

Package: flextrat.hydrus1d (via r-universe)

October 9, 2026

Title R Package for Soil Water Balance and Solute Transport Modelling
Scenarios for Project Flextrat

Version 0.2.0

Description R Package for Soil Water Balance and Solute Transport
Modelling Scenarios for Project Flextrat.

License MIT + file LICENSE

URL <https://github.com/KWB-R/flextrat.hydrus1d>

BugReports <https://github.com/KWB-R/flextrat.hydrus1d/issues>

Depends R (>= 2.10)

Imports dplyr, ggplot2, kwb.hydrus1d, kwb.utils, lubridate, magrittr,
measurements, plotly, pkgbuild, rlang, scales, stringr, tibble,
tidyr, tidyselect

Suggests covr, DT, future.apply, fs, knitr, kwb.dwd, kwb.pkgbuild,
kwb.python, openxlsx, reticulate, rmarkdown, soilDB

VignetteBuilder knitr

Remotes github::kwb-r/kwb.dwd@get-rid-of-rgdal,
github::kwb-r/kwb.hydrus1d, github::kwb-r/kwb.pkgbuild,
github::kwb-r/kwb.python, github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs cmake make libarchive-dev libicu-dev libuv1-dev libssl-dev libx11-dev

Repository <https://kwb-r.r-universe.dev>

Date/Publication 2025-01-10 06:14:52 UTC

RemoteUrl <https://github.com/KWB-R/flextrat.hydrus1d>

RemoteRef HEAD

RemoteSha bc8a160b91c73966875e14439e33aeebab10f4e

Contents

add_times	2
aggregate_atmosphere	3
aggregate_solute	3
aggregate_tlevel	4
combine_dataframes	4
copy_column	5
create_monthly_atm	6
evapo_p	6
get_hydrologic_years	7
get_traveltimes	8
interpolate_time	8
irrigation	9
materials	10
plot_atmosphere	11
plot_monthly_atm	11
plot_solute	12
plot_traveltimes	12
plot_waterbalance	13
precipitation_daily	13
precipitation_hourly	15
prepare_atmosphere	16
prepare_atmosphere_data	17
recalculate_ctop_with_virtualstorage	17
select_hydrologic_years	18
Index	19

add_times	<i>Helper function: add times</i>
-----------	-----------------------------------

Description

Helper function: add times

Usage

```
add_times(df, sim_datetime_start = as.POSIXct("2017-05-01", tz = "UTC"))
```

Arguments

df	df
sim_datetime_start	simulation start as datetime object (default: as.POSIXct("2017-05-01", tz = "UTC"))

Value

data frame with added times

aggregate_atmosphere	<i>Aggregate Atmosphere</i>
----------------------	-----------------------------

Description

for hydrologic years

Usage

```
aggregate_atmosphere(atm_selected, format = "wide")
```

Arguments

atm_selected	atm_selected as retrieved by prepare_atmosphere_data
format	"wide" or "long"

Value

aggregated data

aggregate_solute	<i>Aggregate Solute</i>
------------------	-------------------------

Description

Aggregate Solute

Usage

```
aggregate_solute(  
  solute,  
  sim_datetime_start = as.POSIXct("2017-05-01", tz = "UTC"),  
  col_aggr = "date"  
)
```

Arguments

solute	solute as retrieved by read_solute
sim_datetime_start	simulation start as datetime object (default: as.POSIXct("2017-05-01", tz = "UTC"))
col_aggr	column to be aggregated. One of "date", "yearmonth", "year" or "year_hydrologic" (default: "date")

Value

aggregated solute

aggregate_tlevel	<i>Aggregate t_level</i>
------------------	--------------------------

Description

only for columns starting with "sum" and matching "volume"

Usage

```
aggregate_tlevel(  
  t_level,  
  sim_datetime_start = as.POSIXct("2017-05-01", tz = "UTC"),  
  col_aggr = "date"  
)
```

Arguments

t_level	t_level as retrieved by read_tlevel
sim_datetime_start	simulation start as datetime object (default: as.POSIXct("2017-05-01", tz = "UTC"))
col_aggr	column to be aggregated. One of "date", "yearmonth", "year" or "year_hydrologic" (default: "date")

Value

aggregated t_level data

combine_dataframes	<i>Helper function: combine dataframes</i>
--------------------	--

Description

Helper function: combine dataframes

Usage

```
combine_dataframes(df_list)
```

Arguments

df_list	df_list
---------	---------

Value

df

copy_column

Helper function: copy column

Description

Helper function: copy column

Usage

```
copy_column(df, column, number_of_new_columns)
```

Arguments

df	data fram
column	name of colum to copy
number_of_new_columns	number of columns to be copied

Value

df with copied column names

Examples

```
# Beispiel-Tibble
tb <- tibble::tibble(
  ID = 1:5,
  Wert = c(10, 20, 30, 40, 50)
)

# Anwendung der Funktion
tb_neu <- copy_column(tb, "Wert", 3)
print(tb_neu)
```

create_monthly_atm	<i>Aggregate Atmospheric Data to Monthly Values</i>
--------------------	---

Description

Aggregate Atmospheric Data to Monthly Values

Usage

```
create_monthly_atm(atm = prepare_atmosphere_data())
```

Arguments

atm	atm as retrieved by prepare_atmosphere_data
-----	---

Value

tibble with yearly atmospheric data values

Examples

```
create_monthly_atm()
```

evapo_p	<i>DWD: Potential Evaporation, Daily</i>
---------	--

Description

Median daily potential evaporation for irrigation area (i.e. ~44km²), based on 1x1km² grids of DWD. Downloaded with [read_daily_data_over_shape](#) between 2017-01-01 and 2020-12-31

Usage

```
evapo_p
```

Format

A data.frame with 1461 rows and 10 variables:

file name of downloaded grid file source

date date

year year

month month

day day

mean spatially averaged, mean
sd spatially averaged, standard deviation
min spatially averaged, min
max spatially averaged, max
n_values number of grid-cells used for spatial averaging

Examples

```
## Not run:
### Data download
remotes::install_github("kwb-r/kwb.dwd")
shape_file <- system.file("extdata/input-data/gis/Abwasserverregnungsgebiet.shp",
package = "flectreat.hyrus1d")

# Only data of full months can currently be read!
evapo_p <- kwb.dwd::read_daily_data_over_shape(
  file = shape_file,
  variable = "evapo_p",
  from = "201701",
  to = "202012"
)
## End(Not run)
head(flectreat.hyrus1d::evapo_p)
```

get_hydrologic_years *Get Hydrologic Years*

Description

Get Hydrologic Years

Usage

```
get_hydrologic_years(datetime)
```

Arguments

datetime datetime string

Value

integer vector with hydrologic year

get_traveltimes	<i>Get traveltimes (for conservative tracers)</i>
-----------------	---

Description

Get traveltimes (for conservative tracers)

Usage

```
get_traveltimes(solute_files, dbg = TRUE)
```

Arguments

solute_files	paths to solute files, with good naming convention for monthly solute exposition (e.g. 0110, solute1: first month, solute10: tenth month after simulation start)
dbg	debug messages (default: TRUE)

Value

tibble with time of substance at top/bottom and diff time. Note that the percentile relate to the substance load

interpolate_time	<i>Helper function: interpolate time</i>
------------------	--

Description

Helper function: interpolate time

Usage

```
interpolate_time(
  solute,
  percentiles = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99)
)
```

Arguments

solute	solute result tibble
percentiles	(default: c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99))

Value

tibble with interpolated times for percentiles


```

        .data$month)),
days_in_month = as.numeric(lubridate::days_in_month(.data$date_start)),
date_end = as.Date(sprintf("%d-%02d-%02d",
        .data$year,
        .data$month,
        .data$days_in_month)),
source = kwb.utils::multiSubstitute(.data$source,
        replacements = list("Grundwasser" = "groundwater.mmPerDay",
        "Klarwasser" = "clearwater.mmPerDay")),
irrigation_cbmPerDay = .data$irrigation_m3/.data$days_in_month,
irrigation_area_sqm = irrigation_area_sqm,
irrigation_mmPerDay = 1000*irrigation_cbmPerDay/irrigation_area_sqm) %>%
dplyr::select(.data$year,
        .data$month,
        .data$days_in_month,
        .data$date_start,
        .data$date_end,
        .data$source,
        .data$irrigation_mmPerDay,
        .data$irrigation_area_sqm) %>%
tidyr::pivot_wider(names_from = .data$source,
        values_from = .data$irrigation_mmPerDay)

## End(Not run)
head(flextreat.hydrus1d::irrigation)

```

materials

Materials

Description

A dataset containing hydraulic soil characteristics from Hydrus1D GUI database

Usage

```
materials
```

Format

A data frame with 12 rows and 7 variables:

material_name Name of soil

Qr Residual soil water content

Qs Saturated soil water content

Alpha Parameter a in the soil water retention function (L-1, here: cm)

n Parameter n in the soil water retention function

Ks Saturated hydraulic conductivity (unit: LT-1, here: cm/day)

I Tortuosity parameter in the conductivity function (-)

Source

Hydrus1D GUI -> Soil Hydraulic Properties (Pre-Processing Menu, Water Flow Submenu)

Examples

```
materials
```

plot_atmosphere	<i>Plot Atmosphere</i>
-----------------	------------------------

Description

Plot Atmosphere

Usage

```
plot_atmosphere(atm_selected_hydro_long)
```

Arguments

atm_selected_hydro_long
as retrieved by [aggregate_atmosphere](#) in "long" format

Value

plot atmosphere

plot_monthly_atm	<i>Plot Monthly Atmospheric Data</i>
------------------	--------------------------------------

Description

Plot Monthly Atmospheric Data

Usage

```
plot_monthly_atm()
```

Value

plot of monthly atmospheric data

Examples

```
plot_monthly_atm()
```

plot_solute	<i>Plot Solute</i>
-------------	--------------------

Description

Plot Solute

Usage

```
plot_solute(solute_aggr, y_label = "Share of 'clearwater' (%)")
```

Arguments

solute_aggr	solute_aggr as retrieved by aggregate_solute
y_label	y_label (default: "Share of 'clearwater' (%)")

Value

plot solute

plot_traveltimes	<i>Plot traveltimes</i>
------------------	-------------------------

Description

Plot traveltimes

Usage

```
plot_traveltimes(solute_travel, title = "", ylim = NULL)
```

Arguments

solute_travel	tibble as retrieved by get_traveltimes
title	optional title (default: "")
ylim	optional ylim (default: NULL)

Value

plotly of traveltimes for different percentiles

plot_waterbalance	<i>Plot Water Balance</i>
-------------------	---------------------------

Description

Plot Water Balance

Usage

```
plot_waterbalance(  
  tlevel_aggr,  
  y_label = "Water Balance Component",  
  unit_org = "cm",  
  unit_target = "mm"  
)
```

Arguments

tlevel_aggr	aggregated t_level as retrieved by aggregate_tlevel
y_label	default: "Water Balance Component (mm)"
unit_org	original unit in "t_level" (default: "cm")
unit_target	target unit for plot (default: "mm")

Value

plot water balance

precipitation_daily	<i>Precipitation: Daily</i>
---------------------	-----------------------------

Description

Hourly precipitation data downloaded from DWD for monitoring station Braunschweig (id = 662) between 1997-10-22 and 2021-12-31, which were aggregated to daily values within R

Usage

```
precipitation_daily
```

Format

A data.frame with 8835 rows and 3 variables:

year year

month month

days_in_month days in month

date_start date start

date_end date end

irrigation_area_sqm irrigation area in squaremeter

"groundwater.mmPerDay" irrigation using "groundwater" (mm/sqm)

"clearwater.mmPerDay" irrigation using "clearwater" (mm/sqm))

Examples

```
## Not run:
install.packages(c("dplyr", "rdwd"))
library(dplyr)
rdwd::updateRdwd()
rdwd::findID("Braunschweig")
rdwd::selectDWD(name = "Braunschweig", res = "daily")

url_bs_rain <- rdwd::selectDWD(name = "Braunschweig",
                              res = "hourly",
                              var = "precipitation",
                              per = "historical" )

bs_rain <- rdwd::dataDWD(url_bs_rain)

precipitation_hourly <- rdwd::dataDWD(url_bs_rain) %>%
  dplyr::select(.data$MESS_DATUM, .data$R1) %>%
  dplyr::rename("datetime" = "MESS_DATUM",
               "precipitation_mm" = "R1")

precipitation_daily <- precipitation_hourly %>%
  dplyr::mutate("date" = as.Date(datetime)) %>%
  dplyr::group_by(date) %>%
  dplyr::summarise(rain_mm = sum(precipitation_mm))

## End(Not run)
head(flextreat.hydrus1d::precipitation_daily)
```

precipitation_hourly *Precipitation: Hourly*

Description

Hourly precipitation data downloaded from DWD for monitoring station Braunschweig (id = 662) between 1997-10-22 and 2021-12-31

Usage

precipitation_hourly

Format

A data.frame with 211629 rows and 2 variables:

datetime date time

precipitation_mm precipitation in mm

Examples

```
## Not run:
install.packages(c("dplyr", "rdwd"))
library(dplyr)
rdwd::updateRdwd()
rdwd::findID("Braunschweig")
rdwd::selectDWD(name = "Braunschweig", res = "daily")

url_bs_rain <- rdwd::selectDWD(name = "Braunschweig",
                              res = "hourly",
                              var = "precipitation",
                              per = "historical" )

bs_rain <- rdwd::dataDWD(url_bs_rain)

precipitation_hourly <- rdwd::dataDWD(url_bs_rain) %>%
  dplyr::select(.data$MESS_DATUM, .data$R1) %>%
  dplyr::rename("datetime" = "MESS_DATUM",
               "precipitation_mm" = "R1")

## End(Not run)
head(flextreat.hydrus1d::precipitation_hourly)
```

```
prepare_atmosphere      Prepare Atmosphere
```

Description

Prepares atmospheric input data structure required by HYDRUS1D and by default uses a conservative tracer in irrigation source "clearwater" (set to 1) in order to track the share of cleaned wastewater in the system inflow rate (as "Prec" column is a combined value of irrigation using either "groundwater" or "clearwater" and real "rainfall").

Usage

```
prepare_atmosphere(  
  atm,  
  conc_irrig_clearwater = 100,  
  conc_irrig_groundwater = 0,  
  conc_rain = 0,  
  defaults = kwb.hydrus1d::defaults_atmosphere()  
)
```

Arguments

atm	atm as retrieved by prepare_atmosphere_data
conc_irrig_clearwater	substance concentration in source "clearwater" used for irrigation (default: 100, set all other source concentrations in default to 0 in order to calculate share of "clearwater" infiltration to groundwater)
conc_irrig_groundwater	substance concentration in source "groundwater" used for irrigation (default: 0)
conc_rain	substance concentration in rainfall (default: 0)
defaults	defaults for undefined parameters <code>kwb.hydrus1d::defaults_atmosphere()</code>

Value

tibble with peoered

Examples

```
atm <- prepare_atmosphere_data()  
atm_selected <- select_hydrologic_years(atm)  
prepare_atmosphere(atm_selected)
```

```
prepare_atmosphere_data
```

Prepare Atmospheric Data

Description

Prepare Atmospheric Data

Usage

```
prepare_atmosphere_data()
```

Value

data frame with atmospheric data for Braunschweig

Examples

```
atm <- prepare_atmosphere_data()
atm
```

```
recalculate_ctop_with_virtualstorage
```

Recalculate c_top with virtual storage

Description

Recalculate c_top with virtual storage

Usage

```
recalculate_ctop_with_virtualstorage(atm, tlevel, crit_v_top = -0.05)
```

Arguments

atm	atmosphere time series data
tlevel	tlevel of model output (use <code>kwb.hydrus1d::read_tlevel</code>)
crit_v_top	critical v_top rate (default: - 0.05)

Value

tibble with modified c_top values in order to respect that evaporation is often larger compared to precipitation

```
select_hydrologic_years
```

Select hydrologic years

Description

Select hydrologic years

Usage

```
select_hydrologic_years(atm = prepare_atmosphere_data())
```

Arguments

atm atm as retrieved by prepare_atmosphere_data

Value

select hydrologic years

Examples

```
atm <- prepare_atmosphere_data()
atm_selected <- select_hydrologic_years(atm)
head(atm_selected)
tail(atm_selected)
```

Index

- * **datasets**
 - evapo_p, [6](#)
 - irrigation, [9](#)
 - materials, [10](#)
 - precipitation_daily, [13](#)
 - precipitation_hourly, [15](#)
- add_times, [2](#)
- aggregate_atmosphere, [3](#), [11](#)
- aggregate_solute, [3](#), [12](#)
- aggregate_tlevel, [4](#), [13](#)
- combine_dataframes, [4](#)
- copy_column, [5](#)
- create_monthly_atm, [6](#)
- evapo_p, [6](#)
- get_hydrologic_years, [7](#)
- get_traveltimes, [8](#)
- interpolate_time, [8](#)
- irrigation, [9](#)
- kwb.hydrus1d::defaults_atmosphere(),
[16](#)
- materials, [10](#)
- plot_atmosphere, [11](#)
- plot_monthly_atm, [11](#)
- plot_solute, [12](#)
- plot_traveltimes, [12](#)
- plot_waterbalance, [13](#)
- precipitation_daily, [13](#)
- precipitation_hourly, [15](#)
- prepare_atmosphere, [16](#)
- prepare_atmosphere_data, [3](#), [17](#)
- read_daily_data_over_shape, [6](#)
- read_solute, [3](#)
- read_tlevel, [4](#)
- recalculate_ctop_with_virtualstorage,
[17](#)
- select_hydrologic_years, [18](#)