A simple two-dimensional parameterisation for Flux Footprint Prediction (FFP)

For details of the derivation of the footprint parameterisation, see
Kljun, N., P. Calanca, M.W. Rotach, H.P. Schmid, 2015: A simple two-dimensional parameterisation for Flux Footprint Prediction (FFP). Geosci. Model Dev., 8, 3695-3713. doi:10.5194/gmd-8-3695-2015.

Please acknowledge the source of your footprint estimates by citing the above article. Thanks!

How to use FFP Matlab code

The FFP function is not meant to be a stand-alone function, but a function that can be called from within your own data processing code. For example, FFP can be called to calculate a series of footprints for a selected time series of your flux data.

1) Single footprint

To calculate a single FFP flux footprint, call **calc_footprint_FFP** as described below. To rotate a single flux footprint into the main wind direction, call calc_footprint_FFP with an optional input value for the wind direction. To derive the source area of R% of the flux footprint, call calc_footprint_FFP with an optional additional single value of R (e.g., 80 for 80%), or with an array of Rs (e.g., [20, 40, 60, 80] or [10:10:80]). You can also plot an example figure of your footprint by setting fig = 1.

[FFP,flag_err] = calc_footprint_FFP(zm,z0,umean,h,ol,sigmav,ustar,varargin)

FFP Input

All inputs as scalars

zm	= Measurement height above displacement height (i.e. z-d) [m]
z0	= Roughness length [m] - enter [NaN] if not known
umean	= Mean wind speed at zm [ms-1] - enter [NaN] if not known
h	= Boundary layer height [m]
ol	= Obukhov length [m]
sigmav	= Standard deviation of lateral velocity fluctuations [ms^{-1}]
ustar	= Friction velocity [ms^{-1}]

Note: Either z0 or umean is required. If both are given, z0 is selected to calculate the footprint.

Optional input (varargin):

Enter as calc_footprint_FFP(...,'OptionalInput',InputValue)

wind_dir	= Wind direction in degrees (of 360) for rotation of the footprint
r	= Percentage of source area, i.e. a value between 10% and 90%. Can be either a single value (e.g., "80") or an array of increasing percentage values (e.g., "[10:10:80]") Expressed either in percentages ("80") or in fractions of 1 ("0.8") Default is [10:10:80]. Set to "NaN" for no output of percentages
nx	= Integer scalar defining the number of grid elements of the scaled footprint. Large nx results in higher spatial resolution and higher computing time. Default is 1000, nx must be >=600.
rslayer	= Calculate footprint even if zm within roughness sublayer: set rslayer = 1. Note that this only

gives a rough estimate of the footprint as the model is not valid within the roughness sublayer. Default is 0 (i.e. no footprint for within RS). z0 is needed for estimation of the RS.

crop = Crop output area to size of the 80% footprint or the largest r given if crop=1

fig = Plot an example figure of the resulting footprint on the screen: set fig = 1. Default is 0 (i.e. no figure).

FFP output

FFP	= Structure array with footprint data for measurement at [0 0 zm] m
FFP.x_ci_max	= x location of footprint peak (distance from measurement) [m]
FFP.x_ci	= x array of crosswind integrated footprint [m]
FFP.f_ci	= Footprint function values of crosswind integrated footprint [m^{-1}]
FFP.x_2d	= x-grid of 2-dimensional footprint [m], rotated if wind_dir is provided
FFP.y_2d	= y-grid of 2-dimensional footprint [m], rotated if wind_dir is provided
FFP.f_2d	= Footprint function values of 2-dimensional footprint [m^{-2}]
FFP.r	= Percentage of footprint as in input, if provided
FFP.fr	= Footprint value at r, if r is provided
FFP.xr	= x-array for contour line of r, if r is provided
FFP.yr	= y-array for contour line of r, if r is provided For array of percentage values, structure entries can be accessed as FFP(1).r, FFP(1).xr
flag_err	= 1 in case of error, 0 otherwise

Example

```
[FFP,flag_err] = calc_footprint_FFP(20,0.1,NaN,2000,-100,0.6,0.4,'wind_dir',30,'r',[10:10:80])
```

2) Single footprint within a given, fixed domain

In some cases it may be useful to derive a footprint for a pre-set given domain. For such a case, use **calc_footprint_FFP_climatology** with a single set of input parameters. For details of input and output parameters, see Section 3 below.

3) Footprint climatology

A footprint climatology is an aggregation of footprints over several time steps. To calculate a footprint climatology with FFP, call **calc_footprint_FFP_climatology** as described below. Again, optional input parameters can be provided to rotate each single flux footprint of the footprint climatology into the wind direction of the corresponding time step. To derive the source area of R% of the flux footprint climatology, call calc_footprint_FFP_climatology with an optional additional single value of R (e.g., 80 for 80%), or with an array of Rs (e.g., [20, 40, 60, 80] or [10:10:80]). You can also plot an example figure of your footprint climatology by setting fig = 1.
This function calculates footprints within a fixed physical domain (either default area or user input). For determining the optimal extent of the domain (large enough to include the footprints) use the function calc_footprint_FFP as described in Section 1.

FFP Input

All vectors need to be of equal length (one value for each time step, scalars possible)

zm	= Measurement height above displacement height (i.e. z-d) [m] Usually a scalar, but can also be a vector
z0	= Roughness length [m] - enter [NaN] if not known Usually a scalar, but can also be a vector
umean	= Vector of mean wind speed at zm [ms^{-1}] - enter [NaN] if not known

Either z0 or umean is required. If both are given, z0 is selected to calculate the footprint

h = Vector of boundary layer height [m]

ol = Vector of Obukhov length [m]

sigmav = Vector of standard deviation of lateral velocity fluctuations [ms^{-1}]

ustar = Vector of friction velocity [ms^{-1}]

wind_dir = Vector of wind direction in degrees (of 360) for rotation of the footprint

Optional input (varargin):

Enter as calc_footprint_FFP_climatology(...,'OptionalInput',InputValue)

domain = Domain size as an array of [xmin xmax ymin ymax] [m].
Footprint will be calculated for a measurement at [0 0 zm] m
Default is smallest area including the r% footprint or [-1000 1000 -1000 1000]m, whichever smallest (80% footprint if r not given).

dx, dy = Cell size of domain [m]
Small dx,dy result in higher spatial resolution and higher computing time
Default is dx = dy = 2 m (if neither domain nor nx and ny are given).
If only dx is given, dx=dy.

nx, ny = Two integer scalars defining the number of grid elements in x and y
Large nx and ny result in higher spatial resolution and higher computing time
Default is nx = ny = 1000. If only nx is given, nx=ny
If dx,dy and nx,ny are given, dx,dy is given priority

r = Percentage of source area for which to provide contours, must be between 10% and 90%.
Can be either a single value (e.g., "80") or an array of percentage values (e.g., "[10:10:80]")
Expressed either in percentages ("80") or in fractions of 1 ("0.8")
Default is [10:10:80]. Set to "NaN" for no output of percentages

rslayer = Calculate footprint even if zm within roughness sublayer: set rslayer = 1. Note that this only gives a rough estimate of the footprint as the model is not valid within the roughness sublayer. Default is 0 (i.e. no footprint for within RS). z0 is needed for estimation of the RS.

smooth_data = Apply convolution filter to smooth footprint climatology if smooth_data=1 (default)

crop = Crop output area to size of the 80% footprint or the largest r given if crop=1

pulse = Display progress of footprint calculations every pulse-th footprint (e.g., "100")

fig = Plot an example figure of the resulting footprint on the screen: set fig = 1.
Default is 0 (i.e. no figure).

FFP output

FFP = Structure array with footprint climatology data for measurement at [0 0 zm] m

FFP.x_2d = x-grid of footprint climatology [m]

FFP.y_2d = y-grid of footprint climatology [m]

FFP.fclim_2d = Normalised footprint function values of footprint climatology [m^{-2}]

FFP.r = Percentage of footprint as in input, if provided

FFP.fr = Footprint value at r, if r is provided [m^{-2}]

FFP.xr = x-array for contour line of r, if r is provided [m]

FFP.yr = y-array for contour line of r, if r is provided [m]
For array of percentage values, structure entries can be accessed as FFP(1).r, FFP(1).xr

FFP.n = Number of footprints calculated and included in footprint climatology

flag_err = 1 in case of error, 2 if not all contour plots (r%) within specified domain, 0 otherwise
If the source area is calculated for 20%, 40%, 60% and 80%, and the 80% contour is extending further than the domain (but the other r's are within the domain), flag_err = 2 and all results are provided apart from those for the contour at 80%.

Example

```
zm=15; z0=0.01; umean=NaN;
h=[2000 1800 1500]; ol=[-10 -100 -500]; sigmav=[0.9 0.7 0.3]; ustar=[0.5 0.3 0.4]; wind_dir=[30 50 70];

[FFP,flag_err]=calc_footprint_FFP_climatology(zm,z0,umean,h,ol,sigmav,ustar,wind_dir, ...
'domain',[-100 1000 -100 1000],'nx',1100,'r',[10:10:80],'smooth_data',1)
```

4) Plotting footprints

To plot the footprint in matlab, type, for example:

Crosswind-integrated footprint

```
plot(FFP(1).x_ci, FFP(1).f_ci,'k-',[ FFP(1).x_ci_max FFP(1).x_ci_max],[0 max(FFP(1).f_ci)],'b--')
```

Two-dimensional footprint with contour lines at R = 10 to 80%

```
surf(FFP(1).x_2d, FFP(1).y_2d, FFP(1).f_2d); shading flat; view(2);
hold all;
for i=1:8
    z = FFP(i).fr.*10.*ones(size(FFP(i).yr));
    plot3(FFP(i).xr,FFP(i).yr,z,'r')
end
```

Three-dimensional footprint surface

```
surf(FFP(1).x_2d, FFP(1).y_2d, FFP(1).f_2d); shading flat; view(-10,20); box on; colormap jet
```

For the footprint climatology, f_2d above needs to be replaced by fclim_2d.

Please note that the plotting convention for matrices varies with software package or even with the selected plotting command, i.e. point (1/1) of the matrix may be the lower left corner or the upper left corner. It hence is suggested that **the footprint plot is always checked against a wind rose**. For complex footprint climatologies, it is sufficient to check just one single footprint. It may be necessary to transpose the footprint matrix depending on the plotting tool.

Note: This version of the code is developed and tested for Matlab version 2019b.