Python Workshop
How we use python

Joseph D. Hughes

U.S. Geological Survey
Florida Water Science Center, Tampa, Florida USA

USGS National Groundwater Workshop, August 2012

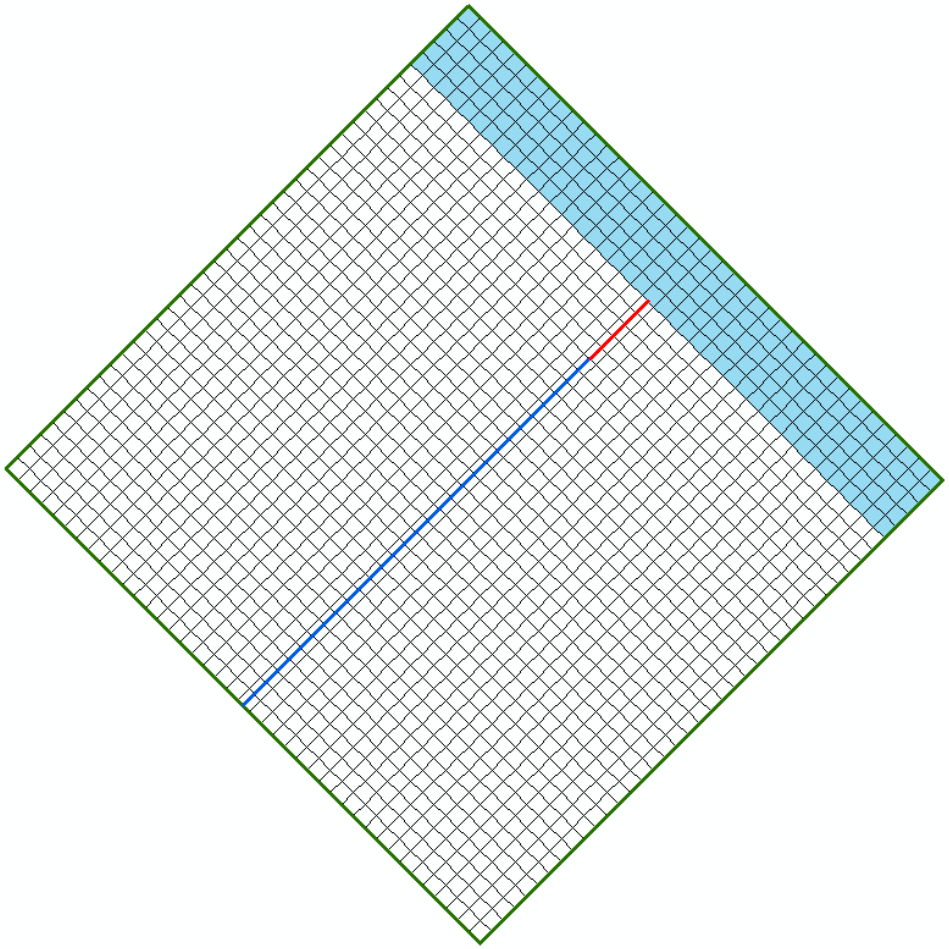
How we use python

1. Creating input files
   - From raw data
   - From model results
2. Processing output files
   - Reading binary head file
3. Making figures and animations
   - Using matplotlib
   - Animations
4. Making shapefiles
   - Get data from MODFLOW file
   - Make shapefile

USGS science for a changing world

Create GHB file from raw data (1)

[xoff,yoff,rot]=[1000,1000,45]



Create GHB file from raw data (2)

CreateCoastalGHB.py

```
1  import numpy as np
2  import MFArrayUtil as au
3  #--function for calculating equivalent freshwater head
4  def eqfwh( rho, h, z ):
5      return ( rho / 1000. ) * h - ( ( rho - 1000. ) / 1000. ) * min( h, z )
6  #--main script
7  #--spatial dimensions
8  nlay, nrow, ncol  = 1, 41, 40
9  dx, dy = 500.0, 500.0 #m
10 #--temporal dimensions
11 nper = 3
12 #--boundary condition data
13 icbc, start_coast = 0, 36
14 kh     = 10.000 #m/d
15 cond   = kh * dx * dy
16 #--read MODFLOW data from external files
17 ibound  = au.loadArrayFromFile(nrow,ncol,'..\\ref\\ibound.ref')
18 top     = au.loadArrayFromFile(nrow,ncol,'..\\ref\\top.ref')
```

Create GHB file from raw data (3)

CreateCoastalGHB.py

```
19  #--ghb dataset
20  nghb = 0
21  for ir in range(0,nrow):
22      for ic in range(0,ncol):
23          if ibound[ir,ic] == 2:
24              nghb += 1
25  f = open('Model.ghb','w')
26  #--dataset 0
27  f.write( '#Coastal Aquifer GHB Package\n' )
28  #--dataset 1
29  f.write( '{0:10d}{1:10d}  NOPRINT\n'.format( nghb, icbc ) )
30  #--write header for ghb file -- stress period 1
31  f.write( '{0:10d}          0          #STRESS PERIOD {1:05d}\n'.format( nghb, 1 ) )
32  for ir in range(0,nrow):
33      for ic in range(0,ncol):
34          if ibound[ir,ic] == 2:
35              f.write( ' {0:9d} {1:9d} {2:9d} {3:9.5f} {4:9.3g}\n'.format(nlay,ir+1,ic+1, \
36                       eqfwh( 1025., 0.0, top[ir,ic]),cond ) )
37  #--reuse data for the remaining stress period(s)
38  for iper in range(1,nper):
39      f.write( '{0:10d}          0          #STRESS PERIOD {1:05d}\n'.format(-1,iper+1) )
40  f.close()
```

Output GHB file from raw data (4)

Model.ghb

```
1   #Coastal Aquifer GHB Package
2          205          0  NOPRINT
3          205          0            #STRESS PERIOD 00001
4            1          1         36   0.00000   2.5e+06
5            1          1         37   0.05000   2.5e+06
6            1          1         38   0.10000   2.5e+06
7            1          1         39   0.15000   2.5e+06
8            1          1         40   0.20000   2.5e+06
9            1          2         36   0.00000   2.5e+06
10           1          2         37   0.05000   2.5e+06
11           1          2         38   0.10000   2.5e+06
12           1          2         39   0.15000   2.5e+06
13           1          2         40   0.20000   2.5e+06

210          1         41         36   0.00000   2.5e+06
211          1         41         37   0.05000   2.5e+06
212          1         41         38   0.10000   2.5e+06
213          1         41         39   0.15000   2.5e+06
214          1         41         40   0.20000   2.5e+06
215         -1          0            #STRESS PERIOD 00002
216         -1          0            #STRESS PERIOD 00003
```

Extract heads to create a new initial head array

extractSteadyHead.py

```
1  import numpy as np
2  import MFBinaryClass as mfb
3  #--problem size
4  nlay, nrow, ncol = 1, 41, 40
5  nper, nstp = 1, 100
6  #--name of MODFLOW head file
7  head_file = '..\\Results.SWI\\CoastalAquifer.hds'
8  #--read head data
9  #--create instance of head object from MFBinaryClass
10 headObj = mfb.MODFLOW_Head(nlay,nrow,ncol,head_file)
11 #--read array
12 totim,kstp,kper,h,success = headObj.get_record(nstp,nper)
13 #--save array or print error message
14 if success:
15     np.savetxt('..\\ref\\steady_ihead.ref',h[0,:,:])
16 else:
17     print 'Could not read Stress Period {0} Time Step {1}\n  from {2}'.format(nper,nstp,\
18                                                                              head_file)
```

Binary head data

plotHeads.py

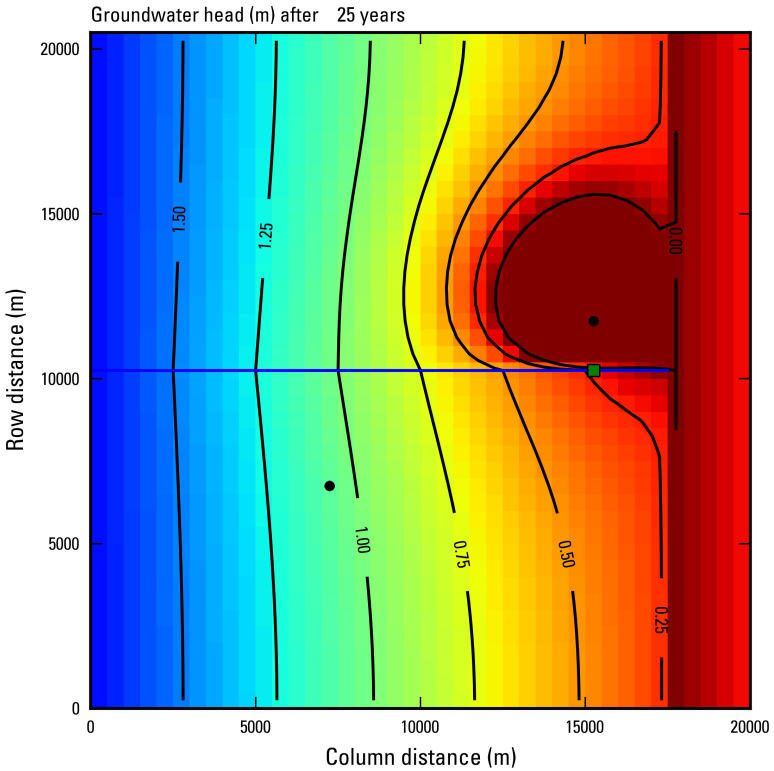
```
43  #--get available times
44  headObj = mfb.MODFLOW_Head(nlay,nrow,ncol,head_file)
45  t = headObj.get_gage(1)
46  ntimes = t.shape[0]
47  mf_times = np.zeros( (ntimes), np.float )
48  for i in range(0,ntimes):
49      mf_times[i] = t[i,0]


55  #--create figures for each output time
56  for ipos,on_time in enumerate( mf_times ):
57      #--build output file name
58      output_name = '{0}{1}_{2:05d}.{3}'.format(base_dir,base_name,int(ipos),extension)
59      fnames.append( output_name )
60      #--read head data
61      headObj = mfb.MODFLOW_Head(nlay,nrow,ncol,head_file)
62      totim,kstp,kper,h,success = headObj.get_record(on_time)
63      hd = np.copy( h[0,:,:] )
```

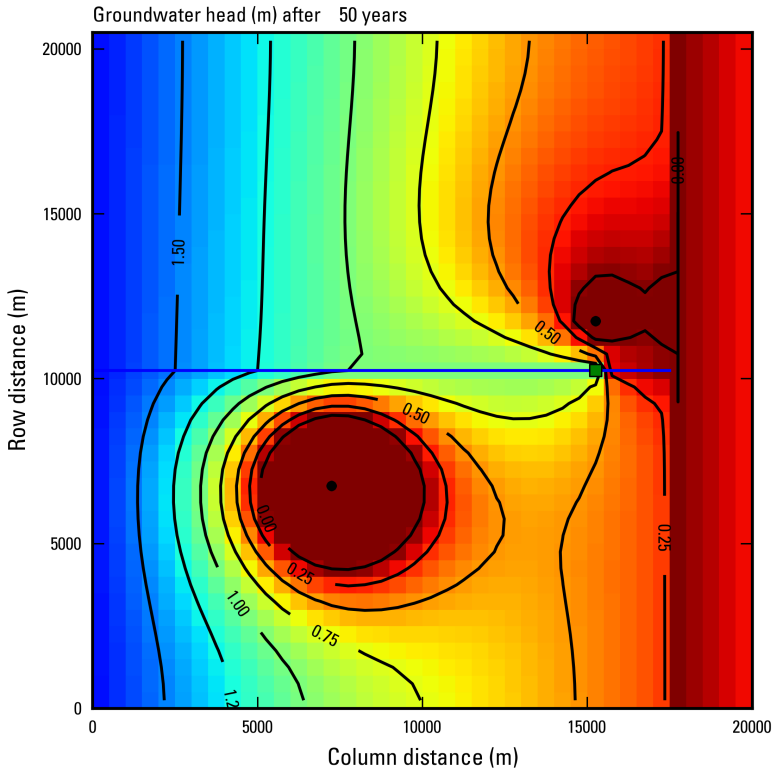
Figures (1)

plotHeads.py

```
84      hp = ax.pcolor(Xedge,Yedge,hd,\
85                     vmin=0,vmax=2,cmap='jet_r',alpha=1.0,edgecolors='None')
86      ch = ax.contour(xcell,ycell,hd,\
87                      levels=hdcontour,colors='k',linewidths=1)
88      ax.clabel(ch,inline=1,fmt='%5.2f',fontsize=6)
89      ax.plot([xedge[0],xedge[35]],[ycell[20],ycell[20]],linewidth=1,color='b',label='River')
90      ax.plot(xcell[struct_loc[1]],ycell[struct_loc[0]],'gs',markersize=4,label='Structure')
91      ax.plot(xcell[well_loc[0,1]],ycell[well_loc[0,0]],'ko',markersize=3,label='PW-1')
92      ax.plot(xcell[well_loc[1,1]],ycell[well_loc[1,0]],'ko',markersize=3,label='PW-2')
```

Figures (2)

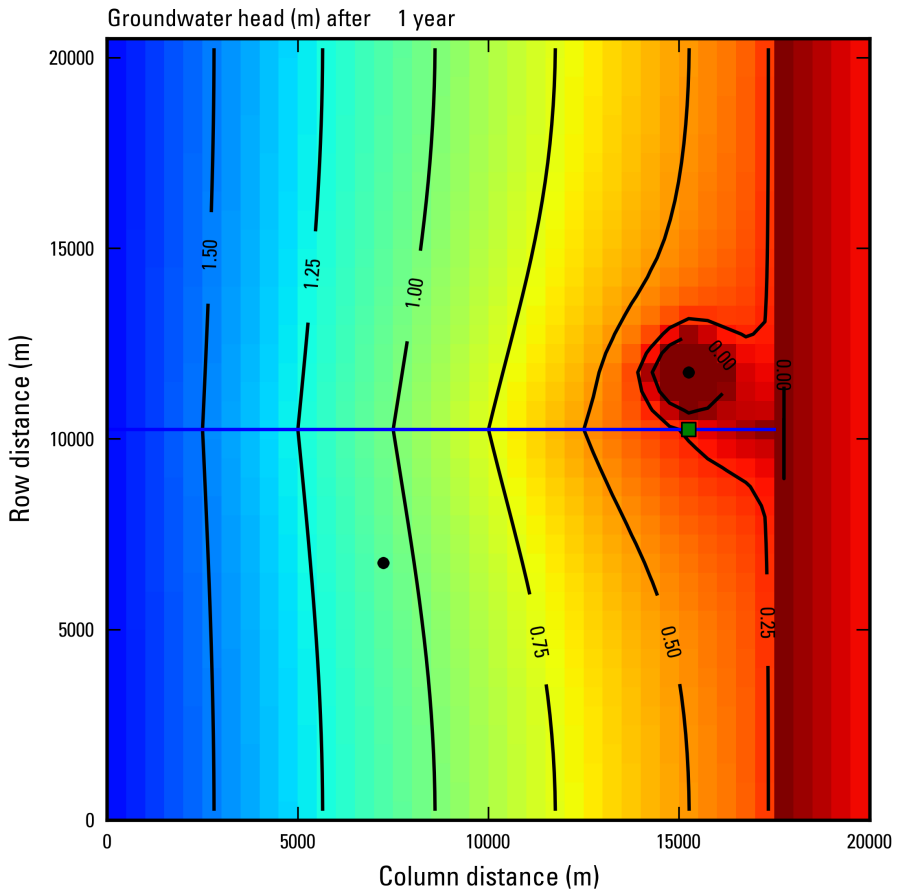

Groundwater head (m) after 25 years
Row distance (m)
Column distance (m)
0
5000
10000
15000
20000
1.50
1.25
1.00
0.75
0.50
0.25
0.00


Groundwater head (m) after 50 years
Row distance (m)
Column distance (m)
0
5000
10000
15000
20000
1.50
1.00
0.75
0.50
0.25
0.00

Using `ffmpeg.exe`

```
101  #--animate head data
102  coutf = '{0}{1}.swf'.format(base_dir,base_name)
103  cline = 'ffmpeg.exe -i {0}{1}_%05d.png {2} -y'.format( base_dir,base_name,coutf )
104  try:
105      os.remove(coutf)
106  except:
107      print 'could not remove...{0}'.format( coutf )
108  subprocess.call(cline, stdin=None, stdout=None, stderr=None, shell=False)
```

Using `ffmpeg.exe`



Making shapefiles (1)

makeShapefileFromDIS.py

```
4   import numpy as np
5   import shapefile
6   import MFArrayUtil as au


89  filename = '..\\data\\CoastalAquifer.dis'
90  offset,nlay,nrow,ncol,delr,delc = load_dis_file(filename)
91  #--flip the delc since we moved the orgin to lower left
92  delc = np.flipud( delc )
93  #--sum the lengths along the distance vectors
94  delr_cum = np.cumsum(delr)
95  delc_cum = np.cumsum(delc)
96  #--flip the delc since we moved the origin to lower left
97  delc_cum = np.flipud( delc_cum )
98  #--insert '0' in the first position
99  xedge = np.hstack((0,delr_cum))
100 yedge = np.hstack((delc_cum,0))
101 #--read MODFLOW data from external files
102 ibound  = au.loadArrayFromFile(nrow,ncol,'..\\ref\\ibound.ref')
103 top     = au.loadArrayFromFile(nrow,ncol,'..\\ref\\top.ref')
```

Making shapefiles (2)

makeShapefileFromDIS.py

```
104  #--create polygon shapefile of grid
105  wr = shapefile.Writer()
106  wr.field('row',fieldType='N',size=20)
107  wr.field('column',fieldType='N',size=20)
108  wr.field('delx',fieldType='N',size=20)
109  wr.field('dely',fieldType='N',size=20)
110  wr.field('cellnum',fieldType='N',size=20)
111  wr.field('ibound',fieldType='N',size=20)
112  wr.field('elev_m',fieldType='N',size=16,decimal=7)
113  #--create each polygon
114  cell_count = 0
115  for ir in range(0,nrow):
116      for ic in range(0,ncol):
117          #--calc the box points relative to the grid
118          lowleft = [xedge[ic],yedge[ir]]
119          lowright = [xedge[ic+1],yedge[ir]]
120          upright = [xedge[ic+1],yedge[ir+1]]
121          upleft = [xedge[ic],yedge[ir+1]]
122          closeit = [xedge[ic],yedge[ir+1]-0.0001]
123          this_box = [upleft,upright,lowright,lowleft,closeit]
```

Making shapefiles (3)

makeShapefileFromDIS.py

```
124             #--if rotation is non-zero
125             if offset[2] != 0.0:
126                 this_box = rotate(this_box,offset[2])
127             #--add the offset in after the rotation
128             this_box = add_offset(this_box,offset)
129             ibt    = ibound[ir,ic]
130             ttop   = top[ir,ic]
131             wr.poly(parts=[this_box], shapeType=5)
132             wr.record([ir+1,ic+1,delc[ir],delr[ic],cell_count+1,ibt,ttop])
133             cell_count += 1
134     #--save polygon shapefile
135     wr.save(target='..\\data\\CoastalAquifer_grid')
```