KLATra documentation

KLATra_v1.py

Paule Hainz

September 13, 2024

No Software Warranty
User acknowledges and agrees that the software is provided "as is", with all faults and without warranty of accuracy or warranty of any other kind.

1 Using KLATra

KLATra needs as input a FORTRAN namelist like the example namelist KLATra_in.in. Inside this namelist all needed information can be passed to the script.

You can run KLATra in your command line like this:

```
python KLATra\_v1.py -f namelist.in
```

`-f` or `--file` is necessary to show that you pass the namelist file.

KLATra reads the namelist and the namelist of the KLAM_21 simulation for which trajectories should be calculated. It checks whether there are files for the simulation begin and creates them if necessary. As there is no cold air layer at the simulation start, u, v, Hx and Ex are all zero over the whole model area.

KLATra uses either timesteps from KLAM_21 dt file or a fixed timestep for the calculation of trajectories. This can be decided in the namelist.

The input coordinates (i.e. the position where trajectory calculation starts) can be created in various modes:

- centers of random cells from one land use class
  - in the whole model area ('landuse')
  - in a rectangle ('landuse_area')
  - in a polygon ('landuse_poly')
- centers of random cells

  - in a rectangle ('rectangle')
  - in a polygon ('polygon')
- centers of cells at the edge of a rectangle ('rectangle_border')
- specific coordinates (these do not need to be centers of cells) ('single')

Which mode should be used can be specified via the input file. For randomly distributed input coordinates a trajectory starting density $[\frac{1}{ha}]$ has to be set in the input file. A numpy random generator is used with a seed, so with unchanged input coordinate parameters the input coordinates are reproducible. Based on the way you want to use, you have to specify the coordinates in y_koord and x_koord. If you want something with a rectangle you need to specify only two diagonal corners. For a polygon you have to write all the corners.

If you want to use a version based on landuse, KLATra needs a polygonized version of the landuse file. It should be saved as a .geojson and set in the namelist in the block 'trajectories' with the variable 'lu_file'. You can polygonize your raster file using gdal_polygonize.py (https://gdal.org/en/latest/programs/gdal_polygonize.html) in your command line.

Output can be saved as KLAM_21 compatible pl_files, so that they can be viewed in KLAM_21's GUI, or as geojson files. For the geojson output two files are created - one time LineStrings and the other time Points. It is not possible to save information for every output position of a trajectory inside the linestrings, so it is saved for each point separately. The LineStrings are saved as well as the are useful for a first overview.

2 Overview KLATra

The post-processing tool to calculate trajectories from KLAM_21 simulation results was implemented using python. Necessary parameters are passed to the program as a FORTRAN namelist as possible users know this format. KLATra interpolates KLAM_21 results temporally linear and uses inverse-distance-weighting (IDW) for spatial interpolation of the four grid cells closest to the position of the trajectory. The power parameter is set to p = 2, so the weight decreases with the square of the distance (Shepard, 1968). Which four grid cells are to be interpolated is decided by the position of the trajectory inside the grid cell in which it is. If it is in the middle of a grid cell or in a grid cell at the border of the model area, no spatial interpolation takes place. The distance is calculated from the trajectory's position and the grid cells' centre.

$$g_i = \frac{\sqrt{(x-x_i)^2+(y-y_i)^2}^{-p}}{\sum_{i=1}^{4}\sqrt{(x-x_i)^2+(y-y_i)^2}^{-p}} \tag{1}$$

$$var_1 = \sum_{i=1}^{4} g_i * var_{1i} \tag{2}$$

$$var = var_1 * \frac{t_2 - t}{t_2 - t_1} + var_2 * \frac{t - t_1}{t_2 - t_1} \tag{3}$$

Indices 1 and 2 in the equations show that the value is for one of the KLAM_21 output times, index i means the value is that of one of the four grid cells and no index means it is the spatially and temporally interpolated value of the trajectory. var can be one of the scalar variables u, v, Hx or Ex; g is the weight from IDW; t is time and $t_1 \leq t < t_2$. var_2 is calculated analogue to eq. (2). All variables are converted to SI-units in KLATra. For u and v normally the vertically averaged values are used, but it is possible to use the wind field for a selected height above ground (if KLAM_21 was run with this height; there is no estimation of wind speed in a specific height above ground in KLATra).

KLATra uses KLAM_21's calculation time steps for numerically stable trajectory calculations without implementing criteria for it. It reads the time step file from KLAM_21 and searches for the time steps which were used closest to simulation time t. The smaller one of these two is then used as Δt. Alternatively, a fixed time step can be used. The new coordinates are then:

$$y_{t+\Delta t} = y_t + \Delta t * v \tag{4}$$

$$x_{t+\Delta t} = x_t + \Delta t * u \tag{5}$$

u and v are fixed for Δt.

Starting and ending times of trajectories can be chosen freely from the whole simulation time. Based on these times trajectories are calculated forwards or backwards. In one run trajectories can be calculated for different time intervals.

The results can be saved in two formats: as a pl-file (Sievers, 2005), which allows displaying of the trajectories in the KLAM_21 GUI, or as two GeoJSON-files (Butler et al., 2016) for use in GIS. By default the position of the trajectories is saved approximately every 300 seconds, but this interval can be changed. Besides position also time, time step, land use class, wind speed, trajectory number, and interpolated values of u, v, Hx and Ex are saved. As it is not possible to save GeoJSON LineStrings with attributes for each position, there is also a Point file where this information is stored. In the pl-file there are some columns for line style next to the trajectory results. Backward trajectories are normally set to be displayed red and forward trajectories blue in the KLAM_21 GUI. For better visualization of the progression over time in the input file the parameter "colorgradient" in the block "output" can be set as `.true.`. This results in the trajectories being cut into half-hour blocks with sequential colors.

References

Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., & Hagen, S. (2016). The GeoJSON Format. https://doi.org/10.17487/RFC7946

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, 517–524. https://doi.org/10.1145/800186.810616

Sievers, U. (2005). *Das Kaltluftabflussmodell KLAM_21: Theoretische Grundlagen, Anwendung und Handhabung des PC-Modells*. Selbstverl. des Deutschen Wetterdienstes
CD-ROM u.d.T.: Das Kaltluftabflussmodell KLAM_21, PC-Demonstrationsversion.