Building the Stream Package
RGDSS Memorandum
FINAL

To: File
From: Dr. Willem A. Schreuder, Principia Mathematica
Subject: Building the Stream Package
Date: December 29, 2015

The MODFLOW stream package is used to represents natural streams, drains, canals modeled as streams and McIntire Springs in the groundwater model. This document describes how the stream package input file is built using a **makefile** to control the individual steps.

This document includes the following sections:

0.0 Disclaimer

This program is furnished by The State of Colorado (State) and is accepted and used by the recipient upon the expressed understanding that the State makes no warranties, express or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the State shall be under no liability whatsoever to any person by reason of any use made thereof.

The program herein belongs to the State of Colorado. Therefore, the recipient further agrees not to assert any proprietary rights therein or to further represent this program to anyone as other than a State program.

1.0 Acknowledgment

Willem A Schreüder of Principia Mathematica developed the programs used to build the stream package using funding provided by the Colorado Division of Water Resources and the Rio Grande Water Conservation District.

2.0 Introduction

The MODFLOW stream package is used to represents natural streams, drains, canals modeled as streams and McIntire Springs in the groundwater model. The MODFLOW stream package in preference to the older river package as many of these streams and drains dry out in places. The original stream package was modified in order to correctly simulate diversions where the diverted amount exceeds the flow in the stream, and to calculate the stage from a stage-discharge curve.

Building the stream package is a multi-step process. It merges data from numerous sources. The build process was therefore set up as dependency tree specified as a **makefile**. This ensures that when files are updated, all the necessary steps are executed in the right order.

The stream package is built as a sequence of steps, starting with the GIS coverage of the streams to be modeled, and ending with the MODFLOW stream package. Along the way, data from other sources are merged in order to provide all the other information required such as flows, diversion locations, and the like.

Several programs are used to perform the various steps. Of these, the major programs are name **mkstr**, **mkq**, **build** and **ModEx**.

The GIS coverage of the streams is read by **mkstr**. This program takes individual line segments that make up stream segments in the GIS coverage and combines them into a single compound line segment for each stream reach. The National Elevation Dataset (NED) Digital Elevation Model (DEM) is then used to turn the two-dimensional GIS coverage into a three dimensional coverage, which determines the downstream direction.

The **mkq** program reads flow quantities for all stream flows, diversions, and return flows, and prepares files with flows for each of the stress periods to be simulated. It is assumed that the stream network remains static over all stress periods, and only the user specified flows, diversions and flows change from one stress period to the next.

The **build** program does most of the data integration. The three-dimensional stream network without diversions produced by **mkstr** is combined with vertical survey information to produce a stream network with the elevation of the stream bottom. The virgin stream network is then broken into segments separated by diversions and return flow locations. The program also specifies the stream parameters such as stage, width, conductance and depth.

Finally the **ModEx** program is used to discretize the stream network and produce a MODFLOW stream package.

3.0 Miscellaneous Pre-processing

The process of building the stream package works in the groundwater model coordinate system. Most data file, however, are in UTM coordinates. A number of different pre-processing steps therefore create .mod files which are text files with the relevant data in model coordinates. Most of these files are used by the **build** program to create the final GIS stream description.

The **mkgage** program creates the gage.mod file using the list of gages in gage.dat and locations in GIS/General/div3_flowstations.dbf.

The **mkdiv** program creates the diversions.mod file, which specifies the diversions along the stream network. Diversion locations are specified in GIS/General/div3_diversions.dbf. The diversions to simulate are specified in StreamGW/diversions.dat. This file lists each stream and then the diversions to simulate from that stream in order from top to bottom. The location of special diversions can be specified in divloc.mod.

The **mkret** program creates the return.mod file, which specifies the location of surface returns to the stream network. The locations are specified relative to diversion locations as either above or below them. The %fix structure in **mkret** is used to specify locations that are difficult to otherwise set such as where the Empire Canal dumps into the Alamosa and La Jara creeks.

The flowline.mod file is created using the c_survey script. This sctipt specifies surveyed elevations along various streams.

4.0 The mkstr program

The GIS coverage of the stream network is the starting point of building the stream package. This coverage consists of a set of polylines, defined as a set of *(x,y)* points. The coverage originated with the USGS 1:100,000 digital line graphs (DLGs), which the RGDSS GIS contractor coded by associating a stream name.

The DLGs are a topological coverage, which means that wherever lines in the coverage join, a separate polyline starts. Within each 7½ minute quadrangle, the polylines join exactly. However, between 7½ minute quadrangles, the polylines defining a stream may be disjoint by about a hundred feet. The polylines in the coverage also has no direction of hydrologic significance. Therefore some of the polylines are in the downstream direction while others are in the upstream direction.

The **mkstr** program overcomes these difficulties by as follows:

1. Each stream is stored in a separate file. The name of the file defines the name of the stream used by **mkstr.** The extension **.str** is used to define stream segments, while the extension **.div** is used to define diversions.

2. The file consists of a set of polylines, starting with the word LINE and ending with the word END. Between the key words, a series of *(x,y)* pairs appears. After the *x* and *y* values any comments can appear. The program **shape2ste** was used to convert the ArcView GIS coverage to a suitable text format. The **rgrep** program was then used to extract the lines corresponding to the appropriate stream. The coordinates are in UTM Zone 13 NAD 27.

3. When a polyline is not preceded by LINE, the line is ignored. Since the GIS coverage extends beyond the groundwater model domain, the LINE key word was changed to XLINE for those polylines that should be removed. In addition, any *(x,y)* pair preceded by a pound sign (#) are ignored. The text files extracted from the GIS coverage was hand edited to remove those polylines or parts of polylines outside of the domain.

4. The set of polylines that make up a stream are combined into a single polyline by matching the head and tail of individual polylines. This procedure was used to degenerate sets of polylines into longer polylines that represent major reaches in the stream network.

5. Where more than two polylines meet, a confluence or diversion in the stream network occurs. Diversions are identified by the **.div** file extension. Individual stream segments are given unique names by adding an integer 1, 2, … to the stream name.

6. Whenever the same stream continues at the join between exactly two polylines, the **mkstr** program will join those polylines except when the key words <BREAK> or <BREAK~> occurs after LINE. These key words can be used to start a new segment at the beginning or end of the polyline, respectively. This key word was used to break long segments of streams where a change in stream properties occur.

7. The order of the *(x,y)* pairs are not important as **mkstr** will determine the downstream direction for itself based on the DEM data. However, when the key

word <FORWARD> or <REVERSE> appears after LINE, the order of the *(x,y)* pairs are taken to be downstream or upstream, respectively.

8. After joining as many polylines as possible in the stream network, **mkstr** uses the National Elevation Dataset (NED) to evaluate the elevation of the ground surface at each of the *(x,y)* coordinates. This provides the best estimate of the ground surface elevation at the location of the stream. The NED data are spaced one third arc second apart on a regular grid. At 40 degrees north, the spacing is approximately 33 feet in the north-south direction and approximately 26 feet in the east-west direction. The elevation at each *(x,y)* location is interpolated from four surrounding elevations using bilinear interpolation.

9. The **mkstr** program then uses the NED elevations to determine the downstream direction for each stream segment, unless the <FORWARD> or <REVERSE> key words are specified.

10. Finally, the **mkstr** program outputs a virgin stream network in downstream order. Downstream order is defined as an order that the stream flow at any point in the stream network is completely determined by upstream stream segments. For each stream segment, the output consists of LINE <segmentname>, followed by *(x,y,z)* triplets in downstream order, followed by END. The **mkstr** algorithm first outputs all stream segments where neither the top nor bottom of the segment is connected to another stream. These are the so-called orphan streams, that is streams not connected to the rest of the stream network. Next the headwater segments are output, that is streams with no connection at the upstream end. Finally, the rest of the stream network is output by iteratively scanning the remaining stream segments and outputting them after the upstream segment have been output. An error is generated if the entire stream network cannot be output in this way, which is indicative of a stream network beyond the capability of the MODFLOW stream package.

5.0 The mkq program

The purpose of the **mkq** program is to produce a single flow file with all the flows required as input to the MODFLOW stream package for each run. The input to the **mkq** program are a set of files extracted from hydrobase.

The **mkq** program reads estimates of the stream inflow on selected streams at the model boundaries, diversions for relevant diversions and return flows as estimated by **ModFate** from HydMod style files. From these flows, **mkq** calculates the flow in the North Branch of the Conejos and a corrected flow on La Jara Creek.

Return flows are named for the place where they return to the stream network. Where more than one return flow returns to the same location, **mkq** sums the values.

mkq is used to generate the flows needed for all the simulations. For every stress period, **mkq** calculates the appropriate average or copies the appropriate monthly volume. It then saves a flow file for each simulation containing flow volumes for all the stress periods appropriate to that simulation.

The **mkq** program is controlled by the **.qpr** file. The **.qpr** file consists of five sections. Sections are separated by one or more blank lines. Within a section a line starting with a # can be used as a comment without causing the next line to be the next section.

The first section contain keyword value pairs to specify the run types. The GRID keyword is used to specify a StatePP grid file. This grid file specifies the period for the different types of runs. The key words SS, NP, IP, AM MO and CY are used to specify the root for the output file name according to the type of run.

The second section contain name translations. This is primarily used to specify the inflow at the boundary in terms of a gage or similar quantity but renaming it to match the stream name. So, for example, the lines

```
08261000    Costilla
SANDUNCO    Sand
```

are used to set the flow on Costilla creek at the state line to the measured gage flow at USGS gage 08261000 and the flow on Sand Creek at the model boundary to the SANDUNCO gaged flow.

The third section specifies the input file names. Each line specifies an input file which is read. The key word **sum** may follow the file name. If a structure is named multiple times in the input file, the **sum** directive indicates that all the recorded flows will be added together. Absent the **sum** directive, multiple occurrences of a structure name for the same year will be treated as an error.

The fourth section allows new flows to be calculated. In Phase 6 this is used to calculate the flow into the North Branch of the Conejos and North Branch of the Rio Grande from diversions along those branches. The format is the new name (e.g. RioGrandeNB) followed by a sequence of quantities to add. The quantities are a plus or minus sign followed by a # followed by the structure number, e.g. +#2200593.

The fifth section allows flow quantities to be deleted. Each line consists of one or more values to delete. Unless a quantity is deleted or renamed, all the quantities in the input files are echoed to the output files.

The output files consist of a very wide table. Each line represents a time period, and the quantities are different columns on that line. The first line in the file names the structures, gages and similar locations. Subsequent lines start with the period name and a flow rate in cubic feet per day. For the steady state and similar data, the value represents the average for that period.

6.0 The build program.

The **build** program is the penultimate step in building the stream package. The purpose of the **build** program is to integrate the stream data and create files that can be read by **ModEx** to build the MODFLOW stream package.

The **build** program is controlled by the .bld file. This file contains two sections. The first section lists the stream name, layer, stream bed Kv, width, thicknes, incision depth and Manning roughness for every stream segment. A layer of 0 means the top layer. The incision depth can be specified as a range linearly applied to the segment. Instead of a roughness, a stage/discharge coefficient pair may be specified.

The second section in the .bld file is the output sequence. In this section, the user must specify the order in which the stream segments should be output. Each line consists of a stream name, a segment or range of segments, and optionally other parameters to add to the **ModEx** input file. Most commonly, the parameters are FLOW= and DIV= to specify the flow at the top of the segment.

It is very important that the output sequence be top to bottom as **ModEx** will build the stream segments in the same order as is specified here. **ModEx** will attempt to catch backwards references in the stream package, but MODFLOW will not flag such an error.

The **build** program starts with the output from the **mkstr** program. This consists of the three dimensional polylines defining the streams. This first step is to determine the elevations of the streams. This is done by combining the incision depth data with the survey data. The survey locations and elevations are read from the flowline.mod file. The survey locations are mapped to the streams to establish the elevation at the various points along the streams.

The elevations along the streams are then established by linear interpolation along the streams. Elevations are also interpolated between stream segments. Where two segments segments with elevation meet and the segments are upstream and downstream of the point where they meet, the elevation at that junction is set by linear interpolation. If more than two segments meet and at least one of the segments are upstream and at least one of the segments are downstream, the elevation at the junction is established by interpolation weighted by inverse distances. Whenever the elevation at a junction is established, the elevation is applied to all stream segments at the junction. When the upstream and downstream neighboring segments of a stream contains surveys, the elevations are also established by interpolation across the center segment.

While linear interpolation is applicable in most situation, there are a few exceptions where linear interpolation between surveys are clearly incorrect. In order to correct these situations, auxiliary surveys can be specified. When an asterisk (*) is specified as the elevation in flowline.mod, a survey point is generated based on the DEM data.

The incision depth is used to establish the elevation when survey data is not available. Since the elevations are interpolated from 30 foot spaced DEM data, narrowly incised stream channels would not be reflected in the NED elevations. By subtracting the incision depth from this elevation, a better estimate of the stream bottom elevation is obtained. The incision depth is specified for each segment as a single value, or a pair of values that are linearly interpolated from the start to the end of the segment.

Linear interpolation is also used to transition between surveyed elevations and the elevations derived from the DEM. A user specified distance (Smax, currently set at two miles) specifies the distance across which the transition takes place. The transition takes place starting with the last surveyed elevation by linearly transitioning between the extrapolated survey elevation and the elevation set by the DEM minus the incision depth.

After establishing the elevation at every point along the stream network, the next step is to insert point diversions, inflows and gage locations. Diversions are specified in the file diversions.mod. Each diversion is specified as an ***(x,y)*** location, diversion name and stream from which to divert. The diversion amount is determined by the name of the diversion in the flow file generated by **mkq**. The diversion is mapped to the nearest point on the named stream. When the distance from the mapped location to the stream is greater than the specified tolerance (two miles), the diversion is omitted and a warning message generated.

Gages locations are similarly specified in the file gage.mod. Gage locations are explicitly represented in the stream network, but cause no change in flow.

Return flows locations are specified in the file return.dat. The return is specified as a return name followed by the stream to which the return should occur. The location can be specified as an *(x,y)* location, or as a location relative to a diversion or gage. A relative location is specified as a plus or minus sign and the name of a diversion or gage. The return flow is located upstream or downstream along the stream at a fraction Rfrac (0.2) of the distance between the specified location and the next diversion or gage location, but no more than Rdist (one mile) from the specified location.

Diversions and returns are specified as stub flows. These are inflows or diversions that have no length and therefore have no stream/aquifer interaction. They are, however, significant to the stream as they represent an increase or decrease in the modeled stream flow. The flow volumes, like all other flows, are specified in the flow file. By default, the flow volume for each stress period is the name of the return flow. Therefore, return flow names must be valid variable names. To make the names into valid variable names, return flows upstream (+) and downstream (-) are named XXXX#YYYYYY and XXXX_YYYYYY, respectively, where XXXX is an abbreviation of the name of the stream and YYYYYY is the number of the diversion.

When necessary, the return flow amount can be specified as an arbitrary expression which appear as the optional fifth parameter in the return.dat file. This feature is used to represent the Empire Canal imports to the Alamosa and La Jara.

As diversion, return flow or gage location are inserted into the stream network, the segments in the virgin stream network are broken into sub-segments. Sub-segments are named for the segment with a period and a three digit sequence number. When multiple diversions occur at the same location, **build** degenerates the individual diversions into a single composite diversion. The composite diversion is simply the summation of all the individual diversions.

Finally, **build** saves the stream information to a file for **ModEx**. The order in which the stream segments are save are specified in the second section in the .bld file. When the stream segments are saved, parameters such as the stream width, stream bed vertical conductivity and thickness, and the like are added to each stream segment. These parameters are specified in the first section of the .bld file. When the roughness parameter is specified, the stage is calculated by MODFLOW in the regular manner. When the STAGE parameter is specified, the stage will be calculated using the formula

$$\text{Stage} = C\ Q^n$$

When the stage is specified in this way, the STAGE parameter is saved to the output file instead of the ROUGH parameter.

While the **build** does not test to ensure that the order in which stream segments are save are appropriate, an error will occur when **ModEx** is run with an inappropriate order in the stream network.

7.0 Using ModEx to generate the stream the stream package.

ModEx is a multi-purpose program for the generation of MODFLOW related calculations. One of the functions in **ModEx** builds a MODFLOW stream package from the polylines and parameters saved by **build**.

Each polyline is preceded by a command line. This command line consists of the name of the stream segment and a set of optional parameters. The parameters can be specified in terms of constants, or in terms of values that vary along the polyline, by using the reserved variable names *#1*, *#2*, etc. to denote the columns after the *(x,y,z)* values.

The **ModEx** input file defines the model grid, and then uses the STREAM command to turn polylines into a MODFLOW steam package. The FILE parameter is used to set the output file. The SOURCE parameter is used to define the input file containing the polylines. The **ModEx** program calculates the intersection of the polylines with the model grid and thence the length of the stream in each model cell. The stream conductance is calculated as this length times the specified width times the specified stream bed hydraulic conductivity divided by the stream bed thickness. The stream bed elevation **Stop** is calculated as the average *z* value from the polyline. The point at which the connection to the stream is broken **Sbot** is calculated as **Stop** minus the stream bed thickness.

The **ModEx** program outputs the stream cells in the order that they occur as defined by the polylines. Where multiple polyline segments occur in the same cell for the same polyline, only one cell that is the net result is generated.

When the stream stage is to be calculated, the **ModEx** program will output a set of parameters used for the stage calculations. The stream width, slope and roughness is the standard set of parameters used to calculate the stage. The slope is automatically calculated from the polylines. The MINSLOPE parameter can be used to specify a minimum slope to be used, as a slope of zero causes division by zero in the stage calculation. Alternately, when the stage is specified using the cQ^n equation, a negative width is saved, *c* replaces the slope and *n* replaces the roughness.

The **ModEx** program also determines tributaries and diversions from the endpoints of the polylines. When the DIV parameter is specified on the polyline header, a diversion is set. All other meeting points are assumed to be tributaries to the lower streams. An error is generated when the order in which stream segments are specified does not meet MODFLOW's strictly downstream order.

The stress periods are specified using the NPER and FLOW parameters. The number of stress periods are set using the NPER parameter. For each of the stress periods, a set of flows are reading from the FLOW file. The flow amounts at headwaters and diversions are determined for each stress period.

The **ModEx** program assumes that the stream network remains the same for all stress periods, and that only the flows change from stress period to stress period. The absence of diversions can be simulated using zero flows.

The **ModEx** input file for building the stream package contains STREAM commands for all the different simulations. The commands differ only in the number of stress periods specified and the flow file used to set the flows for each stress period in the simulation, as well as the name of the output file.

The **ModEx** parameter INSTRUMENT is used to instrument the stream package. It causes a HYDMOD file to be written that contains entries for the beginning and end of every stream segment, or the inflow or outflow for stub segments. Since the net gain or loss for a segment is simply outflow minus inflow, this allows gains and losses to be calculated for each segment, and also permits detailed budgets to be calculated by post-processing programs such as **mksum**.

8.0 Miscellaneous post-processing.

After the stream package is built, several post-processing steps are performed.

The **mkhyd** program is run in the **hyd** directory. This builds the HYDMOD package input file used to extract flows from the model.

The **mkseq** program reads the stream package an builds the **doc/sequence.htm** file which describes the stream segments in detail for further review.

The **mkz** program calculates the average stream elevation for each model cell with streams. This is used in the generation of the ET ground surface in the model.

The **mklab** program program creates a file showing the location of all the stream segments, diversions and gages for plotting.

9.0 Comments and Concerns.

The stream network uses GIS coverages developed during Phase 3 of the RGDSS. This has subsequently been refined. For example, the rerouting of Sand Creek and Cotton Creek are reflected in the current stream network. However, the National Hydrography Dataset (NHD) is now available and could potentially be used to replace some or all of the older dataset developed for the RGDSS.

The Stream Routing package was modified to overcome problems with the original

USGS package. These features are now represented in newer packages available with MODFLOW. These packages were not adopted before because they were not available or still being developed. However, that decision should be revisited occasionally. If so the above procedures would need to be adapted to the new package.

10.0 References

Appendix B1: Canals as Streams, RGDSS Memorandum Task 41.4, Ray R. Bennett, November 7, 2002.

Appendix C1: Stage-Discharge Reltionship, RGDSS Memorandum, Willem A. Schreüder, July 26, 2004.

Appendix U: Building the Steam Package, RGDSS Memorandum, Willem A. Schreüder, July 26, 2004.