# LagTrack

Functions are written to be used both as part of the GUI and as standalone. The main GUI is opened using:

```
LagTrack
```

Note that the GUI requires the GUI Layout toolbox. A summary of all functions is printed with the command:

```
help LagTrack_functions
```

## Table of content



## Input parameters

### DEM

The DEM is automatically retrieved using the readhgt function.

#### **Download the DEM**

Using the GUI:

```
downloadSRTM
```

Using the command line:

```
downloadSRTM(latMin, latMax, lonMin, lonMax, resolution, name)
```

- `latMin`, `latMax`: Minimum and maximum latitudes in decimal degrees. Negative in southern hemisphere
- `lonMin`, `lonMax`: Minimum and maximum longitudes in decimal degrees. Negative in western hemisphere
- `resolution`: Resolution to interpolate the DEM (m). This is obsolete (but still required for now), as the code will automatically attempt retrieving SRTM1 data
- `name`: Name of the DEM dataset stored in `input/dem/`

**DEM format**

The DEM format in LagTrack is a Matlab structure containing called `dem` and containing the following fields, where *m* and *n* are the number of cells in the *y* and *x* dimensions, respectively. Adopt this convention to use a DEM obtained from a different source in LagTrack.

- `X`: *[m×n]* matrix of longitudes
- `Y`: *[m×n]* matrix of latitudes. In the Matlab matrix, `dem.Y(1,:)` should be the southernmost points and `dem.Y(end,:)` the northernmost
- `Z`: *[m×n]* matrix of elevations (m asl). The orientation should be the same as `dem.Y`
- `res`: Cell size (m) (obsolete)

**Empty grid**

For calculations of particles trajectories that do not require a DEM, LagTrack can create an empty calculation grid specifying only an elevation used to stop the particle:

```
makeDefaultGrid(alt, name)
```

- `alt`: Mean elevation of the grid (m asl)
- `name`: Name of the grid stored in `input/dem/`

## Atmospheric data

Atmospheric data in LagTrack can be retrieved from the NCEP/NCAR Reanalysis 1, the NCEP-DOE Reanalysis 2, the ECMWF Era-Interim and the ECMWF Era-5 datasets. To access Era-Interim data, it is assumed that the procedure described here has been followed. To test the installation, run the following command in Matlab. If no message is output, then the ECMWF library is working properly:

```
!python -c "from ecmwfapi import ECMWFDataServer"
```

**Download atmospheric data**

Using the GUI:

```
downloadATM
```

Using the command line:

```
downloadATM(latMin, latMax, lonMin, lonMax, yearMin, yearMax, monthMin,
monthMax, name, dataset, varargin)
```

- `latMin`, `latMax`: Minimum and maximum latitudes in decimal degrees. Negative in southern hemisphere
- `lonMin`, `lonMax`: Minimum and maximum longitudes in decimal degrees. Negative in western hemisphere
- `yearMin`, `yearMax`: Minimum and maximum years to retrieve (e.g. 2017). For a single year, `yearMin`=`yearMax`
- `monthMin`, `monthMax`: Minimum and maximum months to retrieve (e.g. 02 for Feb). For a single month, `monthMin`=`monthMax`
- `name`: Name of the atmospheric dataset stored in `input/wind/`
- `dataset`: Reanalysis dataset, accepts `'ERA5'`, `'Interim'`, `'Reanalysis1'` and `'Reanalysis2'`
- If using `'ERA5'`, an additional optional parameter can be passed representing the number of hours between two data points (accepts `1`, `2`, `3`, `4`, `6`, `8`, `10`, `12`, `24`, where `6` represents one data point every 6 hours, i.e. 4 data points per day) (default=`6`)

**Format of atmospheric data**

The format of atmospheric data in LagTrack is a Matlab structure called `atm` and containing the following fields:

- `lat`, `lon`: Latitude and longitude vectors. `lat` is a *[m×1]* vector and `lon` is a *[n×1]* vector, where *m* and *n* are the number of points along latitude and longitude, respectively
- `level`: Geopotential height (mb). *[l×1]* vector, where *l* is the number of levels
- `time`: Date vector of each data point in number of days from January 0, 0000 (see Matlab function `datenum`). *[t×1]* vector, where *t* is the number of data point in time
- `temp`: Temperature (deg K). *[m×n×l×t]* matrix
- `alt`: Altitude (m asl). *[m×n×l×t]* matrix
- `humid`: Relative humidity (%). *[m×n×l×t]* matrix
- `u`, `v`: U and V components of wind (m/s). Each is a *[m×n×l×t]* matrix
- `rhoair`: Atmosphere density (kg/m2). *[m×n×l×t]* matrix
- `muair`: Atmosphere dynamic viscosity viscosity (Pa s). *[m×n×l×t]* matrix

**Post processing the atmospheric data**

Post-processing of the atmospheric dataset should be automatic upon successful completion of the download step. To manually post-process wind data, use:

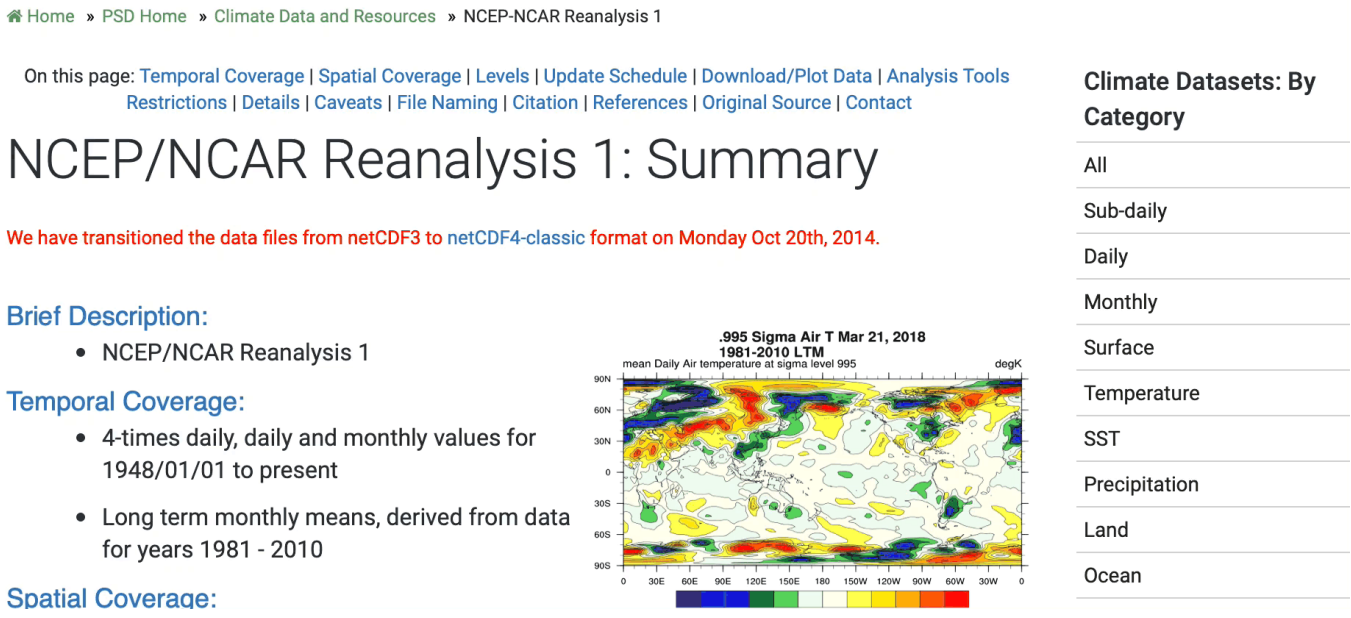
```
processATM(name, dataset, latMin, latMax, lonMin, lonMax, yearMin,
yearMax, monthMin, monthMax)
```

⚠️ All variables except `rhoair` and `muair` must be provided in the NetCDF file. `rhoair` and `muair` are automatically computed by the `processATM` function.

### Manually downloading Reanalysis data

**Era-Interim:** There are two possible options. To access data in batch access mode outside of LagTrack, use the python template provided in `code/functions/dependencies/ecmwf-api-client-python/download_ECMWF_tmp.py`. Otherwise, NetCDF should be manually downloaded *for each separate month* from here and placed in the folder `input/wind/windName/`. NetCDF files should contain the following variables: `latitude`, `longitude`, `time`, `level`, `u`, `v`, `z`, `t` and `r`. More details on the procedure is available here.

**Reanalysis 1/2:** Reanalysis 1 and Reanalysis 2 must be retrieved *for entire years* at *pressure levels*. Variables to be retrieved are `Air temperature`, `Geopotential height`, `Relative humidity`, `u-wind` and `v-wind`. The yearly NetCDF files must be placed in the folders `input/wind/_Reanalysis1_Rawdata/` or `input/wind/_Reanalysis2_Rawdata/`.




### Standard atmosphere

As an alternative to Reanalysis dataset, LagTrack also offers to create a user-defined wind field in a US 1976 standard atmosphere:

```
makeStandardAtm(uwind, vwind, name)
```

- `uwind`, `vwind`: Trigonometric *u* and *v* components of the wind (m/s)
- `name`: Name of the dataset stored in `input/wind/`

# Running the model

## Defining particles

In LagTrack, each particle belongs to a **run**. Multiple particles can depend on a same run. Each particle is a Matlab structure containing various fields described below. The following list describes the structure's fields, whose descriptions are also applicable in the main GUI.

- `run_name`: Run name to which particles are associated
- `run_mode`: Forward (`1`) or backward (`2`) runs (see next section)
- `vent`: Structure containing vent properties
  - `lat`, `lon`: Vent latitude and longitude (decimal degree)
  - `alt`: Vent elevation (m asl)
- `date`: Eruption date (number of days since Jan 0, 0000)
- `path`: Structure containing paths to input parameters
  - `nc`: Path to the atmospheric data
  - `dem`: Path to the dem
- `part`: Structure containing the particle's aerodynamical properties
  - `name`: Particle name
  - `diam`: Particle diameter (m)
  - `dens`: Particle density (kg/m3)
  - `flat`: Flatness (0-1)
  - `elon`: Elongation (0-1)
- `rel`: Structure containing the particle's release properties
  - `x`, `y`: Horizontal displacement (m) of release point relative to the vent; positive towards N and E, negative towards S and W
  - `z`: Release elevation (m relative to vent)
  - `t`: Time offset (s relative to `date`)
  - `vx`, `vy`: Initial release velocities along the x and y axes (m/s). Use a value of *-1* to set particle's initial velocities to be equal to the u and v components of wind
  - `vz`: Initial vertical velocity (m/s), positive upwards, negative downwards. **Must not be null**
- `adv`: Structure containing advanced properties
  - `solution`: Accepts 'euler' or 'analytical'
  - `dt`: Time step (s)
  - `drag`: Region of reduced drag (m) around initial particle release point
  - `interp`: Interpolation of the atmospheric data. Accepts 'subset' (default; only a subset of the domain is interpolated), 'complete' (the entire domain is interpolated) and 'none' (no interpolation)
  - `method`: Interpolation method. Accepts 'linear', 'nearest', 'pchip', 'cubic' and 'spline' (see `interpn` Matlab function)
  - `range`: If the interpolation iis set to 'subset', defines a range of points in each direction around the current particle location to interpolate the atmospheric data
  - `skip`: Number of `dt` between interpolations

## Run mode

LagTrack can be used calculate the trajectory of particles either in a *forward* or *backward* mode.

- In *forward* mode (`run_mode = 1`), particles are release in the atmosphere at the coordinates [`vent.lat`, `vent.lon`, `vent.alt`] (± `rel.x`, `rel.y`, and `rel.z`) at time `date` (± `rel.t`) and their trajectories are computed by advancing in time (`dt = dt`) and with a gravity force oriented downwards (`g = -9.81`) until the particle intersects the DEM;
- In *backward* mode (`run_mode = 2`), particles are release in the atmosphere at the coordinates [`vent.lat`, `vent.lon`] (± `rel.x`, `rel.y`) at the DEM elevation at this location (± `rel.z`) at time `date` (± `rel.t`) and their trajectories are computed **backwards** in time (`dt = -dt`) and with a gravity force oriented upwards (`g = 9.81`) until the particle intersects the altitude defined by `vent.alt`.

### Default particle

It is possible to create a default particle with the command:

```
load default_part
```

Details of particles requirements can be displayed in Matlab with the command:

```
help LagTrack_particle
```

### Model run

The trajectory of particles is computed with the function `get_trajectory`, where `part` is a particle previously defined:

```
get_trajectory(part)
```

It is possible to run several particles in one command by grouping the into a cell array, in which case they are run in parallel using the Parallel Computing Toolbox (if available):

```
get_trajectory({part1, part2, partn})
```

## Results

### Trajectory

Upon run completion, each particle is saved as a `.mat` file in `project/runName/particleName.mat`. Three fields are appended to the original structure:

- `run_check`: Status of the run - 1 if completed normally, 0 if the particle landed outside of the domain or did not hit the ground before the end of the atmospheric dataset
- `timestamp`: Time of impact with the ground

- `traj`: Structure containing the particle's properties at each time step

### Plot results

Particles can be visualised using the functions `map_part` and `plot_part`. These functions open GUIs and take no input arguments.

## Additional information

### Requirements

LagTrack has been initially developed using Matlab 2014b and has been tested to run on all versions up to 2018b.

### Dependencies

| Function | Description | Credit |
|---|---|---|
| `readhgt` | Downloads SRTM data | François Beauducel |
| `ll2utm`, `utm2ll` | Latitude/longitude to and from UTM coordinates | François Beauducel |
| `GUI Layout toolbox` | GUI tools | David Sampson |
| `lat_lon_proportions` | Constrains map proportions | Jonathan Sullivan |