terraforming manual

Marco Günther

October 17, 2006

```
                                  *                   *       *
   o                      *           /\_ *                    *
*       o         *                  /    \    \_  /
           o_                 *    /\/  *  \_  \ /      *
 \__/__|_>     *            / /   /\     \  |_/
  oo    o-o-(__________/_/__/__\____\_|  *      *
                                                     ------------
```

Abstract

This document includes ideas, technology and tutorial for terraforming, a grid based generator for digital elevation models (DEM) and related applications like soil- and use tasks. There are many tools for topologic analysis of digital elevation models, so we only deal with topology needed for further genesis like using a rivernetwork for soil derivation. The intention is to create grids with predicted parameters like forcing an outlet or hypsometric curve. To get these results we didn't use erosion models but a kind of randomwalk. Generated or loaded surfaces also can be transformed, to realize a set of digital height models with discrete parameter in an interval.

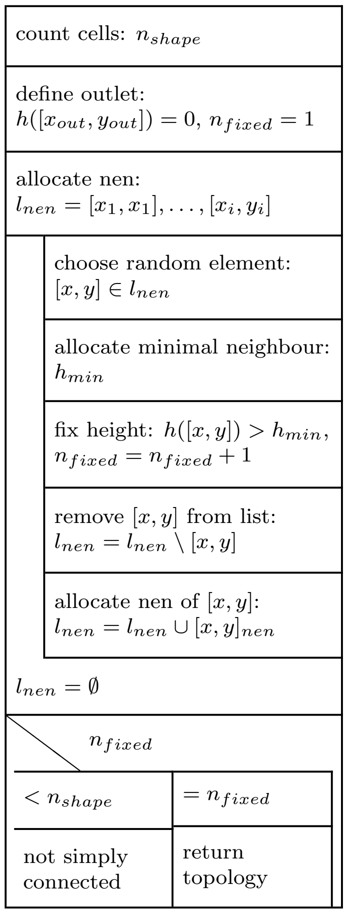
In an automatic mode, arbitrary digital elevation models with parameters in intervalls are generated. WaSiM ETH can be executed within a batch skript to simulate the rainfall runoff process for each DEM. It is also possible to randomize some of the WaSiM parameters and to analyse the effects of random simulation and topography by the DYNIA method.

Contents

1 Deriving Height

1.1 Topological Correctness

Terraforming creates digital elevation models (DEM) without local minima. This topological correctness is achieved by assigning heights to cells based on minimal height of the existing, fixed neighbours: a newly assigned cell must have at least the minimal neighbour height to ensure the DEM-drainage. In general terraforming uses $D4$ or $D8$-methods (drainage structure) for generation and analysis. A DEM created by using $D8$-neighbourhood cannot drain in $D4$ flow analysis.
Height allocation starts with a given outlet at the border of a given grid area (e.g. a set of n_{shape} *ones* are indicating an area in a matrix with otherwise *zero* entries). Temporarily its height is set to zero. In Dx at the maximum $x-1$ ($x = 4, 8$) NEN (next empty neighbours: cells without fixed height but attribute *area*) are written in a list and one is chosen randomly to be the next in line to get its height. After the height is fixed, its coordinates are removed from the list and all new possible NEN are added. The next cell is chosen randomly then for height assignment. If the list runs out of coordinates and $n_{fixed} < n_{shape}$, the n_{shape} cells aren‘t simply connected with respect to Dx.



1.2 Assigning Heights

A simple way to define the height of a chosen empty cell is to increase the minimal needed height by a positive value Δ, e.g. randomly from interval I. To force geomorphological parameters the interval I can be modified, e.g. as a function of distance between two points: outlet x_{out} and temporarily considered gridcoordinate x: $I = I(|x_{out} - x|)$. A grid G storing those values will lead to concentric surfaces with center x_{out}. If we do the randomwalk by using the increase of $G(x)$, it can project characteristics of the modelmatrix G to the

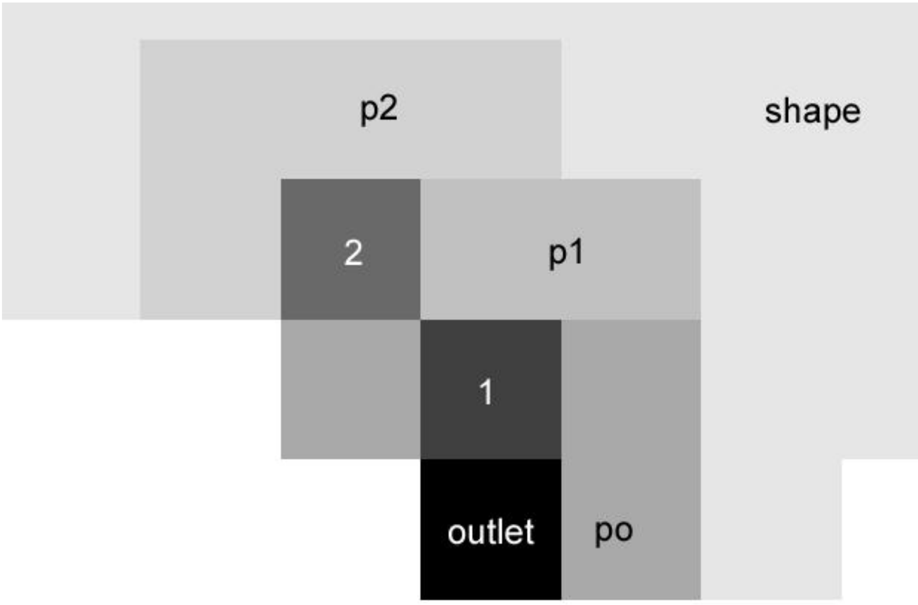


Figure 1: Fixing height D8: *po* (including 1) are the NEN of the outlet. 1 is the first fixed height, choosen from *po*. p1 (including 2) are all new available cells to set the second cell height 2.

deduced dhm. Terraforming uses normalized grids G, wich means that the point farest away from the outlet equals 1: $I(|x_{out} - x_{farest}|) = 1$.

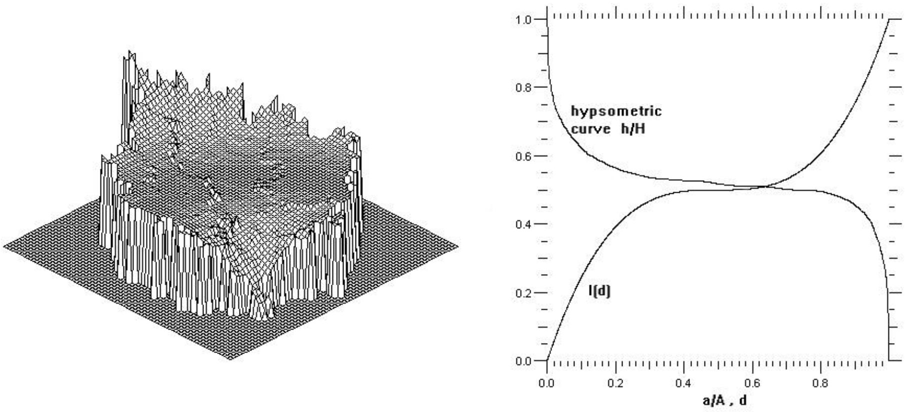


Figure 2: sigmoid dhm generated by normalized $I(d)$

In Figure 2 a uniform value $u \in U = [0.9, 1.1]$ is multiplied by the absolute increase of G, generated by $I(d) = 4(d - \frac{1}{2})^3 + \frac{1}{2}$, in direction of the minimal neighbour, to get a sigmoid result.

Other normalized functions are available ($f(0) = 0, f(1) = 1$):

linear:	x
exponential:	$\frac{e^x-1}{e-1}$
invers exp:	$\ln((e-1)x+1)$
sigmoid:	$4(x-\frac{1}{2})^3+\frac{1}{2}$.

The gradient in a direction of equal height in G equals 0. This leads to the height of the minimal neighbour. Therefore,to avoid flat plains in the constructed DEM, a value of $0.0001h$ is added for topological correctness.

After assigning heights to all cells, the DEM itself is normalized to: $h_{out} = 0, h_{max} = 1$. Because of there is no absolut height yet assigned, only the total range of interval U affects the degree of randomisation in a derived DEM.

This way of randomwalk leads to steep gradients on the surface in a case of two near regions of expansion: Accidental series of high and low values of random additions u_i^l and u_i^h are not compensated for in this model. Assigning height by using the average height of already fixed next neighbours (instead of minimal neighbour height) smoothes the gradient. This could be enhanced by considering more cells than next neighbours.

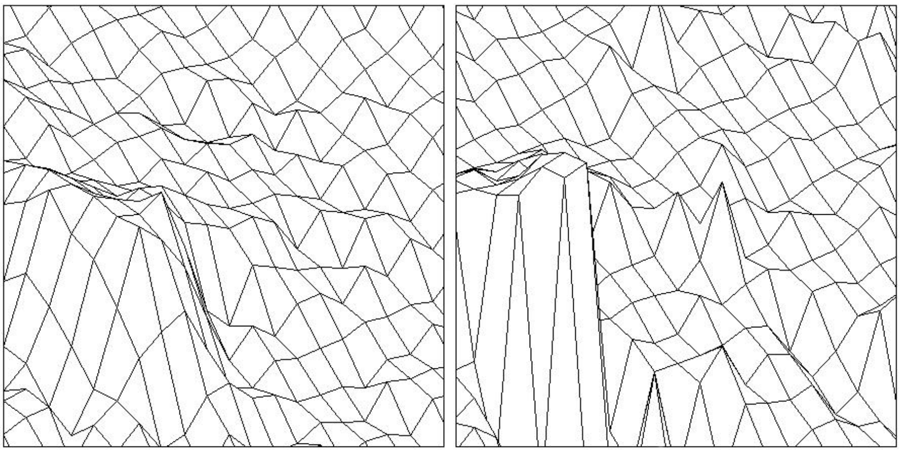


Figure 3: examples of average height and minimal height model

1.3 Scaling and Forcing Hypsometric Curve

The height of the assigned grid can be scaled and rescaled at any time. This is also valid for any externally produced and loaded DEM. Because the tangent is monotone, drainage topology and the hypsometric curve are invariant under this linear transformation of all cells. This contrasts the forcing of the hypsometric curve: Sorting cell coordinates by height results in a monotone series. A mapping of another monotonic series, e.g. a discretisation of $[f(0), f(1)]$ using the

ratio	sigmoid	linear	exp	inverse exp	exp, uniform assumption
R_N	3.65	3.62	3.61	3.64	3.90
R_L	1.55	1.44	1.49	1.46	1.64
R_A	4.15	4.14	4.13	4.17	4.44
n	92	96	90	90	89

Table 1: Horton's laws, 10000 cells, average method

normalized functions mentioned above, to the ordered grid coordinates creates a new DEM with a new hypsometric curve. Normalized series mapped on the grid will return the same discretisation to the hypsometric curve. Scaling back to minimal and maximal height will not affect the curve. This transformation also does not interfere with the topological correctness: the order of height remains constant. Thus a drainage network derived in $D4$ method holds. $D8$ networks vary because of the factor $\sqrt{2}$ for diagonal neighbour distances.

1.4 Horton's Laws

The ratios of stream numbers R_B, lenghts R_L and areas R_A are important characters of the channel network. They are defined by:

$$R_B = \frac{N(\omega)}{N(\omega+1)} \qquad R_L = \frac{L(\omega+1)}{L(\omega)} \qquad R_A = \frac{A(\omega+1)}{A(\omega)}$$

where $N(\omega)$ is the total number of streams, $L(\omega)$ the average lenght an $A(\omega)$ the average total catchment area of order ω. Testing Hortens Laws with spherical shape, random outlet and a minimum number of 15 cells for a stream threshold, leads to the ratios of Table 1 (except the last column). These values are obtained by using the average neighbour method, where n is the number of digital height models draining to order 5.

As one can see in Figure 4 (black), the maximum order stream lenght doesn't fullfill Horton's law. In general the ratios depend on the water gauging station, in particular the lenght of the maximum ordered streams S_i^{max}, while the total contributing area and the numbers of lower ordered streams are estimated at less dependency. Assuming that lenghts of the maximum ordered streams fullfill a uniform distribution, the value of its arithmetic average will be half of the full estimated lenght. Assuming further that the density of streams flowing into those maximum ordered is constant, one can double the number of lower ordered streams in a area $A_{total} \setminus A_{start}$, where A_{total} is the total watersheed and A_{start} is the watersheed associated to the point where the S_i^{max} starts. Doing so and resimulating some digital height models, Horton's law looks like the red values of Figure 4 and the power law changes to the last column of Table 1.

The result brings little advantage for values of the S_i^{max}. In fact the assumption of uniform distribution could be enhanced, if we'd have a look at the relative numbers of digital height models draining to order $max+1$ and $max-1$

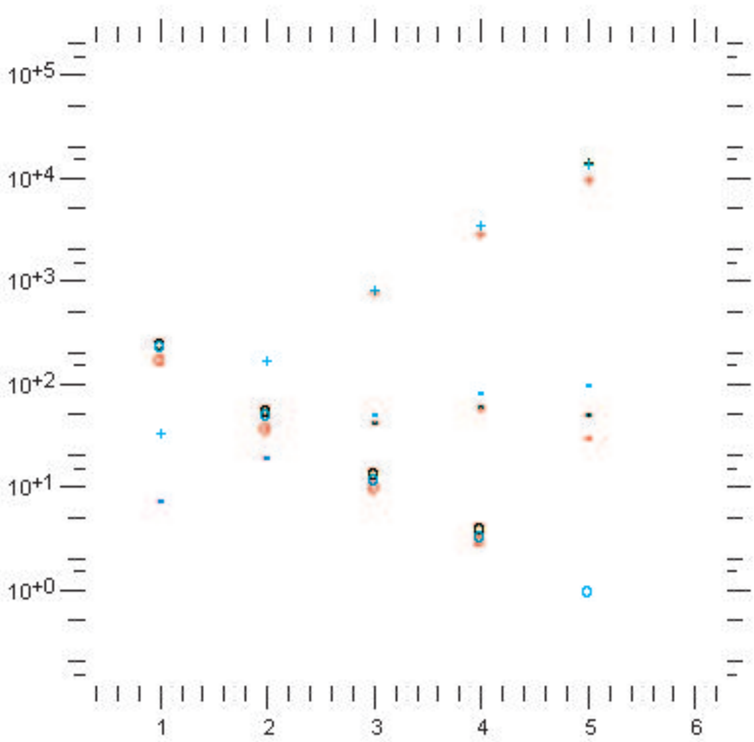


Figure 4: Horton's laws, 10000 cells, average method, sigmoid

and deriving a new average. Replacing the spherical shapes by streched ones also results in a more power-like law (Figure 4: blue). Finally the use of nonconcentrical models should significantly influence the behavior of streams.

2 Using Terraforming

Terraforming is written in *Python*, thus it cannot be used as a standalone application. It requires the interpreter at any time. Terraforming is developed in *Python 2.4* with the module *numpy*. The software is freely available on the www: for *Python 2.4* see *www.python.org,* numpy for *Python 2.4 at* www.scipy.org.

Terraforming has been developed to generate DEMs and a control model running as well as the DYNIA analysis of these runs. It is therefore restricted to a simple user interface which makes use of a console window and a control file. In the main console window the user is able to activate the principal functionality of terraforming by pressing the numbers of the sub menus offered. Sub menus again consist of functions which may be activated pressing the corresponding numbers on the keyboard. There are two possibilities to work with terraforming: Dealing with single digital height models and their evaluation and visualisation can be realized in a menu (console) mode, described in subsection 1. In the automatic mode terraforming can be used to create an arbitrary number of DEMs including soil- and landuse-grids. The terraforming script generates a batch file controlling the hydrological catchment model WaSiM-ETH. Restarting terraforming in the menu mode calculates and visualizes the results of the WaSiM-ETH model runs. The boolean parameter `auto` (`0` for menu mode and `1` for automatic genesis) in the control file `terra_man.gen` is used to switch between the automatic and manual modes of the program. To execute terraforming in the auto mode, you have to run the batch file `batch.bat`.

2.1 Menu Mode

Starting terraforming in the menu mode will display the main menu in the console window. Navigation is controlled by the numbers displayed in brackets (+enter). Available items in because the tangent is monotone. The available items in the main menu are:

(1) edit shape: DEM shape related tasks

(2) fill shape: height assignation related tasks,hypsometric curve

(3) analysis: flow net, soil and use tasks

(6) wasim output: see paragraph 3

(7) save/load: save: shape, DEM, landuse, soil; load: shape, dhm

(8) options: setup cellsize, nodata value and grid size

(9) exit: exit

2.1.1 edit shape

The shape of the DEM is visualized in the upper left square of the output window. It is not used for restricting the area of analysis, but to define the area for assigning heights.

(1) add/remove elements: The script prompts the user for the number I of elements to be added or removed. The typed integer I is treated as a circle of radius r, where the area of a single cell is 1. A left click on the shape square will set a discrete circle to attribute *area* and shift+left click click removes a cell from the shape. Because of the discrete, spheric symmetry, the number of added cells is not expected at I. Adding and removing elements by clicking on the grid will not activate the search for holes in the shape. A left click exits the sub menu.

(4) add random elements: I cells are randomly chosen step by step for the DEM shape. Every potential cell is added to the DEM shape with equal probability. The first cell of an empty shape is always situated in the middle of the base grid. After I cells are chosen, terraforming checks for holes in the shape. The holes are filled and an equal amount of cells is randomly removed from the boarder of the DEM in order to guarantee the total number I of cells for the watershed.

(6) get shape from dhm: Get the shape of the actual DEM and reject the old one.

(7) reduce dhm by shape: Use the overlapping area of any shape and the actual DEM to create a new DEM.

(9) back: Back to the main menu.

2.1.2 fill shape

The DEM with assigned heights is visualized in the upper left square too. A right click on the square switches between the DEM heights and shape.

(1) randomwalk D8: D8 randomwalk with given parameters (standard heights and hypsometric curve) is used to assign cell heights to the chosen shape. Before height assignment, the user has to specify the position of the outlet by clicking on the DEM shape. A shape can be refilled at any time.

(2) randomwalk D4: : D4 randomwalk is used analogously to the D8 procedure.

(7) hypsometric curve: The hypsometric curve will automatically be forced after height assignment, if it isn't set to *none*. Any hypsometric curve can be rescaled by using this tool (but not back to *none*). Available functions $f(x)$ are those from section **??** including an additional powerlaw function with exponent p. Because of the definition of the hypsometric curve, the used normalized functions: $f(0) = 0$ and $f(1) = 1$ are mirrored at $x = 0.5$, so the hypsometric curve becomes $h(0) = 1$ and $h(1) = 0$.

(8) rescale height: The minimal and maximal height values are read from the file `terra_man.gen`. To rescale manually, the user is able to specify new minimal and maximal values.

2.1.3 analysis

(1) plot hypsometric curve: This will plot the hypsometric curve in the lower left corner of the output window.

(3) flownet D8: Analysis of the river network is realized in this sub menu. The drainage density and the ordering of streams is displayed in the upper right of the visualization window (right click to switch). Horton's law is checked in the lower right.

(4) flownet D4: All river network applications are available for D4 too.

Some commands directly execute a tool from everywhere in the menu: Type `rwd4` or `rwd8` to assign heights to a shape by randomwalk. `fnd4` and `fnd8` derives the D4 and D8 river network of the actual DEM. Typing `exit` will exit terraforming immediatly. Some parameters used in menu mode will be saved in the file `terra_man.gen`: grid file header data, hypsometric curve and generator model (see section **??**), minimal and maximal height, threshold number of cells to define a first order stream.

2.2 Auto Mode

Before using terraforming in auto mode, it has to be assured that the local path is set to the *Python24* root directory and *Windows\system32* in file `pack\batch.bat`. Terraforming is executed in auto mode by running the batch file (after setting `auto=1` in the control file `terra_man.gen`).

To start the automatic genesis of digital height models, the file `terra_auto.gen` must be edited. The first parameter is related to the terraforming output path and should be set to `terra_o\\` . The doubled backslash is important for python internal usage. Output files in `outpath` are labeled in the form `outprefix_x.outsuffix` where `x` stands for the number of the generated DEM without leading zero. The next lines are definitions for the output suffix of DEM, shape, soil and landuse data grids. The values for the standard ascii header are the parameters 8 to 13. They will be ignored, if any input like a shape- or DEM-file is used. Variable `dhm_files` defines the number of files to generate. The parameter `dhm_shape` is either a valid file name that will be used for assigning heights, or can be set to `'random'`, to use a number of random cells (parameter `random_elements`) to build a shape. The outlet of a given shape can be defined with the variable `outlet`. `dhm_shape` will be ignored if `dhm_file` is not `'random'`. In this case terraforming will alter properties of the given grid but not change the topology. The outlet will be detected for a given dhm if `outlet = 'detect'`. The variables `generator_dx` and `generator_model` switch between `d4` and `d8` randomwalk and define the matrix G introduced in chapter 1.2.

In the version of terraforming, minimal and maximal height of the resulting n dhms can be linearly increased step by step by using the keyword `minimal_height = 'linear'`. The values `min_min_height` and `max_min_height` are the lowest and the highest minimum value. Likewise, the highest grid values can be scaled. The `flownet_parameter` and its minimal and maximal values are working in the same way, to alter minimal and maximal drainage area for the first order streams linearly. For holding those values instead of increasing, a value of choice can be used instead of `'linear'` or equal values for minimal and maximal parameters can be assigned.

The keyword `hypsometric_curve` stands for the curvature to be used. Valid entries are `'linear'`, `'sigmoid'`, `'exp'`, `'invexp'` or `'none'`.

If `hypsometric_curve = 'powerlaw'`, then the next keyword `hypsometric_power_mod` stands for the exponent to use (e.g. a floating point number). If further `hypsometric_power_mod = 'monte_carlo'`, one can set up the lower and upper boundary for a random manipulation of the powerlaw by `hypsometric_power_min` and `hypsometric_power_max`.`hypsometric_power_mod = 'linear'` is also possible.

The parameter `analysis_dx` sets drainage network to `'d4'` or `'d8'`.

If `exec_wasim = 0`, *WaSiM Eth* is not executed.

Finally terraforming writes a log file called `terra.log` in the output path. This file contains average stream numbers, areas and lenghts of all DEMs generated.

2.3 Auto Mode Example

Exemplarily, a 500m grid of the Schwarze Pockau watershed at Zblitz gauging station was rescaled for automatic genesis. The natural maximum height is 901m. 10 DEMs with maximal height ranging from 550m to 1450m (holding minimal height at 450m) and natural soil and landuse grids were used for a *WaSiM-ETH* analysis. In Figure 4 the 10 hydrographs are displayed on the left side. On the right side n represents the number of the DEM. $n = 0$ characterizes the natural DEM. The realisations are displayed with an abscise representing the peak flow difference of the n^{th} DEM with respect to the natural DEM hydrograph. From $n = 1$ to 11 the scaled DEMs are distruibuted linearly from 550m to 1450m for the maximum height. The file `terra_auto.gen` used to produce this result is commented below:

`outpath="terra_o\\"`
`outprefix="z_500"`
`outsuffix_dhm="dhm"`
`outsuffix_ezg="ezg"`
`outsuffix_use="use"`
`outsuffix_stp="stp"`
`outsuffix_alb="alb"`
	The outpath for dhm will be `terra_o\`
	with file names `z_500_1.dhm`
	to `z_500_10.dhm`

`ncols = 75`
`nrows = 75`
`xllcorner = 56.56`
`yllcorner = 56.56`
`cellsize = 500`
`nodata_value = -9999`
	This ascii header
	will be ignored, because
	a file is loaded.
	Terraforming uses the
	file header instead.

dhm_files = 10	10 DEMs are generated and evaluated
dhm_shape = 'random'	Ignored, shape of dhm_file will be used.
dhm_file = 'terra_i\z_fix_d8.txt' outlet = 'detect'	This file will be loaded. outlet must be set to 'detect'.
random_elements = 10000	Ignored
generator_dx = 'd8' generator_model = 'linear'	Ignored, because there is already a topology.
minimal_height = 450 min_min_height = 450 max_min_height = 1450	If minimal_height is set to a value, the min and max parameters are ignored.
maximal_height = 'linear' min_max_height = 550 max_max_height = 1450	Height is increased linearly from 550m to 1450m in ten steps.
hypsometric_curve = 'none'	The original DEM hypsometric curve is used.
hypsometric_power_mod = 'monte_carlo' hypsometric_power_min = 2.0 hypsometric_power_max = 3.5	Only valid for 'powerlaw' Lower and upper boundaries for a monte carlo simulation
analysis_dx = 'd8'	D8 method for stream analysis
flownet_parameter = 5 min_flownet_parameter = 3 max_flownet_parameter = 16	Threshold for stream order 1 Ignored Ignored
soil_grid_model = 'ref' use_grid_model = 'ref' albedo_grid_model = 'ref'	*WaSiM-ETH* uses the reference grids for its simulation.
execwasim = 1 shuffle_wasim = 0 stop_error = 1	*WaSiM* will be executed No monte carlo simulation for *WaSiM* Terraforming waits for 'enter' at errors

2.4 File Structure

Terraforming uses several files and a special directory structur. The whole package uses various applications and links them via batch files. Directories in pack\ are:

backup Before running terraforming in the auto mode, the content of this directory will be replaced by out\. This creates a backup of the last model run.

manual contains the manual.

meteo *WaSiM* The meteorological data is stored here by default.

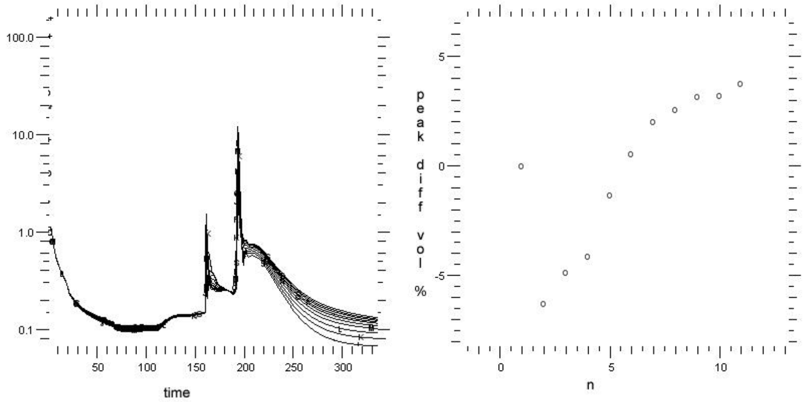


Figure 5: 10 digital height models: outlet direct flow (ls) and percentage difference of the peak to reference peak (rs)

out This directory contains the output of *tanalys* (in `out\x\`) and *WaSiM* (in `out\wasx\`), where `x` labels the DEM without leading zero. `out\ref\` and `out\wasref\` are locations for the output of the reference data.

ref The reference input data: `dhm.dhm` must be an ascii file with valid header. `dhm.ezg`, `dhm.stp`, `dhm.use`, `dhm.alb` are ascii-files with DEM-shape, soil type, landuse and albedo information.

temp Temporary files used in the process of batching are written here.

terra_i It's the default shape and DEM input directory of terraforming.

terra_o In the output directory of terraforming the auto-mode saves files, including the log file.

was_i The batch file will copy files located in this directory into the *WaSiM* model run directory.

ascigrid.exe Converts ascii files to grid format (Schulla 1997).

batch.bat This batch script starts terraforming in the auto mode (if `auto = 1` in `terra_man.gen`).

gridasci.exe Converts grid files to ascii format (Schulla 1997).

mc.exe Thix executable is used to manage the monte carlo randomization of `was_steuer.was`

modell.log *WaSiM-ETH* log file

tanalys.str Is the *tanalys* control file.

tanalys.exe *tanalys* (Schulla 1997)

terra_auto.gen This is the setup file for the terraforming auto mode.

terra_man.gen In this file the initialisation data for the terraforming menu mode is saved.

terraforming.py The terraforming python script file.

was_batch.bat Is a batch script written by terraforming in auto mode to manage the *WaSiM* simulation of the DEMs.

was005.exe *WaSiM Eth* executable

was_steuer.ref A *WaSiM* control file used for the reference model.

was_steuer.was This is the *WaSiM* control file used for the generated dhms.

3 Using *WaSiM Eth* in Auto Mode

If `exec_wasim = 1` in the auto mode setup file `terra_auto.gen`, terraforming writes a batch script to run *WaSiM* for each DEM. Meteorological data is stored by default in the directory `meteo\` and *WaSiM* control files pointing them. *tanalys* writes its output files in `temp\` and *WaSiM* in `temp\was\`. After a simulation is done, the *tanalys* files are moved to `out\x\` and the *WaSiM* output data into `out\wasx\`, where `x` is for the number of the DEM. Because the moving of the files is realized by a batch, it must be assured that none of those files or directories are opened during the simulation.

3.1 Shuffle *WaSiM*

If `shuffle_wasim = 1` in `terra_auto.gen`, `was_ batch.bat` calls `mc.exe`, the monte carlo program. In automatic mode `was005.exe` will read its control file `was_steuer.was` for parameters. The k_i, k_d and f_d parameters can be randomized by editing this control file: a line `#monte_carlo` (without any spaces) in `was_steuer.was` will cause `mc.exe` to skip the next line (which is important for further dynia analysis of the parameter) and read an interval from the third line `#(min,max)`, with integers *min* and *max*. Finally a value with precision 1 will be written in line four. It is important to break the lines after the monte_carlo command, keyword and interval without any additional spaces.

`# codes of the subbasins` `1`	

`#monte_carlo`
`# kelsqi`
`#(1,6)`
`3.3 #`
	the monte_carlo command;
	the keyword characterizes K_i;
	the random value interval is $[1, 6]$;
	the random value is 3.3

`# monte_carlo`
`# kelsqd`
`#(4,12)`
`6.6 #`
	this is not the exact command;
	anyway the keyword `'# kelsqd'` is
	used to identify the parameter;
	6.6 is valid for every simulation

`#monte_carlo`
`# flow density (for Interflow, channels per km)`
`#(4,12)`
`5.8 #`
	again the command is found;
	parameter f_d comes
	with a very long keyword

3.2 DYNIA analysis

DYNIA (DYNamic Identifiability Analysis) is a method for parameter identification introduced by Thorsten Wagner (2003). This approach evaluates changes in parameter identifiability for any set of time series depending on a Monte Carlo parameter distribution. In a moving window every time series leads to a identifiability measure (e.g. mean absolute error). A best fitting subset of parameters (identified with the best fitting time series, e.g. 20%) in every time step is used to plot a histogram of parameters over time.

This parameter analysis is implemented in the main menu as item `(6) wasim output`. First, simulated data has to be loaded: Subitem `(1) load wasim data` reads the root directory for the `wasx\` directory locations of the simulations, e.g. `out\`. The second string is for the name of the files to be read in those directories, e.g. `qdirdhm.s1996`. A third input asks for the number of the last `x` in `wasx\` to be processed. `(2) load wasim control files and topological parameters` will search for the wasim monte carlo values in the `wasx\was_steuer.was`. `terra.log` is opened in the `wasx\`-root directory.

Item `(6) dynia` starts the DYNIA analysis: Typing `s` will set up the size of the moving window as a relativ value of moving window size to the total number of time series, `t` the relativ number of best subset to total set of simulation and `c` the number of the containers of the histogram. The available parameters are listed and typing their item number leads to the DYNIA analysis. After every step of the moving window procedure is finished, the top level or the number of containers and the observed parameter can be resetted without running the moving window procedure again. Each container is normalized by the total number of time series in this container, to minimize the effects of discontinual Monte Carlo simulations. Every time step is normalized, so that the sum of all containers equals 1. The resulting parameter probability density of the best simulations is plotted in the DYNIA window.