MODFLOW with Flopy (4)

advanced topics

Harry Lee

3/19/2018

CEE 696

Announcement

1. Update Flopy to version 3.2.9 if you haven't
2. The next class will continue in two weeks (4/9) email me if you have any question on
3. Sign up for HI-DSI Python Workshop (4/17 - 18)

What we haven’t covered

- Transient Simulation
- Boundary conditions
- Recharge package
- Load existing model inputs
- Unsaturated flow (Later)

Transient GW flow simulation

```python
# Time step parameters
nper = 3
perlen = [1, 100, 100]
nstp = [1, 100, 100]
steady = [True, False, False]
dis = flopy.modflow.ModflowDis(mf, nlay, nrow, ncol,
                               delr=delr, delc=delc,
                               top=ztop, botm=botm[1:],
                               nper=nper, perlen=perlen,
```

Boundary conditions

MODFLOW has packages for five boundary conditions

- Well
- General-Head Boundary
- Drain
- River
- Time-Variant Constant Head

General-Head Boundary (GHB) Package

```
flopy.modflow.mfghb.ModflowGhb(model,
                               stress_period_data)
```

stress_period_data list/numpy.recarray/dict of boundaries

GHB flux is defined as

$$Q = -C(h - h_0)$$

- C = conductance
- h = head at a cell determined by initial condition (at t = 0.) or simulation
- h_0 = constant head source (stage)

General-Head Boundary (GHB) Package

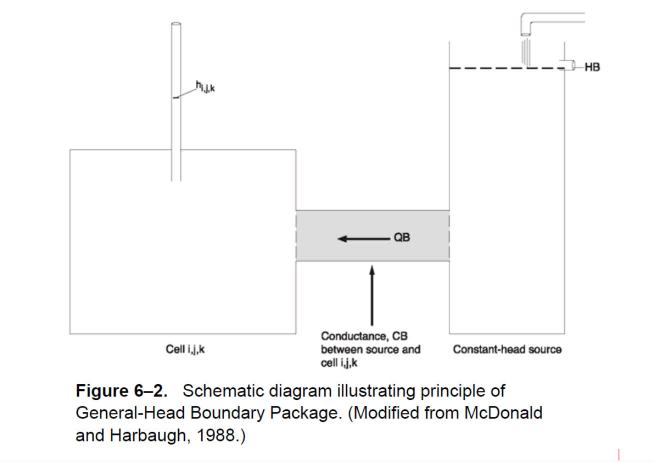


Figure 6–2. Schematic diagram illustrating principle of General-Head Boundary Package. (Modified from McDonald and Harbaugh, 1988.)

General-Head Boundary (GHB) Package

```
flopy.modflow.mfdrn.ModflowGhb(mf, stress_preiod_data = spd)
spd =
{0: [
    [lay, row, col, stage, cond],
    ...,
    [lay, row, col, stage, cond]
    ],
 1: [...], ...,
 kper:
    [
    [lay, row, col, stage, cond],
    ...,
    [lay, row, col, stage, cond]
    ]
}
```

You can use list, numpy.recarray and dict to assign your boundary conditions. See below:

`https://github.com/modflowpy/flopy/blob/develop/examples/Notebooks/flopy3_modflow_boundaries.ipynb`

Comments on head-dependent boundary conditions

- if no values are specified for a certain stress period, then the list of boundaries for the previous stress period for which values were defined is used
- If the number of lists is smaller than the number of stress periods, the last list will apply until the end of the simulation

Drain (Drn) Package

```
flopy.modflow.mfdrn.ModflowDrn(model,
                               stress_period_data)
```

stress_period_data list/numpy.recarray/dict of boundaries

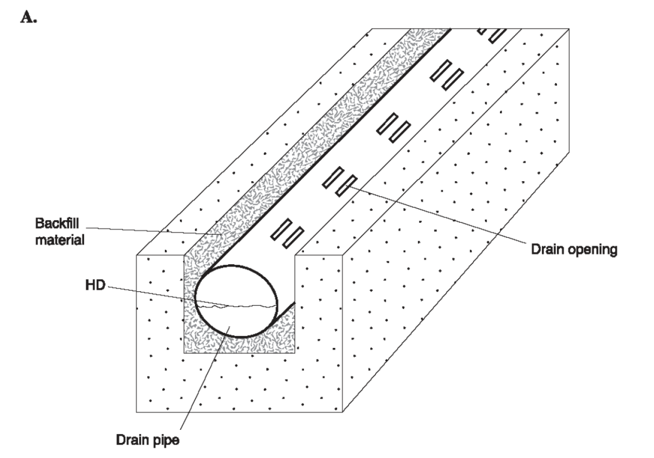
Drainage flux is defined as

$$Q = -C(h - d)$$

- C = conductance
- h = head at a cell determined by initial condition (at t = 0.) or simulation
- *d* = drainage elevation (stage)

Note that water is only allowed to flow out of the groundwater system, i.e., for h < d, Q = 0

Drain (DRN) Package



River (RIV) package

```
flopy.modflow.mfriv.ModflowRiv(model,
                               stress_period_data=spd)
```

stress_period_data list/numpy.recarray/dict of boundaries (lay, row, col, stage, cond, rbot)

- rbot : the elevation of the bottom of the riverbed

River (RIV) Package

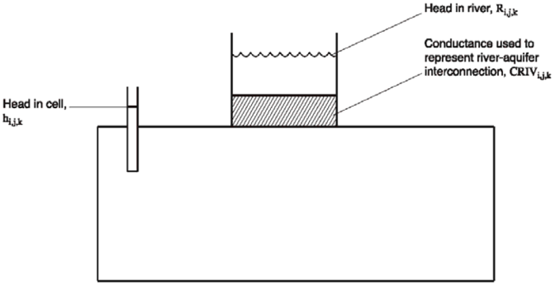

Head in river, $R_{i,j,k}$
Conductance used to represent river-aquifer interconnection, $CRIV_{i,j,k}$
Head in cell, $h_{i,j,k}$

Figure 2–4. Conceptual representation of leakage through a riverbed into a cell. (Modified from McDonald and Harbaugh, 1988.)

River (RIV) package - stress_period_data

```
spd =
{0: [
    [lay, row, col, stage, cond, rbot],
    [lay, row, col, stage, cond, rbot]
    ],
1:  [
    [lay, row, col, stage, cond, rbot],
    [lay, row, col, stage, cond, rbot]
    ], ...
kper:
    [
    [lay, row, col, stage, cond, rbot],
    [lay, row, col, stage, cond, rbot]
    ]
}
```

Exercise (RIV) package

```
#layer, row, column, stage, conductance, river bottom
spd = [
                      [2, 3, 4, 10.7, 5000., -5.7],
mf = flopy.modflow.Modflow(modelname='test')
riv = flopy.modflow.ModflowRiv(mf,
                               stress_period_data=spd)
mf.write_input()

# you print out the file content from ipython console
!more test.riv
```

Constant Head Boundary (CHD) package

```
flopy.modflow.mfchd.ModflowChd(model,
                               stress_period_data=spd)
```

stress_period_data list/numpy.recarray/dict of boundaries (lay, row, col, shead, ehead)

- shead: the head at the start of the stress period
- ehead: the head at the end of the stress period

Constant Head Boundary (CHD) package - stress_period_data

```
spd =
{0: [
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead]
    ],
1:  [
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead]
    ], ...
kper:
    [
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead],
    [lay, row, col, shead, ehead]
    ]
}
```

Recharge (RCH) package

```
flopy.modflow.mfrch.ModflowRch(model, nrchop=3,
ipakcb=None, rech=0.001,
irch=0, extension='rch',
unitnumber=None, filenames=None)
```

nrchop 1: Recharge to top grid layer only 2: Recharge to layer defined in irch 3: Recharge to highest active cell (default is 3).

rech the layer to which recharge is applied in each vertical column (only used when nrchop=2). (default is 0), the recharge flux array(nrow,ncol)

irch array(norw, ncol)

rech recharge flux (default 1e-3)

Load from existing model inputs

```
ml = flopy.modflow.Modflow.load(f, exe_name='mf2005.exe'
                                 verbose = False,
                                 model_ws='.',
                                 check =True)
```

- f : MODFLOW name file
- exe_name : if you want to run again, input a relevant executable
- model_ws : model workspace path; for current directory, use ”.”
- check : check model inputs

https://github.com/modflowpy/flopy/blob/develop/examples/Notebooks/flopy3_modflow_boundaries.ipynb

Example : Freyberg [1988]

Download MODFLOW inputs from `https://www.dropbox.com/s/h7qtas50d7xtivv/data.zip?dl=0` and unzip the files into your working directory

```python
import flopy
import os
model_ws = "freyberg"
ml = flopy.modflow.Modflow.load("freyberg.nam", model_ws
```

Load existing model inputs

Plotting results

```
# plot K
mf.lpf.hk.plot(mflay=0,colorbar=True)
plt.show()

# plot h
h = flopy.utils.HeadFile(os.path.join(model_ws,
                        "freyberg.hds"),model=mf)
h.plot(totim=h.times[-1], contour=True, grid=True,
       colorbar=True, figsize=(10,10), vmin=0,vmax =30
plt.show()
```

shapefile export

install package “pyshp”

```
mf.dis.export("freyberg/freyberg_dis.shp")
h = flopy.utils.HeadFile(os.path.join(model_ws,
                        "freyberg.hds"),model=mf)

#let's write these heads to shapefile
h.to_shapefile(os.path.join(mf.model_ws,
                            "freyburg_head.shp"))
```

Exercise 2 - transient modeling

Download input files from `https://www.dropbox.com/s/jwhsw0pcyp0i63i/freyberg_transient.zip?dl=0` and copy them to your working directory.

Run example below

`https://github.com/modflowpy/flopy/blob/develop/examples/Notebooks/flopy3_working_stack_demo.ipynb`

References for your project

- Flopy notebook examples `https://github.com/modflowpy/flopy/blob/develop/docs/notebook_examples.md`
- Prof. Anderson's lecture slides `http://www.geology.wisc.edu/courses/g724/`
- Prof. Poeter's lecture notes `http://inside.mines.edu/~epoeter/583/lesson_list.html`
- Flopy code documentation is always your friend: `https://modflowpy.github.io/flopydoc/code.html`