MODFLOW with Flopy (2)

simulation with flopy

Harry Lee

1/22/2018

CEE 696 & Stanford CEE 268

GW rap

https://youtu.be/cuJP4kdi6Og

By Prof. Roseanna Neupauer at CU-Boulder

Flopy Example 1

Example 1

A horizontal confined aquifer (1000 x 1000 x 50 m) with constant head on the western and eastern boundaries (h_{west} = 10 m, h_{east} = 0 m), no flow condition on northern and southern boundaries. Horizontal and vertical hydraulic conductivity are given by 10 m/d.

Please run the tutorial code below: `https://github.com/modflowpy/flopy/blob/master/examples/Tutorials/Tutorial01/tutorial01.py`

my_first_flopy_scipt.py (1) - Create MODFLOW Model Object

```python
import flopy
import numpy as np

# Assign name and create modflow model object
modelname = 'ex1'
mf = flopy.modflow.Modflow(modelname,exe_name='mf2005')
```

modelname name of model. This will be used to name the MODFLOW input files (default: ‘modflowtest’)

exe_name the name of the executable to use (the default is ‘mf2005’)

my_first_flopy_scipt.py (2) - Discretization File (DIS)

```
# Model domain and grid definition
Lx, Ly = 1000., 1000. # aquifer length in x/ y
ztop, zbot = 0., -50. # aquifer top/bottom elev.
nlay, nrow, ncol = 1, 10, 10 # # of layers/rows/cols
delr, delc = Lx/ncol, Ly/nrow # spacing along rows/cols
delv = (ztop - zbot) / nlay # sapacing along depth
botm = np.linspace(ztop, zbot, nlay + 1) # bot elev.

# Create the discretization object
dis = flopy.modflow.ModflowDis(mf, nlay, nrow,
ncol, delr=delr, delc=delc,top=ztop, botm=botm[1:])
```

- More input parameters, see below
- `https://modflowpy.github.io/flopydoc/mfdis.html`

my_first_flopy_scipt.py (3) - Basic Package (BAS)

```
# Variables for the BAS package
ibound = np.ones((nlay, nrow, ncol), dtype=np.int32)
ibound[:, :, 0] = -1 # constant head at the first col
ibound[:, :, -1] = -1 # constant head at the last col
strt = np.ones((nlay, nrow, ncol), dtype=np.float32)
strt[:, :, 0] = 10.
strt[:, :, -1] = 0.
bas = flopy.modflow.ModflowBas(mf, ibound=ibound,
strt=strt)
```

IBOUND boundary variable. Constant head for IBOUND < 0, inactive for IBOUND = 0, active (determined by simulation) for IBOUND > 0.

STRT initial (starting) head

https://modflowpy.github.io/flopydoc/mfbas.html

my_first_flopy_scipt.py (4) - Layer Property Flow (LPF) package

```
# Add LPF package to the MODFLOW model
lpf = flopy.modflow.ModflowLpf(mf, hk=10., vka=10.,
ipakcb=53)
```

HK hydraulic conductivity, can be an array

VKA vertical hydraulic conductivity, can be an array

ipakcb Fortran Unit Number for cell-by-cell budget data . See below for more information

```
https://modflowpy.github.io/flopydoc/mflpf.html
https://water.usgs.gov/ogw/modflow/
MODFLOW-2005-Guide/index.html?lpf.htm
```

my_first_flopy_scipt.py (5) - Output Control (OC)

```
spd = {(0, 0): ['print head', 'print budget',
'save head', 'save budget']}
oc = flopy.modflow.ModflowOc(mf, stress_period_data=spd,
compact=True)
```

stress_period_data python dictionary for output control: which output is saved for the corresponding stress period and time step with zero-based indexing

compact boolean indicating whether results will be saved in compact budget form. (default is True).

stress_period_data with (i,j) indicates the j+1-th time step in the i+1-th stress period. If None, heads are saved for the last time step of each stress period.

For example, stress_period_data = {(0,1):['save head']} would save the head for the second timestep in the first stress period.
`https://modflowpy.github.io/flopydoc/mfoc.html`

my_first_flopy_scipt.py (6) - Preconditioned Conjugate-Gradient (PCG) Package

```
# Add PCG package to the MODFLOW model
pcg = flopy.modflow.ModflowPcg(mf)
```

Solver for $Ax = b$. We will get back to this later when we learn numpy and optimization techniques

my_first_flopy_scipt.py (7) - Generate MODFLOW Inputs

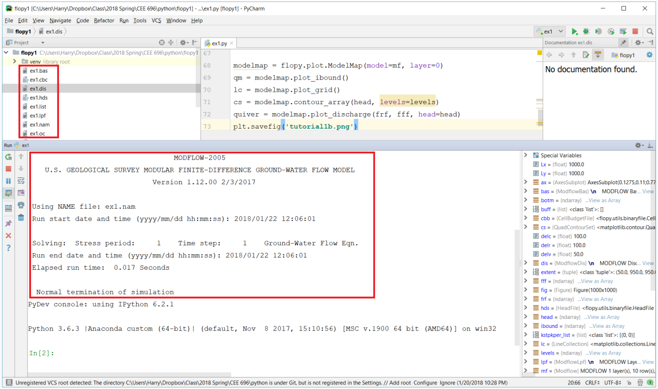
```
# Write the MODFLOW model input files
mf.write_input()
```

What do you see in your project directory?

`https://modflowpy.github.io/flopydoc/mf.html`

my_first_flopy_scipt.py (8) - Run MODFLOW

```python
# Run the MODFLOW model
success, buff = mf.run_model() # success should be True
```



flopy1 [C:\Users\Harry\Dropbox\Class\2018 Spring\CEE 696\python\flopy1] - ...\ex1.py [flopy1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
flopy1 〉 ex1.dis 〉
Project
flopy1 C:\Users\Harry\Dropbox\Class\2018 Spring\CEE 696\python\flopy1
venv library root
ex1.bas
ex1.cbc
ex1.dis
ex1.hds
ex1.list
ex1.lpf
ex1.nam
ex1.oc
ex1.py
ex1
Documentation ex1.dis
flopy1
Run ex1

```python

modelmap = flopy.plot.ModelMap(model=mf, layer=0)
qm = modelmap.plot_ibound()
lc = modelmap.plot_grid()
cs = modelmap.contour_array(head, levels=levels)
quiver = modelmap.plot_discharge(frf, fff, head=head)
plt.savefig('tutorial1b.png')
```

No documentation found.

```
                                  MODFLOW-2005
    U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER FLOW MODEL
                             Version 1.12.00 2/3/2017

Using NAME file: ex1.nam
Run start date and time (yyyy/mm/dd hh:mm:ss): 2018/01/22 12:06:01

Solving:  Stress period:     1    Time step:     1    Ground-Water Flow Eqn.
Run end date and time (yyyy/mm/dd hh:mm:ss): 2018/01/22 12:06:01
Elapsed run time:  0.017 Seconds

 Normal termination of simulation
PyDev console: using IPython 6.2.1

Python 3.6.3 |Anaconda custom (64-bit)| (default, Nov  8 2017, 15:10:56) [MSC v.1900 64 bit (AMD64)] on win32

In[2]:
```

```
Special Variables
Lx = {float} 1000.0
Ly = {float} 1000.0
ax = {AxesSubplot} AxesSubplot(0.1275,0.11;0.77
bas = {ModflowBas} \n    MODFLOW Ba:... View
botm = {ndarray} ...View as Array
buff = {list} <class 'list'>: []
cbb = {CellBudgetFile} <flopy.utils.binaryfile.Cell
cs = {QuadContourSet} <matplotlib.contour.Qua
delc = {float} 100.0
delr = {float} 100.0
delv = {float} 50.0
dis = {ModflowDis} \n    MODFLOW Dis... View
extent = {tuple} <class 'tuple'>: (50.0, 950.0, 950.
fff = {ndarray} ...View as Array
fig = {Figure} Figure(1000x1000)
frf = {ndarray} ...View as Array
hds = {HeadFile} <flopy.utils.binaryfile.HeadFile
head = {ndarray} ...View as Array
ibound = {ndarray} ...View as Array
kstpkper_list = {list} <class 'list'>: [(0, 0)]
lc = {LineCollection} <matplotlib.collections.Line
levels = {ndarray} ...View as Array
lpf = {ModflowLpf} \n    MODFLOW Laye... View
mf = {Modflow} MODFLOW 1 layer(s), 10 row(s),
```

Unregistered VCS root detected: The directory C:\Users\Harry\Dropbox\Class\2018 Spring\CEE 696\python is under Git, but is not registered in the Settings. // Add root Configure Ignore (1/20/2018 10:28 PM)   20:66 CRLF÷ UTF-8÷

https://modflowpy.github.io/flopydoc/mf.html

my_first_flopy_scipt.py (9) - Post-processing

```
import matplotlib.pyplot as plt
import flopy.utils.binaryfile as bf

fig = plt.figure(figsize=(10,10)) # w,h tuple in inches
ax = fig.add_subplot(1, 1, 1, aspect='equal')

hds = bf.HeadFile(modelname+'.hds') # heads saved in hds
times = hds.get_times() # plz print times
# get data at a specific simulation time (totim)
head = hds.get_data(totim=times[-1])

levels = np.linspace(0, 10, 11) # contour level for plot
```

Since we didn’t specify the total simulation time for steady-state simulation, default 1.0 is used. You can print “times”.

my_first_flopy_scipt.py (10) - Post-processing

```
# budget saved in modelname.cbc
cbb = bf.CellBudgetFile(modelname+'.cbc')

#kstpkper_list = cbb.get_kstpkper()

frf = cbb.get_data(text='FLOW RIGHT FACE',
totim=times[-1])[0] # flow in cols
fff = cbb.get_data(text='FLOW FRONT FACE',
totim=times[-1])[0] # flow in rows
```

get_kstpkper() Get a list of unique stress periods (kper) and time steps (kstp) in the file

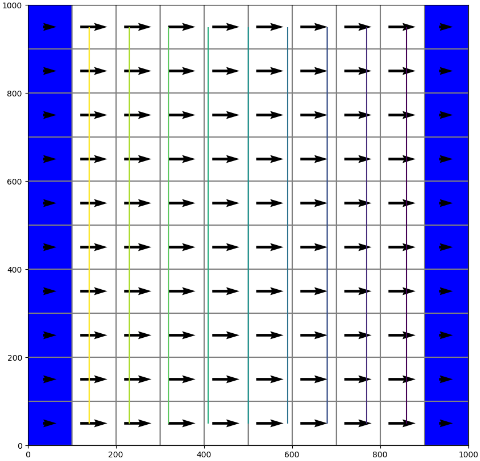
my_first_flopy_scipt.py (11) - Post-processing

```
modelmap = flopy.plot.ModelMap(model=mf, layer=0)
qm = modelmap.plot_ibound() # plot ibound
lc = modelmap.plot_grid() # plot grid
cs = modelmap.contour_array(head, levels=levels) # head
quiver = modelmap.plot_discharge(frf, fff, head=head)
plt.show()
```

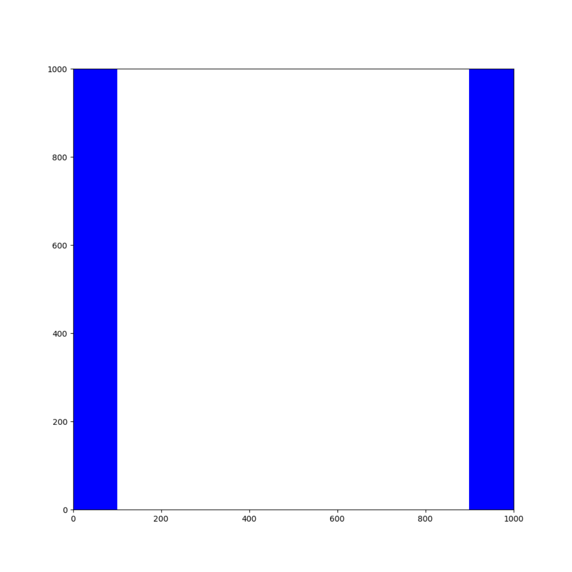
You can plot ibound, grid, head, and quiver separately.

`https://modflowpy.github.io/flopydoc/map.html`

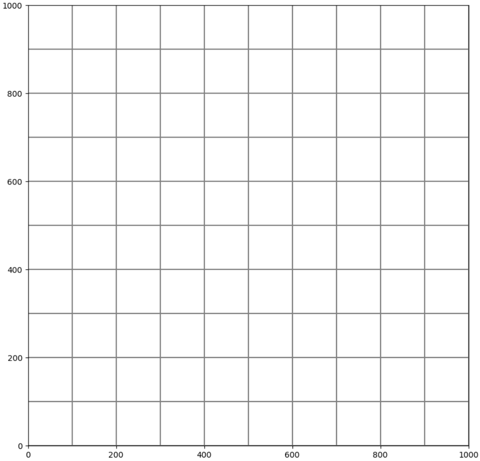
my_first_flopy_scipt.py (12) - Results



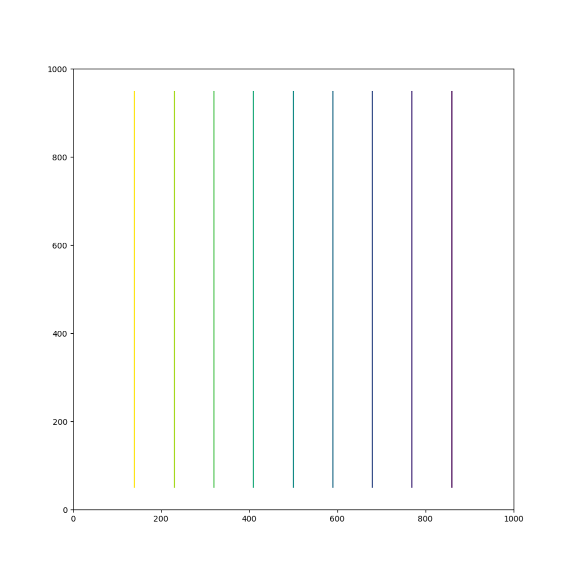
my_first_flopy_scipt.py (13) - IBOUND



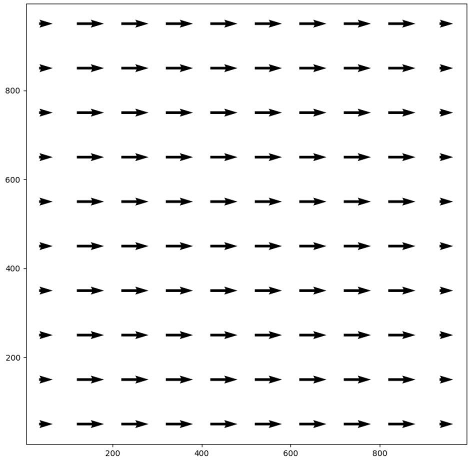
my_first_flopy_scipt.py (14) - Grid



my_first_flopy_scipt.py (15) - Head



my_first_flopy_scipt.py (16) - Quiver


800
600
400
200
200
400
600
800