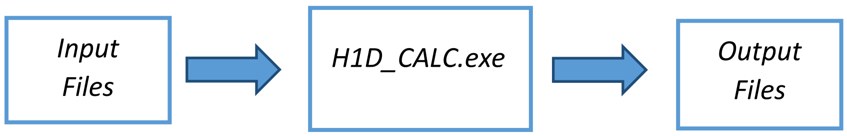
Running Hydrus (1D) in Matlab

E. Diamantopoulos

ediamant@ucdavis.edu



Hydrus

Hydrus uses text format for the input and output files. The GUI creates the input text files, runs the executable H1D_CALC.EXE (default location: "C:\Program Files (x86)\PC-Progress\Hydrus-1D 4.xx\H1D_CALC.exe") and reads the output text files. Sometimes it is needed to conduct multiple Hydrus simulations (e.q., for a sensitivity analysis) and doing that using Hydrus' GUI is very difficult. This tutorial discusses how we can use MATLAB to create Hydrus input files, run Hydrus in Matlab and read the output files for a simple simulation scenario.

Simulation Scenario

This is a simple constant flux infiltration simulation.

Inputs (please refer to the attached project).

Units: cm and days

Total simulation time: 10 days

Soil Profile: 100 cm deep

Soil Hydraulic Properties (SHPs): Loamy Sand (from Hydrus Database)

Boundary conditions: upper BC constant flux (-0.5 cm/day) and lower BC free drainage.

initial condition: uniform pressure head (-300cm)

We used Hydrus-1D GUI to create the reference project. The project is located in "C:\"

Our goal is to evaluate the effect of the van Genuchten (1980) parameter alpha on the soil water profiles after 10 days (end of simulation). For this reason, we add normally distributed noise to the loamy sand alpha parameter and run Hydrus for all the realizations.

For this project the H1D_CALC.exe file should be located in "C:\" and our working directory is "C:\Simulations". Moreover, for this simple project only two input files are mandatory. The profile.dat (the same for all realizations) and the selector.in (different for each realization).

Calculate the reference SHPs

```
clear all;clc;
vg_par0=[0.057 0.41 0.124 2.28 350.2 0.5]; %[teta_r, teta_s, alpha,n,Ks,tau]
h=0:.1:300;h=-h;
teta_0=zeros(3001,1);tcond_0=zeros(3001,1);
% call VG function
for i=1:length(h)
    [teta_0(i,1),tcond_0(i,1)] =
vg(h(i),vg_par0(5),vg_par0(4),vg_par0(3),vg_par0(6),vg_par0(2),vg_par0(1));
end
```

This block calculates the SHPs for the reference case. As discussed previously we chose the Loamy Sand type from Hydrus Database. Vg is a simple Matlab function which calculates the water retention and hydraulic conductivity values for a given pressure head value and soil type.

Add noise to the alpha parameter

```
%add noise to alpha parameter
num_sim=100;
alpha=vg_par0(3)+0.02*randn(1,num_sim);
```

This block creates a vector with 100 values of alpha parameters. We assume that the error is normally distributed with a standard deviation of 0.02.

Calculate the SHPs for all realizations

```
teta=zeros(3001,num_sim);tcond=zeros(3001,num_sim);
for i=1:length(alpha)
    for j=1:length(h)
    [teta(j,i),tcond(j,i)] =
vg(h(j),vg_par0(5),vg_par0(4),alpha(i),vg_par0(6),vg_par0(2),vg_par0(1));
    end
end
```

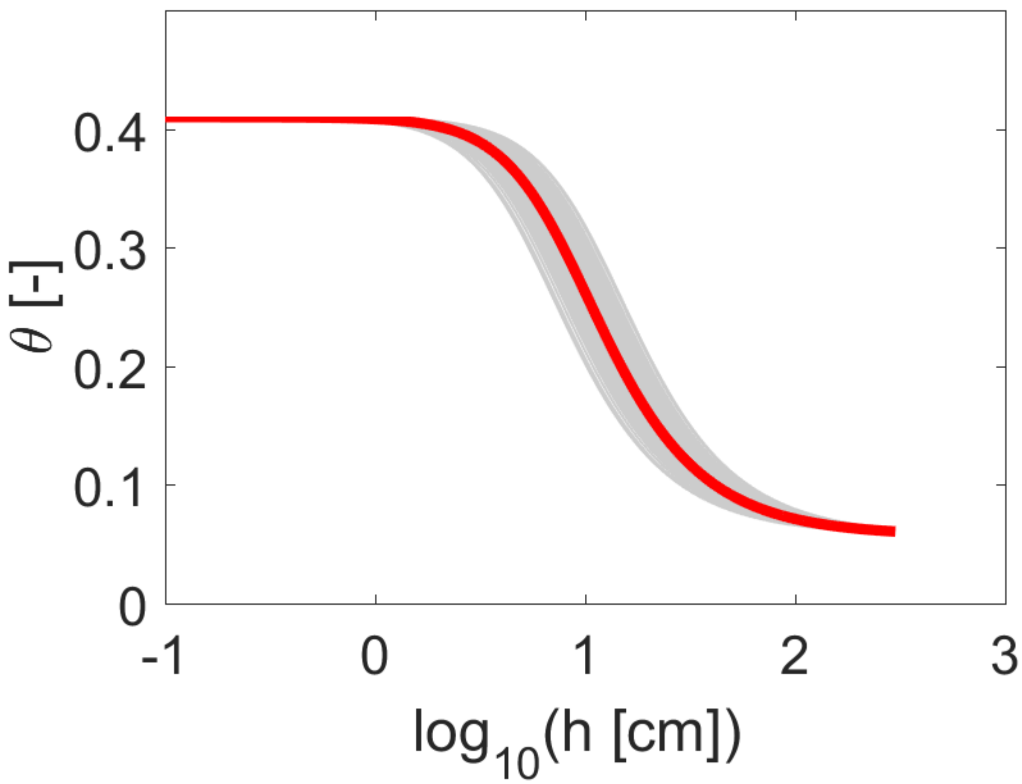
This block calculates the SHPs for all different alpha values (100)

Plot the reference and the generated Water Retention Curves

```
figure(1)
hold on
for i=1:length(alpha)
     plot(log10(abs(h)),teta(:,i),'-','color',[0.8 0.8 0.8],'linewidth',1);
end
plot(log10(abs(h)),teta_0,'-','color','red','linewidth',4);
```

```
xlabel('log_{10}(h [cm]) ');
ylabel('\theta [-]');
set(gca,'fontsize',20);
box on;
```

This block plots the reference Water Retention Curve (WRC, red) and the WRC for all 100 realizations (grey).



Initialize files and directories

```
hydrus_exec='C:\H1D_calc.exe';
hydrus_ref='C:\test_ref';
profileDAT='C:\test_ref\profile.dat';
selectorIN='C:\test_ref\selector.in';
work_dir='C:\Simulations\';
mkdir(work_dir);
```

Create directories and input files

```
path=cell(1,num_sim);
for i=1:num_sim;

    %create folder for each run
    path{i}=strcat(work_dir,'run_',num2str(i));
    mkdir(path{i});
    %copy profileDAt from reference directory to the simulation directory
    copyfile(profileDAT,path{i});
    %manipulate Selector.in for each run
    fileID_out=fopen(strcat(path{i},'\selector.in'),'wt');
    fileID_in=fopen(selectorIN);
    skip_lines=28;
   for k=1:(skip_lines);
         x=fgetl(fileID_in);
         fprintf(fileID_out,'%s\n',x);
   end;
    out_par=vg_par0;out_par(3)=alpha(i);
    fprintf(fileID_out,'%f %f %f %f %f %f\n',out_par');
    fgetl(fileID_in);
    skip_lines_end=10;
   for k=1:(skip_lines_end);
         x=fgetl(fileID_in);
         fprintf(fileID_out,'%s\n',x);
   end;
    fclose('all');
end
```

This block is doing the work. Creates the input files for all realizations. Copies the mandatory file Profile.dat to the corresponding folder and creates the corresponding Selector.in. The Selector.in file is different for each realization since the alpha parameter is variable. The code reads the reference Selector.in, copies the first 28 lines to the new file, write the van Genuchten parameters to the 29th line, and finally copies the last 10 lines from the reference Selector.in to the new file.

Run Hydrus

```
myCluster = parcluster('local');
myCluster.NumWorkers = 8;  % 'Modified' property now TRUE
saveProfile(myCluster);
parpool('local',8);

parfor i=1:num_sim;
   exec_path=[hydrus_exec ' ' path{i}];
   [x, y]=dos(exec_path);
end
```

```
Starting parallel pool (parpool) using the 'local' profile ... connected to 8 workers.
```

Prepare Matlab for using parallel for (parfor). In this example we use 8 precessors/workers. The last lines of this block shows how to run Hydrus for each realization. The parfor loop can be replaced by a for loop for a single core calculation.

Read Hydrus output files

```
prof_out=zeros(101,num_sim+1);
for i=1:num_sim;
   fileID_out=fopen(strcat(path{i},'\nod_inf.out'));
   skip_lines=123;
  for k=1:(skip_lines);
       x=fgetl(fileID_out);
  end;
   temp=fscanf(fileID_out,'%f',[11,101])';
   prof_out(:,i+1)=temp(:,4);
   fclose(fileID_out);
end
   prof_out(:,1)=temp(:,2);
```

This block reads for all the realizations of the Nod_inf.out file and stores the water content profiles for t=10 days in Prof_out table.

Read reference solution

```
fileID_out=fopen('C:\test_ref\\nod_inf.out');
skip_lines=123;
for k=1:(skip_lines);
    x=fgetl(fileID_out);
end;
prof_out_ref=fscanf(fileID_out,'%f',[11,101])';
fclose(fileID_out);
```

Plot simulated water content profiles

```
figure(2)
hold on
for i=1:num_sim
    plot(prof_out(:,i+1),prof_out(:,1),'-','color',[0.8 0.8 0.8],'linewidth',1);
end
plot(prof_out_ref(:,4),prof_out_ref(:,2),'-','color','red','linewidth',4);
ylabel('depth [cm]');
xlabel('\theta [-]');
set(gca,'fontsize',20);
box on;
```

The final block plots the calculated water content profiles for *t*=10 days for all the realizations, including the reference one.