

# Package ‘ambhasGW’

August 2, 2017

**Title** Ground Water Modelling

**Version** 0.0.2

**Description** Implements distributed transient groundwater modelling. The model is based on the groundwater flow equation solved numerically using the finite difference explicit scheme.

**Depends** R (>= 3.2.3)

**Imports** yaml, raster, stats, rgdal

**License** GPL (>= 3)

**Repository** CRAN

**Encoding** UTF-8

**RoxygenNote** 6.0.1.9000

**NeedsCompilation** no

**Author** Sat Tomer [aut, cre],  
Sekhar Muddu [aut],  
Vishal K. Mehta [aut],  
Subash Yegina [aut],  
Thiyaku S. [aut]

**Maintainer** Sat Tomer <sat.kumar@aapahinnovations.com>

**Date/Publication** 2017-08-02 13:53:06 UTC

## R topics documented:

ambhasGW . . . . .	2
create_inputs . . . . .	2
parameter_definition . . . . .	3
update_horizontally . . . . .	3
update_vertically . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

ambhasGW

*Ground Water Modelling*

---

### **Description**

ambhasGW Computes the groundwater head for each time step in raster format

### **Usage**

```
ambhasGW(input.file)
```

### **Arguments**

input.file      yml file having input parameters information

### **Examples**

```
# Create necessary input file to make dummy run
# Dummy directory to run
dummy.dir <- tempdir()

# Make dummy run
create_inputs(dummy.dir)
input.file <- file.path(dummy.dir , 'input/parameters.yml')
ambhasGW(input.file)
```

---

create\_inputs

*Create dummy input data*

---

### **Description**

Creates dummy data and corresponding yml file to run the model.

### **Usage**

```
create_inputs(dummy.dir)
```

### **Arguments**

dummy.dir      Dummy directory to input and output data

### Examples

```
# Creates input data and yml file
create_inputs(tempdir())
```

---

parameter\_definition    *Parameters definition*

---

### Description

Prints the parameters definition of given parameter

### Usage

```
parameter_definition(par)
```

### Arguments

par                    Parameter name

### Examples

```
parameter_definition('hini')
```

---

update\_horizontally    *Horizontal distribution of the flow in grid.*

---

### Description

Simulates the horizontal distribution of the flow in grids based on the diffusivity.

### Usage

```
update_horizontally(h, beta)
```

### Arguments

h                      Initial head  
beta                    a parameter based on the diffusivity, time step and grid size.

**Value**

Returns the updated head.

**Examples**

```
# create synthetic head:
h <- cbind(c(1,2,3),c(4,5,6),c(7,8,9))
h
beta <- 0.5
h1 <- update_horizontally(h, beta)
h1

# increase the value of beta:
beta <- 1.0
h1 <- update_horizontally(h, beta)

# decrease the value of beta:
beta <- 0
h1 <- update_horizontally(h, beta)
h1
```

---

update_vertically	<i>Update the vertical flux components</i>
-------------------	--

---

**Description**

Update the grids with net recharge and discharge, also computes the baseflow from each grid.

**Usage**

```
update_vertically(h, sy, net.recharge, p.baseflow, hmin.baseflow)
```

**Arguments**

h	Initial head raster
sy	Specific yield [0-1].
net.recharge	Net recharge (recharge - draft)
p.baseflow	parameter for the baseflow [0-1].
hmin.baseflow	head corresponding to zero baseflow.

**Value**

Returns a list containing the updated head and computed baseflow.

**Examples**

```
# create synthetic head and parameters
h <- 10
sy <- 0.01
net.recharge <- 1
p.baseflow <- 0.7
hmin.baseflow <- 5

# update the head and compute baseflow
out <- update_vertically(h, sy, net.recharge, p.baseflow, hmin.baseflow)
out
```

# Index

`ambhasGW`, [2](#)

`create_inputs`, [2](#)

`parameter_definition`, [3](#)

`update_horizontally`, [3](#)

`update_vertically`, [4](#)