Performing Groundwater Model Runs
RGDSS Memorandum
FINAL

To: File
From: Principia Mathematica – Willem Schreuder
Subject: RGDSS Ground Water – Steps used to perform a complete set of groundwater model simulations for a particular version of the RGDSS Groundwater Model
Date: March 30, 2016

An update of the RGDSS groundwater model generally involves building the model input files, building the PEST control files, using PEST to estimate new best aquifer parameters, performing a suite of model runs with the new parameters, performing the impact runs, performing the response function runs, and post-processing of the results. This document provides an overview of the steps involved. Other documents describe some of the individual steps in more detail. This document includes the following sections:

0.0 Disclaimer

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

1.0 Acknowledgment

The procedures used during Phase 6 of the Rio Grande Decision Support System (RGDSS) Groundwater Model (Model) as defined by Model Version 6P98, were developed by Willem A. Schreüder of Principia Mathematica under the direction of the Technical Advisory Committee, also known as the Peer Review Team, using funding provided by the Colorado Division of Water Resources and the Rio Grande Water Conservation District. Some procedures were developed by Jim Slattery, Chuck Brendecke, Jim Brannon and James Heath.

2.0 Introduction

The purpose of this memorandum is to describe at a high level, all the processing that is involved in creating a new version of the model. Since modeling is inherently an iterative process, it is assumed that a previous version of the model is available to provide initial values of, for example, aquifer parameters.

The RGDSS Groundwater Model (the "Model") is the culmination of a long line of data processing operations that builds input data sets for the Model. The groundwater model than calculates the resulting flows through the aquifer and predicts heads and stream gains as a result of these stresses. The Model is then also applied to evaluate various changes in stresses such as switching off some wells. Calculating the difference between heads or stream gains from one simulation to another is used to calculate the impacts, so these runs are typically referred to as impact runs.

The data processing is done using scripts rather than manual manipulation of the data. This approach is desirable because it makes it easier to perform the steps consistently and correctly. Generally speaking, the scripts produce only minimal output to the screen, unless errors are encountered.

This memorandum does not discuss the algorithm involved in each processing step in detail because there are numerous other memoranda that delve into those details. Instead, the intention here is to describe an outline of the process.

A new version of the Model is typically created when one or more of the modeled stresses are changed, or some refinement to the model calibration is applied. The term "version" does not imply an entirely new model, but rather reflects gradual refinements that are applied.

For each version of the model, there is typically a set of calibration runs performed using PEST. This may involve thousands of individual model runs which PEST uses to optimize the parameters and the files associated with these runs are typically not retained.

Once the parameters are optimized, a suite of model simulations are run that represent different conditions such as a steady state, average monthly, initial period, monthly transient, and the no pumping verification run. Some of these runs are simply diagnostic in nature. The primary simulation is the monthly transient which is considered to be the best representation of historical conditions and is used during both the calibration and evaluation of the model.

Impact runs are typically performed using the monthly transient run but with the historical stresses altered to represent the changed condition for which the impact is evaluated. These runs evaluate the alternative historical condition assuming that, for example, a certain set of wells would not be operating. By calculating the difference in, for example, stream gains between the historical and the impact simulation, the stream depletions attributable to these wells are calculated.

To simplify the stream depletion calculations for administrative purposes, response functions can also be calculated for a given version of the model. The response functions use a set of model runs that create a one year long impulse followed by nineteen years of average monthly conditions. These responses can then be calibrated to match the model predicted impacts for the historical period.

In this memorandum, for consistency between sections, "pumping" is used to describe groundwater withdrawals whether the withdrawal is by a mechanical pump or if the well flows under artesian pressure.

The following section describes the process of creating a new version of the model and how all the simulations are performed.

3.0 Model Process

The naming convention used in the model files are described in *Directory and File Naming Convention*. Instead of naming all files associated with a particular simulation with the name of that simulation, the approach is to name the input files with the version number of that input source.

For example, the 6P98 model has a simulation name M6A00P98 which is the monthly transient simulation of historical conditions. This simulation uses stream package version M6P064, StateCU data version M6P033, boundary data version X6P018, aquifer property data version M6P098 and aquifer geometry data version X6P008. The purpose is to clearly identify which element(s) of the simulation has changed, so that there are not numerous files with different file names but containing the identical information. This improves the ability to track changes to model input data and reduces duplication of files.

3.1 Directory Layout

In order to avoid making copies of files which could lead to the accidental use of outdated files, a file layout is used so that a simulation can directly access files using a relative path.

Each phase of the RGDSS modeling is stored in a directory named for the phase, e.g. phase6. This is not required in order to run a simulation for a particular phase, but better compartmentalize different phases.

Appendix A shows the directory layout used for Phase 6 of the RGDSS. This layout must be maintained because various scripts expect data files to be located in specific directories.

3.2 Setting up the Model Geometry

The Model geometry is rarely changed. However, as new information becomes available, it may be necessary to change the active cells in the model or the thickness of layers.

The Model geometry is defined in the **prop** directory. The geometry of the model is defined in terms of a MODFLOW grid. For Phase 6, the model was updated to the North American Datum of 1983. To guard against using the older coordinate system by mistake, mod coordinates was redefined as an easting of 120939.39 m and a northing of 4095055.99 m in UTM Zone 13 with axes rotated 0.40807 degrees east of north around this point. Distances are measured in survey feet (1m=39.37 inches exactly).

The five layers in the model were defined by HRS during the initial phases of the model. These layers are largely used in Phase 6. However, based on a closer look at some localized areas, the layering has been refined during Phases 5 and 6. These refinements are discussed in various HRS memoranda.

Creating an updated model grid involves two steps. The first step is to create a new PM-DIAMOND input file by copying the previous grid generation input file, and adapting it to represent the needed changes to the grid. This script will generally read the old definition of the grid, activate or inactivate new cells as desired, determine the ground surface elevation from the 1/3 Arc Second National Elevation Dataset, calculate the bottom elevation for each layer as successive depths below this ground surface, and saving the grid definition as MODFLOW IBOUND files, a ground surface and aquifer bottom files, and create shape files for export.

A new version of the Model grid is generated using the command

diamond im8 out 1

The command will save the shape files in a ZIP file named grid8.zip. This file must be imported into the GIS processing to reflect the new spatial discretization of the model.

The second step involves creating a **.grd** file in the StatePP directory. The **.grd** file specifies the spatial and temporal discretization for the StatePP program. The **.grd** file also defined time periods such as the simulation period for the monthly model run. Therefore the **.grd** must therefore also be updated when the simulation period is changed.

3.3 Data Import

During Phase 6, the GIS processing, extracting data from HydroBase, and performing the StateCU and StateFate simulations were performed by DWR staff. The groundwater model runs were performed by Willem Schreuder at Principia Mathematica.

The first step in creating a new version of the Model is therefore to update the files located in the folders **GIS**, **Diversions**, **MIPumping**, **RimRecharge**, **StreamInflow**, **StreamGW** and **StateFate**. During Phase 6, this was accomplished by performing all the processing necessary to populate these directories and then uploading the files to the Principia web server. Updating the data in these directories are described in various other memoranda. The transfer step is, of course, only required if the groundwater model update is performed on a different file system.

3.4 GIS Processing

The GIS data are processed for individual structures. However, in StateCU and StateFate, some structures are combined into MultiStructures. It is therefore necessary to combine some of the GIS data so that the GIS mapping matches the structures in StateCU and StateFate. This is accomplished by the **agg** preprocessor.

The control file for **agg** is called **agg.par** and is describe in a separate memorandum. The **agg.par** file defines which files are used to perform the aggregation. The multi-structure definition is typically **Diversions/rg-ms.csv.** The **agg** program is run for each of the subdirectories representing a snapshot. On a Linux system this is done with the following command executed from the GIS directory:

```
for yr in 1936 1998 2002 2005 2009 2010;
do
  (cd $yr;ln ../agg.par $yr.par;agg $yr >$yr.err);
done
```

This step creates the files **irrwell.dat**, **divleak.dat divirln.dat** and **ditch.fact** in each of the year subdirectories.

The second step in the GIS preprocessing is to create GIS files for the “other” category of subirrigation. This is done by running the **mksub3** program as:

../src/mksub3 1936 1998 2002 2005 2009 2010

This creates a file **subacres3.dat** in each of the year subdirectories.

The final step in GIS preprocessing is to run the **mkprf** preprocessor to create the irrigated and non-irrigated precipitation fraction files. The command is:

../src/mkprf prf.par

and it creates the files **pptfrac.non** and **pptfrac.irr** in each year subdirectory.

3.5 StateFate Processing

In order to apply the recharge decrees against pumping on the ditches that have recharge decrees, it is necessary to compare the amount of consumption permitted under the recharge decrees to the actual consumption as represented in StateCU.

During Phase 6, Davis Engineering provided a spreadsheet which shows the total consumable water under the recharge decrees for each canal. The annual total consumable water for each canal is created by extracting the appropriate columns from the spreadsheet and summarized in the file **subdistricts/recharge-credits.txt**. **StateFate** summarizes the groundwater consumptive use in the .Xpp file. The ratio of actual use to permitted use is calculated using the **mkrc** program which is run in the **subdistricts** directory using the command:

mkrc ../StateFate/rg2012_FactorSoUMeter.Xpp

where the argument is the Xpp file to use. The file **ratio.dat** is written in the subdistricts file and defines the ratio of actual to permitted groundwater use for ditches with recharge decrees.

This ratio is applied to the Xpp files by running the **mkrcdwb** program in the StateFate directory. The commands:

../src/mkrcdwb rg2012_FactorSoUMeter.Xpp rg2012_NoQ.Xpp rg2012_rc.Xpp
ln -s rg2012_FactorSoUMeter.Xgw rg2012_rc.Xgw

create a new Xpp file named **rg2012_rc.Xpp** by using the **ratio.dat** file in the subdistricts folder and scaling the groundwater pumping and recharge shown in the input file **rg2012_FactorSoUMeter.Xpp** with values adjusted by the ratio. For ditches that do not have a recharge decree, the values from **rg2012_NoQ.Xpp** are used, which are the **StateFate** results assuming that there is no groundwater pumping. The merged output **rg2012_rc.Xpp** is therefore the no pumping for all ditches without a recharge decree and pumping matching what is permitted under the recharge decree for ditches with a recharge decree.

This output file will be used to build the input data files for the impact runs.

3.6 Precipitation Preprocessing

Precipitation data for the model is derived from the PRISM data. The PRISM precipitation data consist of gridded data for the entire US by month. The PRISM gridded data are resampled to the RGDSS grid, saved to text files, and used by StatePP.

The resampling is done using a PM-DIAMOND script. This script is executed in the **ppt** directory using the command:

diamond im out 0

The program saves a text file named **yyyymm.ppt** in the data subdirectory, where yyyy and mm refers to the year and month. The file contains the total precipitation for the month in that model cell in inches.

3.7 StatePP Preprocessing

StatePP is the pre-processing program used to create the MODFLOW input files from the GIS and other pre-processing steps. For the RGDSS, **StatePP** consists of two executables, **statepp** and **mket.**

StatePP is discussed in more detail in the memorandum named RGDSS_P6_MOD_StatePP.pdf Here it suffices to say that the **StatePP** input files are created using the **mkpar** script.

All processing using **StatePP** occurs in the **StatePP** directory. In order to streamline the update process, the **mkpar** script is used to create the parameter files used by **statepp** and **mket**. For every update, this script edited to specify a summary title, and version number, and a list of StateFate and M&I Pumping input files to use in the **StatePP** processing. **mkpar** also has the ET versus depth to water and maximum subirrigation rates so that the same curves are written to the **mket** parameter files. **mkpar** creates the following output files:

File	Description
X6P0**xx**.par	**statepp** parameter file for the Steady State, Average Monthly, Initial Period and Monthly simulations
N6P0**xx**.par	**statepp** parameter file for the no pumping verification run
M6P0**xx**-rec.par	**statepp** parameter file for the monthly impact data set
X6P0**xx**.etp	**mket** parameter file for the Steady State, Average Monthly, Initial Period and Monthly simulations

File	Description
N6P0**xx**.etp	**mket** parameter file for the no pumping verification run
M6P0**xx**-rec.etp	**mket** parameter file for the monthly impact data set

The three different parameter files for **statepp** are identical except that they reference different ***.Xpp** files, and name the output files for the particular application. Similarly the three different parameter files for **mket** are identical except for the ***.Xpp** file referenced and the output file names.

StatePP is run using the commands:

statepp X6P033.par >X6P033.lst
statepp N6P033.par >N6P033.lst
statepp M6P033-rc.par >M6P033-rc.lst
mket X6P033.etp
mket N6P033.etp
mket M6P033-rc.etp

The processing takes a few minutes, but these six commands can be executed in parallel.

StatePP creates an input data set for several simulation types. The following is the file root for each of the simulations

Root	Description	Type
S6P0xx	Steady State (1990-1998)	Steady state
I6P0xx	Initial Period (1965-1969)	Steady state
M6P0xx	Monthly transient (1970-2010)	Monthly transient 41 years
A6P0xx	Periodic transient (monthly average 1990-1998)	Monthly transient 1 year
N6P0xx	No pumping	Steady state
C6P0xx	Annual transient[1]	Annual transient 41 years
M6P0xx-rc	Monthly impact (1970-2010)	Monthly transient 41 years

1 The annual transient was used as a calibration technique that captured gross variations in water levels and stream gains when sufficient computation power was not available to do the monthly transient runs. These runs were performed early in Phase 6 but are no longer used.

For each simulation, **StatePP** produces a set of stress files. Multiple stress files are produced for some packages in order to distinguish the quantities in the water budget. For each of the simulation types described above, the following table shows the stress files produced. In addition to these files, a ***.err** file is also produced which lists warnings produced by the program. A summary of stresses is produced to standard output, which in the command set above is redirected to a ***.lst** file.

Extension	Package	Description
mi	Well	Municipal, indistrial, commercial and other pumping
ag	Well	Agricultural well pumping
ppt	Recharge	Recharge from precipitation
rim	Recharge	Recharge from rim recharge
clk	Recharge	Recharge from canal leakage
irr	Recharge	Recharge from irrigation return flow
ets	ET segment	Native evapotranspiration
sub1	ET segment	Meadow subirrigation
sub2	ET segment	Alfalfa subirrigation
sub3	ET segment	Other subirrigation

Separating the stresses by source is instructive for analyzing the Model results. However, during calibration runs performed by PEST, these distinctions are not important and adds to the computational effort involved. The **mkcal** program combines the ***.mi** and ***.ag** files into a single ***.wel** file and the ***.ppt**, ***.rim**, ***.clk** and ***.irr** files into a single ***.rch** file. The **mkcal** program is run using the commands:

mkcal I6P033
mkcal M6P033

The Response Area impact run data sets are created using the **mksub** command. The **mksub** command is a post-processing step that uses the MODFLOW input files from the monthly transient (M6P0xx) and the monthly impact (M6P0xx-rc) files to create the monthly impact data set for each Response Area. This approach is required because the ditch service areas cross Response Area boundaries. The **mksub** command is run using the command:

mksub M6P033 M6P033-rc a:1 b:2 c:3 d:4 e:5 f:6 g:7 h:8 i:9

The **mksub** command uses the files **subdistricts/dist.1** and **subdistricts/dist.2** files to define the spatial extent of each Response Area in the unconfined and confined aquifers, respectively. The letters a, b, c, etc. are used to identify the Response Areas, while the numerical values are used to define the Response Areas in the corresponding flag files.

The **mksub** program produces a new set of stress files by appending the Response Area letter to each of the files. The **mksub** program does not produce new ***.ppt**, ***.clk** or ***.ets** files because it is assumed that precipitation recharge, canal leakage and native ET potential do not change in the impact runs.

Further processing in the **StatePP** directory is done to generate input files for the Response Function runs. A discussion of that processing is separately provided in the RGDSS Phase 6 memorandum named RGDSS_P6_MOD_StatePP.pdf.

3.8 Building the Stream Package

Building the stream package involves many steps. This process is described in detail in a separate RGDSS Phase 6 memorandum named RGDSS_P6_MOD_Stream.pdf. In this document we will simply cover the process at a macro level.

The stream package is built in the **str** directory. The steps required to build the stream package are explicitly defined in a file named **Makefile**. This file defines the dependency tree of the files required to build the stream package, and the commands that must be executed in order to build new versions of the files. The Unix **make** utility compares the time stamps on the various files. When file **a** is shown as depending on file **b**, but the time stamp on file **b** is newer than the time stamp of file **a**, the appropriate command is executed to create a new version of file **a**. The **make** utility builds a dependency tree such that if any data file is changed, it will run the commands needed to recreate the new stream package. The advantage of using the **make** utility is that the entire suite of commands does not have to be run if only one data file is changed. Instead, only those commands in the effected sub-tree are executed. The **make** utility can also be safely executed to check if anything has changed. If it determines that none of the data files have been modified, the **make** utility will indicate that the stream package is already up to date.

Typing **make** will build a stream package for all the steady state and transient simulations used. To create a new version of the stream package, it is necessary to create four new files named X6Pxxx.bld, X6Pxxx.qpr, X6Pxxx.par and X6Pxxx.cst, where xxx is the new version number of the stream package. There is also a need to edit the **Makefile** and set the VER variable to match the new version number and the DIR variable to indicate the directory in which the new version of the model would be run. For example, at the end of Phase 6 these variables are set to VER=6P064 and DIR=x6A00P98.

The ***.bld** file names the streams used to build the stream package. This file is used by the **build** program to build a three dimensional GIS coverage of the stream network including attributes such as stream conductance, stage-discharge rating curve, and similar information.

The ***.qpr** file is used by the **mkq** program to generate time series that represent inflows into and diversions from the stream network. The **mkq** program creates summaries appropriate for the types of simulations of monthly, annual or average flows for the specific locations needed, based on the instructions in the ***.qpr** file. For example, the **mkq** estimates the flow split between the north and south branch of the Rio Grande based on diversions from the north branch.

The stream package is then built by the **modex** program using instructions provided in the ***.par** file. The **modex** program uses the GIS coverage created by **build** and the flow time series created by **mkq** to build the MODFLOW stream package files.

The **modex** program also generates a ***.loc** file that lists the segment and reach in the stream package for all stream gages and the top and bottom of every stream segment in the stream package. The **mkhyd** program uses this information to create a HYDMOD file that is used to summarize the inflow and outflow for every stream segment. This file is used by the **mksum** program to calculate the stream-aquifer interactions.

The ***.cst** file is used by the **mkstr** program to adjust the stream bed conductance as part of the calibration. This file is further described in section Building the PEST Data Files.

3.9 Building the Boundaries

The boundary flows are computed in the **bnd** directory. This directory contains three MODFLOW packages that control boundary fluxes in the model. Although these packages could be given different version numbers, the practice has been to keep the version numbers of all three the same to simplify keeping track of the current boundary flows.

The Flow and Head Boundary (FHB) package used to specify the fixed fluxes into the model which represents the subsurface inflows into the domain along the western boundary of the domain (also called the San Juan underflow), the eastern boundary of the domain south of La Veta Pass (also called the Culebra underflow), and the unsaturated flow across the state line in layers 1, 2 and 3.

The HFB package is generated using the **mkbnd** program. The **mkbnd** program is run using the command:

mkbnd X6P018

where X6P018 represents the version of the boundary. Note the prefix X indicating that these boundary flows will be used in all versions of the model. The **mkbnd** program reads two input files**, X6P018.v**ol an**d X6P018.b**nd and writes the output file **X6P016.fhb**.

The ***.vol** file contains the annual volume in acre feet for each boundary flow. Positive values indicate an inflow, while negative values indicate an outflow. The volumes specified in this file consist of a symbolic name followed by the amount.

The ***.bnd** file specifies the model cells where the volume should be applied. Each line in the ***.bnd** file are of the format:

k i j weight name

where k, i and j represent the model cell, weight is a relative weight for the cell and the name represents the symbolic name used in the ***.vol** file. The **mkbnd** program distributes the flow to individual model cells specified in the ***.bnd** file such that the total flow over all the cells with a given name matches the volume specified in the ***.vol** file, but the relative amount in each cell is proportional to the weight assigned to the cell. Therefore if all cells are given a weight of 1 in the ***.bnd** file, the flow is evenly distributed to all the named cells.

The boundary flows were estimated by HRS by area and layer. These values were represented in the ***.vol** file. The cells in the ***.bnd** file were selected based on the current active cells in the model. Note that these are specified fluxes and therefore these fluxes do not change as a function of the simulated head in the groundwater model.

The General Head Boundary (GHB) package used to represent saturated flow across the state line. These flows occur in Layer 4 and are a function of the calculated heads in the model. The GHB package is built in PM-DIAMOND using two input files. The ***.slf** file is a flag file indicating cells along the southern boundary of the model where the boundary flow will occur. Any digit from 1-9 can be used to indicate a cell where the boundary flows occurs, while 0 is used for the rest of the domain.

The ***.href** file specifies an easting in model coordinates and an elevation. The reference heads in the GHB package is calculated by fitting a Hermite spline through these elevations to obtain cell by cell values.

The GHB conductance is calculated using a hydraulic conductivity of 44 ft/day, the thickness of model layer 4, the cell width (2640 feet) and a reference distance of 5 miles.

The ***.ghb** package is built by running the command:

diamond in out 1 -sGHB

The model version should be set in the **in** file. This command will create a graphic showing a cross section of the model at the state line and the reference heads.

The Horizontal Flow Barrier (HFB) package used to specify the horizontal barrier to flow represented by the Mesita Fault. The location of the Mesita fault as mapped by HRS based on the Tweto work is hard coded into the **mkhfb** program. The program is run using command:

mkhfb X6P018

The horizontal conductance is set to 10^{-4} for the cells along the Mesita fault. Note that during Phase 6, the Mesita fault extends about 2.5 miles further north in layers 2, 3 and 4 than in layer 1.

3.10 Building the PEST Data Files

PEST is used to estimate the aquifer parameters. This process involves creating the appropriate pest control, instruction and template files and running PEST.

To do a PEST optimization, an initial set of model parameters are needed. This is typically the best parameters from the previous optimization. So, for example, for model version 6P96, you would copy the parameters from 6P95 called **X6P095fin.par** to a file called **X6P096ini.par** in the directory **x6A00P96**. If desired, these ***.par** files contain the best parameters which can be used as the initial parameters, or the user may edit this file to alter the values. However, the better approach is to explicitly modify the values using the procedure described below.

The PEST files are created by a program called **mkpestX**X where XX is the model run. So, for example, the program to create run 6P96 is called **mkpest96**. The **mkpestXX** program is created by copying the previous **mkpest** program and simply altering it to reflect changes unique to this model run. While it would be possible to create a single **mkpest** program and control it via data files, the modeling process generally does not lend itself to a sufficiently structured process due to the complex nature of the calibration process. In the rest of this section, this program will be generically called **mkpest**.

The important variables to set in the **mkpest** program are the model version number, the version of the geometry and stream packages, and the parameters to be set.

The **mkpest** program reads Kh data from pump tests from the file **k.dat**. This file contains the location and layer as well as the estimated Kh value for each location. These values are set as fixed parameters for calculating Kh.

The location for the K pilot points are read from **prop.aux**. Each location specified in **prop.aux** becomes a Kh pilot in all the layers unless there is an observation in **k.dat** within 1 mile from that point in the same layer. The locations are specified as an (x,y) location in model coordinates followed by the pilot point name. If the pilot point name is repeated, all those pilot points will have the same value.

The **prop.rng** file contains the plausible range for each parameter. The file consists of the Response Area name, parameter name, an initial value, and the maximum and minimum plausible values. This plausible range is used to set the minimum and maximum values of the parameters in the PEST control file.

The **%adjust** structure in the **mkpest** program is the recommended procedure to alter parameter values from their values in the ***ini.par** file. Specifying a new value here documents how values are changed from one optimization to the next.

When the initial value is at the edge of the range, PEST often has difficulty with finding an optimal value for that parameter. The **%edge** structure sets a tolerance by which the parameter is changed if it is at the edge of the range. For example, Kh1 => 0.02 specifies that if a Kh value is at the lower (or upper) edge of the range, the initial value will be placed 2% from the lower (or upper) end of the plausible range.

The **mkpest** program creates a template file matching the model version called, for example, **X6P096k.tpl**. This template file will be used to create a file named **prop.dat.** The **prop.dat** will then be used by **modex** to calculate the BCF package arrays.

The *.cst file from the stream package is used to set the streambed conductance multipliers for each of the streams. The format of the ***.cst** file is the name of the stream

reach, the PEST parameter name, the type and the range of the multiplier. If the type is fixed, the parameter will not be optimized. If the type is log, the parameter will be optimized within the specified range.

The **mkpest** program creates a template file **X6P096s.tpl** which will be used to create a file **str.par** which is then used by the **mkstr** program to update the streambed conductance values for the stream package.

Head Observations

Head observations for PEST are read from a file called **slvp-tr.res**. As part of the monthly update of the SLVP Water Level Database that Principia maintains for the Rio Grande Water Conservation District, a set of water level observations are created that is stored in this file. This file contains all water level observations from the monitor well network, except that in the case of the wells with continuous recorders that store daily values, the average value for each month is stored instead of the daily values.

Stream Gains and Losses

Stream gain observations are read from the file **baseflow.dbf**, which is created by **mkbf**. This file contains monthly and seasonal observations of baseflow gain estimates. The file **weight.txt** controls how baseflow gains are used. For each stream reach, this file contains the name of the stream reach, the type and weight. The type may be Monthly, Average, Seasonal, Unmet, NoNovMar or NoAprOct. A type of Monthly, Average or Seasonal is used to indicate the time period over which the gain-loss estimates should be compared. The Unmet type sets unmet diversions as an annual target. NoAprOct and NoNovMar sets a target indicating that there should be no flow past the last diversion on the stream for the April-October or November-March period.

The weight in the **weight.txt** file is used to set the relative weight of the observations. The line may also contain a list of weights for each of the years in the simulation. If annual weights are specified, the weight for that year's observation will be determined by multiplying the annual and global weights. When a particular observation for a year has a weight of zero, it is omitted.

Flooded and Dry Cells

The **mkpest** program also generates targets called **flood** and **dry**. These targets assert that there should be no cells that are dry (heads below the bottom) or flooded (heads above ground surface).

The **%weight** structure in **mkpest** is used to assign weights to individual targets. This can be used to increase or decrease the weights of individual observations. The **%mul** structure performs a similar function, but instead adjusts the weight by multiplying it by the set value.

Since most of the observations are heads, the **mkpest** program applies an algorithm to balance the weights between the groups of head observations. Most observations are in layer 1, so these are weighted less for each observation. Observations in Layer 5 are the fewest, and are therefore weighted more. The RG series wells are weighted more as they are of high quality. Water levels derived from well permits are weighted least as they are of the lowest quality.

The **mkpest** program will then create the pest control file by writing the parameters and observations to the PEST control file **X6P098.pst**. The observations are also written to the PEST observation file **X6P098.pof** which will be used by the **hyd2res** program to extract modeled results from the HYDMOD binary output files and save it in a text file that can be read by PEST. The **mkpest** program will also write an instruction file **X6P098.ins** to read the results. Finally, the **mkpest** program will also create the **X6P098.sync** file which is used by **beorun** to copy the input files to all the cluster nodes.

The **mkpest** program is run in the **prop** directory. At the same time, the **mkbcf** program should be run to create the **modex** input file **X6P098.par**, the ***.bcf** files for MODFLOW and the **X6P098.dpr** file for **mkdrn**. These files are named for the model run. So, for example, the 6P96 input files are created using the command:

mkbcf X6P096

The **mkbcf** program must be run in the **prop** directory.

3.11 Running the Model

A PEST optimization consists of three distinct operations.

The first operation is to calculate the Jacobian matrix. The number of model runs required to calculate the Jacobian is equal to the number of adjustable parameters in the model. During Phase 6, this required 1251 model simulations. Each model simulation takes around 60 to 90 minutes. Using BeoPEST on a computer cluster allows these simulations to be done in parallel, which greatly reduces the time needed to perform these simulations.

Once the Jacobian is calculated, the second operation is to optimize using super-parameters. The super-parameters are calculated using the largest eigenvalues of the Jacobian. During Phase 6, 240 super-parameters were used. These super-parameters are then optimized to minimize the residual. Each optimization iteration requires 241 or 481 model simulations, depending on whether forward differences or central differences are used. Between optimization iterations, a parallel lambda search is done.

Once an optimal set of parameters are found, the third PEST operation is to run PEST one last time with the base parameters. This run provides the final residuals and base parameter values for the model.

Once the optimization is complete, the entire suite of steady state and transient model runs are performed using the separate stress files, followed by a suite of model runs to do the impact evaluation.

The model runs are controlled by several scripts. These scripts are copied from one model version to the next, and modified to adjust to the particular run.

The files to copy are named **mkrun**, **mksubnam**, **run**, **runsam**, **runsub**, **runsvd**, **svd.in**, **Mkbgt** and **RUN**.

After copying the scripts, edit them. In each script alter the THIS variable to set the current model run, LAST to set the previous model run, CU to set the StateCU data version, STR to set the stream data versions and BND to set the boundary data version. In the Mkbgt script, also set the date of the simulation.

The **run** script is invoked by PEST to perform the model run. During Phase 6, the monthly transient model is used. PEST will create the **prop.dat** and **str.par** files. The **run** script then runs **modex** to create files named *.kh?, *.vc??, *.sy and *.sc? which have the cell by cell values of hydraulic conductivity, vertical conductance, specific yield and storage coefficient. Next it runs the **src/mkdrn** script which creates the Drain Return (DRT) input file from the **prop/*.dpr** file and the *.kh? files. Next it runs the **src/mkstr** program to adjust the stream bed conductance from the **str.par** file and the stream package for both the Initial Period and Monthly Transient runs. It then runs MODFLOW for the Initial Period, uses the predicted head from the initial period as the starting heads for the transient run, and runs the transient run. Finally it runs the **hyd2res** program to extract the heads and flows from the ***.sfi** file using instructions provided in the ***.pof** file, and creates the **target.dat** file which contains the targets in a text file. During Phase 6, the ***.pof** file also causes the **sumflood** program to be run to calculate how often cells are flooded and the **sumdry** program to calculate how often the heads drop below the bottom of the aquifer. The **mksum** program is also run from the **hyd2res** program with the **-u** flag to summarize the unmet diversions and flows past the last diversion.

The **runsvd** script is identical to the **run** script except that it invokes **parcalc** and **picalc** to map the super-parameters estimated during the SVD optimization to base parameters.

The **mkrun** script is used to set up the run. It will create the necessary ***.NAM** files used by MODFLOW to access the various packages using the **src/mkrun** program, create the starting head files using the last best simulation using the **src/mkshead** program, build a mapping using the stream package from MODFLOW stream segments to HYDMOD names**,** run **mkpest** to create the PEST control files and check that the PEST control files are properly constructed.

Next, run **mksubnam** to create the ***.NAM** files for the Response Area impact runs.

The **RUN** script performs the rest of the simulations. Depending on the run, this can take from a day to a week or more on a computer cluster. The **RUN** command assumes that the work is being performed on a cluster. It invokes the **beorun** command to run BeoPEST on as many cores as are specified in the **beohost** file. First **beorun** is invoked with the **-r** parameter which uses the UNIX **rsync** command to copy the model input files to all the cluster nodes. Next **beorun** is invoked on the PEST control file written by **mkpest** which will compute the Jacobian. This takes about a day on the computer cluster.

Once the Jacobian is computed, the **svdaprep** command is run using the **svd.in** commands to set up the SVD PEST run. To customize the resulting PEST control file, the UNIX **sed** command is used to modify some of the parameters, such as the number of optimization cycles to perform. Then **beorun** is again used to first use **rsync** to copy the Jacobian and related files to each of the nodes on the cluster, and then perform the SVD PEST run. This can take anywhere from a day to a week or more on the computer cluster.

The final PEST step is to use the **parrep** command to map the optimized super-parameters to base parameters and then do a single PEST run in the local directory in order to get the PEST output files for this best model run.

Once the optimization is complete, the **runsam** script is invoked from **RUN.** This script runs **src/mktarstat** which reads the individual head observations from **target.dat** and creates summaray statistics for each well in **target.dbf.** Next it runs **src/mklim** which creates a summary table showing the pilot points used and uses different symbols to show whether the optimized value is within the specified range, or has reached the upper or lower bound of the range. Then it uses the **src/mkstr** program and the **str.par** file to create the stream packages for the entire suite of model runs.

Next the **runsam** script invokes the **src/runam** scrip to perform the suite of model runs. Note that during the optimization, the monthly model is run with a single recharge and well file. In the **src/runsam** script, the No Pumping, Steady State, Initial Period and Annual Average runs are also executed. The **src/runam** script is in turn invoked to perform the Average Monthly run. This simulation runs the model using average monthly conditions for a year, uses the year end heads as starting heads for the next year, and repeats this process 20 times. To speed convergence, the starting heads for the next yearly cycle is taken as 5% of the previous starting head and 95% of the new starting head. Once these simulations are completed, the **src/runsam** script invokes the **mkbgt** program to calculate the detailed water budget for all of the simulations. Next the **mksum** script is invoked to calculate the detailed stream budgets for all the simulations.

Lastly, the **runsam** script invokes the **src/mksubsum, src/mkgl** and **src/mkcond** programs. These programs create summary results of subirrigation, stream gains and losses and stream conductances that are used in graphical displays of the results.

After the suite of model runs are completed and post processed, the Response Area impact runs are performed using the **runsub** script which is invoked from **RUN**. The **src/mksubstr** program adds groundwater return flow discharges that occur directly to streams in to the stream package for the various Response Areas. Since the Response Area runs are independent of each other, the MODFLOW runs can be performed in parallel on the cluster. Once the Response Area impact runs are complete, the results are post-processed and summarized using the **Mkbgt** script.

The **Mkbgt** script runs **mkbgt** and **mksum** in difference mode. In this mode, the programs calculate how the aquifer stresses are different between the historical or reference run, and the Response Area impact run. These differences are interpreted as the impacts caused by the activities within the Response Area . After **mksum** is run, **mkimp**

is used to summarize the stream impacts in the ***-imp.dbf** file which are the impacts from the Response Area. The locations of the impacts are summarized using the **src/mkgl** program.

Once all the Response Area impacts are calculated, the **Mkbgt** program invokes **src/mkdet** program to summarize where the ET and subirrigation salvage occurs. Finally, the **Mkbgt** script calls the **src/mktab** program to summarize the impacts by Response Area and stream reach for the last 10 years.

3.12 Performing the Response Function Runs

Additional model runs are completed to create the outputs needed to calculate response functions.

The response function simulations involve using paired simulation where the initial year corresponds to a historical year which is then followed by 19 years of average conditions. The simulation in a pair of runs differ only with respect to the first year of the simulation, where in one simulation the historical conditions are simulated, while in the other simulation the well pumping and recharge is altered for a Response Area to evaluate the impact condition.

The input files for the response function run is created in the **StatePP** directory by running the **mkurfdat** program. The model version, StateCU version and stream version is set at the top of the script. The years to consider (1988-2010 in Phase 6) and the Response Area are also specified. The program will then create files named **H6PXXXzYY.*** where XXX is the StateCU data version, z is a letter representing the Response Area and YY is the last two digits of the year. The symbol = is used to represent no Response Area, in other words the reference condition. The **mkurfdat** will create packages for well pumping, irrigation returns and subirrigation for each Response Area. The precipitation recharge, rim inflow, native ET and stream packages are the same in the reference and impact runs, so only a single package is created for each response year.

The **mkurfnam** script is used to create the MODFLOW runs that must be performed. This script is run in the same directory as the model runs. The script is similar to the **mksubnam** script, except that it does runs for each of the response years.

During Phase 6, a total of 253 response runs were required. These runs are independent of each other and can be run in parallel on the cluster.

The **mkurfbgt** script was used to post-process the response function results. The differencing feature of the **mkbgt** and **mksum** programs were used to isolate the impacts. The **mkimp** program was then used to extract only the impacts to surface water from these results. Since this post-processing step is quite computationally intensive, the **mkurfbgt** script performs the calculations in parallel on the cluster.

Finally the **mkurftab** script is used to export the response function values in a format suitable for importing into Excel for the purpose of calibrating the response functions. This script calculates the magnitude of the responses in the impact runs, extracts those where the impacts exceed the criteria and saves the results to text files containing the individual responses, the Response Area responses to which to calibrate and summary tables of the responses for each Response Area.

4.0 Comments and concerns

Following are comments and concerns associate with this process.

At this time, the scripts are set for a Linux system. Most of the commands will work on a Windows system, but may need modifications to deal with operating system specific issues such as directory separators.

The **RUN** script for performing the bulk of the optimization is specific to the particular cluster on which the optimizations have been performed. This could be readily adapted to other clusters. Performing this model setup, calibration, simulations, and summaries other than using a cluster is not recommended as it would take many months if not years on a single desktop computer. The cluster used in Phase 6 has 128 cores, and typical speedup achieved was around 100 times. Attempting an optimization that would take 3 days on this cluster on a single computer would extend the run time to a year.

Many of the adjustments from one model version to the next require flexibility and do not follow a set pattern. This necessitated an approach of copying scripts and modifying them to suit the adjustments required. These scripts were retained as a record of all the adjustments made. Where it was observed that the scripts start to follow a pattern, a script was created to write the scripts instead of copying them. However, as the process continues, this documentation will require updating to adapt to evolving procedures.

5.0 References

Appendix N: Directory and File Naming Convention, RGDSS Memorandum Task 41.1, Ray Bennett, Brian Ahrens and Willem Schreuder, March 3, 2004.

RGDSS Phase 6 Memorandum, Building the Stream Package, RGDSS_P6_MOD_Stream.pdf.

RGDSS Phase 6 Memorandum, Groundwater GIS Processing, RGDSS_P6_GIS_Processing.pdf.

RGDSS Phase 6 Memorandum, Model Enhancements, RGDSS_P6_MOD_Enhanc.pdf.

Appendix A: Directory Structure

Directory	Description
baseflow	Directory where baseflow calculations are performed
bnd	Directory where state line and boundary inflow packages are built
Diversions	Output from TSTool scripts to summarize diversion amounts.
GIS	Output from StateDGI that maps ditches, wells, rim inflow areas, native ET, etc. to model cells. Subdirectories **1936**, **1998**, **2002**, **2005**, **2009** and **2010** contain snapshots corresponding to those particular years. Subdirectory **General** contains shape files with the locations of diversions.
hyd	Directory where HYDMOD package is built
ManualEntry	Data files used by the aggregate structure too **agg**
MIPumping	Output from TSTool scripts that summarize pumping and return flows from wells not simulated within StateCU such as municipal, industrial or commercial pumping, and pumping by the Closed Basin Project. Also contains description of pumping returns to streams.
misc	Directory for solver control files
ppt	Directory where PRISM precipitation data is stored
prop	Directory where aquifer geometry and property data are built. This directory is also where the PEST data files are built.
RimRecharge	Output from TSTool scripts that summarize recharge from small stream inflows around the rim of the valley.
src	Directory for various processing scripts
StateFate	Output from the StateFate program
StatePP	Directory where the StatePP program is run.
str	Directory where stream package is built.
StreamInflow	Output from TSTool scripts that summarize surface inflow for streams explicitly modeled.
StreamGW	Output from TSTool scripts that summarize stream flow quantities required by the stream package such as outflow from transfer ditches to streams and data files describing diversions to be modeled.
subdistricts	Directory where recharge decree calculations are performed
x6A00P**xx**	Directory for simulation 6P**xx**, where **xx** is a two digit number.