

# Package ‘fluxible’

April 22, 2025

**Title** Ecosystem Gas Fluxes Calculations for Closed Loop Chamber Setup

**Version** 1.1.0

**Date** 2025-04-07

**Description** Processes the raw data from closed loop flux chamber (or tent) setups into ecosystem gas fluxes usable for analysis. It goes from a data frame of gas concentration over time (which can contain several measurements) and a meta data file indicating which measurement was done when, to a data frame of ecosystem gas fluxes including quality diagnostics. Functions provided include different models (exponential as described in Zhao et al (2018) <[doi:10.1016/j.agrformet.2018.08.022](https://doi.org/10.1016/j.agrformet.2018.08.022)>, quadratic and linear) to estimate the fluxes from the raw data, quality assessment, plotting for visual check and calculation of fluxes based on the setup specific parameters (chamber size, plot area, ...).

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), vdiff, forcats, tidyverse, fs

**Config/testthat/edition** 3

**Imports** broom, dplyr, ggforce, ggplot2, haven, lubridate, rlang, purrr, stats, stringr, tidyr, zoo, progress, purrrlyr, tidyselect

**Depends** R (>= 4.1)

**LazyData** true

**URL** <https://plant-functional-trait-course.github.io/fluxible/>,  
<https://github.com/Plant-Functional-Trait-Course/fluxible>

**VignetteBuilder** knitr

**BugReports** <https://github.com/Plant-Functional-Trait-Course/fluxible/issues>

**NeedsCompilation** no

**Author** Joseph Gaudard [aut, cre] (<<https://orcid.org/0000-0002-6989-7624>>),  
Richard James Telford [aut] (<<https://orcid.org/0000-0001-9826-3076>>)

**Maintainer** Joseph Gaudard <joseph.gaudard@pm.me>

**Repository** CRAN

**Date/Publication** 2025-04-22 07:50:02 UTC

## Contents

co2_conc . . . . .	3
co2_conc_mid_missing . . . . .	4
co2_conc_missing . . . . .	5
co2_df_missing . . . . .	6
co2_df_short . . . . .	6
co2_fluxes . . . . .	7
co2_liahovden . . . . .	8
flux_calc . . . . .	8
flux_check_item . . . . .	10
flux_cut . . . . .	11
flux_fitting . . . . .	11
flux_fitting_exptz . . . . .	13
flux_fitting_hm . . . . .	14
flux_fitting_lm . . . . .	15
flux_fitting_quadratic . . . . .	16
flux_fitting_zhao18 . . . . .	16
flux_fit_type . . . . .	18
flux_flag_count . . . . .	18
flux_fun_check . . . . .	19
flux_gep . . . . .	20
flux_match . . . . .	21
flux_match_col . . . . .	22
flux_match_fixed . . . . .	23
flux_param_exp . . . . .	23
flux_param_kappamax . . . . .	24
flux_param_lm . . . . .	24
flux_param_qua . . . . .	25
flux_plot . . . . .	25
flux_plot_exp . . . . .	27
flux_plot_flag . . . . .	27
flux_plot_lin . . . . .	28
flux_plot_quadratic . . . . .	28
flux_quality . . . . .	29
flux_quality_exp . . . . .	32
flux_quality_kappamax . . . . .	33
flux_quality_lm . . . . .	34
flux_quality_qua . . . . .	35
raw_twogases . . . . .	37
record_liahovden . . . . .	37
record_short . . . . .	38
slopes0_temp . . . . .	38

<code>co2_conc</code>	3
<code>stupeflux</code> . . . . .	39
<code>twogases_record</code> . . . . .	43
<b>Index</b>	<b>45</b>

---

<code>co2_conc</code>	<i>CO2 concentration</i>
-----------------------	--------------------------

---

**Description**

CO2 concentration with measurements meta data

**Usage**

`co2_conc`

**Format**

A tibble with 1251 rows and 13 variables

- datetime** Datetime at which CO2 concentration was recorded.
- temp\_air** Air temperature inside the flux chamber in Celsius.
- temp\_soil** Ground temperature inside the flux chamber in Celsius.
- conc** CO2 concentration in ppm.
- PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.
- turfID** Unique ID of the turf in which the measurement took place.
- type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).
- f\_start** Datetime at which the measurement was started.
- f\_end** Datetime at which the measurement ended.
- f\_fluxid** Unique ID for each flux.
- f\_n\_conc** Number of data point per flux.
- f\_ratio** Ratio of n\_conc over length of the measurement (in seconds).
- f\_flag\_match** Data quality flags.

**Examples**

`co2_conc`

---

co2_conc_mid_missing	<i>CO2 concentration with missing data</i>
----------------------	--

---

## Description

CO2 concentration with measurements meta data and missing data in the middle of the measurements

## Usage

```
co2_conc_mid_missing
```

## Format

A tibble with 1251 rows and 13 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**f\_start** Datetime at which the measurement was started.

**f\_end** Datetime at which the measurement ended.

**f\_fluxid** Unique ID for each flux.

**f\_n\_conc** Number of data point per flux.

**f\_ratio** Ratio of n\_conc over length of the measurement (in seconds).

**f\_flag\_match** Data quality flags.

## Examples

```
co2_conc_mid_missing
```

---

co2_conc_missing	<i>CO2 concentration</i>
------------------	--------------------------

---

### Description

CO2 concentration with measurements meta data, with missing data.

### Usage

```
co2_conc_missing
```

### Format

A tibble with 668 rows and 13 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**f\_start** Datetime at which the measurement was started.

**f\_end** Datetime at which the measurement ended.

**f\_fluxid** Unique ID for each flux.

**f\_n\_conc** Number of data point per flux.

**f\_ratio** Ratio of n\_conc over length of the measurement (in seconds).

**f\_flag\_match** Data quality flags.

### Examples

```
co2_conc_missing
```

---

co2_df_missing	<i>CO2 concentration with missing data</i>
----------------	--

---

### Description

Continuous CO2 concentration as measured on the field, with missing data.

### Usage

co2\_df\_missing

### Format

A tibble with 1148 rows and 5 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

### Examples

co2\_df\_missing

---

co2_df_short	<i>CO2 concentration</i>
--------------	--------------------------

---

### Description

Continuous CO2 concentration as measured on the field

### Usage

co2\_df\_short

### Format

A tibble with 1801 rows and 5 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

**Examples**

```
co2_df_short
```

---

co2_fluxes	<i>CO2 fluxes</i>
------------	-------------------

---

**Description**

Calculated CO2 fluxes

**Usage**

```
co2_fluxes
```

**Format**

A tibble with 6 rows and 11 variables

**f\_fluxid** Unique ID for each flux.

**f\_slope\_tz** Slope of C(t) at t zero.

**f\_temp\_air\_ave** Air temperature inside the flux chamber in Celsius averaged over the flux measurement.

**f\_flux** CO2 flux in mmol/sqm/hour.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm averaged over the flux measurement.

**temp\_soil** Ground temperature inside the flux chamber in Celsius averaged over the flux measurement.

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**f\_start** Datetime at which the measurement started.

**temp\_fahr** Air temperature inside the flux chamber in Fahrenheit averaged over the flux measurement.

**temp\_kelvin** Air temperature inside the flux chamber in Kelvin averaged over the flux measurement.

**Examples**

```
co2_fluxes
```

---

co2_liahovden	<i>CO2 concentration at Liahovden</i>
---------------	---------------------------------------

---

### Description

CO2 concentration at Liahovden site, used in example in readme file

### Usage

```
co2_liahovden
```

### Format

A tibble with 89692 rows and 5 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

### Examples

```
co2_liahovden
```

---

flux_calc	<i>calculates ecosystem gas fluxes</i>
-----------	--

---

### Description

calculates a flux based on the rate of change of gas concentration over time

### Usage

```
flux_calc(
  slopes_df,
  slope_col,
  datetime_col,
  temp_air_col,
  chamber_volume,
  atm_pressure,
  plot_area,
  f_fluxid = f_fluxid,
  conc_unit,
  flux_unit,
```



```

    cols_keep = c(),
    cols_ave = c(),
    cols_sum = c(),
    cols_med = c(),
    tube_volume,
    temp_air_unit = "celsius",
    f_cut = f_cut,
    keep_arg = "keep",
    cut = TRUE,
    fit_type = c()
)

```

### Arguments

slopes_df	dataframe of flux slopes
slope_col	column containing the slope to calculate the flux (in $ppm * s^{-1}$ or $ppb * s^{-1}$ )
datetime_col	column containing the datetime of each gas concentration measurements in slopes_df. The first one after cutting will be kept as datetime of each flux in the output.
temp_air_col	column containing the air temperature used to calculate fluxes. Will be averaged with NA removed.
chamber_volume	volume of the flux chamber in L, can also be a column in case it is a variable
atm_pressure	atmospheric pressure, can be a constant (numerical) or a variable (column name)
plot_area	area of the plot in $m^2$ , can also be a column in case it is a variable
f_fluxid	column containing the flux IDs
conc_unit	unit in which the concentration of gas was measured ppm or ppb
flux_unit	unit in which the calculated flux will be mmol outputs fluxes in $mmol * m^{-2} * h^{-1}$ ; micromol outputs fluxes in $micromol * m^{-2} * h^{-1}$
cols_keep	columns to keep from the input to the output. Those columns need to have unique values for each flux, as <a href="#">distinct</a> is applied.
cols_ave	columns with values that should be averaged for each flux in the output. Note that NA are removed in mean calculation.
cols_sum	columns with values for which is sum is provided for each flux in the output. Note that NA are removed in sum calculation.
cols_med	columns with values for which is median is provided for each flux in the output. Note that NA are removed in median calculation.
tube_volume	volume of the tubing in L, can also be a column in case it is a variable
temp_air_unit	units in which air temperature was measured. Has to be either celsius (default), fahrenheit or kelvin.
f_cut	column containing cutting information
keep_arg	name in f_cut of data to keep
cut	if 'TRUE' (default), the measurements will be cut according to 'f_cut' before calculating fluxes. This has no influence on the flux itself since the slope is provided from <a href="#">flux_fitting</a> , but it will influence the values of the columns in cols_ave.

fit\_type (optional) model used in [flux\\_fitting](#). Will be automatically filled if slopes\_df was produced using [flux\\_fitting](#).

### Value

a dataframe containing flux IDs, datetime of measurements' starts, fluxes in  $\text{mmol} * \text{m}^{-2} * \text{h}^{-1}$  or  $\text{micromol} * \text{m}^{-2} * \text{h}^{-1}$  (f\_flux) according to flux\_unit, temperature average for each flux in Kelvin (f\_temp\_ave), the total volume of the setup for each measurement (f\_volume\_setup), the model used in [flux\\_fitting](#), any column specified in cols\_keep, any column specified in cols\_ave with their value averaged over the measurement after cuts and discarding NA.

### Examples

```
data(co2_conc)
slopes <- flux_fitting(co2_conc, conc, datetime, fit_type = "exp_zhao18")
flux_calc(slopes,
  f_slope,
  datetime,
  temp_air,
  conc_unit = "ppm",
  flux_unit = "mmol",
  chamber_volume = 24.5,
  tube_volume = 0.075,
  atm_pressure = 1,
  plot_area = 0.0625)
```

---

flux_check_item	<i>check the items inside flux_fun_check</i>
-----------------	--

---

### Description

check the items inside flux\_fun\_check

### Usage

```
flux_check_item(arg, fn, msg, nargs, df_name = NA)
```

### Arguments

arg	argument to be checked by fn
fn	function to check arg
msg	message to display in case arg is the wrong class
nargs	name of arg
df_name	name of arg in case it is a data frame

### Author(s)

Adam Klimes

---

flux_cut	<i>filter cut data before calculating fluxes</i>
----------	--

---

**Description**

filter cut data before calculating fluxes

**Usage**

```
flux_cut(slopes_df, cut_col, keep_arg)
```

**Arguments**

slopes_df	dataset containing slopes and cut column
cut_col	column containing cutting information
keep_arg	name in cut_col of data to keep

---

flux_fitting	<i>Fitting a model to concentration data and estimating the slope</i>
--------------	---

---

**Description**

fits gas concentration over time data with a model (exponential, quadratic or linear) and provides the slope later used to calculate gas fluxes with [flux\\_calc](#)

**Usage**

```
flux_fitting(  
  conc_df,  
  conc_col,  
  datetime_col,  
  f_start = f_start,  
  f_end = f_end,  
  f_fluxid = f_fluxid,  
  start_cut = 0,  
  end_cut = 0,  
  cz_window = 15,  
  b_window = 10,  
  a_window = 10,  
  roll_width = 15,  
  t_zero = 0,  
  fit_type  
)
```

**Arguments**

conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration data
datetime_col	column with datetime of each concentration measurement Note that if there are duplicated datetime in the same f_fluxid only the first row will be kept
f_start	column with datetime when the measurement started (ymd_hms)
f_end	column with datetime when the measurement ended (ymd_hms)
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)
end_cut	time to discard at the end of the measurements (in seconds)
cz_window	window used to calculate $C_z$ , at the beginning of cut window (exponential fit)
b_window	window to estimate $b$ . It is an interval after $t_z$ where it is assumed that the model fits the data perfectly (exponential fit)
a_window	window at the end of the flux to estimate $a$ (exponential fit)
roll_width	width of the rolling mean for $CO_2$ when looking for $t_z$ , ideally same as $cz\_window$ (exponential fit)
t_zero	time at which the slope should be calculated (for quadratic and exp_tz fits)
fit_type	exp_zhao18, exp_tz, exp_hm, quadratic or linear. exp_zhao18 is using the exponential model $C(t) = C_m + a(t - t_z) + (C_z - C_m) \exp(-b(t - t_z))$ from Zhao et al (2018). exp_tz is a modified version which allows the user to fix $t\_zero$ : $C(t) = C_m + a * t + (C_z - C_m) \exp(-b * t)$ . exp_hm is using the HM model (Pedersen et al., 2010; Hutchinson and Mosier, 1981) $C(t) = C_m + (C_z - C_m) \exp(-b * t)$

**Value**

a dataframe with the slope at  $t\_zero$  (f\_slope), a datetime column of  $t\_zero$  (f\_start\_z), a factor column indicating the cuts (f\_cut), the time in seconds since the start of the measurement (f\_time), the modeled fit (f\_fit), the modeled slope (f\_fit\_slope), the parameters of the fit depending on the model used, and any columns present in the input. The type of fit is added as an attribute for use by the other functions.

**References**

Pedersen, A.R., Petersen, S.O., Schelde, K., 2010. A comprehensive approach to soil-atmosphere trace-gas flux estimation with static chambers. *European Journal of Soil Science* 61, 888–902. <https://doi.org/10.1111/j.1365-2389.2010.01291.x>

Hutchinson, G.L., Mosier, A.R., 1981. Improved Soil Cover Method for Field Measurement of Nitrous Oxide Fluxes. *Soil Science Society of America Journal* 45, 311–316. <https://doi.org/10.2136/sssaj1981.0361599500450>

Zhao, P., Hammerle, A., Zeeman, M., Wohlfahrt, G., 2018. On the calculation of daytime  $CO_2$  fluxes measured by automated closed transparent chambers. *Agricultural and Forest Meteorology* 263, 267–275. <https://doi.org/10.1016/j.agrformet.2018.08.022> exponential is equal to exp\_zhao18, for backwards compatibility

**See Also**[selectfluxes HMR](#)**Examples**

```
data(co2_conc)
flux_fitting(co2_conc, conc, datetime, fit_type = "exp_zhao18")
flux_fitting(co2_conc, conc, datetime, fit_type = "quadratic",
t_zero = 10, end_cut = 30)
```

---

flux_fitting_exptz	<i>Fitting a model to the gas concentration curve and estimating the slope over time, using a modified version of the model from Zhao et al (2018) that allows the user to fix t_zero.</i>
--------------------	--

---

**Description**

Fits the exponential expression to the concentration evolution  $C(t) = C_m + a * t + (C_z - C_m) \exp(-b * t)$

**Usage**

```
flux_fitting_exptz(
  conc_df_cut,
  conc_df,
  conc_col,
  f_start,
  f_fluxid,
  start_cut,
  cz_window,
  b_window,
  a_window,
  roll_width,
  t_zero
)
```

**Arguments**

conc_df_cut	dataframe of gas concentration over time, cut
conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration
f_start	column with datetime when the measurement started
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)
cz_window	window used to calculate Cz, at the beginning of cut window

b_window	window to estimate b. It is an interval after tz where it is assumed that C fits the data perfectly
a_window	window at the end of the flux to estimate a
roll_width	width of the rolling mean for CO2 when looking for tz, ideally same as cz_window
t_zero	time at which the slope should be calculated (for quadratic fit)

**Value**

a dataframe with the slope at t zero, modeled concentration over time and exponential expression parameters

**References**

Zhao, P., Hammerle, A., Zeeman, M., Wohlfahrt, G., 2018. On the calculation of daytime CO2 fluxes measured by automated closed transparent chambers. Agricultural and Forest Meteorology 263, 267–275. <https://doi.org/10.1016/j.agrformet.2018.08.022>

---

flux_fitting_hm	<i>Fitting a model to the gas concentration curve and estimating the slope over time, using the HM model (Pedersen et al., 2010; Hutchinson and Mosier, 1981).</i>
-----------------	--

---

**Description**

Fits the exponential expression to the concentration evolution  $C(t) = C_m + (C_z - C_m) \exp(-b * t)$

**Usage**

```
flux_fitting_hm(  
  conc_df_cut,  
  conc_df,  
  conc_col,  
  f_start,  
  f_fluxid,  
  start_cut,  
  cz_window,  
  b_window,  
  roll_width,  
  t_zero  
)
```

**Arguments**

conc_df_cut	dataframe of gas concentration over time, cut
conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration

f_start	column with datetime when the measurement started
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)
cz_window	window used to calculate Cz, at the beginning of cut window
b_window	window to estimate b. It is an interval after tz where it is assumed that C fits the data perfectly
roll_width	width of the rolling mean for CO2 when looking for tz, ideally same as cz_window
t_zero	time at which the slope should be calculated (for quadratic fit)

**Value**

a dataframe with the slope at t zero, modeled concentration over time and exponential expression parameters

**References**

Pedersen, A.R., Petersen, S.O., Schelde, K., 2010. A comprehensive approach to soil-atmosphere trace-gas flux estimation with static chambers. *European Journal of Soil Science* 61, 888–902. <https://doi.org/10.1111/j.1365-2389.2010.01291.x>

Hutchinson, G.L., Mosier, A.R., 1981. Improved Soil Cover Method for Field Measurement of Nitrous Oxide Fluxes. *Soil Science Society of America Journal* 45, 311–316. <https://doi.org/10.2136/sssaj1981.0361599500450>

---

flux_fitting_lm	<i>linear fit to gas concentration over time</i>
-----------------	--

---

**Description**

fits a linear model to the gas concentration over time

**Usage**

```
flux_fitting_lm(conc_df_cut, conc_df, conc_col, f_fluxid, start_cut)
```

**Arguments**

conc_df_cut	dataframe of gas concentration over time, cut
conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)

**Value**

a df with the modeled gas concentration, slope, intercept, std error, r square and p value of the linear model

---

flux_fitting_quadratic	<i>quadratic fit to gas concentration over time</i>
------------------------	---

---

**Description**

fits a quadratic model to the gas concentration over time

**Usage**

```
flux_fitting_quadratic(  
  conc_df_cut,  
  conc_df,  
  conc_col,  
  f_start,  
  f_fluxid,  
  start_cut,  
  t_zero  
)
```

**Arguments**

conc_df_cut	dataframe of gas concentration over time, cut
conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration
f_start	column with datetime when the measurement started
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)
t_zero	time at which the slope should be calculated

**Value**

a df with the modeled gas concentration, slope, intercept, std error, r square and p value of the quadratic model

---

flux_fitting_zhao18	<i>Fitting a model to the gas concentration curve and estimating the slope over time, using the exponential model from Zhao et al (2018)</i>
---------------------	--

---

**Description**

Fits an exponential expression to the concentration evolution



**Usage**

```
flux_fitting_zhao18(  
  conc_df_cut,  
  conc_df,  
  conc_col,  
  f_start,  
  f_fluxid,  
  start_cut,  
  cz_window,  
  b_window,  
  a_window,  
  roll_width  
)
```

**Arguments**

conc_df_cut	dataframe of gas concentration over time, cut
conc_df	dataframe of gas concentration over time
conc_col	column with gas concentration
f_start	column with datetime when the measurement started
f_fluxid	column with ID of each flux
start_cut	time to discard at the start of the measurements (in seconds)
cz_window	window used to calculate Cz, at the beginning of cut window
b_window	window to estimate b. It is an interval after tz where it is assumed that C fits the data perfectly
a_window	window at the end of the flux to estimate a
roll_width	width of the rolling mean for CO2 when looking for tz, ideally same as cz_window

**Value**

a dataframe with the slope at t zero, modeled concentration over time and exponential expression parameters

**References**

Zhao, P., Hammerle, A., Zeeman, M., Wohlfahrt, G., 2018. On the calculation of daytime CO2 fluxes measured by automated closed transparent chambers. *Agricultural and Forest Meteorology* 263, 267–275. <https://doi.org/10.1016/j.agrformet.2018.08.022>

---

flux_fit_type	<i>to check the type of fit</i>
---------------	---------------------------------

---

### Description

extracts the type of fit that was applied in flux\_fitting or checks that the fit\_type provided by the user is compatible with Fluxible

### Usage

```
flux_fit_type(
  df,
  fit_type = c(),
  fit_type_list = c("exp_hm", "exp_tz", "exp_zhao18", "exponential", "linear",
    "quadratic")
)
```

### Arguments

df	any dataframe
fit_type	type of fit that was applied in flux_fitting. Needs to be filled only if the df was produced outside of the Fluxible workflow.
fit_type_list	list of fit types in use with Fluxible.

---

flux_flag_count	<i>counts quality flags</i>
-----------------	-----------------------------

---

### Description

provides a table of how many fluxes were attributed which quality flag. This function is incorporated in flux\_quality (output as a message) but can be used alone to extract a dataframe with the flag count.

### Usage

```
flux_flag_count(
  slopes_df,
  f_fluxid = f_fluxid,
  f_quality_flag = f_quality_flag,
  f_cut = f_cut,
  f_flags = c("ok", "discard", "zero", "force_discard", "start_error", "no_data",
    "force_ok", "force_zero", "force_lm"),
  cut_arg = "cut"
)
```

**Arguments**

slopes_df	dataframe of flux slopes
f_fluxid	column containing fluxes unique ID
f_quality_flag	column containing the quality flags
f_cut	column indicating which part of the flux is being cut
f_flags	list of flags used in the dataset (if different from default from flux_quality). If not provided, it will list only the flags that are present in the dataset (no showing 0).
cut_arg	argument defining that the data point should be cut out

**Value**

a dataframe with the number of fluxes for each quality flags and their proportion to the total

**Author(s)**

Vincent Belde

**Examples**

```
data(co2_conc)
slopes <- flux_fitting(co2_conc, conc, datetime, fit_type = "exp_zhao18")
slopes_flag <- flux_quality(slopes, conc)
flux_flag_count(slopes_flag)
```

---

flux_fun_check	<i>checking that arguments and columns are in the correct class</i>
----------------	---

---

**Description**

checking that arguments and columns are in the correct class

**Usage**

```
flux_fun_check(args, fn, msg, name_df = NA)
```

**Arguments**

args	list of arguments or dataframe to check
fn	list of functions used to check (is.numeric, is.character, ...)
msg	list of messages to return in case of failed check
name_df	in case args is a df with selected columns to check origdf is the original df to take the name from for a more obvious error message

**Author(s)**

Adam Klimes

flux\_gep

*Calculates GEP***Description**

to calculate gross ecosystem production (GEP) from net ecosystem (NEE) exchange and ecosystem respiration (ER) as  $GEP = NEE - ER$ . Datetime and other variables to keep will be taken from the NEE measurement. Fluxes presents in the dataset that are neither NEE nor ER (soilR, LRC or other) are not lost.

**Usage**

```
flux_gep(
  fluxes_df,
  type_col,
  datetime_col,
  f_flux = f_flux,
  id_cols,
  nee_arg = "NEE",
  er_arg = "ER",
  cols_keep = "none"
)
```

**Arguments**

fluxes_df	a dataframe containing NEE and ER
type_col	column containing type of flux (NEE or ER)
datetime_col	column containing start of measurement as datetime
f_flux	column containing flux values
id_cols	columns used to identify each pair of ER and NEE
nee_arg	argument designating NEE fluxes in type column
er_arg	argument designating ER fluxes in type column
cols_keep	columns to keep from fluxes_df. Values from NEE row will be filled in GEP row. none (default) keeps only the columns in id_cols, flux, type and datetime columns; all keeps all the columns; can also be a vector of column names.

**Value**

a dataframe with  $GEP = NEE - ER$  in long format with GEP, NEE, and ER as flux type, datetime, and any column specified in cols\_keep. Values of datetime and columns in cols\_keep for GEP row are taken from NEE measurements.

**Examples**

```
data(co2_fluxes)
flux_gep(co2_fluxes, type, f_start, id_cols = "turfID",
  cols_keep = c("temp_soil"))
```

---

flux_match	<i>Matching continuously measured fluxes with measurement IDs and meta data</i>
------------	---

---

## Description

Matching a dataframe of continuously measured gas concentration data with measurement metadata from another dataframe. Measurements are paired with their metadata based on datetime. Extra variables in both dataframes are kept in the output.

## Usage

```
flux_match(
  raw_conc,
  field_record,
  datetime_col,
  start_col,
  conc_col,
  end_col,
  startcrop,
  measurement_length,
  fixed_length = TRUE,
  ratio_threshold = 0.5,
  time_diff = 0
)
```

## Arguments

raw_conc	dataframe of CO2 concentration measured continuously. Has to contain at least a datetime column in ymd_hms format and a gas concentration column as double.
field_record	dataframe recording which measurement happened when. Has to contain at least a column containing the start of each measurement, and any other column identifying the measurements.
datetime_col	datetime column in raw_conc (ymd_hms format)
start_col	start column in field_record (ymd_hms format)
conc_col	concentration column in raw_conc
end_col	end column in field_record (ymd_hms format)
startcrop	how many seconds should be discarded at the beginning of the measurement
measurement_length	length of the measurement (in seconds) from the start specified in the field_record
fixed_length	if TRUE (default), the measurement_length is used to create the end column. If FALSE, end_col has to be provided.

ratio_threhold	ratio (number of concentration measurement compared to length of measurement in seconds) below which the data should be flagged as too little
time_diff	time difference (in seconds) between the two datasets. Will be added to the datetime column of the raw_conc dataset. For situations where the time was not synchronized correctly.

Value

a dataframe with concentration measurements, corresponding datetime, flux ID (f\_fluxid), measurements start (f\_start) and end (f\_end), flags in case of no data or low number of data (f\_flag\_match), the number of datapoints per measurement (f\_n\_conc), the ratio of number of datapoints over the length of each measurement in seconds (f\_ratio), and any variables present in one of the inputs.

Examples

```
data(co2_df_short, record_short)
flux_match(co2_df_short, record_short, datetime, start, conc, startcrop = 10,
measurement_length = 180)
```

---

flux_match_col	<i>Using an already existing end column to slice measurements</i>
----------------	---

---

Description

Provides the f\_end column for flux\_match

Usage

```
flux_match_col(field_record, start_col, end_col, name_field_record)
```

Arguments

field_record	dataframe recording which measurement happened when. Has to contain at least a column containing the start of each measurement, and any other column identifying the measurements.
start_col	start column in field_record (ymd_hms format)
end_col	end column in field_record (ymd_hms format)
name_field_record	name of the df (for error message)

---

flux_match_fixed	<i>Using a fixed measurement length to slice the measurements</i>
------------------	---

---

**Description**

Provides the f\_end column for flux\_match

**Usage**

```
flux_match_fixed(field_record, start_col, measurement_length)
```

**Arguments**

field_record	dataframe recording which measurement happened when. Has to contain at least a column containing the start of each measurement, and any other column identifying the measurements.
start_col	start column in field_record (ymd_hms format)
measurement_length	length of the measurement (in seconds) from the start specified in the field_record

---

flux_param_exp	<i>prepares text to print for flux_plot function</i>
----------------	--

---

**Description**

creates a df with quality flags and quality diagnostics to print on the plots produced by flux\_plot. flux\_param\_lm is for fit in the lm family (linear and quadratic) flux\_param\_exp is for the exponential fit

**Usage**

```
flux_param_exp(slopes_df, conc_col)
```

**Arguments**

slopes_df	the slopes_df that is being provided to flux_plot
conc_col	column with gas concentration

---

flux_param_kappamax	<i>prepares text to print for flux_plot function</i>
---------------------	--

---

### Description

creates a df with quality flags and quality diagnostics to print on the plots produced by flux\_plot. flux\_param\_lm is for fit in the lm family (linear and quadratic) flux\_param\_exp is for the exponential fit

### Usage

```
flux_param_kappamax(slopes_df, conc_col)
```

### Arguments

slopes_df	the slopes_df that is being provided to flux_plot
conc_col	column with gas concentration

---

flux_param_lm	<i>prepares text to print in flux_plot</i>
---------------	--

---

### Description

creates a df with quality flags and quality diagnostics to print on the plots produced by flux\_plot. flux\_param\_lm is for fit in the lm family (linear and quadratic) flux\_param\_exp is for the exponential fit

### Usage

```
flux_param_lm(slopes_df, conc_col)
```

### Arguments

slopes_df	the slopes_df that is being provided to flux_plot
conc_col	column with gas concentration



---

flux_param_qua	<i>prepares text to print in flux_plot</i>
----------------	--

---

### Description

creates a df with quality flags and quality diagnostics to print on the plots produced by flux\_plot. flux\_param\_lm is for fit in the lm family (linear and quadratic) flux\_param\_exp is for the exponential fit

### Usage

```
flux_param_qua(slopes_df, conc_col)
```

### Arguments

slopes_df	the slopes_df that is being provided to flux_plot
conc_col	column with gas concentration

---

flux_plot	<i>plotting fluxes for visual evaluation</i>
-----------	--

---

### Description

plots the fluxes, fit and slope in facets with color code indicating quality flags This function takes time to run and is optional in the workflow, but it is still highly recommended to use it to visually check the measurements. Note that 'flux\_plot' is specific to the [flexible](#) package and will work best with datasets produced following a flexible workflow.

### Usage

```
flux_plot(
  slopes_df,
  conc_col,
  datetime_col,
  color_discard = "#D55E00",
  color_cut = "#D55E00",
  color_ok = "#009E73",
  color_zero = "#CC79A7",
  scale_x_datetime_args = list(date_breaks = "1 min", minor_breaks = "10 sec",
    date_labels = "%e/%m \n %H:%M"),
  f_ylim_upper = 800,
  f_ylim_lower = 400,
  f_plotname = "",
  facet_wrap_args = list(ncol = 4, nrow = 3, scales = "free"),
  y_text_position = 500,
```

```

    print_plot = "FALSE",
    output = "print_only",
    ggsave_args = list()
  )

```

## Arguments

slopes_df	dataset containing slopes, with flags produced by <a href="#">flux_quality</a>
conc_col	column with gas concentration
datetime_col	column with datetime of each data point
color_discard	color for fits with a discard quality flag
color_cut	color for the part of the flux that is cut
color_ok	color for fits with an ok quality flag
color_zero	color for fits with a zero quality flag
scale_x_datetime_args	list of arguments for <a href="#">scale_x_datetime</a>
f_ylim_upper	y axis upper limit
f_ylim_lower	y axis lower limit
f_plotname	filename for the extracted pdf file; if empty, the name of slopes_df will be used
facet_wrap_args	list of arguments for <a href="#">facet_wrap_paginate</a>
y_text_position	position of the text box
print_plot	logical, if TRUE it prints the plot as a ggplot object but will take time depending on the size of the dataset
output	pdfpages, the plots are saved as A4 landscape pdf pages; ggsave, the plots can be saved with the ggsave function; print_only (default) prints the plot without creating a file (independently from 'print_plot' being TRUE or FALSE)
ggsave_args	list of arguments for <a href="#">ggsave</a> (in case output = "ggsave")

## Value

plots of fluxes, with raw concentration data points, fit, slope, and color code indicating quality flags and cuts. The plots are organized in facets according to flux ID, and a text box display the quality flag and diagnostics of each measurement. The plots are returned as a ggplot object if print\_plot = TRUE; if print\_plot = FALSE it will not return anything but will produce a file according to the output argument.

## Examples

```

data(co2_conc)
slopes <- flux_fitting(co2_conc, conc, datetime, fit_type = "exp_zhao18")
slopes_flag <- flux_quality(slopes, conc)
flux_plot(slopes_flag, conc, datetime)

```

---

flux_plot_exp	<i>plotting fluxes with exponential fit</i>
---------------	---

---

**Description**

plots the fluxes that were fitted with an exponential model

**Usage**

```
flux_plot_exp(slopes_df, conc_col, datetime_col, y_text_position)
```

**Arguments**

- slopes\_df        dataset containing slopes
- conc\_col        column with gas concentration
- datetime\_col    column with datetime of each data point
- y\_text\_position        position of the text box

---

flux_plot_flag	<i>creates the flag column to be used by flux_plot</i>
----------------	--

---

**Description**

creates a column with quality flags (from flux\_quality) for the part of the rows to be kept, and cut flag for rows to be discarded

**Usage**

```
flux_plot_flag(slopes_df, param_df)
```

**Arguments**

- slopes\_df        as provided in flux\_plot
- param\_df        as provided by flux\_param

---

flux_plot_lin	<i>plotting fluxes with linear fit</i>
---------------	--

---

**Description**

plots the fluxes that were fitted with a linear model

**Usage**

```
flux_plot_lin(slopes_df, conc_col, datetime_col, y_text_position)
```

**Arguments**

- slopes\_df        dataset containing slopes
- conc\_col        column with gas concentration
- datetime\_col    column with datetime of each data point
- y\_text\_position        position of the text box

---

flux_plot_quadratic	<i>plotting fluxes with a quadratic fit</i>
---------------------	---

---

**Description**

specific part of flux\_plot for quadratic fit

**Usage**

```
flux_plot_quadratic(slopes_df, conc_col, datetime_col, y_text_position)
```

**Arguments**

- slopes\_df        dataset containing slopes
- conc\_col        column with gas concentration
- datetime\_col    column with datetime of each data point
- y\_text\_position        position of the text box

---

flux_quality	<i>assessing quality of slopes calculated with <a href="#">flux_fitting</a></i>
--------------	---

---

### Description

indicates if slopes should be discarded or replaced by 0 according to quality thresholds set by user

### Usage

```
flux_quality(  
  slopes_df,  
  conc_col,  
  f_fluxid = f_fluxid,  
  f_slope = f_slope,  
  f_time = f_time,  
  f_start = f_start,  
  f_end = f_end,  
  f_fit = f_fit,  
  f_cut = f_cut,  
  f_pvalue = f_pvalue,  
  f_rsquared = f_rsquared,  
  f_slope_lm = f_slope_lm,  
  f_fit_lm = f_fit_lm,  
  f_b = f_b,  
  force_discard = c(),  
  force_ok = c(),  
  force_zero = c(),  
  force_lm = c(),  
  force_exp = c(),  
  ratio_threshold = 0,  
  gfactor_threshold = 10,  
  fit_type = c(),  
  ambient_conc = 421,  
  error = 100,  
  pvalue_threshold = 0.3,  
  rsquared_threshold = 0.7,  
  rmse_threshold = 25,  
  cor_threshold = 0.5,  
  b_threshold = 1,  
  cut_arg = "cut",  
  instr_error = 5,  
  kappamax = FALSE  
)
```

### Arguments

slopes\_df      dataset containing slopes

conc_col	column containing the measured gas concentration (exponential fits)
f_fluxid	column containing unique IDs for each flux
f_slope	column containing the slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_time	column containing the time of each measurement in seconds (exponential fits)
f_start	column with datetime of the start of the measurement (after cuts)
f_end	column with datetime of the end of the measurement (after cuts)
f_fit	column containing the modeled data (exponential fits)
f_cut	column containing the cutting information
f_pvalue	column containing the p-value of each flux (linear and quadratic fits)
f_rsquared	column containing the r squared of each flux (linear and quadratic fits)
f_slope_lm	column containing the linear slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_fit_lm	column with the fit of the linear model. (as calculated by the <a href="#">flux_fitting</a> function)
f_b	column containing the b parameter of the exponential expression (exponential fits)
force_discard	vector of fluxIDs that should be discarded by the user's decision
force_ok	vector of fluxIDs for which the user wants to keep the calculated slope despite a bad quality flag
force_zero	vector of fluxIDs that should be replaced by zero by the user's decision
force_lm	vector of fluxIDs for which the linear slope should be used by the user's decision
force_exp	vector of fluxIDs for which the exponential slope should be used by the user's decision (kappamax method)
ratio_threshold	ratio of gas concentration data points over length of measurement (in seconds) below which the measurement will be considered as not having enough data points to be considered for calculations
gfactor_threshold	threshold for the g-factor. Defines a window with its opposite outside which the flux will be flagged discard (exponential quadratic fits).
fit_type	model fitted to the data, linear, quadratic or exponential. Will be automatically filled if slopes_df was produced using <a href="#">flux_fitting</a>
ambient_conc	ambient gas concentration in ppm at the site of measurement (used to detect measurement that started with a polluted setup)
error	error of the setup, defines a window outside of which the starting values indicate a polluted setup
pvalue_threshold	threshold of p-value below which the change of gas concentration over time is considered not significant (linear and quadratic fits)

rsquared_threshold	threshold of r squared value below which the linear model is considered an unsatisfactory fit (linear and quadratic fits)
rmse_threshold	threshold for the RMSE of each flux above which the fit is considered unsatisfactory (exponential fits)
cor_threshold	threshold for the correlation coefficient of gas concentration with time below which the correlation is considered not significant (exponential fits)
b_threshold	threshold for the b parameter. Defines a window with its opposite inside which the fit is considered good enough (exponential fits)
cut_arg	argument defining that the data point should be cut out
instr_error	error of the instrument, in the same unit as the gas concentration
kappamax	logical. If TRUE the kappamax method will be applied.

### Details

the kappamax method (Hüppi et al., 2018) selects the linear slope if  $|b| > kappamax$ , with  $kappamax = |slope_{lm}/instr\_error|$ . The original kappamax method was applied to the HMR model (Pedersen et al., 2010; Hutchinson and Mosier, 1981), but here it can be applied to any exponential fit.

### Value

a dataframe with added columns of quality flags (f\_quality\_flag), the slope corrected according to the quality flags (f\_slope\_corr), and any columns present in the input. It will also print a summary of the quality flags. This summary can also be exported as a dataframe using [flux\\_flag\\_count](#)

### References

- Pedersen, A.R., Petersen, S.O., Schelde, K., 2010. A comprehensive approach to soil-atmosphere trace-gas flux estimation with static chambers. *European Journal of Soil Science* 61, 888–902. <https://doi.org/10.1111/j.1365-2389.2010.01291.x>
- Hüppi, R., Felber, R., Krauss, M., Six, J., Leifeld, J., Fuß, R., 2018. Restricting the nonlinearity parameter in soil greenhouse gas flux calculation for more reliable flux estimates. *PLOS ONE* 13, e0200876. <https://doi.org/10.1371/journal.pone.0200876>
- Hutchinson, G.L., Mosier, A.R., 1981. Improved Soil Cover Method for Field Measurement of Nitrous Oxide Fluxes. *Soil Science Society of America Journal* 45, 311–316.

### See Also

[selectfluxes](#)

### Examples

```
data(co2_conc)
slopes <- flux_fitting(co2_conc, conc, datetime, fit_type = "exp_zhao18")
flux_quality(slopes, conc)
```

---

flux_quality_exp	<i>quality assessment for the slopes estimated by flux_fitting</i>
------------------	--

---

### Description

indicates if fluxes should be discarded or replaced by 0 according to parameters set by user. flux\_quality\_lm is for the model of the lm family. flux\_quality\_exp is for the exponential model.

### Usage

```
flux_quality_exp(
  slopes_df,
  conc_col,
  f_fluxid,
  f_slope,
  f_time,
  f_fit,
  f_cut,
  f_slope_lm,
  f_b,
  force_discard,
  force_ok,
  force_zero,
  force_lm,
  gfactor_threshold,
  rmse_threshold,
  cor_threshold,
  b_threshold,
  name_df
)
```

### Arguments

slopes_df	dataset containing slopes, fluxID, and parameters of the exponential expression
conc_col	column with gas concentration
f_fluxid	column of ID for each measurement
f_slope	column containing the slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_time	column containing the time of each measurement in seconds
f_fit	column containing the modeled data
f_cut	column containing the cutting information
f_slope_lm	column containing the linear slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_b	column containing the b parameter of the exponential expression



force_discard	vector of fluxIDs that should be discarded by the user's decision
force_ok	vector of fluxIDs for which the user wants to keep the calculated slope despite a bad quality flag
force_zero	vector of fluxIDs that should be replaced by zero by the user's decision
force_lm	vector of fluxIDs for which the linear slope should be used by the user's decision
gfactor_threshold	threshold for the g-factor. Defines a window with its opposite outside which the flux will be flagged discard.
rmse_threshold	threshold for the RMSE of each flux above which the fit is considered unsatisfactory
cor_threshold	threshold for the correlation coefficient of gas concentration with time below which the correlation is considered non significant
b_threshold	threshold for the b parameter. Defines a window with its opposite inside which the fit is considered good enough.
name_df	name of slopes_df

**Value**

same dataframe with added flag and corrected slopes columns

---

flux\_quality\_kappamax    *selecting linear slope with kappamax method*

---

**Description**

selecting linear slope with kappamax method

**Usage**

```
flux_quality_kappamax(
  slopes_df,
  f_slope,
  f_fit,
  f_fluxid,
  f_slope_lm,
  f_fit_lm,
  f_b,
  force_exp,
  fit_type,
  instr_error,
  name_df
)
```

**Arguments**

slopes_df	dataset containing slopes
f_slope	column containing the slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_fit	column containing the modeled data (exponential fits)
f_fluxid	column of ID for each measurement
f_slope_lm	column containing the linear slope of each flux
f_fit_lm	column with the fit of the linear model.
f_b	column containing the b parameter of the exponential expression
force_exp	vector of fluxIDs for which the exponential slope should be used by the user's decision (kappamax method)
fit_type	model fitted to the data, linear, quadratic or exponential. Will be automatically filled if slopes_df was produced using <a href="#">flux_fitting</a>
instr_error	error of the instrument, in the same unit as the gas concentration
name_df	name of slopes_df

---

flux_quality_lm	<i>quality assessment for the slopes estimated by flux_fitting</i>
-----------------	--

---

**Description**

indicates if fluxes should be discarded or replaced by 0 according to parameters set by user. flux\_quality\_lm is for the model of the lm family. flux\_quality\_exp is for the exponential model.

**Usage**

```
flux_quality_lm(
  slopes_df,
  conc_col,
  f_fluxid,
  f_slope,
  f_cut,
  f_pvalue,
  f_rsquared,
  force_discard,
  force_ok,
  force_zero,
  pvalue_threshold,
  rsquared_threshold,
  name_df
)
```

**Arguments**

slopes_df	dataset containing slopes, fluxID, p.value and r.squared
conc_col	column with gas concentration
f_fluxid	column of ID for each measurement
f_slope	column containing the slope of each flux (as calculated by the flux_fitting function)
f_cut	column containing the cutting information
f_pvalue	column containing the p-value of each flux
f_rsquared	column containing the r squared to be used for the quality assessment
force_discard	vector of fluxIDs that should be discarded by the user's decision
force_ok	vector of fluxIDs for which the user wants to keep the calculated slope despite a bad quality flag
force_zero	vector of fluxIDs that should be replaced by zero by the user's decision
pvalue_threshold	threshold of p-value below which the change of gas concentration over time is considered not significant (user decided)
rsquared_threshold	threshold of r squared value below which the linear model is considered an unsatisfactory fit
name_df	name of slopes_df (used for error message)

**Value**

same dataframe with added flag and corrected slopes columns

---

flux_quality_qua	<i>quality assessment for the slopes estimated by flux_fitting</i>
------------------	--

---

**Description**

indicates if fluxes should be discarded or replaced by 0 according to parameters set by user. flux\_quality\_lm is for the model of the lm family. flux\_quality\_exp is for the exponential model.

**Usage**

```
flux_quality_qua(
  slopes_df,
  conc_col,
  f_fluxid,
  f_slope,
  f_cut,
  f_pvalue,
  f_rsquared,
```

```

    f_slope_lm,
    force_discard,
    force_ok,
    force_zero,
    force_lm,
    gfactor_threshold,
    pvalue_threshold,
    rsquared_threshold,
    name_df
)

```

### Arguments

slopes_df	dataset containing slopes, fluxID, p.value and r.squared
conc_col	column with gas concentration
f_fluxid	column of ID for each measurement
f_slope	column containing the slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
f_cut	column containing the cutting information
f_pvalue	column containing the p-value of each flux
f_rsquared	column containing the r squared to be used for the quality assessment
f_slope_lm	column containing the linear slope of each flux (as calculated by the <a href="#">flux_fitting</a> function)
force_discard	vector of fluxIDs that should be discarded by the user's decision
force_ok	vector of fluxIDs for which the user wants to keep the calculated slope despite a bad quality flag
force_zero	vector of fluxIDs that should be replaced by zero by the user's decision
force_lm	vector of fluxIDs for which the linear slope should be used by the user's decision
gfactor_threshold	threshold for the g-factor. Defines a window with its opposite outside which the flux will be flagged discard.
pvalue_threshold	threshold of p-value below which the change of gas concentration over time is considered not significant (user decided)
rsquared_threshold	threshold of r squared value below which the linear model is considered an unsatisfactory fit
name_df	name of slopes_df (used for error message)

### Value

same dataframe with added flag and corrected slopes columns

---

raw_twogases	<i>CO2 and CH4 concentration</i>
--------------	----------------------------------

---

**Description**

CO2 and CH4 measured simultaneously

**Usage**

```
raw_twogases
```

**Format**

A tibble with 21681 rows and 4 variables

**co2\_conc** CO2 concentration in ppm

**ch4\_conc** CH4 concentration in ppb

**datetime** Datetime on the datapoint

**temp\_air** Air temperature inside the chamber in Celsius

**Examples**

```
raw_twogases
```

---

record_liahovden	<i>Measurements meta data at Liahovden</i>
------------------	--

---

**Description**

Measurements meta data as recorded on the field at site Liahovden

**Usage**

```
record_liahovden
```

**Format**

A tibble with 138 rows and 3 variables

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**round** Round of measurement.

**start** Datetime at which the measurement was started.

**Examples**

```
record_liahovden
```

---

record_short	<i>Measurements meta data</i>
--------------	-------------------------------

---

### Description

Measurements meta data as recorded on the field

### Usage

record\_short

### Format

A tibble with 6 rows and 3 variables

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**start** Datetime at which the measurement was started.

### Examples

record\_short

---

slopes0_temp	<i>Slopes for each flux</i>
--------------	-----------------------------

---

### Description

Slopes of C(t) for each flux with air temperature in various units.

### Usage

slopes0\_temp

### Format

A tibble with 1251 rows and 29 variables

**datetime** Datetime at which CO2 concentration was recorded.

**temp\_air** Air temperature inside the flux chamber in Celsius.

**temp\_soil** Ground temperature inside the flux chamber in Celsius.

**conc** CO2 concentration in ppm.

**PAR** Photosynthetically active radiation inside the chamber in micromol/s/sqm.

**turfID** Unique ID of the turf in which the measurement took place.

**type** Type of measurement: ecosystems respiration (ER) or net ecosystem exchange (NEE).

**f\_start** Datetime at which the measurement was started.

**f\_end** Datetime at which the measurement ended.

**f\_fluxid** Unique ID for each flux.

**f\_ratio** Ratio of number of datapoints over length of measurement in seconds.

**f\_flag\_match** Flags from flux\_match.

**f\_time** Time variable of the flux in seconds.

**f\_cut** Indicating if the measurement should be kept (keep) or discarded (cut).

**f\_Cz** Cz parameter of the C(t) function.

**f\_Cm** Cm parameter of the C(t) function, calculated by optim() with Cm\_est as starting point.

**f\_a** a parameter of the C(t) function, calculated by optim() with a\_est as starting point.

**f\_b** b parameter of the C(t) function, calculated by optim() with b\_est as starting point.

**f\_tz** tz parameter of the C(t) function, calculated by optim() with tz\_est as starting point.

**f\_slope** Slope of C(t) at tz

**f\_fit** C(t), modeled CO2 concentration as a function of time.

**f\_fit\_slope** Output of linear model of CO2 concentration passing by C(tz) and a slope of slope\_tz.

**f\_start\_z** Datetime format of tz

**f\_cor\_coef** Correlation coefficient of concentration over time.

**f\_RMSE** RMSE of the fit.

**f\_quality\_flag** Quality flags according to flux\_quality.

**f\_slope\_corr** Slope as advised by quality flags.

**temp\_fahr** Air temperature inside the flux chamber in Fahrenheit averaged over the flux measurement.

**temp\_kelvin** Air temperature inside the flux chamber in Kelvin averaged over the flux measurement.

## Examples

```
slopes0_temp
```

---

stupeflux

*From raw gas concentration over time to clean fluxes*

---

## Description

Wrapper function for the Fluxible workflow. We recommend using the step-by-step workflow for more control over the process.

**Usage**

```
stupeflux(  
  raw_conc,  
  field_record,  
  datetime_col,  
  start_col,  
  end_col,  
  conc_col,  
  startcrop,  
  measurement_length,  
  fit_type,  
  temp_air_col,  
  chamber_volume,  
  atm_pressure,  
  plot_area,  
  conc_unit,  
  flux_unit,  
  fixed_length = TRUE,  
  cols_keep = c(),  
  cols_ave = c(),  
  cols_sum = c(),  
  cols_med = c(),  
  tube_volume,  
  ratio_threshold = 0.5,  
  time_diff = 0,  
  start_cut = 0,  
  end_cut = 0,  
  cz_window = 15,  
  b_window = 10,  
  a_window = 10,  
  roll_width = 15,  
  t_zero = 0,  
  force_discard = c(),  
  force_ok = c(),  
  force_zero = c(),  
  ambient_conc = 421,  
  error = 100,  
  pvalue_threshold = 0.3,  
  rsquared_threshold = 0.7,  
  rmse_threshold = 25,  
  cor_threshold = 0.5,  
  b_threshold = 1,  
  temp_air_unit = "celsius",  
  cut = TRUE,  
  slope_correction = TRUE  
)
```



**Arguments**

raw_conc	dataframe of CO2 concentration measured continuously. Has to contain at least a datetime column in ymd_hms format and a gas concentration column as double.
field_record	dataframe recording which measurement happened when. Has to contain at least a column containing the start of each measurement, and any other column identifying the measurements.
datetime_col	datetime column in raw_conc (dmy_hms format)
start_col	start column in field_record (dmy_hms format)
end_col	end column in field_record (ymd_hms format)
conc_col	concentration column in raw_conc
startcrop	how many seconds should be discarded at the beginning of the measurement
measurement_length	length of the measurement (in seconds) from the start specified in the field_record
fit_type	exp_zhao18, exp_tz, quadratic or linear. exp_zhao18 is using the exponential model $C(t) = C_m + a (t - t_z) + (C_z - C_m) \exp(-b (t - t_z))$ from Zhao et al (2018). exp_tz is a modified version which allows the user to fix t_zero: $C(t) = C_m + a * t + (C_z - C_m) \exp(-b * t)$ .
temp_air_col	column containing the air temperature used to calculate fluxes. Will be averaged with NA removed.
chamber_volume	volume of the flux chamber in L, can also be a column in case it is a variable
atm_pressure	atmospheric pressure, can be a constant (numerical) or a variable (column name)
plot_area	area of the plot in m^2, can also be a column in case it is a variable
conc_unit	unit in which the concentration of gas was measured ppm or ppb
flux_unit	unit in which the calculated flux will be mmol outputs fluxes in $\text{mmol} * \text{m}^{(-2)*\text{h}^{(-1)}}$ ; micromol outputs fluxes in $\text{micromol} * \text{m}^{(-2)*\text{h}^{(-1)}}$
fixed_length	if TRUE (default), the measurement_length is used to create the end column. If FALSE, end_col has to be provided.
cols_keep	columns to keep from the input to the output. Those columns need to have unique values for each flux, as distinct() is applied.
cols_ave	columns with values that should be averaged for each flux in the output. Note that NA are removed in mean calculation.
cols_sum	columns with values for which is sum is provided for each flux in the output. Note that NA are removed in sum calculation.
cols_med	columns with values for which is median is provided for each flux in the output. Note that NA are removed in median calculation.
tube_volume	volume of the tubing in L, can also be a column in case it is a variable
ratio_threshold	ratio of gas concentration data points over length of measurement (in seconds) below which the measurement will be considered as not having enough data points to be considered for calculations

time_diff	time difference (in seconds) between the two datasets. Will be added to the datetime column of the raw_conc dataset. For situations where the time was not synchronized correctly.
start_cut	time to discard at the start of the measurements (in seconds)
end_cut	time to discard at the end of the measurements (in seconds)
cz_window	window used to calculate Cz, at the beginning of cut window (exponential fit)
b_window	window to estimate b. It is an interval after tz where it is assumed that the model fits the data perfectly (exponential fit)
a_window	window at the end of the flux to estimate a (exponential fit)
roll_width	width of the rolling mean for CO2 when looking for tz, ideally same as cz_window (exponential fit)
t_zero	time at which the slope should be calculated (for quadratic and exp_tz fits)
force_discard	vector of fluxIDs that should be discarded by the user's decision
force_ok	vector of fluxIDs for which the user wants to keep the calculated slope despite a bad quality flag
force_zero	vector of fluxIDs that should be replaced by zero by the user's decision
ambient_conc	ambient gas concentration in ppm at the site of measurement (used to detect measurement that started with a polluted setup)
error	error of the setup, defines a window outside of which the starting values indicate a polluted setup
pvalue_threshold	threshold of p-value below which the change of gas concentration over time is considered not significant (linear and quadratic fit)
rsquared_threshold	threshold of r squared value below which the linear model is considered an unsatisfactory fit (linear and quadratic fit)
rmse_threshold	threshold for the RMSE of each flux above which the fit is considered unsatisfactory (exponential fit)
cor_threshold	threshold for the correlation coefficient of gas concentration with time below which the correlation is considered not significant (exponential fit)
b_threshold	threshold for the b parameter. Defines a window with its opposite inside which the fit is considered good enough (exponential fit)
temp_air_unit	units in which air temperature was measured. Has to be either celsius (default), fahrenheit or kelvin.
cut	if 'TRUE' (default), the measurements will be cut according to 'f_cut' before calculating fluxes. This has no influence on the flux itself since the slope is provided from <a href="#">flux_fitting</a> , but it will influence the values of the columns in cols_ave.
slope_correction	logical. If TRUE, the flux will be calculated with the slope corrected according to the recommendations of the quality flags.

Value

a dataframe containing flux IDs, datetime of measurements' starts, fluxes in  $\text{mmol}\cdot\text{m}^{-2}\cdot\text{h}^{-1}$  or  $\text{micromol}\cdot\text{m}^{-2}\cdot\text{h}^{-1}$  (f\_flux) according to flux\_unit, temperature average for each flux in Kelvin (f\_temp\_ave), the total volume of the setup for each measurement (f\_volume\_setup), the model used in flux\_fitting, any column specified in cols\_keep, any column specified in cols\_ave with their value averaged over the measurement after cuts and discarding NA.

References

Zhao, P., Hammerle, A., Zeeman, M., Wohlfahrt, G., 2018. On the calculation of daytime CO2 fluxes measured by automated closed transparent chambers. Agricultural and Forest Meteorology 263, 267–275. <https://doi.org/10.1016/j.agrformet.2018.08.022> exponential is equal to exp\_zhao18, for backwards compatibility

Examples

```
data(co2_df_short)
data(record_short)
stupeflux(
  raw_conc = co2_df_short,
  field_record = record_short,
  datetime_col = datetime,
  start_col = start,
  conc_col = conc,
  startcrop = 10,
  measurement_length = 180,
  fit_type = "exp_zhao18",
  temp_air_col = temp_air,
  conc_unit = "ppm",
  flux_unit = "mmol",
  chamber_volume = 24.5,
  tube_volume = 0.075,
  atm_pressure = 1,
  plot_area = 0.0625
)
```

---

twogases_record	Field record
-----------------	--------------

---

Description

Field record

Usage

```
twogases_record
```

**Format**

A tibble with 12 rows and 1 variable

**start** Start datetime of each flux measurement

**Examples**

`twogases_record`

# Index

## \* datasets

- co2\_conc, [3](#)
- co2\_conc\_mid\_missing, [4](#)
- co2\_conc\_missing, [5](#)
- co2\_df\_missing, [6](#)
- co2\_df\_short, [6](#)
- co2\_fluxes, [7](#)
- co2\_liahovden, [8](#)
- raw\_twogases, [37](#)
- record\_liahovden, [37](#)
- record\_short, [38](#)
- slopes0\_temp, [38](#)
- twogases\_record, [43](#)

co2\_conc, [3](#)

co2\_conc\_mid\_missing, [4](#)

co2\_conc\_missing, [5](#)

co2\_df\_missing, [6](#)

co2\_df\_short, [6](#)

co2\_fluxes, [7](#)

co2\_liahovden, [8](#)

distinct, [9](#)

facet\_wrap\_paginate, [26](#)

flux\_calc, [8](#), [11](#)

flux\_check\_item, [10](#)

flux\_cut, [11](#)

flux\_fit\_type, [18](#)

flux\_fitting, [9](#), [10](#), [11](#), [29](#), [30](#), [32](#), [34](#), [36](#), [42](#), [43](#)

flux\_fitting\_exptz, [13](#)

flux\_fitting\_hm, [14](#)

flux\_fitting\_lm, [15](#)

flux\_fitting\_quadratic, [16](#)

flux\_fitting\_zhao18, [16](#)

flux\_flag\_count, [18](#), [31](#)

flux\_fun\_check, [19](#)

flux\_gep, [20](#)

flux\_match, [21](#)

flux\_match\_col, [22](#)

flux\_match\_fixed, [23](#)

flux\_param\_exp, [23](#)

flux\_param\_kappamax, [24](#)

flux\_param\_lm, [24](#)

flux\_param\_qua, [25](#)

flux\_plot, [25](#)

flux\_plot\_exp, [27](#)

flux\_plot\_flag, [27](#)

flux\_plot\_lin, [28](#)

flux\_plot\_quadratic, [28](#)

flux\_quality, [26](#), [29](#)

flux\_quality\_exp, [32](#)

flux\_quality\_kappamax, [33](#)

flux\_quality\_lm, [34](#)

flux\_quality\_qua, [35](#)

fluxible, [25](#)

ggsave, [26](#)

HMR, [13](#)

raw\_twogases, [37](#)

record\_liahovden, [37](#)

record\_short, [38](#)

scale\_x\_datetime, [26](#)

selectfluxes, [13](#), [31](#)

slopes0\_temp, [38](#)

stupeflux, [39](#)

twogases\_record, [43](#)