

# Modflow User Tools (MUT) Version 1.31

## User's Guide



October 2024

Rob McLaren, Young-jin Park, Sorab Panday

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Software Installation and Useage</b>	<b>5</b>
<b>3</b>	<b>Model Build</b>	<b>14</b>
3.1	Defining the Units of Length and Time . . . . .	18
3.2	Defining the Template Mesh . . . . .	19
3.3	Groundwater Flow(GWF) Domain . . . . .	22
3.3.1	Generating a Layered GWF Domain . . . . .	23
3.3.1.1	Defining the Top Elevation . . . . .	23
3.3.1.2	Adding Layers . . . . .	25
3.3.1.3	Cell Connection Properties . . . . .	28
3.3.1.4	Assigning Material Properties . . . . .	29
3.3.1.5	Initial Conditions . . . . .	38
3.3.1.6	Boundary Conditions . . . . .	39
3.4	Surface Water Flow(SWF) Domain . . . . .	40
3.4.1	Generating a SWF Domain . . . . .	40
3.4.1.1	Cell Connection Properties . . . . .	41
3.4.1.2	Assigning Material Properties . . . . .	41
3.4.1.3	Initial Conditions . . . . .	43
3.4.1.4	Boundary Conditions . . . . .	44
3.5	Connected Linear Network(CLN) Domain . . . . .	46
3.5.1	Generating a CLN Domain . . . . .	46
3.5.1.1	Cell Connection Properties . . . . .	48
3.5.1.2	Assigning Material Properties . . . . .	48

3.5.1.3	Initial Conditions . . . . .	49
3.5.1.4	Boundary Conditions . . . . .	49
3.6	Stress Periods . . . . .	49
3.7	Output Control . . . . .	53
3.8	Solver Parameters . . . . .	54
3.9	3D Model Build Visualization . . . . .	56
3.9.1	GWF Domain . . . . .	58
3.9.2	SWF Domain . . . . .	65
3.9.3	CLN Domain . . . . .	72
<b>4</b>	<b>Modflow-Usg<sup>Swf</sup> Execution and Post-Processing</b>	<b>75</b>
4.1	MODFLOW-USG <sup>Swf</sup> Execution . . . . .	75
4.2	MUT Post-processing . . . . .	77
4.3	Volumetric Water Budget Plots . . . . .	77
4.4	3D Visualization of Model Results . . . . .	80
4.4.1	Solution Times and Animation . . . . .	81
4.4.2	Data Set Infomation . . . . .	82
4.4.3	Inactive Cells and Value Blanking . . . . .	83
4.4.4	Slices and Fence Diagrams . . . . .	84
4.4.5	Defining New Variables . . . . .	85
4.4.6	Water Table Isosurface Plot . . . . .	87
4.4.7	Infiltration Plot . . . . .	88
<b>5</b>	<b>Model Verification</b>	<b>90</b>
5.1	1D Variably-saturated Flow in a Column . . . . .	90
5.2	2D Variably-saturated Flow in a Hillslope: Drains vs Surfacewater Flow . . . . .	92
5.3	3D Fully-coupled Groundwater-Surface Water Flow: Abdul's Experiment . . . . .	93
<b>A</b>	<b>Microsoft Excel Database Files</b>	<b>95</b>

# Chapter 1

## Introduction

This document describes a new MODFLOW-USG<sup>1</sup> development environment which has these features:

- We refer to it as Modflow User Tools, or MUT for short.
- MUT is designed to work with a modified version of MODFLOW-USG, where a new surface water flow package, called SWF, has been added. Like the Connected Linear Network (CLN) package, the SWF package represents a new domain type that is fully-coupled to the 3D groundwater flow (GWF) domain. There can also be cell-to-cell flows between the SWF and CLN domains. The SWF domain uses the diffusion-wave approach so simulate 2D surface-water flow. We will refer to this new version of MODFLOW-USG as MODFLOW-USG<sup>Swf</sup> in this manual.
- We currently develop and run it on a MICROSOFT WINDOWS 10-based computing platform, writing software using the INTEL FORTRAN compiler running inside the MICROSOFT VISUAL STUDIO interactive development environment, which includes software version control tools through GITHUB.
- A text-based approach is used for the MUT interface, in which we first develop an input file of instructions that define our MODFLOW-USG<sup>Swf</sup> project, then run MUT to read it and write a complete MODFLOW-USG<sup>Swf</sup> data set. MUT also writes output files for TECPLOT, a third-party visualization software package, which provides a 3D graphical visualization tool to review the model numerical mesh and material properties in the data set. In future, MUT could be extended to support other third-party visualization packages, for example the open source program Paraview.
- MUT can post-process a MODFLOW-USG<sup>Swf</sup> simulation to provide a TECPLOT visualization of temporal model results, including hydraulic heads, saturations, water depths and flow budget data. *If applied to output files which were produced by an earlier version of Modflow, results may be mixed. It is not our intent here to support all existing Modflow packages, many of which have been superceded.*

This document is subdivided into these sections:

**Chapter 2 Installation and Setup:** How to install MUT, MODFLOW-USG<sup>Swf</sup> and TECPLOT and define MICROSOFT WINDOWS environment variables.

---

<sup>1</sup><https://www.gsienv.com/software/modflow-usg/modflow-usg/>

**Chapter 3 Mut Execution and Pre-processing** How to build a MUT input file, produce a MODFLOW-USG<sup>Swf</sup> compatible data set and TECPLOT compatible output files with MUT, then review the results of the model build with TECPLOT.

**Chapter 4 Modflow-Usg<sup>Swf</sup> Execution and Post-Processing** How to run MODFLOW-USG<sup>Swf</sup>, convert the output to TECPLOT-compatible files with MUT, then visualize them with TECPLOT.

**Chapter 5 Model Verification** Examples used to verify the accuracy of MODFLOW-USG<sup>Swf</sup> models built using MUT.

**Chapter ?? Illustrative Example** An example which illustrates the use of MUT and MODFLOW-USG<sup>Swf</sup> to simulate variably-saturated, fully-coupled GWFSWF flow in a large-scale watershed.

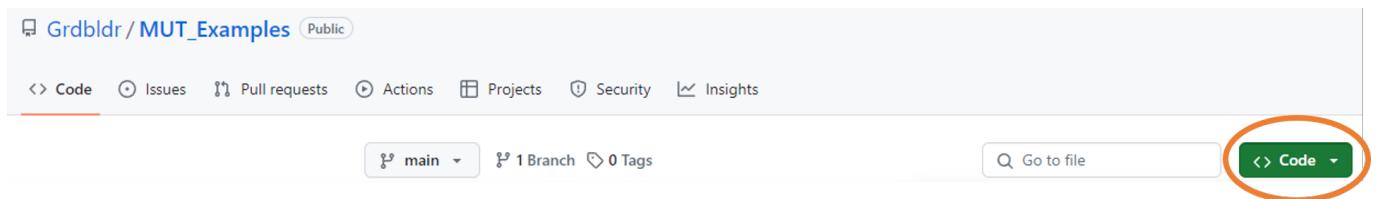
**Appendix A Microsoft Excel Database Files** Details about using the provided MICROSOFT EXCEL database files, which are currently used to store MODFLOW-USG<sup>Swf</sup> model material property and solver parameter data sets.

# Chapter 2

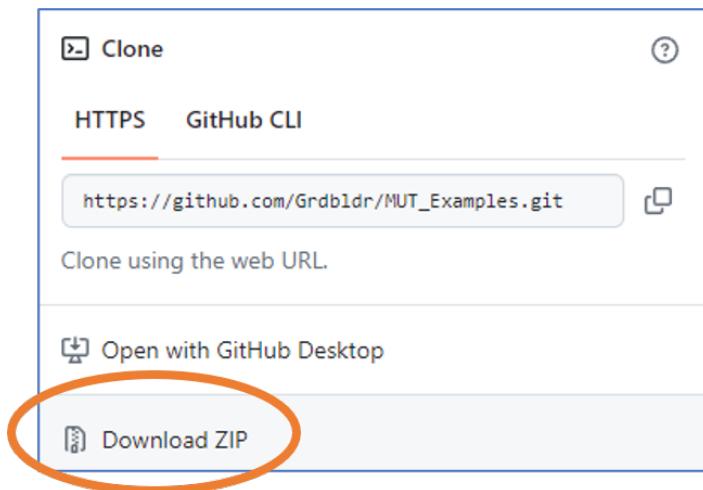
## Software Installation and Useage

The first step in the software installation process is to obtain the MUT examples, executables and database files from GITHUB. To do this:

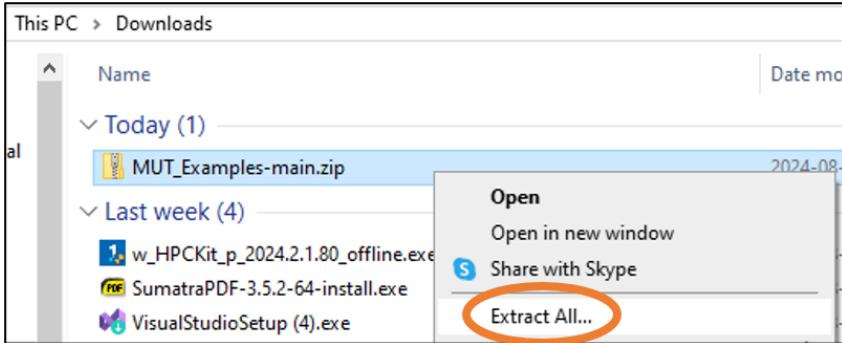
- Click on this link, [https://github.com/Grdbldr/MUT\\_Examples.git](https://github.com/Grdbldr/MUT_Examples.git), which will take you the MUT\_Examples GITHUB page.
- Click on the green 'Code' button.



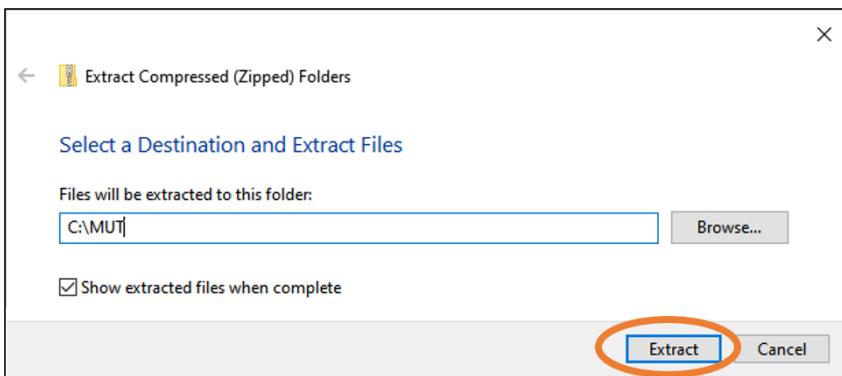
- Choose 'Download ZIP' from the drop-down menu.



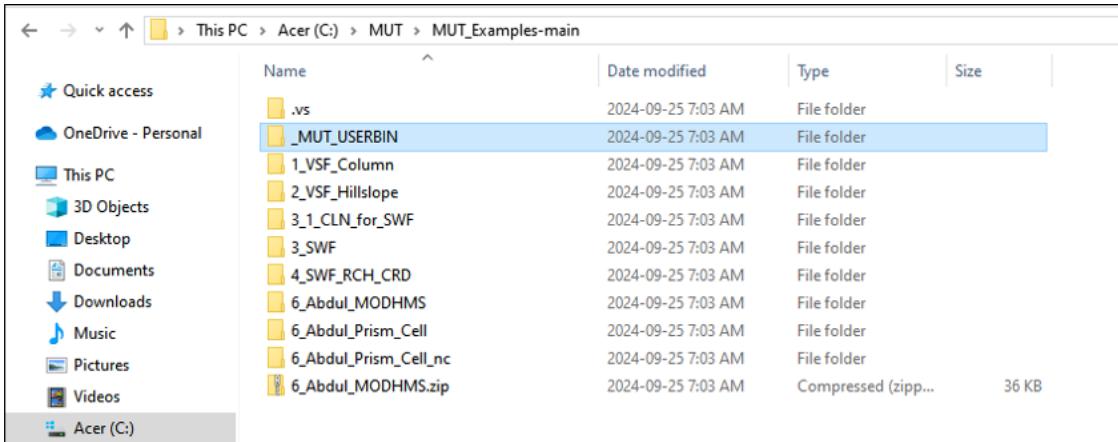
Once the download is complete, the zip file can be found in the MICROSOFT WINDOWS Downloads folder. The contents need to be extracted to a local directory by right-clicking on the download file and choosing 'Extract All...' from the drop-down menu:



This opens the Extract dialogue, where you are free to choose a different drive and folder to store the extracted files. Here we changed the destination folder to C:\MUT. Click the 'Extract' button:



The extracted contents can now be found in the specified destination folder:



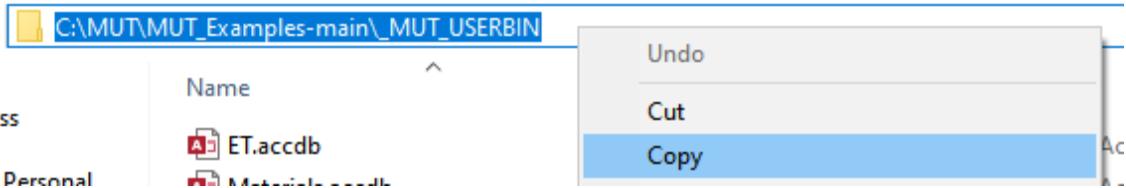
This folder contains a subfolder called '\_MUT\_USERBIN', which contains the following files:

Name	Date modified	Type	Size
mut.exe	2024-10-17 4:07 PM	Application	6,746 KB
USGS_1.exe	2024-08-26 6:57 AM	Application	4,002 KB
tecio.dll	2018-12-19 5:26 AM	Application exten...	1,732 KB
CLN.csv	2024-10-18 10:47 AM	Microsoft Excel C...	1 KB
ET.csv	2024-10-17 10:21 AM	Microsoft Excel C...	1 KB
GWF.csv	2024-10-18 11:24 AM	Microsoft Excel C...	3 KB
LAI.csv	2024-10-16 9:54 AM	Microsoft Excel C...	1 KB
LAI_default.csv	2024-10-17 10:10 AM	Microsoft Excel C...	1 KB
SMS.csv	2024-10-17 10:18 AM	Microsoft Excel C...	2 KB
SWF.csv	2024-10-18 11:30 AM	Microsoft Excel C...	1 KB
CLN.xlsx	2024-10-17 7:32 AM	Microsoft Excel W...	13 KB
ET.xlsx	2024-10-17 10:20 AM	Microsoft Excel W...	13 KB
GWF.xlsx	2024-10-17 7:22 AM	Microsoft Excel W...	14 KB
SMS.xlsx	2024-10-17 10:18 AM	Microsoft Excel W...	13 KB
SWF.xlsx	2024-10-17 10:24 AM	Microsoft Excel W...	13 KB

These include the executable and supporting files for MUT and the executable for MODFLOW-USG<sup>Swf</sup>.

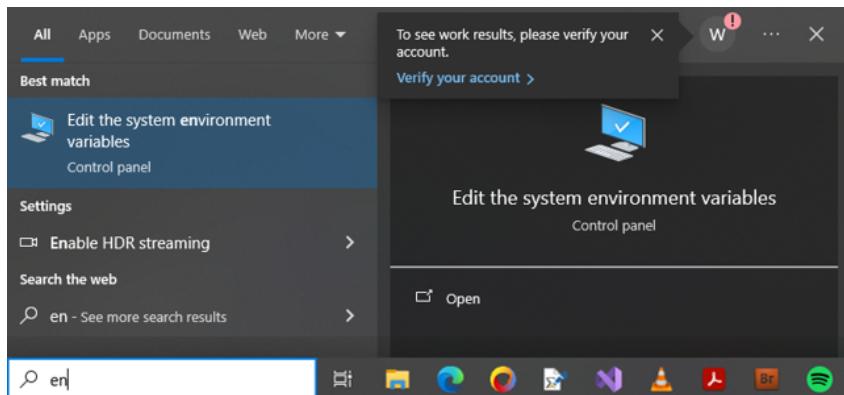
Before you run MUT for the first time, you need to define a windows environment variable called USERBIN, which contains the path to the \_MUT\_USERBIN folder, and modify the existing PATH variable.

First, highlight the path by clicking on it in the File Explorer window, then copy it by pressing CTRL-C or by right-clicking and choosing 'Copy' from the drop-down menu, as shown here:

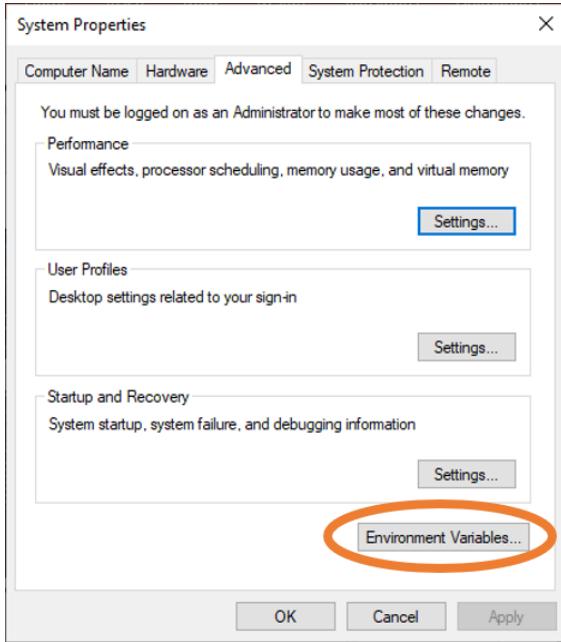


To define the environment variables:

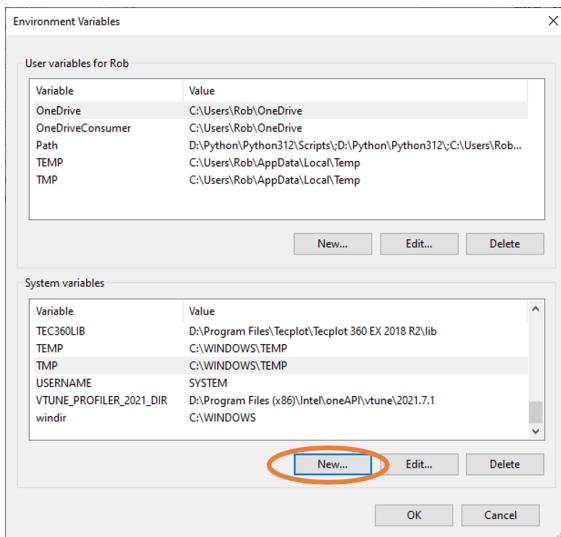
- Type the string 'en' in the windows taskbar search field and open the 'Edit the system environment variables' dialogue:



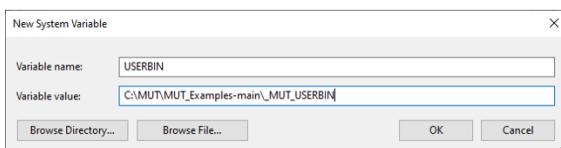
- Click on the 'Environment variables...' button at the bottom of the dialogue:



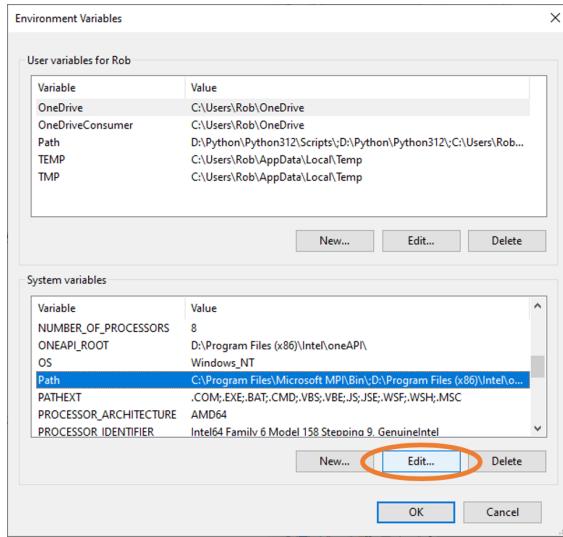
- Click on the 'New' button:



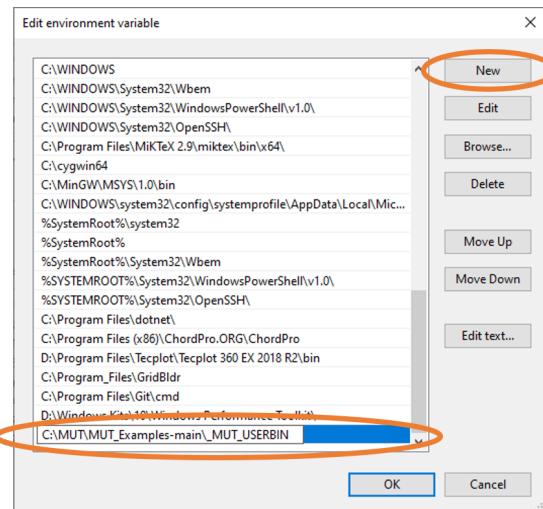
- Add a new variable named USERBIN and define the variable value by pasting in the path copied earlier, then click the 'OK' button:



- Add the path to USERBIN to the existing Path variable. First, choose Path, then click the 'Edit...' button:



- Click the 'New' button and paste in the path copied earlier, then click the 'OK' button:



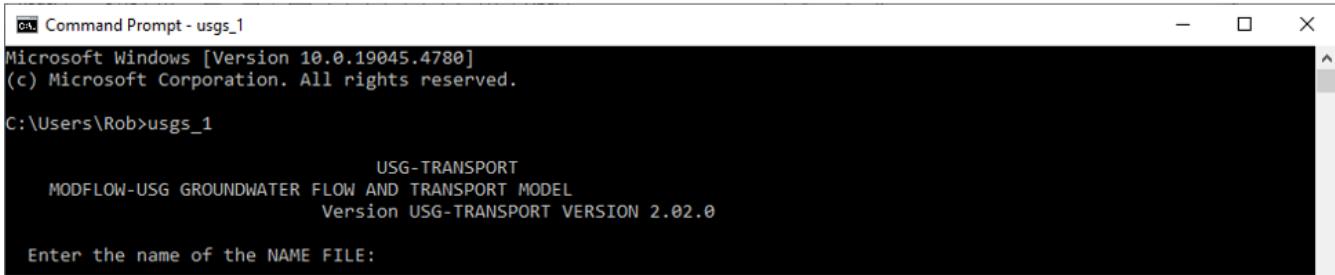
You should now be able to run MUT and MODFLOW-USG<sup>Swf</sup> from the command prompt. To test this, start a new command prompt, then type `mut`, you should see the MUT header:

```
Command Prompt - mut
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rob>mut
MUT version 1.25
No command line prefix
No file: _mut.pfx
Checking for default file: a.mut
No file: a.mut
Enter a prefix for a mut file:
```

Type **ctrl-C** to stop the program.

Run MODFLOW-USG<sup>Swf</sup> by typing `usgs_1`. You should see the MODFLOW-USG<sup>Swf</sup> header:



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rob>usgs_1

USG-TRANSPORT
MODFLOW-USG GROUNDWATER FLOW AND TRANSPORT MODEL
Version USG-TRANSPORT VERSION 2.02.0

Enter the name of the NAME FILE:
```

Type **ctrl-C** to stop the program.

If this is not the case, check the definitions of the **USERBIN** and **PATH** variables. If they are correct, you may need to re-boot your computer and try again.

A licensed version of **TECPLOT** can be obtained from <https://tecplot.com/products/tecplot-360/>. They have a free 30-day trial option for those who want to assess the software before purchase. They also offer educational discounts.

Those of you who are just interested in running the **MUT** and **MODFLOW-USG<sup>Swf</sup>** programs have completed the required software installation tasks and can proceed to Chapter 3, **Model Build**.

Those who want to view and possibly modify and re-compile the source code for **MUT** and **MODFLOW-USG<sup>Swf</sup>** should proceed with these instructions for setting up your **MICROSOFT WINDOWS** programming environment.

As was stated earlier, we use and recommend **MICROSOFT VISUAL STUDIO** and **INTEL FORTRAN**. You should install **MICROSOFT VISUAL STUDIO** before **INTEL FORTRAN**, which will then be automatically integrated into **MICROSOFT VISUAL STUDIO**.

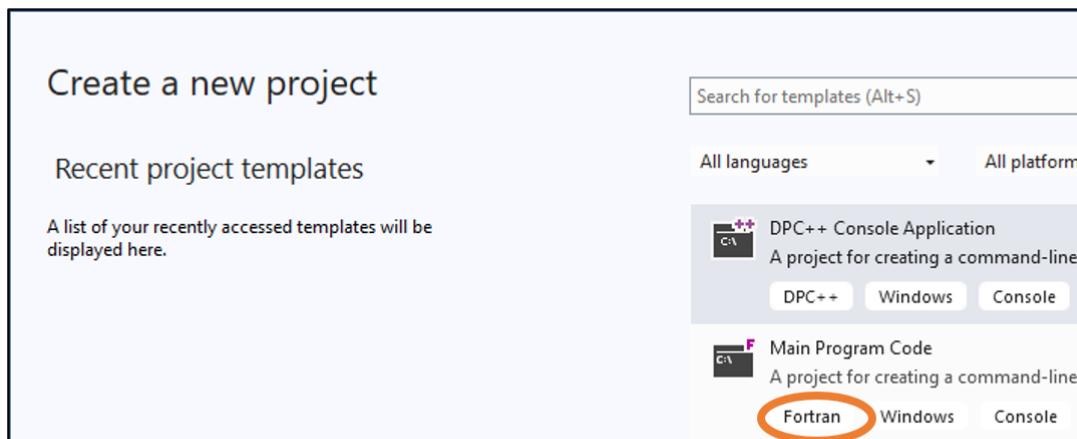
A free version of the latest **MICROSOFT VISUAL STUDIO** (currently 2022) can be obtained from <https://visualstudio.microsoft.com/vs/community/>. Once you are on the site just click the **Download** button. This will download a file (e.g. **VisualStudioSetup.exe**) which can be run to install **MICROSOFT VISUAL STUDIO**. If you already have a version of **MICROSOFT VISUAL STUDIO**, you can choose to keep your old version and add the latest version. When you come to the installation options 'Workloads' page, be sure to check the option for **Desktop development with C++**, shown here:



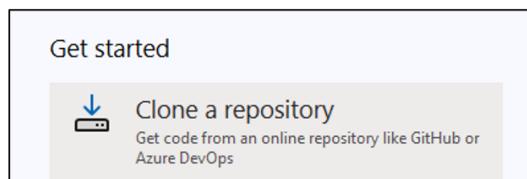
A free version of the latest **INTEL FORTRAN** compiler can be obtained from <https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit.html>. Once you are on the site just click the **Get It Now** button to download the Intel® HPC Toolkit, which includes **INTEL FORTRAN**. Choose the **Windows** option then the **Offline Installer** option. Now you can either fill in the required information and start the download or choose to **Continue as guest** (download starts immediately).

This will download a file (e.g. `w_HPCKit_p_2024.2.1.80_offline.exe`) which can be run to install INTEL FORTRAN.

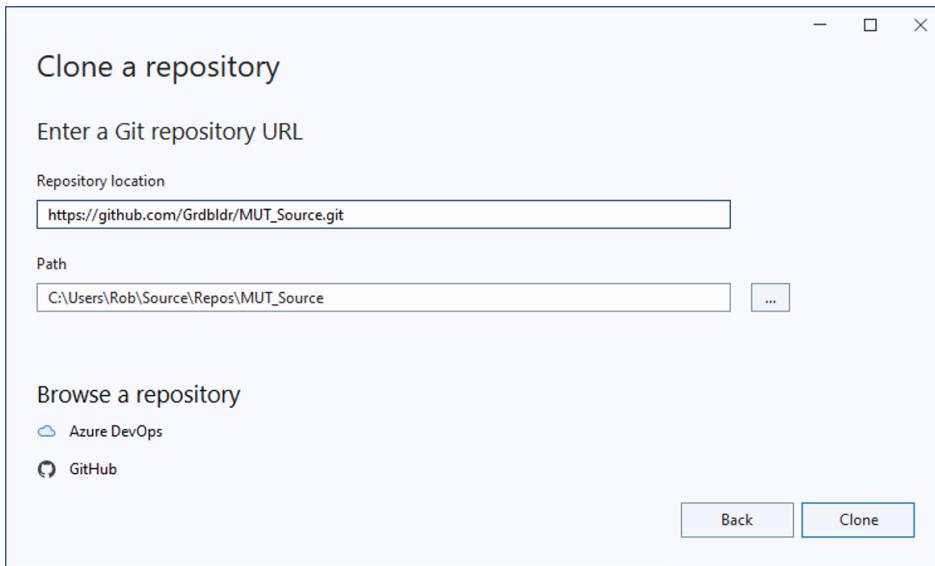
You can check the installation of MICROSOFT VISUAL STUDIO and INTEL FORTRAN by starting MICROSOFT VISUAL STUDIO and choosing **Create a new project**. The window that appears should have links for creating Fortran projects, as shown here:



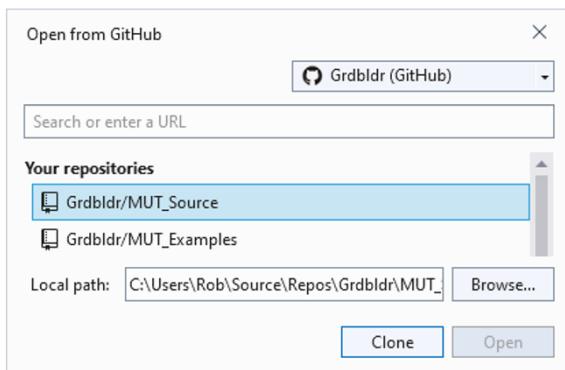
The MUT source files can be obtained from a GITHUB repository at [https://github.com/Grdbldr/MUT\\_Source.git](https://github.com/Grdbldr/MUT_Source.git). Since GITHUB has been integrated into MICROSOFT VISUAL STUDIO we will use it to download the MUT repository. When you start MICROSOFT VISUAL STUDIO choose this option:



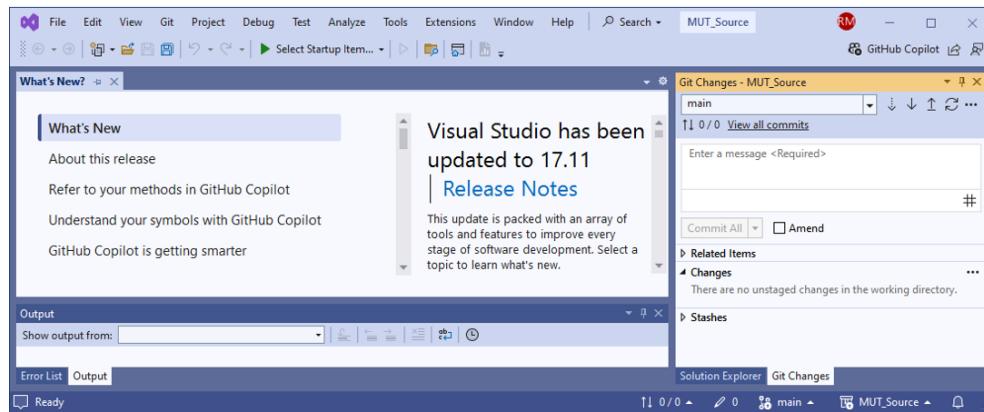
This opens the dialogue box shown below, where you can define the repository location on GITHUB and the path to the local repository. You can copy the link from the PDF file by right-clicking on it and choosing **Copy Link Address**.



Now choose the **GitHub** option under **Browse a Repository** and you will see this dialogue shown below, Choose **Grdbldr/MUT\_Source** from the list of repositories then click the **Clone** button.



This shows the MICROSOFT VISUAL STUDIO window after **Grdbldr/MUT\_Source** has been cloned. Note the **GITHUB** window on the right side, and information along the bottom about the project:



Details about using GITHUB in MICROSOFT VISUAL STUDIO are given in Tutorial ??.

The software has been developed and tested under:

- Windows 10
- TECPLT360 EX 2018 R2
- Microsoft Visual Studio Community 2022, Version 17.11.1
- Intel® Fortran Compiler 2024.1

# Chapter 3

## Model Build

The first step in any model build is to develop a conceptual model, which defines the extent, inflows and outflows, material distributions and physical properties of a hydrogeologic flow system, real or imaginary. The intent of MUT is then to facilitate the production of a set of MODFLOW-USG<sup>Swf</sup> input files by minimizing the amount of time we spend building and testing it. This chapter describes our current model build workflow, which can provide a sound basis for developing your own personal workflow.

The steps in our model build workflow are:

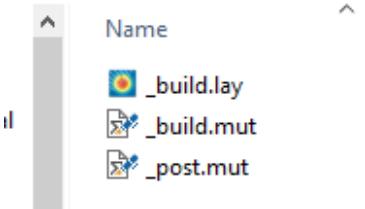
1. Create a new working folder or copy an existing MUT project folder.
2. Modify the MUT input file (and other input files if necessary) to reflect the new Modflow project.
3. Run MUT to build the new Modflow project, which also produces TECPLOT output files for the various Modflow domains (i.e. GWF,SWF and/or CLN) created during the build process.
4. Run TECPLOT and examine the build output files.
5. Repeat steps 2-4 until the new project is defined correctly.

A MUT input file is a plain ascii text file that you can edit with your preferred editor (e.g. Windows Notepad<sup>1</sup>). The MUT input file name can have a prefix of your choice, followed by the extension .mut. Examples of valid MUT input file names are \_build.mut or good.mut. Most often, the easiest approach is to copy an existing input file and modify it as required. This helps reduce set-up time and avoid potential errors that are introduced when creating input files from scratch.

To illustrate our model build workflow, we will refer to the various conceptual models developed for our existing suite of verification examples described in Chapter 5. As you read along, we urge you to carry out the steps we describe as we move through the workflow. We recommend copying the contents of an existing model to a new location (e.g. copy the folder MUT\_Examples\1\_VSF\_Column to C:\SandBox). If you did so, your working directory would look something like this:

---

<sup>1</sup>Our personal favourite editor is WinEdt (<https://www.winedt.com/snap.html>), which also provides a nice LATEX document development environment when coupled with the T<sub>E</sub>X software package MiK<sub>T</sub>eX. This manual was produced using these word processing tools.



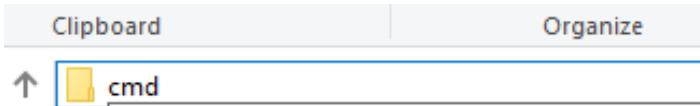
In this example, there are two MUT input files, one for the model build called `_build.mut`, one for post-processing called `_post.mut` (discussed later in chapter 4) and a TECPLOT layout file called `_build.lay` used to visualize the model build results.

In our preferred workflow we would first start a command prompt in the folder which contains the MUT input file by:

1. Navigating to the folder in File Explorer (e.g. C:\SandBox\1\_VSF\_Column).
2. Highlighting the path in File Explorer:

C:\SandBox\1\_VSF\_Column

3. Replacing the existing path with the string 'cmd':



4. Pressing Enter/Return.

A command prompt window rooted at the input folder should appear:

```
C:\> C:\WINDOWS\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\SandBox\1_VSF_Column>
```

When you run MUT it will try to obtain a prefix in the following order:

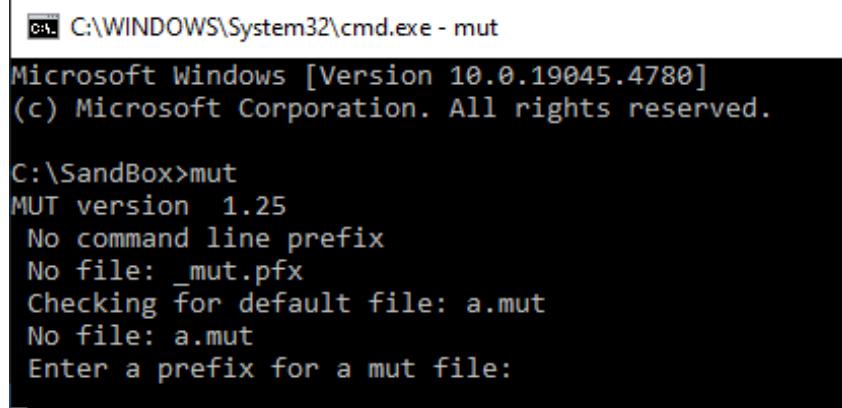
1. **From a command line argument:** At the command prompt, MUT checks for the presence of a command line argument. For example, typing this:

`mut Good`

would cause MUT to process the input file `Good.mut`.

2. **From a prefix file:** If there is no command line argument, MUT checks for the presence of the file `.mut.pfx` in the folder. If present, MUT will read the prefix from it. For example, if the mut file was called `Good.mut` then the file `.mut.pfx` would contain a single line
3. **From the default input file:** If there is no command line argument or prefix file in the folder, MUT checks for the presence of the file `a.mut`. If present in the folder, MUT will process it.

4. **From the keyboard:** If none of these methods are successful, MUT will prompt for a prefix as shown here:



```
C:\WINDOWS\System32\cmd.exe - mut
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\SandBox>mut
MUT version 1.25
No command line prefix
No file: _mut.pfx
Checking for default file: a.mut
No file: a.mut
Enter a prefix for a mut file:
```

The user would type the prefix e.g.: Good and press Enter.

To build the `1_VSF_Column` example, we would run MUT using the input file `_build.mut` by typing:

```
mut _build
```

which uses the first method to supply the prefix.

If you open the file `_build.mut` in a text editor you will see the first couple of lines are comments (which begin with an exclamation point character: '!') describing the problem:

```
! Examples\1_VSF_Column:
!   A modflow project of a 1D column generated from a simple 2d rectangular mesh
```

MUT first creates a clean copy of the input file called `_buildo.input` by removing all comment lines.

As MUT processes the input file, output is written to both the screen and to the file `_buildo.eco`. If you open `_buildo.eco`, you will see the first thing written is the MUT header, which contains the version number and build date.

These are followed by the stripped comments, which can provide a synopsis of the input file contents. The rest of the cleaned input file contains MUT instructions, which may require data in the form of numbers (e.g. parameter values) or alphanumeric strings (e.g. file names).

The first instruction in the cleaned input file begins the model build:

---

## build modflow usg

This is a *subtask* that defines the characteristics of the MODFLOW-USG<sup>Swf</sup> model including:

- Units of length and time.
- Numerical model meshes.
- Material properties.
- Boundary conditions.
- Time-stepping, stress period and output control parameters.

- Solver parameters.

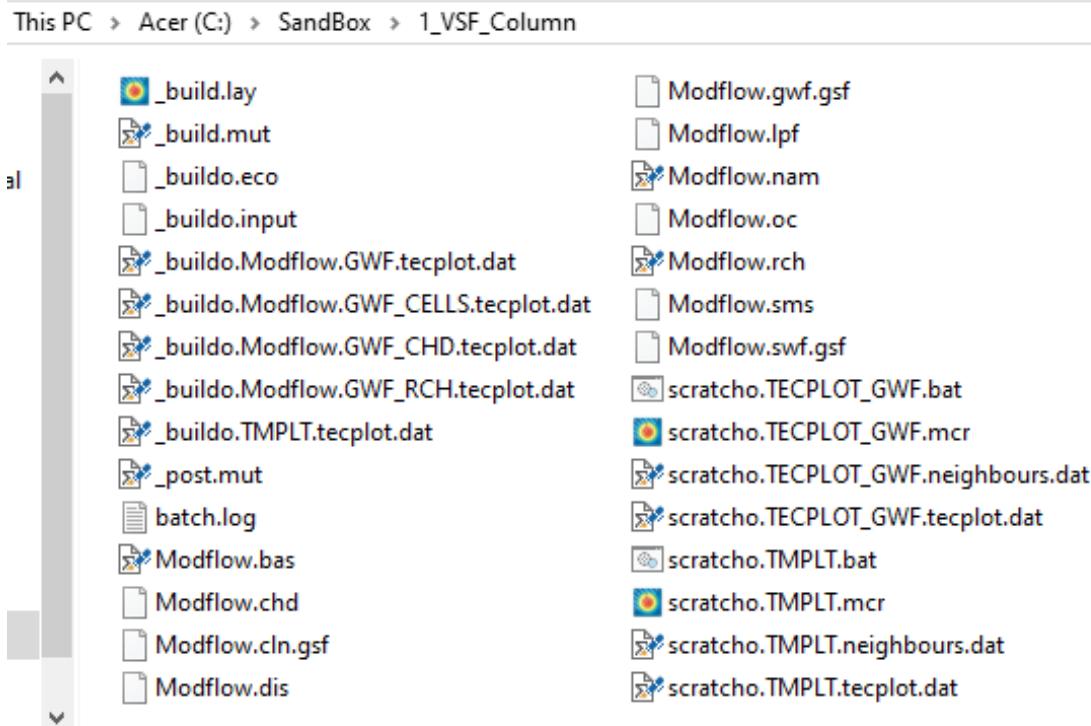
Subtasks have their own unique set of instructions, which are read and processed until an `end` instruction is encountered. We suggest appending the subtask name to the `end` instruction, which makes debugging easier, especially if subtasks are nested:

**end build modflow usg**

We will use the formatting convention shown above when documenting new instructions:

- Heavy upper and lower lines frame the instruction documentation.
- The instruction name is presented in a large sans-serif font.
- Data inputs, if required, are presented and described in a numbered list.
- General notes about instruction usage are presented.
- In the case of a subtask instruction, a suggested `end` instruction is presented. The first three non-blank characters must be the string `end`, but the rest is optional.

After MUT finishes, the working folder should look something like this:



Several new output files have been created, of which it may be noted:

- Build output files, which have the prefix `_buildo`, appear near the start of the list if sorted by name.

- TECPLOT output files are indicated by the suffix `.tecplot.dat`.
- Modflow model input files are written using the default prefix `Modflow`, (e.g. `Modflow.nam`, `Modflow.bas` etc.) The prefix can be customized if desired but there are advantages to keeping this 'generic' one, such as portability of post-processing scripts or TECPLOT layout files that follow this generic naming convention.
- Several scratch files (with prefix `scratcho`) are written. These are used for debugging during code development and can be ignored in most cases.
- MUT deletes previously generated output files and writes a fresh set each time it is run. This can prevent confusion that might arise if out-of-date output files were present.<sup>2</sup>
- If the run is successful the last line written will be `Normal exit`, otherwise an error message will be given.

### 3.1 Defining the Units of Length and Time

By default, MUT uses meters and seconds as the units of length and time respectively.

The instruction `units of length` is used to change the default value:

---

#### **units of length**

1. `$_units` The desired units of length: feet, meters or centimeters.

So, for example, to use units of centimeters instead of meters, we would put the string `centimeters` in the input file, which would then be assigned to the string variable `$_units`.

---

This instruction requires one line of input, which in this case is a variable named `$_units`. When naming input variables in the command description, the following conventions will be used:

- Alphanumeric string variable names will begin with the string '`$_`'
- Real number (i.e. containing a decimal point) variable names will begin with the string '`R_`'
- Integer number (i.e. *nota* containing decimal point) variable names will begin with the string '`I_`'

The instruction `units of time` is used to change the default value:

---

#### **units of time**

---

<sup>2</sup>For example, if we define a recharge boundary condition, MUT will create the file `prefixo.Modflow.SWF_RCH.Tecplot.dat` which shows the locations and recharge values assigned to Modflow cells. If we then removed the recharge condition from the input file, but did not delete this output file, we may assume the recharge condition still applies.

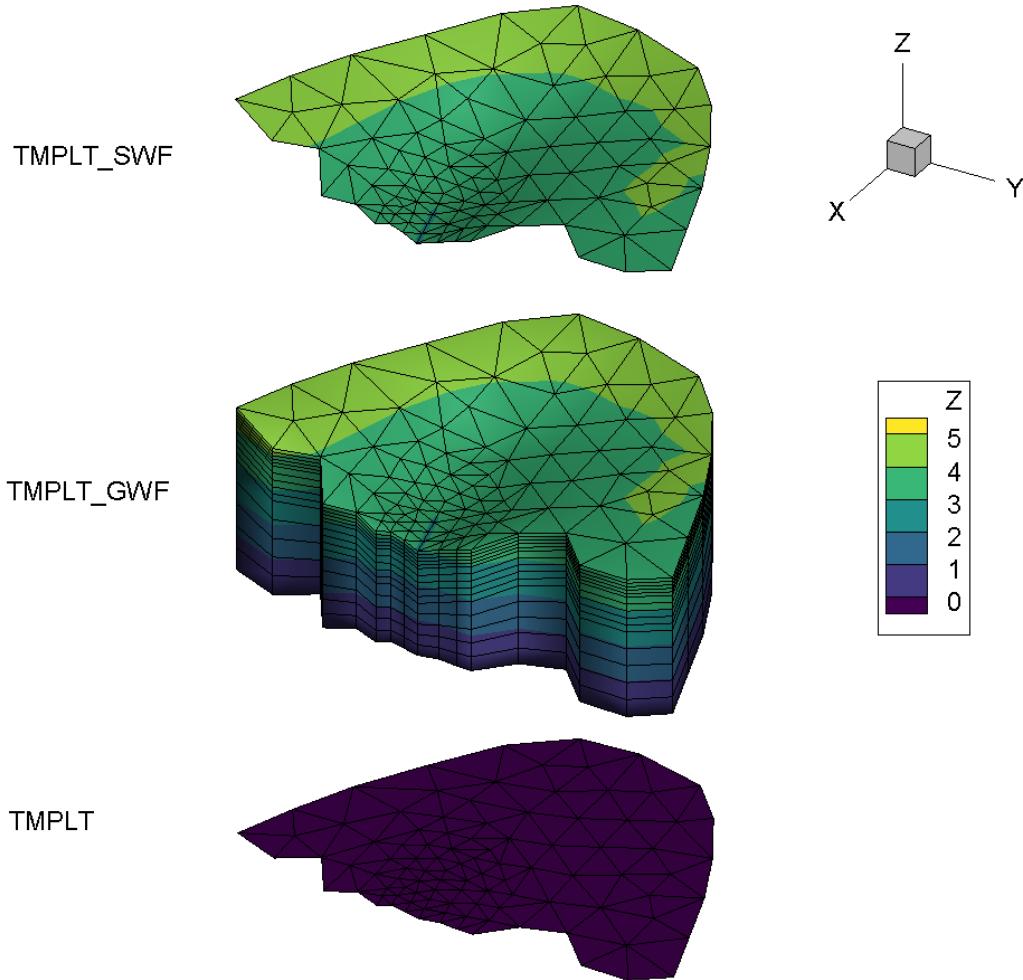
1. **`$_units`** The desired unit of time: seconds, minutes, hours, days or years.

So, for example, to use units of days instead of seconds, we would put the string `days` in the input file, which would then be assigned to the string variable `$_units`.

MUT converts the string variable `$_units` to its numeric equivalent and passes that to MODFLOW-USG<sup>*Swf*</sup> through the variables `ITMUNI` and `LENUNI`.

## 3.2 Defining the Template Mesh

The next step in the model build workflow is to define a template mesh, which is a 2D finite-element mesh that is used to generate a 3D GWF (and possibly a 2D SWF) finite-element mesh. Below is an example<sup>3</sup> showing an exploded view of a template mesh (bottom image) that was used as a basis for generating finite-element meshes for the GWF (middle image) and SWF (upper image) domains:



Some key features of this example are:

<sup>3</sup>This example was generated using the TECPLOT layout file `MUT_Examples\6_Abdul_Prism_Cell\FIG Template Abdul.lay`.

- The template mesh is assigned an elevation of zero, and only the *xy* coordinate data are used to define the other domains.
- The GWF domain has been assigned a base elevation of zero, and a variable top elevation.
- The SWF domain has been assigned the same elevation as the GWF domain i.e. they are coincident.

In this example, the template mesh was defined using these instructions:

```
2d mesh from gb
.\gb\grid
```

The instruction **2d mesh from gb**, which requires a single line of input (i.e.: `.\gb\grid`), is documented as shown here:

## 2d mesh from gb

1. **\$\_Prefix** The GRID BUILDER<sup>4</sup> dataset prefix, including the path to it.

Given **\$\_Prefix**, this instruction reads the 2D finite-element grid data and uses it to define the 2D template mesh. **\$\_Prefix** should contain a relative path to the dataset. Examples of relative paths are:

- `.\gb\grid` The MUT input folder contains a local folder **gb** with the data set prefix **grid**.
- `..\gb\grid` The parent folder to the MUT input folder contains a folder **gb** with the data set prefix **grid**.
- `C:\gb\grid` Absolute path to a drive C: with a folder **gb** with the data set prefix **grid**.  
Absolute paths are not recommended as they may lead to portability issues.

Most input instructions, including **2d mesh from gb**, do not supply length (or time) units explicitly, so MUT assumes the defined units of length or time (e.g. the default units of meters and seconds) apply to the given input data. When supplying input data *you must be careful to supply the values in the unit system defined for the model*. As a reminder, MUT will echo the assumed units to the screen and `o.eco` file, for example:

```
2d mesh from gb
Number of nodes:          1372
Number of elements:        2651
Assumed length Units:     METERS
```

When databases are used to define material parameter values (see Section 3.3.1.4), fields defining the length and time units of each record are included. In this case MUT will automatically convert the parameter values to the unit system that has been defined for the MODFLOW-USG<sup>Swf</sup> model.

---

<sup>4</sup>GRID BUILDER is a legacy 2D triangular finite-element grid generator.

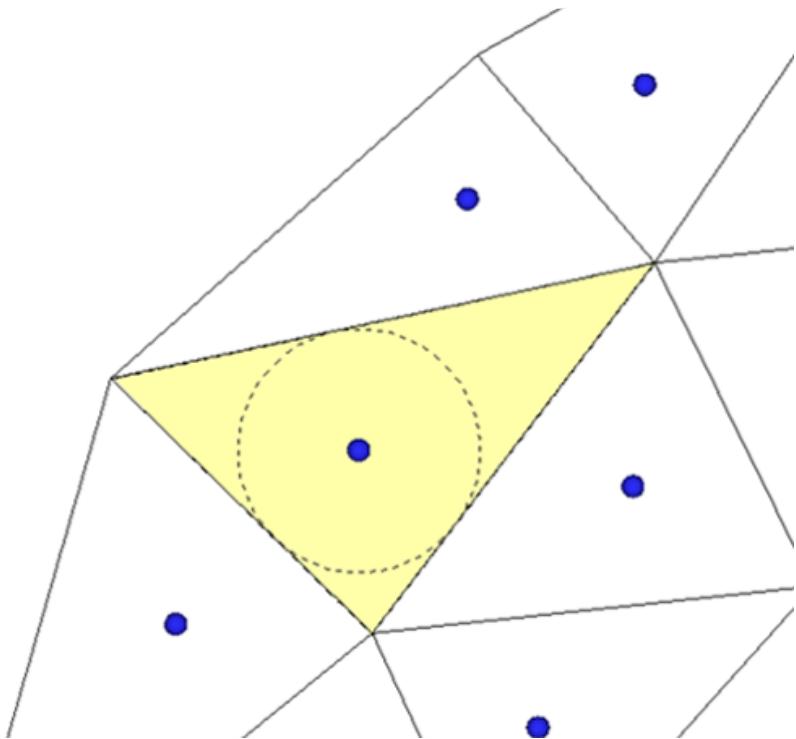
To generate uniform 2D rectangular element template meshes<sup>5</sup> use this instruction:

## generate uniform rectangles

1. **R\_xl, I\_nbz** Domain length and number of blocks in the *x*-direction
2. **R\_yl, I\_nby** Domain length and number of blocks in the *y*-direction

A 2D finite-element mesh composed of uniform rectangular elements will be generated. In this case, the grid is formed by subdividing the domain in the *x*-direction into **I\_nbz** blocks, each of length **R\_xl/I\_nbz**. The domain is subdivided in a similar fashion in the *y*-direction, using the other input parameters.

There are two ways that MODFLOW cell control volumes can be defined from the template mesh. By default, MUT uses a mesh-centred approach as shown here for a triangular-element template mesh:



Some key features to note are:

- Inner circles, which are tangent to all three element sides, are defined for each triangular element. An example is shown by the dashed circular line in the yellow-shaded element.
- The blue-filled circles show the locations of the defined MODFLOW cell control volumes.
- The vertical connection area of the cell is defined by the triangular element area (yellow-shaded triangle).
- The horizontal connection length of the cell is defined by the triangular element side length between neighbouring elements.

<sup>5</sup>See for example the verification cases `MUT_Examples\1_VSF_Column` or `MUT_Examples\6_Abdul_MODHMS`

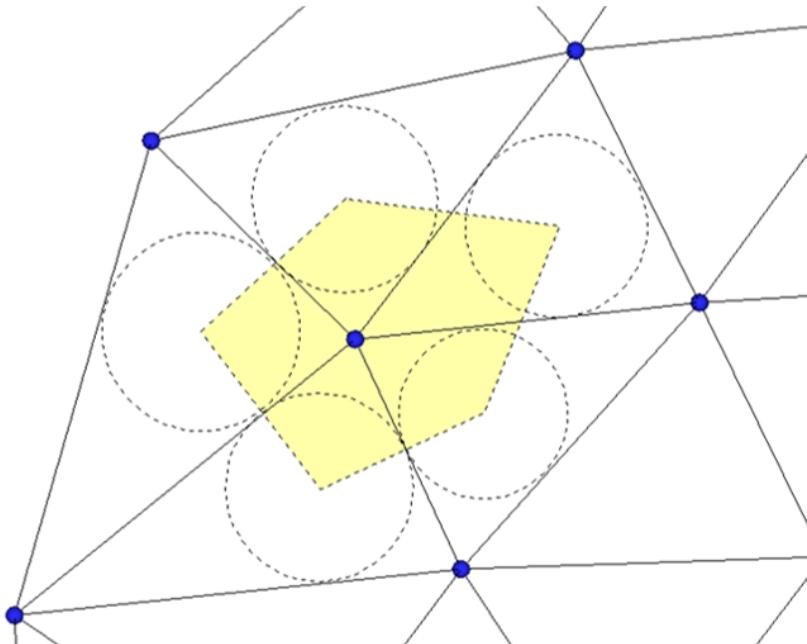
The mesh-centred approach is similar when using a rectangular-element template mesh, with the rectangular element area and side lengths defining the vertical connection area and horizontal connection length respectively.

To use a node-centred control volume approach, add this instruction *before* defining any GWF or SWF model domains:

### nodal control volumes

The node-centered approach will be used to define MODFLOW cell centres instead of the default mesh-centered approach.

The result of using a node-centred approach is shown here for a triangular-element template mesh:



Some key features to note are: .

- The blue-filled circles show that the locations of the defined MODFLOW cell control volumes are now located at template mesh node locations.
- The vertical connection area of the cell (yellow-shaded polygon) is defined by the contributing area formed by joining the inner circle centres of each element containing the template mesh node.
- The horizontal connection length of the cell is defined by the distance to a neighbouring node.

### 3.3 Groundwater Flow(GWF) Domain

Currently, every MODFLOW-USG<sup>Swf</sup> model must contain a GWF domain, which may be reduced to a single layer of very low hydraulic conductivity in cases where GWF flow and interaction with other model domains is to be neglected.

### 3.3.1 Generating a Layered GWF Domain

A MODFLOW-USG<sup>Swf</sup> 3D groundwater flow (GWF) domain can be generated from the template using this instruction:

---

#### generate layered gwf domain

This subtask has instructions that are used to define:

- Element zone numbering scheme
- Top elevation (i.e. z-coordinate)
- Mesh layers and vertical discretization

Subtask instructions will be read and processed until an `end` instruction is encountered. We suggest appending the subtask name to the `end` instruction:

---

`end generate layered gwf domain`

The construction of the 3D GWF finite-element mesh proceeds from top to bottom. First, we define the top elevation, then add layers one at a time until we reach the base of the domain. By default, element zone numbers will be assigned by layer number. If the template mesh is divided into horizontal patches with unique zone numbers, these can be assigned instead to the 3D GWF mesh using this instruction<sup>6</sup>:

---

#### Zone by template

Causes MUT to assign the template mesh element zone number to the corresponding 3D GWF element.

*This instruction should appear in the input file at the beginning of the `generate layered gwf domain` subtask before new layers are added.*

#### 3.3.1.1 Defining the Top Elevation

To assign an elevation to the top layer of template nodes use this instruction:

---

#### top elevation

This subtask defines the elevation (i.e. z-coordinate) of the top layer of nodes in the GWF finite-element template mesh in one of these ways:

- By assigning a given elevation to all nodes
- By reading variable elevation data from a file
- By interpolating elevation data from a function  $z(x)$  where the elevation  $z$  varies by the nodes  $x$  coordinate.

---

<sup>6</sup>The verification example `MUT_Examples\6_Abdul_Prism_Cell` uses this option to define SWF domain zones.

Once the elevation is defined, an `end` instruction is required to stop the subtask e.g.

## **end top elevation**

The top elevation can be defined by one of these instructions:

### **elevation constant**

1. **R\_elev** The elevation **R\_elev** will be assigned to all top layer nodes.

### **elevation from gb file**

1. **\$\_file** The elevation data in the GRID BUILDER nodal property file named **\$\_file** will be assigned to the top layer nodes.

The GRID BUILDER nodal property file uses a legacy binary file format. You can develop your own ascii input files and read them using this instruction:

### **elevation from list file**

1. **\$\_file** The elevation data in the ascii file named **\$\_file** will be assigned to the top layer nodes.

Part of a sample list file <sup>7</sup> is shown here:

```
Kriged cell top elevation for layer 1
4.414571762E+000
4.415914536E+000
...
4.415914536E+000
```

Some key features of this example are:

- The first line of the file is discarded, and in this case contains a string describing the data.
- You must supply a value for each node in the template finite-element mesh.
- The data is read in free format so there can be more than one value entered per line.
- Only the start and end of the file are shown here, with the string '...' replacing the middle portion.

<sup>7</sup>The verification example `MUT_Examples\6_Abdul_MODHMS` uses an ascii file input to define nodal elevations.

To define the top elevation as a function of  $x$  (usually used for cross-sectional models) use this instruction:

---

### elevation from xz pairs

1. **R\_x(1), R\_y(1)** First  $x, z$  coordinate pair.
2. ...
3. **R\_x(n), R\_y(n)** nth  $x, z$  coordinate pair.

An elevation is calculated for each chosen cell, based on its  $x$ -coordinate location, by interpolating an elevation from the given list of  $xz$ -coordinate pairs.

This subtask reads a list of  $xz$ -coordinate pairs until an **end** instruction is encountered e.g.

### end elevation from xz pairs

---

Here is an example showing the use of this instruction <sup>8</sup>:

```
elevation from xz pairs
    0.0, 0.0
    1000.0, 100.0
end elevation from xz pairs
```

Some key features of this example are:

- The two given  $xz$  pairs define a line that slopes from  $z = 0$  at  $x = 0$  to  $z = 100.0$  at  $x = 1000$ . You may supply as many pairs as needed to define the top of your cross-section.
- $x$  coordinates must increase continuously from the top of the list to the bottom.
- the  $x$ -range of the supplied pairs should cover the entire  $x$ -range of the template mesh.
- For each node in the template mesh, the  $x$  coordinate is used to interpolate an elevation (i.e.  $z$  value) using the appropriate  $xz$  pair.

#### 3.3.1.2 Adding Layers

*NOTE: The term layers used here should not be confused with the MODFLOW term of the same name. A MODFLOW layer is one cell thick, while a MUT layer can be one or more elements thick.*

A MODFLOW-USG<sup>Swf</sup> model must contain at least 1 layer, and each layer is defined using this instruction:

---

### new layer

This subtask adds a new layer to the GWF domain by defining the layer:

---

<sup>8</sup>The verification example `MUT_Examples\1_VSF_Hillslope` uses the `elevation from xz pairs` pairs instruction to define the top elevation of the cross-sectional domain.

- Base elevation
- Vertical discretization

It reads instructions until an `end` instruction is found e.g.

`end new layer`

The base elevation is defined using the elevation instructions described on page [24](#) that are given for the `top elevation` instruction.

By default, a new layer will be assigned the name '`Layer n`' where  $n$  is the current layer number. If you want to assign your own layer name use this instruction:

### Layer name

1. `$_layer_name` Layer name.

By default, MUT will stop and issue a warning message if the computed layer base elevation is greater than or equal to the current layer top elevation. This instruction forces the base to be below the top by a set amount:

### Minimum layer thickness

1. `R_MinThick` Minimum thickness value[L].

This instruction causes MUT to enforce a minimum thickness constraint for the current layer. At nodes where the computed layer base elevation is greater than or equal to the current top elevation, `R_MinThick` will be subtracted from the current top elevation to get the base elevation.

By default, a new layer will not be subdivided vertically unless one of the following two instructions is issued. The first creates a uniform subdivision:

### Uniform sublayering

1. `I_nsublayer` Number of sublayers.

This instruction divides the layer vertically into `I_nsublayer` elements, which will each have the same element height, equal to the top elevation minus the current base elevation divided by `I_nsublayer`.

This instruction creates a non-uniform subdivision:

### Proportional sublayering

1. **I\_nsublayer** Number of proportional sublayers.
2. **R\_sub\_thick(i),i=1,I\_nsublayer** Proportional thicknesses in order from top to bottom.

This instruction can be used if you want to refine the **GWF** domain mesh vertically, for example, in the active zone with the **SWF** domain the ground surface in the .

It is important to understand that the variable **R\_sub\_thick** is not a true thickness, but is instead a relative thickness, which is used along with the layer thickness to determine the element heights in the current column.

---

## **Proportional sublayering**

```
3
0.1
1.0
10.0
end
```

would subdivide the current layer vertically into three elements, between the current base and top elevation, with element height proportions of .1, 1 and 10 from top to bottom.

This instruction is most often used to define a layer of uniform thickness relative to an uneven top elevation:

---

## **Offset base**

1. **R\_value** Thickness value (L) by which to offset the layer base elevation.

This instruction causes the elevation of the base of the layer to be offset vertically by the given value. This can be used to create a surface a given distance below another surface.

---

## **For example, these instructions:**

```
top elevation
  elevation from list file
  elev.list
end top elevation

new layer
  uniform sublayering
  3

  elevation from list file
```

```

elev.list

offset base
-1.0

end new layer

end generate layered gwf domain

```

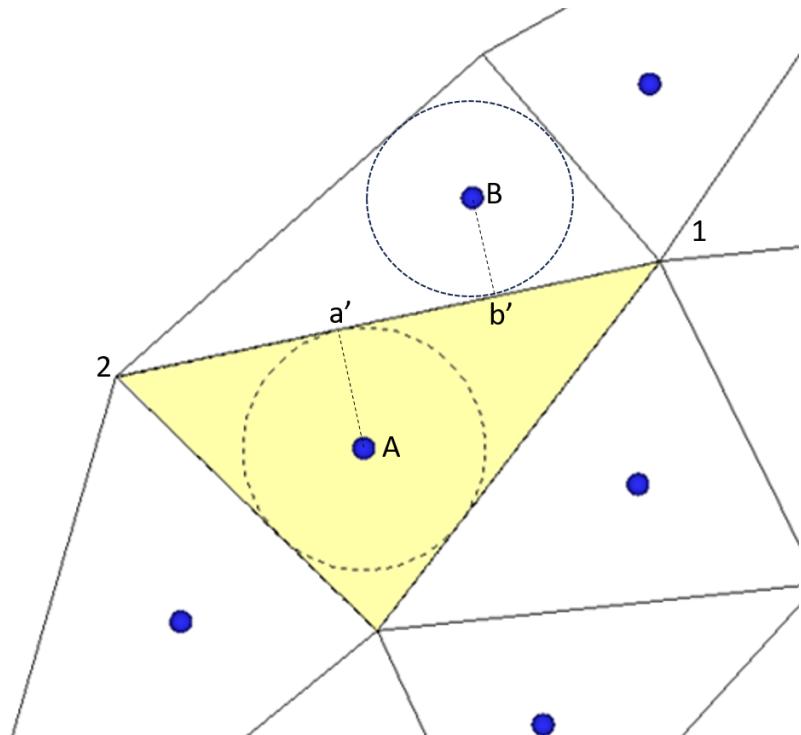
create a layer with a top elevation 1 metre below the elevation defined in the raster file `elev.list`:

### 3.3.1.3 Cell Connection Properties

Currently, MUT defines lateral connections between neighbouring cells (i.e. that share a common side) in the template 2D finite element mesh, and vertical connections between neighbouring cells (i.e. that share a common face) in the stacked 3D GWF finite-element mesh.

The GWF-GWF cell connection length (distance from cell to cell) and area (cross-sectional area perpendicular to the direction of flow) vary on a cell-by-cell basis, and depend on control volume approach (mesh- or node-centred) and element shape (triangle or rectangle).

Consider the mesh-centred case, where cell control volumes are defined at the triangular element centroid:

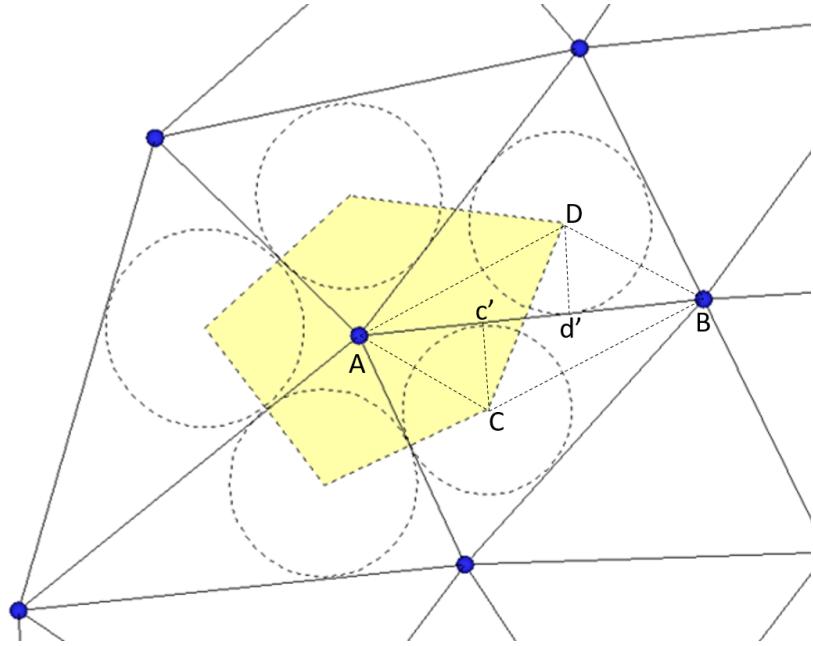


The yellow triangle shows the footprint of a MODFLOW-USG<sup>Swf</sup> cell.

Cell B is a lateral neighbour of Cell A and shares side 1-2. The lateral connection length between A and B is the sum of the lengths of the inner circle radii for each cell:  $A-a' + B-b'$ . The lateral connection area is the product of the length of the shared side 1-2 and the cell thickness.

In the vertical direction, cells above and below cell A are its neighbours. The vertical connection length is the sum of the two half-cell thicknesses. The vertical connection area equals the area of the template mesh element (yellow triangle).

Here is the node-centred case, where cell control volumes are defined at the triangular element node:



The yellow polygon shows the footprint of a MODFLOW-USG<sup>Swf</sup> cell.

Cell B is a lateral neighbour of Cell A. The lateral connection length between A and B is the distance from node A to node B. The lateral connection area is defined by the product of the sum of the inner circle radii for each triangle: C-c' + D-d' and the cell thickness.

In the vertical direction, cells above and below cell A are its neighbours. The vertical connection length is the sum of the two half-cell thicknesses. The vertical connection area is the area of the polygon formed by joining the inner circle centres of the common template mesh elements (yellow polygon).

Cell connection data are define in a similar fashion for model domains that are generated from a rectangular element template mesh. In this case, element side lengths are used to define connection lengths and areas instead of inner circle radii.

### 3.3.1.4 Assigning Material Properties

GWF domain material properties may vary on a cell-by-cell basis. MUT has instructions for selecting subsets of cells. A cell is selected when a true/false attribute (referred to as the cells **Chosen** attribute) is set to true. Selected cells can be assigned material property values. This instruction:

---

#### choose all cells

Select all cells in the active model domain.

---

selects all cells in the *active* model domain. This is an example of what we refer to as a *generic*

instruction, which means it will be applied to the currently active model domain: GWF, SWF or CLN. To choose all cells in the GWF domain, we first need to activate it using this instruction:

## active domain

1. **\$\_Domain** The name of the domain to be activated: GWF, SWF or CLN

This instruction activates the given domain named in **\$\_Domain** so that it will be used with generic instructions such as **choose all cells**.

So to activate the GWF domain, we would insert these instructions in the input file:

```
active domain  
gwf
```

These instructions can be used to choose cells in various ways:

## choose cell at xyz

1. **R\_x1, R\_y1, R\_z1** An *xyz* coordinate triplet.

The cell closest to the given *xyz* coordinate triplet will be chosen.

## choose cells by layer

1. **I\_layer** The number of the layer to be chosen.

The cells in Modflow layer number **I\_Layer** will be chosen. Remember that Modflow layers are one cell high and are numbered from the top to the bottom of the model domain.<sup>9</sup>

## choose cells from file

1. **\$\_file** The file **\$\_file** containing a list of cell numbers.

The cells listed in the file **\$\_file** will be chosen. <sup>10</sup>

<sup>9</sup> See the verification example **MUT\_Examples\1\_VSF\_Column** which uses the previous two instructions to define a constant head at the base and a recharge boundary condition at the top of the 1D column.

<sup>10</sup> See the verification example **MUT\_Examples\1\_Abdul\_MODHMS** which uses this instruction to assign some cells as inactive.

## choose cells from gb elements

1. **\$\_file** The GRID BUILDER chosen element file **\$\_file** containing information concerning the status, chosen or not chosen, of each element in the GRID BUILDER model domain.

If an element is flagged as chosen in the GRID BUILDER model domain then the corresponding cell will be chosen in the MODFLOW-USG<sup>Swf</sup> model domain. <sup>11</sup>

## choose cells from gb nodes

1. **\$\_file** The GRID BUILDER chosen node **\$\_file** containing information concerning the status, chosen or not chosen, of each node in the GRID BUILDER model domain.

If a node is flagged as chosen in the GRID BUILDER model domain then the corresponding cell will be chosen in the MODFLOW-USG<sup>Swf</sup> model domain. <sup>12</sup>

The previous two instructions are used to choose cells for the mesh-centered and node-centered approaches respectively.

Cell selection instructions are cumulative. For example, you can modify the current selection by repeating instructions like `choose cell at xyz` or `choose cells by layer` with different inputs and then assign properties to the current selection. This instruction clears the selection before beginning new cell selection(s):

### clear chosen cells

Clears the current cell selection.

It is good practice to clear the selection before starting a new selection. MUT echoes the results of the selection instructions to the screen and .eco file as shown in this example:

```
clear chosen cells
GWF Cells chosen:      0

choose all cells
GWF Cells chosen:      39765
```

If a cell selection instruction has unexpected results these are good places to check.

The recommended way to assign domain material properties is through the use of a lookup table. Lookup tables are provided for the each domain in the USERBIN directory, as outlined on page [7](#). In this case, the file `GWF.csv` contains the lookup table for the GWF domain.

<sup>11</sup> See the verification example `MUT_Examples\1_Abdul_Prism_Cell` which uses this instruction to assign some cells as inactive.

<sup>12</sup> See the verification example `MUT_Examples\1_Abdul_Prism_Cell_nc` which uses this instruction to assign some cells as inactive.

In order for MUT to access the lookup table, you first need to provide a link to this file using the instruction:

---

### gwf materials database

1. **\$\_file** GWF material properties lookup table file name.

MUT uses the file **\$\_file** to look up GWF material properties.

---

In the case of the GWF domain, the instructions:

```
gwf materials database  
GWF.csv
```

would be used to link to the lookup table.

You can now assign a set of GWF material properties to the current cell selection using this instruction:

---

### chosen cells use gwf material number

1. **I\_val** GWF material ID number.

A unique set of GWF material properties is retrieved from a lookup table, using the given material ID number **I\_val**, and assigned to the chosen cells.

---

The verification example **MUT\_Examples\1\_VSF\_Column** uses these instructions to assign units of length and time for the MODFLOW-USG<sup>Swf</sup> model and properties for material number 1 as shown here:

```
build modflow usg  
    units of length  
    centimeters  
  
    units of time  
    days  
  
    ... etc  
  
gwf materials database  
GWF.csv  
  
active domain  
gwf  
    choose all cells  
  
    chosen cells use gwf material number  
    1
```

The line '... etc' indicates a section of the input file that did not need to be shown for the purposes of this example.

The first few lines of the `GWF.csv` file are shown here:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	Material ID	Material name	Porosity	Kh (Kx)	Kv (Kz)	Ky	Specific storage	Specific Yield	Function Type	Alpha	Beta	Sr	Brooks-Corey Exponent	Length Unit	Time Unit	Notes
2	1	1D Column	0.3	10	10	0	0.000001	0.43	Van Genuchten	0.036	1.56	0.18	6.5714	CENTIMETERS	DAYS	Rob copied these p
3	2	1D Column Brooks	0.3	10	10	0	0.000001	0.43	Brooks-Corey	0.036	1.56	0.18	6.5714	CENTIMETERS	DAYS	-

Some key features to note are:

- CSV files can be loaded and examined using MICROSOFT EXCEL.
- The first line is a header with contains field (i.e. column) names.
- The first column contains the material ID number, followed by the material parameters (one per column), unit definitions and notes.
- Material 1 defines length units of centimeters and time units of days. These are the same as the defined MODFLOW-USG<sup>Swf</sup> units so no unit conversion is required in this case.

Shown below is the output echoed to the screen and `_buildo.eco` file for the verification example `MUT_Examples\1_VSF_Column`:

```

gwf materials database
Materials file C:\MUT\MUT_Examples-main\_MUT_USERBIN\GWF.csv

active domain
gwf

choose all cells
GWF Cells chosen: 100

chosen cells use gwf material number
Assigning all chosen GWF cells properties of material 1, 1D Column
Kh_Kx: 10.000 CENTIMETERS DAYS^(-1)
Kv_Kz: 10.000 CENTIMETERS DAYS^(-1)
Specific Storage: 1.00000E-07 CENTIMETERS^(-1)
Specific Yield: 0.43000 DIMENSIONLESS
Alpha: 3.60000E-02 CENTIMETERS^(-1)
Beta: 1.5600 DIMENSIONLESS
Sr: 0.18140 DIMENSIONLESS
Unsaturated Function Type: Van Genuchten

```

Some key features to note are:

- The location and name of the materials database file used are shown.
- The material name (**1D Column**) associated with material ID number 1 is shown.
- The units defined in the database for each property are given. For example, property **Kh\_Kx** has units of **CENTIMETERS DAYS<sup>-1</sup>**, where the string <sup>-1</sup> means 'raised to the power of minus 1', giving units of **CENTIMETERS/DAYS**.

Editing the **csv** files directly is not recommended. To modify or define database files, please refer to the guidelines given in Appendix A.

The following section describes instructions used to assign values to individual materials properties for the current cell selection. When using these instruction, *you must be careful to supply the values in the unit system defined for the model*. This first instruction is used to define the horizontal hydraulic conductivity, **Kh**:

---

### **gwf kh**

1. **R\_val** Horizontal hydraulic conductivity [ $L T^{-1}$ ].

A horizontal hydraulic conductivity of **R\_val** is assigned to the chosen cells.

---

As a reminder, MUT will echo the assumed units to the screen and **o.eco** file.

---



---

### **gwf kv**

1. **R\_val** Vertical hydraulic conductivity [ $L T^{-1}$ ].

A vertical hydraulic conductivity of **R\_val** is assigned to the chosen cells.

---



---

### **gwf ss**

1. **R\_val** Specific storage [ $L^{-1}$ ].

A specific storage of **R\_val** is assigned to the chosen cells.

---



---

### **gwf sy**

1. **R\_val** Specific yield (-).

A specific yield of **R\_val** is assigned to the chosen cells.

---



---

### **gwf alpha**

1. **R\_val** Van Genuchten/Brooks-Corey Alpha [ $L^{-1}$ ].

A Van Genuchten/Brooks-Corey Alpha of **R\_val** is assigned to the chosen cells.

### gwf beta

1. **R\_val** Specific yield (-).

A specific yield of **R\_val** is assigned to the chosen cells.

### gwf sr

1. **R\_val** Residual saturation [-].

A residual saturation of **R\_val** is assigned to the chosen cells.

### gwf brooks

1. **R\_val** Brooks-Corey exponent.

A Brooks-Corey exponent of **R\_val** is assigned to the chosen cells.

### choose all zones

Select all zones in the active model domain.

### choose zone number

1. **I\_value** The number of the zone to be chosen.

### clear chosen zones

Clears the current zone selection.

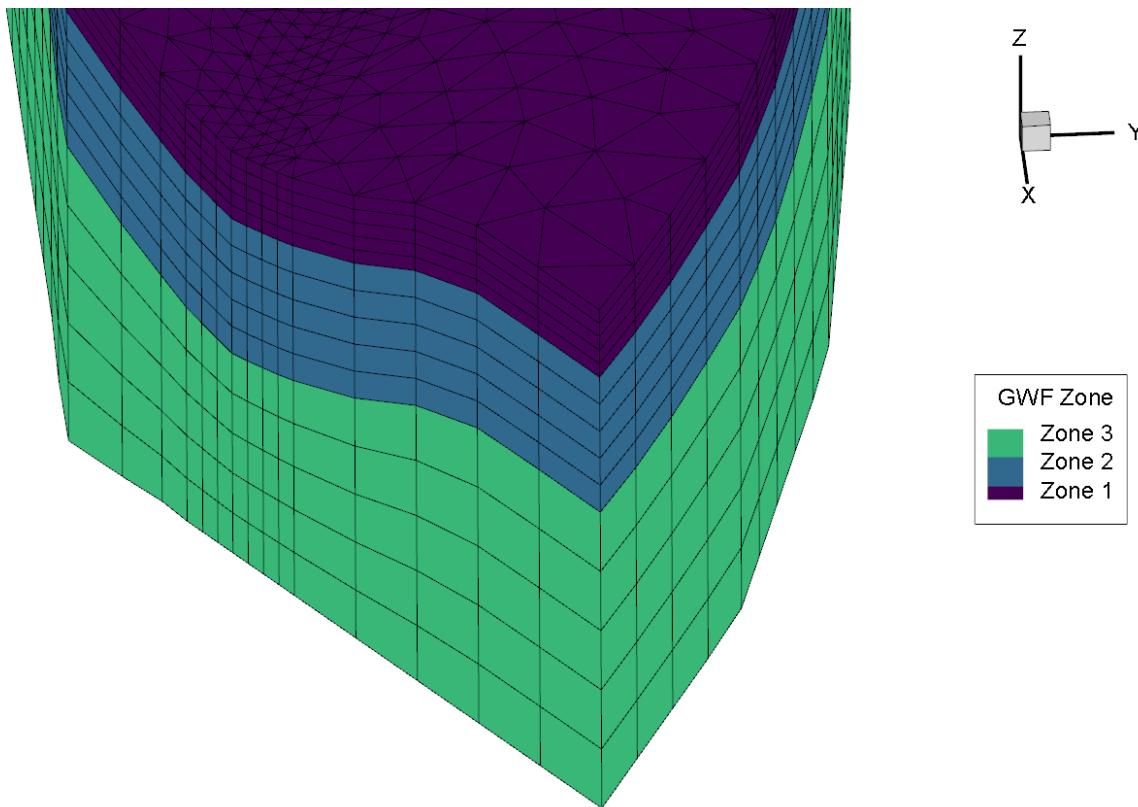
The zone selection can be converted into a cell selection using this instruction:

## choose cells by chosen zones

If a zone is currently chosen, any cell which has that zone number will be chosen.

This cell selection can now be used to assign material properties.

Below is an example<sup>13</sup> of a case in which the element zone numbers have been assigned by layer number for a 3-layer case:



Some key features to note are:

- There are 3 zones, corresponding to the layers 1 to 3.
- 'Zone 1', coloured dark blue, corresponds to layer 1. Recall that the MODFLOW-USG<sup>Swf</sup> mesh is generated from the top down, so layer 1 is at the top of the model domain.
- Each layer is composed of multiple MODFLOW layers, which are each one cell thick.

The following section from the input file shows how material properties were assigned for the 3-layer case:

```
gwf materials database  
GWF.csv
```

```
active domain
```

<sup>13</sup>See verification example MUT\_Examples\6\_Abdul\_Prism\_Cell

```
gwf
```

```
clear chosen zones
choose zone number
1
choose zone number
2
choose zone number
3

clear chosen cells
choose cells by chosen zones

chosen cells use gwf material number
5
```

The following section from the `eco` file shows zone selection output for the 3-layer case:

```
clear chosen zones
GWF Zones chosen:          0

choose zone number
Adding zone number:        1
GWF zone numbers currently chosen:
  1

choose zone number
Adding zone number:        2
GWF zone numbers currently chosen:
  1
  2

choose zone number
Adding zone number:        3
GWF zone numbers currently chosen:
  1
  2
  3

clear chosen cells
GWF Cells chosen:          0

choose cells by chosen zones
GWF Cells chosen:          39765
```

Some key features to note are:

- As we choose zone numbers, the list of currently chosen zones grows accordingly.
- The final number of GWF cells chosen is equal to the total number of cells in the domain, since we had selected all 3 layers prior to converting the zone selection to a cell selection.

Since we are assigning uniform properties to the entire model domain, this example could be simplified by simply choosing all cells then assigning properties. However, if you wanted to assign different material properties to each zone, you can easily do so by modifying the example to assign properties to one zone at a time, being careful to clear the zone and cell selections for each layer.

### 3.3.1.5 Initial Conditions

An initial (or starting) head should be assigned to each cell in the GWF domain. This could be an initial guess at the beginning of a transient stress period or a set of hydraulic heads from a previous run.

To assign a uniform hydraulic head to the GWF model domain, you must first make a cell selection as described on page [30](#), then use this instruction:

---

#### **gwf initial head**

1. **R\_value** Initial (or starting) hydraulic head [L].

An initial hydraulic head of **R\_value** is assigned to the chosen cells.

---

As a reminder, MUT will echo the assumed units to the screen and **o.echo** file:

```
gwf initial head
Assigning all chosen GWF cells starting heads of      2.7800      METERS
```

To assign a linearly varying head that is a function of  $z$  (i.e. depth or elevation), use this instruction:

---

#### **gwf initial head function of z**

1. **R\_z(1), R\_head(1)** First  $z, head$  pair.
2. **R\_z(2), R\_head(2)** Second  $z, head$  pair.
3. ...
4. **R\_z(n), R\_head(n)** nth  $z, head$  pair.

An initial head is calculated for each chosen cell, based on its  $z$ -coordinate location, by interpolating a head from the given list of  $z, head$  pairs.

This subtask reads a list of  $z, head$  pairs until an **end** instruction is encountered e.g.

**end gwf initial head function of z**

This is commonly used to generate an initial head for a simple column model<sup>14</sup> as shown here:

```
gwf initial head function of z
! z    head
 0.0  -100.0
100.0     0.0
```

Some key features of this example are:

- The two given  $z, head$  pairs define an initial head that varies from  $head = -100.0$  at  $z = 0$  to  $head = 0.0$  at  $z = 100.0$ . You may supply as many pairs as needed to define the initial head.
- $z$ -coordinates must increase continuously from the top of the list to the bottom.
- the  $z$ -range of the supplied pairs should cover the entire  $z$ -range of the model domain.
- For each node in the model domain mesh, the  $z$  coordinate is used to interpolate an initial head (i.e.  $head$  value) using the appropriate  $z, head$  pair.

### 3.3.1.6 Boundary Conditions

A constant head boundary condition fixes the head at a GWF cell at a given value, allowing water to flow into or out of the GWF model domain depending on surrounding conditions. To assign a uniform constant head to the GWF model domain use this instruction:

#### gwf constant head

1. **R\_value** Constant hydraulic head [L].

An constant hydraulic head of **R\_value** is assigned to the chosen cells.

A drain boundary condition allows water to flow out of the GWF model domain if the hydraulic head of the cell is higher than the drain elevation. To add a drain to the GWF model domain use this instruction:

#### gwf drain

1. **R\_value** Drain conductance [L/T].

A drain conductance of **R\_value** is assigned to the chosen cells. The top elevation of the cell is assigned automatically as the drain elevation

<sup>14</sup>The verification example `MUT_Examples\1_VSF_Column` uses the `gwf initial head function of z` instruction to define the initial head of the model domain

A recharge boundary condition forces water to flow in to the GWF model domain at a specified rate. To add a recharge to the GWF model domain use this instruction:

---

### gwf recharge

1. **R\_value** Recharge rate [L/T].
2. **I\_option** Recharge option.

A recharge rate of **R\_value** is assigned to the chosen cells.

The recharge option **I\_option** is used to define where the recharge is to be applied and can have one of the following values:

1. To top layer
2. To one specified node in each vertical column
3. To highest active node in each vertical column
4. To the swf domain on top of each vertical column

---

## 3.4 Surface Water Flow(SWF) Domain

The SWF domain is a 2D network of cells which is usually, but not necessarily, coincident with the top of the GWF domain.

MODFLOW-USG<sup>Swf</sup> allows individual (i.e. GWF, SWF and CLN) processes to add to the global conductance matrix in order to represent fluxes between cells within a process as well as with cells of other processes. MODFLOW-USG<sup>Swf</sup> provides a framework for tightly coupling multiple hydrologic processes. The tight coupling, in contrast to a sequential or iterative coupling approach, occurs through the formulation of a global conductance matrix that includes the cells for all processes.

The flows between SWF cells are governed by the diffusion-wave equations, which ultimately provide a pressure head (i.e. surface water depth) at each cell.

### 3.4.1 Generating a SWF Domain

A SWF domain can be easily added to the MODFLOW-USG<sup>Swf</sup> model using the same template mesh that was used to define the GWF mesh, as described in Section 3.2.

The SWF domain is generated using this instruction:

---

### generate swf domain

This subtask currently has only one instruction that is used to define:

- Top elevation (i.e.  $z$ -coordinate, ground surface elevation)

It reads instructions until an **end** instruction is found e.g.

**end generate swf domain**

---

Here, the **top elevation** instruction has the same options as described in Section 3.3.1.1 for the GWF domain.

Currently, the element zone numbering for the SWF domain is determined by the instruction used to generate the template mesh:

**2d mesh from gb** The GRID BUILDER element area numbers are used to define the MODFLOW-USG<sup>Swf</sup> element zone numbers.

**generate uniform rectangles** The element zone numbers default to 1.

### 3.4.1.1 Cell Connection Properties

The SWF-SWF lateral cell connection lengths and areas are defined in the same way as they are for the GWF domain (see Section 3.3.1.3), except lateral connection area uses surface water depth instead of cell thickness.

SWF-GWF cell connection properties vary on a cell-by-cell basis:

- By default, the SWF-GWF connection length is assigned a value of 0.001.
- The vertical SWF-GWF connection area is defined in the same way as it is for vertical GWF-GWF cell connections (see Section 3.3.1.3).

You can use this instruction to change the SWF-GWF vertical cell connection length after making a cell selection (see the instructions described on page 30 for the GWF domain):

---

**swf to gwf connection length**

1. **R\_value** SWF cell to GWF cell connection length [ $L$ ].

A SWF cell to GWF cell connection length **R\_value** is assigned to the chosen cells.

---

### 3.4.1.2 Assigning Material Properties

Unlike the GWF domain, SWF domain material properties vary on a zone-by-zone basis, which means assigning material property values are done using zone selections instead of cell selections.

Prior to making cell or zone selections and assigning properties, we need to activate the SWF domain using these instructions:

```
active domain  
swf
```

Zone selections must first be made using the instructions described on page [35](#) for the GWF domain, then these instructions can be used to assign material properties to the current zone selection:

### swf manning

1. **R\_value** Manning's coefficient of friction [ $L^{-1/3}T$ ].

A Manning's coefficient of **R\_value** is assigned to the chosen cells.

### swf depression storage height

1. **R\_value** Depression storage height [ $L$ ].

A depression storage height of **R\_value** is assigned to the chosen zones.

### swf obstruction storage height

1. **R\_value** Obstruction storage height [ $L$ ].

An obstruction storage height of **R\_value** is assigned to the chosen zones.

### swf depth for smoothing

1. **R\_value1** Depth for smoothing height 1 [ $L$ ].
2. **R\_value2** Depth for smoothing height 2 [ $L$ ].

Two depth for smoothing heights are read in **R\_value1** and **R\_value2** and assigned to the chosen zones.

A lookup table of SWF material properties is provided in the file `qrySWFMaterials.txt`, located in the `USERBIN` directory as outlined on page [7](#).

In order for MUT to access the lookup table, you first need to provide a link to this file using the instruction:

### swf materials database

1. **\$\_file** SWF material properties lookup table file name.

MUT uses the file **\$\_file** to look up SWF material properties.

You can now assign a full set of SWF material properties to the current zone selection, as described on page [35](#), using this instruction:

### chosen zones use swf material number

1. **I\_val** SWF material ID number.

A unique set of SWF material properties is retrieved from a lookup table, using the given material ID number **I\_val**, and assigned to the chosen zones.

The assigned SWF material properties are written to the screen and .eco file:

```
chosen zones use swf material number
Assigning all chosen SWF zones properties of material 1.0000, Streambed
Manning's Coefficient:          3.00000E-02
Depression Storage Height:     0.10000
Obstruction Storage Height:    0.00000
SWF Smoothing Depth 1:        1.00000E-06
SWF Smoothing Depth 2:        1.00000E-06
```

You can find detailed information about how to use MICROSOFT EXCEL to modify or define your own lookup tables in Appendix [A](#).

#### 3.4.1.3 Initial Conditions

An initial (or starting) head should be assigned to each cell in the SWF domain. This could be an initial guess at the beginning of a transient stress period or a set of hydraulic heads from a previous run.

To assign an initial head to the SWF model domain, you must first make a cell selection as described on page [30](#), then this instruction can be used to calculate an initial (or starting) head for the flow solution given an initial surface water depth:

### swf initial depth

1. **R\_value** Initial depth [L].

An initial depth of **R\_value** is used to calculate an initial head at each of the chosen cells.

### 3.4.1.4 Boundary Conditions

A constant head boundary condition fixes the head at a SWF cell at a given value, allowing water to flow into or out of the SWF model domain depending on surrounding conditions. To assign a uniform constant head to the SWF model domain use this instruction:

#### swf constant head

1. **R\_value** Constant hydraulic head [L].

An constant hydraulic head of **R\_value** is assigned to the chosen cells.

A recharge boundary condition forces water to flow in to the SWF model domain at a specified rate. To add recharge to the SWF model domain use this instruction:

#### swf recharge

1. **R\_value** Recharge rate [L/T].
2. **I\_option** Recharge option.

A recharge rate of **R\_value** is assigned to the chosen cells.

The recharge option **I\_option** is used to define where the recharge is to be applied and in this case should be set to a value of 4, which applies the recharge to the SWF domain.

A critical depth boundary condition assigned to a SWF cell allows water to flow out of the SWF model domain at a rate that depends on the surface water depth and a contributing length (i.e. representing the length of the cell side over which the outflow occurs).

Cell selections must first be made then one of the following two instructions can be used to assign a critical depth outflow boundary condition to the SWF model domain:

#### swf critical depth with sidelength1

A critical depth outflow boundary condition is assigned to the chosen cells.

It is assumed that an accurate estimate of the contributing length of a cell can be based on a template element side length. In this case we have arbitrarily used side 1.

#### swf critical depth

A critical depth outflow boundary condition is assigned to the chosen cells.

The contributing length of the cell outflow boundary is calculated from the SWF mesh outer boundary nodes, with each outer boundary node connected to a chosen cell contributing a half-element side length in both directions along the outer boundary.

Although `swf critical depth` is less convenient than `swf critical depth with sidelength1`, it does calculate a contributing length that matches the actual length along the SWF mesh outer boundary.

The verification example `MUT_Examples\6_Abdul_Prism_Cell` uses the second approach to define the critical depth outflow boundary condition:

```
clear chosen nodes
choose gb nodes
./gb/grid.nchos.Outer boundary nodes
flag chosen nodes as outer boundary

clear chosen cells
clear chosen nodes
choose cells from gb elements
./gb/grid.echoes.Critical depth outlet
swf critical depth
```

Some key features of this example are:

- SWF *nodes* are chosen and flagged to be on the outer boundary with the instruction `flag chosen nodes as outer boundary`.
- SWF *cells* are chosen using a GRID BUILDER chosen *elements* file with the instruction `choose cells from gb elements`. Because this example was generated using the mesh-centred control volume approach, there is a 1-to-1 correspondence between template mesh elements and SWF cells.

In the example `MUT_Examples\6_Abdul_Prism_Cell_nc`, the node-centred control volume approach is used and the instruction `choose cells from gb nodes` is used instead, because in this case there is a 1-to-1 correspondence between template mesh *nodes* and SWF cells.

The SWF and GWF meshes that MUT generates inherit node and element information from the template mesh. Currently, you can define node selections using these instructions:

---

### choose all nodes

Select all nodes in the active model domain.

---

---

### choose node at xyz

1. **R\_x1, R\_y1, R\_z1** An *xyz* coordinate triplet.

The node closest to the given *xyz* coordinate triplet will be chosen.

---

---

### choose gb nodes

1. **\$\_file** The GRID BUILDER chosen node **\$\_file** containing information concerning the status, chosen or not chosen, of each node in the GRID BUILDER model domain.

If a node is flagged as chosen in the GRID BUILDER model domain then the corresponding node will be chosen in the MODFLOW-USG<sup>Swf</sup> model domain.

---

### clear chosen nodes

Clears the current node selection.

---

## 3.5 Connected Linear Network(**CLN**) Domain

A CLN domain is a quasi-3D network of modflow cells which are each defined by individual line segments. The flows between CLN cells are governed by either open- or closed-channel flow equations, depending on surrounding conditions, which ultimately provide a pressure head or water depth at each cell.

MUT adds the CLN process equations to the global conductance matrix in order to represent fluxes between cells within the process as well as with cells of other processes.

The current version of MUT has the following limitations for the definition of the CLN domain:

1. General CLN domains, where the cell geometry is independent of the GWF mesh and cell-to-cell connection can be one-to-many or many-to-one are not yet implemented. MUT assumes a 1-to-1 connection exists between CLN cells and GWF layer 1 cells (i.e. top layer).
2. CLN flows to the SWF domain are not yet implemented, just CLN flows to the GWF domain.
3. No CLN domain boundary conditions have been implemented.

This is sufficient for the short-term purpose of solving the verification example `MUT_Examples\3_1_CLN_for_SWF`, which compares the use of a CLN versus an SWF domain for simulating flow in a surface water domain coupled to a GWF domain, but not for solving more general problems of interest.

### 3.5.1 Generating a CLN Domain

A CLN domain can be added to the MODFLOW-USG<sup>Swf</sup> model using this instruction:

---

#### generate cln domain

This subtask is currently limited to defining the CLN domain from a single pair of *xyz* coordinates and a specified number of cells.

It reads instructions until an end instruction is found e.g.

**end generate cln domain**

This instruction can be used to define a simple CLN domain:

### cln from xyz pair

1. **R\_x1, R\_y1, R\_z1** First *xyz* coordinate triplet.
2. **R\_x2, R\_y2, R\_z2** Second *xyz* coordinate triplet.
3. **I\_nCells** Number of cells in the CLN.

The 2 given *xyz* coordinates define the endpoints of a line defining the CLN. The CLN is subdivided into **I\_nCells** individual cells.

In the verification example `MUT_Examples\3_1_CLN_for_SWF`, the inputs are defined so the CLN cells coincide exactly with the top layer of GWF cells as shown below:

```
generate uniform rectangles
101.0, 101    ! Mesh length in X-direction and number of rectangular elements
1.0, 1        ! Mesh length in Y-direction and number of rectangular elements

generate layered gwf domain

    top elevation
        elevation from xz pairs
            0.0, 2.0
            101.0, 1.0
        end elevation from xz pairs
    end top elevation

    ...

generate cln domain
    cln from xyz pair
        0.0      0.5      2.0
        101.0    0.5      1.0
        101    ! number of new CLN cells

end generate cln domain
```

Some key features of this example are:

- A template mesh of length 101.0 and with 101 elements is used to define the GWF domain.
- The top of the GWF domain slopes from  $z = 2.0$  at  $x = 0.0$  to  $z = 1.0$  at  $x = 101.0$ .

- Because CLN domains are not necessarily dependent on GWF meshes, it does not use the template mesh, but instead generates a CLN domain using the `cln from xyz pair` instruction. The instruction is set up to generate 101 CLN cells that also slope from  $z = 2.0$  at  $x = 0.0$  to  $z = 1.0$  at  $x = 101.0$

### 3.5.1.1 Cell Connection Properties

CLN domain cells are currently limited to have a direct one-to-one correspondence with cells in the top layer of the GWF domain, as is the case for the simple verification example `3_1_CLN_for_SWF`. A more general approach, in which CLN cells do not have to conform to the GWF or SWF meshes will be developed in subsequent version of MUT.

CLN-CLN cells are connected through neighbouring nodes and connection properties are defined by the channel geometry inputs.

CLN-GWF cell vertical connections are defined by element areas.

### 3.5.1.2 Assigning Material Properties

CLN domain material properties vary on a zone-by-zone basis. In the current version of MUT the assignment of CLN material properties is very rudimentary.

Prior to assigning properties, we need to activate the CLN domain using these instructions:

```
active domain
cln
```

A lookup table of CLN material properties is provided in the file `qryCLNMaterials.txt`, located in the `USERBIN` directory as outlined on page [7](#).

In order for MUT to access the lookup table, you first need to provide a link to this file using the instruction:

---

#### cln materials database

- `$_file` CLN material properties lookup table file name.

MUT uses the file `$_file` to look up CLN material properties.

---

Zone selections must first be made using the instructions described on page [35](#) for the GWF domain.

You can now assign a full set of CLN material properties to the current zone selection, as described on page [35](#), using this instruction:

---

#### chosen zones use cln material number

- `I_val` CLN material ID number.

A unique set of CLN material properties is retrieved from a lookup table, using the given material ID number **I\_val**, and assigned to the chosen zones.

The assigned CLN material properties are written to the screen and .eco file:

```
chosen zones use cln material number
Assigning all chosen CLN zones properties of material           3, CLN_for_SWF
Geometry:          Rectangular
Rectangular Width:    1.0000
Rectangular Height:   1.0000
Direction:          Horizontal
Flow Treatment:      Unconfined/Mannings
Longitudinal K:      5.48000E-02
```

You can find detailed information about how to use MICROSOFT ACCESS to modify or define your own lookup tables in Tutorial ??.

### 3.5.1.3 Initial Conditions

An initial (or starting) head should be assigned to each cell in the CLN domain. This could be an initial guess at the beginning of a transient stress period or a set of hydraulic heads from a previous run.

To assign an initial head to the CLN model domain, you must first make a cell selection as described on page [30](#), then this instruction can be used to calculate an initial (or starting) head for the flow solution given an initial water depth:

#### cln initial depth

1. **R\_value** Initial depth [L].

An initial depth of **R\_value** is used to calculate an initial head at each of the chosen cells.

### 3.5.1.4 Boundary Conditions

No CLN domain boundary conditions have been implemented in the current version of MUT.

In the verification example `MUT_Examples\3_1_CLN_for_SWF`, all boundary conditions are applied to the GWF domain.

## 3.6 Stress Periods

A MODFLOW-USG<sup>Swf</sup> simulation can be broken up into separate periods of time called "stress periods". Boundary conditions can be defined at the beginning of each stress period and changed in subsequent stress periods.

At least one stress period must be defined using this instruction:

---

## stress period

This subtask has several instructions that can be used to define the duration, type and timestepping parameters of the stress period.

It reads instructions until an **end** instruction is found e.g.

---

## end stress period

---

These instructions can be used to define the stress period parameters:

---

### type

1. **\$\_type** Stress period type.

The stress period type is defined by the string **\$\_type**. It can be one of the following:

- **SS** A steady-state stress period in which the simulation is carried out until it reaches a state of equilibrium with the defined boundary conditions.
- **TR** A transient stress period in which the simulation is carried out for a specified duration with the defined boundary conditions.

---

### duration

1. **R\_value** Stress period duration [T].

The stress period duration, is defined by the string **R\_value**. It should be entered using the correct units of time as outlined in Section [3.1](#).

---

### number of timesteps

1. **I\_value** Number of timesteps to be used for this stress period.

You can change the default starting time step size of  $1 \times 10^{-3}$  seconds with this instruction:

---

### deltat

1. **R\_value** Starting time step size [T].

The starting time step size used for the stress period is defined by the string **R\_value**. It should be entered using the correct units of time as outlined in Section [3.1](#).

You can change the default minimum time step size of  $1 \times 10^{-5}$  seconds with this instruction:

### tminat

1. **R\_value** Minimum time step size [T].

The minimum time step size to allow for the stress period is defined by the string **R\_value**. It should be entered using the correct units of time as outlined in Section [3.1](#).

You can change the default maximum time step size of 60.0 seconds with this instruction:

### tmaxat

1. **R\_value** Maximum time step size [T].

The maximum time step size to allow for the stress period is defined by the string **R\_value**. It should be entered using the correct units of time as outlined in Section [3.1](#).

You can change the default multiplier for time step size of 1.1 with this instruction:

### tadjat

1. **R\_value** Multiplier for time step size [T].

The multiplier for adjusting time step size when using adaptive time-stepping is defined by the string **R\_value**.

You can change the default divider for time step size of 2.0 with this instruction:

### tcutat

1. **R\_value** Divider for time step size [T].

The divider for adjusting time step size when using adaptive time-stepping is defined by the string **R\_value**.

To add more stress periods, repeat the **stress period** subtask instructions and boundary condition definitions as many times as required. Stress periods are numbered automatically as they are added.

Here is an example which could be used to define two stress periods:

```

! stress period 1
stress period
    type
    TR

    duration
    3000.0d0
end stress period

active domain
swf
    choose all cells
    swf recharge
    5.56d-6
    4

    clear chosen nodes
    choose cell at xyz
    0.0 0.0 0.0
    swf constant head
    1.0

! stress period 2
stress period
    type
    TR

    duration
    3000.0d0
end stress period

active domain
swf
    choose all cells
    swf recharge
    0.0d0
    4

```

Some key features of this example are:

- Both stress periods are transient (type TR) with a duration of 3000.
- The recharge applied to the SWF domain (recharge option 4) is 5.5e-6 for the first stress period, then is reduced to 0.0 in the second stress period.
- The constant head applied to the SWF domain in stress period 1 is maintained for the entire simulation. By default, a boundary condition is maintained through subsequent stress periods unless it is redefined.

- Any boundary conditions given after an `end stress period` instruction apply to that stress period until another `stress period` instruction is encountered.

## 3.7 Output Control

This instruction can be used to generate a MODFLOW-USG<sup>*Swf*</sup> output control file:

---

### generate output control file

1. `R_t(1)` First output time [T].
2. ...
3. `R_t(n)` nth output time [T].

This subtask reads a list of output times `R_t()` until an `end` instruction is encountered e.g.

---

### end generate output control file

---

The verification example `MUT_Examples\1_Abdul_prism_cell` generates an output control file with 10 output times using these instructions:

```
! -----Output Control
generate output control file
 1e-4
 60.
 300.0
 600.0
 900.0
 1200.0
 1500.0
 3000.0
 4500.0
 6000.0
end generate output control file
```

The output control file looks like this:

```
# MODFLOW-USG OC file written by Modflow-User-Tools version 1.28
ATSA NPTIMES    10
 9.99999747378752E-005    60.00000000000000    300.000000000000
 600.000000000000    900.000000000000    1200.000000000000
 1500.000000000000    3000.000000000000    4500.000000000000
 6000.000000000000
HEAD SAVE UNIT    114
HEAD PRINT FORMAT 0
```

```

DRAWDOWN SAVE UNIT    115
DRAWDOWN PRINT FORMAT 0
PERIOD      1
  DELTAT     1.0000E-03
  TMINAT     1.0000E-05
  TMAXAT     60.00
  TADJAT     1.100
  TCUTAT     2.000
    SAVE HEAD
    PRINT HEAD
    SAVE DRAWDOWN
    SAVE BUDGET
    PRINT BUDGET
PERIOD      2
  DELTAT     1.0000E-03
  TMINAT     1.0000E-05
  TMAXAT     60.00
  TADJAT     1.100
  TCUTAT     2.000
    SAVE HEAD
    PRINT HEAD
    SAVE DRAWDOWN
    SAVE BUDGET
    PRINT BUDGET

```

Some key features of this example are:

- MUT automatically inserts the adaptive time-stepping option (ATSA) in the file, defines the number of print times in the simulation (NPTIMES 10) and the list of print (i.e. output) times.
- Two stress periods were defined and the listed parameters are using the default values.

## 3.8 Solver Parameters

A lookup table of MODFLOW-USG<sup>Swf</sup> solver parameters is provided in the file `qrySMS.txt`, located in the `USERBIN` directory as outlined on page [7](#).

In order for MUT to access the lookup table, you first need to provide a link to this file using the instruction:

---

**sms database**

1. `$_file` Solver parameters lookup table file name.

MUT uses the file `$_file` to look up the solver parameter values.

You can now assign the full set of solver parameters using this instruction:

## sms parameter set number

1. **I\_val** Solver parameter set ID number.

A unique set of solver parameter values is retrieved from a lookup table, using the given parameter set ID number **I\_val**.

You can find detailed information about how to use MICROSOFT ACCESS to modify or define your own lookup tables in Tutorial ??.

Currently, all of the verification examples use default solver parameters, which are defined in the input file as shown in this example:

```
sms database
qrySMS.txt
sms parameter set number
1
```

The solver parameter values are written to the screen and .eco file:

```
sms parameter set number
Using SMS parameter set 1, Default
OUTER ITERATION CONVERGENCE CRITERION (HCLOSE) 1.00000E-03
INNER ITERATION CONVERGENCE CRITERION (HICLOSE) 1.00000E-05
MAXIMUM NUMBER OF OUTER ITERATIONS (MXITER) 250
MAXIMUM NUMBER OF INNER ITERATIONS (ITER1) 600
SOLVER PRINTOUT INDEX (IPRSMS) 1
NONLINEAR ITERATION METHOD (NONLINMETH) 1
LINEAR SOLUTION METHOD (LINMETH) 1
D-B-D WEIGHT REDUCTION FACTOR (THETA) 0.70000
D-B-D WEIGHT INCREASE INCREMENT (KAPPA) 0.10000
D-B-D PREVIOUS HISTORY FACTOR (GAMMA) 0.10000
MOMENTUM TERM (AMOMENTUM) 0.0000
MAXIMUM NUMBER OF BACKTRACKS (NUMTRACK) 200
BACKTRACKING TOLERANCE FACTOR (BTOL) 1.0000
BACKTRACKING REDUCTION FACTOR (BREDUC) 0.20000
BACKTRACKING RESIDUAL LIMIT (RES_LIM) 1.0000
TRUNCATED NEWTON FLAG (ITRUNCNEWTON) 0
Options SOLVEACTIVE DAMPBOT
ACCELERATION METHOD (IACL) 1
EQUATION ORDERING FLAG (NORDER) 0
LEVEL OF FILL (LEVEL) 7
```

MAXIMUM NUMBER OF ORTHOGONALIZATIONS	(NORTH)	14
INDEX FOR USING REDUCED SYSTEM	(IREDSYS)	0
RESIDUAL REDUCTION CONVERGE CRITERION	(RRCTOL)	0.0000
INDEX FOR USING DROP TOLERANCE	(IDROPTOL)	1
DROP TOLERANCE VALUE	(EPSRN)	1.00000E-03

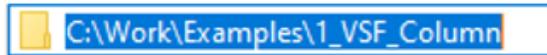
## 3.9 3D Model Build Visualization

The current version of MUT writes Tecplot-compatible output files that can be visualized to check the following attributes for GWF, SWF and CLN model domains:

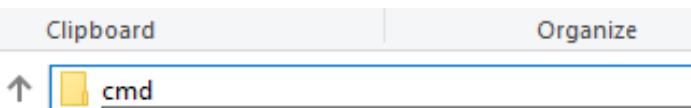
- Finite-element mesh and MODFLOW-USG<sup>Swf</sup> cell locations derived from it
- Material properties
- Initial conditions
- Boundary conditions (if specified)

A TECPLOT layout file, `_build.lay`, has been created for each verification example and provides a quick way to view the results of the model build. We will demonstrate some basic concepts using the verification example `1_VSF_Column`, which has a GWF domain defined by a simple 1D column mesh with boundary conditions assigned to the top and bottom cells.

To load `_build.lay` in TECPLOT first navigate to the folder in File Explorer (e.g. `C:\Work\Examples\1_VSF_Column`) then highlight the path in File Explorer:



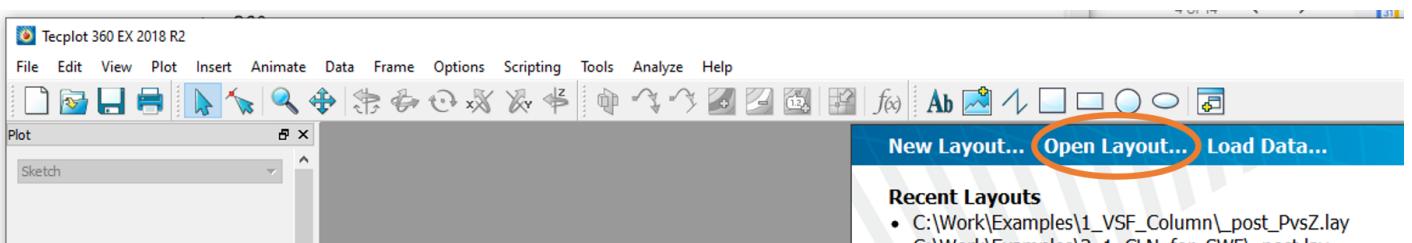
Replace the existing path with the string 'cmd':



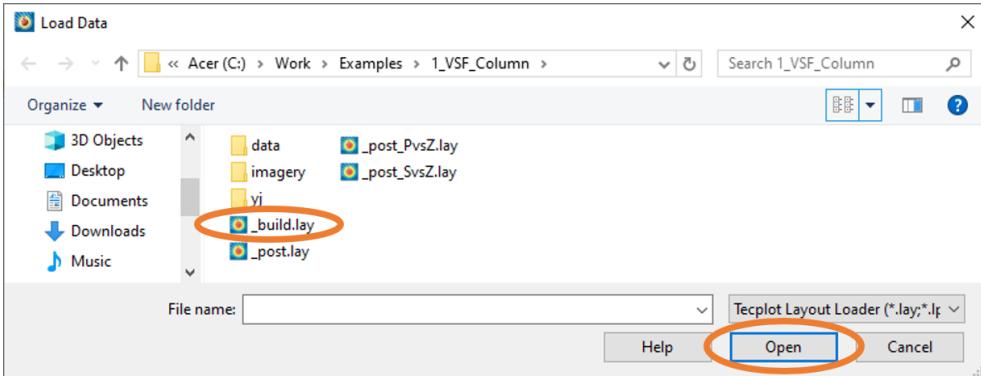
Press Enter/Return. A command prompt window rooted at the input folder should appear. To start TECPLOT type:

```
tec360
```

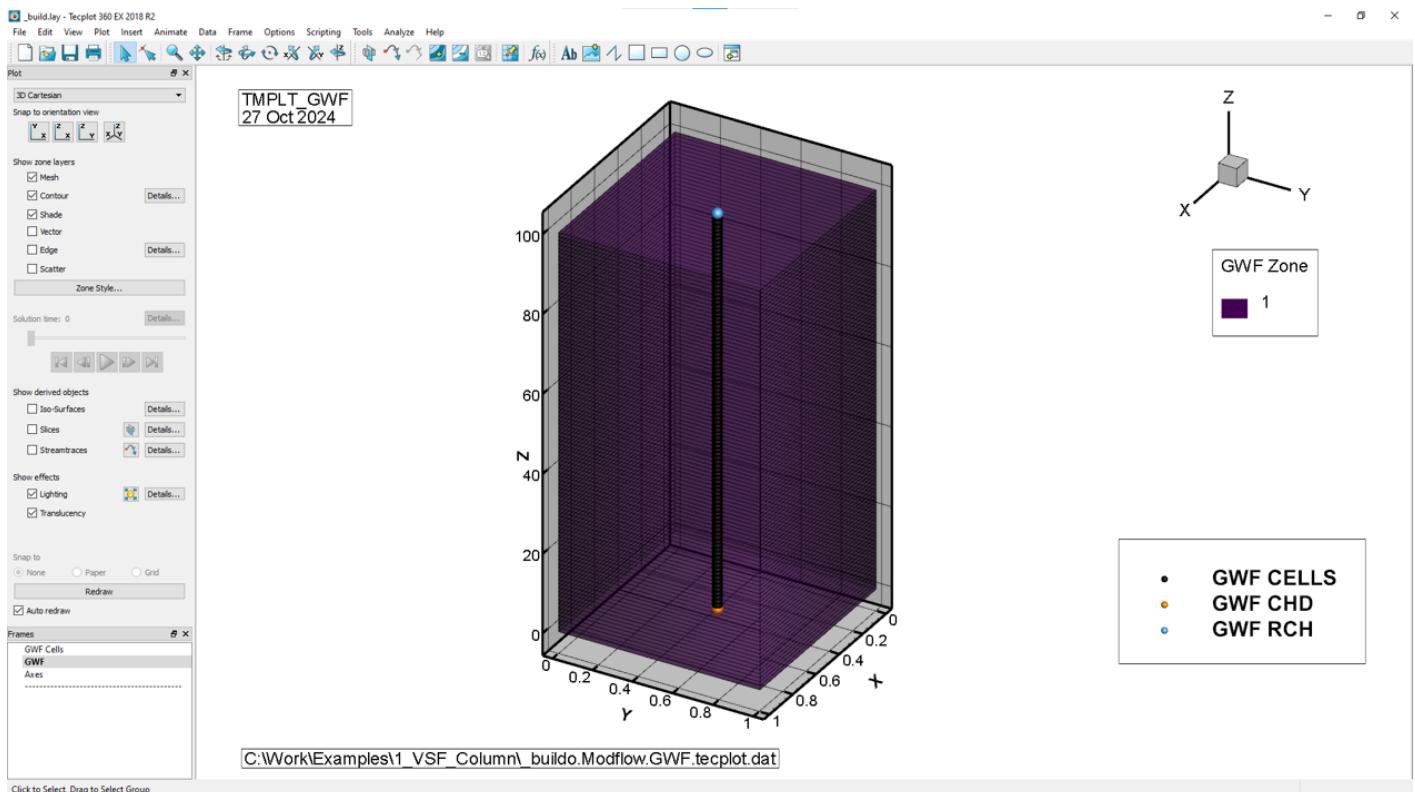
Choose 'Open Layout' to open a file selection dialogue:



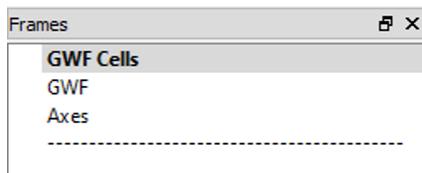
Select and open the file `_build.lay`:



You should now see the following 3D visualization of the example:



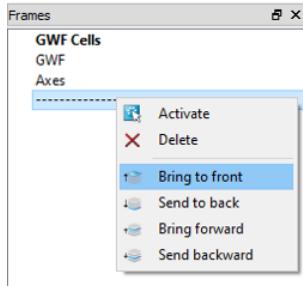
There are 4 Tecplot 'frames' that make up this image. Each frame can house it's own data for plotting, and have unique settings for visualization. The 'Frames' window at the bottom left corner shows the frame names and plotting order:



The frame at the front, called **GWF Cells**, is at the top of the list, and the bold font indicates it is the currently active frame.

The frame at the bottom, indicated by the dashed line, is a special frame we will refer to as the

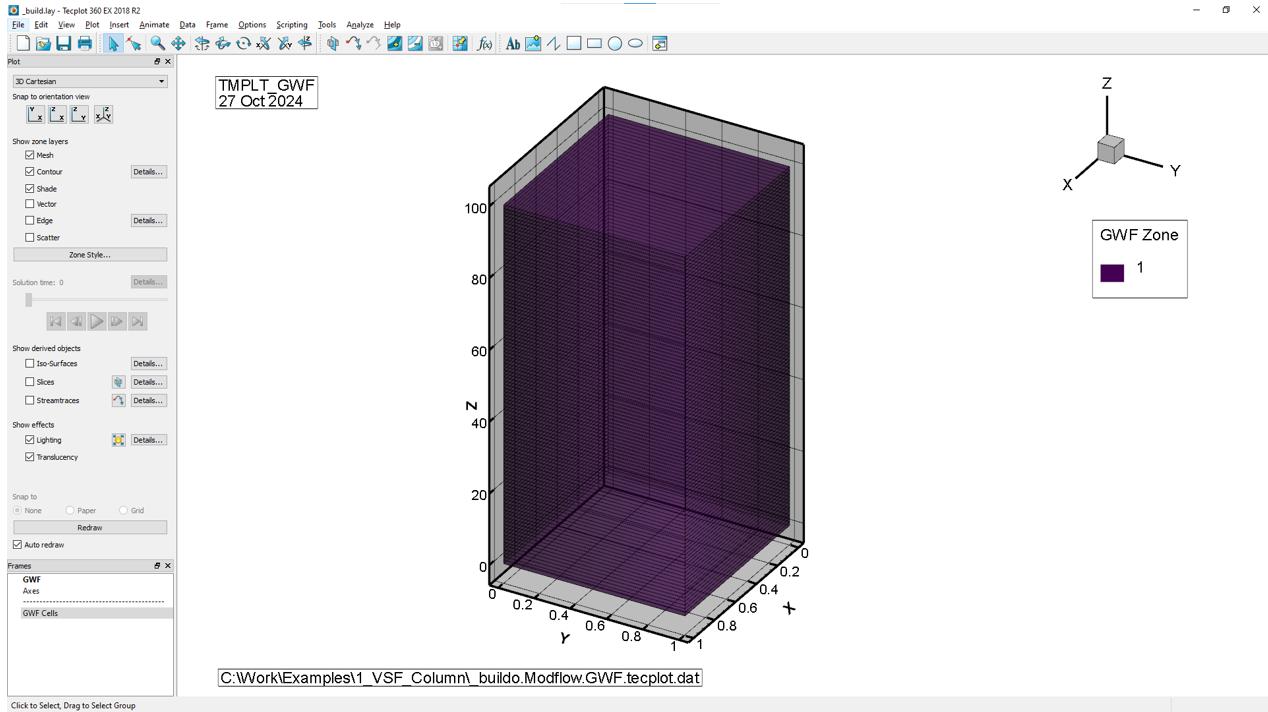
background. The contents of any frame above it may be partly or completely visible, depending on which other frames are in front of it. Move the background to the front by right-clicking on the name and selecting 'Bring to front'.



You should now see an empty white TECPLT image.

### 3.9.1 GWF Domain

Right-click on the GWF frame and bring it to the front to see it in isolation.



Every frame below the background frame is now invisible.

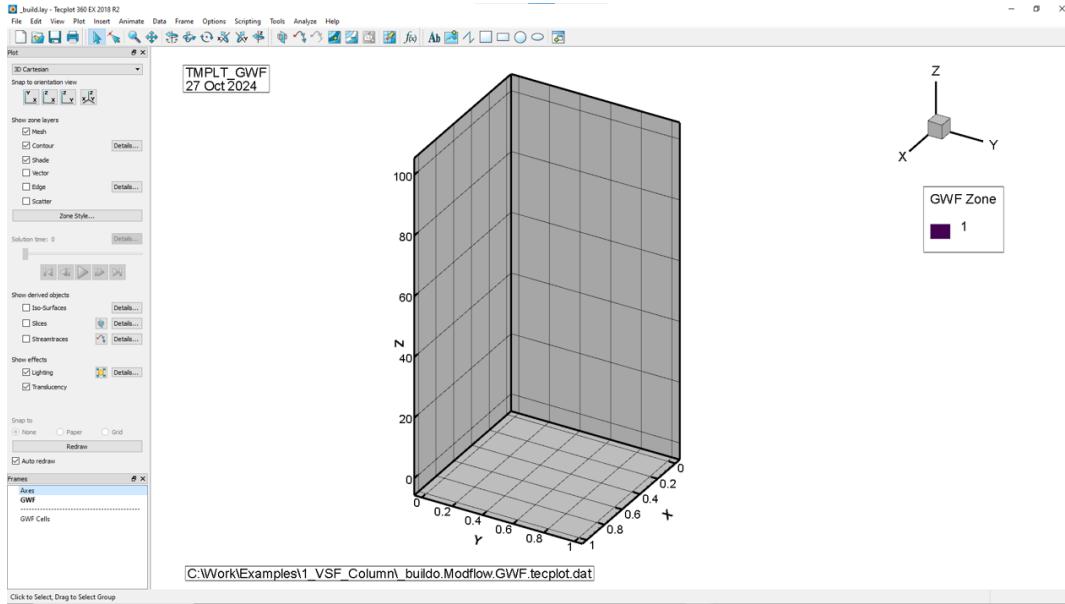
The GWF frame has the following contents:

- The finite-element mesh is shown by the translucent blue-shaded volume and wireframe block elements.
- The names of the data files loaded into the frame are shown at the bottom left corner.
- The data set title and current date (on the day the file was loaded) are shown at the top left corner. The data set title 'TMPLT\_GWF' indicates that this is the GWF domain mesh that was

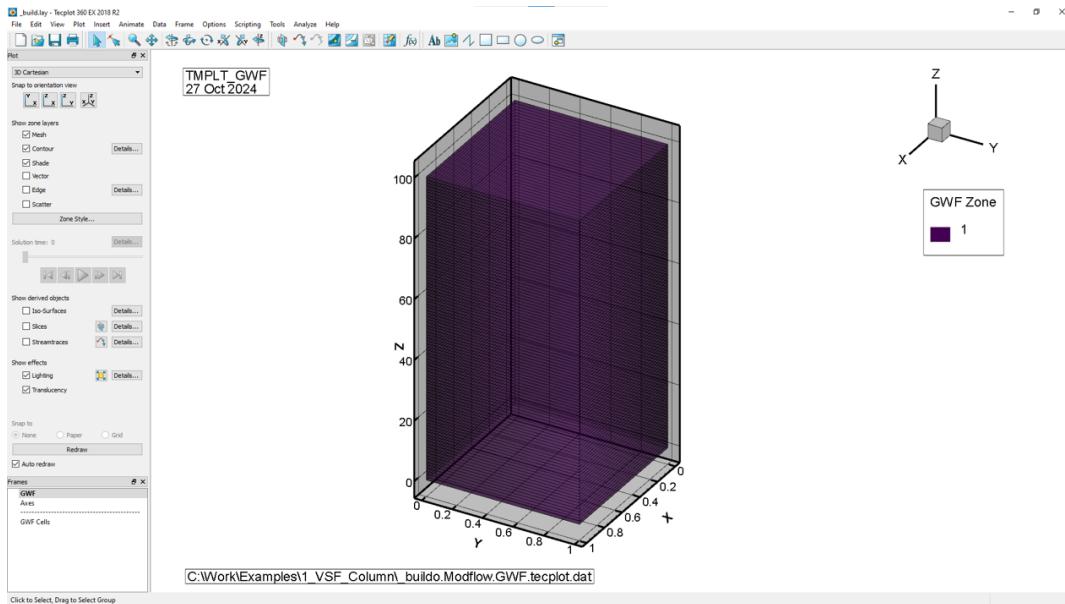
created from the template mesh.

- The contouring legend, showing there is one GWF zone, is shown at the middle right side.
- The 3D orientation axis is shown at the top right corner.

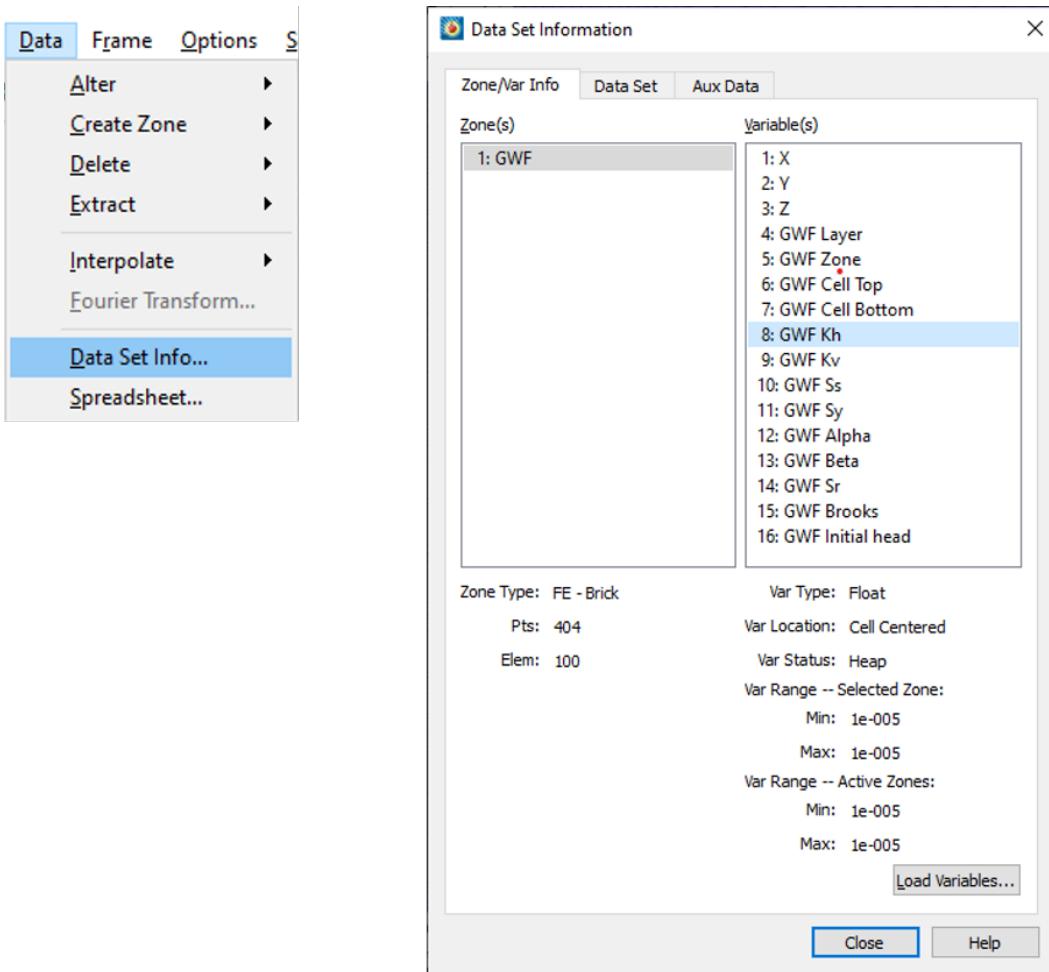
Right-click on the Axes frame and bring it to the front.



Some of the contents of the GWF frame are still visible but the finite-element mesh is obscured by the axes. Right-click on the GWF frame and bring it to the front.



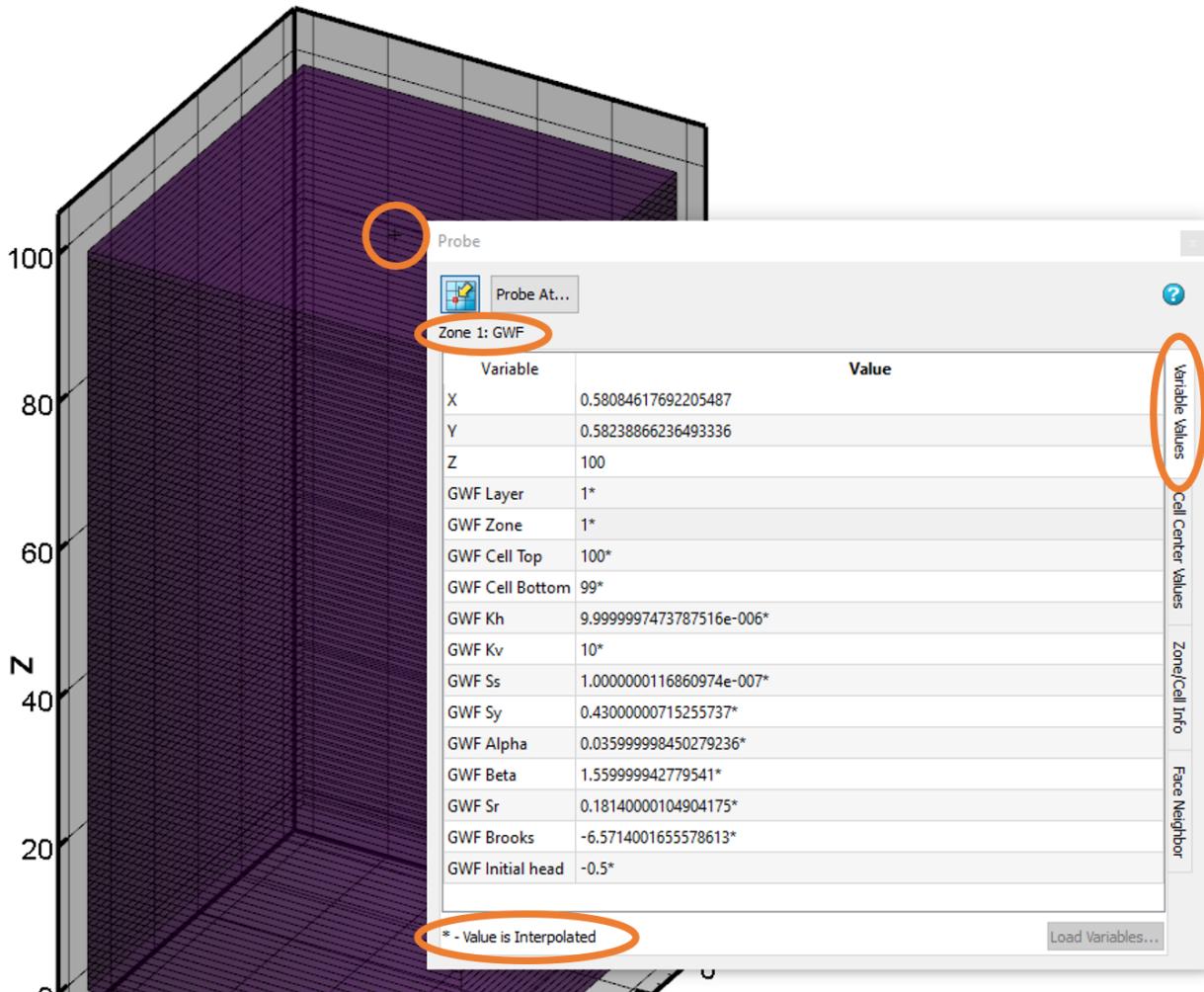
The menu option Data\DataSet Info... (shown below left), brings up the Data Set Information dialogue (shown below right):



The currently active frame title **GWF** is shown in the **Zone(s)** field, while the data set variables are listed in the **Variable(s)** field. The variables include the *xyz* coordinates of the **TMPLT\_GWF** mesh nodes, followed by the Layer and Zone numbers assigned to the **TMPLT\_GWF** mesh elements. The remainder of the list shows the MODFLOW-USG<sup>*swf*</sup> cell properties that were defined during the model build. The **Var–Range Selected Zone:** area shows the variable **Min:** and **Max:** values for the currently chosen (highlighted) zone and variable (currently **GWF** and **GWF Kh** respectively).

 The TECPLT Probe Tool is used to probe for values of the dataset's variables at a particular point. When selected, the mouse cursor changes to a modified cross-hair which indicates the Probe Tool is active. To obtain *interpolated* values of the dataset variables at the specified location, click at any point in the data region.

Shown below are the results of probing the top cell of the **GWF** domain:



The probe location is indicated by the small '+' sign in the top cell (left orange circle). This opens the Probe dialogue which shows the zone probed (Zone 1: GWF), and a table of Variable names and values.

It is important to understand that in TECPLT nomenclature, the Variable Values tab refers to values assigned to TMPLT\_GWF mesh nodes, while the Cell Centred Values tab (*Not to be confused with MODFLOW cells!*) refers to values assigned to TMPLT\_GWF mesh elements.

If the mesh-centred control volume approach is used, as is the case for this example, then MODFLOW cell locations align with TMPLT\_GWF mesh element centroids, and all values except the XYZ coordinates (which are defined at TMPLT\_GWF mesh nodes) are interpolated (as indicated by an asterisk \* appended to the value). The XYZ coordinates show the exact location of the small '+' sign.

If we select the Cell Centred Values tab, the Probe output looks like this:

Probe

Zone 1: GWF

Variable	Value
X	0.5*
Y	0.5*
Z	99.5*
GWF Layer	1
GWF Zone	1
GWF Cell Top	100
GWF Cell Bottom	99
GWF Kh	9.999997473787516e-006
GWF Kv	10
GWF Ss	1.0000000116860974e-007
GWF Sy	0.43000000715255737
GWF Alpha	0.035999998450279236
GWF Beta	1.559999942779541
GWF Sr	0.18140000104904175
GWF Brooks	-6.5714001655578613
GWF Initial head	-0.5

\* - Value is Interpolated

Load Variables...

Variable Values  
Cell Center Values  
Zone(Cell Info)  
Face Neighbor

Now the *XYZ* coordinates are interpolated and show the approximate TMPLT\_GWF mesh element centroid location, while for all other variables, exact values (i.e. what was input) are shown.

If the node-centred control volume approach is used, as is the case for the 6\_Abdul\_Prism\_Cell\_nc example, then MODFLOW cell locations align with TMPLT\_GWF mesh nodes, and no variable values are interpolated when the Cell Centred Values tab is selected.

Probe

Zone 1: GWF

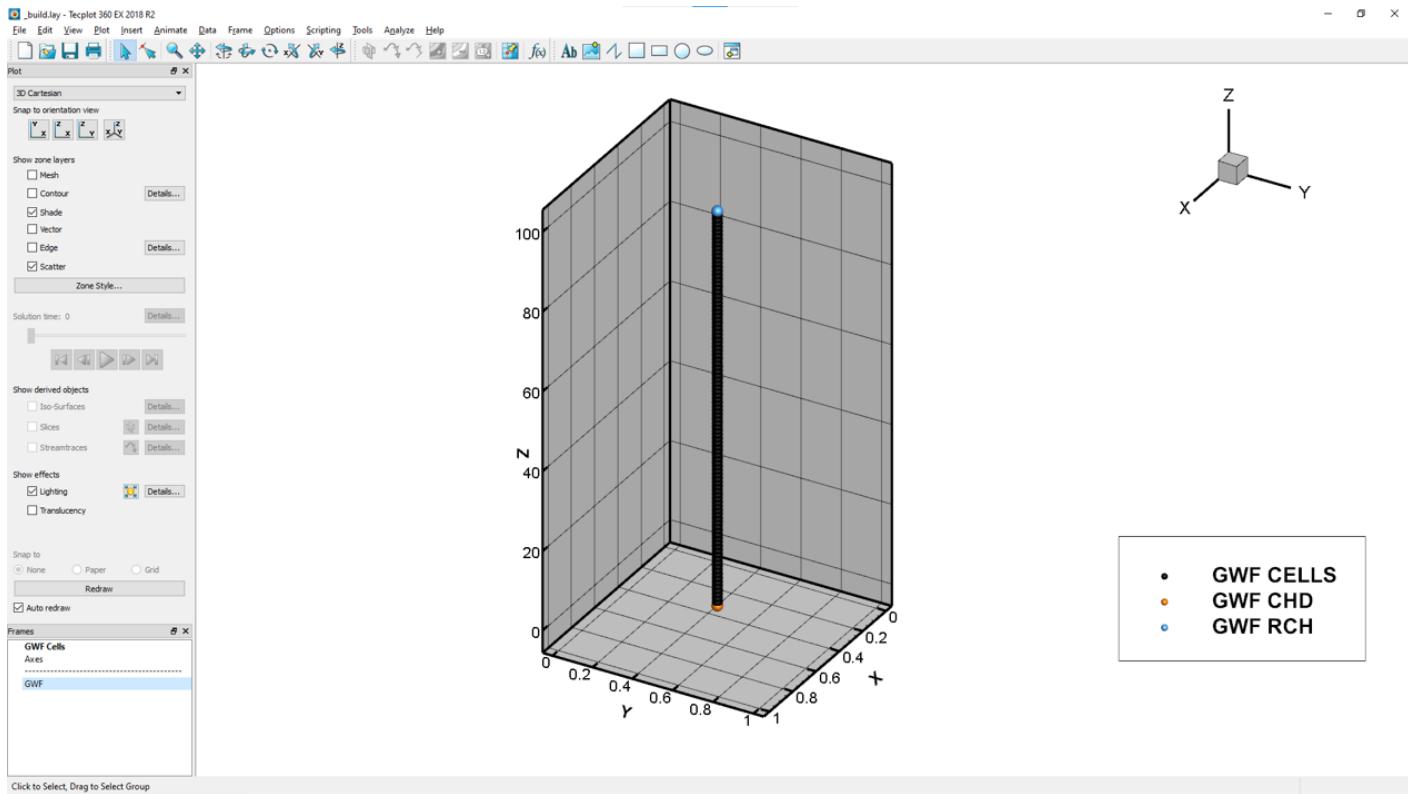
Variable	Value
X	47.937981085299597
Y	11.83543514715265
Z	3.2684222284199946
GWF Layer	1
GWF Zone	1
GWF Cell Top	3.2684222284199946
GWF Cell Bottom	3.21842227610371
GWF Kh	9.999997473787516e-006
GWF Kv	9.999997473787516e-006
GWF Ss	1.19999957179898e-007
GWF Sy	0.34000000357627869
GWF Alpha	1.899999761581421
GWF Beta	6
GWF Sr	0.18000000715255737
GWF Brooks	-1
GWF Initial head	2.7799999713897705

\* - Value is Interpolated

Load Variables...

Variable Values  
Cell Center Values  
Zone(Cell Info)  
Face Neighbor

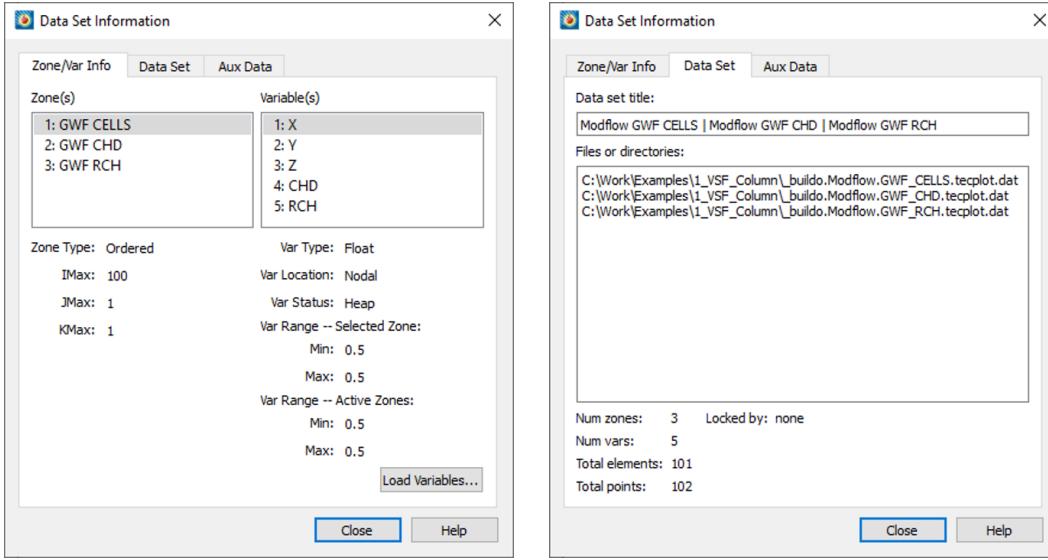
Bring the GWF Cells frame to the front, and send the GWF frame to the back.



The GWF Cells frame has the following contents:

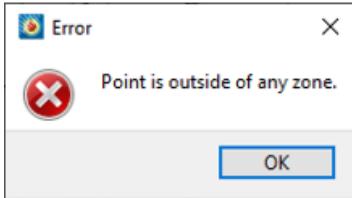
- The legend, which shows three types of scatter points called GWF Cells, GWF CHD and GWF RCH, is shown at the middle right side.
- The MODFLOW-USG<sup>Swf</sup> cell locations are shown by the black spheres.
- The cell assigned recharge is shown by the large blue sphere.
- The cell assigned a constant head is shown by the large green sphere.

If boundary conditions are assigned to the GWF domain, as they are in this case, then Tecplot output files will be produced for each type. The `_build.lay` file has been configured to load these files.

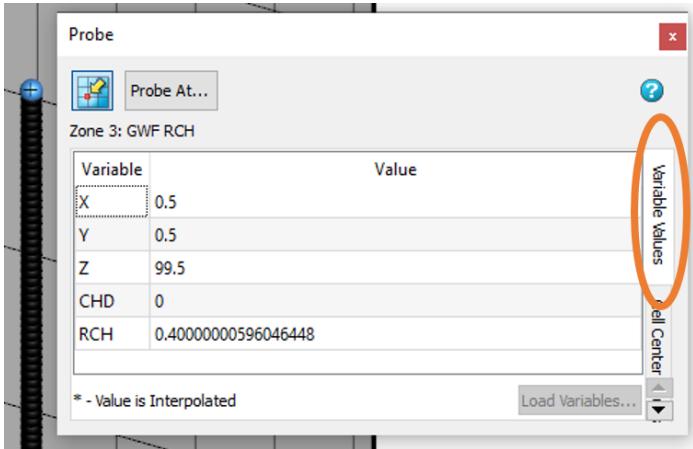


In the Zone/Var Info tab, there are 3 zones, and 5 variables. In the data Set tab, we can see that 3 files have been loaded.

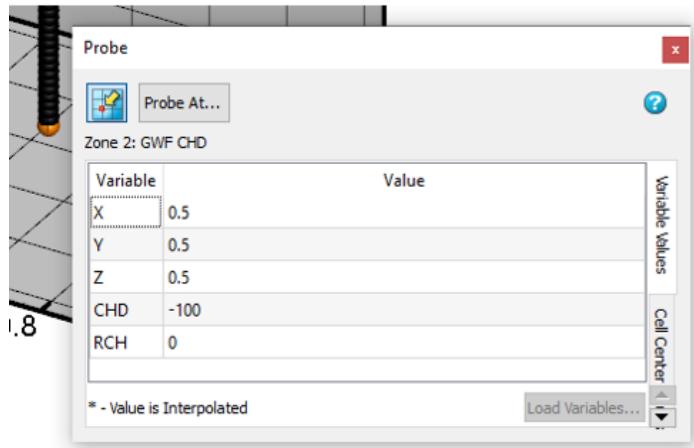
Try to probe the blue sphere, and you will likely get the following warning:



This is because the cell is located at an exact *XYZ* point, and the chance of clicking the mouse right there is very small. In this case, to obtain *exact* values for the data point nearest the specified location, hold down the Control key while clicking the mouse at the desired location.



You must make sure to have the Variable Values tab selected to see values. Here, we see the *XYZ* location of the nearest cell, and the recharge (RCH) value 0.4. Note that the CHD value of 0 is shown by default at non-constant head nodes, and does not mean a constant head of zero has been assigned. If we probe the green sphere we see this.

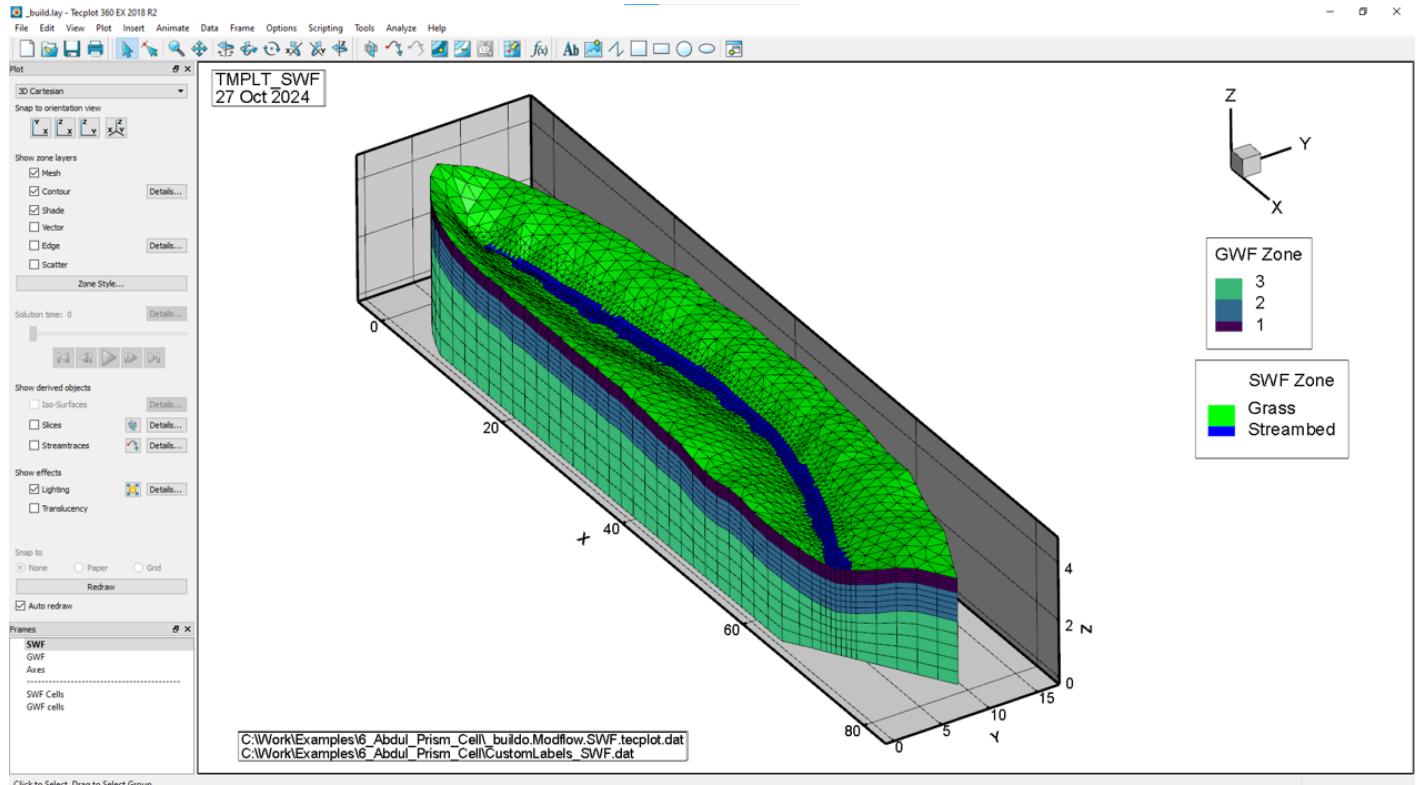


Now the constant head (CHD) value -100 is shown while the RCH value is 0, because this is not an assigned recharge cell.

### 3.9.2 SWF Domain

The verification example 6\_Abdul\_Prism\_Cell has a SWF domain defined by an irregular 2D surface with a recharge boundary condition assigned to the entire domain and a critical depth outflow boundary condition assigned at the downstream outlet. Since it also has a GWF domain the \_build.lay file is a bit more complicated, but has many of the previous example.

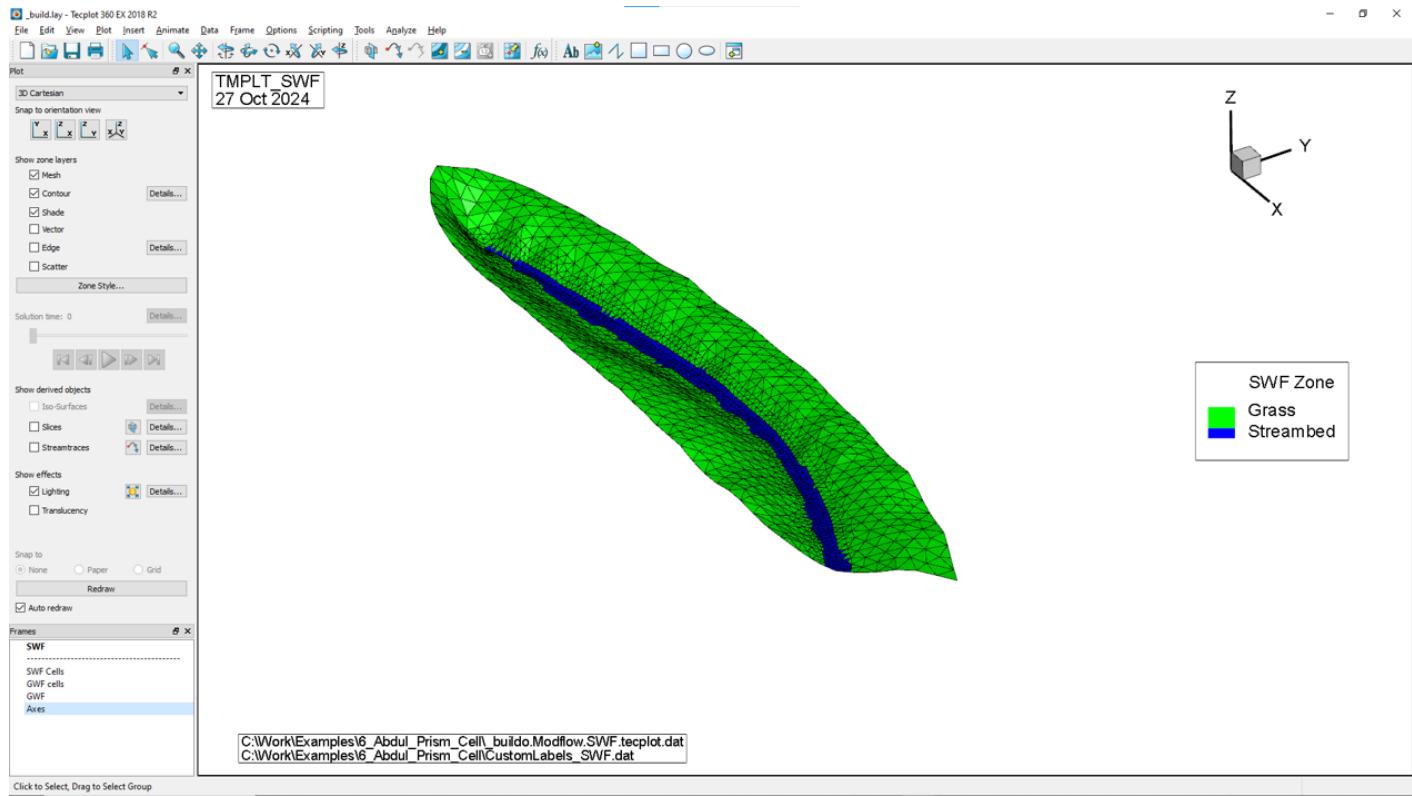
Start a new command prompt, run TECPLOT and load the \_build.lay file.



Here we can see the SWF, GWF and Axes frames are visible (i.e. placed above the background frame in

the list). The **SWF** is currently active (i.e. the name is bolded) and placed at the front of the list (i.e. at the top of the list). It uses a different colormap to make it easier to distinguish the GWF domain below.

Send the **GWF** and **Axes** frames to the back to see only the contents of the **SWF** frame.



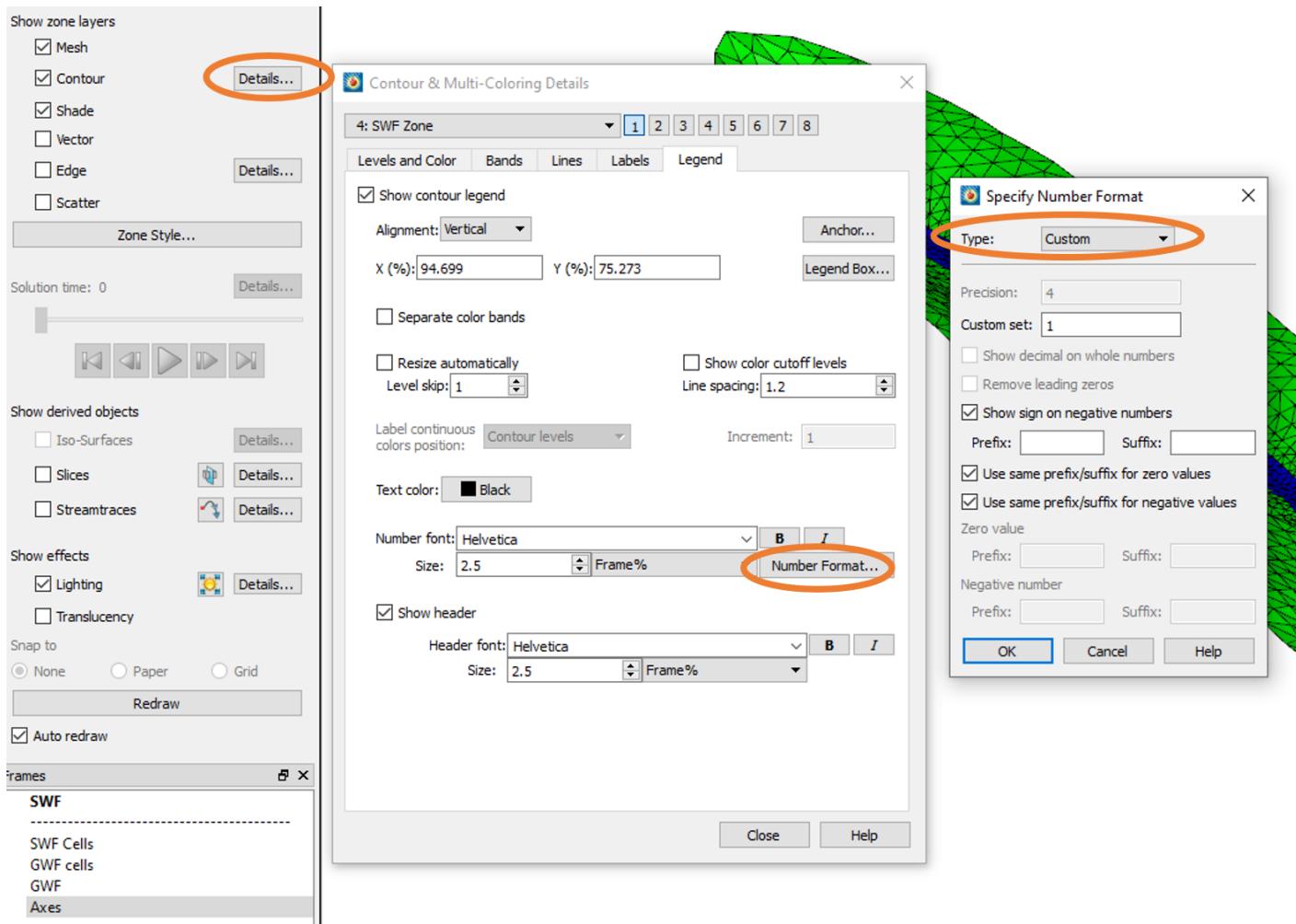
The **SWF** frame has very similar contents to the **GWF** frame described earlier, but note that:

- The data set title **TMPLT\_SWF** indicates that this is the **SWF** domain mesh that was created from the template mesh.
- The names of the data files loaded into the frame are for the **SWF** domain, and there is an extra data set that has been loaded called **CustomLabels\_SWF.dat**.
- The contouring legend, showing there are two **SWF** zones called **Grass** and **Streambed** is a bit more descriptive than a list of zone numbers. These are referred to a custom labels in **TECPLOT**.

To define custom labels as shown for the **SWF** legend, you must define a tecplot file that contains the custom label set. This is in fact the **CustomLabels\_SWF.dat** file loaded in the **SWF** frame, which has the following contents:

```
CUSTOMLABELS
"Streambed",
"Grass",
"Zone 3",
"Zone 4",
"Zone 5",
"Zone 6",
```

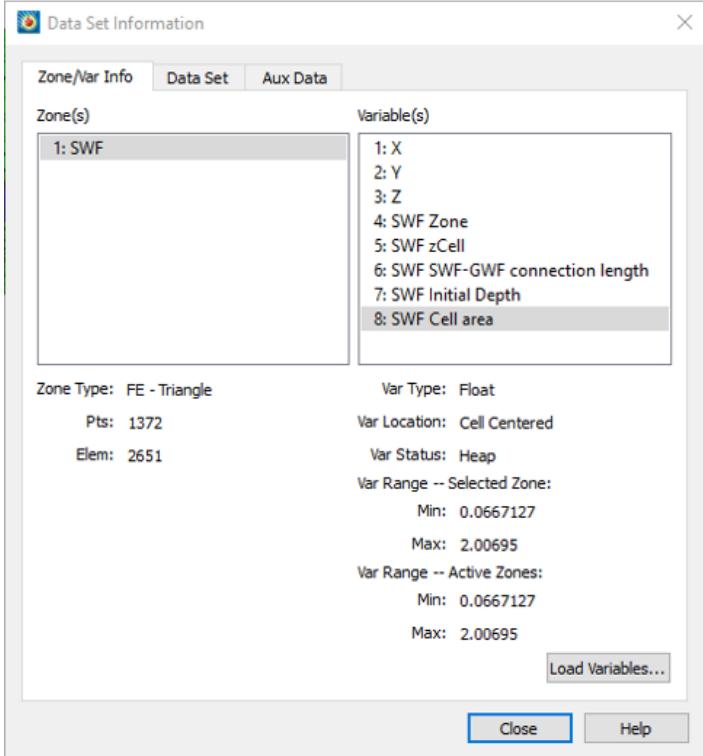
Although the SWF domain only has 2 zones, the set contains enough entries for 6 zones. The extra entries are ignored.



As shown above, the use of custom labels is configured by first choosing the contouring Details... button, which opens the Contour & Multi-Coloring Details dialogue. Choose the Number Format... button to open the Specify Number Format dialogue, then choose Custom from the Type: drop-down menu.

Custom labels can also be defined for the GWF and CLN domains if desired.

Data Set Information defined for the SWF domain is shown below:



The currently active frame title **SWF** is shown in the **Zone(s)** field, while the data set variables are listed in the **Variable(s)** field. The variables include the *xyz* coordinates of the **TMPLT\_SWF** mesh nodes, followed by the Zone numbers assigned to the **TMPLT\_SWF** mesh elements. The remainder of the list shows the MODFLOW-USG<sup>*swf*</sup> cell properties that were defined during the model build. The **Var–Range Selected Zone:** area shows the variable **Min:** and **Max:** values for the currently chosen (highlighted) zone and variable (currently **SWF** and **SWF Cell Area** respectively).

For zoned variables, the **SWF Zone** number can be used to check the assigned properties by referring back to the **\_build0.eco** file output. For the **6\_Abdul\_Prism\_Cell** example, the properties of the first zone are shown as being for the material Streambed:

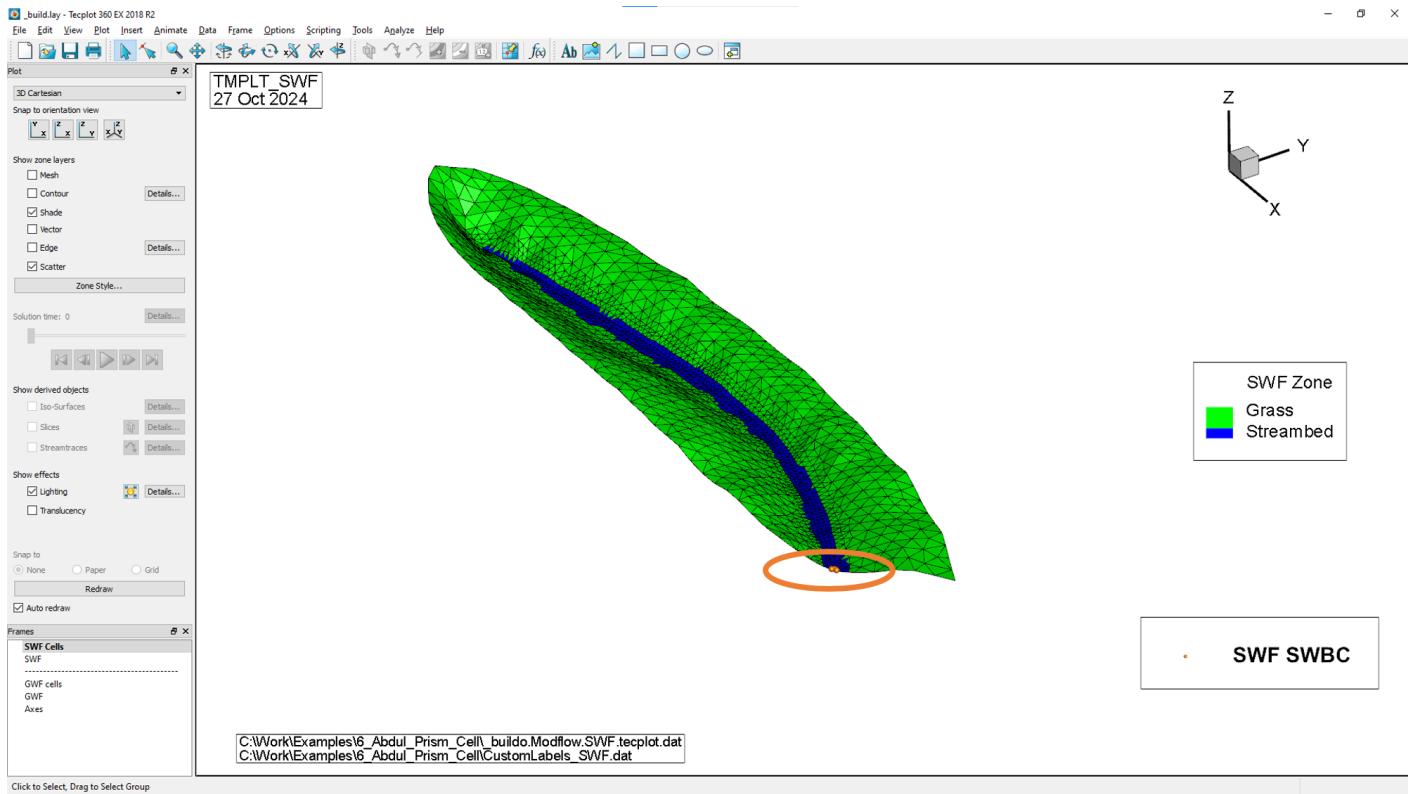
```

choose zone number
Adding zone number:      1
SWF zone numbers currently chosen:
  1

chosen zones use swf material number
Assigning all chosen SWF zones properties of material      3, Streambed
Manning's Coefficient:      3.00000E-02      METERS^(-1/3)  SECONDS
Depression Storage Height:  0.10000      METERS
Obstruction Storage Height: 0.0000      METERS
SWF Smoothing Depth 1:     1.00000E-06      METERS
SWF Smoothing Depth 2:     1.00000E-06      METERS

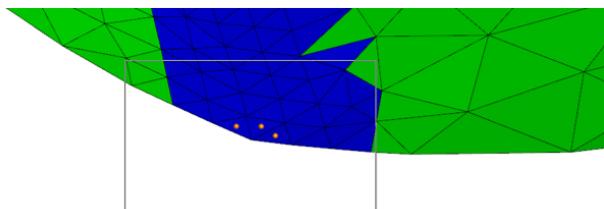
```

Bring the **SWF Cells** frame to the front.

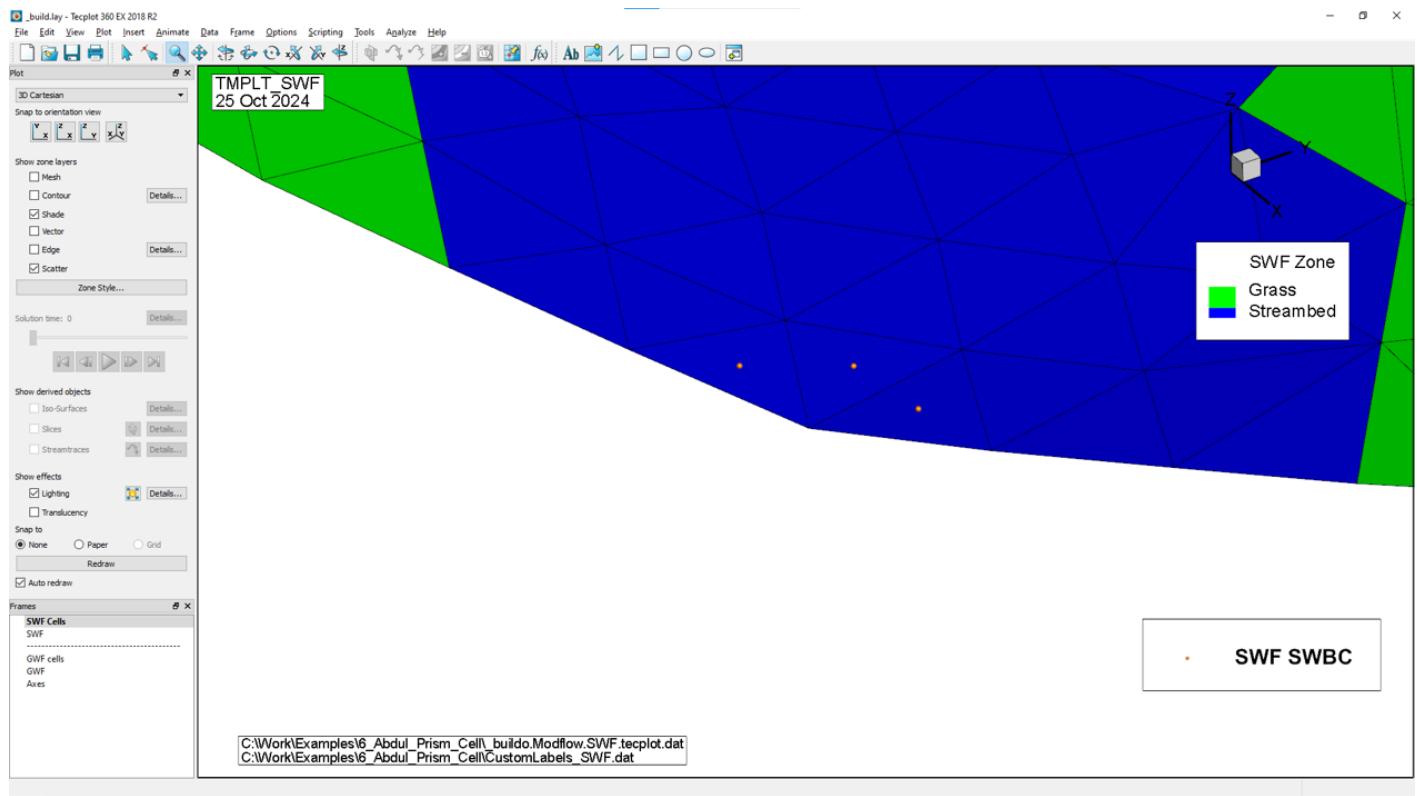


By default, only one scatter point set called **SWF SWBC** is being displayed. **SWBC** stands for Surface Water Boundary Condition, and contains values for the assigned critical depth boundary condition, shown as 3 orange spheres at the downstream end of the streambed, as indicated by the orange oval.

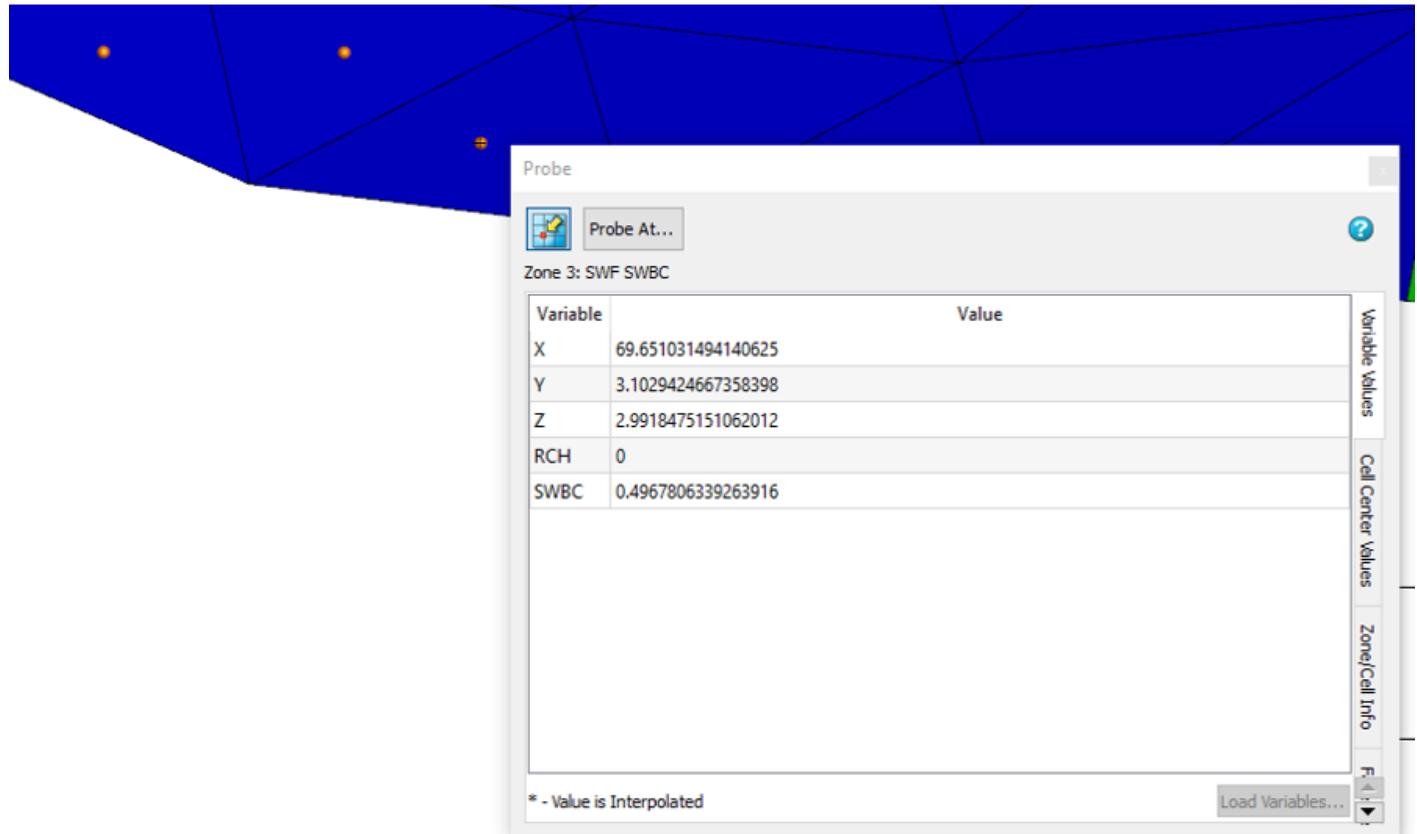
The Zoom Tool can be used to expand the scale around the outlet. Drag the magnifying glass cursor to draw a box around the region that you want to fit into the frame.



The region within the view box will be resized to fit into the frame.

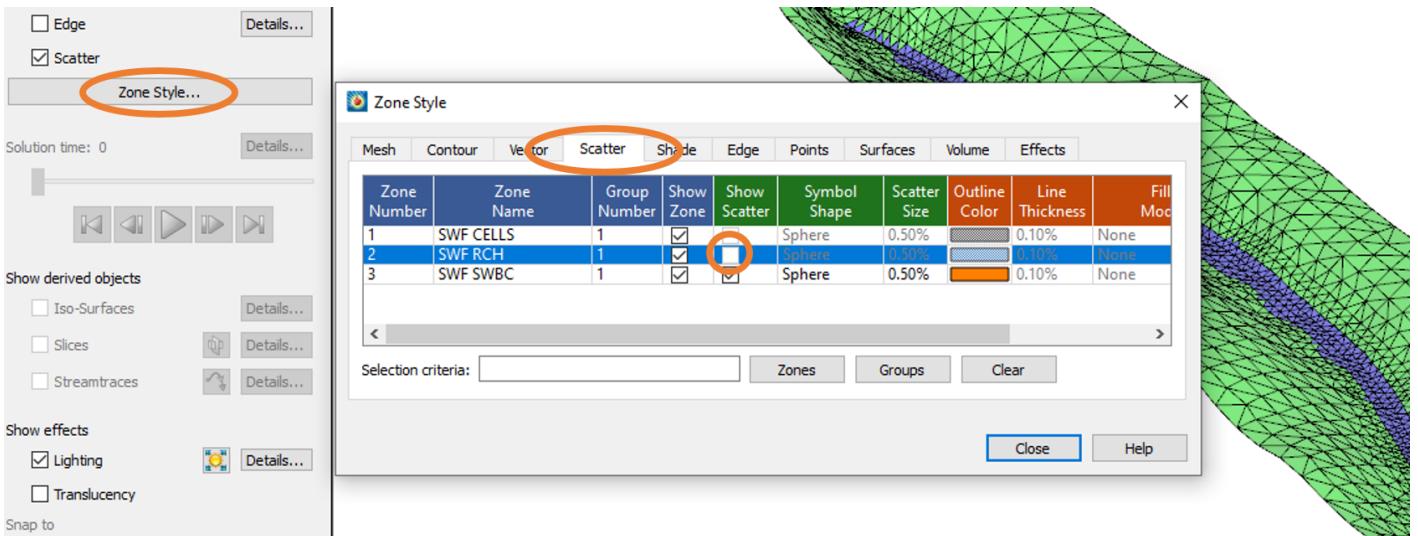


Using the Probe Tool, hold down the Control key and click the mouse on one of the orange spheres.

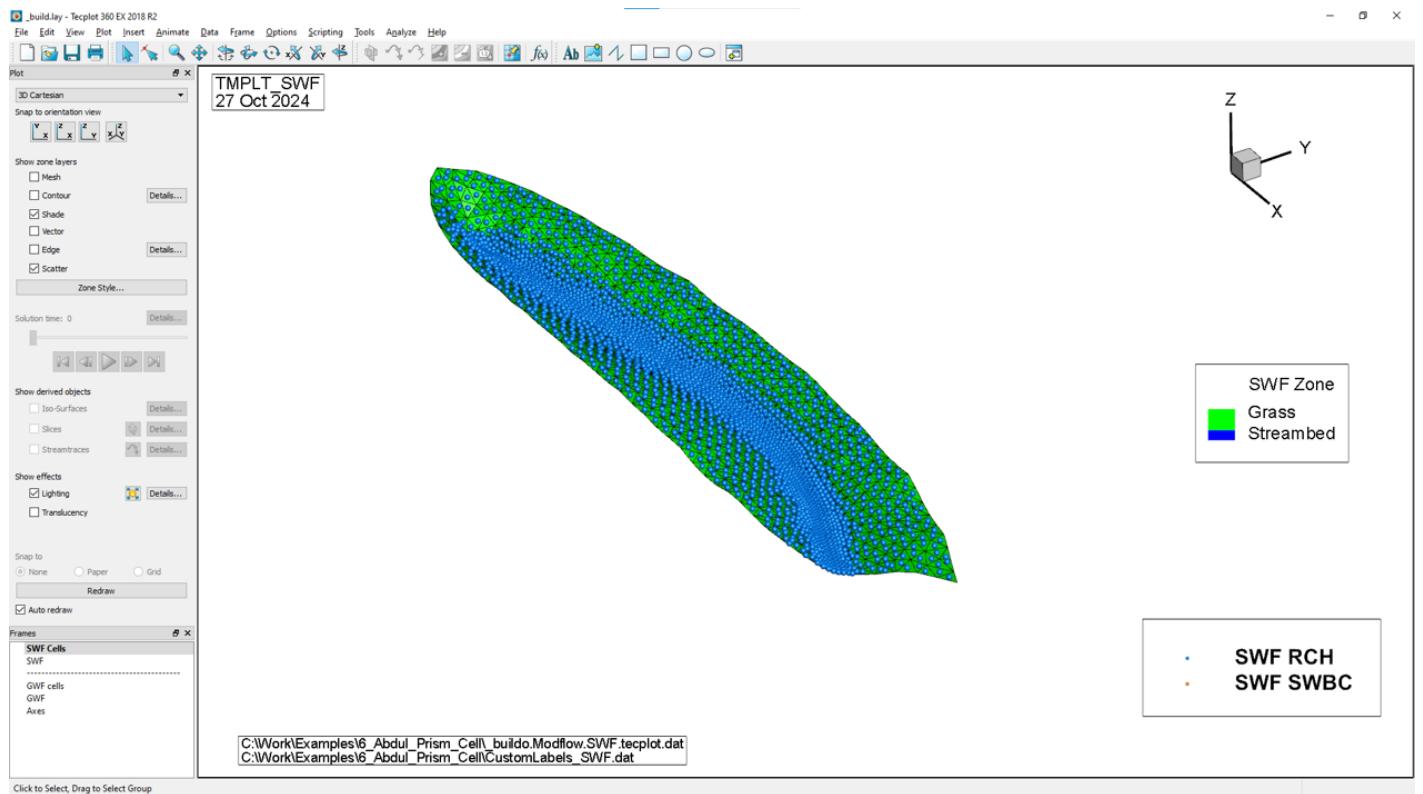


Although the critical depth boundary condition does not require any input, it does use a contributing

length over which it is applied, which is the value shown above for SWBC. The presence of the variable RCH indicates it has been defined for the SWF domain. To show the RCH cells:



Click the Zone Style... button, which opens the Zone Style dialogue. Select the Scatter tab, then click in the empty box in the Show Scatter column for the SWF RCH zone.

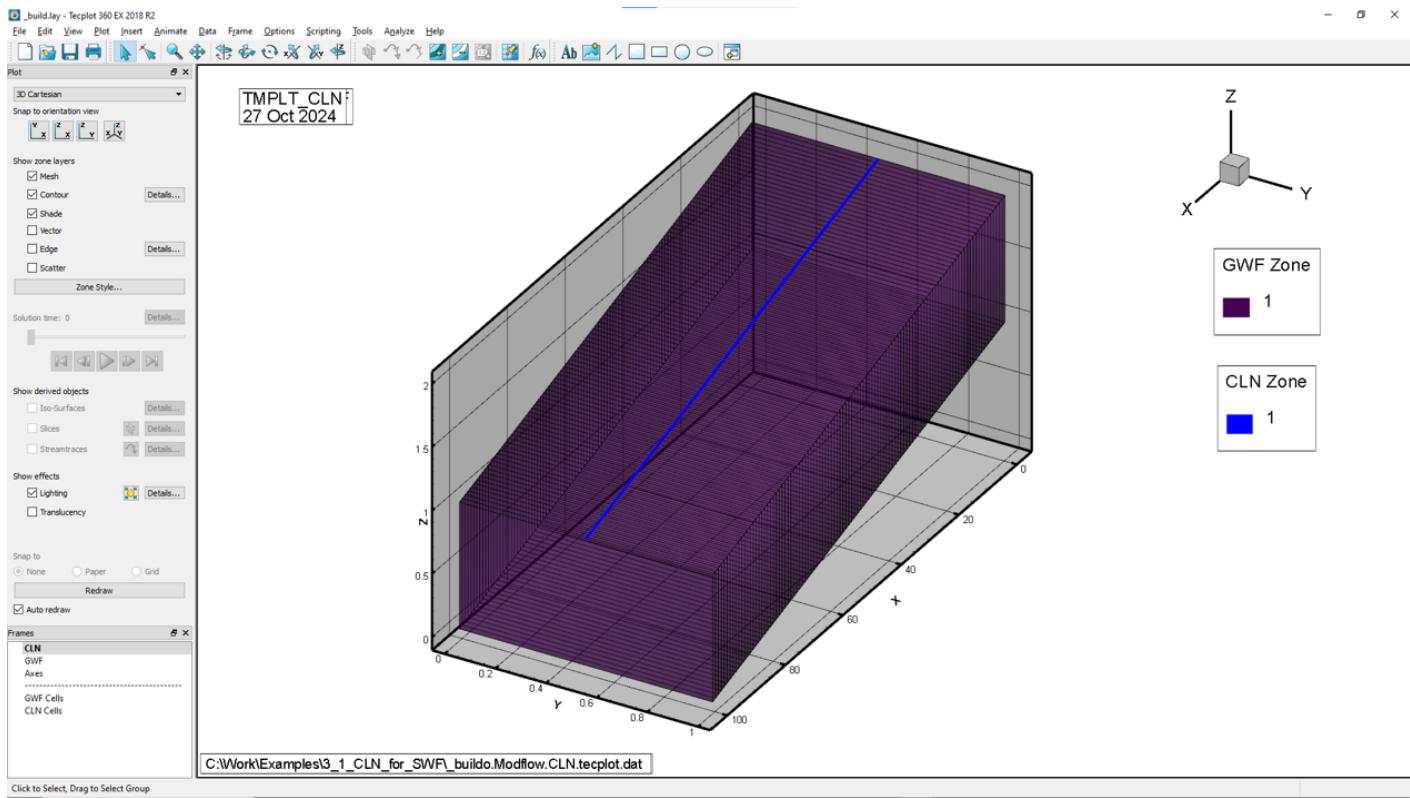


If you probe the recharge cells, you will see values of zero everywhere. This is because the values shown are for the last defined stress period, which for this example is the second one, where the recharge (rainfall) was stopped (i.e. set to zero).

### 3.9.3 CLN Domain

The verification example 3\_1\_CLN\_for\_SWF has a CLN domain defined by the end coordinates of a sloping straight line with constant head boundary conditions defined at each end of the GWF domain.

Start a new command prompt, run TECPLOT and load the \_build.lay file.

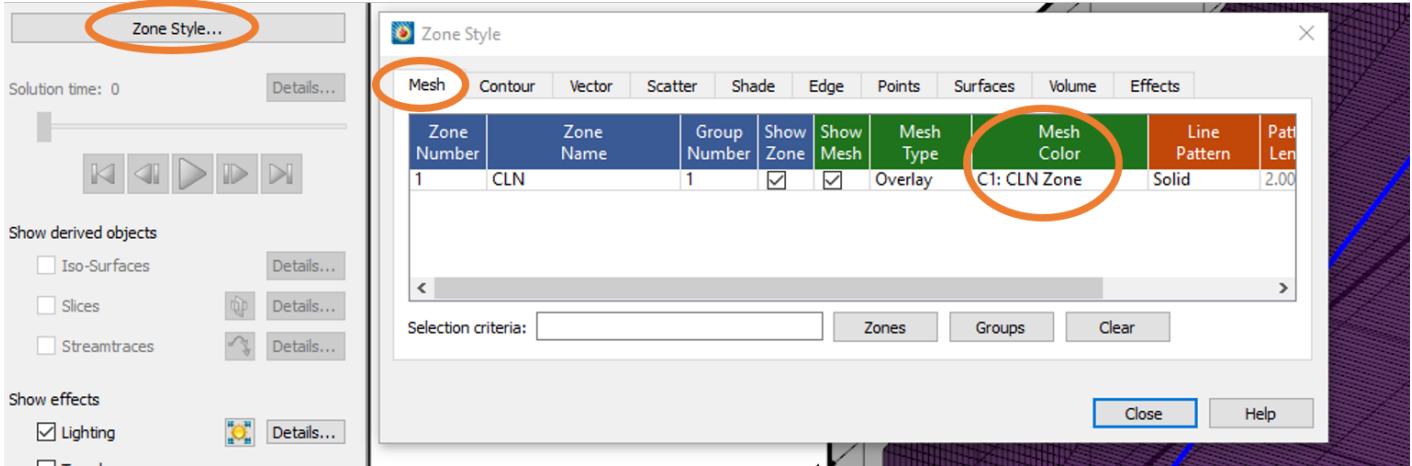


Here we can see the **CLN**, **GWF** and **Axes** frames are visible (i.e. placed above the **background** frame in the list). The **SWF** is currently active (i.e. the name is bolded) and placed at the front of the image (i.e. at the top of the list). It uses a different colormap to make it easier to distinguish the **GWF** domain below.

The **CLN** frame has very similar contents to the **GWF** frame described earlier, but note that:

- The names of the data files loaded into the frame are for the **SWF** domain.
- The data set title **TMPLT\_CLN** indicates that this is the **SWF** domain mesh that was created from the template mesh.
- The contouring legend, showing there is one **CLN** zone.

CLN domain meshes cannot be contoured like GWF or SWF meshes because the CLN element is a 1D straight line and contouring can only be shown for 2D or 3D elements. In this case, the finite-element mesh has been coloured by the **CLN Zone** number.



For zoned variables, the CLN Zone number can be used to check the assigned properties by referring back to the `_buildo.eco` file output. For the `3_1_CLN_for_SWF` example, the properties of all zones are shown as being for material 1, 2D Hillslope 100 m length:

choose all zones

chosen zones use cln material number

Assigning all chosen CLN zones properties of material 1, 2D Hillslope 100 m length

Geometry: Rectangular

Rectangular Width: 1.0000 METERS

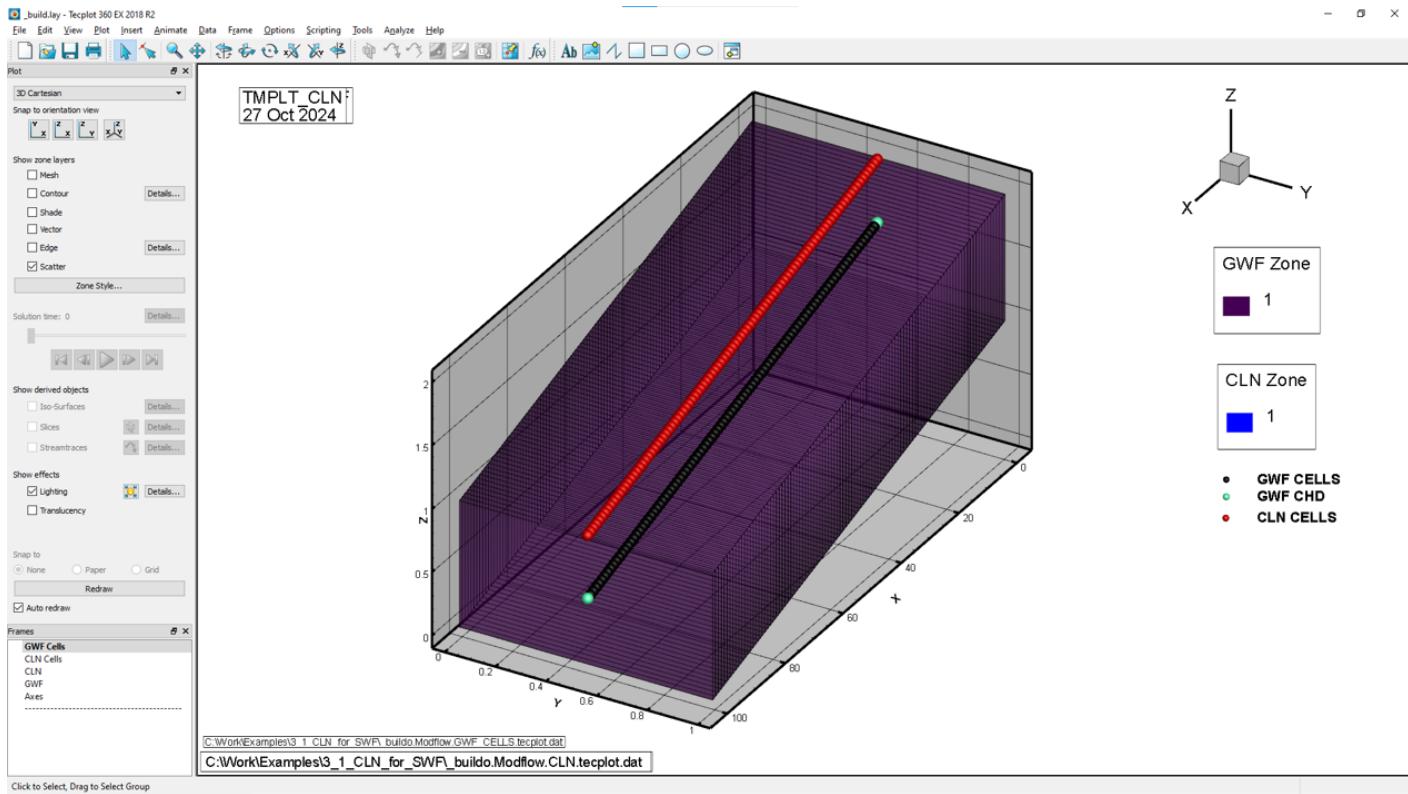
Rectangular Height: 1.0000 METERS

Direction: Horizontal

Flow Treatment: Unconfined/Mannings

Longitudinal K: 5.48000E-02 METERS SECONDS<sup>-1</sup>

Bring the CLN Cells and GWF Cells frames to the front.



The CLN cells (red spheres) show that it is located along the top surface of the GWF domain. The GWF cells (black spheres) are located at the centroid of the GWF elements (i.e. mesh-centred control volumes), and constant head cells (light green spheres) are located at each end of the GWF domain.

# Chapter 4

## Modflow-Usg<sup>*Swf*</sup> Execution and Post-Processing

The steps in our model excution workflow are:

1. Run MODFLOW-USG<sup>*Swf*</sup>to create the new project output files (e.g. time-varying hydraulic head, drawdown etc).
2. Run MUT to post-process the Modflow project, which produces TECPLOT output files for the various Modflow domains (i.e. GWF, SWF and/or CLN) created during the Modflow simulation.
3. Run TECPLOT and examine the Modflow output files.

### 4.1 Modflow-Usg<sup>*Swf*</sup> Execution

To run MODFLOW-USG<sup>*Swf*</sup> on the `1_VSF_Column` example, and assuming we have a command prompt open at the appropriate directory, we can type:

```
usgs_1 modflow
```

which obtains the prefix for the MODFLOW-USG<sup>*Swf*</sup> input files, in this case the default prefix `modflow`.

As MODFLOW-USG<sup>*Swf*</sup> processes the input file, output is written to the screen:

```

C:\WINDOWS\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Work\Examples\1_VSF_Column>usgs_1 modflow

      USG-TRANSPORT
MODFLOW-USG GROUNDWATER FLOW AND TRANSPORT MODEL
      Version USG-TRANSPORT VERSION 2.02.0

Using NAME file: modflow.nam
Run start date and time (yyyy/mm/dd hh:mm:ss): 2024/10/28 5:59:18

Solving: Stress period: 1 Time step: 1 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 2 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 3 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 4 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 5 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 6 Groundwater Flow Eqn.

```

If execution is successful you will see the Normal termination of simulation message:

```

C:\WINDOWS\System32\cmd.exe
Solving: Stress period: 1 Time step: 843 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 844 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 845 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 846 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 847 Groundwater Flow Eqn.
Solving: Stress period: 1 Time step: 848 Groundwater Flow Eqn.
Run end date and time (yyyy/mm/dd hh:mm:ss): 2024/10/28 5:59:19
Elapsed run time: 0.946 Seconds

Normal termination of simulation

C:\Work\Examples\1_VSF_Column>

```

Every MODFLOW-USG<sup>Swf</sup> simulation generates a run-time listing file, in this case called `modflow.lst`, that consists of the input data for the simulation; the solver and nonlinear outputs at user-requested detail; head and drawdown solutions, if requested; mass-balance information; and time-step information for the simulation. It also produces binary files of head, drawdown, saturation and cell-by-cell flows for each model domain.

To post-process the output produced by MODFLOW-USG<sup>Swf</sup> for the `1_VSF_Column` example, we would run MUT using the input file `_post.mut` by typing:

```
mut _post
```

If you open the file `_post.mut` in a text editor you will see the first line is a comment followed by one instruction and input:

```

! This example reads a modflow project and postprocesses it
postprocess existing modflow model
    modflow
```

As in the model build, MUT first creates a clean copy of the input file called `_posto.input` by removing all comment lines. As it processes the input file, output is written to both the screen and to the file `_posto.eco`.

## 4.2 Mut Post-processing

The instruction to post-process the MODFLOW-USG<sup>Swf</sup> model after execution is:

### postprocess existing modflow model

1. **\$\_Prefix** The MODFLOW-USG<sup>Swf</sup> model prefix.

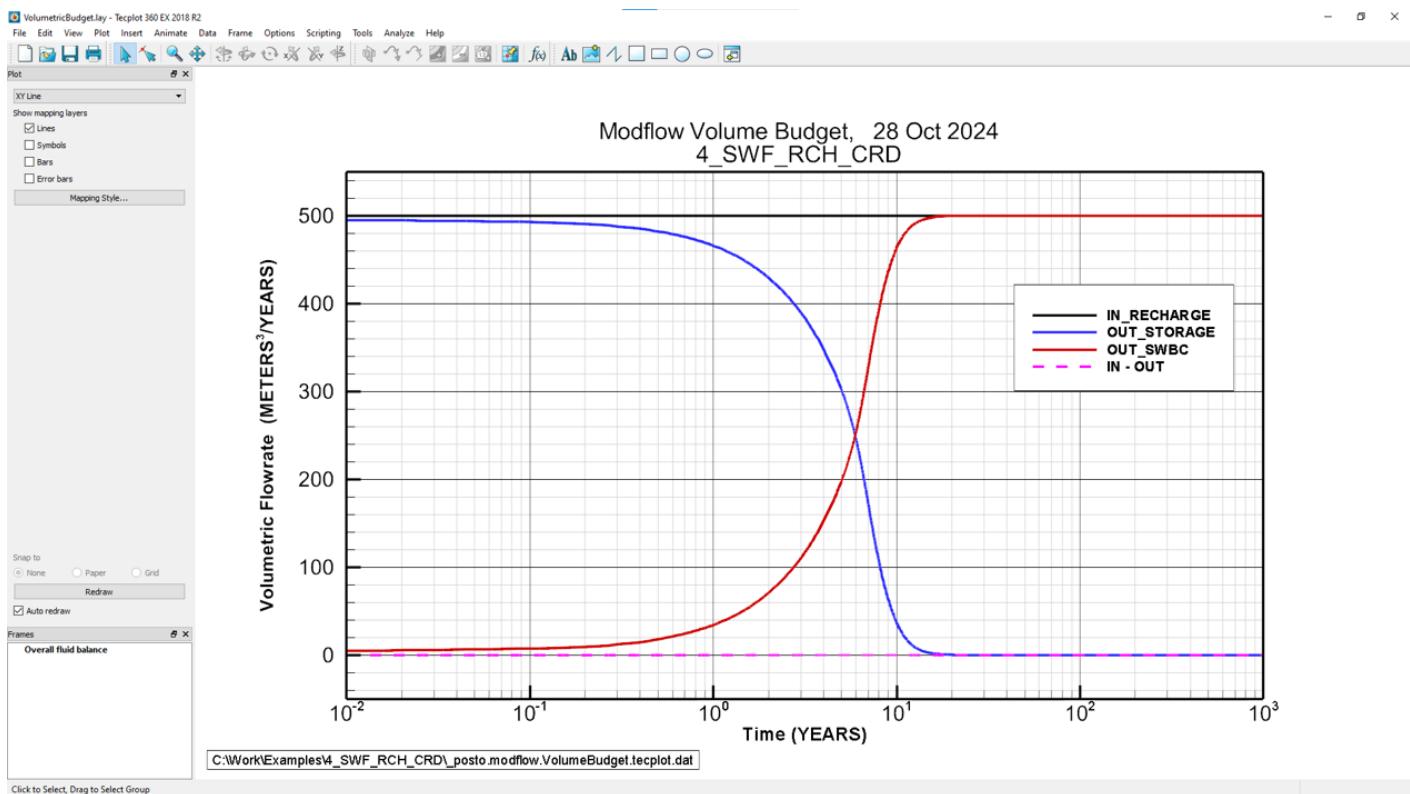
Given **\$\_Prefix**, this instruction:

- Reads head, drawdown and cell-by-cell flow binary output files for each output time and writes the results to the file `_posto.modflow.GWF.tecplot.dat`
- scans the MODFLOW-USG<sup>Swf</sup> listing file, extracts volumetric budget data at each time step and writes the results to the file `_posto.modflow.VolumeBudget.tecplot.dat`

## 4.3 Volumetric Water Budget Plots

Volumetric water budget data is useful for checking the fluid mass balance of the model run. The example 4\_SWF\_RCH\_CRD has a TECPLOT layout file `VolumetricBudget.lay` which you can load directly into TECPLOT from the command prompt by typing:

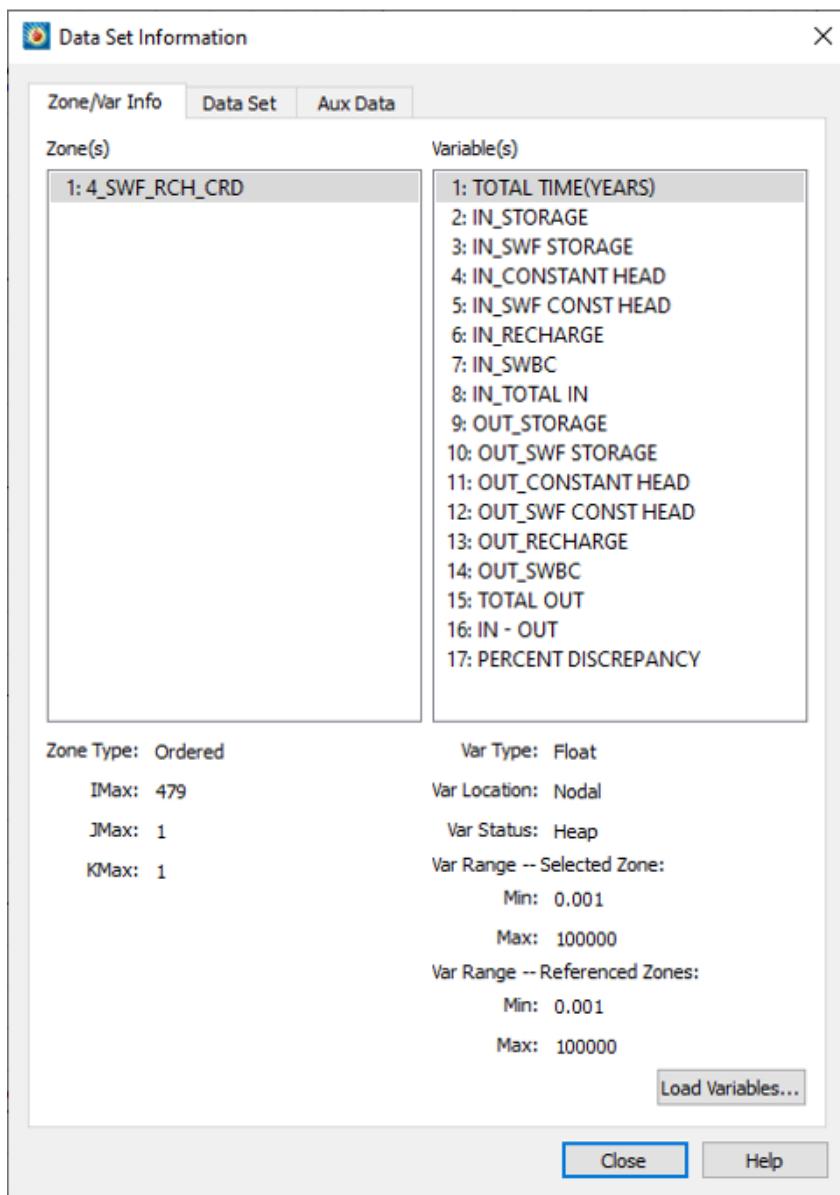
```
tec360 VolumetricBudget.lay
```



This TECPLOT frame has the following features and contents:

- It is an XY Line plot showing the volumetric flowrate versus time for selected components of the model.
- The name of the data file loaded into the frame is shown at the bottom left corner.
- The plot title, current date (on the day the file was loaded) and data set title (4\_SWF\_RCH\_CRD) are shown centred above the plot.
- The line legend is shown on the right side of the plot.
- The X-axis uses a log time scale.

The TECPLOT date set information dialogue shows all of the variables available for plotting:

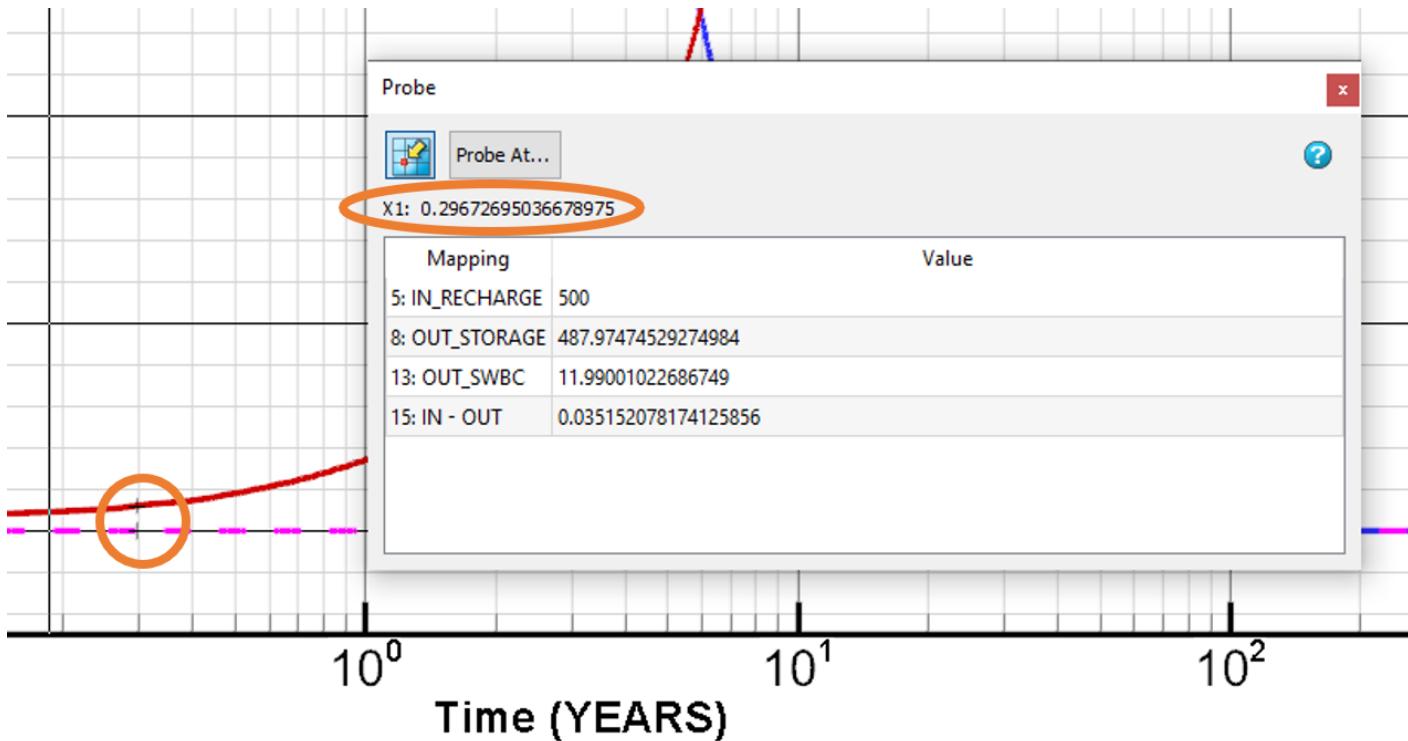


Variable names are derived from the `modflow.1st` file:

CUMULATIVE VOLUMES	L**3	RATES FOR THIS TIME STEP	L**3/T
IN:		IN:	
---		---	
STORAGE =	3.0881E-04	STORAGE =	5.6023E-09
SWF STORAGE =	9.9743E-03	SWF STORAGE =	2.1841E-13
CONSTANT HEAD =	0.0000	CONSTANT HEAD =	0.0000
SWF CONST HEAD =	0.0000	SWF CONST HEAD =	0.0000
RECHARGE =	50000000.0000	RECHARGE =	500.0000
SWBC =	0.0000	SWBC =	0.0000
TOTAL IN =	50000000.0103	TOTAL IN =	500.0000
... etc			
PERCENT DISCREPANCY =	0.01	PERCENT DISCREPANCY =	0.01

This example has a uniform recharge rate of 0.5 *meters/year* which results in a total recharge of 500 *meters<sup>3</sup>/year* when multiplied by the 1000 *meter* length of the cross-section. Initially, water comes out of storage then but this declines to zero at equilibrium. Water exiting the surface water outflow critical depth boundary OUT\_SWBC is initially zero then rises to equal the total recharge at equilibrium. Fluid balance error IN-OUT is essentially zero throughout the simulation.

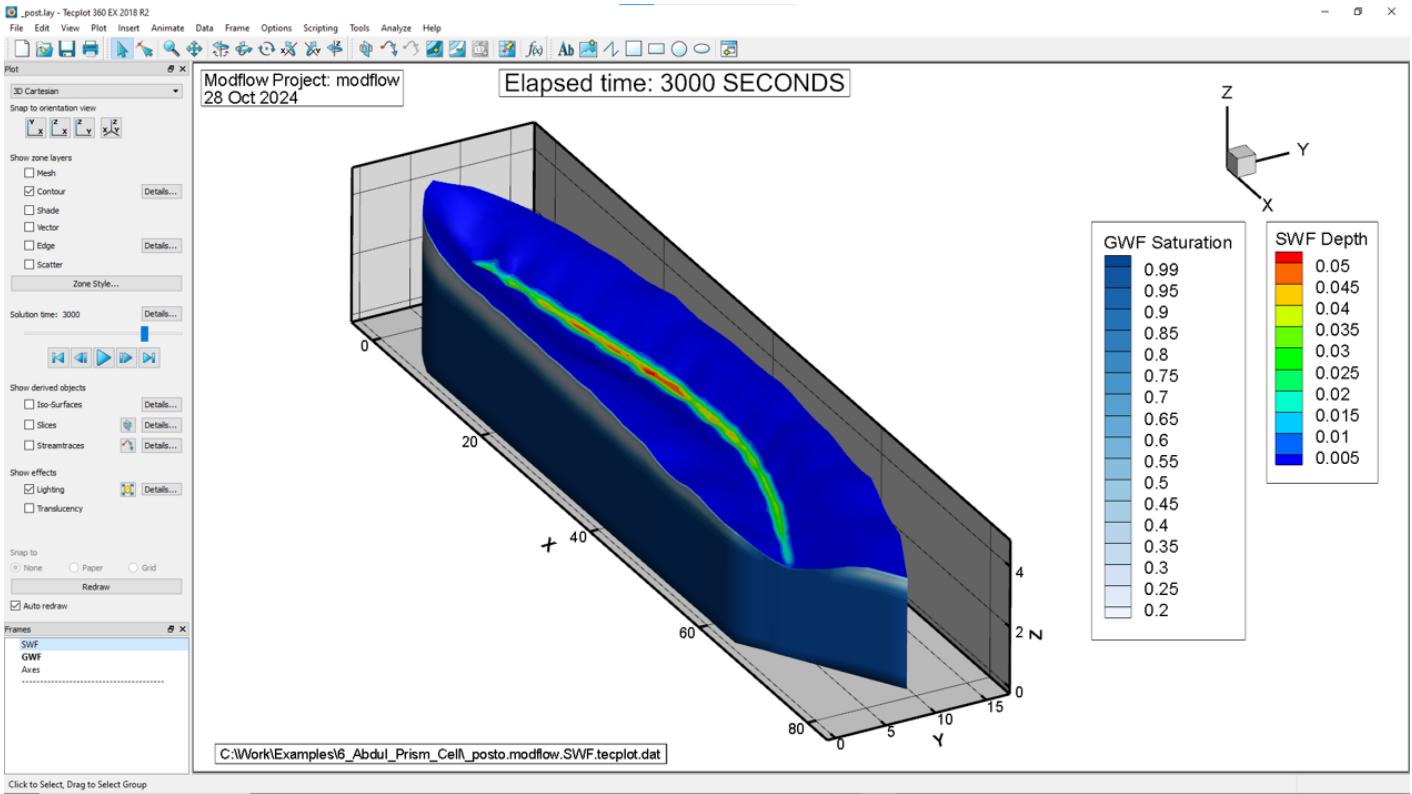
The probe tool can be used in XY Line plots to get exact variable values at a chosen location along the X or Y axis. The cursor is shown as a vertical line if probing values on the X-axis (i.e. over time):



Here, the location of the probe is shown by small vertical lines placed where the probe crossed the plotted lines. The exact coordinate is given as X1: 0.2967... years. At this early time, the OUT\_STORAGE value is still near its initial value of 500 and the OUT\_SWBC has just started rising.

## 4.4 3D Visualization of Model Results

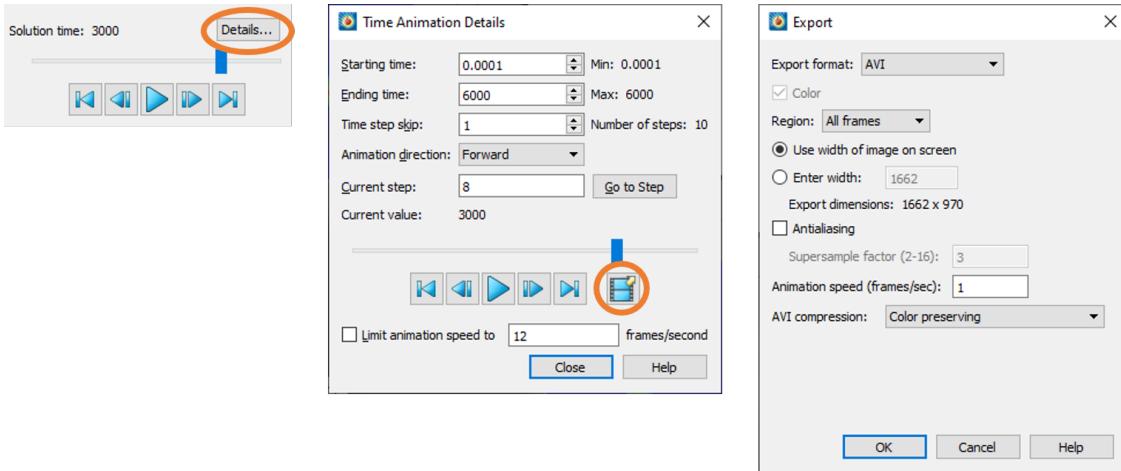
A TECPLLOT layout file, `_post.lay`, has been created for each verification example and provides a quick way to view MODFLOW-USG<sup>Swf</sup> model solution results. This result is from the example `6_Abdul_Prism_Cell`:



The `_post.lay` layout file is similar to the `_build.lay` file shown earlier for the same problem: SWF, GWF and Axes frames are visible (i.e. placed above the background frame in the list). The SWF is currently active (i.e. the name is bolded) and placed at the front of the image (i.e. at the top of the list). It uses a different colormap to make it easier to distinguish the GWF domain below. In this case though, the contoured variables are SWF Depth and GWF Saturation results from the MODFLOW-USG<sup>*Swf*</sup> simulation.

#### 4.4.1 Solution Times and Animation

The model output includes data for several output times, and the image above is showing conditions at a solution time of 3000 seconds, as indicated by the label near the top center of the plot. The solution time shown is controlled by the slider and button controls near the centre of the Plot frame on the left hand side of the image:

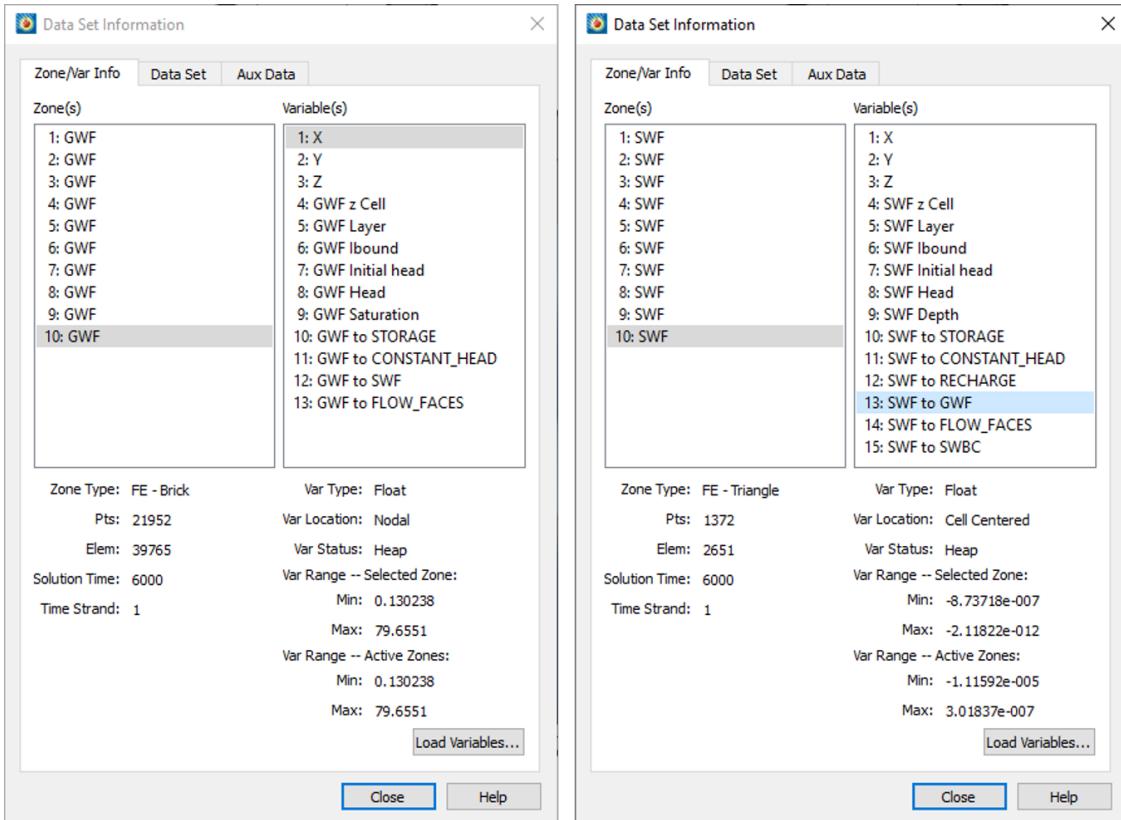


The Details... button leads to the Time animation details dialogue which allows you to control and

save animations of transient model results using the Export To File button . There is an example animation in \MUT\_Examples-main\6\_Abdul\_Prism\_Cell folder in the powerpoint file Abdul Problem Animation.pptx. It used the settings shown in the Export dialogue above to limit animation speed and write to an AVI-formatted file. This was then inserted in powerpoint where it can be viewed.

#### 4.4.2 Data Set Information

The Data Set Information dialogues for the SWF and GWF frames are shown below:



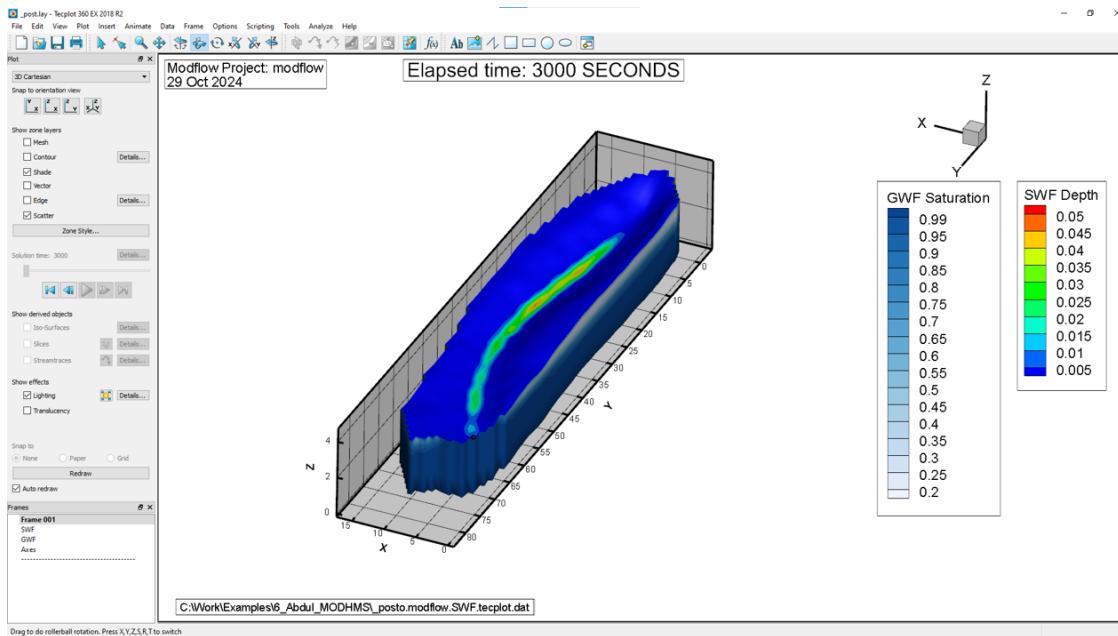
These are similar to those shown earlier during the model build except there are now multiple Zone(s), one for each output time. The Solution Time (6000) is shown for the highlighted zone, in this case zone 10.

Included are these cell properties:

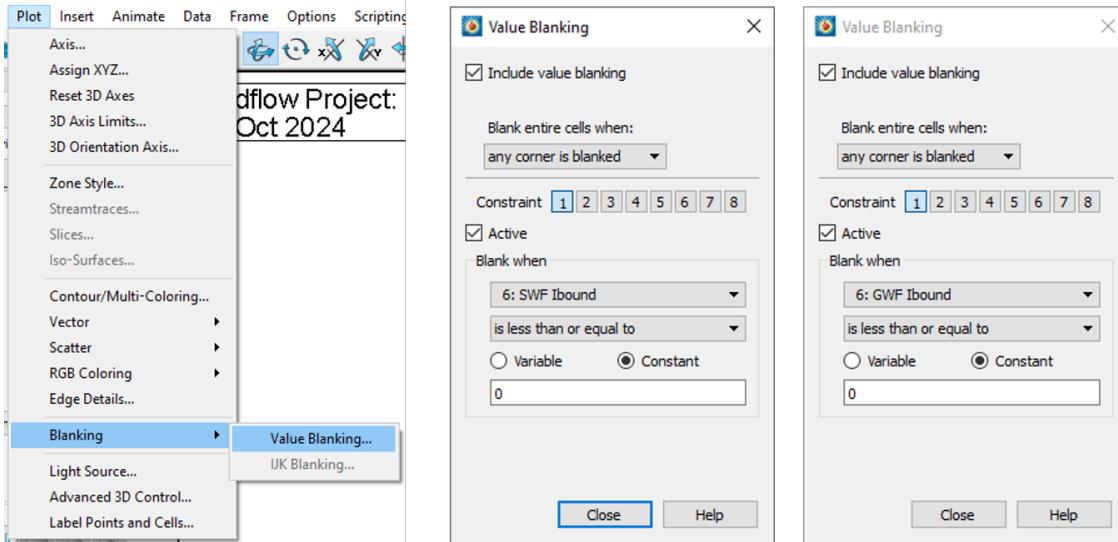
- Elevation SWF z Cell and GWF z Cell.
- MODFLOW layer number SWF layer and GWF layer.
- MODFLOW boundary number SWF lbound and GWF lbound.
- Initial head.
- Hydraulic head result.
- GWF saturation and SWF depth. These are stored in the MODFLOW DDN (drawdown) file.
- In this case, cell-by-cell flows are stored in GWF variables 10 to 13 and SWF variables 10 to 15.

#### 4.4.3 Inactive Cells and Value Blanking

The verification example 6\_Abdul\_MODHMS was generated from a 2D rectangular template mesh. To make it conform to the shape of the original triangular mesh used in example 6\_Abdul\_Prism\_Cell, a region of inactive cells was assigned around the outside of the domain. The layout file \_plot.lay has been configured to show only active cells:



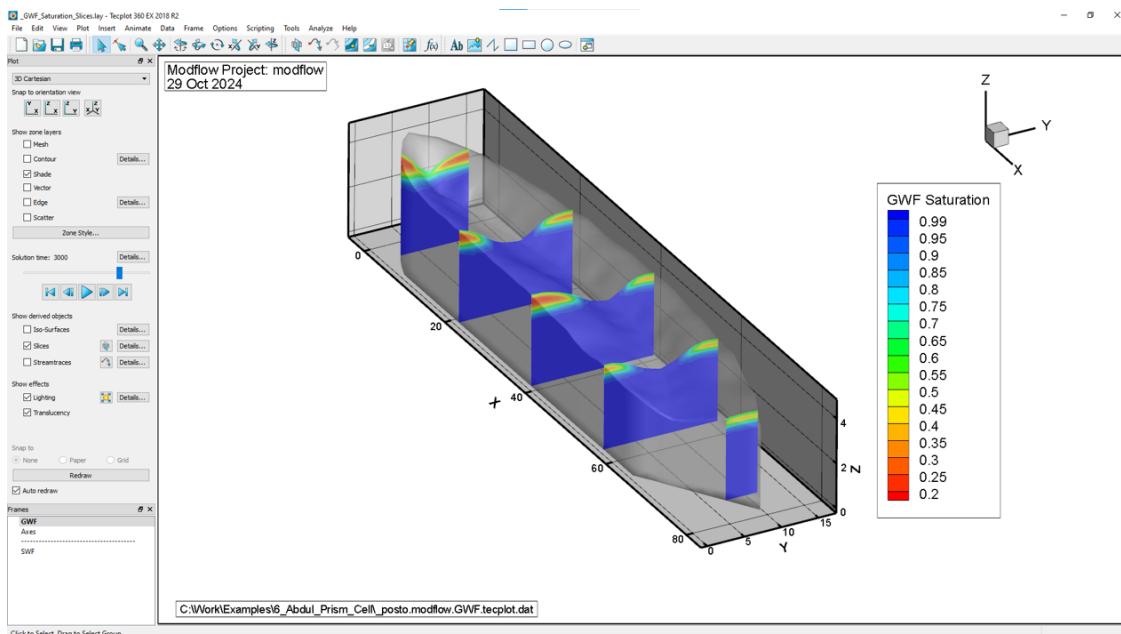
The layout uses TECPLLOT value-blanking (in this case based on the value of the blanking variables SWF lbound and GWF lbound) to prevent cells (or portions of cells) from appearing in the image. The Plot\Blanking\Value Blanking... menu option leads to the Value Blanking dialogue for the currently active frame:



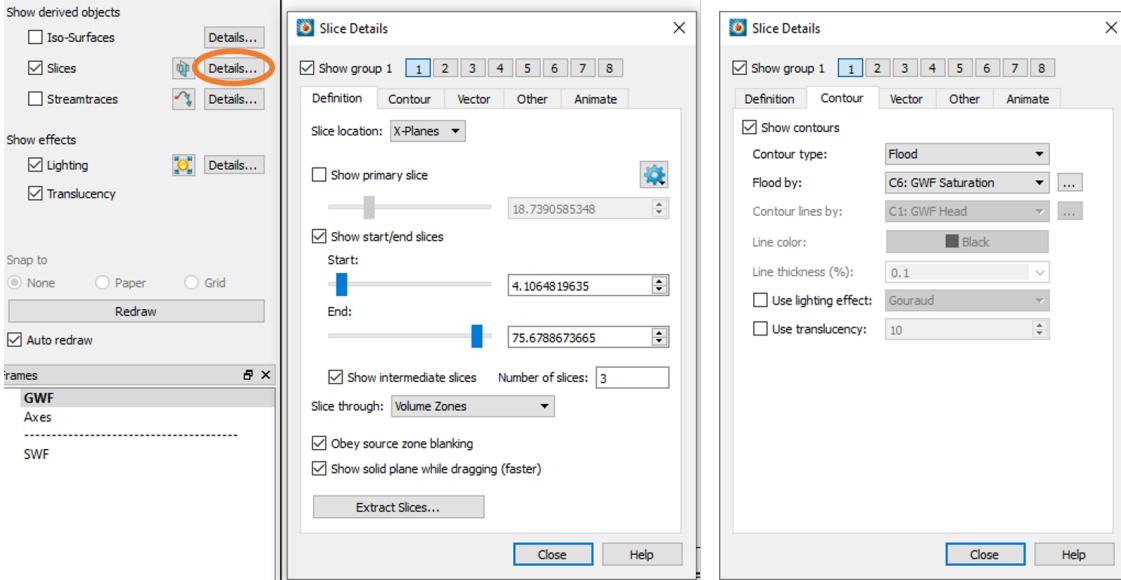
Shown here are the dialogues for both the SWF and GWF domains, where value blanking constraints are set to prevent SWF and GWF domain cells from being shown if the SWF Ibound or GWF Ibound value is less than or equal to zero.

#### 4.4.4 Slices and Fence Diagrams

The layout file `_GWF_Saturation_Slices.lay` has been configured to show a series of cross-sections in the GWF domain:



The layout uses the TECPLLOT derived object **Slices** to define the slice locations and content. The **Details...** button leads to the **Slice Details** dialogue for the currently active frame:



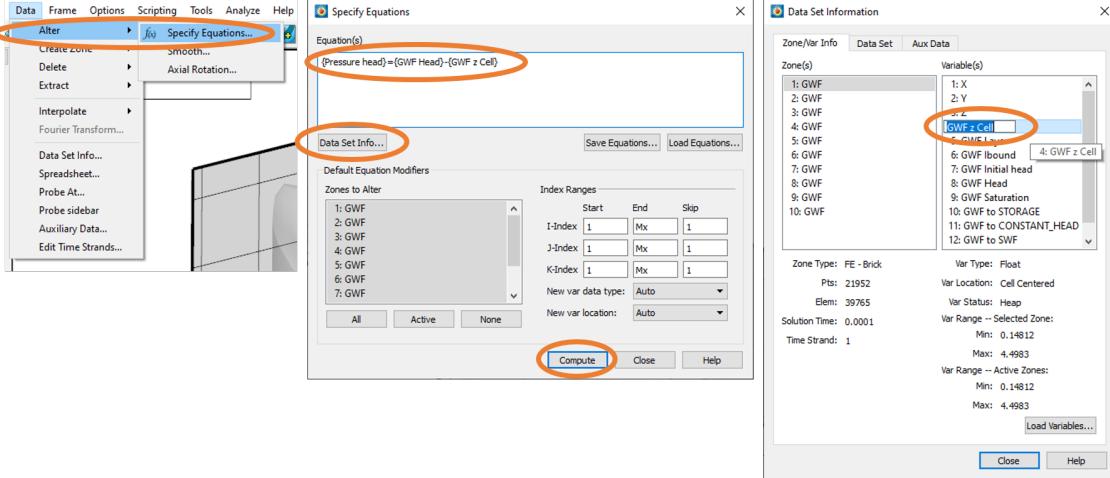
The Definition tab is configured to position the start and end slice positions and set the number of intermediate slices at 3.

The Contour tab is configured to contour-flood the slices by GWF Saturation value.

The Shade and Translucency boxes are checked to show the slice position relative to the model domain.

#### 4.4.5 Defining New Variables

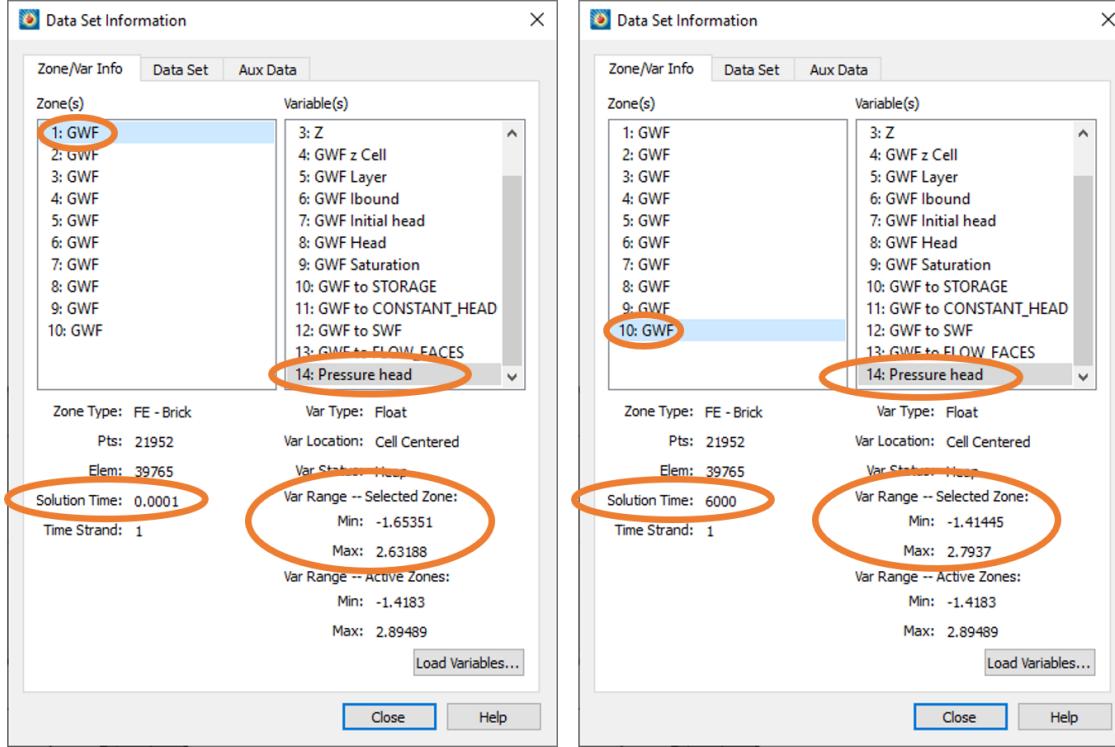
New variables can be defined in TECPLLOT using the Data\Alter\Specify equations... menu option which leads to the Specify equations dialogue:



This dialogue has an Equation(s) window where we have defined a new variable, in this case Pressure Head, that is set equal to two existing variables according to the mathematical expression GWF Head - GWF z Cell. In equation syntax, variable names must be placed within curly brackets: {}.

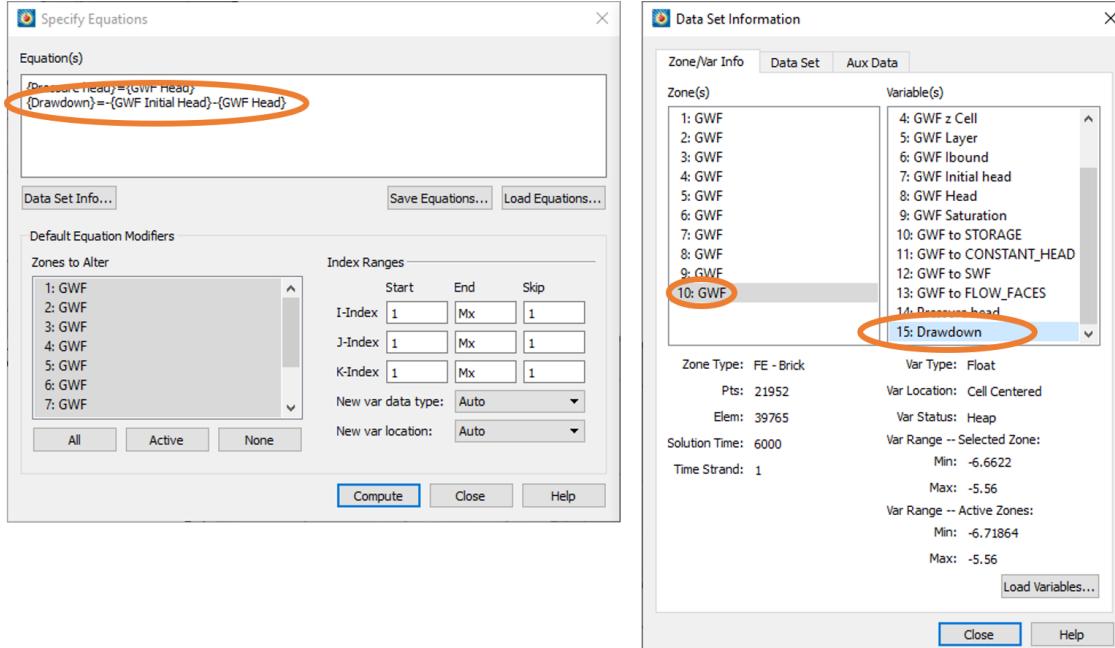
The Data Set Info... tab leads to the Data Set Information dialogue where variable names can be cut and pasted into the equation window. Here the GWF z Cell variable has been selected.

Once the equation is complete, click the Compute button to generate the new variable, which will now show up in the Data Set Information dialogue:



Here we see the variable 14:Pressure Head has been added. Because the GWF Head variable changes over time and is defined at every timestep, the calculated variable does also. The two panes above show the pressure head range at solution time 0.0001 is different at time 6000.

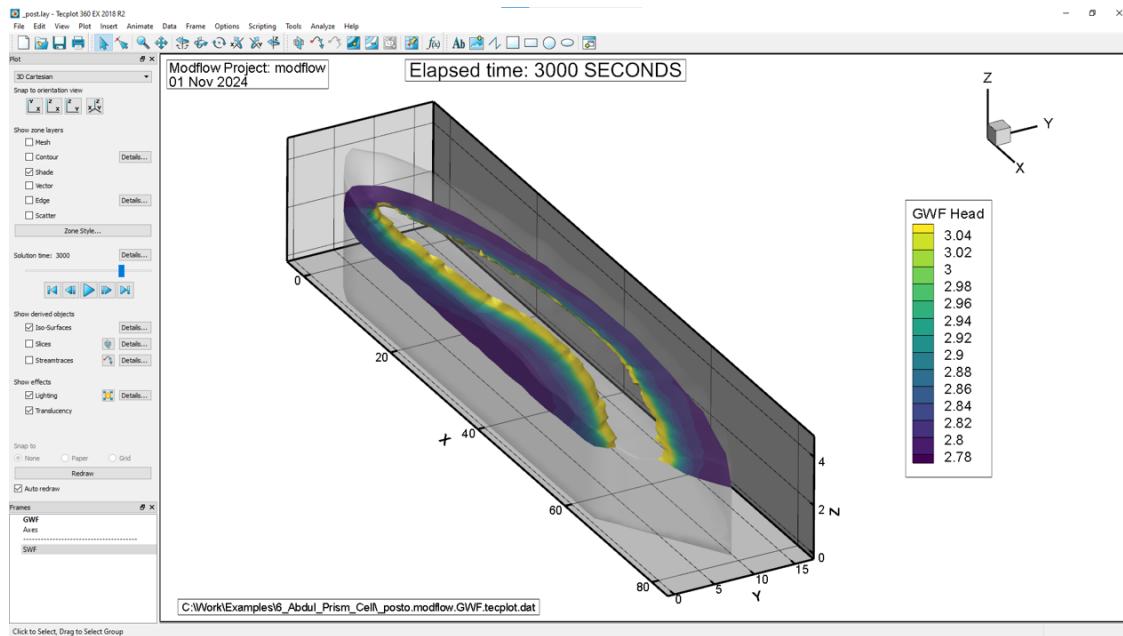
MODFLOW-USG<sup>Swf</sup> uses the MODFLOW DDN (i.e. drawdown) file to store saturation values for the GWF domain and surface water depth values for the SWF domain. If you want to plot drawdowns they can be calculated by adding an equation to the previous example:



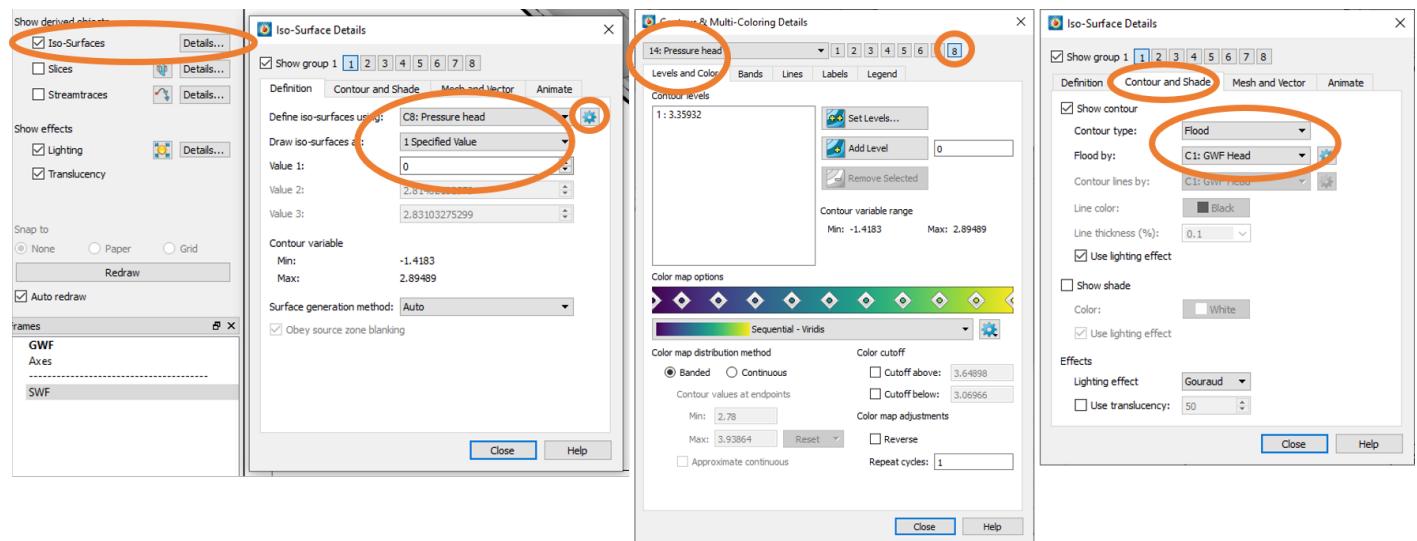
Here we see the equation for calculating the variable Drawdown has been added to the Equations(s) window and the variable 15:Drawdown appear in the Data Set Information dialogue. The range shown for Drawdown values is negative because simulated heads were always higher than the initial flat water table condition.

#### 4.4.6 Water Table Isosurface Plot

The layout file 6\_Abdul\_Prism\_Cell\\_Water\_Table.lay has been configured to show an isosurface of the water table in the GWF domain at a time of 3000 seconds :



The layout uses the TECPLLOT derived object Iso-surfaces to define the slice locations and content. The Details... button leads to the Iso-Surface Details dialogue for the currently active GWF frame:



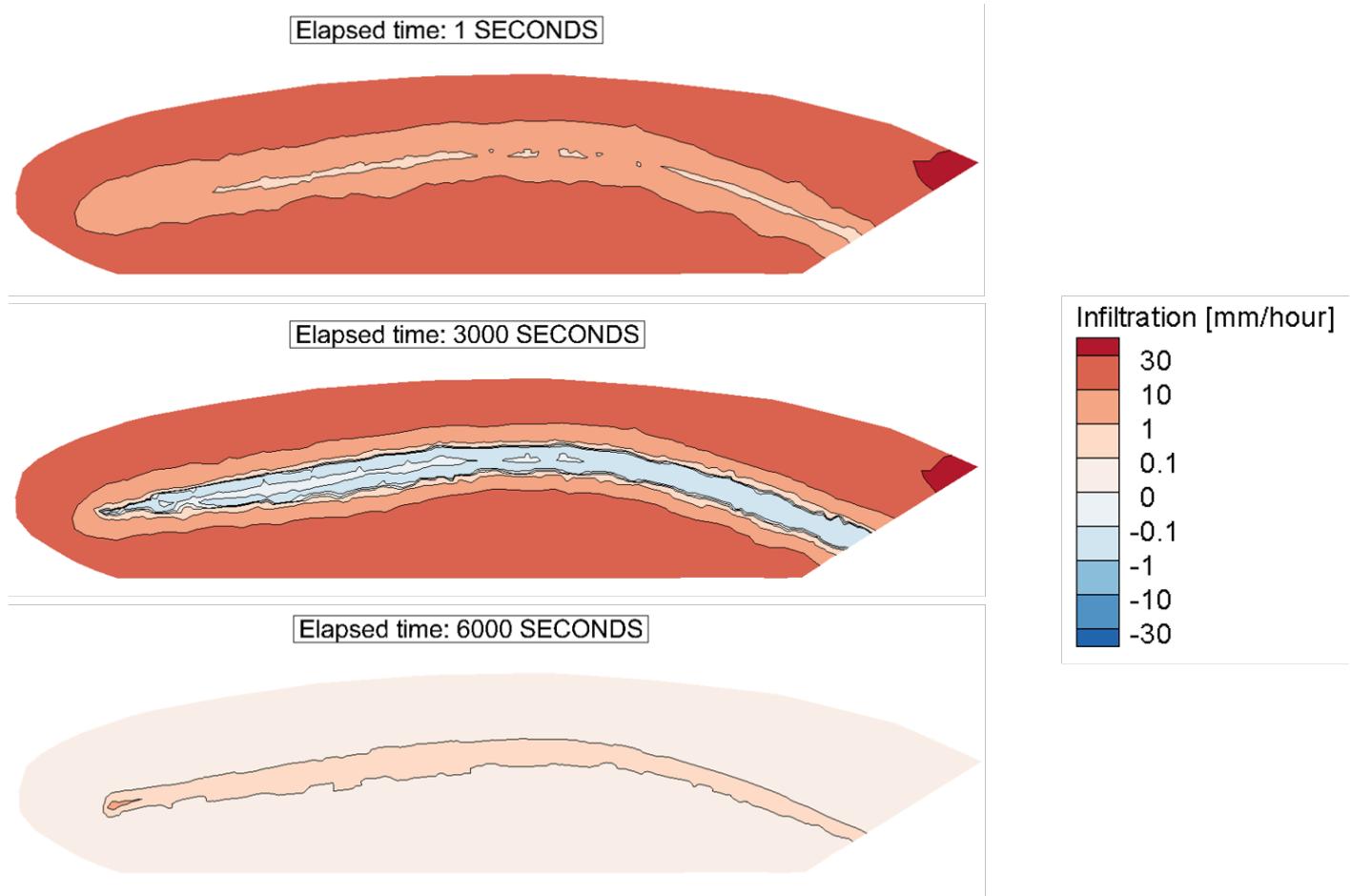
The Definition tab is configured to define iso-surfaces using the Pressure Head variable. This had to be defined as one of the eight contouring variables, and there is a short-cut to the Contour & Multi-Colouring

Details dialogue using the  button, where we set the eighth contouring variable (number 8 in orange circle) as 14:Pressure Head. By definition, the water table is the surface where the pressure head is equal to zero, so back on the Definitions tab we chose to draw iso-surfaces at one specified value: zero.

Finally, the Contours and shade tab is configured to contour flood the isosurface with the GWF Head variable.

#### 4.4.7 Infiltration Plot

The layout file 6\_Abdul\_Prism\_Cell\\_Infiltration defines a new variable Infiltration [mm/hour] and uses contour flood and lines to show it. The contour levels are set up with a red-blue colour map that is symmetric about the value zero. Red shading indicates zones of infiltration, while blue shading indicates zones of GWF discharge to the SWF domain.



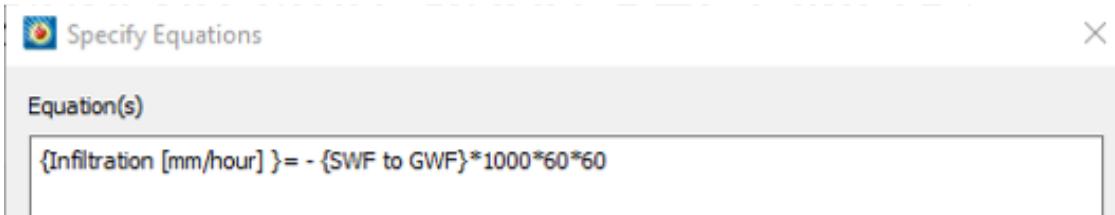
Here we can see the infiltration at 3 times:

**1 second** The application of water to the SWF domain has just started. The entire SWF domain is shaded red, showing that water is infiltrating everywhere, with the highest rate of infiltration occurring at the highest elevations in the domain (i.e. at hilltops)

**3000 seconds** Application of water stops. Infiltration is still occurring on the hills but discharge is occurring along most of the length of the swale, as shown by the blue shaded region.

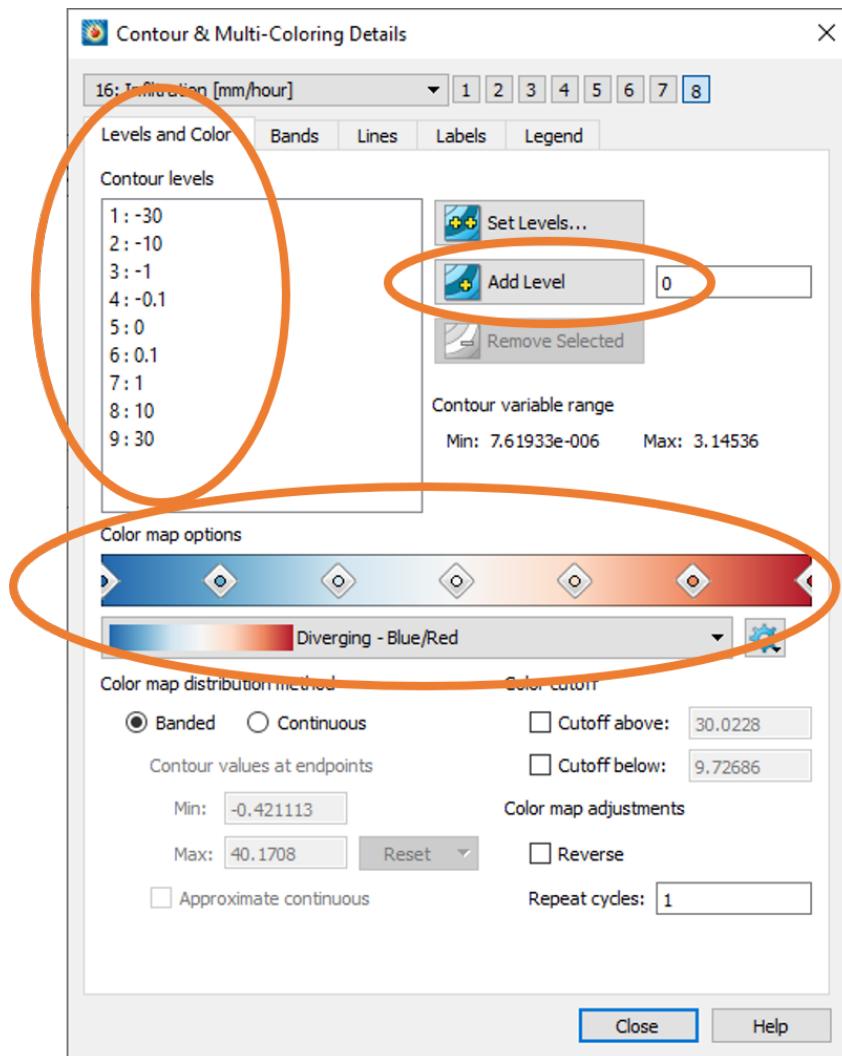
**6000 seconds** End of simulation. The stream has drained and water again infiltrating over the entire domain.

Here, the new variable Infiltration [mm/hour] is calculated by the following equation in TECPLT



The variable SFW to GWF contains the cell-by-cell flows from the SWF domain to the GWF domain, with negative values indicating a loss relative to the SWF domain (i.e. infiltration). We want infiltration to be shown as a positive value in this case so we add a negative sign at the beginning of the equation. The final part of the equation \*1000\*60\*60 converts the value from units of meters per second to millimetres per hour.

The contour levels were set up as shown here:



The levels were entered manually using the Add Level and Remove selected buttons. The value zero is

defined at the boundary between blue and red colours by using an equal number of positive and negative values with the same absolute range.

The color map is set to Diverging - Blue/Red using the drop-down menu.

# Chapter 5

## Model Verification

### 5.1 1D Variably-saturated Flow in a Column

This example <sup>1</sup> simulates variably-saturated flow in a 1D column of homogeneous sand 100 m thick. Parameter values used in this example are shown in Table 5.1.

The Van Genuchten unsaturated function type was used.

A uniform rainfall of 0.4 m/s was applied to the top of the column and the base was fixed at a pressure head of -100 m.

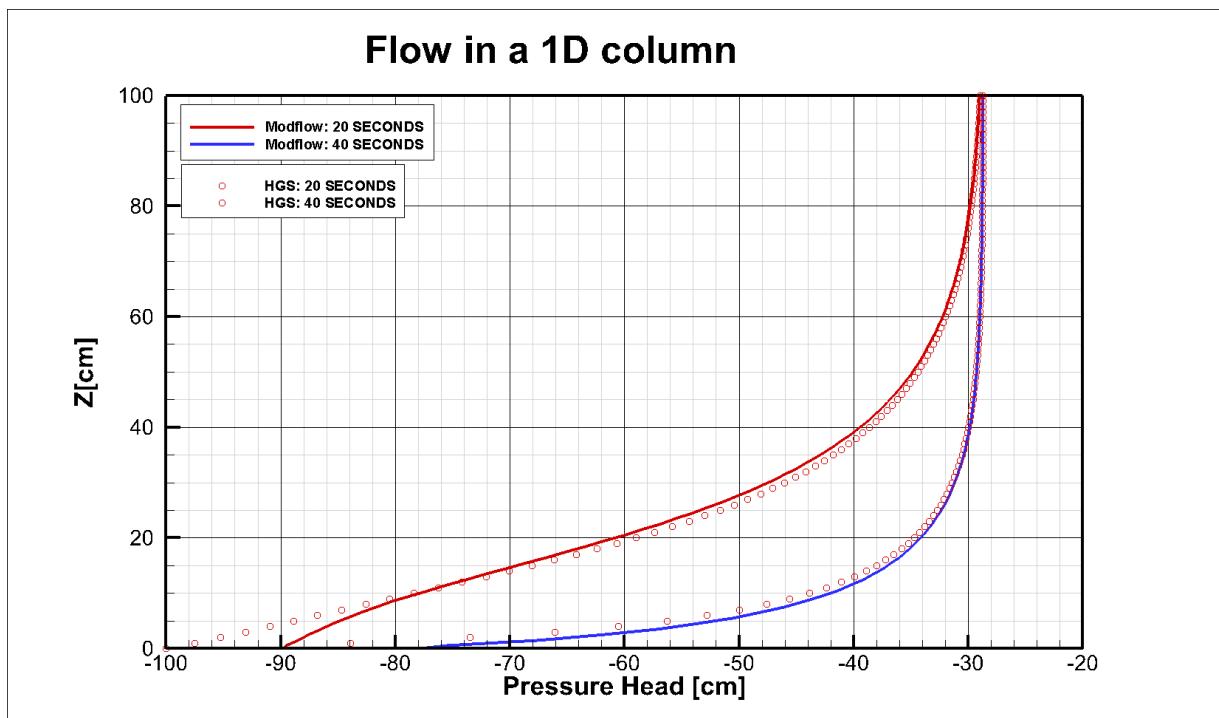
---

<sup>1</sup>See verification example MUT\_Examples\1\_VSF\_Column

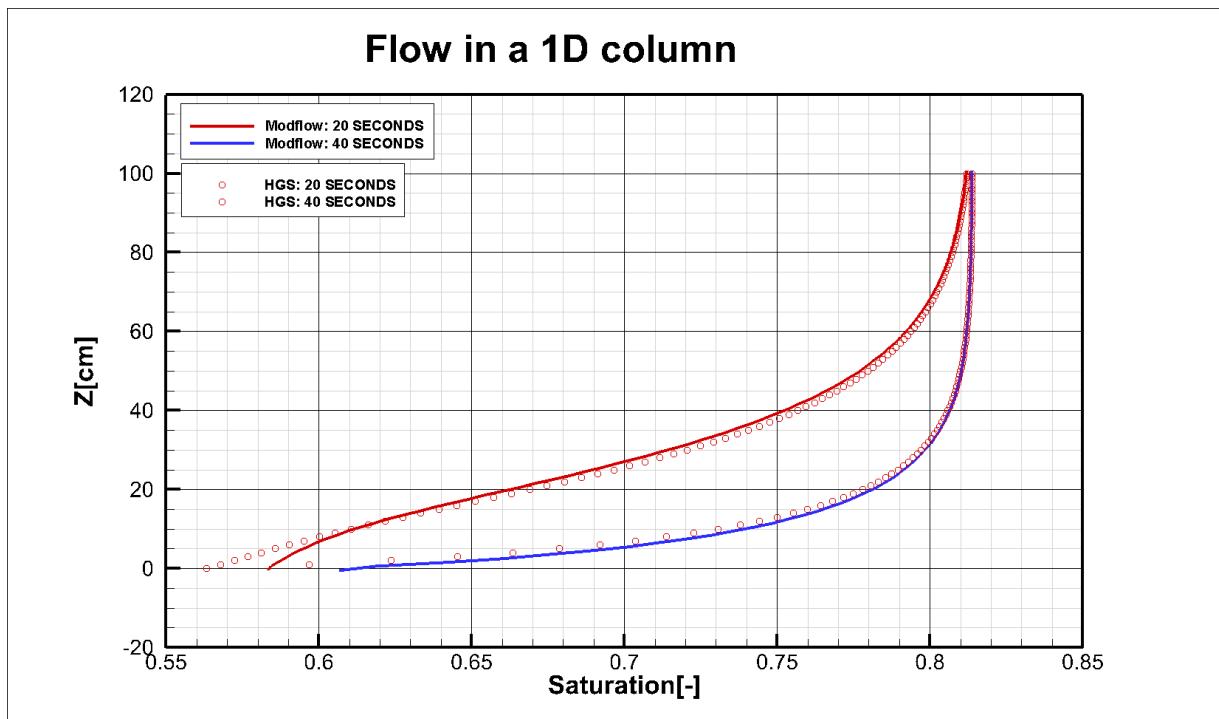
Table 5.1: Parameter Values for Simulation of the 1-D Column.

Parameter	Value	Unit
specific yield (porosity)	0.34	
hydraulic conductivity	$1 \times 10^{-5}$	$\text{m s}^{-1}$
specific storage coefficient	$1.2 \times 10^{-7}$	$\text{m}^{-1}$
Van Genuchten parameter	1.9	$\text{m}^{-1}$
Van Genuchten parameter	6	
residual saturation	0.18	
Manning coefficient for plot	0.3	$\text{s m}^{-1/3}$
Manning coefficient for channel	0.03	$\text{s m}^{-1/3}$
Initial water table elevation	2.78	m

Here is a comparison of pressure head versus depth results for MODFLOW-USG<sup>*Swf*</sup> and HYDROGEO-SPHERE at 20 and 40 seconds:



Here is a comparison of saturation versus depth results for MODFLOW-USG<sup>*Swf*</sup> and HYDROGEOSPHERE at 20 and 40 seconds:



## 5.2 2D Variably-saturated Flow in a Hillslope: Drains vs Surfacewater Flow

This example <sup>2</sup> simulates variably-saturated flow in a 2D hillslope of homogeneous material which receives a uniform recharge rate of  $1.27 \times 10^{-8}$  m/s was applied directly to the GWF domain at the ground surface. .

These are the properties we used:

Variable name	Value	Units
Kh_Kx:	1.0E-06	m/s
Kv_Kz:	1.0E-06	m/s
Specific Storage:	1.0E-05	1/m
Specific Yield:	0.1	-
Alpha:	3.34E-02	1/m
Beta:	1.982	1/m
Sr:	0.2771	-

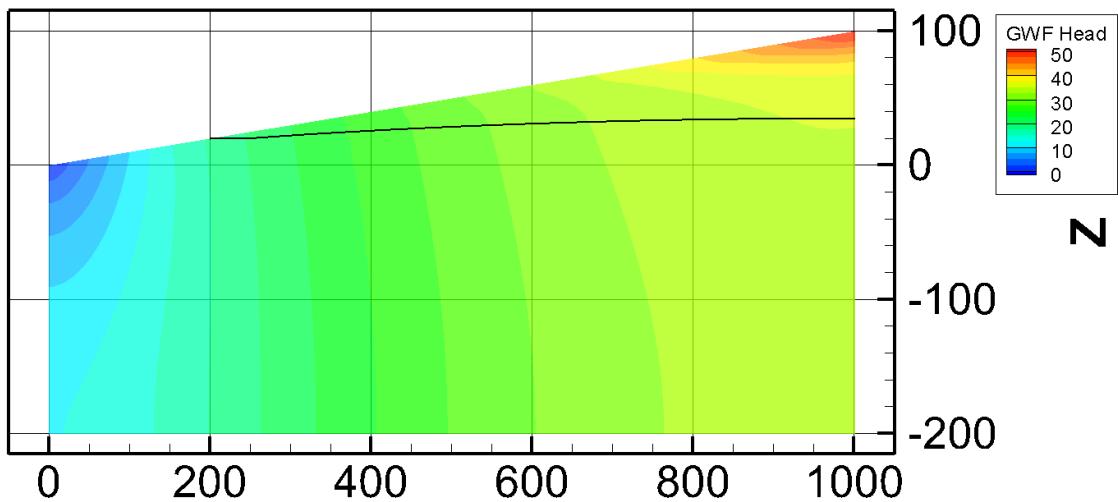
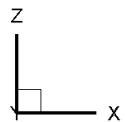
The Van Genuchten unsaturated function type was used.

A drain boundary condition was with a drain conductance of 1000 m/s was applied to the GWF domain at the ground surface.

Here is the hydraulic head distribution at 242 years (equilibrium):

---

<sup>2</sup>See verification example MUT\_Examples\2\_VSF\_Hillslope



C:\temp\TestExamples\2\_VSF\_Hillslope\posto.modflow.GWF.tecplot.dat

Some key features of this example are:

- The hill slopes from an elevation of 0 m at  $x = 0$  to 1000 m at  $x = 1000$ .
- The base is flat at an elevation of -200 m.
- The water table is shown as heavy black line.

### 5.3 3D Fully-coupled Groundwater-Surface Water Flow: Abdul's Experiment

Properties from the HGS manual are shown in Table 5.2.

Table 5.2: Parameter Values for Simulation of the 3-D Field Scale Study of *Abdul* [1985].

Parameter	Value	Unit
porosity, $\Theta$	0.37	
hydraulic conductivity, $K$	$1 \times 10^{-5}$	$\text{m s}^{-1}$
storage coefficient, $S_s$ ,	$1.2 \times 10^{-7}$	$\text{m}^{-1}$
Van Genuchten parameter, $\alpha$	1.9	$\text{m}^{-1}$
Van Genuchten parameter, $\beta$	6	
residual saturation, $S_r$	0.18	
Brooks-Corey coefficient, $n$	3.4	
Manning coefficient for plot	0.3	$\text{s m}^{-1/3}$
Manning coefficient for channel	0.03	$\text{s m}^{-1/3}$
Initial water table elevation	2.78	m

# Appendix A

## Microsoft Excel Database Files

MICROSOFT EXCEL files are used to store information in the following databases:

- `GWF.xlsx` for GWF domain material parameters.
- `SWF.xlsx` for SWF domain material parameters.
- `CLN.xlsx` for CLN domain material parameters.
- `SMS.xlsx` for solver parameters.
- `ET.xlsx` and `LAI.xlsx` for evapotranspiration (ET) and leaf area index (LAI) parameters respectively. *These are currently just placeholders for future development and will not be discussed further at this time.*

Modifications can be made to the database by editing the `.xlsx` file in MICROSOFT EXCEL and exporting the results to a `csv`-formatted version of the file which is then read and processed by MUT. We will use the GWF database to illustrate the modification workflow, which can then be applied to the other databases.

If you are a MUT end-user, you should edit the database files in the `USERBIN` directory, as outlined on page [7](#).

If you are a developer and have downloaded the `MUT_Examples` repository, you should edit the database files in the directory `patho\Grdbldr\MUT_Examples\_MUT_USERBIN`, where the string `patho` represents the local path to the repository (e.g. `c:\repos`).

First, open the `GWF.xlsx` in MICROSOFT EXCEL:

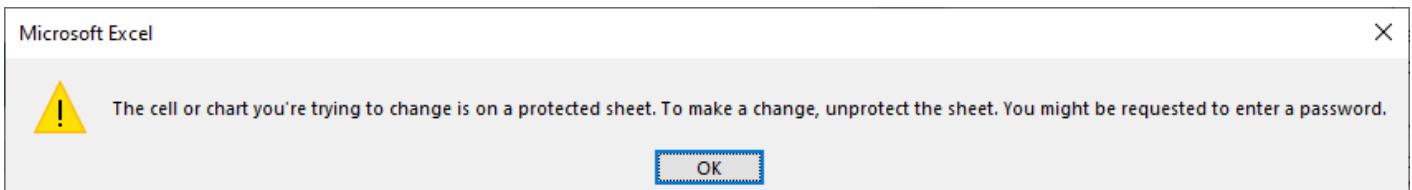
B11 : 2D Hillslope seconds

Material	ID	Material name	Porosity	Kh (Kx)	Kv (Kz)	Ky	Specific storage	Specific Yield	Unsaturated Function Type	Alpha	Beta	Sr	Brooks-Corey Exponent	Length Unit	Time Unit	Notes
1	1	1D Column	0.3	10	10	0	1E-07	0.43	Van Genuchten	0.036	1.56	0.1814	6.5714	CENTIMETER DAYS		Rob copied these properties from
2	2	1D Column Brooks	0.3	10	10	0	1E-07	0.43	Brooks-Corey	0.036	1.56	0.1814	6.5714	CENTIMETER DAYS		-
3	3	2D Hillslope	0.1	31.536	31.536	31.536	1E-07	0.1	Van Genuchten	3.34	1.982	0.2771	6.5714	METERS YEARS		-
4	4	2D Hillslope 100 m len	0.3	0.000147	0.000147	0	0.0001	0.3	Van Genuchten	1	5	0.3	1	METERS SECONDS		-
5	5	3D Abduls Problem	0.34	0.00001	0.00001	0	1.2E-07	0.34	Van Genuchten	1.9	6	0.18	1	METERS SECONDS	Properties from Joel Vanderkwa	
6	6	Borden sand	0.34	0.00001	0.00001	0	1.2E-07	0.34	Van Genuchten	1.9	6	0.18	1	METERS SECONDS	Properties from Joel Vanderkwa	
7	7	Sand	0.3	0.00001	0.00001	0	0.0001	0.3	Van Genuchten	1	5	0.3	1	METERS SECONDS	Made up for illustrative purposes	
8	8	Clay	0.5	1E-08	1E-08	0	0.0001	0.5	Van Genuchten	1	5	0.5	1	METERS SECONDS	Made up for illustrative purposes	
9	9	Silt	0.4	0.000001	0.000001	0	0.0001	0.4	Van Genuchten	1	5	0.5	1	METERS SECONDS	Made up for illustrative purposes	
11	10	2D Hillslope seconds	0.1	31.536	31.536	0.00001	0.00001	0.1	Van Genuchten	0.0334	1.982	0.2771	1	METERS SECONDS	Rob copied these properties from	
12																
13																

Some key features to note are:

- The first row contains database field (i.e. column) names.
- The existing database contains data for 10 materials stored in rows 2 to 11.

As a precaution, the database is protected. If you try to modify it you will receive the following warning:



To unprotect the sheet, choose 'Review', then 'Unprotect sheet':

GWF.xlsx - Excel

Data Review View Help Tell me what you want to do

Translate New Delete Previous Next Show/Hide Comment Show All Comments Unprotect Sheet Protect Workbook Allow Edit Ranges Unshare Workbook

Comments

Kv (Kz)	Ky	Specific storage	Specific Yield	Unsaturated Function Type	Alpha	Unprotect Sheet	Length Unit
.0	10	0	1E-07	0.43	Van Genuchten	0.0	CENTI
.0	10	0	1E-07	0.43	Brooks-Corey	0.0	CENTI
.6	31.536	31.536	1E-07	0.1	Van Genuchten	3.34	METER
.7	0.000147	0	0.0001	0.3	Van Genuchten	1.9	METER
11	0.00001	0	1.2E-07	0.34	Van Genuchten	1	METER
11	0.00001	0	1.2E-07	0.34	Van Genuchten	1.9	METER
11	0.00001	0	0.0001	0.3	Van Genuchten	1	METER

You will be prompted to enter a password, which is 'mut'.

To protect the sheet, choose 'Review', then 'Protect sheet', then enter and confirm a password. We suggest you use the 'mut' password. If not, don't forget the new password!

The purpose for protecting the database is to prevent accidental changes from being made to critical material properties so future runs referring to protected material ID's yield repeatable results. This would be the case for material properties calibrated for an engineering project or verification example material properties, which occupy the first 5 materials in the GWF database).

Although the database is protected, you can add your own materials starting at row 12. The easiest way to add a material is to copy an existing one. For example, we can copy row 10 (silt) to row 12:

1. Click on the number 10 at the left end of row 10 to select the entire row.

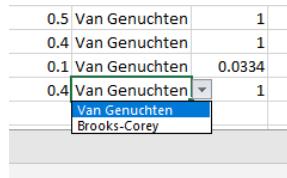
9	8 Clay	0.5	1E-08	1E-08	0	0.0001	0.5 Van Genuchten	1	5	0.5	1 METERS	SECONDS	Made up for illustrative purposes
10	9 Silt	0.4	0.000001	0.000001	0	0.0001	0.4 Van Genuchten	1	5	0.5	1 METERS	SECONDS	Made up for illustrative purposes
11	10 2D Hillslope seconds	0.1	31.536	31.536	0.00001	0.00001	0.1 Van Genuchten	0.0334	1.982	0.2771	1 METERS	SECONDS	Rob copied these properties from
12													
13													

2. Type 'CTRL-C' to copy it.
3. Click on the number 12 at the left end of row 12 to select the entire row.
4. Type 'CTRL-V' to paste it.

10	9 Silt	0.4	0.000001	0.000001	0	0.0001	0.4 Van Genuchten	1	5	0.5	1 METERS	SECONDS	Made up for illustrative purposes
11	10 2D Hillslope seconds	0.1	31.536	31.536	0.00001	0.00001	0.1 Van Genuchten	0.0334	1.982	0.2771	1 METERS	SECONDS	Rob copied these properties from
12	11 Silt	0.4	0.000001	0.000001	0	0.0001	0.4 Van Genuchten	1	5	0.5	1 METERS	SECONDS	Made up for illustrative purposes
13													

Note that the new material ID number is automatically set to 11, which is the previous number of materials plus 1. You should now change the material name and properties as desired.

Columns I, N and O are special fields where the input is restricted by a list of choices. For example, if you click on cell I12, the unsaturated function type for the new material, you will see a drop-down list arrow appear beside the cell. To choose, for example, the 'van Genuchten' function type, select the arrow and highlight the option in the list and press enter:



The second worksheet, 'Options', contains the following data:

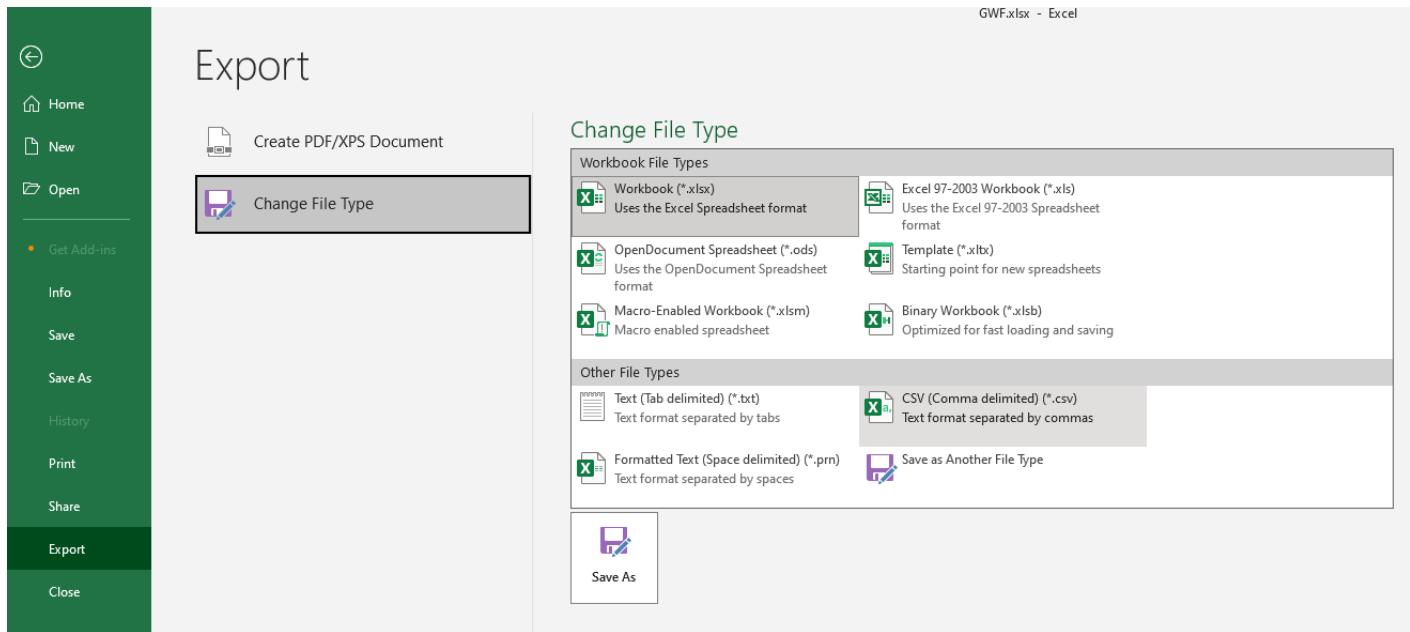
	A	B	C	D
1	Length	Time	Mass	Unsaturated Function Type
2	FEET	SECONDS	GRAM	Van Genuchten
3	METERS	MINUTES	KILOGRAM	Brooks-Corey
4	CENTIMETERS	HOURS		
5		DAYS		
6		YEARS		
7				
8				
9				
10				
11				
12				
13				

GWF Options

These are the drop-down lists for the unsaturated function type, length unit, time unit in fields I, N and O respectively. This worksheet is also protected with the password 'mut'.

Once you have finished modifying the material properties, you should save the **.xlsx** file, then export them to a **.csv** file by:

1. Selecting File/Export/Change File Type/CSV:



2. Double-clicking CSV or clicking Save As.

# Index

## CLN Domain

- boundary conditions, 49
- cell connection properties, 48
- initial condition
  - initial (starting) depth, 49
- material properties, 48
- \_build.lay
  - CLN frame, 72

## GWF Domain

- boundary conditions, 39
- cell connection properties, 28
- initial condition
  - initial (starting) head, 38
- layering, 25
- material properties, 29
- \_build.lay
  - GWF Cells frame, 63
  - GWF frame, 58

## SWF Domain

- boundary conditions, 44
- cell connection properties, 41
- initial condition
  - initial (starting) depth, 43
- material properties, 41
- \_build.lay
  - SWF frame, 66

## TECPLOT

- Custom labels, 66
- data set information, 60
- Equations and defining new variables, 85
- frame order, 57
- Infiltration Plot, 88
- Isosurfaces, Water Table Plot, 87
- probe tool, 60
- Slices and Fence diagrams, 84, 87
- value blanking, 83
- Zoom Tool, 69

generate uniform rectangles, 21

Grid generation

## GWF Domain

- New layer, 25

## Input instructions

- 2d mesh from gb, 20
- active domain, 30
- build modflow usg, 17
- choose all cells, 29
- choose all nodes, 45
- choose all zones, 35
- choose cell at xyz, 30
- choose cells by chosen zones, 36
- choose cells by layer, 30
- choose cells from file, 30
- choose cells from gb elements, 31
- choose cells from gb nodes, 31
- choose gb nodes, 46
- choose node at xyz, 45
- choose zone number, 35
- chosen cells use gwf material number, 32
- chosen zones use cln material number, 49
- chosen zones use swf material number, 43
- clear chosen cells, 31
- clear chosen nodes, 46
- clear chosen zones, 35
- cln from xyz pair, 47
- cln initial depth, 49
- cln materials database, 48
- deltat, 51
- duration, 50
- elevation constant, 24
- elevation from gb file, 24
- elevation from list file, 24
- elevation from xz pairs, 25
- generate cln domain, 47
- generate layered gwf domain, 23
- generate output control file, 53
- generate swf domain, 41
- generate uniform rectangles, 21
- gwf alpha, 35

gwf beta, 35  
gwf brooks, 35  
gwf constant head, 39  
gwf drain, 40  
gwf initial head, 38  
gwf initial head function of z, 39  
gwf kh, 34  
gwf kv, 34  
gwf materials database, 32  
gwf recharge, 40  
gwf sr, 35  
gwf ss, 34  
gwf sy, 34  
Layer name, 26  
Minimum layer thickness, 26  
new layer, 26  
nodal control volumes, 22  
number of timesteps, 50  
Offset base, 27  
postprocess existing modflow model, 77  
Proportional sublayering, 27  
sms database, 55  
sms parameter set number, 55  
stress period, 50  
swf constant head, 44  
swf critical depth, 45  
swf critical depth with sidelength1, 44  
swf depression storage height, 42  
swf depth for smoothing, 42  
swf initial depth, 43  
swf manning, 42  
swf materials database, 43  
swf obstruction storage height, 42  
swf recharge, 44  
swf to gwf connection length, 41  
tadjat, 51  
tcutat, 51  
tmaxat, 51  
tminat, 51  
top elevation, 24  
type, 50  
Uniform sublayering, 26  
units of length, 18  
units of time, 19  
Zone by template, 23  
Installation, 5

Model Build  
    visualization, 56  
Volumetric water budget, 77