Description of PSD code

V2 2020-08-06

Classes

There are four classes within the code: GWNode, GWGrid, DEM and River. These classes provide the basic functionality so that a groundwater head to be calculated at any point on a user defined grid. The classes and a brief description are presented in the table below:

Class	Description
GWNode	Groundwater node – contains the calculation method for change in groundwater head at a particular location. The method used is based on an Explicit solution of the 2D groundwater flow equation. The change in groundwater head resulting from a instantaneous change in stage is provided by the function "getGWHeadChange"
GWGrid	Groundwater grid – acts as a container for the groundwater nodes. Input data is read within the constructor. These data include the groundwater parameters (k, S) as well as recharge and the river node associated with each groundwater node. A function "getGWHeads" is provided to output the groundwater heads at any time to the file "GridTest.out"
River	Holds the location of a series of river nodes. The X and Y position is provided to other objects by two functions "getXpos" and "getYpos"
DEM	Contains the DEM at the same grid size as for the GWGrid. The DEM value is given via the function "getDEMValue".

Flowchart

Figure 1 illustrates the main flow through the PSD code, contained in PSDModel.cpp. First the text files used for input and output are created. Then the GWGrid object is instantiated which results in the data in "`GridTest.in`" being read and a grid of GWNode objects being created. After this the River object is instantiated which reads in the river node locations from "`GridTestRiver.in`". After this the DEM object is instantiated which reads in the river node locations from "`GridTestDEM.in`". The model then enters the time step loop which means the groundwater head at each grid location is calculated for a given time and written to "`GridTestHeads.out`", "`GridTestFLows.out`", and "`GridTestHYdro.out`", . Finally the input and output files are closed.

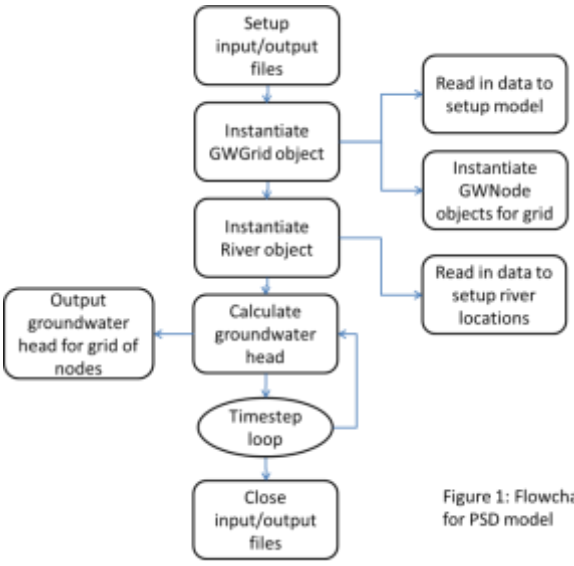


Figure 1: Flowchart for PSD model

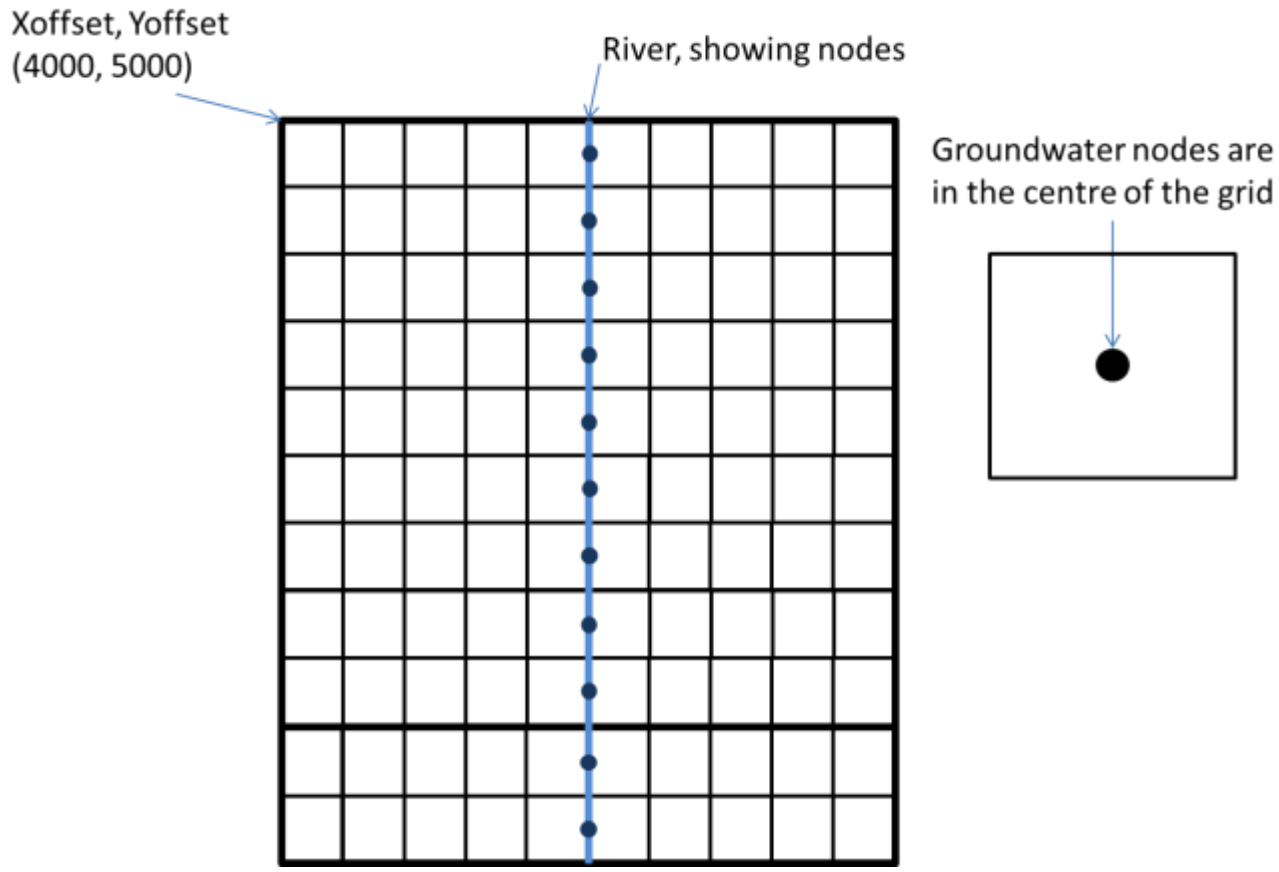
Methods in the main part of the code

The following table outlines the methods used to control the running of the model as well as enabling the state variable (groundwater heads) and input flux (recharge) to be interrogated and to be set to a value.

Method	**Description**
`initialise_model();`	Initialises model: opens all channels to read input files and write output files; instantiates objects for the River, DEM and Grids classes
`run_model();`	Runs the model one time step at a time (via `do_timestep`)
`finalise_model();`	Finalises model: closes all open files
`do_timestep()`	Updates the groundwater heads for the grid nodes
`getGWHead(int valuei, int valuej)`	Enables the groundwater head value to be interrogated at a node (i, j)
`getRecharge(int valuei, int valuej)`	Enables the recharge value to be interrogated at a node (i, j)
`setGWHead(double valueGWHead, int valuei, int valuej)`	Enables the groundwater head value to be set to a value at a node (i, j)
`setRecharge(double valueRecharge, int valuei, int valuej)`	Enables the recharge value to be set to a value at a node (i, j)

Input files

A simple example has been setup using a rectangular grid and a straight-line river. The parameterisation of this is used to provide a simple example to illustrate the input files.



The model time control is determined by `GridTestMain.in` which sets the initial time, time step and end time:

`GridTestMain.in`	Read by the main class

`0.05`
`1`
`2`
	Time step (days)
	start time (days
	end time (days)

Three input files are required one for the setting up the grid of groundwater nodes, `GridTest.in` one, `GridTestRiver.in`, for setting the position of the river nodes and another one `GridTestDEM.in`, for setting up the DEM.

`GridTest.in`	Read by the constructor of the GWGrid class

`5.`
`0.005`
`10.0`
`0.001`
`10.5`
`10`
`0 0`
`1 1.`
`2 0.`
`3 0.`
`4 0.`
`5 0.`
`6 0.`
`7 0.`
`8 0.`
`9 0.`
`10 0.`
`10 11`
`50 50`
`4000 5000`
`0 0 0 0 0 1 0 0 0 0 0`
`0 0 0 0 0 2 0 0 0 0 0`
`0 0 0 0 0 3 0 0 0 0 0`
`0 0 0 0 0 4 0 0 0 0 0`
`0 0 0 0 0 5 0 0 0 0 0`
`0 0 0 0 0 6 0 0 0 0 0`
`0 0 0 0 0 7 0 0 0 0 0`
`0 0 0 0 0 8 0 0 0 0 0`
`0 0 0 0 0 9 0 0 0 0 0`
`0 0 0 0 0 10 0 0 0 0 0`
`0 0 0 0 0 11 0 0 0 0 0`
`0 0 0 0 0 12 0 0 0 0 0`
`0 0 0 0 0 13 0 0 0 0 0`
`0 0 0 0 0 14 0 0 0 0 0`
`0 0 0 0 0 15 0 0 0 0 0`
	Hydraulic conductivity (m/d)
	Specific storage (/m)
	Saturated thickness (m)
	Recharge rate (m/d)
	Initial head (m)
	River stage rise data
	Number of days to discretise the surface water hydrograph
	Change in stage (m) for each day of the hydrograph
	
	No. of nodes X direction, no. of nodes Y direction
	DeltaX, DeltaY
	XOffset, YOffset (X, Y of Top Right Hand Corner)
	Grid of river nodes for each groundwater head node, note this code corresponds to the river node specified in “GridTestRiver.in”

`GridTestRiver.in`	Read by the constructor of the River class
`11`	Number of river locations

1 4250 5050 2 4250 5100 3 4250 5150 4 4250 5200 5 4250 5250 6 4250 5300 7 4250 5350 8 4250 5400 9 4250 5450 10 4250 5500 11 4250 5550	River location ID, X, Y Note: These do not have to be in numerical order, i.e. as long as there is the correct number of River IDs and associated X and Y values

GridTestDEM.in	Read by the constructor of the Dem class
1 11 15 50 50 4000 5000 50.	Switch for surface-groundwater interaction (1 on; 0 off) No. of nodes X direction, no. of nodes Y direction DeltaX, DeltaY XOffset, YOffset (X, Y of Top Right Hand Corner) Grid of river nodes for each groundwater head node, note this code corresponds to the river node specified in "GridTestRiver.in" Grid of surface elevations

Finally `GridTestHydro.in` is used to specify the hydrographs output by the model at each time step.

GridTestHydro.in	Read by the main class
2 1 1 10 10	Number of hydrographs (-) I, j of first hydrograph 1, j of second hydrograph

Output files

The code produces three output files:

1. `GridTestHeads.out`, which is the groundwater head calculated for a time t at all the nodes and written as a grid
2. `GridTestFlows.out`, which is the flows calculated for a time t at all the nodes and written as a grid
3. `GridTestHydro.out`, which is the groundwater head calculated at a point a time t at all the selected nodes in `GridTestHydro.in`

`GridTestHeads.out`	

`[ input is echoed first ]`
`Time is: 0.1`
`10.671 10.7543 10.8406 …`
`10.671 10.7543 10.8406 …`
`10.671 10.7543 10.8406 …`
`10.671 10.7543 10.8406 …`
…
…
…
…
…
…
…
	Time as a header
	Head (m) for each groundwater node

`GridTestFlows.out`	

`Time is: 0.1`
`0.61 0.75 0.84 …`
`0.61 0.75 0.84 …`
`0.61 0.75 0.84 …`
…
…
…
…
…
…
…
	Time as a header
	Flows (m3/m2/d) for each groundwater node

`GridTestHydro.out`	

`0.5,10.51,10.51,`
`1,10.52,10.9,`
`1.5,10.53,11.3656,`
`2,10.54,11.8155,`
`2.5,10.55,12.2255,`
`3,10.56,12.5924,`
`3.5,10.5702,12.9198,`
`4,10.5813,13.2127,`
`4.5,10.5938,13.4758,…`
…
	Time, head1, head2,

... ...	