and the water level is relative low, we may assume only horizontal flow. To calculate this analytically, other assumptions were steady state rainfall, vertical flow from the surface to the water table, no soil evaporation of plant transpiration, parallel ditches and a subsurface impermeable layer.

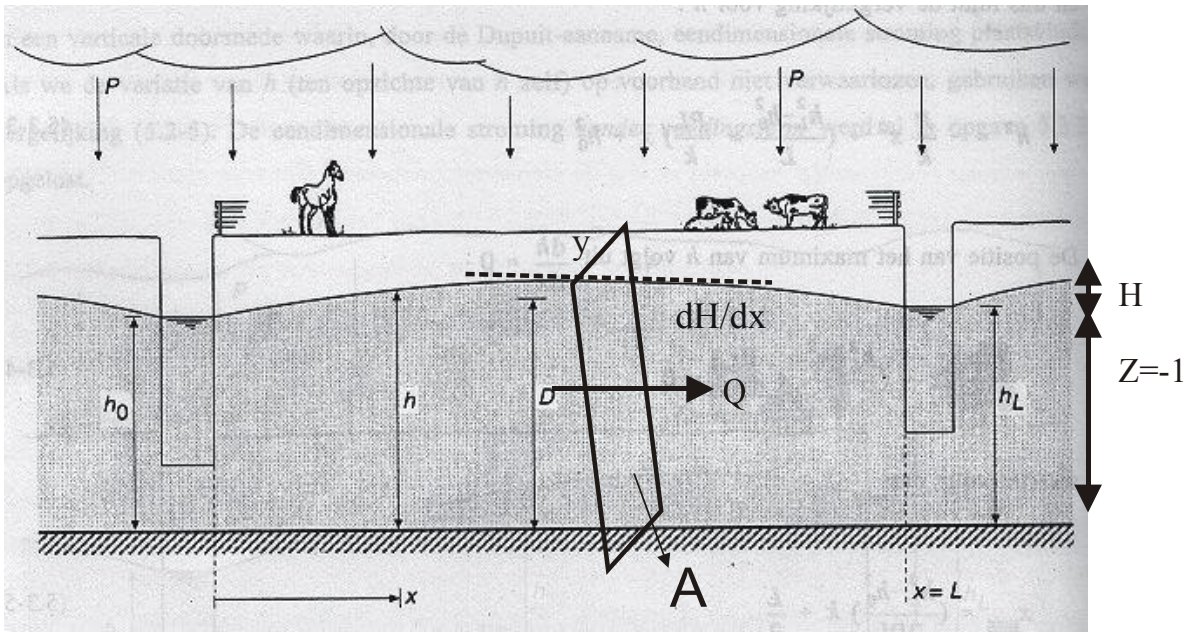


Figure 3.1: Scheme of Hooghoudt, in which A is the cross-sectional area, Q is the volume flux of water, dH/dx is the slope of the water table and y is the distance in the y-direction. In this example, reference height is at the top of the water level in the ditches and the impermeable layer is 1 meter below this reference height; P = precipitation (assumed constant at the whole field) in m day^{-1}; h_0 = water level in the ditch in comparison to reference level (impermeable soil layer); L = Length of parcel. D = Thickness of the aquifer in meters. In this case, D is approximately equal to h_0; x = horizontal position in the parcel (see illustration).

A disadvantage of the analytical form of Hooghoudt is that the model is based on steady state flow, meaning that there is no change in water storage, just like in the experiment of Darcy. If we assume steady-state flow, then the continuity equation is zero:

$$\frac{\partial h}{\partial t} = 0 \qquad (3.2)$$

in which *h* is the water storage.

However, in many environmental problems, we also want to calculate non-steady state flow, meaning that the water storage can change in time. For instance, if we would like to model soil evaporation or plant transpiration during periods without rainfall, the storage capacity will decrease. Therefore, we will solve the Hooghoudt model in a numerical way with a discrete approach (see also Chapter 2).

From our horizontal point of view, we can describe the water flow equation law in two directions:

$$Q = -KA\left(\frac{dH}{dx} + \frac{dH}{dy}\right) \text{ which can be split into}$$

$$Q_x = -KA\frac{dH}{dx} \text{ and } Q_y = -KA\frac{dH}{dy}$$

Before we continue, it is important that we use a standard grid discretization. As shown in Figure 3.2, having a horizontal view, the numbering of the rate (FlowX, FlowY) and state (H) variables is important. In Figure 3.2, we focus on only one grid cell (i,j).

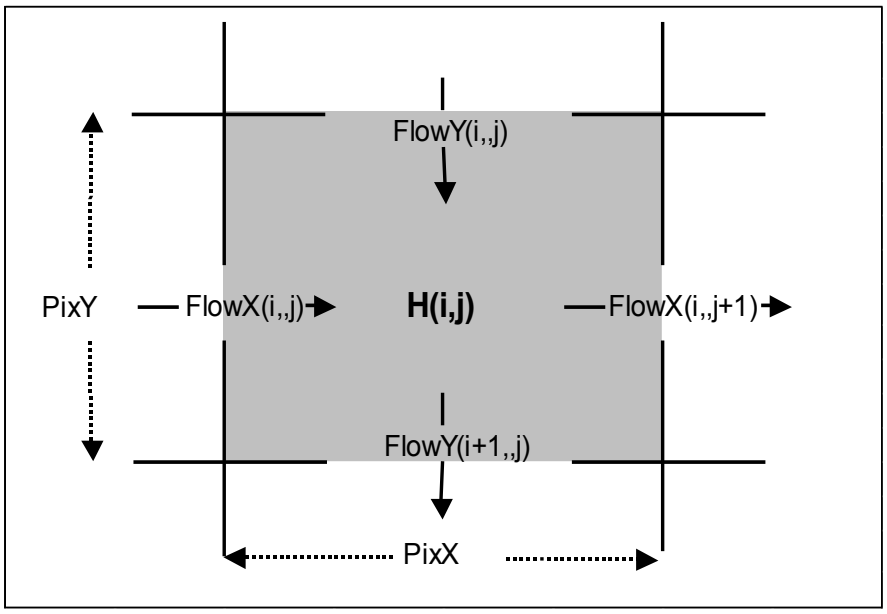


Figure 3.2: Standard grid discretization in two directions. H(i,j) means the hydraulic head of grid cell (i,j). Four different flows can come in or go out this grid cell.

As a next step, we can calculate the water flow in time. It is very important to use a standard order in which statements are being processed:
(i) calculation of all flows in and out of one grid cell;
(ii) calculation of the net balance of water flow within one grid cell;
(iii) integration to calculate the new amount of water in one grid cell at time $t = t+\Delta t$;
(iv) time update.

(i) Flow calculation

In simulation models, the boundaries of the study area have specified conditions. These conditions will not change during the dynamic simulation. In the Hooghoudt example, we assume specific boundary conditions (see more in section 3.1.1). The flows of all other cells can be calculated with two nested for-loops. As standard grid discretization we use the Appendix III on Blackboard: number grid cells: 2-dimensional discretization. Try to understand all fluxes and states.

If we assume a raster containing (NPixY×NPixX) cells, with cell dimensions PixY (m) and PixX (m), and if we only calculate the flow in x-direction:

```
for i=1 to NPixY do,
    for j=2 to NPixX do,
        KDX(i,j) =  K * PixY* ((H(i,j-1)-DemClay(i,j-1)) + (H(i,j)-DemClay(i,j)))/2;
        FlowX(i,j) = -KDX(i,j) * (H(i, j)-H(i,j-1))/PixX;
    end;
end;
```

Remember that the indices are (NPixY,NPixX) in MATLAB, so i is in the y-direction and j is the x-direction.

We use j=2 to NPixX because the border fluxes (j=1 and j=NPixX+1) of the grid has specific boundary conditions (see 3.1.1). As you see, with NPixX grid cells we have NPixX+1 fluxes.

First, the resistance of the total grid cell is calculated (KDX(i,j)), which is the resistance (K) times the cross-sectional area between two cells (PixY times the average height of the water column). Hereafter, the water flow from one cell to the next can be calculated, which is the difference in hydraulic head divided by the path length of the flow, PixX.

With MATLAB we can write this loop in a more efficient way, in which we do not need for-loops:

```
KDX = K * (0.5*(H(:,1:NPixX-1) + H(:,2:NPixX)) - DemClay) * PixY;

FlowX(:,2:NPixX) = -1 * KDX .* (H(:,2:NPixX)-H(:,1:NPixX-1)) / PixX;
```

in which

KDX= k * A = k * pixY * average height of water column [m^3 day^{-1}]
H = Hydraulic head (m) with respect to reference height
DemClay = height of impermeable layer, which is constant
FlowX = water flow = k*A* dH/dx [m^3 d^{-1}];
: means the total array (1:NPixY)

and the same can be done for the y-direction.

(ii) Net flow

After all flows are calculated, the net flow for one grid cell can be calculated:

```
NetFlow(1:NPixY,2:NPixX-1) = FlowX(1:NPixY,2:NPixX-1) - FlowX(1:NPixY,3:NPixX) + ...
                             FlowY(1:NPixY,2:NPixX-1) - FlowY(2:NPixY+1,2:NPixX-1);
```

NetFlow = net balance of water flow in one grid cell [m^3 day^{-1}]. This is the net change of the state variables.

(iii) Integration

To calculate the new amount of water in one grid cell, we need to update the old state with the new change:

```
I(1:NPixY, 2:NPixX-1) = I(1:NPixY,2:NPixX-1) + NetFlow(1:NPixY,2:NPixX-1) * dt
```

I = amount of water in one grid cell [m^3]

In this integration step, we calculate the amount of water at time $t = t + \Delta t$ from the old amount at time t and the net flow from both directions. For simplicity, we use the forward rectangular integration (Euler-method).

If there is precipitation above the grid cell, this amount of water must be taken with the integration step:

```
I(1:NPixY, 2:NPixX-1) = I(1:NPixY,2:NPixX-1) + (NetFlow(1:NPixY,2:NPixX-1)+PrecVol)
* dt
```

The unit of precipitation must be equal to water flow, meaning [m^3 day^{-1}], therefore the rain intensity (m day^{-1}) must be multiplied with the area of the grid cell.

Finally, the new water volume can be substituted to hydraulic head based on the amount of pores (PorVol):

```
H(:,2:NPixX-1) = H(:,2:NPixX-1) + ...
        (PrecVol(:,2:NPixX-1)+ NetFlow(1:NPixY,2:NPixX-1))* dt /(PorVol*PixX*PixY);
```

This final statement will be used in the program.

(iv) Time step update
After the integration step, in which new amounts at time = t+Δt are calculated, the time can be updated with Δt.

```
Time = Time + dt;
```

3.1.1 Boundary conditions

Until now, we did not calculate the water flow at the boundaries of the research area. For these left and right columns, at the position of the ditches, we assume a constant water level:

```
H(:,1) = 0;
H(:,NPixX) = 0;
```

For the uppermost and lowermost row, we assume no flow in or out the system:

```
FlowY(1,:) = 0;
FlowY(NPixY+1,:) = 0;
```

On Blackboard, in Appendix III-a a total overview of the grid cell discretization is given. In Appendix III-b, an example programme of the groundwater flow program based on Hooghoudt is shown.

3.2 Exercise Hooghoudt and two-dimensional discretization

3.2.1 As shown in Appendix III-a, the hydraulic head is valid in a grid cell, while the flows are valid between cells. Which variables are valid within a grid cell or between grid cells?

	within a cell	between cells
PixX		
DemClay		
Net		
PorVol		
PixY		
Q		
I		
Prec		
KDX		

3.2.2 In the groundwater-Hooghoudt model, the water level in the ditches will not change in time. Which part of the source code of the program is responsible for it? Tip: Also look to the dynamic part of the model.

What does this mean in a hydrological context?

3.2.3 Why is the distance between the ditches about 8 meter?

3.2.4 In the statement to calculate KDX, you find (0.5*(H(:,1:NPixX-1) + H(:,2:(NPixX)).

- Why is it 0.5?
- What is the dimension of the ":" in (H(**:**,1:NPixX-1))?
- Why is it 1:NPixX-1 and not 1:NPixX?

3.2.5 In the statement to calculate KDX we first multiply with PixY, and in FlowX we divide by PixX. Can you explain why we do this?

3.2.6 You can find the Hooghoudt model on Blackboard. The flow in the y-direction has been left out.

- Include this in the script;
- Run the model for 50 and 100 days. In the 100-day run you find some peaks. This is due to a numerical problem, so the used discretization is not accurate enough. How can you solve this problem?

3.2.7 Run the model again for 10, 50 and 100 days.

- What is the equilibrium state? And why do you think that the simulation goes to an equilibrium?
- Will an increase of precipitation rate change the equilibrium? Why?
- You can also change the hydraulic conductivity of the system. If you use a more sandy soil (k = 0.5 m d^{-1}), what will happen with the groundwater level? (Use a smaller dt.)

3.2.8 To make sure that you do not make numerical errors, it is important to check the water balance of the model. The total water balance of the system (continuity equation) must be 0, so check=in-out-change of storage;

- IN: Calculate for every time step the total amount of precipitation.
  Tip 1: Do it within the while-loop and intitialize with 0; use TotPrec=TotPrec+(..total amount of prec on the grid); the unit is m^3;
  Tip 2: Do not calculate the amount of rain falling on the ditches.
- OUT: Use the following statement for the outflow:
  `TotFlow=TotFlow-sum(FlowX(:,2))*dt+sum(FlowX(:,NPixX))*dt;`
  Why do we use (:,2) and (:,NPixX) and why is it – and +?
- CHANGE in STORAGE:
  `Storage=(sum(sum(H-InitH)))*PixX*PixY*PorVol;`

3.2.9 Include a slope of the impermeable layer (DemClay) in such a way that the depth at (NpixY,:) is –0.5 m and at the other side (1,:) still –1 m. Explain the results after 5 days.
Tips:

- The DemClay variable changes from one single number to a matrix. This matrix can be created from a column vector. You can make it using a for-loop:

```
for i=1:NpixX
     (Copy the vector);
end
```

Do it first outside the programme by copy-paste before you run the whole programme.
- The thickness of the water layer will also change by calculating KDX and KDY.

3.2.10 Go back to the normal Hooghoudt model without a slope. Until now we didn't use the Digital elevation model above reference level. For instance, if you use a precipitation rate of 0.1 m d^{-1}, the groundwater level will come above the ground level.

We now will use a real DEM (Digital Elevation Map) of a polder. Download from Blackboard the file DEMpolder.txt. For this polder, the following initial conditions should change:
- Pixel size is 2×2 m;
- dt = 0.1 day
- K = 0.5 m day^{-1}
- Rain intensity remains 1 cm day^{-1}
- Pore Volume is 0.35

Now we will calculate the amount of Runoff and assume that if the ground water level is above the DEM, then this water will be transported away at once:

```
Runoff=(H>DEMPolder).*(H-DEMPolder)*PorVol*PixX*PixY;              % m3
H=H-Runoff/(PorVol*PixX*PixY);                                     % m
```

Change the water balance, because now we have an extra loss.

- Run the model for 10 days and make a contour plot of the runoff. Show the groundwater level at a cross-section of the groundwater level at y=3 and y=29 and hand this in.
- Run the model for 100 days. Hand in some plots and explain what you see.

```
%  Groundwater flow
%  Based on Hooghoudt-situation
%
%%%%%%%%%%%%%% INITIALISATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
% system discretisation
PixX = 0.079;                                % Pixel length in X-direction           [m]
PixY = 0.10;                                 % Pixel length in Y-direction           [m]
NPixX=100;                                   % Number of Pixels in X-direction       [-]
NPixY=10;                                    % Number of Pixels in Y-direction       [-]
x(1:NPixX)=[0:PixX:(NPixX-1)*PixX]+PixX/2;   % x-grid                                [-]
y(1:NPixY)=[0:PixY:(NPixY-1)*PixY]+PixY/2;   % y-grid                                [-]
FieldLx = NPixX*PixX;                        % FieldLength in X-direction            [m]
FieldLy = NPixY*PixY;                        % FieldLength in Y-direction            [m]

% control constants
StartTime=0;                                 % Start time for simulation           [day]
EndTime=50;                                  % time at which simulation ends       [day]
dt = 0.005;                                  % calculation time step               [day]

% system constants
DemClay = -1;                                % Dem of Clay above reference level     [m]
K = 0.1;                                     % hydraulic Conductivity            [m/day]
PorVol = 0.25;                               % Pore Volume                           [-]

% initialisation and boundary conditions
Time = StartTime;                            % Initialisation of time              [day]
InitH=0.0;                                   % Initial hydraulic Head (H) value      [m]
H(1: NPixY,2: NPixX-1) = InitH;              % field of H values                     [m]
H(1:NPixY,1) = 0;                            % height in drain = 0 level (left)      [m]
H(1:NPixY,NPixX) = 0;                        % height in drain = 0 level (right)     [m]
FlowY(1:100,:) = 0; FlowY(NPixY+1,:) = 0;       % bound.con.no flow in/out to Y-
dir[m3/day]
Prec(1: NPixY,1: NPixX) = 0.01;              % field of Precipitation Rate       [m/day]

%%%%%%%%%%%%% DYNAMIC LOOP %%%%%%%%%%%%%%%%%%%%%%%%%%%

while Time <= EndTime

  % calculate Flow in x-direction : Flow = -kD * dH/dt; (kD = k * average height of watercolumn * PixY)
  KDX = K * (0.5*(H(:,1:NPixX-1) + H(:,2:(NPixX))) -DemClay) * PixY;             %[m3/day]
  FlowX(:,2:NPixX) = -1 * KDX .* (H(:,2:(NPixX))-H(:,1:NPixX-1)) / PixX;        %[m3/day]


  % calculate Flow in y-direction: Flow = -kD * dH/dt; (kD = k * average height of watercolumn * PixX)
  KDY(???, ???) = ???
  FlowY(???,???) = ???

  % calculate net flow and precipitation rate in Volume per Pixel;
  NetFlow(1:NPixY,2:NPixX-1) = FlowX(1:NPixY,2:NPixX-1) - FlowX(1:NPixY,3:NPixX) + ...
                       FlowY(1:NPixY,2:NPixX-1) - FlowY(2:NPixY+1,2:NPixX-1); %  [m3/day]
  PrecVol = Prec * PixX * PixY;                                                %  [m3/day]

  % calculate new H values by forward integration
  H(1:NPixY,2:NPixX-1) = H(1:NPixY,2:NPixX-1) + ...
                        (PrecVol(1:NPixY,2:NPixX-1)+ NetFlow(1:NPixY,2:NPixX-1))* dt/ ...
                        (PorVol*PixX*PixY);                                    %  [m]


   Time = Time + dt;

end
%%%%%%%%%%%%%% END DYNAMIC LOOP %%%%%%%%%%%%%%%%%%%%%%%%%%%
```