Methods Note/

ListingAnalyst: A Program for Analyzing the Main Output File from MODFLOW

Richard B. Winston[1] and Scott Paulinski[2]
[1]Corresponding Author: U. S. Geological Survey, MS 431
12201 Sunrise Valley Dr.
Reston, VA 20192
USA
rbwinst@usgs.gov

[2] U. S. Geological Survey
California Water Science Center
Sacramento, CA 95819
USA
spaulinski@usgs.gov

Received November 2012, accepted February 2013.

doi: 10.1111/gwat.12054

Abstract

ListingAnalyst is a Windows® program for viewing the main output file from MODFLOW-2005, MODFLOW-NWT, or MODFLOW-LGR. It organizes and displays large files quickly without using excessive memory. The sections and subsections of the file are displayed in a tree-view control, which allows the user to navigate quickly to desired locations in the files. ListingAnalyst gathers error and warning messages scattered throughout the main output file and displays them all together in an error and a warning tab. A grid view displays tables in a readable format and allows the user to copy the table into a spreadsheet. The user can also search the file for terms of interest.

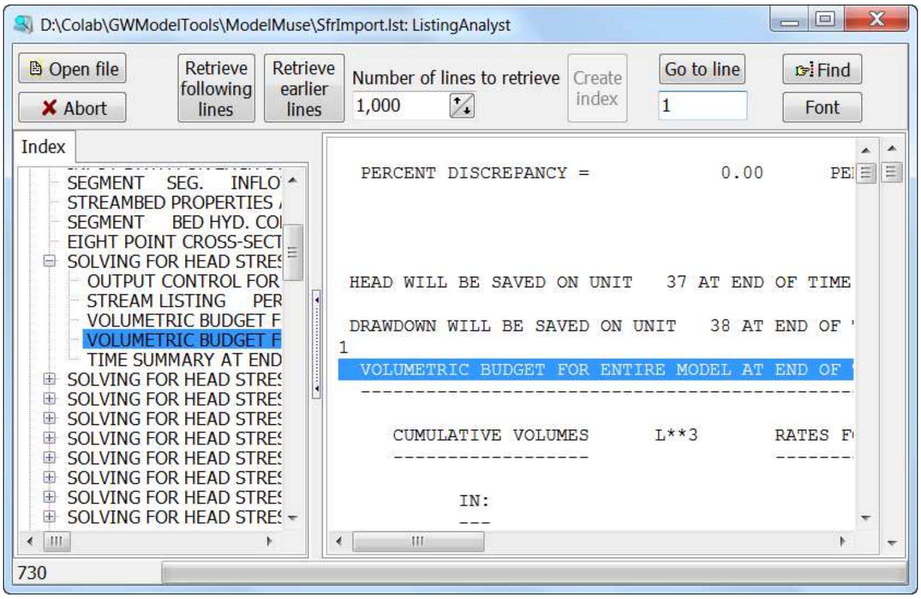
Introduction

As computers have become more and more powerful, hydrologists have been able to simulate groundwater flow in greater and greater detail. Models that once would have been impractical because of their long run time are now feasible. With these larger models come larger output files. The main output (listing) file produced by MODFLOW-2005 (Harbaugh, 2005) contains a wealth of information including printouts of the input data, calculated heads, drawdowns, warning messages, and error messages. Analyzing these large output files can present a challenge to the modeler. Large files are slow to open in some text editors. While text editors that efficiently read large files do exist, they do not organize listing files by section and subsection. It can be difficult to locate the information of interest because information, including error and warning messages, is scattered throughout the file. Searching for error and warning messages can be especially troublesome. Although the user can search for the terms “error” and “warning,” not all the error and warning messages contain those terms so some could easily be overlooked. This article describes a new computer program designed to alleviate problems handling large listing files from MODFLOW.

Description of ListingAnalyst

ListingAnalyst is a new, public-domain, MODFLOW utility program intended to facilitate analysis of listing files for MODFLOW models. To facilitate navigation through the listing file, the program automatically searches for headings in the listing file and displays them in a tree-view control when a file is first opened (Figure 1). The headings for which it searches were determined through examination of the source code of MODFLOW-2005, MODFLOW-NWT (Niswonger and others, 2011), MODFLOW-LGR (Mehl and Hill, 2005), and GSFLOW (Markstrom and others, 2008). The program also searches for error and warning messages and lists them separately (Figure 2). These error and warning messages were also identified through examination of the source code.

When ListingAnalyst opens a MODFLOW listing file, instead of attempting to load the entire file in memory, it only displays a user-specified number of lines. The user can choose which lines to display by specifying a starting line or using a scroll bar. Because only a small part of the file is displayed at any time, the amount of memory used is manageable. The program has a search function to allow the user to search the entire file for any text of interest.



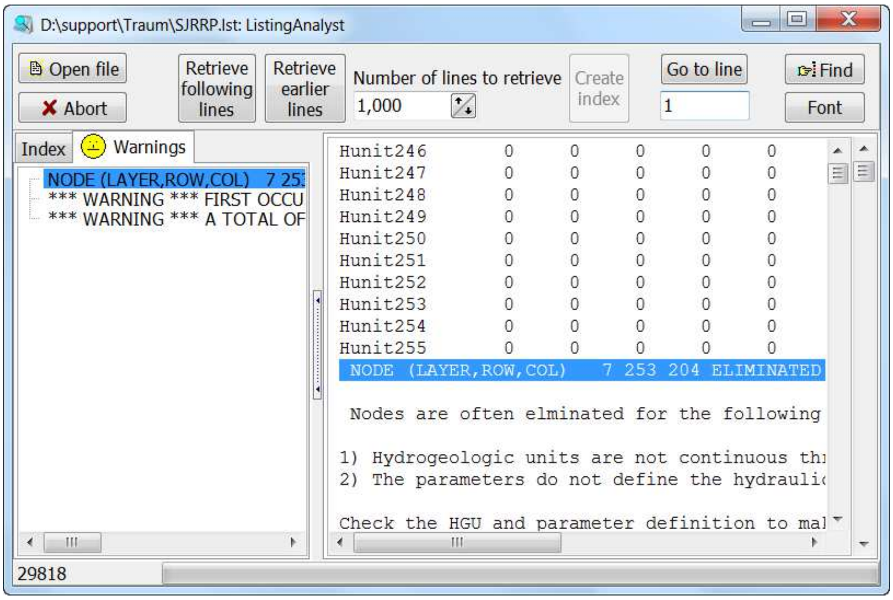
Number of lines to retrieve 1,000

Go to line 1

730

```
  PERCENT DISCREPANCY =              0.00       PE



 HEAD WILL BE SAVED ON UNIT    37 AT END OF TIME

 DRAWDOWN WILL BE SAVED ON UNIT    38 AT END OF '
1
  VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF
  ---------------------------------------------

     CUMULATIVE VOLUMES      L**3       RATES F
     ------------------                 -------

          IN:
          ---
```

Figure 1. ListingAnalyst showing extracted section headings of the listing file.



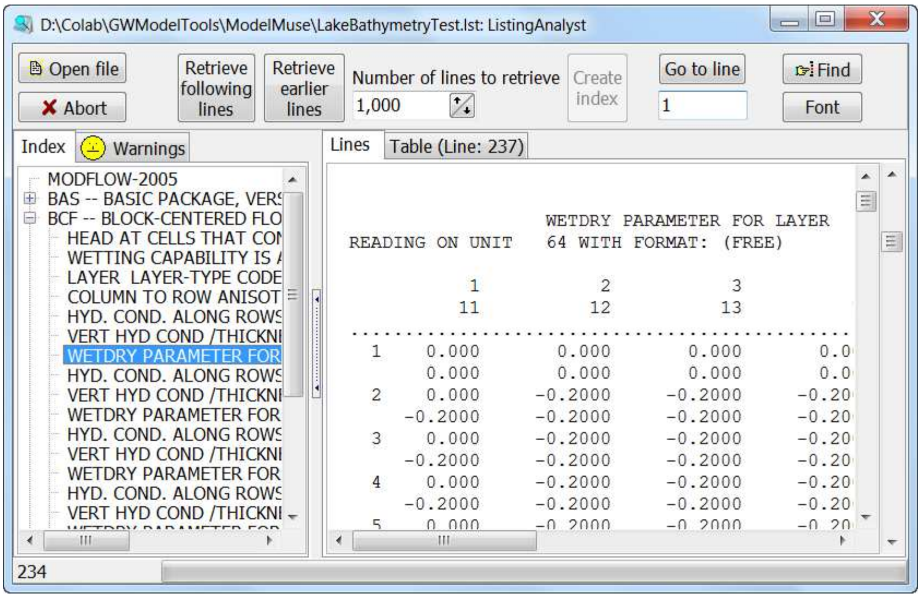
```
NODE (LAYER,ROW,COL)   7 25
*** WARNING *** FIRST OCCU
*** WARNING *** A TOTAL OF
```

```
Hunit246          0       0       0       0       0
Hunit247          0       0       0       0       0
Hunit248          0       0       0       0       0
Hunit249          0       0       0       0       0
Hunit250          0       0       0       0       0
Hunit251          0       0       0       0       0
Hunit252          0       0       0       0       0
Hunit253          0       0       0       0       0
Hunit254          0       0       0       0       0
Hunit255          0       0       0       0       0
 NODE (LAYER,ROW,COL)    7 253 204 ELIMINATED

 Nodes are often elminated for the following

1) Hydrogeologic units are not continuous thı
2) The parameters do not define the hydrauli(

Check the HGU and parameter definition to mal
```

Figure 2. ListingAnalyst screen showing extracted warning messages.

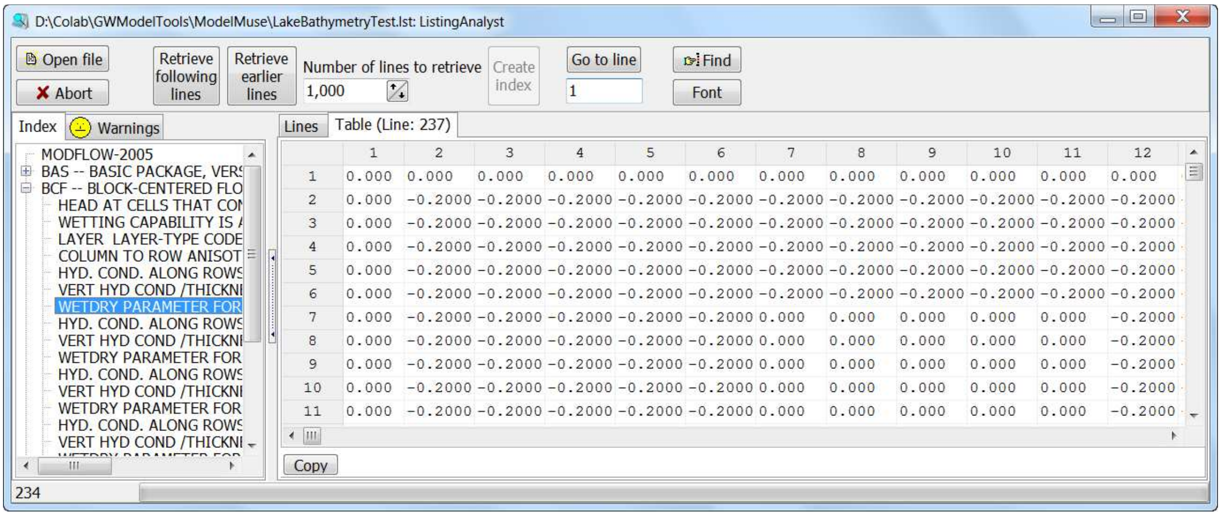
Implementation

Using a method described below, ListingAnalyst can recognize tables of heads, drawdown, input arrays, and observations in a listing file. When it recognizes a table, it places the content of the MODFLOW table in a table view from which it can be easily copied to the clipboard in a format recognized by spreadsheet programs (Figures 3 and 4). MODFLOW writes arrays in either a “strip” format or a “wrap” format. In the “strip” format, each block of up to 10 columns is printed in a separate table. In the “wrap” format (Figure 3), all the data are in one table but only 10 values are printed on each line so multiple lines may be required to print the data for one row. ListingAnalyst reformats tables in wrap format so that each model row is represented by one row in the table, and each model column is represented by one column in the table (Figure 4).



```
                          WETDRY PARAMETER FOR LAYER
 READING ON UNIT   64 WITH FORMAT: (FREE)

             1           2           3
            11          12          13

 ...............................................
   1     0.000       0.000       0.000       0.0
         0.000       0.000       0.000       0.0
   2     0.000     -0.2000     -0.2000     -0.20
       -0.2000     -0.2000     -0.2000     -0.20
   3     0.000     -0.2000     -0.2000     -0.20
       -0.2000     -0.2000     -0.2000     -0.20
   4     0.000     -0.2000     -0.2000     -0.20
       -0.2000     -0.2000     -0.2000     -0.20
   5     0.000     -0.2000     -0.2000     -0.20
```

Figure 3. MODFLOW data displayed in "wrap" format.



	1	2	3	4	5	6	7	8	9	10	11	12
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000
3	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000
4	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000
5	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000
6	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000
7	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	0.000	0.000	0.000	0.000	0.000	-0.2000
8	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	0.000	0.000	0.000	0.000	0.000	-0.2000
9	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	0.000	0.000	0.000	0.000	0.000	-0.2000
10	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	0.000	0.000	0.000	0.000	0.000	-0.2000
11	0.000	-0.2000	-0.2000	-0.2000	-0.2000	-0.2000	0.000	0.000	0.000	0.000	0.000	-0.2000

Figure 4. The same data in Figure 3 displayed in a ListingAnalyst table from which it can easily be copied to the clipboard.

To identify a table, the program first tests whether the current line is one of the lines written at the beginning of a table of observation results such as “HEAD AND DRAWDOWN OBSERVATIONS.” If it is, it searches for a line starting with “----” and identifies the following line as the start of the table of observation results. If a table of observation results is not found, the program starts searching from the current line for a line starting with “........” If it finds such a line before encountering 3 blank lines, it tentatively identifies the following line as the start of a table. It then searches backwards for either a blank line or a line containing “ 1 ” to identify the column labels for the table. Once a table has been identified, it is read and displayed in a new tab (Figure 4). A separate tab shows the normal view of the listing file (Figure 3).

The method used to find the titles, error messages, and warning messages (collectively referred to as "search terms") has been optimized for speed. Finding search terms has been optimized by minimizing the number of comparison operations required during indexing. This has been accomplished by storing the search terms in a trie structure (Parsons, 1995). Using a trie structure minimizes the number of comparison operations when indexing multiple search terms. In addition, the number of comparison operations has been minimized by eliminating lines and blocks of text from the search. There are only a few cases where a line containing a heading, a warning message or an error message starts with a number. ListingAnalyst first tests whether a block of lines contains one of those rare instances using the Boyer-Moore string searching algorithm (Boyer and Moore, 1977; Bucknall, 2007). If the block of lines does not contain one of those cases, then any line whose first non-blank character is a number or negative sign can be skipped. The trie structure is then used to search the remaining lines for any of the search terms. There are a few cases where a search term is included in a line but it is not appropriate to create an index entry for the line. For example, when the term "RECHARGE" is included in the budget for a time step, it should not be indexed but if it introduces an array of recharge values, it should be indexed. Such cases are handled by using some other characteristic of the line to determine whether it should be indexed or not. In the case of "RECHARGE," the presence of "RECHARGE =" twice within the same line can be used to identify a line that should not be indexed.

The time required to read the listing file and identify the search terms is approximately linearly related to the file size (Figure 5). If a file contains a large number of error or warning messages, the program may take longer to read and identify the search terms than would be expected based on its file size alone. For example, one of the tested files in Figure 5 was 181 MB in size but took 12 seconds to read and identify the search terms whereas a 186 MB file took only 7 seconds. The longer time was required for the smaller file because it contained a large number of error messages. The times shown in Figure 5 are representative values. However, the time to read and identify the search terms has been observed to vary by up to 10% when measured on the same computer with the same file. More consistent speeds can be achieved by limiting as much as possible all other processes on the computer. All speed tests were performed on a computer with an Intel® Xeon® CPU with a rated speed of 1.87 GHz and 6 GB of RAM. The operating system was Windows 7. The hard drives containing the files both had speeds of 7200 RPM.

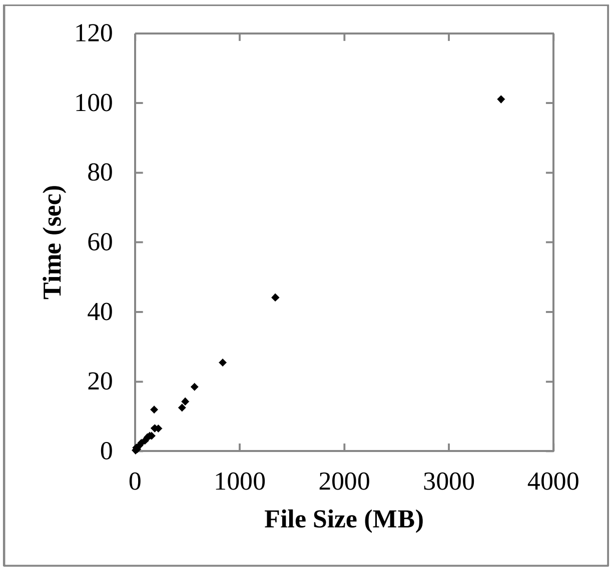


Figure 5. The time required to read a file increases linearly with file size. The time required to index a file increases with file size but also displays variability.

Typical usage

The user begins a session by opening ListingAnalyst and then clicking the Open File button to select a MODFLOW listing file. An extra checkbox is included on the Open File dialog box labeled “Index file.” If checked, an index of the file will be created. Otherwise the file will be opened without being indexed. Other ways to open a file are to drag and drop a file into ListingAnalyst or to start ListingAnalyst from the command line and specify the name of the file to open on the command line. In these cases, the file will be indexed automatically.

Once a file is opened, the lines at the beginning of the file are placed in a text box and the user is given several choices.

1. The user can navigate to the section of listing file that contains the index entry or error or warning message by double clicking on an entry in the index or one of the error or warning messages.
2. The user can navigate to a specific line by entering the line number and clicking the “Go to Line” button.
3. If there is a particular word that the user wants to find in the file, the user can click the Find button and search for it within the file.
4. The user can use the outer scroll bar to move to a different section of the file. (The inner scroll bar moves to a different location in the portion of the file that is currently being displayed.)

If too many or not enough lines are included in the text box, the user can control the number of lines in the text box with the “Number of lines to retrieve” edit box. This controls the number of lines that will be placed in the text box the next time the user navigates to a different point of the file. The user can also click the “Retrieve earlier lines” or “Retrieve following lines” buttons to show lines before or after the lines that are currently in the text box. The number of additional lines that will be placed in the text box is controlled by the “Number of lines to retrieve” edit box. If the time required to open or index a file is too long, the user can click the “Abort” button to stop reading or indexing the file. The user can control the font used to display the file by clicking the “Font” button.

The listing file may contain tables of heads, drawdown, input arrays, and observations. When the user selects a section from the tree-view containing one of these tables, ListingAnalyst displays these tables in a grid view. For tables containing spatial information, the rows and columns of the grid view are numbered according to the rows and columns of the MODFLOW model. The grid view both allows the user to view tables in an easy to read format and allows the user to copy/paste the table into a spreadsheet.

Limitations

ListingAnalyst is designed to display large files, but it is not designed to edit or print files. Users who wish to edit or print large files should choose another program. ListingAnalyst requires exclusive access to the file that it is displaying so no other program can open the file while ListingAnalyst has it open. In particular, if the user wishes to rerun the model and generate a new listing file, ListingAnalyst will need to be closed first.

Availability

ListingAnalyst is available through http://water.usgs.gov/software/lists/groundwater/ and/or http://water.usgs.gov/nrp/gwsoftware/modflow.html.

Conclusion

ListingAnalyst is a useful tool for displaying the listing file from various MODFLOW programs. By indexing the file, it allows the user to navigate quickly to the sections of the file that are of interest. It also creates lists of any error and warning messages. Without such a list, the user might not realize that the error or warning messages were present. ListingAnalyst does not keep the entire file in memory. This allows it to handle large files that might otherwise be too large to open or take an excessively long time to open.

Acknowledgements

We would also like to thank Leonard Konikow and Glen Carleton and four anonymous reviewers for suggestions on improving the manuscript and program.

References

Boyer, R.S., and J.S. Moore, 1977. A fast string searching algorithm: *Communications of the ACM* 20, no. 10: 762–772.

Bucknall, J. M. 2007. String searching: *PCPlus* 260: 146-147.

Harbaugh, A.W. 2005. MODFLOW-2005, the U.S. Geological Survey modular ground-water model -- the Ground-Water Flow Process. U.S. Geological Survey Techniques and Methods 6-A16. Reston, Virginia: USGS.

Markstrom, S.L., R.G. Niswonger, R.S. Regan, D.E. Prudic, and P.M. Barlow. 2008. GSFLOW-Coupled Ground-water and Surface-water FLOW model based on the integration of the Precipitation-Runoff Modeling System (PRMS) and the Modular Ground-Water Flow Model (MODFLOW-2005). U.S. Geological Survey Techniques and Methods 6-D1. Reston, Virginia: USGS.

Mehl, S.W., and M.C. Hill. 2005. MODFLOW-2005, the U.S. Geological Survey modular ground-water model -- documentation of shared node local grid refinement (LGR) and the Boundary Flow and Head (BFH) Package. U.S. Geological Survey Techniques and Methods 6-A12. Denver, Colorado: USGS.

Niswonger, R.G., S. Panday, and M. Ibaraki. 2011. MODFLOW-NWT, A Newton formulation for MODFLOW-2005. U.S. Geological Survey Techniques and Methods 6-A37. Reston, Virginia: USGS.

Parsons, T.W. 1995. *Introduction to Algorithms in Pascal.* New York, John Wiley & Sons, Inc., 447 p.