Package: dwc.ar4gw (via r-universe)

September 11, 2024

Title R Package for Preparing Modflow Output Data for Articifical Reality Visualisation

Version 0.0.0.9000

Description R Package for preparing Modflow output data for articifical reality visualisation.

License MIT + file LICENSE

URL https://github.com/KWB-R/dwc.ar4gw

BugReports https://github.com/KWB-R/dwc.ar4gw/issues

Imports dplyr, ggplot2, magrittr, reshape2, reticulate, rlang, tidyr

Suggests covr, knitr, kwb.nextcloud, kwb.python, kwb.utils, remotes, rmarkdown, stringr

VignetteBuilder knitr

Remotes github::kwb-r/kwb.nextcloud, github::kwb-r/kwb.python, github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

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

RemoteUrl https://github.com/KWB-R/dwc.ar4gw

RemoteRef HEAD

RemoteSha 8c06d2e7a9d4c8308c84d9a60f2e46210f9422d8

Contents

get_extended_budget *get_extended_budget*

Description

Get the flow rate across cell faces including potential stresses applied along boundaries at a given time. Only implemented for "classical" MODFLOW versions where the budget is recorded as FLOW RIGHT FACE, FLOW FRONT FACE and FLOW LOWER FACE arrays.

Usage

```
get_extended_budget(cbcfile, ...)
```

Arguments

`cbcfile`	Cell by cell file produced by Modflow.
`...`	additional arguments passed to flopy$utils$postprocessing$get_extended_budget for information see references below

Value

return sub-list for with "Qx_ext", "Qy_ext", "Qz_ext" for each budget output timestep. Flow rates across cell faces. Qx_ext is a array of size (nlay, nrow, ncol + 1). Qy_ext is a array of size (nlay, nrow + 1, ncol). The sign is such that the y axis is considered to increase in the north direction. Qz_ext is a ndarray of size (nlay + 1, nrow, ncol). The sign is such that the z axis is considered to increase in the upward direction.

References

https://flopy.readthedocs.io/en/latest/source/flopy.utils.postprocessing.html#flopy.utils.postprocessing.get_extended_budget

Examples

```
 ## Not run:
 flopy <- import_flopy()
 reticulate::py_help(object = flopy$utils$postprocessing$get_extended_budget)

## End(Not run)
```

`import_flopy` *Import "flopy" Python Package*

Description

Import "flopy" Python Package

Usage

```
import_flopy(convert = TRUE, ...)
```

Arguments

`convert` Boolean (default: TRUE); should Python objects be automatically converted to their R equivalent? If set to FALSE, you can still manually convert Python objects to R via the py_to_r function.

`...` additional arguments passed to import

Value

imports "flopy" python package

`plot_data` *plot_data*

Description

plot_data

Usage

```
plot_data(
  multiarray,
  title = "",
  value_min = NULL,
  value_max = NULL,
  fill_gradient_low = "grey90",
  fill_gradient_high = "red"
)
```

Arguments

`multiarray` multidimensional MODFLOW array

`title` title for plot

`value_min` minimum value. All smaller values will be excluded from dataset (default: NULL)

`value_max` maximum value. All larger values will be excluded from dataset (default: NULL)

`fill_gradient_low`
fill_gradient_low (default: "grey90")

`fill_gradient_high`
fill_gradient_high (default: "red")

Value

plot all Modflow layers

`to_long` *Helper function: convert multiarray to long format*

Description

Helper function: convert multiarray to long format

Usage

```
to_long(multiarray)
```

Arguments

`multiarray` multidimensional MODFLOW array

Value

convert multiarray to long format with columns: layer, column, row, value `to_wide` *to_wide*

Description

to_wide

Usage

```
to_wide(multiarray_long, parameter = "")
```

Arguments

`multiarray_long`
multiarray in long format (as retrieved by `to_long`)

`parameter` prefix to use for parameter in wide format (default: "")

Value

tibble with column, row, parameter_layer1-n

Index