Listing of Python Utilities Used to Support WWUM Model Unit Response Function (URF) Update Process – April/May 2013

Filename: 20130423_WWUM_URFupdate.py

Other Versions: *20130408_WWUM_URFupdate.py (deprecated – reference only)*

Python Version: Python 3.x (written and interpreted through Python 3.2 using IDLE v3.2)

Description: The processes defined in this Python script/program allow the user to distribute some defined rate of pumping (injection or extraction) across MODFLOW groundwater model cells as defined by zones in a shapefile. The process makes use of functions that read and write MODFLOW-2000 well (*.WEL) files *[WelRead, WelWrite]* and assumes that the distributed pumping will be superimposed on an existing WEL file.

Inputs:

1) Path to land use shapefile – should be complete, valid file path and file name; can use *.shp, *.dbf, or *.shx file as target; this shapefile should be a complete representation of the model grid (inactive and active cells) and have an attribute field with zone identifiers

2) Path to the baseline/original pumping (WEL) file onto which additional pumping will be superimposed

3) Path to folder where new well (WEL) files should be written (input is folder, not file)

4) Pumping rate that should be distributed across zones defined in 1)

5) Interactive inputs for selecting attribute fields from input SHP

6) Must enter model dimensions (row, column, layers) manually (could be changed in future)

Outputs:

1) MODFLOW well (*.WEL) files for each unique zone identifier in the land use shapefile

2) A log file that records progress through the process (time, inputs, outputs, etc)

Listing of Python Utilities Used to Support WWUM Model Unit Response Function (URF) Update Process – April/May 2013

Filename: 20130417_ModelRunControl.py

Other Versions: *none*

Python Version: Python 3.x (written and interpreted through Python 3.2 using IDLE v3.2)

Description: The processes defined in this Python script/program allow the user to automate, with minimal inputs of filenames and locations, the process of running many repetitions of MODFLOW and ZoneBudget using different inputs for each run. This process uses function *NAMread031713* to incorporate filenames and references from a designated MODFLOW name file into a Python dictionary that can be used by other processes and functions (particularly to call and interact with other model files). This program was written to cycle through all files having a certain extension (in this case ".wel") in a directory and run MODFLOW with that file taking the place of whatever was there originally, a process that could easily be extended to other applications.

The process assumes that the MODFLOW and ZoneBudget executables are present in the home directory containing the MF name file. The option is available via the 'BatchZbFlag' variable to call ZoneBudget via a batch file rather than directly via the Python if desired.

Inputs: 1) **targetDir** : directory containing the WEL files you want to use during the batch model run - this directory must be within same root structure as the main mudflow files to be used - i.e. it cannot be in a branched directory or MODFLOW can't find them because absolute pathnames are too long [4.17.2013, jg]

2) **MfNamPath:** directory path to the MF Name file that will be the basis for this batch; this will be re-written to use the well files in the targetDir

3) **zoneFile:** the name (ONLY!) of the zone file in the SAME directory as the mudflow files that is to be used in running ZoneBudget

4) BatchZbFlag: manually change in code as desired: 1 = ZB run with batch file, 0 = ZB run using python interface

Outputs: 1) "RunLogs" directory (if one doesn't exist) and log files for MF and ZB run processes

2) ZB output directory (if one doesn't exist) to store ZoneBudget output files (assumes using ZoneBudget 3.01 with LST and 2.csv outputs – this

can be changed as needed) and ZoneBudget output files for each model run

Listing of Python Utilities Used to Support WWUM Model Unit Response Function (URF) Update Process – April/May 2013

Filename: 20130423_WWUM_URFupd_cxfreezeSetup.py

Other Versions: *none*

Python Version: Python 3.x (written and interpreted through Python 3.2 using IDLE v3.2)

Description: Setup script to guide the "freeze" process (using cx_Freeze package) of converting an interpreted Python 3.x version of WWUM_URFupdate to an executable and associated library files for portability onto machines without Python or needed packages

Inputs: Script file - 20130423_WWUM_URFupdate.py

Outputs: Executable (build\exe.win-amd64-3.2\20130423_WWUM_URFupdate.exe) and associated python and windows libraries (pyd and dll)

Filename: 20130513_ZBsummarize.py

Other Versions: *20130501_ZBsummarize.py (deprecated – reference only)*

Python Version: Python 3.x (written and interpreted through Python 3.2 using IDLE v3.2)

Description: This script performs automated data reduction and summary operations on ZoneBudgetv3.01 *.2.csv files within a directory to produce a set of summary files that include a time series of net rate, volume, and cumulative volume differences, all in model units. This process presumes there is a ZB file that represents some baseline conditions against which a comparison is desired that is reflected in water budget terms broken out in zones. The process will only work with information available in the ZoneBudget files – if a budget term was not saved to the CBC file to begin with, the results will be incomplete or erroneous.

Specifically, this process will determine the number of budget terms in a ZB *.2.csv file, automatically calculate the IN-OUT difference and can then subtract a designated baseline condition from those differences by stress period, time step, and zone.

The process uses function *DISread* which finds model dimensions in space and time as defined by MF discretization file and stores as Python dictionary. This information is used for dimensioning arrays and for calculating volumes from rates.

Inputs: 1) **BLzbout_fp** – full file name and path to baseline condition ZB file (*.2.csv)

2) zbnet_fp – optional ZB net budget term output

3) zbRunMinusBL - full file name and path to file that will be written as output from this process – analysis run minus baseline run results as volumetric rate (in model units)

4) zbRmBLcumul - full file name and path to file that will be written as output from this process – analysis run minus baseline run results as net volume difference at stress period as well as cumulative volume difference (in model units)

5) targetDir – full path to a folder (not file) that contains ZB files you'd like to process. Output files will be saved here as well

6) DisFilePath – full file name and path to MF discretization file (used for model dimensions and stress perid/time step information)

7) Naming/labeling conventions can be adjusted in code as needed

Outputs:

1) "xxxxx_NetRateDiffs.csv" – comma-separated formatted text file containing net volumetric rate differences for budget terms by zone, unique runID, stress period, and time step

2) "xxxx_NetVolDiffs.csv" - comma-separated formatted text file containing net volume and cumulative volume differences for budget terms by zone, unique runID, stress period, and time step