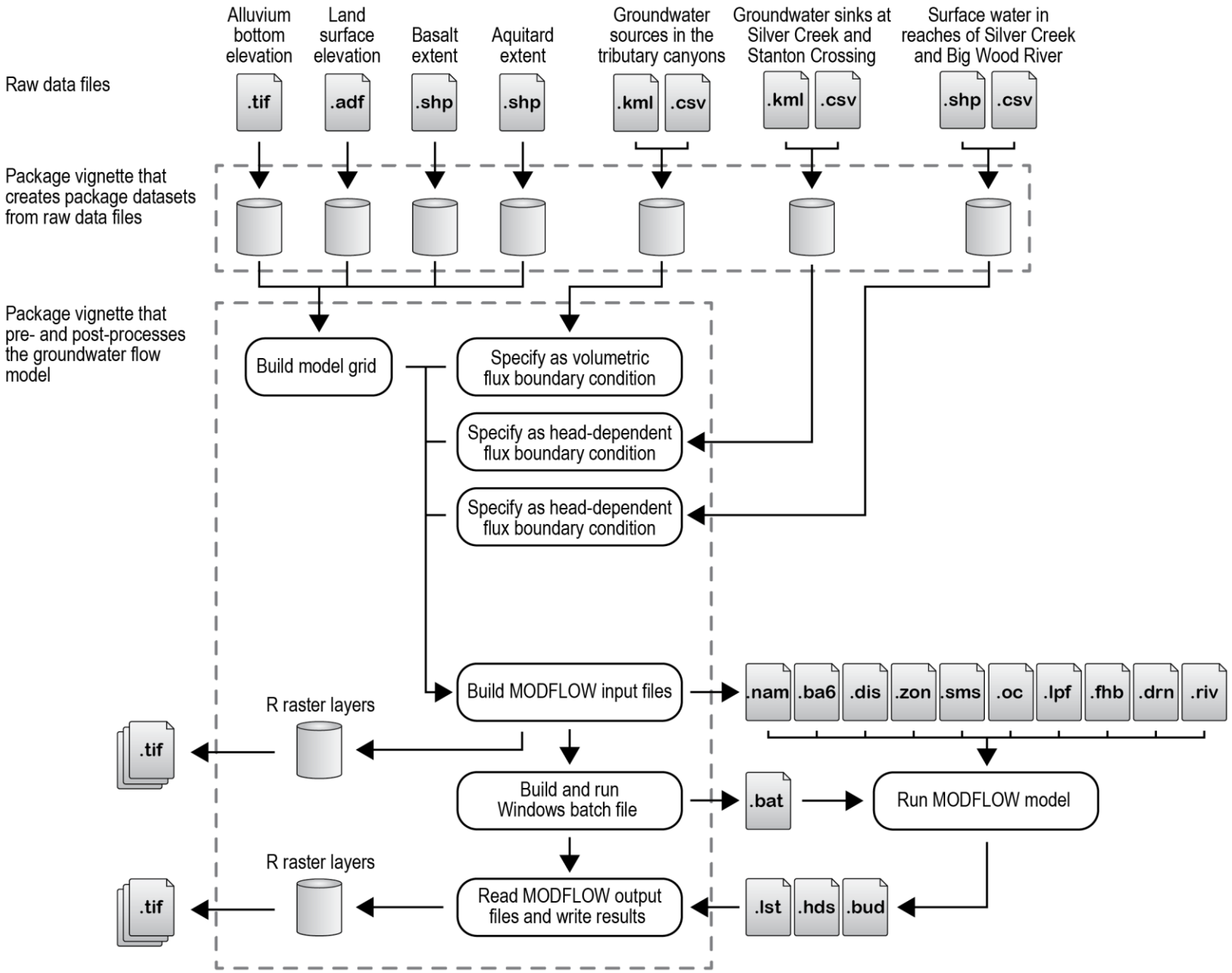
Wood River Valley
Revision to the Model-Building Vignette

Jason C. Fisher

U.S. Geological Survey
Idaho Department of Water Resources

PROVISIONAL: FOR INFORMATIONAL PURPOSES ONLY

These slides were presented at the Wood River Valley Modeling Technical Advisory Committee meeting Thursday, 4/3/2014, 10am-3pm at the Community Campus, Bullion Room, in Hailey. Taken outside the context of the original presentation, these slides may not provide a complete or accurate representation of the speaker’s intent.


Alluvium bottom elevation
Land surface elevation
Basalt extent
Aquitard extent
Groundwater sources in the tributary canyons
Groundwater sinks at Silver Creek and Stanton Crossing
Surface water in reaches of Silver Creek and Big Wood River
Raw data files
.tif
.adf
.shp
.shp
.kml
.csv
.kml
.csv
.shp
.csv
Package vignette that creates package datasets from raw data files
Package vignette that pre- and post-processes the groundwater flow model
Build model grid
Specify as volumetric flux boundary condition
Specify as head-dependent flux boundary condition
Specify as head-dependent flux boundary condition
Build MODFLOW input files
.nam
.ba6
.dis
.zon
.sms
.oc
.lpf
.fhb
.drn
.riv
R raster layers
.tif
Build and run Windows batch file
.bat
Run MODFLOW model
R raster layers
.tif
Read MODFLOW output files and write results
.lst
.hds
.bud

LaTeX

R

Create Package Data Sets from Raw Data Files

By Jason C. Fisher

Contents

Introduction 2

Software, I/O Paths, and Coordinate Reference System 2

Points 2
- Cities and Towns . . . 2
- Map Labels . . . 3

Lines 3
- Reaches of the Big Wood River and Silver Creek . . . 3
- Rivers and Streams . . . 3

Polygons 3
- Lakes and Reservoirs . . . 3
- Extent of Aquifer System . . . 8
- Extent of Aquitard . . . 8
- Extent of Basalt . . . 8
- Groundwater Sources . . . 8
- Groundwater Sinks . . . 12

Grid 12
- Pre-Quaternary Bedrock Surface and Top of Quaternary Basalt . . . 12
- Topography of Land Surface . . . 12

Reproducibility 16

Introduction

This *package vignette* explains the processing steps for creating R data sets in the **wrv** package. All data sets are projected into a common coordinate reference system (CRS) and converted to a consistent set of units; that is, length and time dimensions in units of meters (m) and days (d), respectively.

Software, I/O Paths, and Coordinate Reference System

Extend the capabilities of R by installing the following user-created packages:

```
library(rgdal)   # bindings for the geospatial data abstraction library
library(raster)  # gridded spatial data toolkit
library(RCurl)   # general network client interface
library(png)     # read and write png images
```

Package data sets are created from raw data files located on GitHub. Specify the top-level uniform resource locator (URL) where raw data files are located:

```
url <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/"
```

The raw data file of land-surface elevations was deemed too large in file size (about 500 MB) to be placed in the **wrv**-package repository. The land-surface elevation data is part of the National Elevation Dataset (NED) 1/3-arc-second raster, and was downloaded on September 22, 2013 in a Esri ArcGRID format using the National Map Viewer. The west, east, south, and north bounding coordinates for this data set are -115, -114, 43, and 44 decimal degrees, respectively. Specify the path name of the downloaded raster elevation data directory:

```
file.ned <- "C:/n44w115/grdn44w115_13"
```

Specify directory path names for output files and create these directories if they do not already exist.

```
dir.data <- file.path(getwd(), "data")
dir.extdata <- file.path(getwd(), "inst", "extdata")
dir.create(dir.data, showWarnings = FALSE)
dir.create(dir.extdata, showWarnings = FALSE, recursive = TRUE)
```

The common CRS applied to all spatial data sets is the Idaho Transverse Mercator projection (IDTM). PROJ.4 projection arguments are used to specify a CRS in R. The common CRS that all raw data files are converted into is specified as:

```
crs <- CRS(paste("+proj=tmerc +lat_0=42 +lon_0=-114 +k=0.9996 +x_0=2500000 +y_0=1200000",
                 "+datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0"))
```

Points

Cities and Towns

Cities and towns in the WRV and surrounding areas (fig. 1).

```
files <- wrv::DownloadFile(paste0(url, "cities.zip"))
layer <- sub(".shp$", "", basename(files[grep("*.shp$", files)]))
cities <- readOGR(dsn = tempdir(), layer = layer, verbose = FALSE)
cities <- spTransform(cities, crs)
```

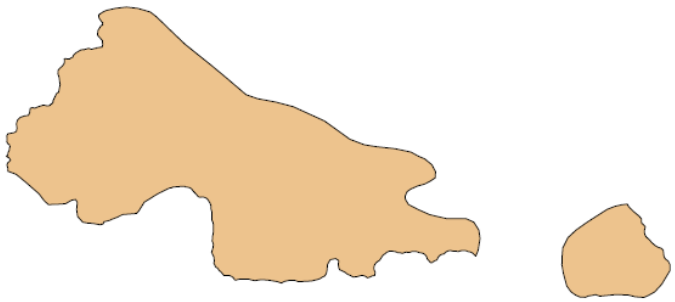


Figure 7: Estimated extent of the confining unit separating the unconfined aquifer from the underlying confined aquifer in the Wood River Valley.

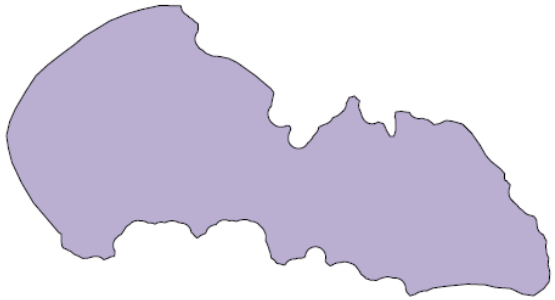


Figure 8: Estimated extent of the basalt underlying the alluvial Wood River Valley aquifer system.


```
source.locations <- spTransform(source.locations, crs)
unlink(file)
file <- wrv::DownloadFile(paste0(url, "source.locations.csv"))
h <- read.csv(file, row.names = 1)
source.locations@data <- cbind(source.locations@data, h)
save(source.locations, file = file.path(dir.data, "source.locations.rda"))
unlink(file)
```

Groundwater Sinks

Polygons used to define boundary conditions at sink locations in the model domain (fig. 10). The polygons clip the line segments along the aquifer boundary (see aquifer.extent), and model cells intersecting these clipped-line segments are defined as boundary cells.

```
file <- wrv::DownloadFile(paste0(url, "sink.locations.kml"))
sink.locations <- suppressWarnings(readOGR(file, basename(file), verbose = FALSE))
sink.locations <- spTransform(sink.locations, crs)
unlink(file)
file <- wrv::DownloadFile(paste0(url, "sink.locations.csv"))
h <- read.csv(file, row.names = 1)
sink.locations@data <- cbind(sink.locations@data, h)
save(sink.locations, file = file.path(dir.data, "sink.locations.rda"))
unlink(file)
```

Grid

Pre-Quaternary Bedrock Surface and Top of Quaternary Basalt

The estimated elevation of the pre-Quaternary bedrock surface and top of Quaternary basalt in the the WRV aquifer system (fig. 11).

```
file <- wrv::DownloadFile(paste0(url, "alluvium.bottom.tif"))
alluvium.bottom <- readGDAL(file, band = 1, silent = TRUE)
proj4string(alluvium.bottom) <- crs
names(alluvium.bottom) <- "alluvium.bottom"
save(alluvium.bottom, file = file.path(dir.data, "alluvium.bottom.rda"))
unlink(file)
```

Topography of Land Surface

The topography of the land surface in the WRV and vicinity (fig. 12).

```
r <- raster(readGDAL(file.ned, band = 1, silent = TRUE))
r <- projectRaster(r, raster(alluvium.bottom), method = "bilinear")
r[is.na(raster(alluvium.bottom))] <- NA
land.surface <- as(r, "SpatialGridDataFrame")
names(land.surface) <- "land.surface"
save(land.surface, file = file.path(dir.data, "land.surface.rda"))
```

Hillshading based on the slope and aspect of land-surface elevations (fig. 13).

Processing Instructions for the Groundwater Flow Model of the Wood River Valley, Idaho

By Jason C. Fisher

Contents

Introduction 3

Software 3

Hydrogeologic Framework 3

Pre-Processing 5
- Conceptualization of Model Grid . . . 5
- Hydrogeologic Zones . . . 15
- Groundwater Flow in the Tributary Canyons and Upper Big Wood River Valley . . . 18
- Groundwater Flow Beneath Silver Creek and Stanton Crossing . . . 20
- Stream-Aquifer Flow Exchange in the Big Wood River and Silver Creek . . . 22
- Model Grid . . . 23
- Initial Head Distribution . . . 23
- Export Raster Layers . . . 25

Model Run 25

Post-Processing 26

Reproducibility 27

Figures

1 Land surface topography and extent of the aquifer system . . . 4
2 Thickness of Quaternary sediment in the aquifer system . . . 6
3 Extent of basalt in the aquifer system . . . 7
4 Thickness of the aquifer system . . . 9
5 Extent of aquitard in the aquifer system . . . 10
6 Vertical connectivity among cells . . . 12
7 Location of source cells in the aquifer system . . . 13
8 Schematic cross-section representation of hydrogeologic units and three-layer model grid. . . 14
9 Hydrogeologic zones in model layer 1 . . . 16
10 Hydrogeologic zones in model layer 2 . . . 17
11 Hydrogeologic zones in model layer 3 . . . 19
12 Location of sink cells in model layer 1 . . . 21
13 Stream-reach types in the Big Wood River and Silver Creek . . . 24
14 Saturated and partially-saturated cells in model layer 1 . . . 28

15 Simulated elevation of the water table . . . 29

Tables

1 Volumetric flux in the tributary canyons and upper Big Wood River valley. . . . 18
2 Specified hydraulic head thresholds for sink cell boundary conditions. . . . 20
3 Description of stream reaches in the Big Wood River and Silver Creek. . . . 22
4 Summary description of the model grid attributes. . . . 23
5 Input parameters for model run. . . . 25
6 Volumetric budget for entire model at end of time step 1 stress period 1. . . . 26

Conversion Factors and Datums

Conversion Factors

Multiply	By	To obtain
	Length	
meter (m)	3.28084	foot (ft)
kilometer (km)	0.621371	mile (mi)
meter (m)	1.09361	yard (yd)
	Area	
square meter (m^2)	0.0002471	acre
square kilometer (km^2)	247.104	acre
square meter (m^2)	10.7639	square foot (ft^2)
square kilometer (km^2)	0.386102	square mile (mi^2)
	Volume	
cubic meter (m^3)	0.0008107	acre foot (acre-ft)
cubic meter (m^3)	35.3147	cubic foot (ft^3)
	Volume per unit time	
cubic meter per day (m^3/d)	0.296107	acre-foot per year (acre-ft/yr)

Datums

Vertical coordinate information is referenced to the North American Vertical Datum of 1988 (NAVD 88).

Elevation, as used in this document, refers to distance above vertical datum.

Horizontal coordinate information is referenced to the North American Datum of 1983 (NAD 83).

Maps are based on the Idaho Transverse Mercator projection (IDTM).

Introduction

The **wrv** package (Fisher, 2014) is a pre- and post-processing program for the numerical groundwater flow model of the Wood River Valley (WRV) aquifer system, south-central Idaho. This document (also known as a *package vignette*) explains the steps taken to process the model; its contents should be viewed as provisional until model and report completion in 2015 (Bartolino *et al.*, 2015). It is assumed that the readers of this vignette are familiar with the R-programming language and have read the help pages for functions and data sets in the **wrv** package.

Software

Software items needed to run the processing instructions for the groundwater flow model include R and MODFLOW-USG. R is a language and environment for statistical computing and graphics (R Core Team, 2014). If R (version $\geq$ 3.0) is not already installed on your computer, download and install the latest binary distribution from CRAN. Extend the capabilities of R by installing user-created packages, such as **wrv**. Start an R session and type the following commands in your R Console window:

```
install.packages(c("rgdal", "raster", "igraph", "rgeos", "RCurl", "png", "xtable"))
install.packages("wrv", repos = "http://jfisher-usgs.github.com/R/")
```

MODFLOW-USG is a computer program for simulating three-dimensional, steady-state and transient groundwater flow using a control volume finite-difference formulation (Panday *et al.*, 2013). If MODFLOW-USG (version $\geq$ 1.1) is not already installed on your computer, download and decompress the latest file archive. The file archive contains an executable file for Windows. Users of a Unix-like operating system will need to compile MODFLOW-USG. Specify the full path name to the executable file using the following R command (change path if needed):

```
file.exe <- "C:/WRDAPP/mfusg.1_1/bin/mfusg_x64.exe"  # path specified with forward slashes
```

This package vignette integrates R code in a LaTeX document. The code is run when the vignette is built, and all data analysis output (figures, tables, etc.) is created on the fly and inserted into the final document. Note that embedded code can be extracted from the vignette, which allows for truly reproducible research; see the 'Reproducibility' section for details. Small chunks of stylized code are shown throughout the vignette and are intended to be used interactively. Commands that comprise these *code chunks* are essential for processing the groundwater flow model, and describe approaches to model development and analysis decisions.

To gain access to the contents of the **wrv** package in R, use the following command:

```
library(wrv)
```

To open help pages for functions and data sets in the **wrv** package, type:

```
help(package = "wrv")
```

Hydrogeologic Framework

The WRV aquifer system is composed of a single unconfined aquifer that underlies the entire valley, an underlying confined aquifer that is present only in the southern part of the valley, and a confining unit separating the two aquifers (Bartolino and Adkins, 2012, pg. 3). The land-surface topography and spatial extent of the aquifer system are shown in figure 1. The aquifer system primarily consists of Quaternary deposits that can be divided into three hydrogeologic units: a coarse-grained sand and gravel unit (alluvium unit), a fine-grained silt and clay unit (clay unit), and a basalt unit (Bartolino and Adkins, 2012, pg. 3).

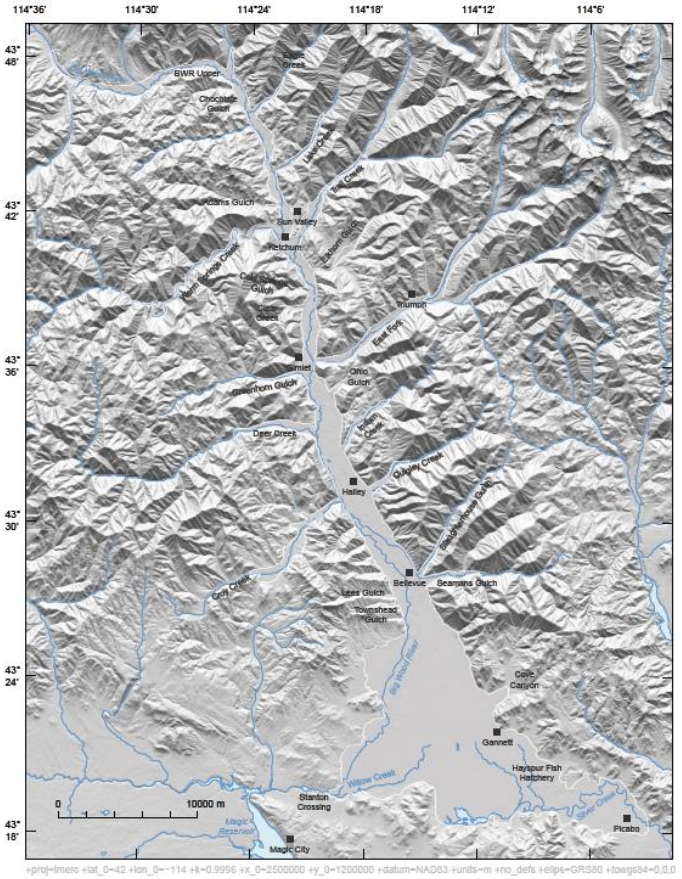


Figure 1: Land surface topography and extent of the aquifer system.

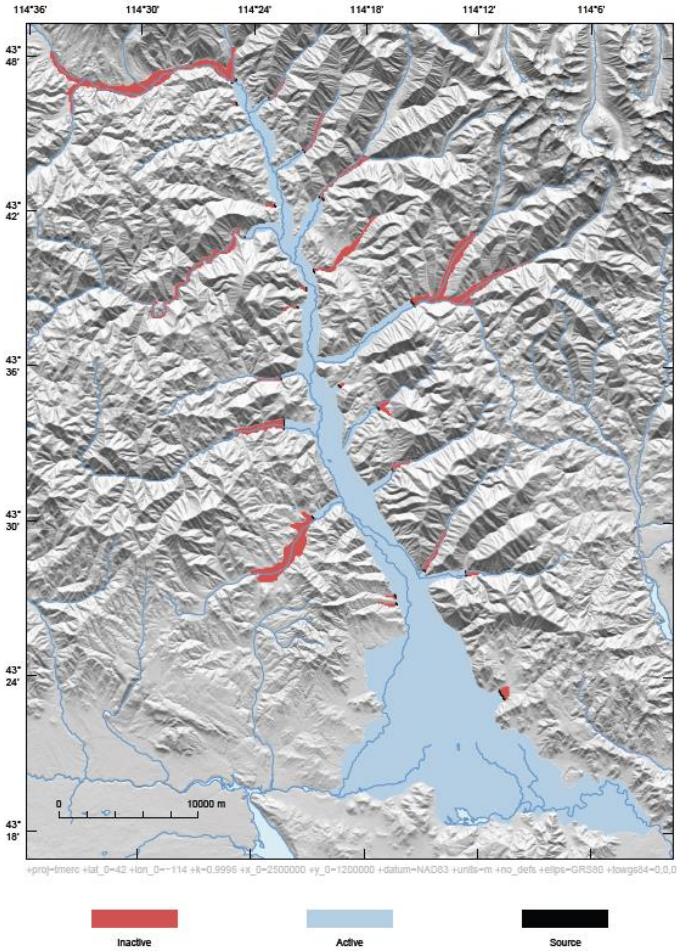


Figure 7: Location of source cells in the aquifer system.

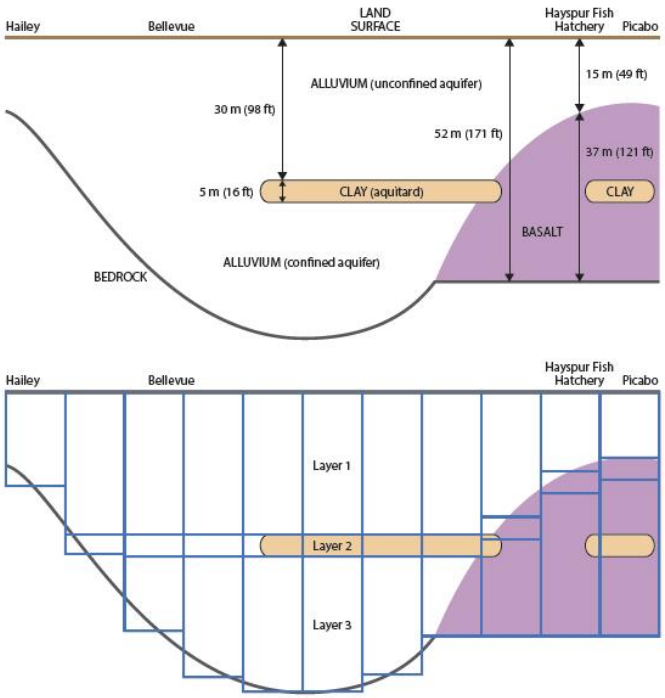


Figure 8: Schematic cross-section representation of hydrogeologic units and three-layer model grid.

```
r <- rs.model[["lay1.bottom"]] - aquitard.thickness
is.below <- rs.data[["bedrock"]] > r
r[is.below] <- rs.data[["bedrock"]][is.below]
r[(rs.model[["lay1.bottom"]] - r) < min.thickness] <- NA  # enforce minimum thickness
r <- ExcludeSmallCellChunks(r)
names(r) <- "lay2.bottom"
rs.model <- stack(rs.model, r)
```

The bottom elevation of model layer 3 is at bedrock.

```
r <- rs.data[["bedrock"]]
r[is.na(rs.model[["lay2.bottom"]])] <- NA
r[(rs.model[["lay2.bottom"]] - r) < min.thickness] <- NA  # enforce minimum thickness
r <- ExcludeSmallCellChunks(r)
```

```
names(r) <- "lay3.bottom"
rs.model <- stack(rs.model, r)
```

Bottom elevations of model layer 1 are adjusted to bedrock where the cell value is above bedrock and its vertically adjacent cell is inactive in model layer 2.

```
r <- rs.model[["lay1.bottom"]]
is.adjusted <- r > rs.data[["bedrock"]] & is.na(rs.model[["lay2.bottom"]])
r[is.adjusted] <- rs.data[["bedrock"]][is.adjusted]
r <- ExcludeSmallCellChunks(r)
rs.model[["lay1.bottom"]] <- r
```

The top elevation of model layer 1 is at land surface.

```
r <- rs.data[["land.surface"]]
r[is.na(rs.model[["lay1.bottom"]])] <- NA
names(r) <- "lay1.top"
rs.model <- stack(rs.model, r)
```

Hydrogeologic Zones

Prior to model calibration, the distribution of hydraulic properties (such as hydraulic conductivity) is based on hydrogeologic zones, groups of model cells with uniform hydraulic properties that compose part or all of a hydrogeologic unit. The model consists of four hydrogeologic zones described as follows:

Zone 1: composed of the alluvium unit under unconfined conditions and located in all three model layers;

Zone 2: composed of the basalt and clay units and located in model layers 2 and 3;

Zone 3: composed of the clay unit and located in model layer 2; and

Zone 4: composed of the alluvium unit under confined conditions and located in model layer 3.

The delineation of hydrogeologic zones in model layer 1 is shown in figure 9.

```
r <- rs.model[["lay1.bottom"]]
r[!is.na(r)] <- 1L  # alluvium unit (unconfined aquifer)
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- "Zone 1"
levels(r) <- rat
names(r) <- "lay1.zones"
rs.model <- stack(rs.model, r)
```

The delineation of hydrogeologic zones in model layer 2 is shown in figure 10.

```
r <- rs.model[["lay2.bottom"]]
r[!is.na(r)] <- 1L  # alluvium unit (unconfined aquifer)
r[!is.na(r) & !is.na(rs.data[["aquitard.extent"]])] <- 3L  # clay unit (aquitard)
r[rs.model[["lay2.bottom"]] < rs.data[["alluvium.bottom"]]] <- 2L  # basalt & clay units
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- c("Zone 1", "Zone 2", "Zone 3")
levels(r) <- rat
names(r) <- "lay2.zones"
rs.model <- stack(rs.model, r)
```

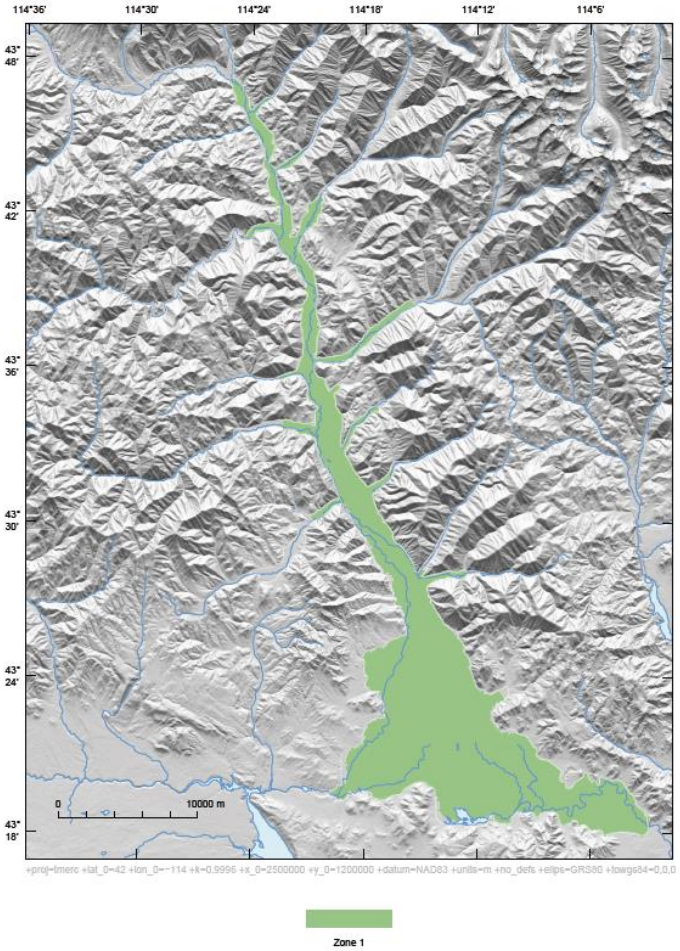



Figure 9: Hydrogeologic zones in model layer 1.

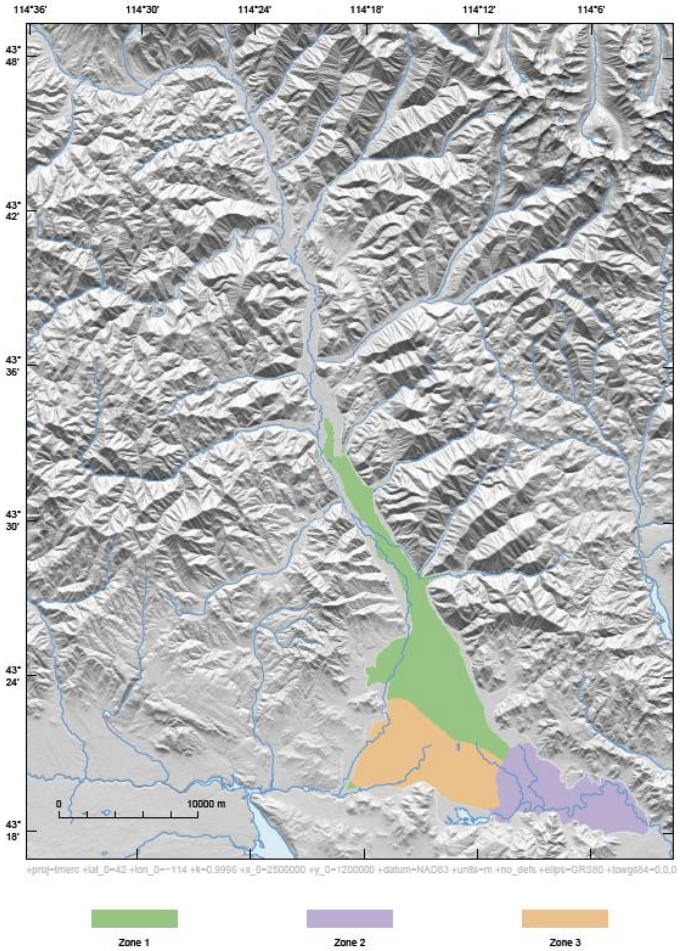


Figure 10: Hydrogeologic zones in model layer 2.

The delineation of hydrogeologic zones in model layer 3 is shown in figure 11.

```
r <- rs.model[["lay3.bottom"]]
r[!is.na(r)] <- 1L  # alluvium unit (unconfined aquifer)
r[!is.na(r) & rs.model[["lay2.zones"]] == 3L] <- 4L  # alluvium unit (confined aquifer)
r[rs.model[["lay3.bottom"]] < rs.data[["alluvium.bottom"]]] <- 2L  # basalt & clay units
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- c("Zone 1", "Zone 2", "Zone 4")
levels(r) <- rat
names(r) <- "lay3.zones"
rs.model <- stack(rs.model, r)
```

Groundwater Flow in the Tributary Canyons and Upper Big Wood River Valley

Groundwater entering the aquifer system (source) through the tributary canyons and upper Big Wood River Valley is simulated using the Flow and Head Boundary Package (Leake and Michael, 1997), a specified flow boundary condition. The estimated volumetric flux for a source area is based on steady-state conditions and given in table 1. For each source area, the volumetric flux is uniformly distributed among model cells where this boundary condition is applied (source cells).

Table 1: Volumetric flux in the tributary canyons and upper Big Wood River valley.

Name	Flow (m^3/d)	Flow (acre-ft/yr)
Adams Gulch	2,874	851
BWR Upper	491	145
Chocolate Gulch	176	52
Clear Creek	463	137
Cold Springs Gulch	675	200
Cove Canyon	490	145
Croy Creek	2,378	704
Deer Creek	4,937	1,462
Eagle Creek	3,428	1,015
East Fork	1,591	471
Elkhorn Gulch	172	51
Greenhorn Gulch	2,303	682
Indian Creek	8,129	2,407
Lake Creek	8,125	2,406
Lees Gulch	453	134
Ohio Gulch	865	256
Quigley Creek	1,891	560
Seamans Gulch	6,582	1,949
Slaughterhouse Gulch	1,709	506
Townshead Gulch	196	58
Trail Creek	9,787	2,898
Warm Springs Creek	1,645	487

```
r <- rs.model[[1]]
r[] <- NA
r.src <- rasterize(source.lines, r)
is.active <- !is.na(rs.model[["lay1.bottom"]][])
r.src[!is.active] <- NA
```

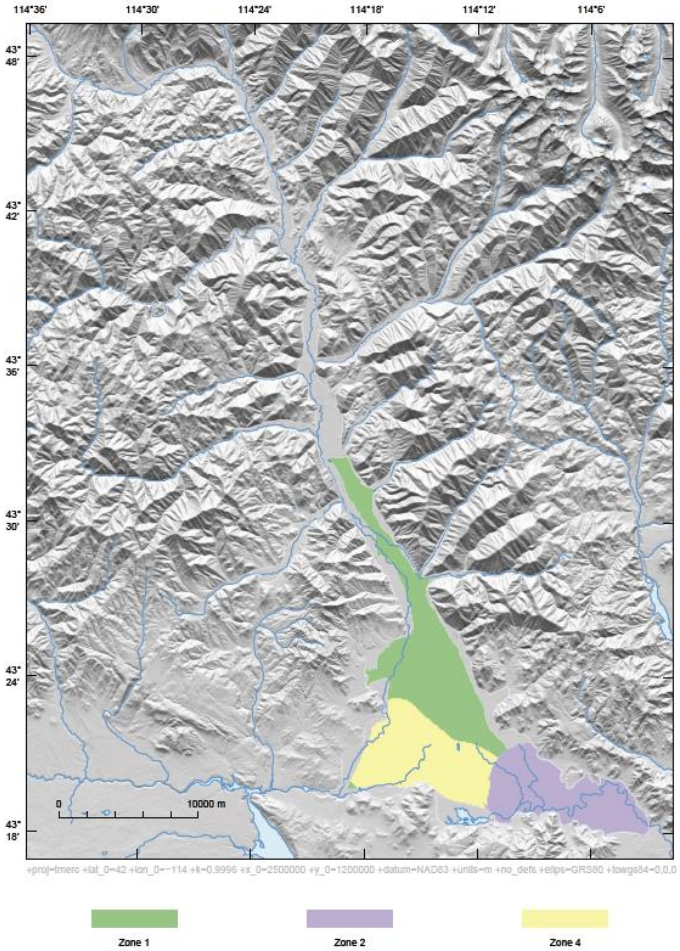


Figure 11: Hydrogeologic zones in model layer 3.


```
src.cells <- which(!is.na(r.src[]))
adj.cells <- adjacent(r.src, src.cells, directions = 4)
is.valid.src <- adj.cells[, 2] %in% which(is.active) & !(adj.cells[, 2] %in% src.cells)
rm.src.cells <- src.cells[!(src.cells %in% unique(adj.cells[is.valid.src, 1]))]
rs.model[["lay1.bottom"]][rm.src.cells] <- NA
rs.model[["lay1.top"]][rm.src.cells] <- NA
r.src[rm.src.cells] <- NA
rat <- levels(r.src)[[1]]
for (i in seq_len(nrow(rat))) {
  trib.cells <- which(r.src[] == rat$ID[i])
  r[trib.cells] <- rat$Flow[i] / length(trib.cells)
}
names(r) <- "lay1.bdry.sources"
rs.model <- stack(rs.model, r)
```

Groundwater Flow Beneath Silver Creek and Stanton Crossing

Groundwater leaving the aquifer system (sink) beneath Silver Creek and Stanton Crossing (fig. 1) is simulated using the MODFLOW Drain Package (Harbaugh *et al.*, 2000), a head-dependent flux boundary condition. If the head in a model cell falls below a certain threshold, the flux drops to zero; therefore, these model cells will only allow groundwater to leave the aquifer system (sink cells). The specified head threshold at Silver Creek and Stanton Crossing are given in table 2. The location of sink cells in model layer 1 is shown in figure 12. Note that the Silver Creek sink cells also reside in model layers 2 and 3.

```
l <- gIntersection(sink.locations, as(aquifer.extent, "SpatialLinesDataFrame"), TRUE)
sink.lines <- SpatialLinesDataFrame(l, data = sink.locations@data, match.ID = FALSE)
r <- rasterize(sink.lines, rs.model, field = "Head")
r[!is.na(r) & is.na(rs.model[["lay1.bottom"]])] <- NA
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- c("Silver Creek", "Staton Crossing")
levels(r) <- rat
names(r) <- "lay1.bdry.sinks"
rs.model <- stack(rs.model, r)
r <- rasterize(sink.lines, rs.model, field = "Head")
r[!is.na(r) & is.na(rs.model[["lay2.bottom"]])] <- NA
names(r) <- "lay2.bdry.sinks"
rs.model <- stack(rs.model, r)
r <- rasterize(sink.lines, rs.model, field = "Head")
r[!is.na(r) & is.na(rs.model[["lay3.bottom"]])] <- NA
names(r) <- "lay3.bdry.sinks"
rs.model <- stack(rs.model, r)
```

Table 2: Specified hydraulic head thresholds for sink cell boundary conditions.

Name	Head (m)	Head (ft)
Silver Creek	1,432	4,698
Stanton Crossing	1,470	4,823

```
initial.head.frac <- 0.90
r <- rs.model[["lay1.bottom"]] + rs.model[["lay1.thickness"]] * initial.head.frac
names(r) <- "lay1.strt"
rs.model <- stack(rs.model, r)
r <- rs.model[["lay1.strt"]]
r[is.na(rs.model[["lay2.bottom"]])] <- NA
names(r) <- "lay2.strt"
rs.model <- stack(rs.model, r)
r <- rs.model[["lay1.strt"]]
r[is.na(rs.model[["lay3.bottom"]])] <- NA
names(r) <- "lay3.strt"
rs.model <- stack(rs.model, r)
```

Export Raster Layers

Create georeferenced image files from layers in the raster stacks. Place these files in a subdirectory of the working directory.

```
dir.out <- file.path(getwd(), paste0("wrv_", format(Sys.time(), "%Y%m%d%H%M%S")))
dir.create(path = dir.out)
ExportRasterStack(file.path(dir.out, "Data"), rs.data)
ExportRasterStack(file.path(dir.out, "Model"), rs.model)
```

Model Run

Steady-state flow in the WRV aquifer system is simulated using the MODFLOW-USG groundwater flow model. This numerical model was chosen for its ability to solve complex unconfined groundwater flow simulations. Input files for MODFLOW-USG are created from raster layers and parameters given in table 5. Horizontal hydraulic conductivity values for the hydrogeologic zones are based on previous estimates by Bartolino and Adkins (2012, table 2, pg. 25-26). Parameter values should be viewed as preliminary and subject to change during model calibration.

```
id <- "wrv_ss_mfusg"  # model run identifier
dir.run <- file.path(dir.out, "Run")
args <- list(rs.model = rs.model, id = id, dir.run = dir.run, perlen = 5479,
             hk = c(21.0, 15.2, 8.6e-7, 12.8), vani = 1000, riv.cond = 850)
do.call(CreateModflowInputFiles, args)
```

Table 5: Input parameters for model run.

Parameter	Value
Length of stress period (d)	5,479
Horizontal hydraulic conductivity of Zone 1 (m/d)	21
Horizontal hydraulic conductivity of Zone 2 (m/d)	15.2
Horizontal hydraulic conductivity of Zone 3 (m/d)	8.6e-07
Horizontal hydraulic conductivity of Zone 4 (m/d)	12.8
Global vertical anisotropy	1,000
Global conductance of riverbed (m^2/d)	850

Create and execute a *batch file* containing commands that run MODFLOW-USG.


```
cmd <- NULL
cmd[1] <- paste("cd", shQuote(dir.run))
cmd[2] <- paste(shQuote(file.exe), shQuote(paste0(id, ".nam")))
file.bat <- file.path(dir.run, "Run.bat")
cat(cmd, file = file.bat, sep = "\n")
Sys.chmod(file.bat, mode = "755")
output <- system(shQuote(file.bat), intern = TRUE)
```

Captured output from running the model is provided below.

```
#                                   MODFLOW-USG
#     U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUNDWATER FLOW MODEL
#                                Version 1.1.00 08/23/2013
#
#  Using NAME file: wrv_ss_mfusg.nam
#  Run start date and time (yyyy/mm/dd hh:mm:ss): 2014/03/25 12:04:56
#
#  Solving:  Stress period:     1    Time step:     1    Groundwater Flow Eqn.
#  Run end date and time (yyyy/mm/dd hh:mm:ss): 2014/03/25 12:07:01
#  Elapsed run time:  2 Minutes,  5.236 Seconds
#
#   Normal termination of simulation
```

The volumetric budget at the end of the 15 year simulation is given in table 6.

Table 6: Volumetric budget for entire model at end of time step 1 stress period 1.

	Vol. (m^3)	Vol. (acre-ft)	Rate (m^3/d)	Rate (acre-ft/yr)
Storage in	0	0	0	0
Constant head in	0	0	0	0
Drains in	0	0	0	0
River leakage in	1,023,633,780	829,869	186,829	55,321
Specified flows in	325,233,440	263,670	59,360	17,577
Total in	1,348,867,220	1,093,538	246,189	72,898
Storage out	0	0	0	0
Constant head out	0	0	0	0
Drains out	277,392,985	224,885	50,628	14,991
River leakage out	1,071,474,524	868,653	195,560	57,907
Specified flows out	0	0	0	0
Total out	1,348,867,509	1,093,538	246,189	72,898
In minus out	-290	-0	-0	-0
Percent discrepancy	-0	-0	-0	-0

Post-Processing

Read simulated hydraulic heads (heads) for each model layer into a raster stack.

```
heads <- ReadModflowBinaryFile(file.path(dir.run, paste0(id, ".hds")))
rs.head <- stack()  # initialize a raster stack for simulated heads
r <- rs.model[[1]]
```

```
r[] <- heads[[1]]$d
r[!rs.model[["lay1.ibound"]]] <- NA
names(r) <- "lay1.head"
rs.head <- stack(rs.head, r)
r[] <- heads[[2]]$d
r[!rs.model[["lay2.ibound"]]] <- NA
names(r) <- "lay2.head"
rs.head <- stack(rs.head, r)
r[] <- heads[[3]]$d
r[!rs.model[["lay3.ibound"]]] <- NA
names(r) <- "lay3.head"
rs.head <- stack(rs.head, r)
```

Determine which model cells are saturated in model layer 1 (fig. 14).

```
r <- rs.head[["lay1.head"]] > rs.model[["lay1.top"]] & rs.head[["lay1.head"]] < 1e30
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- c("Partially Saturated", "Saturated")
levels(r) <- rat
names(r) <- "lay1.saturated"
rs.head <- stack(rs.head, r)
```

Determine the simulated elevation of the water table (fig. 15). Heads above land surface are specified using the land surface elevation.

```
is.above.land.surface <- rs.head[["lay1.head"]] > rs.model[["lay1.top"]]
is.in.lay2 <- rs.head[["lay2.head"]] > rs.model[["lay2.bottom"]] &
              rs.head[["lay2.head"]] < rs.model[["lay1.bottom"]]
is.in.lay3 <- rs.head[["lay3.head"]] < rs.model[["lay2.bottom"]]
r <- rs.head[["lay1.head"]]
r[is.above.land.surface] <- rs.model[["lay1.top"]][is.above.land.surface]
r[is.in.lay2] <- rs.head[["lay2.head"]][is.in.lay2]
r[is.in.lay3] <- rs.head[["lay3.head"]][is.in.lay3]
names(r) <- "water.table"
rs.head <- stack(rs.head, r)
```

Write simulated heads to georeferenced image files.

```
ExportRasterStack(dir.run, rs.head)
```

Reproducibility

To reprocess the groundwater flow model, evaluate R code extracted from this vignette using the following command:

```
source(system.file("doc", "wrv-process.R", package = "wrv"), echo = TRUE)
list.files(dir.out, full.names = TRUE, recursive = TRUE)  # path names of output files
```

Version information about R and attached or loaded packages is as follows:

- R version 3.0.3 (2014-03-06), x86_64-w64-mingw32
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils

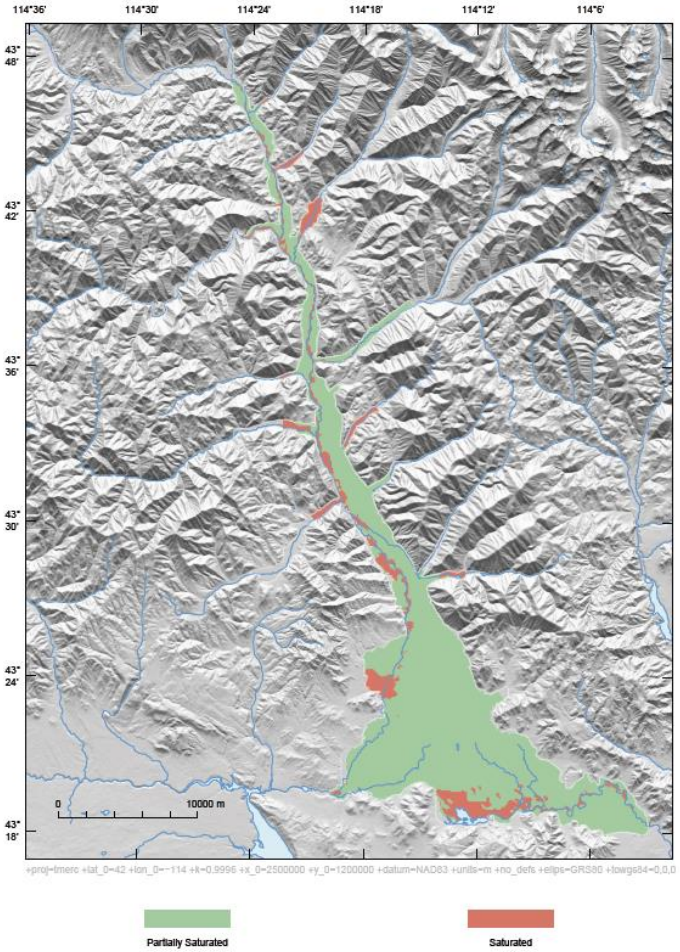


Figure 14: Saturated and partially-saturated cells in model layer 1.

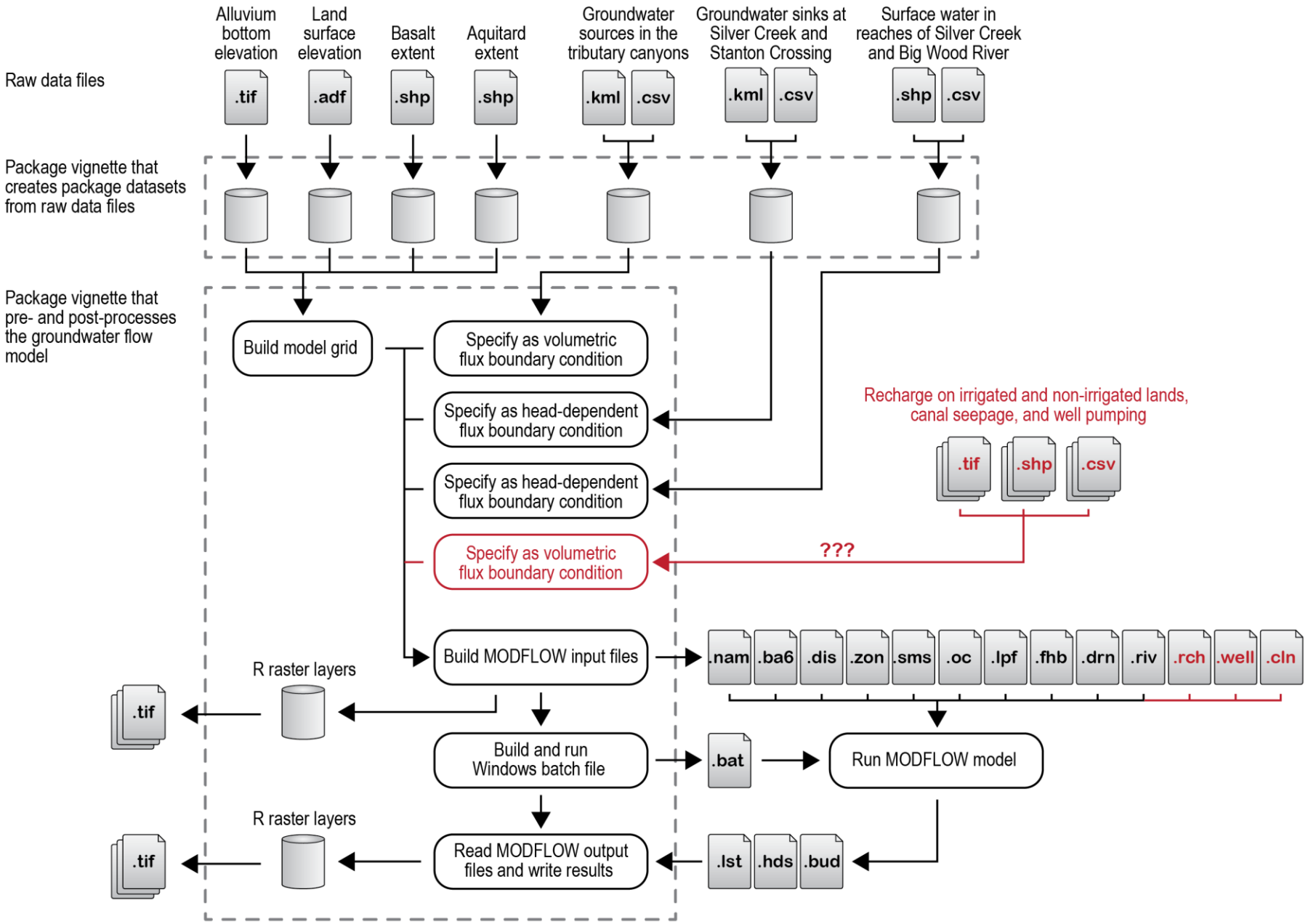
Questions

USGS

Wood River Valley
Update on Processing of Recharge and Well Pumping

Jason C. Fisher

U.S. Geological Survey
Idaho Department of Water Resources


Raw data files
Alluvium bottom elevation
Land surface elevation
Basalt extent
Aquitard extent
Groundwater sources in the tributary canyons
Groundwater sinks at Silver Creek and Stanton Crossing
Surface water in reaches of Silver Creek and Big Wood River
.tif
.adf
.shp
.shp
.kml
.csv
.kml
.csv
.shp
.csv
Package vignette that creates package datasets from raw data files
Package vignette that pre- and post-processes the groundwater flow model
Build model grid
Specify as volumetric flux boundary condition
Specify as head-dependent flux boundary condition
Specify as head-dependent flux boundary condition
Specify as volumetric flux boundary condition
Recharge on irrigated and non-irrigated lands, canal seepage, and well pumping
.tif
.shp
.csv
???
Build MODFLOW input files
.nam
.ba6
.dis
.zon
.sms
.oc
.lpf
.fhb
.drn
.riv
.rch
.well
.cln
R raster layers
.tif
Build and run Windows batch file
.bat
Run MODFLOW model
R raster layers
.tif
Read MODFLOW output files and write results
.lst
.hds
.bud

Recharge on Non-Irrigated Lands

- Precipitation minus evapotranspiration
- Model layer 1 cells in all stress periods
- Recharge Package (non-anthropogenic recharge)

Recharge on Irrigation Entities

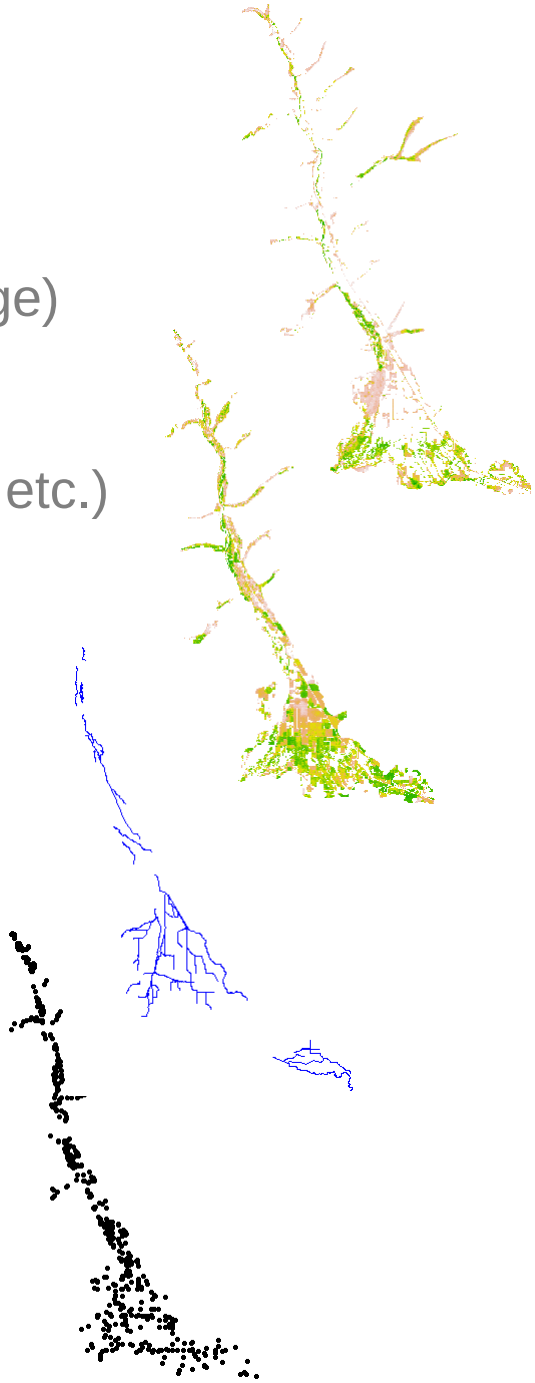
- Complex calculation (dependent on diversions, etc.)
- Model layer 1 cells, all stress periods
- Well Package (anthropogenic recharge)

Seepage along Canals

- Fraction of diversions for irrigation entities
- Model layer 1 cells, all stress periods
- Well Package (anthropogenic recharge)

Pumping at Well Sites

- Observed and estimated values of pumping
- Model layer(s) 1, 2, 3 cells, all stress periods
- Connected Linear Network Package



Recharge on Irrigated Lands, Canal Seepage, and Well Pumping

1. Export WRV_POD.shp to temporary file WRV_POD_YearMo.shp
2. Add field StressPeriodGWRate to WRV_POD_YearMo.shp
3. Join summary table created in e.ii.2 to WRV_POD_YearMo.shp (join GWWRNo to WaterRight), **keep all records**
4. If SWCurt <> Null, calculate *YearMoGWRate = IRRcfs * SWCurt*
5. If SWCurt = Null, calculate *YearMoGWRate = IRRcfs*
6. Remove join

iv. Remove groundwater right records from WRV_POD_YearMo.shp if WMISNumber = Well Number with a record in WRV_GWDiv.csv for this stress period

v. For each EntityName in WRV_POD_YearMo.shp, determine the fraction of GWDivEst to assign to each WMISNumber for this stress period.

1. Calculate sum of YearMoGWRate for each EntityName
2. Add field YearMoFraction and calculate as *YearMoFraction = YearMoGWRate/(sum of YearMoGWRate for EntityName)*
3. Add field YearMoGWDiv and calculate as *YearMoGWDiv = YearMoFraction * (GWDivEst for EntityName)*
4. Repeat for other EntityNames

vi. Summarize WRV_POD_YearMo.shp by WMISNumber with sum of YearMoGWDiv and write result to GWDivEst.csv in format WMISNumber, YearMo, YearMoGWDiv. Diversion volume is in AF/month.

vii. Repeat for other April through October stress periods.

f. **ASSIGN RECORDED AND ESTIMATED GROUNDWATER DIVERSIONS TO MODEL CELLS AND LAYERS.**

i. Add field MdlLayer to WRV_POD.shp and calculate layer(s) from fields TopOpen1, BotOpen1, TopOpen2, BotOpen2 and model layer depths below ground surface (anticipate many wells completed in more than one layer, will we use MNW or CLN instead of WEL package to deal with multi-layer wells???).

ii. For each stress period:

1. Assign recorded groundwater diversions from WRV_GW_Div.csv for Well Number to model

```r

  priority.cut <- priority.cuts[priority.cuts$YearMonth == i, "PdateNum_BWR"]
  is.lt.priority.cut <- !is.sc.src & d$PdateNum < priority.cut
  d$sw.rate[!is.sc.src] <- 0
  d$sw.rate[is.lt.priority.cut] <- d$MaxDivRate[is.lt.priority.cut]

  priority.cut <- priority.cuts[priority.cuts$YearMonth == i, "PdateNum_SC"]
  is.lt.priority.cut <- is.sc.src & d$PdateNum < priority.cut
  d$sw.rate[is.sc.src] <- 0
  d$sw.rate[is.lt.priority.cut] <- d$MaxDivRate[is.lt.priority.cut]

  d.agg <- cbind(aggregate(d$MaxDivRate, by=list(d$WaterRight), sum),
                 aggregate(d$sw.rate, by=list(d$WaterRight), sum)[, 2])
  names(d.agg) <- c("WaterRight", "MaxDivRate", "sw.rate")
  d.agg$sw.curt <- 1 - d.agg$sw.rate / d.agg$MaxDivRate

  pod <- merge(pod.gw@data, d.agg, by="WaterRight", all.x=TRUE)
  pod$gw.rate <- pod$IRRcfs
  is.sw.curt.na <- is.na(pod$sw.curt)
  pod$gw.rate[!is.sw.curt.na] <- pod$IRRcfs[!is.sw.curt.na] *
                                 pod$sw.curt[!is.sw.curt.na]

  pod.agg <- aggregate(pod$gw.rate, by=list(pod$EntityName), sum)
  names(pod.agg) <- c("EntityName", "gw.rate")
  idxs <- match(pod$EntityName, pod.agg$EntityName)
  pod$fraction <- pod$gw.rate / pod.agg$gw.rate[idxs]

  pod$gw.div <- 0  # TODO(jfisher): had to intialize to 0
  div <- div.by.entity[[i]][, c("EntityName", "gw.div.est")]
  idxs <- match(pod$EntityName, div$EntityName)
  pod$gw.div[!is.na(idxs)] <- pod$fraction[!is.na(idxs)] * div$gw.div.est[!is.na(idxs)]

  return(pod)
}
pod.gw.data <- lapply(yr.mo.irr, Fun)
names(pod.gw.data) <- yr.mo.irr

Fun <- function(i) {
  d <- aggregate(pod.gw.data[[i]]$gw.div, by=list(pod.gw.data[[i]]$WMISNumber), sum)
  d <- cbind(d, i)
  names(d) <- c("WMISNumber", "gw.div", "YearMonth")
  return(d)
}
summary.pod.gw.data <- do.call(rbind, lapply(names(pod.gw.data), Fun))


## 3f. assign recorded and estimated groundwater diversions to model cells and layers


```

Recharge on Non-Irrigated Lands

f. Repeat for every Apr, May, Jun, Jul, Aug, Sep, and Oct.

10. <u>Assign Recharge to the model grid (I have not done this step, this is one option for doing it).</u>
    a. This step may need to change to include Jennifer's recharge values on irrigated land.
    b. Assign recharge values to *ACTIVE* model grid cells based on the CELL_INTGR of the cell.
    c. Use the Spatial Analyst extension in ArcMap. Use the Spatial Analysis Tools Toolbox and the Zonal Tool called "Zonal Statistics as Table" which assigns the values from a raster to the zones of a shapefile in a table format.
    d. Not sure if you need to set Environments, but I did it anyway.
        i. Processing Extent: set to "ET.199604.MET.tif"
        ii. Snap Raster: set to "ET.199604.MET.tif"
        iii. Raster Analysis → Cell Size: set to "ET.199604.MET.tif"
    e. An example:

Input raster or feature zone data (this is our Model Grid.shp)
Zone field (this is the CELL_INTGR field in the ModelGrid.shp)
Input value raster (this is our "RECHG.YYYYMM.tif" for winter months or "RECHG.NOIRR.201004.tif" for growing season months. Input for growing season may change as we integrate Jennifer's data with my data).
Output table (this will be the table that is the basis for the recharge file)
Specific example:
In the file "ModelGrid.shp": Open Properties → Definition Query →Query Builder Select "Active Grid" = 1
In Spatial Analysis Toolbox →Zonal Statistics as Table
Input raster or feature zone data = "Model Grid.shp"
Zone field = "CELL_INTGR"
Input value raster = "RECHG.201001.tif"
Output table = "RECHG.201001.dbf"

##Note that Arc Map creates an INFO table if you don't explicitly state the .dbf extension. The INFO tables seem to only be usable within an Arc project. Even the .dbf tables have strange hidden characters that have made my attempts at scripting a way to use this data impossible for me (I hope that you can work around this). When I use these tables I wind up

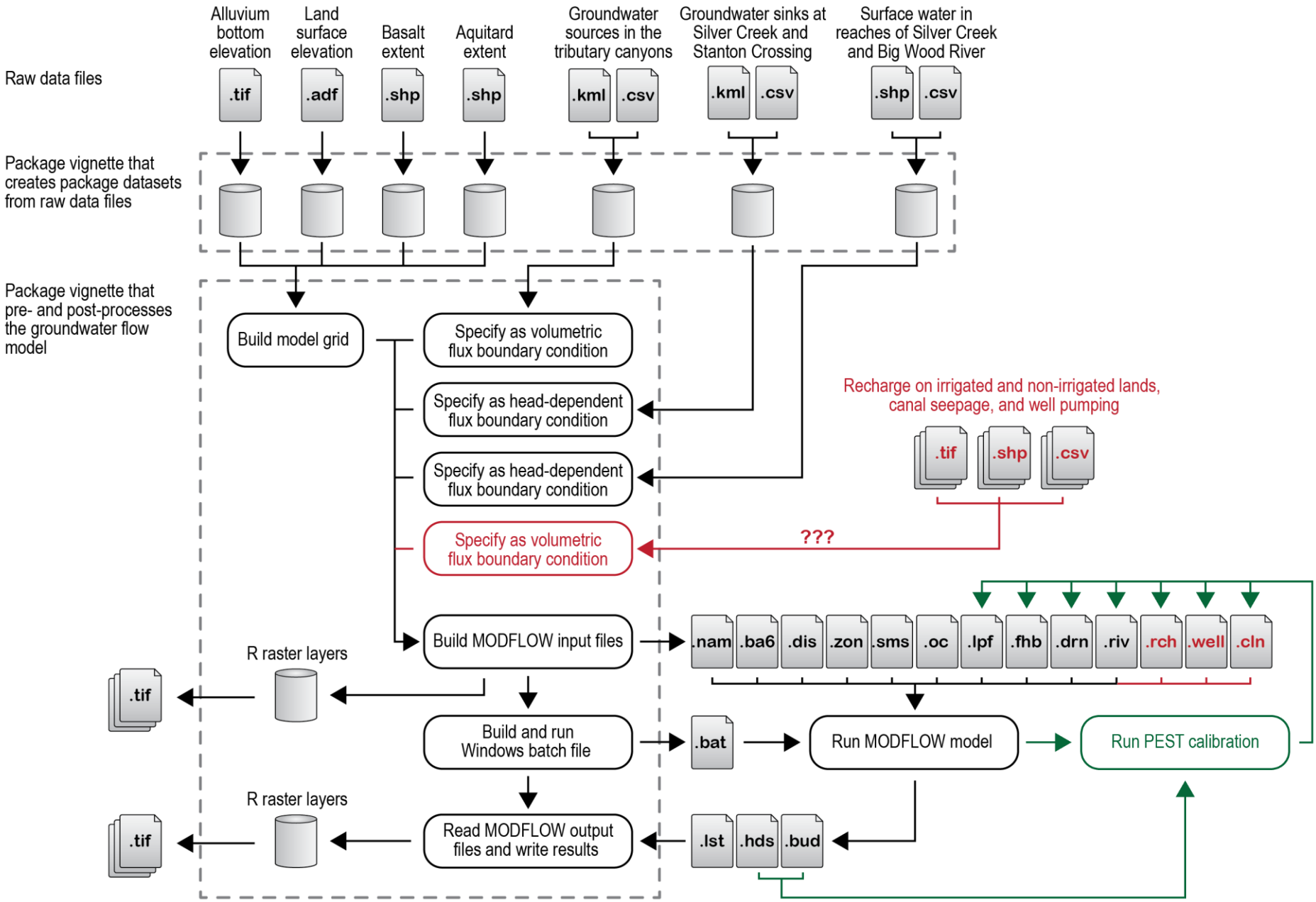
```r
r <- rasterize(precip.zones, rs.et, "ID", silent=TRUE)
r <- ratify(r)
rat <- levels(r)[[1]]
rat$att <- precip.zones@data$PrecipZone
levels(r) <- rat
r.precip.zones <- r

soil.infiltration <- rasterize(soils, rs.et, "INF", silent=TRUE)

Fun <- function(i) {
  p <- precipitation[precipitation$YearMonth == i, c("PrecipZone", "Precip_ft")]
  p <- p[match(p$PrecipZone, levels(r.precip.zones)[[1]]$att), "Precip_ft"]
  names(p) <- levels(r.precip.zones)[[1]]$att
  r <- r.precip.zones
  r[r == 1L] <- p[1]
  r[r == 2L] <- p[2]
  r[r == 3L] <- p[3]
  r <- r - (rs.et[[i]] * mm.to.ft)
  is.pos <- r > soil.infiltration
  r[is.pos] <- soil.infiltration[is.pos]
  if (i %in% yr.mo.irr)
    r[rasterize(gUnaryUnion(et.by.entity[[i]]), r, background=0, silent=TRUE) == 1] <- NA
  names(r) <- i
  return(r)
}
rs.inf <- do.call("stack", lapply(names(rs.et), Fun))
```

TODO List

1. Validate R code pertaining to processing of recharge on irrigated lands, canal seepage, and well pumping
2. Create and document new R datasets: precipitation, diversions, evapotranspiration, canal seepage, etc.
3. Integrate new processing instructions into package vignettes
4. Develop input files for the Recharge, Well, and Connected Linear Network Packages
5. Construct transient model run
6. Decrease run time for a MODFLOW simulation (currently at about 2 minutes)
7. Collaborate on model calibration…


Raw data files
Alluvium bottom elevation
Land surface elevation
Basalt extent
Aquitard extent
Groundwater sources in the tributary canyons
Groundwater sinks at Silver Creek and Stanton Crossing
Surface water in reaches of Silver Creek and Big Wood River
.tif
.adf
.shp
.shp
.kml
.csv
.kml
.csv
.shp
.csv
Package vignette that creates package datasets from raw data files
Package vignette that pre- and post-processes the groundwater flow model
Build model grid
Specify as volumetric flux boundary condition
Specify as head-dependent flux boundary condition
Specify as head-dependent flux boundary condition
Specify as volumetric flux boundary condition
Recharge on irrigated and non-irrigated lands, canal seepage, and well pumping
.tif
.shp
.csv
???
Build MODFLOW input files
.nam
.ba6
.dis
.zon
.sms
.oc
.lpf
.fhb
.drn
.riv
.rch
.well
.cln
R raster layers
.tif
Build and run Windows batch file
.bat
Run MODFLOW model
Run PEST calibration
R raster layers
.tif
Read MODFLOW output files and write results
.lst
.hds
.bud

Questions

USGS