User guide for SMART-SED

In the following we define the SMART-SED root directory the one that contains the README.md file and which has the folders:

- DeterministicProgram
- Geostatistics
- Inputs
- Outputs
- run

For Windows users, we suggest to install Docker (https://www.docker.com/products/docker-desktop). We provide a Dockerfile in the folder run/Docker where one can build the Docker environment for the SMART-SED program.

For UNIX users it is sufficient to install the dependencies listed in the README.md file.

Fist use: Compile

Compiling is necessary at the first stage. Run the Docker Machine, run the file runWindows.bat in the folder run/Docker. Execute the command "make -j" in the directory DeterministicProgram. Wait until it finishes.

Geostatistical part: Input data

In the Inputs directory you can find a folder named Geostatistics_input_data where you must put four files in .tif format and with the following names,

- DEM.tif
- clay.tif
- sand.tif
- silt.tif

they are read by the R script and are the digital elevation model and the coarse scale clay, silt, sand maps respectively. The coarse particle size fractions (psf) maps, which are given e.g., from the online repository SoilGrids (https://www.isric.org/explore/soilgrids), must have the same resolution among them and are going to be downscaled at maximum up to the resolution given by DEM.tif file. All the files should be related to the same geographical region.

Execution script

The execution script is one file named with extension .sub and is a shell script file. Let us consider for the sake of simplicity the file submission_ev_local.sub that reads the file SMARTSED_input_ev.

At line 26, through NPROCS, one must specify the number of MPI processors to use to perform the simulation. We remember to the reader that the code execution remains always serial, it is useful to use more than one processor in case one should perform stochastic simulations. The number of MPI processors to use is upper bounded by the number of stochastic simulations to be performed. This last number is set at line 106 via the variable nsim. In formula we have, $NPROCS \leq \max(nsim, 1)$. The variable nsim could be also -1 or 0. In case nsim=-1 the Gestatistical preprocessor is not run, and the user must specify the psf maps in the file SMARTSED_input_ev, see next sections. In case nsim=0 no conditional simulations are performed and the psf maps are the output just the kriging procedure.

Finally, we have another variable called res at line 107. It represents the scaling factor with respect to the DEM.tif file. To make an example if DEM.tif has resolution 5mx5m and res=4 it means the simulation resolution is set equal to 20mx20m.

To execute the script just type "./submission_ev_local.sub" in the "run" directory. In case enable the execution permissions with "chmod u+x submission_ev_local.sub". Before running the script follow the next section where we set the input parameters for the numerical part.

Numerical part: Input data

It is possible either to insert the input files in single folders or to put all of them in a unique folder. The input file paths must be inserted into the text file SMARTSED_input_ev (folder run). The file path always starts from the Inputs folder. If one wants to insert comments in this file put a hashtag beforehand the comment.

The complete list of the input parameters follows.

DEM

```
[files]
    orography_file              = realOrography/DEM.asc
    mask_file                   = realOrography/Mask_bin.asc
```

First input is the orography (DEM), the second one is the hydrogeological basin mask. Provide both the files in ESRI ASCII format (.asc is the classical extension). Convert the DEM.tif in ESRI ASCII and put the path in orography_file.

Meteo data: Temperature

```
[./meteo_data]
temperature_file              = DatiMeteo_ARPA_LOMBARDIA/lecco/temperature_lecco_ev0.txt
height_thermometer            = 272
time_spacing_temp             = 0.1667
format_temp                   = arpa # comune, arpa
```

Insert one temperature file with the given altitude, standard atmosphere law is applied. The height is expressed in sea level meters. The time_spacing_temp is the spacing in seconds of the temperature file provided.

Then you need to specify the kind of format of the temperature file. Two different formats are supported: comune and arpa. They are both tab delimited files, we recommend to use the arpa format and we describe here this input file for simplicity. The header should be like the following, the one we provide in the folder Inputs/DatiMeteo_ARPA_LOMBARDIA/lecco/temperature_lecco_ev0.txt. We have four columns the first is an unused number related to the seonsor ID, it is not used by the program. The second and the third are the day and time of the day yyyy/mm/dd, hh:mm respectively. The last one is the value in degree Celsius of the temperature.

Meteo data: Rain

```
precipitation                 = true
constant_precipitation        = false  # false: IDW

# comune lecco files
rain_file                     = pioggia_test.txt
time_spacing_rain             = 0.1667

# arpa files
number_stations               = 9
# if constant_precipitation false, provide multiple rain data, insert NODATA = -999
rain_file_1                   = DatiMeteo_ARPA_LOMBARDIA/rotaImagna/rain_rotaImagna_ev0.txt
X_1                           = 539709.2
Y_1                           = 5076380.39
time_spacing_rain_1           = 0.1667
```

The flag precipitation if set to true we have precipitation rates as given by the input files otherwise the files are not read, and we have no rain. The flag constant_precipitation means uniform or not on the whole basin. If this flag is true is read the file provided in the variable rain_file otherwise one should provide the number of rain stations and provide each one with the given geographical coordinate and time spacing (e.g., _1 for the first station, _2 for the second station and so on, in the considered file we have set up to 9). In this last case the rain field is computed via inverse distance weighting method.

In case constant_precipitation=true the input file must be like the file we provide in Inputs/rain/pioggia_id_constant.txt. Otherwise, the rain files are like the one we provide in 2020/cortenova2020_bat.txt. In both the files the last column is related to the rainfall in millimeters. Note also that the day put is important and should be consistent with the one given in the temperature file because quantities like evapotranspiration depend on the day and season.

Initial conditions

Follows initial conditions if provided set to true the corresponding flag. We note that it is possible to provide input psf maps so if restart_soilMoisture is set to true the output of the geostatistical preprocessor is not read. One should equally set nsim=-1 in the execution file.

```
restart_soilMoisture        = false
clay_file                   = ideal/clay_id.asc
sand_file                   = ideal/sand_id.asc
```

We note that one could set a resolution finer to the simulation resolution for the initial condition maps and the program automatically rescale to coarser resolution via bilinear interpolation.

Infiltration

The only infiltration model implemented is the SCS-CN model, one can choose the presence of initial loss (isInitialLoss) and the coefficient of initial loss (perc_initialLoss). We have then three parameters about the roughness factor valid for three different values of slope S, i.e., roughness_scale_factor1 ($S \leq 0.2$), roughness_scale_factor2 ($0.2 < S \leq 0.6$), roughness_scale_factor3 ($S > 0.6$). Finally, in corineCode_file you need to set the path of the corine land cover file.

Evapotranspiration

The only evapotranspiration model implemented is the Hargreaves one (ET_model), you just need then to specify the mean latitude in degrees of the basin (latitude_deg).

Friction and Sediment

Through friction_model you specify the friction model (Rickenmann or Manning). n_manning specifies the value of the Manning friction coefficient, i.e., an uniform value in case of friction_model=Manning, and in case friction_model= Rickenmann it specifies the minimum value of the Manning friction coefficient. The Gavrilovic_txt is an ASCII file with the values of Gavrilovic coefficients used, it is a two-column file, see e.g., coeff_Gav.txt in Inputs directory. Finally, is_sediment_transport if true enables the sediment movement in the computation.

Other options

We mention the most relevant ones,

- FillSinks if true smoothens sinks to avoid erroneous accumulations of water in concentrated areas. It is based on a depression filling algorithm,
- steps_per_hour is the minimum number of time steps per hour we want to perform,
- max_Days is the number of days we want to simulate, we remember that the starting day is given in the meteo files,
- H_min and T_thr are the threshold for conservation purposes and the threshold under which we have snow respectively,
- number_gauges is the number of observation points we want in the numerical model, i.e. where we want to save the temporal sequences if the corresponding flag is set equal to true (save_temporal_sequence). Provide the observation points as X_gauges_1, Y_gauges_1, X_gauges_2, Y_gauges_2, X_gauges_3, Y_gauges_3, …
- delta_gauges delimits the quadrilateral neighbourhood we consider as "tolerance" on the observation points,
- frequency_save is the saving time in hours of the entire solution, i.e., maps solution in ESRI ASCII format.

Output

To save the solution in a different folder, go to file submission_ev.sub and change *JOBID = {name of the new folder}*.

The output files (the maps files in ESRI ASCII format) are the following:

H – water surface elevation [m]

hG – gravitational component of water within the soil (water table) [m]

hsd – sediment accumulation value relevant to the respective timestep, normalized to the cell dimension [m]

hsn – snow depth [m]

w – map of sediment source areas [m]

The ASCII output files are the following:

- *waterSurfaceHeight* : superficial water height [m] at every time step for each control point,
- *waterSurfaceMassFlux* : h_{water} x vel_{water} [m^2/s] at every time step for each control point,

- *SolidFlux* : h_{sed} x vel_{water} [m^2/s] at every time step for each control point,
- *timesteps* : to plot the sequence time step in second against the time in seconds.