UNIVERSITÀ DEGLI STUDI DI PADOVA

Groundwater hydrology project

Precision irrigation Modelling of a cranberry field in Quebec (Canada)

Andrea Gorgi

University of Padova

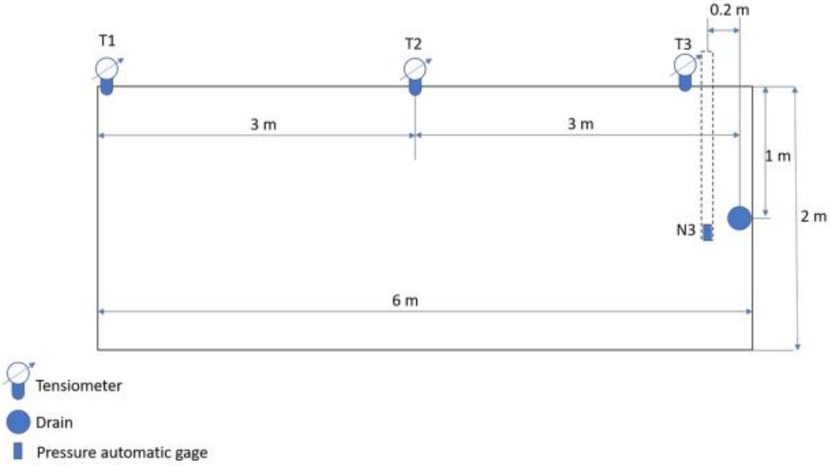
February 2022

Outline

1. First Part: Model setting in Cathy
   - Creation of the geometry
   - Initial and Boundary conditions
   - Atmospheric conditions/ soil parameters
   - First Result
2. Calibration
   - Possible approaches
   - First Improvement Local Search
   - Physics effect of parameters
   - Results
3. Validation
   - Conclusions

Creation of the geometry

Exploit main features of the assignment.



Creation of the geometry

Idea: z-axis horizontal plane 1 meter below surface. z discretization with parallel planes. Drain at $x = 0$ as a long monodimensional tube in the $y-$direction.

Mesh to have exact points at

1. T1 (6,0.5,0.9);
2. T2 (3,0.5,0.9);
3. T3 (0.2,0.5,0.9);
4. Drain (0,0:1,0);

Surface definition: hap.in

Domain extension in x = 6, $\frac{6}{0.2} = 30 \implies$ 30 cells in x.
Problem y-symmetric $\implies$ Consider sufficient 2 cells in y.

```
--------------------------------------------------------------------------------
STRUCTURAL PARAMETERS
--------------------------------------------------------------------------------
Grid spacing along the x-direction =                          0.50
Grid spacing along the y-direction =                          0.50
DEM rectangle size along the x-direction =                   30
DEM rectangle size along the y-direction =                    2
Number of cells within the catchment =                       60
X low left corner coordinate =                                0.00000000
Y low left corner coordinate =                                0.00000000
--------------------------------------------------------------------------------
```

Surface definition: dem13.val

File with 30 columns and 2 rows of "1.0".

Last value equal to 0.99999 to ensure possibility of outflow.

```
fid = fopen('dtm_13.val', 'w');
N_x = 2;
N_y = 30;
for i_x=1:N_x
    for i_y=1:N_y
        fprintf(fid, '1.0  ');
    end
    fprintf(fid,'\n');
end
fclose(fid);
```

```
1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  0.9999999
```

Mesh creation: dem_parameters

Exact points at T1 (6,0.5,0.9), T2 (3,0.5,0.9), T3 (0.2,0.5,0.9), Drain (0,0:1,0).
Layer at 0.1 (5% of 2) from the top (not necessarily the first layer);
Layer at 1 (50% of 2) from the top.
Finer discretization near surface and near drain?

```
0.2
0.5
1.0
1
1 14 25
0 0 2.0
0.01  0.02  0.02  0.03  0.04  0.05  0.06  0.08  0.09  0.10  0.10  0.11  0.13  0.16

delta_x
delta_y
factor
dostep
nzone  nstr  n1
ivert  isp   base
zratio(i).i=1.nstr
```

Mesh creation: Parm file

Set $IPRT1 = 3$ and obtain the mesh.

Adapt some convergence parameter (DELTAT, DTMIN, DTMAGM).

One VTK file for each hour.

Set maximum time equal to 500 for calibration (see next).

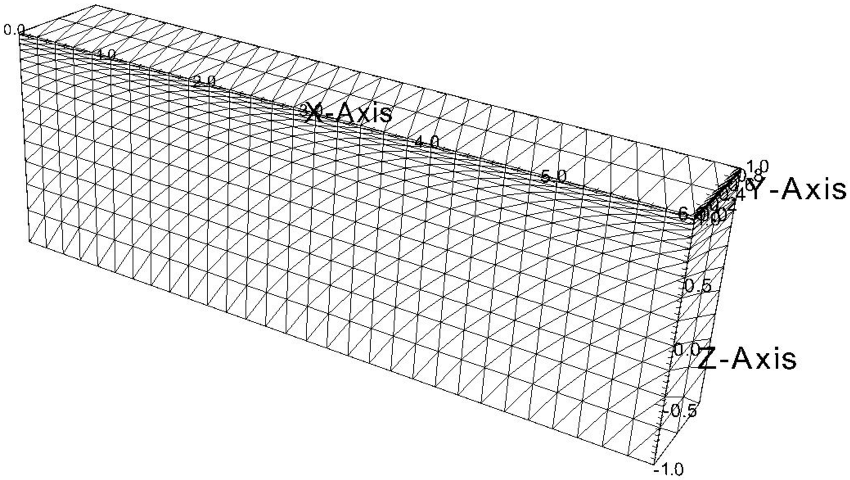
```
2         0         1                 IPRT1  NCOUT  TRAFLAG
2         0.00      0                 ISIMGR PONDH_MIN VELREC
0         0.01                        KSLOPE TOLKSL
-3.0      -1.0      -3.0      -1.0     PKRL   PKRR   PSEL   PSER
-3.0      -2.5      -1.5      -1.0     PDSE1L PDSE1R PDSE2L PDSE2R
0         0         0                 ISFONE ISFCVG DUPUIT
1.        1         1                 TETAF  LUMP   IOPT
0         0.8                         NLRELX OMEGA
0         1.0e-4  1.0e+30    1.0e+30  L2NORM TOLUNS TOLSWI  ERNLMX
15         5         7                 ITUNS  ITUNS1 ITUNS2
2         500         1.0e-10         ISOLV  ITMXCG TOLCG
1        0.0000001        1.0     500.0   DELTAT DTMIN  DTMAX  TMAX
0.0       4.0       0.0        0.5     DTMAGA DTMAGM DTREDS DTREDM
2         2       500
1.0
2.0
3.0
4.0
5.0
6.0
```

Units of measure

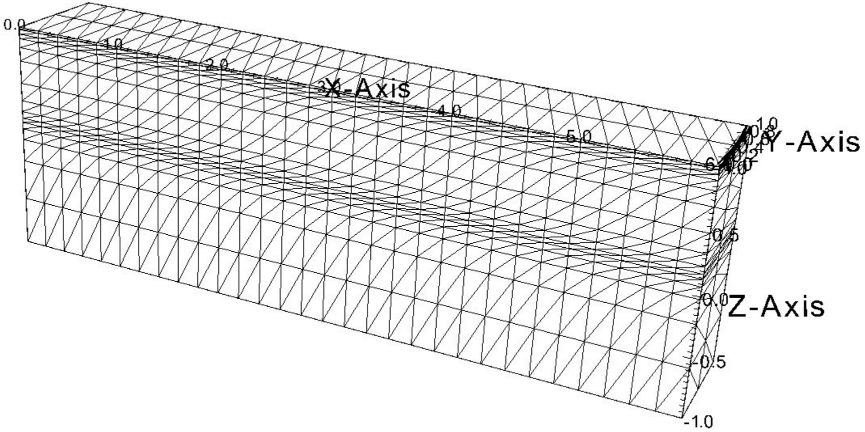
Data for 2018 hours($\sim$ 84days) $\implies$ Time unit: hours (days);
Length unit: meter.

- Pressure head $[\psi] = m$;
- Hydraulic conductivity $[K] = mhours^{-1}$;
- Specific storage $[S] = 1/m$.

Mesh Results



Mesh Results



Initial condition: IC file

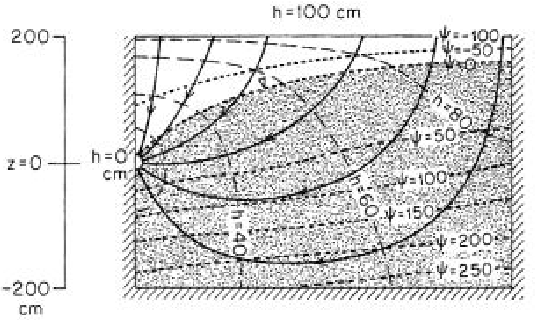
Unsaturated zone $\implies$ partially saturated vertical hydrostatic equilibrium as IC.
WTPOSITION in depth from the surface.
First trial: initial water table depth of 0.4 meters (will be calibrated).

```
4   0 INDP IPOND
0.4   WTPOSITION
```

Boundary Conditions: drain

Three possibilities:

- $\psi = 0$ at the drain;
- $\psi = 0$ at drain + hydrostatic pressure in the vertical;
- $\psi = 0$ at the points in N3 (variable condition in time).



Freeze and Cherry. Ideal drain in saturated-unsaturated region.

Dirichlet BC

```
clear all; close all; clc;
fid = fopen('project/runs/weilletal/input/nansfdirbc','w');
fprintf(fid, '0.0000000000000000E+000                              TIME\n');
flag = 0;
%%% FIND NODES
nodes = Nodes();
% [0.01  0.02  0.02  0.04  0.06  0.10  0.06  0.05  0.03  0.02  0.03  0.07  0.10  0.18  0.21];
z = 2*[0 0.01  0.02  0.02  0.03  0.04  0.05  0.06  0.08  0.09  0.10  0.10  0.11  0.13  0.16];
sum = 0;
Tmax = 2018.0;
for i = 1:length(z)
    sum = sum + z(i);
    z(i) = sum;
end
z = 1 - z;
switch flag
    case 0
        z = z(z<=0);
        id = zeros(length(z),3);
        for i = 1:length(z)
            id(i,1:3) = (nodes.Find(2,0,0,z(i)))';
        end
        fprintf(fid, '      0 %d \n', length(z)*3);
        fprintf(fid,'       ');
        for i = 1:length(z)
            fprintf(fid,' %d %d %d ', id(i,:));
        end
        fprintf(fid,'\n       ');
        for i = 1:length(z)
            fprintf(fid,' %2.1f %2.1f %2.1f ', -z(i), -z(i), -z(i));
        end
```

```
case 1
    input = xlsread('Input_data_completo');

WTD = input(:,4);
id = zeros(1,3);
for i=1:Tmax
    z_search = WTD(i);
    incumb = 1e5;
    for j = 1 : length(z)
        dist = abs(z(j) - z_search);
        if dist < incumb
            incumb = dist;
            index = j;
        end
    end
    id = nodes.Find(2,0,0,z(index));
    %--------------------------------
    if i ~= 1
        fprintf(fid,'\n\n   %4.1f\n', i);
    end
    fprintf(fid, '      0 %d \n', 3);
    fprintf(fid,'       ');
    fprintf(fid,' %d %d %d ', id(:));

    fprintf(fid,'\n       ');
    fprintf(fid,' %2.1f %2.1f %2.1f ', 0.0, 0.0, 0.0);
```

Dirichlet BC: class Nodes

```
%------------------------------------------------------------------
function this = Nodes()
    path = "project/runs/weilletal/output/xyz";
    fileID = fopen(char(path),'r');
    formatSpec = '%f %f %f %f';
    A = textscan(fileID, formatSpec, 'headerLines', 1);
    A = cat(4,A{:});
    this.indeces = A(:,1);
    this.x_coord = A(:,2);
    this.y_coord = A(:,3);
    this.z_coord = A(:,4);
end
%------------------------------------------------------------------
function outIndex = Find(this, flag, x, y, z)
    % flag = 0: just that node;
    %      = 1: all possible x;
    %      = 2: all possible y;
    %      = 3: all possible z;
    toll = 1e-4;
    switch flag
        %----------------------------------------------------------
        case 0
            outIndex = 0;
            for i = 1:this.indeces(end)
                if (abs(this.x_coord(i) - x)<toll && abs(this.y_coord(i) - y)<toll && abs(this.z_coord(i) - z) < toll)
                    outIndex = this.indeces(i);
                    break;
                end
            end
        %----------------------------------------------------------
```

Atmospheric BC: rain+evapotranspiration

Compute NET flow : *drain* $-$ *ET* for each time step and adjust the dimensions.

```
fid = fopen('project/runs/weilletal/input/atmbc', 'w');

input = xlsread('Input_data_completo');
RF  = input(:,6);
ETP = input(:,7);

hours=2018;
atbc=zeros(1,hours);
fprintf(fid, "1 1                   HSPATM IETO\n\n");
for i=1:hours
    atbc(i)=(RF(i)-ETP(i))/1000;
    fprintf(fid, ...
    '%5.1f       TIMEIN \n  %9.7f       ATMINP \n\n',...
                                        (i-1),atbc(i));
end
```

Soil parameters

Assume medium-textured soil (50-70% sand, 25-40% silt, 5-15% clay).

Table 2. Approximate ranges of saturated hydraulic conductivities expected for materials commonly encountered in geoenvironmental engineering based on permeation with water

Material	Saturated hydraulic conductivity (ms^{-1})	Comments
Gravel	10^{-2}–10^{-3}	Values based on "clean" soils;
Sand	10^{-3}–10^{-5}	variation in k_{sat} based on particle size distribution
Silt	10^{-5}–10^{-8}	Variation in k_{sat} based on mineralogical composition of silt particles
Clay	10^{-8}–10^{-12}	Variation in k_{sat} based on mineralogical composition of clay particles
Geosynthetic clay liner	10^{-10}–10^{-11}	Values based on sodium bentonite sandwiched between two geotextiles
Sand–bentonite mixture	10^{-9}–10^{-10}	Values based on a mixture of clean sand (w/o fines) and 4–10% (w/w) sodium bentonite

Samples Number		D1	D2	D3	I1	I2	I3	P1	P2	P3
	Clay (%)	14.09	27.60	25.70	13.71	11.84	13.45	21.58	11.19	11.25
Texture	Silt (%)	12.45	25.88	21.50	21.61	20.63	15.26	18.20	23.34	19.39
	Sand (%)	73.46	46.52	52.80	64.68	67.53	71.18	60.22	65.67	69.36
Bulk density, g $(cm^3)^{-1}$		1.20	1.23	1.21	1.17	1.18	1.14	1.08	1.01	1.16
Particule density, g $(cm^3)^{-1}$		2.53	2.54	2.53	2.50	2.48	2.54	2.54	2.44	2.61
Porosity (%)		52.65	51.49	52.17	53.22	52.43	55.12	57.49	58.61	55.56
Organic matter (%)		2.15	2.01	2.09	2.01	1.93	1.79	1.99	4.22	2.88
Aggregate stability (%)		46.03	32.13	32.17	21.62	16.86	17.22	61.42	88.28	75.85
Lime (%)		1.09	1.09	1.31	1.28	1.40	1.46	2.80	2.58	3.47
Hydraulic conduct, cm h^{-1}		6.04	4.63	5.09	5.25	4.55	4.75	7.35	8.98	9.56

Soil: Range of parameters

Obtain an approximate range of variability of the parameters

- K = 0.02 m/hour ÷ 2.53 m/hour;
- S_s = 1E-5 m^{-1} ÷ 0.01 m^{-1};
- η = 50% ÷ 57%.

The first values used are:

WTPOS	PERMX	PERMY	PERMZ	ELSTOR	POROS	VGN1	VGN2	VGN3
0.4	0.5	0.5	0.5	0.005	0.55	1.46	0.15	0.03125

First Result

Drain applied with second approach.

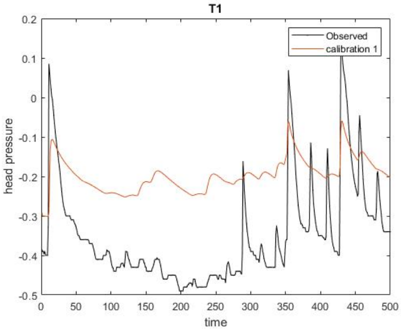


Figure: MSE = 0.033 m^2; NSE = -0.999; KGE = 0.173

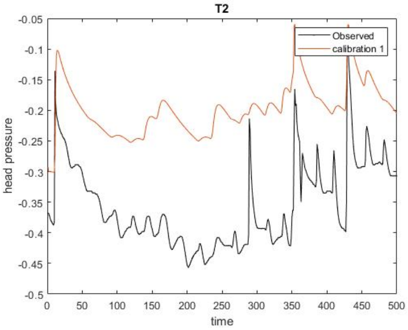


Figure: MSE = 0.027 m^2; NSE = -4.87; KGE = 0.357

First Result

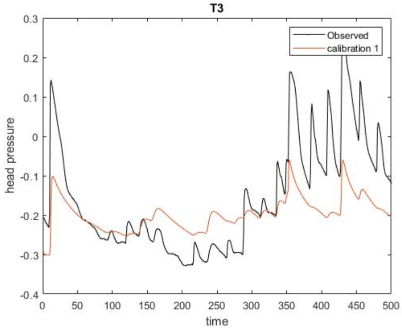
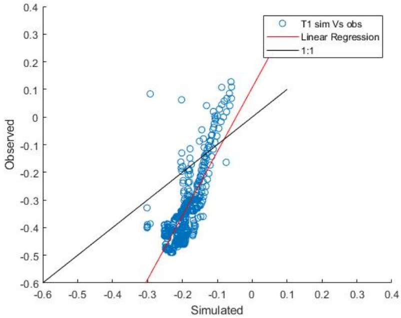


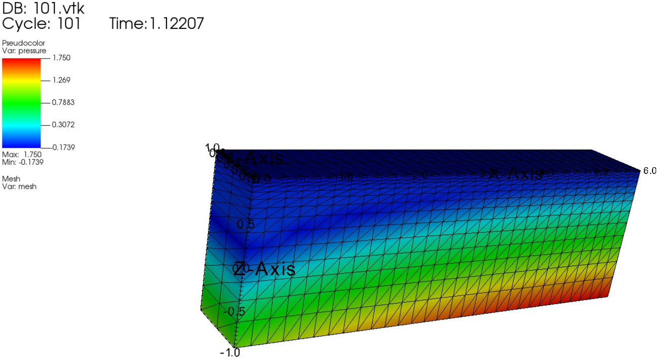
Figure: MSE = 0.0133 m^2; NSE = 0.3144; KGE = 0.233



Scatter plots depicting simulated and observed head pressures. k_s = 2.31.

First Result

Pressure "Pseudocolor" result in Visit at time 1.12 hours.



CALIBRATION

Introduction

- Test on 1/4 of total data (approximately 500 hours). Remaining used for validation.
- Choose and objective function to minimize (MSE) or maximize (NSE or KGE).
- Various possible heuristic models to find "good" values of parameters.
- Just some:
  - First improvement local search;
  - Best improvement local search;
  - Genetic algorithm (?).
- Need to discretize the search space (finite number of possible values combinations).
- Define a "move". Idea: Firstly great steps, find local optima, then reduce steps.

Local Search

```
//-----------
double WTpos  = 0.35;
double PERMX  = 1.0;
double PERMY  = 1.0;
double PERMZ  = 1.5;
double ELSTOR = 0.005;
double POROS  = 0.55;
double VGN1   = 1.46;
double VGN2   = 0.15;
double VGN3   = 0.03185;
//-----------
/// INCREMENTS
double delta_WTpos  = -0.05;
double delta_PERMX  = 0.0125;
double delta_PERMY  = 0.0125;
double delta_PERMZ  = 0.0125;
double delta_ELSTOR = 0.02;
double delta_POROS  = -0.05;
double delta_VGN1   = 0.01;
double delta_VGN2   = 0.01;
double delta_VGN3   = 0.0005;
//-----------
```

Code hints

- Increase first parameter (horizontal permeability).
- Build dynamically IC, parm and soil files.
- Launch the cathy_ft.
- Get the pressure values and the interested nodes.
- Compute KGE (NSE) errors and check for improvements.
    - If improvement restart the cycle and increase horizontal permeability
    - else decrease permeability to previous value and increment the new one.
- Finally, "change move" and check convergence.

```
void setIC(const double& WTdepth) {
    std::ofstream icFile;
    icFile.open( "runs/weilletal/input/ic");
    icFile << 4 << std::setw(3)  << 0 << " INDP IPOND \n";
    icFile <<  WTdepth << "  WTPOSITION";
    icFile.close();
}
```

Objective function

- From results: parameters that increase fit at T1 or T2 decrease it in T3.
- Possible motivations:
  - Drain might not be ideal;
  - Observed data near drain can be affected by errors;
  - Richard's equation is not enough near the drain.
- Idea: Penalize the error in T3 in the objective function.

```
// KGE
double KGE_1 = 1 - ED_1; double KGE_2 = 1 - ED_2; double
std::cout << "KGE1 = " << KGE_1 << "\n  KGE2 = " << KGE_2
curr_err = (KGE_1 + KGE_2 + 0.5*KGE_3) /(2.5);
```

First Improvement Local Search

```cpp
if (curr_KGE > incumbent_KGE) {
    //--------------------------------------------
    iter += 1;
    std::ofstream resultFile;
    snprintf(name, NAME_SIZE, "Results3/result%d.txt", iter);
    char const * filename = (char*)(&name[0]);
    resultFile.open(filename);
    resultFile << std::setw(15) << "T1" << std::setw(15) << "T2" << std::setw(15) << "T3" << std::setw(15) << "TIME" << std::setw(15) << " KGE_ERR\n";
    for (int i = 0; i < Tmax; i++) {
        double T1_curr; double T2_curr; double T3_curr;
        double curr_time;
        if (i == 0){
            double lambda = 1.0/times[i];
            curr_time = lambda * times[i];
            T1_curr = lambda * T1[i]; T2_curr = lambda * T2[i]; T3_curr = lambda * T3[i];
        } else{
            double lambda = (i+1 - times[i-1])/(times[i] - times[i-1]);
            curr_time = lambda * times[i] + (1-lambda)*times[i-1];
            T1_curr = lambda * T1[i] + (1 - lambda) * T1[i-1];
            T2_curr = lambda * T2[i] + (1 - lambda) * T2[i-1];
            T3_curr = lambda * T3[i] + (1 - lambda) * T3[i-1];
        }
        resultFile << std::setw(15) << T1_curr << std::setw(15) << T2_curr << std::setw(15) << T3_curr << std::setw(15) << curr_time << std::setw(15) << curr_KGE;
        resultFile << std::endl;
    }
    resultFile.close();
    incumbent_KGE = curr_KGE;
    /// PRINT RESULTS
    outFile << "SOLUTION IMPROVED!! \n";
    std::cout << "SOLUTION IMPROVED!! \n";
    switch (count)
    {
    case 0:
      std::cout << "IMPROVED WTpos \n";
      outFile << "IMPROVED WTpos \n";
      break;
    case 1:
    std::cout << "IMPROVED PERMX/Y \n";
    outFile << "IMPROVED PERMX/Y \n";
    break;
    case 2:
```

First Improvement Local Search

```
        count = 0;
        PERMX += delta_PERMX;
        PERMY += delta_PERMY;
        // WTpos += delta_WTpos;
        // std::cout << "trial increase WTpos \n";
        // outFile << "trial increase WTpos \n";
    } else {
            count += 1;
            switch (count)
            {
                case 0:
                WTpos += delta_WTpos;
                    break;
                case 1:
                //WTpos -= delta_WTpos;
                PERMX += delta_PERMX;
                PERMY += delta_PERMY;
                std::cout << "trial increase PERMX \n";
                outFile << "trial increase PERMX \n";
                    break;
                case 2:
                PERMX -= delta_PERMX;
                PERMY -= delta_PERMY;
                PERMZ += delta_PERMZ;
                std::cout << "trial increase PERMX \n";
                outFile << "trial increase PERMX \n";
                    break;
                case 3:
                PERMZ -= delta_PERMZ;
                ELSTOR += delta_ELSTOR;
                std::cout << "trial increase POROS \n";
                outFile << "trial increase POROS \n";
                break;
                case 4:
                ELSTOR -= delta_ELSTOR;
                POROS += delta_POROS;
                std::cout << "trial increase POROS \n";
```

```
CALIBRATION
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35     0.9     0.9     1.5   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.421991

SOLUTION IMPROVED!!
IMPROVED PERMX/Y
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35   0.925   0.925     1.5   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.421803

trial increase PERMX
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35     0.9     0.9   1.525   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.423099

SOLUTION IMPROVED!!
IMPROVED PERMZ
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35   0.925   0.925   1.525   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.422666

trial increase PERMX
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35     0.9     0.9    1.55   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.424387

SOLUTION IMPROVED!!
IMPROVED PERMZ
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3

     0.35   0.925   0.925    1.55   0.005    0.55    1.46    0.15 0.03135
KGE ERROR: 0.424806

SOLUTION IMPROVED!!
IMPROVED PERMX/Y
    WTPOS   PERMX   PERMY   PERMZ  ELSTOR   POROS    VGN1    VGN2  VGN3
```

MSE and NSE

- MSE (mean squared error) $[0, +inf]$; MSE $= \frac{\sum_{t=1}^{n}(x_{s,t} - x_{o,t})^2}{n}$
- NSE (Nash–Sutcliffe efficiency) $[-inf, 1]$;
  - $NSE = 1 - \frac{MSE}{\sigma_o^2}$;
  - Classic skill score. If NSE$\leq$0 observed mean is a better predictor;
  - Likely underestimates of the variability in the flows;
  - May lead to a Pareto set of optimal solutions
- Decomposition:
  - MSE $= 2\sigma_s\sigma_o(1 - r) + (\sigma_s - \sigma_o)^2 + (\mu_s - \mu_o)^2$;
  - NSE $= 2\alpha r - \alpha^2 - \beta_n^2$
    $\alpha = \frac{\sigma_s}{\sigma_o}$, $\beta_n = (\mu_s - \mu_o)/\sigma_o$.

KGE

- Ideas:
  - Corrected formulations;
  - Multi-objective perspective (KGE).
- Klinga-Gupta efficiency (KGE) = 1- ED
- ED = $\sqrt{(r-1)^2 + (\alpha - 1)^2 + (\beta - 1)^2}$
- $\beta = \mu_s / \mu_o$

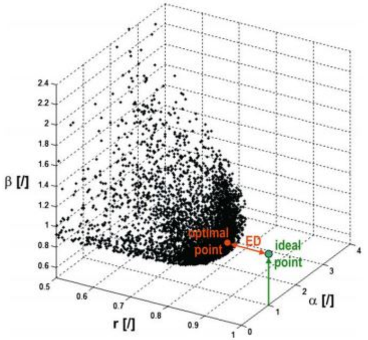
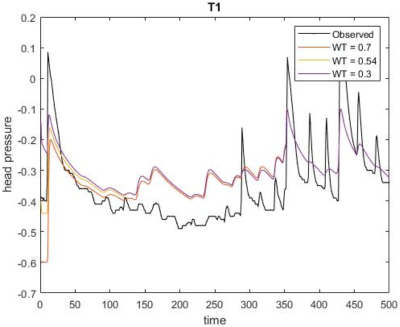
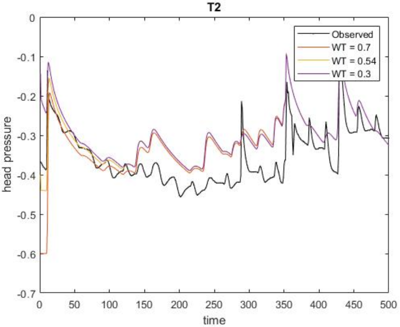


Fig. 2. Example for three-dimensional Pareto front of r, α and β. ED is the Euclidian distance between the optimal point and the ideal point, where all three measures are 1.0. Glan River, Austria, 432 km^2, 5 years daily data, HBV model variant, random parameter sampling.

Move: change one parameter's value

- Useful to understand parameters effects on the solution.
- Not so effective if parameters effects are dependent to each other.

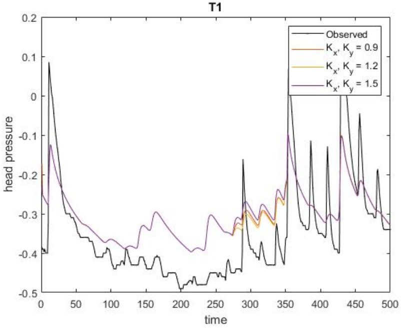


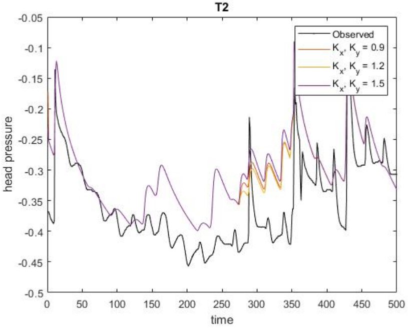


Effects of initial conditions decrease in time and vanish at approximately 350 hours.

Change one parameter's value: K_x/K_y

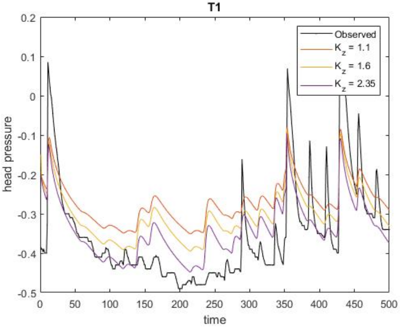
Change the horizontal conductivity.





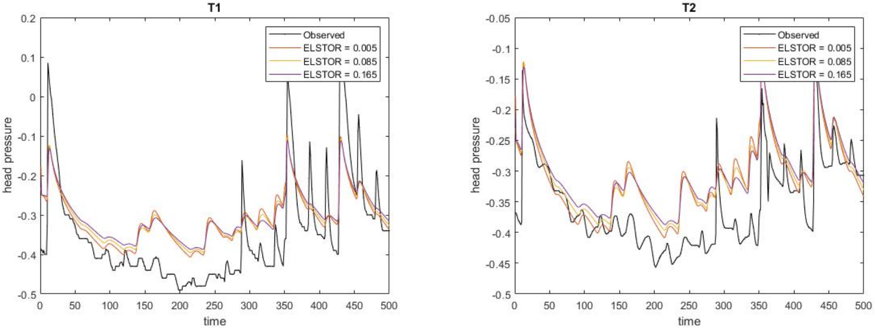
Change one parameter's value: K_z

Change the vertical conductivity.



Change one parameter's value: *ELSTOR*

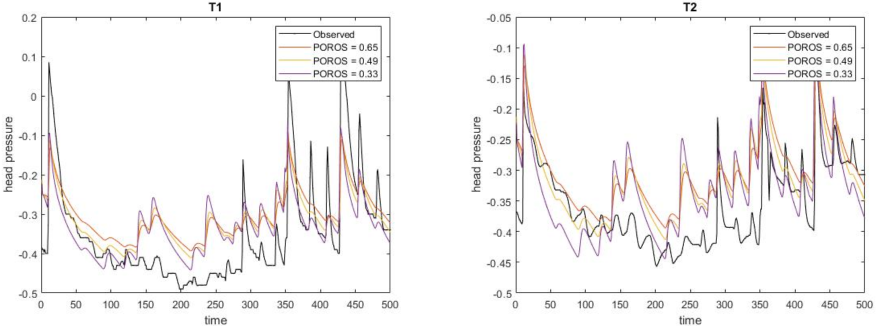
Change the specific storage value.



S_s proportional to volume water released per surface area per head drop.
High $S_s \implies \psi$ less sensible to atmospheric conditions.

Change one parameter's value: *POROS*

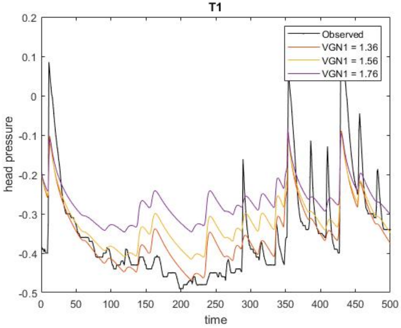
Change the porosity.

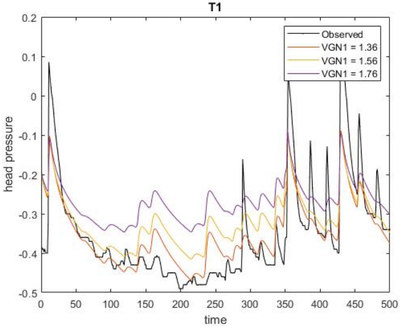


Porosity decrease $\implies$ S_y decrease. More sensible to atmospheric conditions.

Change one parameter's value: $VGN1$

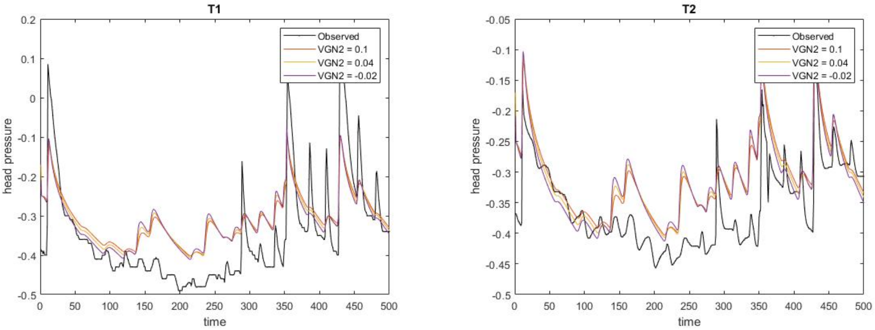
Change N parameter of Van Geneucthen.





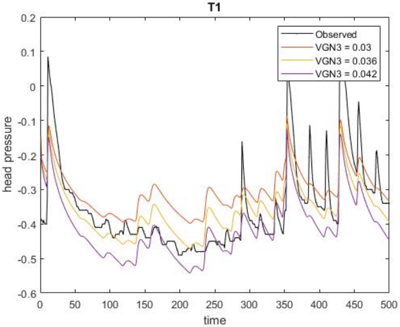
Change one parameter's value: $VGN2$

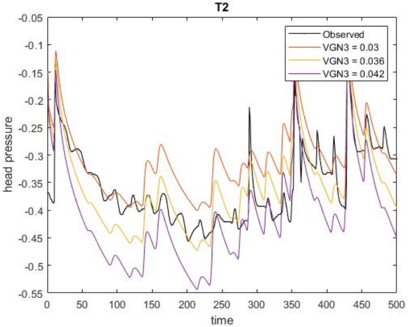
Change theta parameter of Van Geneucthen.



Change one parameter's value: *VGN3*

Change inverse alpha parameter of Van Geneucthen.





Calibration with KGE

Calibration with KGE and first approach for drain boundary conditions.

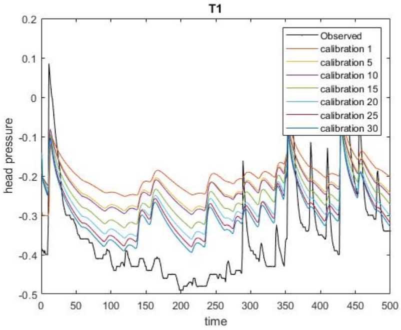


Figure: Best: MSE = 0.0097 m^2; NSE = 0.4122; KGE = 0.450

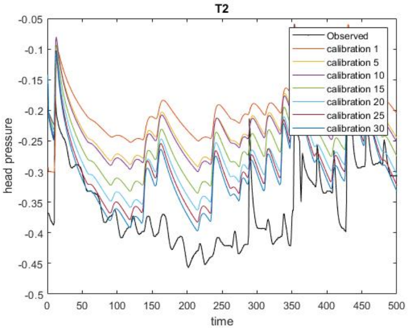


Figure: Best: MSE = 0.0056 m^2; NSE = -0.2154; KGE = 0.6803

Calibration with KGE

See how the best approximation for T1 and T2 is almost the worst for T3.

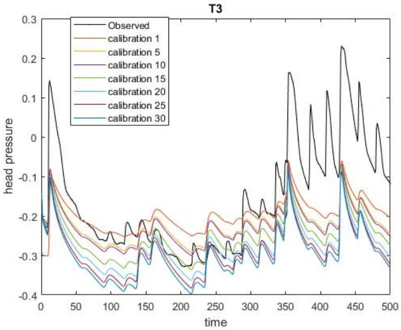


Figure: Best: MSE = 0.0293 m^2; NSE = -0.512; KGE = -0.083

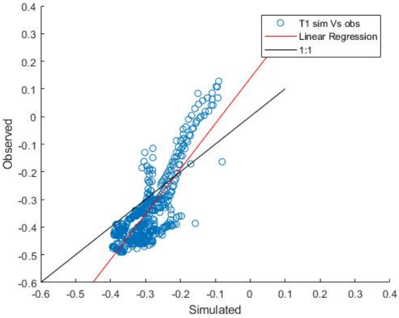
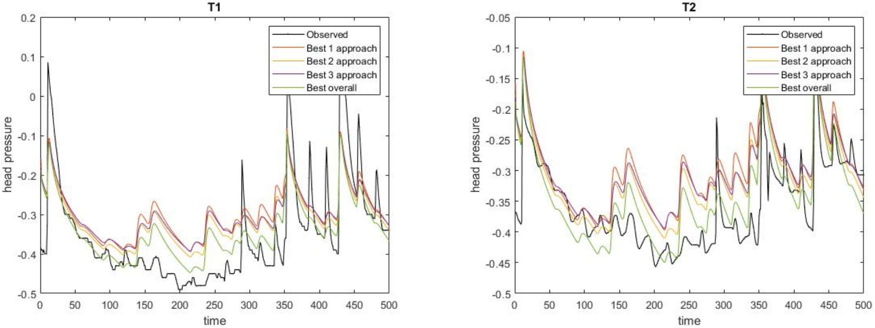


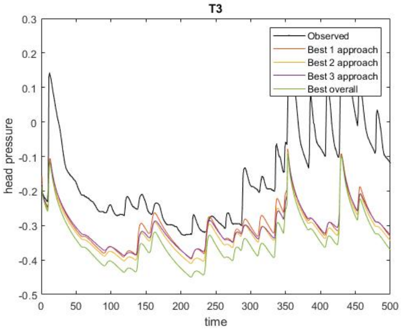
Figure: Scatter plots depicting simulated and observed head pressures. k_s = 1.63.

Calibration with KGE: Comparison



	1 APPROACH			2 APPROACH			3 APPROACH			BEST + NEW MOVE		
	KGE	MSE(cm^2)	NSE	KGE	MSE(cm^2)	NSE	KGE	MSE(cm^2)	NSE	KGE	MSE(cm^2)	NSE
T 1	0.45	0.97	0.41	0.47	0.79	0.52	0.45	0.88	0.47	0.56	0.55	0.66
T 2	0.68	0.56	-0.21	0.73	0.39	0.16	0.72	0.47	-0.012	0.77	0.23	0.5
T 3	-0.083	2.93	-0.51	-0.16	3.34	-0.72	-0.1	3.02	-0.558	-0.29	4.2	-1.15

Calibration with KGE: Comparison



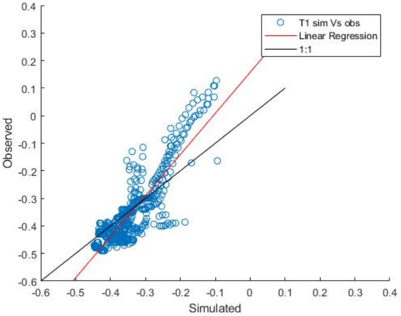


Figure: Scatter plots depicting simulated and observed head pressures. $k\_s = 1.49$.

WTPOS	PERMX	PERMY	PERMZ	ELSTOR	POROS	VGN1	VGN2	VGN3
0.35	1.125	1.125	1.85	0.065	0.55	1.36	0.15	0.03185

KGE ERROR: 0.471666

NSE calibration

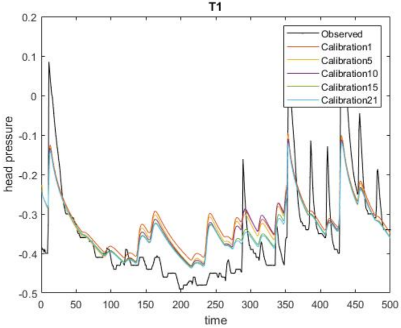


Figure: Best: MSE $= 0.0058\ m^2$; NSE $=$ 0.65; KGE $=$ 0.52

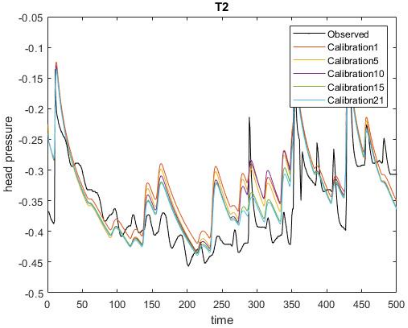


Figure: Best: MSE $= 0.0020\ m^2$; NSE $=$ 0.57; KGE $=$ 0.81

Calibration with NSE

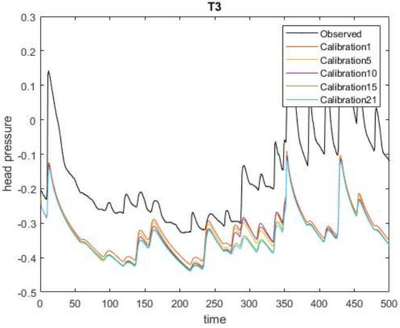


Figure: Best: MSE = 0.0041 m^2; NSE = -1.10; KGE = -0.29

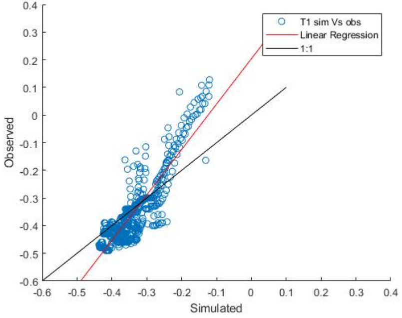


Figure: Scatter plots depicting simulated and observed head pressures. k_s = 1.65.

```
IMPROVED PERMX/Y
   WTPOS    PERMX    PERMY    PERMZ  ELSTOR    POROS     VGN1     VGN2  VGN3

    0.35      1.5      1.5     2.45   0.009      0.7     1.46     0.16 0.03125
NSE ERROR: 0.264315
```

KGE validation

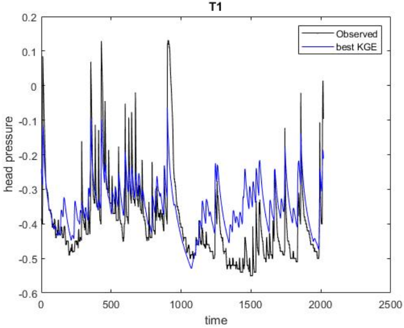


Figure: Best: MSE = 0.0101 m^2; NSE = 0.3315; KGE = 0.49

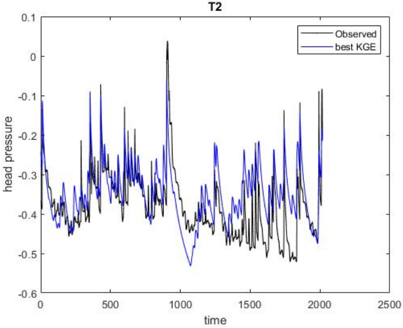


Figure: Best: MSE = 0.0056 m^2; NSE = 0.1503; KGE = 0.61

KGE validation

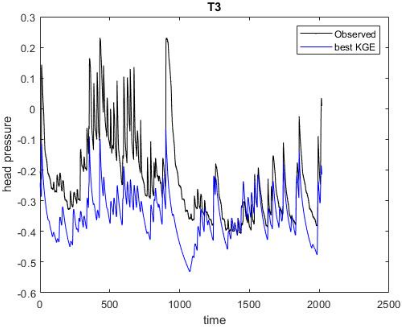


Figure: Best: MSE = 0.0250 m^2; NSE = -0.3285; KGE = 0.23

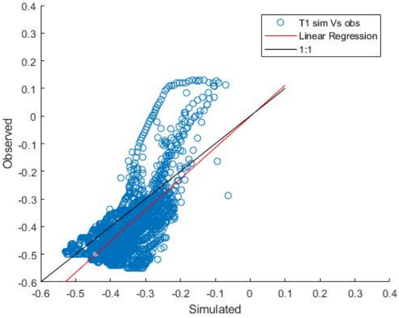


Figure: Scatter plots depicting simulated and observed head pressures. $k_s = 1.12$.

WTPOS	PERMX	PERMY	PERMZ	ELSTOR	POROS	VGN1	VGN2	VGN3
0.35	1.125	1.125	1.85	0.065	0.55	1.36	0.15	0.03185

KGE ERROR: 0.471666

NSE validation

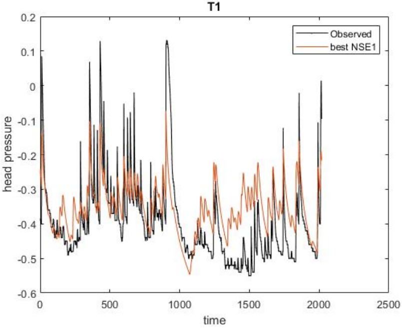


Figure: Best: MSE $= 0.0099\ m^2$; NSE $=$ 0.3324; KGE $= 0.48$

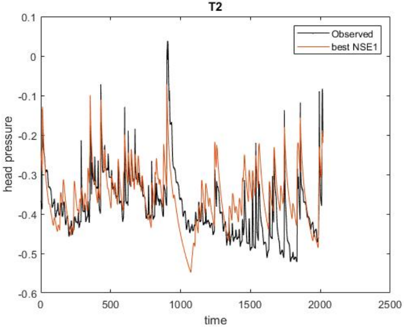


Figure: Best: MSE $= 0.0054\ m^2$; NSE $=$ 0.18; KGE $= 0.61$

NSE validation

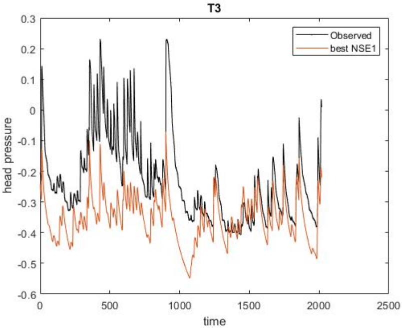


Figure: Best: MSE = 0.0268 m^2; NSE = -0.43; KGE = 0.20

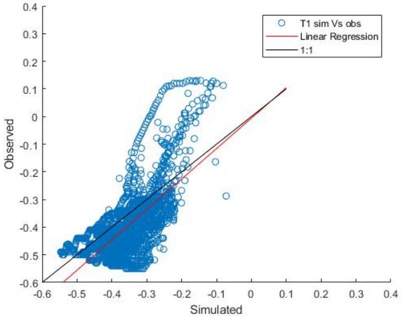


Figure: Scatter plots depicting simulated and observed head pressures. k_s = 1.099.

```
IMPROVED PERMX/Y
   WTPOS    PERMX    PERMY    PERMZ   ELSTOR    POROS     VGN1     VGN2  VGN3

    0.35      1.5      1.5     2.45    0.009      0.7     1.46     0.16 0.03125
NSE ERROR: 0.264315
```

Conclusions

- Better accuracy with second approach for drain, but worst fit in T3;
- Better fit in T3 with third approach, but worst fit overall;
- Calibration on NSE and KGE provides similar results overall of fit, but extremely different paramters:
  - ELSTOR = 0.65 is not so realistic for KGE;
  - PORSO = 0.7 is not so realistic for NSE.