Package 'TUFLOWR'

April 27, 2023

Type Package

Title Helper Functions for 'TUFLOW FV' Models

Version 0.1.1

Author Matt Gibbs [aut, cre]

Maintainer Matt Gibbs <gibbs.ms@gmail.com>

URL <https://github.com/matt-s-gibbs/TUFLOWR>

Description
Helper functions for 'TUFLOW FV' models. Current functionality includes reading in and plotting output POINTS files and generating initial conditions based on point observations.

License GPL-3

Depends R (>= 4.0.0)

Encoding UTF-8

Imports raster, readr, tibble, tidyr, stringr, sf, sp, dplyr, gstat, lubridate, ggplot2, rlang, magrittr

RoxygenNote 7.2.3

Suggests covr

NeedsCompilation no

Repository CRAN

Date/Publication 2023-04-27 02:42:30 UTC

R topics documented:

TFVGetResults   *Read in one variable out of a TFV POINTS File. Assumes the points*
                                 file ID is a observation station site ID.

Description

Read in one variable out of a TFV POINTS File. Assumes the points file ID is a observation station
site ID.

Usage

```
TFVGetResults(
  Resultfile,
  parameter,
  RunName,
  stations = NULL,
  dailyaverage = FALSE
)
```

Arguments

Resultfile	TUFLOW output POINTS file
parameter	- parameter to plot, must be in column heading, e.g. "SAL"
RunName	- name of model run to add to a column, for ease of combining/plotting
stations	- list of stations to return, to subset from everything recorded. default to NULL, which will return all.
dailyaverage	- set to TRUE to average sub daily data to daily time step

Value

tibble with data in long format, with columns: Time, Site, Value, Data

Examples

```
## Not run:
TFVGetResults("Results_POINTS.csv","SAL","TFV",c("A4261209","A4261165"))

## End(Not run)
```

TFVInitCons   *Generate a TFV initial conditions file from point observations*

Description

Generate a TFV initial conditions file from point observations

Usage

```
TFVInitCons(
  file,
  fgdb,
  initialvaluesfile,
  outputfile,
  cellsize = 50,
  ncell = 50000
)
```

Arguments

`file`	.2dm mesh used to generate the cell IDs and centres
`fgdb`	Shape file (.shp and associated .prj) for zones to be preserved
`initialvaluesfile`	
	csv file with observed data to interpolate
`outputfile`	file to write the initial conditions to
`cellsize`	size of cells to interpolate the observations onto
`ncell`	number of cells to interpolate the observations onto

Details

The shapefile should be a polygon shape file, with different zones to interpolate within. This can be useful to ensure interpolations are physically sensible, for example where there is a barrier between fresh and saline water, or a storage above a lower river. Projections are not handled by this function, so the shapefile should be in the same projection as the mesh.

The initialvaluesfile should have at least 6 columns of information:

- Column 1 as point index
- Column 2 and 3, columns named "X" and "Y", with the coordinates of the point observation, in the same projection as the mesh.
- Columns 4 and 5 are reserved for point information, e.g. a station number and station name. These are not used, but all for context to be documented, and helpful and if the same csv file is also used for specifying locations in a points output block in the model .fvc file.
- Columns 6 and beyond should be values and will be interpolated. Each column will be interpolated for each zone, allowing for multiple initial conditions. The column name should match the model scalar (e.g. H, SAL, TEMP, trace_1)

cellsize and ncell are used to specify the resolution of the interpolation. ncell is in units of the mesh coordinates, e.g. m or degrees

interpolation is undertaken using inverse distance weighting from the gstat package, with the default weighting power of 2.

Value

Nothing is returned to the environment, with the initial conditions generated written to outputfile. Summary metrics are printed to screen to allows for some quick QA

Examples

```
## Not run:
file="001_RM_Wetlands_LL_Coorong_MZ.2dm"
fgdb<-"Zones/Zones.shp"
initialvaluesfile<-"CLLMMInitialValues2019.csv"
outputfile<-pate0(tempdir(),"/Initcons2021.csv")
TFVInitCons(file,fgdb,initialvaluesfile,outputfile)

## End(Not run)
```

TFVPlotagainstObserved

ggplot of TFV model runs and observed data

Description

ggplot of TFV model runs and observed data

Usage

```
TFVPlotagainstObserved(
  Sim,
  Obs,
  ylab,
  file,
  width = 17,
  height = 22,
  order = NULL,
  scales = "fixed",
  cols = NULL,
  newnames = NULL,
  ylim = NA,
  nlegendrow = 1
)
```

Arguments

`Sim`	modelled output, imported using 'TFVGetResults()'
`Obs`	observed data. This is bind_rows() to Sim, so requires the same columns and headings. Obs can be NULL if all data of interest is in Sim
`ylab`	label for y axis
`file`	file to save figure to
`width`	width of image in cm, default to 17
`height`	height of image in cm, default to 22
`order`	character vector of site IDs to order the plots in. Default to NULL, which will plot in alphabetical order
`scales`	control the y axis scales across the facets plots. default to fixed, the same scale across all plots. Change to free_y to have scales dependent on the data for each plot
`cols`	vector of colours to plot each line. Defaults to DEW style 2 greens and 2 blues. Defaults will fail if more than 4 RunNames.
`newnames`	named vector to label the facets on the plot. vector elements should be the new text to use, names should be the names in the data
`ylim`	- specify y axis limits manually as two number vector, e.g. c(0,1)
`nlegendrow`	- specify the number of rows in the legend, increase this if the scenario names go off the figure. Defaults to 1

Value

Nothing is returned to the R environment. The generated figure is saved to file.

Examples

```
## Not run:
stations<-c("A4261043", "A4261134","A4261135","A4260572","A4260633","A4261209","A4261165")
TFVPlotagainstObserved(Sim,Obs,"Salinity (g/L)","salinity.png",order=stations)

## End(Not run)
```

TUFLOWR	*TUFLOWR*

Description

Helper functions for 'TUFLOW FV' models. Current functionality includes reading in and plotting output POINTS files and generating initial conditions based on point observations.

Index