A brief guide to openLISEM

Contents

[Introduction 1](#_Toc376415468)

[Installing openLISEM 3](#_Toc376415469)

[The database 3](#_Toc376415470)

[The interface 4](#_Toc376415471)

[Main page 5](#_Toc376415472)

[Map page 7](#_Toc376415473)

[Simulation page 8](#_Toc376415474)

[Display page 9](#_Toc376415475)

[Creating a database with PCRaster 10](#_Toc376415476)

[Tips and tricks 12](#_Toc376415477)

[Importing maps 13](#_Toc376415478)

[Appendix – PCRaster database script 14](#_Toc376415479)

## Introduction

openLISEM is a raster model that simulates the surface water and sediment balance for every gridcell. It is event based and has great spatial and temporal detail. The model is designed to simulate the effects of detailed land use changes or conservation measures on runoff, flooding and erosion during, and immediately after, **rainfall events**. It is a model designed to be used in disaster risk management, not for long term hydrological estimates.

The event based nature has two major implications:

* The model does not simulate ‘slower’ processes such as evapotranspiration, groundwater flow or changes in vegetation because of crop growth, and that means you have to be careful about the initialial values you use in the model (they have to reflect a certain state the catchment is in, like summer-dry, or spring-wet). This is the responsibility of the user.
* The model can handle any size catchment (the largest is several hundred km2 so far), but the grid cell size has to be smaller than 1 ha in order for certain empirical assumptions to be correct. Large areas also need correct input data such as spatially varying rainfall. openLISEM will run with practically any dataset you throw at it, but that is **not a guarantee** for good, logical results. A model performs according to what you put in it. openLISEM only does technical checking of input data, not whether input data are theoretically correct.

*In other words: you have to know what you are doing!*

In the flow chart below you see the basic structure of the hydrological part of openLISEM: most of the water balance processes are calculated for the gridcell itself (hydrological processes in 1D). Spatial processes are the *hydraulic* processes runoff and channel flow (kinematic wave in 1D) and shallow flooding from the channel system (Saintvenant equations in 2D). The addition of flooding makes openLISEM into a combined 1D/2D model. openLISEM is originally an erosionmodel, the sediment related processes are not shown in figure 1.

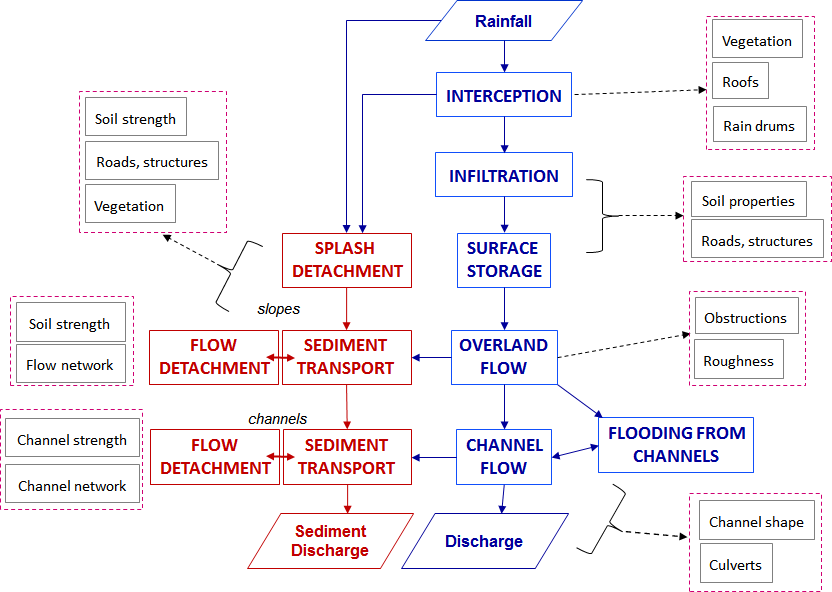


Figure 1. Simplified flowchart of openLISEM with water processes (blue), sediment processes (red) and the main input variables needed as input maps (black).

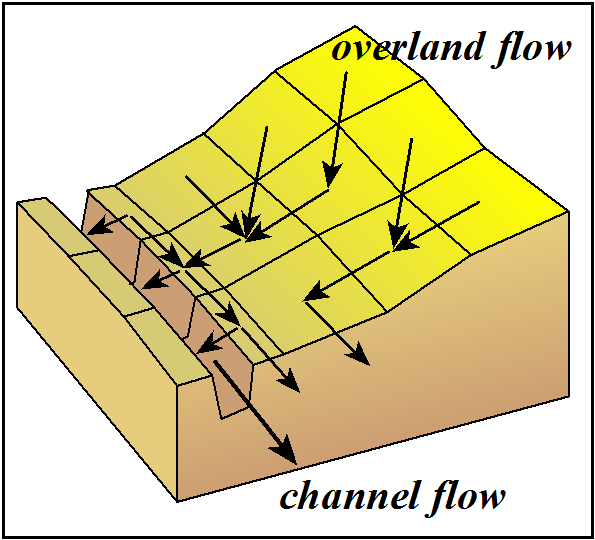


Figure 2. Overland flow (runoff) and channel flow are done with a kinematic wave following a predefined network (usually steepest slope and digitized channel maps).

## Installing openLISEM

openLISEM is a standalone program consisting of an executable and a few libraries that are related to the interface. Note that *openLISEM.ini* is a text file remembering the run files you have used sofar and will be generated when it doesn’t exist yet.

You just unpack the zip in any directory of your choice:

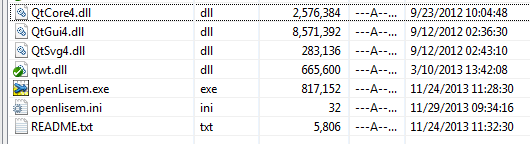


Figure 3. openLISEM executable, no install is needed, just unpack it in a directory of choice.

## The database

openLISEM needs many raster GIS maps and a rainfall text file. Currently the map format used is from PCRaster a freeware GIS and modelling language (http://pcraster.geo.uu.nl/). In the future more formats will be accepted. Also the output of openLISEM consists of PCRaster maps and tables. Simple conversion are available to ASCII maps, which can then be important in other GIS systems (ARCGIS, Ilwis). Therefore PCRaster has to be installed from the website in order to run openLISEM. With PCRaster is a separate development for a windows interface, which is called NutShell. You can find it here: <https://sourceforge.net/projects/nutshellqt/>. Install it after you have installed PCRaster.

PCRaster is not only a GIS but also a spatial modelling language. It can use “scripts” (macros) to create maps with GIS commands. A script exists for openLISEM to create a complete database from a few basic maps and tables. The minimum basic data that is needed to generate an input database for openLISEM is:

* a DEM: used for elevation, slope and creating a flow network using the steepest slope
* a landuse map: used for surface properties that effect surface hydrology and runoff (vegetation properties, surface roughness and resistance, soil strength if erosion is switched on)
* a soil map: used for soil hydrological properties, soil strength and grain size for sedimentation
* tables with soil physical properties (for infiltration and erosion), surface properties (usually related to land use)and vegetation properties
* optional: a series of channel maps specifying riverbed or channel properties and dimensions
* optional: a road network map
* optional: housing densities, roof storage and rainwater storage
* optional: soil conservation measure maps (grass strips, buffers, sediment traps).

**openLISEM has often been criticized for the many maps it needs! Though luck! You cannot analyze complex situations with simplistic datasets and hope to get a reliable answer.** For instance if you want to analyze flooding, you have to tell openLISEM how large and deep your channels are, what shape they have and where possible culverts are. This needs 6 different maps (width, depth, resistance, bottom slope, sidewall slope, culvert flow capacity). So often a single feature like channels need a series of maps to define it.

Fig. 4 shows how openLISEM deals with sub gridcell information. You can add layers with objects smaller than a gridcell, which are then defined as a fraction (buildings and vegetation) or by their size (roads and channels). Note that roads do not act as channels, do not guide the water along the road, only if you make sure the flow direction is also along the road. Roads only influence infiltration (impermeable), runoff (smooth), and there is no interception and sediment detachment but there can be deposition.

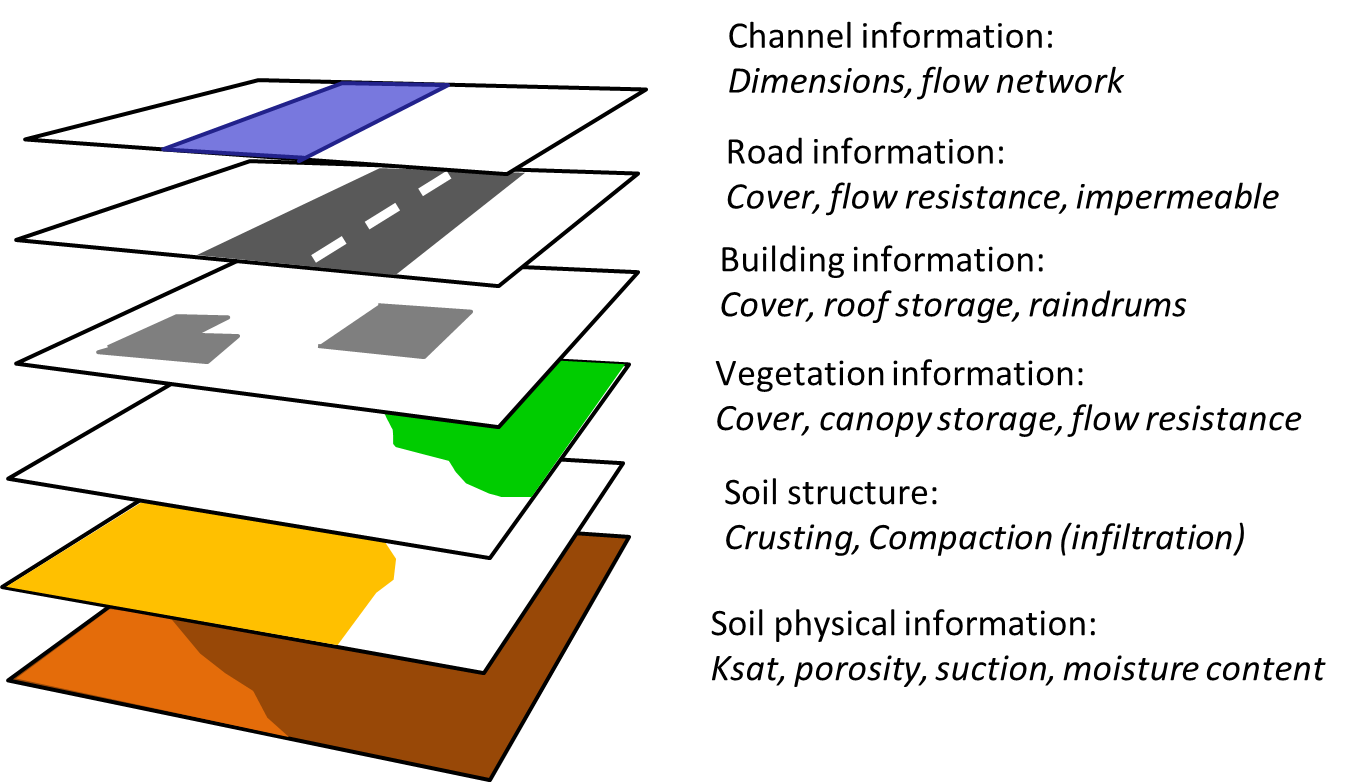


Figure 4. Different information layers are combined into one set of information per gridcell. Vegetation and building information is given as a fraction per cell, roads and channels are given as width in m. The soil layer is the base layer so that we always know wat for instance the infiltration beside a road is.

## 

## The interface

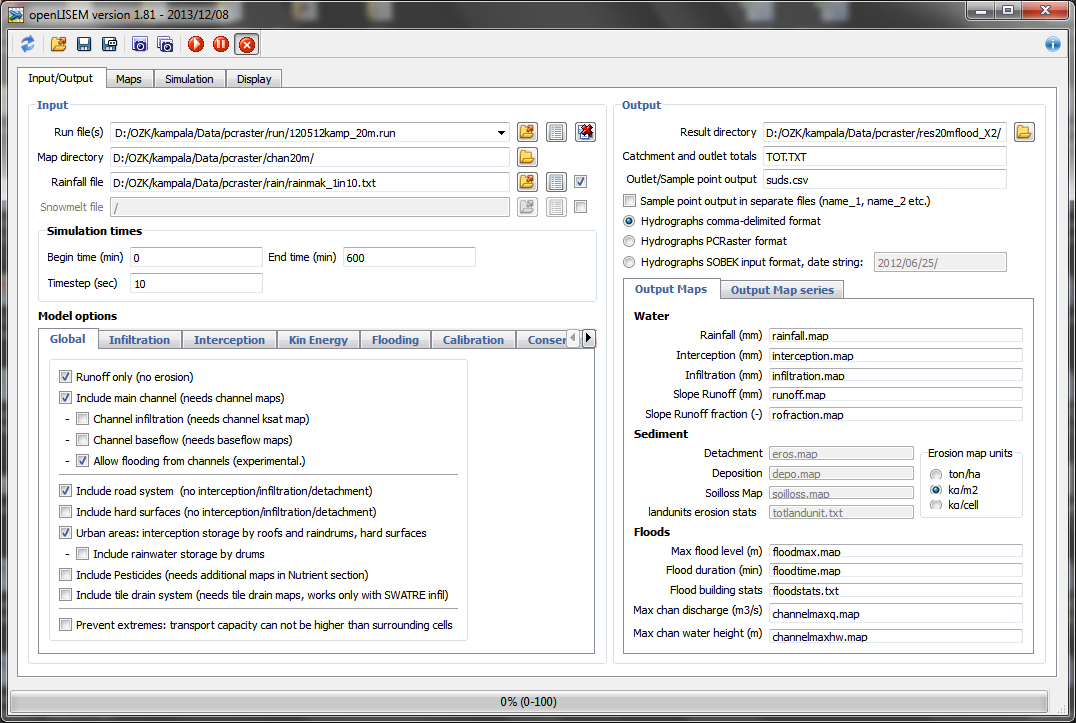
openLISEM is a research model with a complex interface: there are many options that switch on or off certain processes in the model , or offer a selection of algorithms for a single process. There are for instance different ways to calculate infiltration, according to the amount of data you have or the theory you feel is the most appropriate for your problem.

All options in openLISEM are saved in a **run file**. A run file is a text file exactly specifying how openLISEM should be run. We normally work with run files so you don’t have to type in all map names and options each time you run openLISEM.

If you don’t have a run file you can generate one by choosing the main options (see below) and press save, which will save it the first time.

### Main page

The main interface looks like this (greyed out fields depend on options chosen):



***1***

***3***

***2***

***4***

***5***

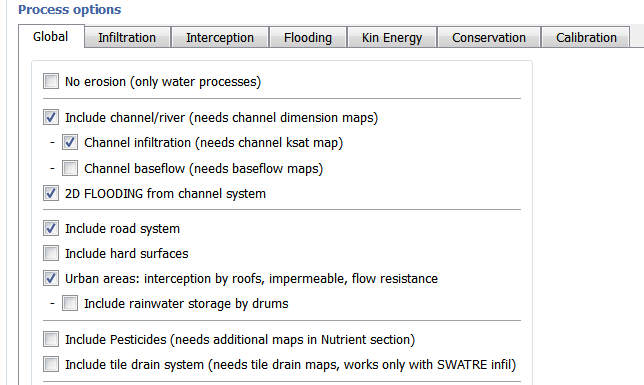
1. The top left section gives the run file name, the input directory to the maps and to the rainfall file. The rainfall can be in a separate directory. All run files are stored in a pull down list, the red cross deletes run files from this list. Generally the database structure used is:

\areaname\maps 🡨 directory with all maps (and a few tables depending on the options)  
\areaname\rain 🡨 directory with rainfall event files

\areaname\run 🡨 directory with collection of run files for different scenarios  
\areaname\res1 🡨result directories, depending on scenarios  
Also the run time is defined here: begin and end time (in minutes) and the timestep (in seconds).

1. The top right section specifies the names of the **result directories** and the main output files. In the hydrograph/sedigraph file is the output per timestep given for the outlet and all other points that are selected for output. These are specified in the map *outpoint.map*. The outlet has value 1, each subsequent point has value 2,3,4 …n. If “Sample point output in separate files” is chosen an output is generated for each point in the map *outpoint.map*, else the values on the output points are given in columns in one file. The output can be plain text, values separated by spaces, comma delimited output for excel, and output in PCRaster table format. The columns in these files are “time, rainfall, discharge, channel water height, sediment discharge and concentration”.   
   If the SOBEK output format is chosen only discharge is saved in a format that the flood model SOBEK can read.
2. A series of tabs that have all **model options** in different categories:

* Global: overall options switching on or off certain processes, often requiring extra data layers.



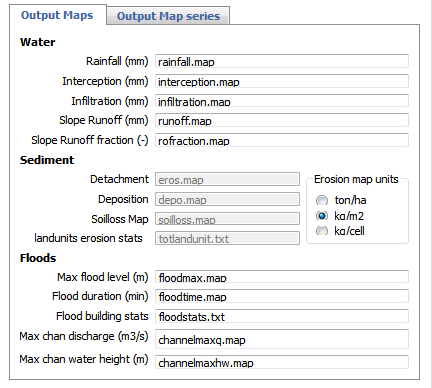
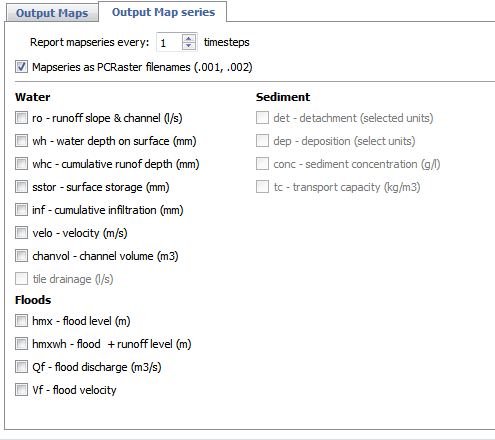
* + **No erosion**: only surface hydrology processes calculated, erosion maps are not needed
  + **Include channel/river system**: overland flow is directed to channels in channel cells and a separate kinematic wave is done for 1D flow in channels. If this option is switched off, all flow is 1D overland flow (kinematic wave). When switched on additional channel maps are needed
  + **Channel infiltration**: water infiltrates in the channel bed, saturated flow is assumed using the ksat of the channel.
  + **Channel baseflow**: not implemented yet, needs assumptions for groundwater flow
  + **2D FLOODING from channel system**: switches on flooding from channels when they overflow. Without flooding all water is assumed to stay in the channel (infinite channel depth). With flooding the channel depth map is needed.   
    NOTE that flooding needs many internal maps and will need use substantial amount of internal memory.  
    NOTE that sediment processes are not yet included in the flooding, so this works only when the "no erosion" option is used.
  + **Include road system**: needs a road width map (in m). Used for tarred roads, assumed smooth, impermeable and no interception. Also no splash and flow detachment, but deposition can take place. The road can be smaller than the gridcell, the information for the roadsides is collected from the soil/land unit maps.
  + **Include hard surfaces**: identical to roads but for used parking lots and so on (map has fraction of hard surface).
  + **Urban areas**: include building structures (map with housing density in fraction of gridcell). Influences interception by roofs, and Manning's n in overland flow and flooding. Houses are impermeable and non-erosive. Building information is also used to produce a flood statistics map.
  + **Include rainwater storage by drums**: water from roofs is assumed to fill up a storage tank first. Map has the tank volume in m3. Works only if there is a building in the same pixel.
  + **Include pesticides**: experimental code to calculate pesticide content in runoff and sediment losses.
  + **Include tile drain system**: experimental code calculating flow through underground tile drains, works only in conjunction with the SWATRE infiltration sub-model.
* Infiltration: select an infiltration model (Green and Ampt, Smith and Parlange, ksat subtraction) and configure if there are 1 or 2 layers, compaction, crusting, closed soil profile, or include percolation. Note that SWATRE is a full finite difference solution of the Richards equation and requires a 3D soil structure defined by the user.
* Interception: interception by vegetation canopy with a choice of empirical equations relating canopy storage (in mm) to LAI (in m2/m2). Choose the predominant vegetation. In case of complex vegetation the user has to supply a map (smax.map) with canopy storage in mm.
* Kin. Energy: equations to calculate kinetic energy for splash detachment from rainfall information.
* Flooding: technical flood parameters related to the mathematical solution of the flood equations based on the FullSWOF2D open source code). Experimental : do not touch if you don’t know what you are doing!
* Calibration: multiplications factors for the most common calibration variables: Ksat, moisture content, roughness. 1.0 means that the original values in the maps are used, 0.5 means the map is multiplied by 0.5.   
  NOTE that multiplying the Ksat by 0.5 means there is less infiltration and more runoff, multiplying the moisture content by 0.5 means the soil is dryer and will generate *more* infiltration and *less* runoff.
* Conservation: configuration of water and sediment buffers, sediment traps and grass strips (these options need additional maps).

1. Lower right is a selection of output maps and output map series (see below). The outputs are divided in water, sediment and flood related output. The water maps are totals in mm:

* Rainfall: sum of all rainfall per gridcell. Useful when you use spatially variable rainfall maps instead of tables;
* Interception: total interception per gridcell by vegetation and buildings (mm);
* Infiltration: total infiltration per gridcell (mm);
* Runoff: total runoff in mm per gridcell. This map can have high values where runoff concentrates in flow lines;
* Runoff Fraction: calculated per gridcell as (runoff out – runoff in)/rainfall. Positive values indicate more water is generated on the cell than is received from upstream (source), negative values indicate that less water is generated and water from upstream infiltrates(sinks). This map can display high values in flow concentration lines, the map is best viewed by scaling between -2.0 and +2.0 for instance.

The second tab has output map time series: these have standard names and are saved every timestep in PCRaster format. For instance the runoff maps (in l/s) are stored as ro000000.001, ro000000.002 etc. for timestep 1 and timestep 2. If you do not want to store every map you can increase the interval number. For instance if your simulation timestep is 20 seconds and the you set the counter to 15, the interval in which maps are saved is 5 minutes (15x20 sec is 300 sec = 5 min). The maps are still called ro00000.001, 002 etc., conform PCRaster conventions. Options are:

* Runoff (ro) : overland flow and channel flow discharge in l/s;
* Water height at the surface (wh) in mm;
* Cumulative runoff depth (whc) in mm;
* Surface storage in micro depressions (sstor) in mm;
* Cumulative Infiltration (inf) in mm;
* Runoff velocity (velo) in m/s;
* Channel water volume (chanvol) in m3;
* Tile drainage (tile) in l/s;
* Flood water level (hmx) in m;
* Flood water level and runoff level (hmxwh) in m;
* Flood discharge (average in x and y direction, Qf) in m3/s;
* Flood velocity (average in x and y direction, Vf) in m/s;
* Detachment, sum of splash and overland flow and channel flow detachment (DET), units to be selected;
* Deposition by overland and channel flow (DEP), units to be selected;
* Concentration of sediment in flow (conc) in kg/m3;
* Transport Capacity of flow (tc) in kg/m3;



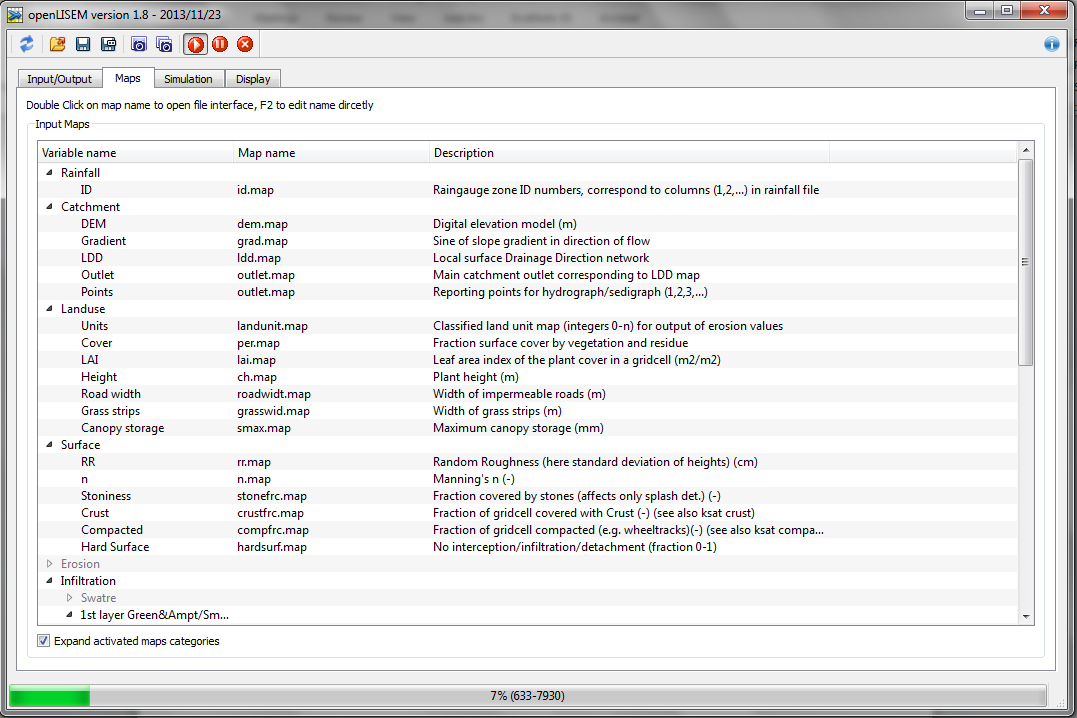
1. Speed buttons to load and save run files, make single screenshots or screenshots of every timestep and run, pause, stop the simulation. If you choose multiple screenshots, these will be taken form the Simulation page window and the Display window. They are saved as .png files in the result directory and can be used with windows moviemaker for instance to make a demonstration movie.

### Map page

The second tab in the interface is the map page. This gives a list of all input maps with a one line description and units in which they have to be supplied. that are used (not used are grey). Normally you do anything in the page, because this information is in the run file. It is easiest to use default names, although some of them are historical and not very logical (e.g. per.map comes from vegetation cover percentage but is a map with vegetation cover fraction).

It is easiest if you use the default map names but you can change these names to your own liking. Double click on the map name in the middle column and a file window will open pointing to the map directory.

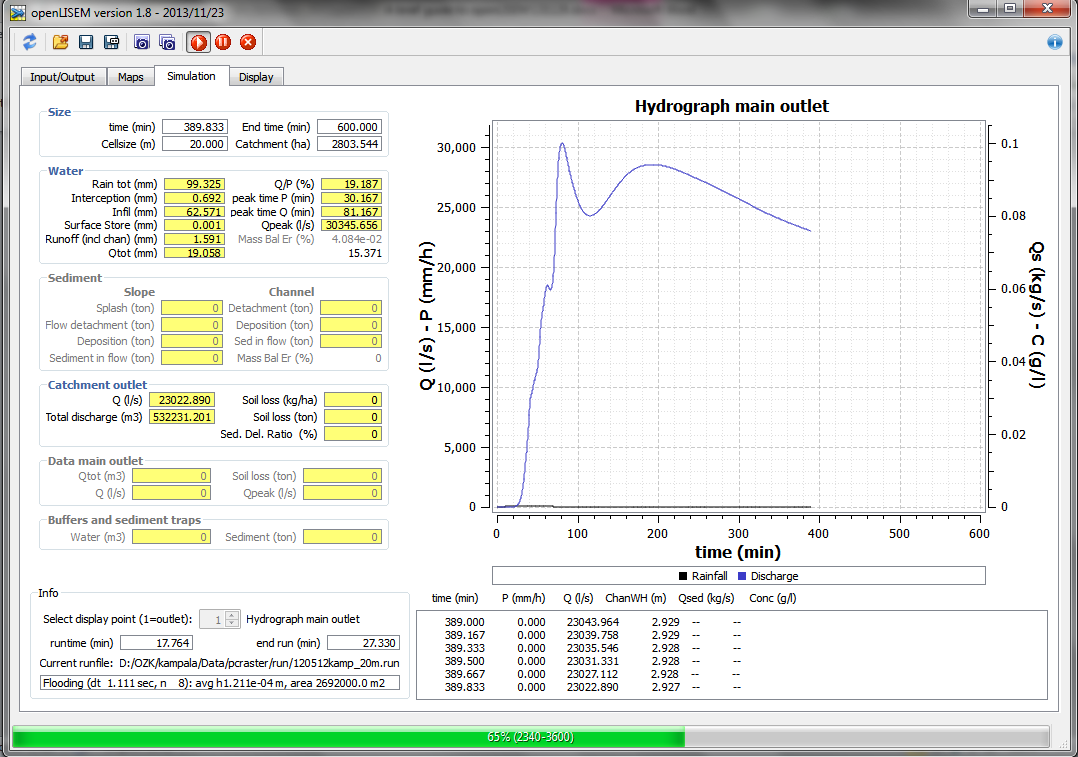
IMPORTANT: when openLISEM reads the map database it does some preliminary checking. The base map against which all maps are checked is the “Local drain direction” or **ldd.map**. PCRaster has a command to generate such a map form the DEM. This is the user defined flow network that connects all cells. If openLISEM cannot find a values of input maps where a network is defined it will stop with a warning. This may happen if the aps have different origins, for instance a vegetation map may be derived from a satellite image, the soil map is digitized and the DEM comes from interpolation of elevation lines. It is the responsibility of the user to make sure all maps have the same size and coordinates. Maps can be bigger that he network map ldd.map, the area outside the network is ignored. But they cannot be smaller.



### Simulation page

This page shows the hydrological and erosion key values and graphs. The following info can be found:

1. “Size”: catchment size and run time values
2. “Water”: catchment totals in mm (left hand side) where the sum of all the processes should equal the rainfall (and or snowmelt) at any given moment. On the right hand side are some values needed to compare different scenarios: average catchment runoff fraction (Q/P), peak times of rainfall and discharge (the difference is the response time of the catchment), and peak discharge.
3. “Sediment”: sediment detachment and deposition totals on slopes and in channels (greyed out in this example because erosion processes are not calculated) .
4. Catchment outlet actual and total discharge and soil loss
5. Additional catchment discharge information at a point selected by the user. The user can define many points to output information in a separate map (outpoint.map) with values 0 (background), 1 (main outlet), 2-n (additional output points). In the info box (6) the point can be selected that is shown. This can only be done before the run (not during a run). Note that the total catchment water and sediment values in boxes 1 to 4 are always shown, while the graph and the info in box 5 change according to the output point selected.
6. Additional information about the run, warnings, some technical information for flooding (needed by the developers).
7. Rainfall, hydrograph and sedigraph of the outlet or the chosen outputpoint. The rainfall scales with the discharge (left axis) so it might become less visible. The row of values below the graph are the last 6 timesteps.



***7***

***6***

***5***

***4***

***3***

***2***

***1***

### Display page

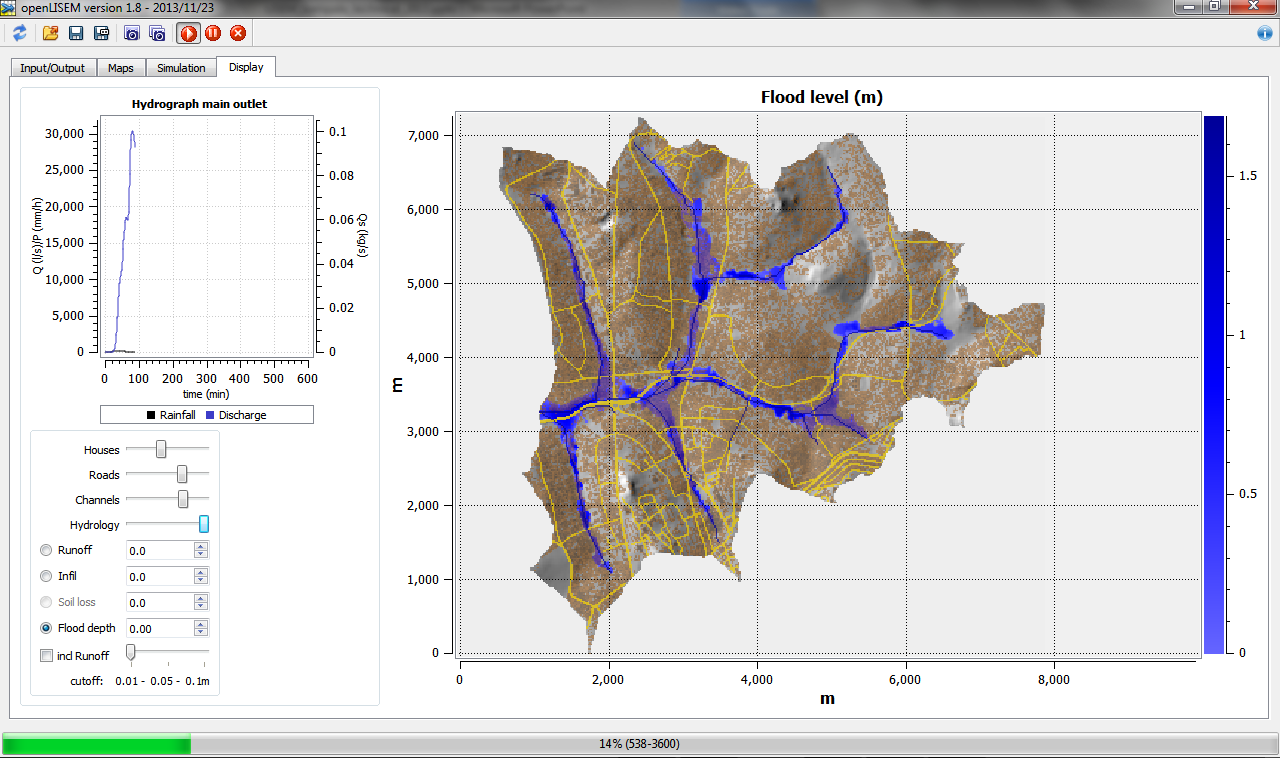
This page shows spatial information of the current timestep shown over he shaded relief map of the catchment. NOTE: you can only see the CURRENT timestep. Information is not saved to go back to earlier timesteps, this would cost too much memory (openLISEM is not a GIS).

There are four dynamic information layers you can show: runoff (l/s), cumulative infiltration (mm), soil loss (ton/cell) and flood level (m). Fixed information shown are the housing densities, roads and channels.

The sliders on the right hand side let show the housing density, roads and channels and the hydrology with different levels of transparency (this does not influence the simulation). The spinboxes allow you to set the maximum values in the display and legend of the hydrological information, so that you can optimize the display.

The flood display can be shown with an additional minimum cutoff value so that only areas with a water level of 0.01, 0.05 or 0.1 m of water level are shown.

NOTE: You can zoom in and out in the map window with the mouse wheel and the value at the cursor is the cell value of the current hydrological information.



## Creating a database with PCRaster

In this section an example is given for a script that generates a standard database for openLISEM. Note that your situation can be different so it is just an example. All map layers are based on a DEM, classified soil map (based on texture classes), a classified land use map and a channel and road map.

A script is given in the appendix. All text preceded by a hash “#” is a comment. The comments help you to understand the script.

Assuming you have a project called “flood”, you canake in windows a directory structure as follows:

\flood\maps

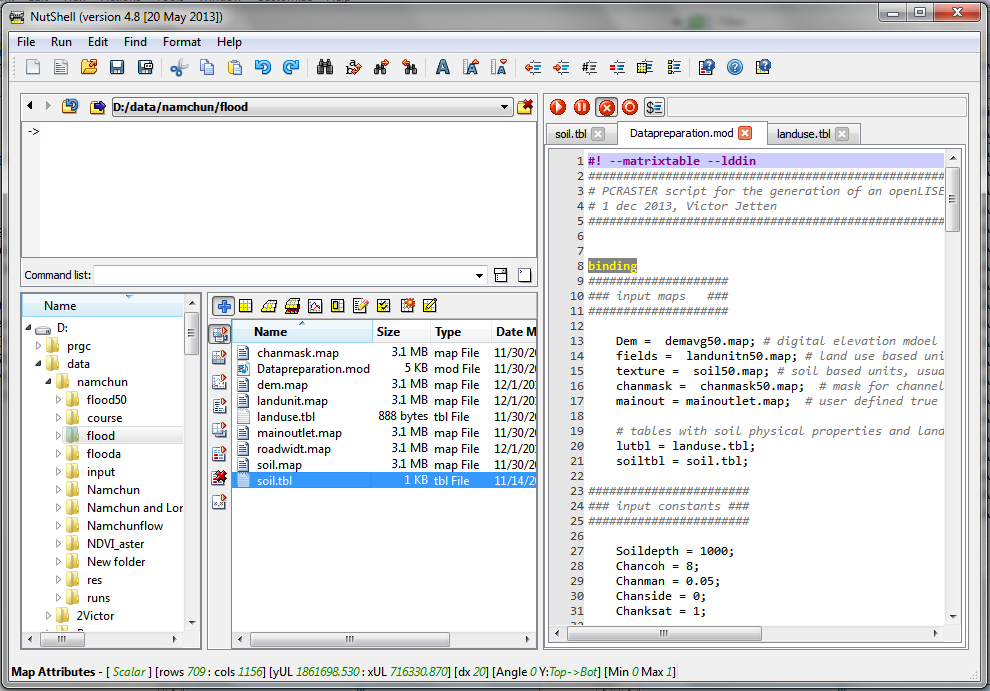
\flood\rain

\flood\run  
\flood\res

Make sure in \flood\maps you have the basic PCRaster maps:   
*dem.map  
landuse.map  
texture.map  
roadmask.map  
channelmask.map*tables:  
*soildata.tbl  
landusedata.tbl*

Note that you can use additional maps such as *mainoutlet.map* which is a map with the true outlet position. Sometimes the digitized channel is not in the location of the lowest DEM position. The script forces the surface runoff to end at the channel and the channel to end at the outlet.

Assuming you have installed PCRaster and NutShell, you see the following screen if you open NutShell:



Go to the directory with your maps and press the icon to set the PCRaster work directory to that folder. Load the script “datapreparation.mod” in the editor on the right by double clicking on it. Then press the red arrow button to run the script. If you double click on a map, it will be opened and shown. A full explanation of PCRaster and Nutshell goes beyond this manual.

## Tips and tricks

This section assumes that you have run a script that makes a lisem database but there are some of the underlying things still to be done.

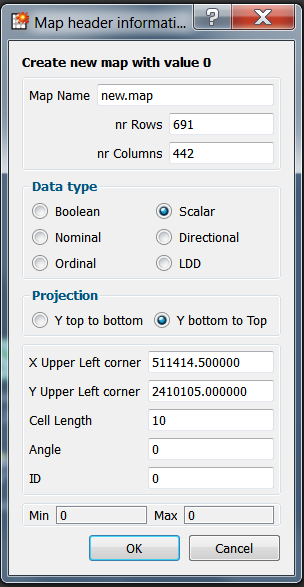
1. If your soils are very homogeneous and the land use determines the infiltration, for instance because of tillage operations, make sure you link variables like Ksat to the land use instead of to the soil.
2. It is up to you to choose correct combinations of values. A fully grown Maize field can have a high cover fraction, but runoff can easily flow between the rows of stalks, so the Manning’s n value is generally low. A grass land has a high dense cover, and also a high Manning’s n value because the grass obstructs the flow. openLISEM makes no assumptions, and simply tries to execute what you put in it.
3. You can further influence infiltration by assuming cells that have compacted or crusted to a certain degree with a compacted or crusted fraction (crustfrc.map, compfrc.map) and a low Ksat (ksarcrst.map, ksatcomp.map). For instance your village may have the same soil type but the soil might be compacted.
4. Note that the channel can only be one cell wide, if you have a mask with channel parts that two cells wide, PCRaster will create a one channel network with small side branches. To remove these small side branches do the following:
   1. make an accumulated channel networks map with the command:  
      pcrcalc --lddin chups.map=accuflux(lddchan.map,1e20,1e20,1e20,1e20)
   2. remove all cells that have a value “1” from this map (this will also make your channel network 1cell shorter:  
      pcrcalc chanmaskn.map=if(chups.map gt 1,scalar(1), 0)
   3. use the new chanmaskn.map in the script and run again.
5. Roads must be given as a separate layer because they can be smaller than a gridcell, but they are often part of a landuse map. In other words you may have a landuse map with unit 20 which is roads but this means this is a 30m wide gridcell with a road inside of 12m wide. Lisem has to know which land use type is *beside* the road, so the road has infiltrating sides. You can solve this in the following way:
   1. Make a scalar roadmask map:  
      pcrcalc roads.map = if(lu.map eq 20,scalar(1), 0)
   2. Assuming the land use map is a nominal (classified) map with units larger than zero, and the road has unit 20. First make a scalar landuse map with the roads set to zero:  
      pcrcalc lus.map=scalar(if(lu.map eq 20,0,lu.map))
   3. Next fill the cells with zero with the maximum of the surrounding cell in a window:  
      pcrcalc lus.map=if(lus.map eq 0, windowmaximum(lus.map, 3\*celllength()),lus.map)
   4. Use the lus.map and roads.map in your script and run again.

## Importing maps

If you have these maps in another system export these as ASCII maps and import them. If they are exported as a matrix of values (with or without a header) you can use the command:

**asc2map -- clone mask.map dem.asc dem.map -a**

Note: “-a” is for ArcGIS ASCII export maps only. The option “--clone mask.map” uses an empty map in PCRaster which you have to make first.

Making a mask map is done in NutShell by pressing in NutShell which opens a window where (see left) where you can fill in the number of rows and columns, upper left coordinates and cell length. The ID is the value that the new map will receive.

ArcMAP ASCII maps have a header that has the *lower left* coordinates while PCRaster uses the *upper left* coordinates (of the UL corner of the cell). So if you have 100 rows and 100 cols of 10m cells, and coordinates in ArcMAP are LLx = 30000 and LLy = 120000, then the PCRaster coordinates are ULx = 30000 and ULy = 120000 + 100\*10 = 121000.

If the map Y coordinate *increases* from top to bottom (certain projections) than the ULy = 120000 – 1000 = 119000.

Note that the cell length is just a header value, changing it will not resize or resample the map. PCRaster has no concept of cell length units in m or cm or km, it is just a value. NOTE THAT OPENLISEM HOWEVER ALWAYS ASSUMES THE CELL LENGTH IS IN METERS!

Export is exactly the same but the reverse operation, using **map2asc** as a command. This produces an ASCII map for import in for instance arcmap.

If your exported map is stored in a 3 column format with x, y and the value, the command is:

**col2map --clone mask.map dem.col dem.map**

You type these commands in the upper left window of NutShell at the “->” prompt. If you type in asc2map or col2map without parameters you get the syntax of thee commands.

MORE CASN BE FOUND ON IMPORT?EXPORT AT **http://www.gdal.org/**.GDAL recognizes PCRaster maps.

## Appendix – PCRaster example database script for openLISEM

#! --matrixtable --lddin

##########################################################################

# PCRASTER script for the generation of an openLISEM input database #

# 1 dec 2013, Victor Jetten #

##########################################################################

binding

####################

### input maps ###

####################

Dem = demavg50.map; # digital elevation mdoel

fields = landunitn50.map; # land use based units

texture = soil50.map; # soil based units, usualyy texture classes

chanmask = chanmask50.map; # mask for channel maps

mainout = mainoutlet.map; # user defined true outlet (1 =voutlet, 0 is rest)

# tables with soil physical properties and land use properties

lutbl = landuse.tbl;

soiltbl = soil.tbl;

#######################

### input constants ###

#######################

Soildepth = 1000;

Chancoh = 8;

Chanman = 0.05;

Chanside = 0;

Chanksat = 1;

#####################

### output maps ###

#####################

# basic topography related maps

Ldd = ldd.map; # Local Drain Direction

grad = grad.map; # sine of slope (not tangent)

id = id.map; # pluviograph influence zones (1,2, ...n)

# if there are more classes the rainfall file needs more columns

outlet = outlet.map; # location of main outlet

outpoint = outpoint.map; # location of additional information points, 1 (outlet) 2-n

# vegetation/crop maps

lu = landunit.map; # for output stats

coverc= per.map; # vegetation cover fraction

lai= lai.map; # leaf area index (m2/m2) for interception storage

cropheight= ch.map; # vegetation/crop height for splash detachment energy

grass = grasswid.map; # grass strip width (m)

# soil maps, "1" stands for layer 1

ksat= ksat1.map; # Saturated hydraulic conductivity (mm/s)

psi= psi1.map; # matrix suction at the wetting front (cm)

pore= thetas1.map; # porosity (-)

thetai= thetai1.map; # initial moisture content (-)

soildep= soildep1.map; # soil depth in mm

# surface maps

rr= rr.map; # random roughness (cm) for surface storge and witdh of flow

mann= n.map; # mannings n, overland flow resistance

stone= stonefrc.map; # stoniness, crusting and compaction fractions

crust= crustfrc.map;

comp= compfrc.map;

# erosion maps

cohsoil = coh.map; # soil shear strength (kPa)

cohplant = cohadd.map; # additional root shear strength (kPa)

D50 = d50.map; # median of texture in micrometers (mu)

aggrstab = aggrstab.map;# aggergate stability, lowe drop test (number of drops)

# channel maps

lddchan = lddchan.map; # channel network

chanwidth = chanwidt.map; # width (m)

changrad = changrad.map; # sine gradient of channel/river bed

chanman = chanman.map; # flow resistance

chanside = chanside.map; # tangent side wall angle, 0 = rechtangular, 1 is 45 degrees

chancoh = chancoh.map; # cohesion (kPa)

chanksat = chanksat.map; # Ksat of channel (mm/h)

chandepth = chandepth.map; # depth (m) only for flooding

chanlevee = chanlevee.map; # height of levees in m

chanmaxq = chanmaxq.map; # maximum discharge of at culvert location in m3/s, 0 means no culverts

barriers = barriers.map; # additional elemants, elevations, dikes, taluts to be added to the DEM (m)

# housing maps

house = housecover.map; # structure cover fraction

drum = drumstore.map; # storage in m3 of water drums

roof = roofstore.map; # roof interception storage in mm

areamap

# MASK

Dem;

initial

######################

### BASE MAPS ###

######################

mask=scalar(Dem/Dem); #make a mask

# make ldd and ensure it flows to channel and to the main outlet

report ldd.map = lddcreate(Dem-chanmask\*2-mainout\*2,1e20,1e20,1e20,1e20);

report outlet = pit(ldd.map);

# sine gradient (-), make sure slope > 0.001

report grad = max(sin(atan(slope(Dem))),0.005);

#########################################

### MAPS WITH RAINFALL INFLUENCE ZONE ###

#########################################

report id = nominal(mask);

#######################

### CROP MAPS ###

#######################

report lu = fields; # copy the landuse to a landunit.map for stats output

# fraction soil cover (including residue), from col 5 of land use table

report coverc = lookupscalar(lutbl, 5, fields);

# crop height (m) from col 7 of land use table

report cropheight = lookupscalar(lutbl, 7, fields)\* mask;#coverc;

# LAI (m2/m2) from cover fraction

report lai = ln(1-coverc)/-0.4;

###########################################################

### INFILTRATION MAPS for option one layer GREEN & AMPT ###

###########################################################

report ksat = lookupscalar(lutbl, 10, fields);

report pore = lookupscalar(lutbl, 11, fields);

report psi = lookupscalar(soiltbl, 5, texture)\* mask;

report thetai = lookupscalar(lutbl, 12, fields);

report soildep = scalar(Soildepth);

#############################

### SOIL SURFACE MAPS ###

#############################

# micro relief, random roughness (=std dev in cm)

report rr = lookupscalar(lutbl, 8, fields);

# Manning's n (-)

report mann = lookupscalar(lutbl, 9, fields);

# profile definition in PROFILE.INP and PROFILE.MAP

report crust = 0 \* mask;

# stone fraction

report stone = 0 \* mask;

#fraction compacted

report comp = 0 \* mask;

######################

### EROSION MAPS ###

######################

#report D50 = 30\*mask;

report D50 = lookupscalar(soiltbl, 8, texture) \*2\* mask;

report cohsoil = lookupscalar(soiltbl, 6, texture);

report cohplant = 0\*cohsoil;

report aggrstab = (1 -lookupscalar(soiltbl, 2, texture))\*100 \* mask;

######################

### CHANNEL MAPS ###

######################

chanmask = chanmask/chanmask;

# channel is 1 and rest missing value

report lddchan = lddcreate(Dem\*chanmask,1e20,1e20,1e20,1e20);

report changrad = max(0.005,sin(atan(slope(chanmask\*Dem))));

report chancoh = chanmask\*scalar(Chancoh);

report chanman = chanmask\*scalar(Chanman);

report chanside = chanmask\*scalar(Chanside);

# width empirical scaled up from 1 to 15 meter

report chanwidth = min(15,max(1,accuflux(lddchan,1)/200));

report chanksat = chanmask\*scalar(Chanksat);

# flood maps

report chandepth = chanmask\*1; #depth set to 1 m everywhere

report chanlevee = mask\*0; # small levee along channel in m

report chanmaxq = mask\*0; # culvert max dlow, 0 for all the rest, if no culverts set to 0

report barriers = mask\*0; # added to dem is needed, like dikes

######################

### house MAPS ###

######################

report house = if(fields eq 9, uniform(1), 0) \* mask;

# random housecover from 0 to 1 in unit 9 settlement.

# Note that 9 has compacted soil physical values

report drum = mask\*0;

report roof = 1\*mask;