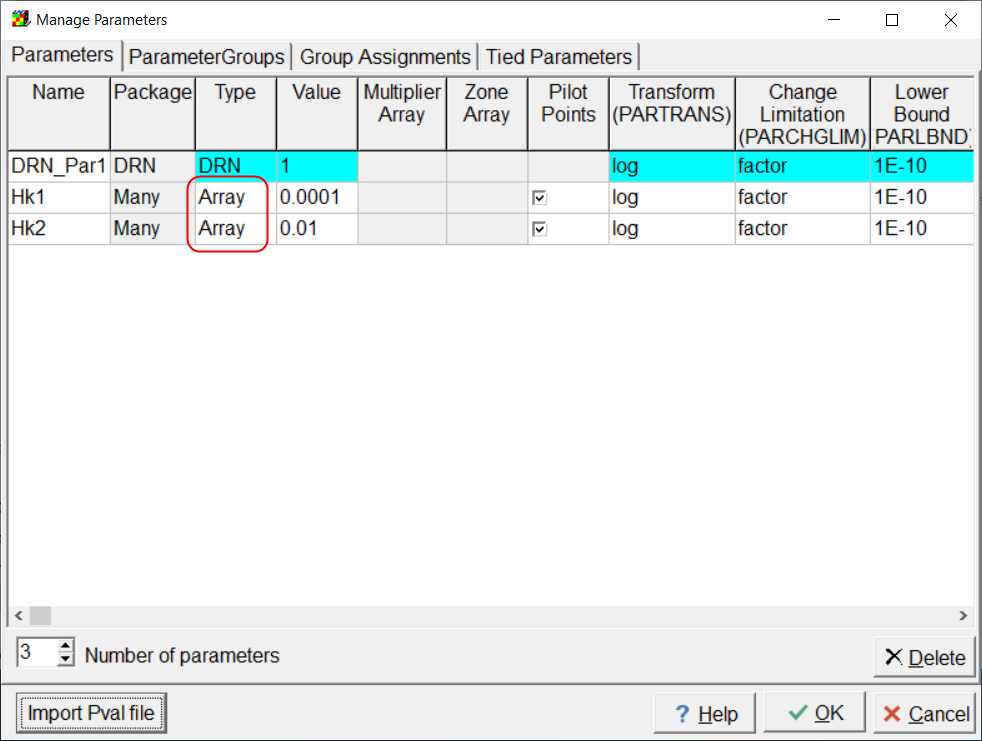
ModelMuse with Support for PEST – Beta 4

This version of ModelMuse adds support for the use of pilot points in PEST.

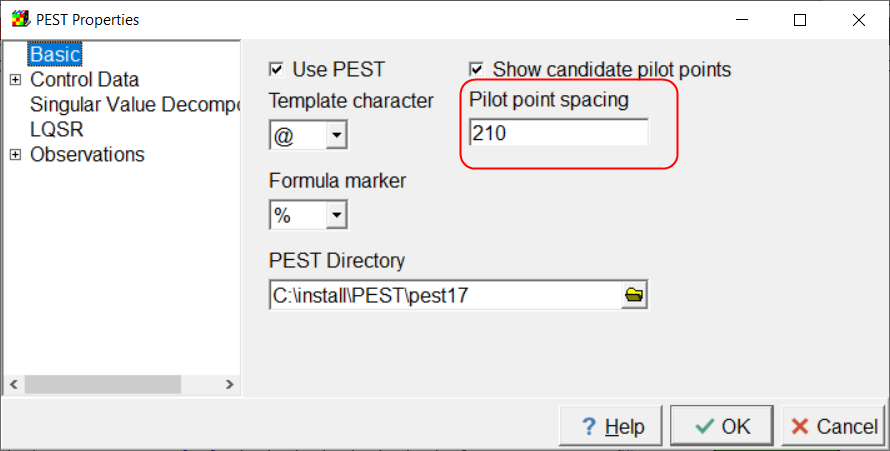
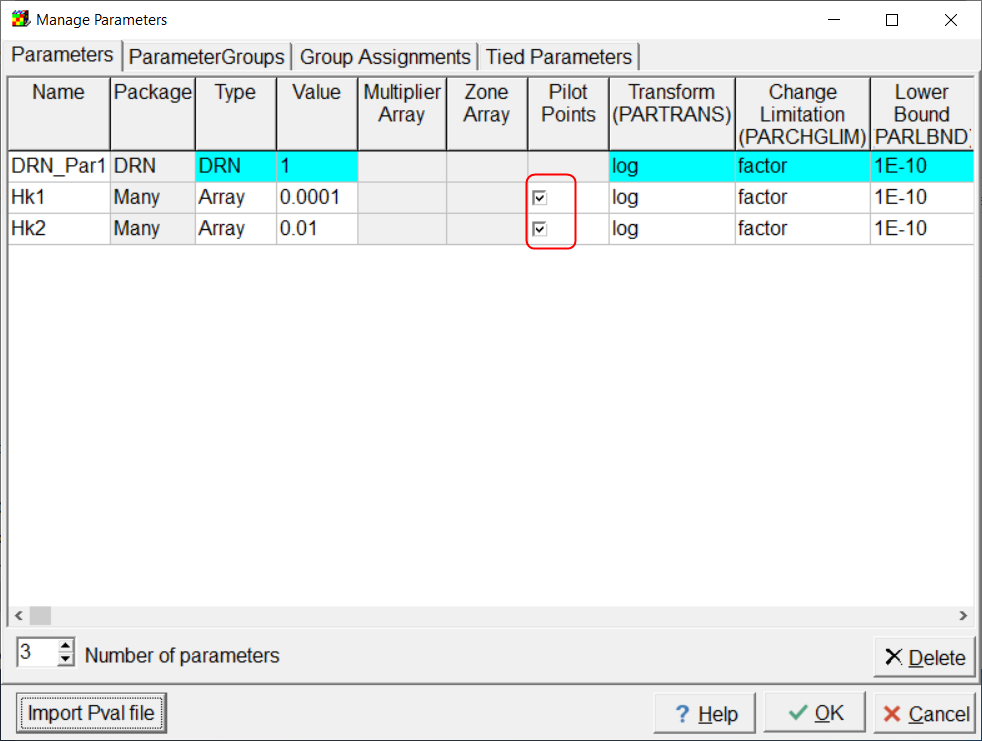
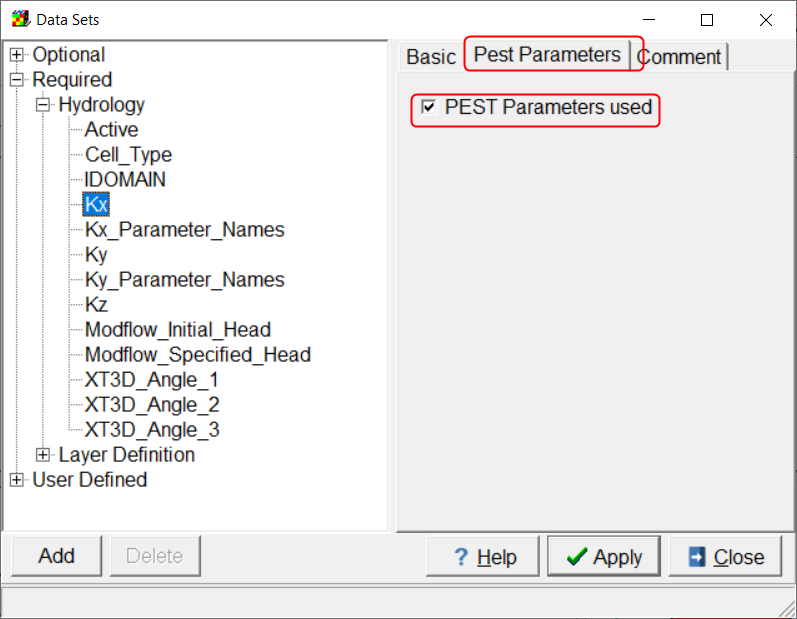
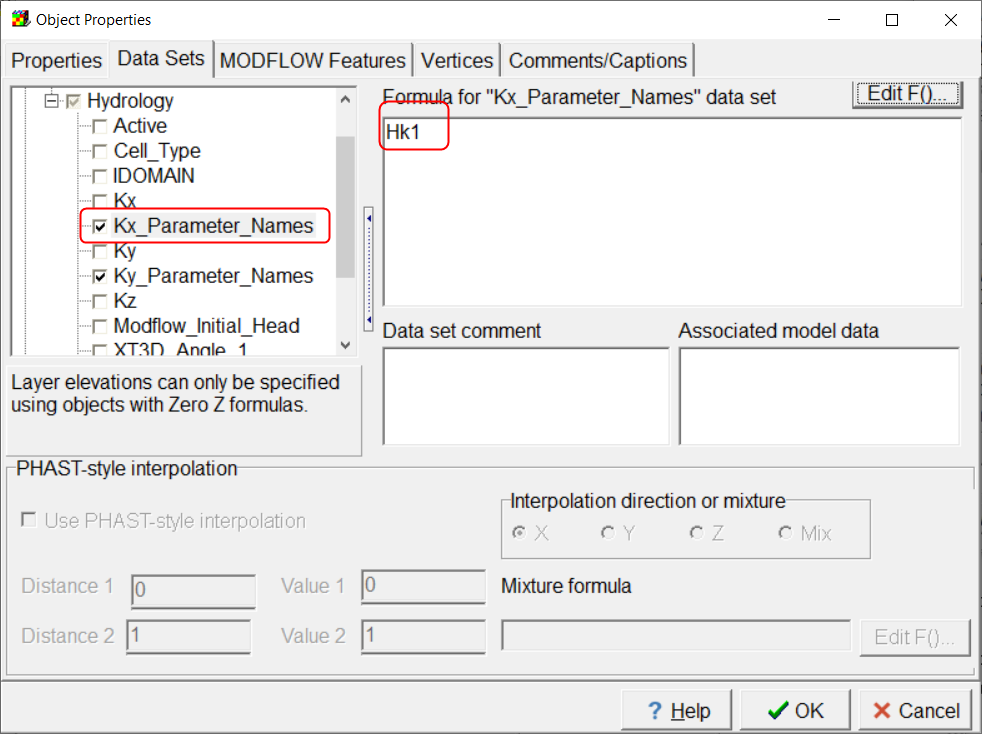
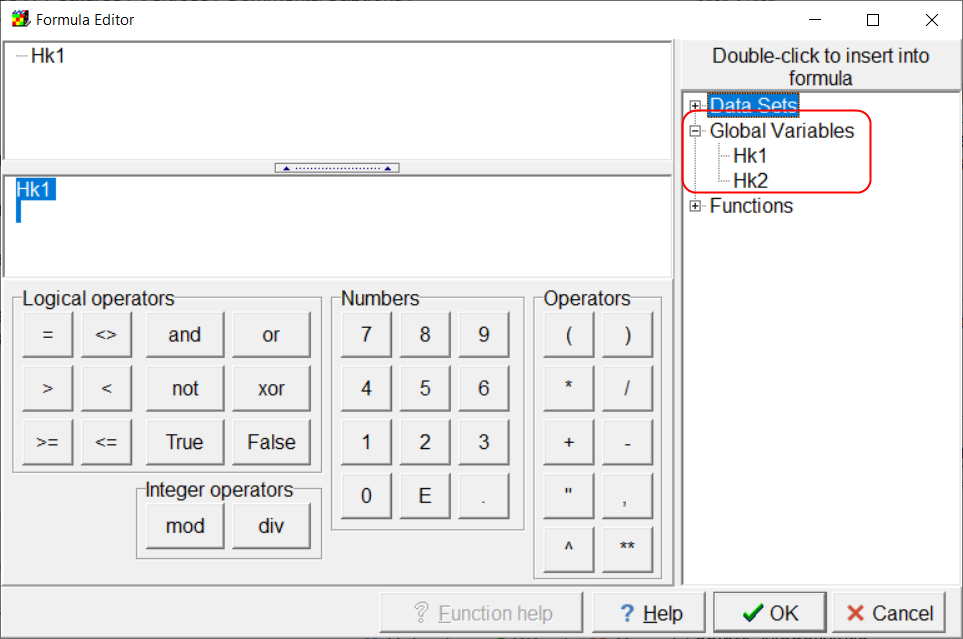
# Changes from Beta 3

1. In the previous version, one of the types of parameter was called PEST. This caused some confusion because some people thought that PEST could only calibrate parameters whose type was “PEST” whereas actually, PEST could be used with any type of parameter. I changed the name of the type from “PEST” to “Array” in the hopes that this would be less confusing. The name may be changed again in the future.
2. For each Array (formerly PEST) parameter, ModelMuse now creates a global variable whose name and value are both the name of the parameter. You can use this global variable in assigning where a parameter will be applied. If you change the name of the parameter, the global variable will be automatically updated so that you don’t need to update the parameter names assigned to cells. Using these global variables ensures that misspelled parameter names won’t be a problem.

# New Features in Beta 4

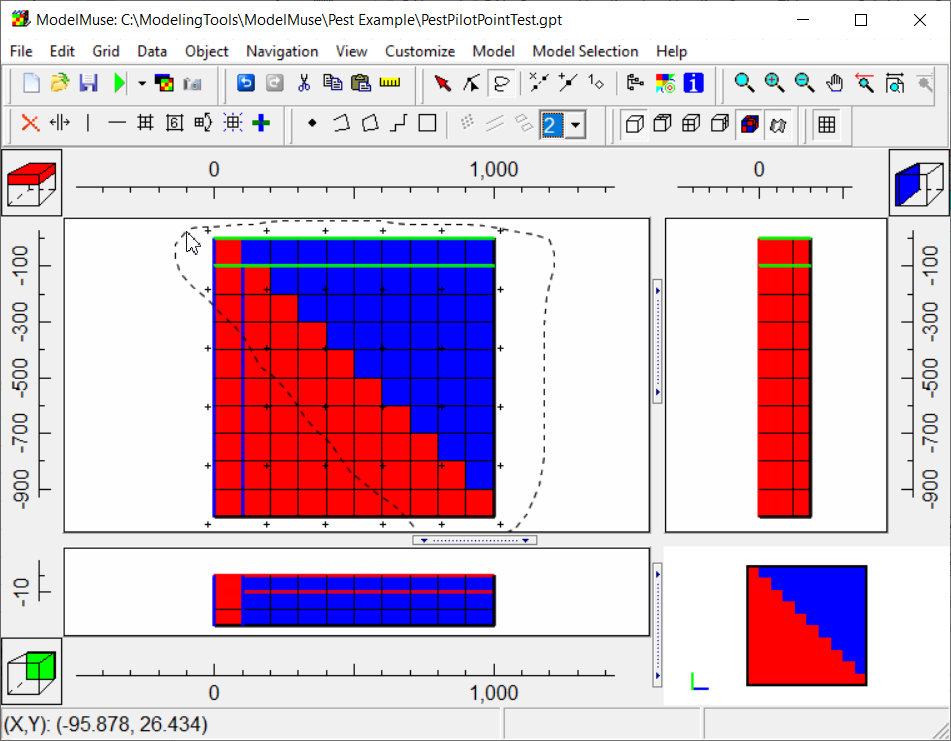
The main change in ModelMuse Beta 4 is that pilot points are (finally) supported. There is a caveat, however; this version doesn’t have any built-in mechanism for including prior information related to the pilot points. If you want to use pilot points with this version of ModelMuse, be sure to check out GENREG in the PEST Groundwater Utilities or some of the programs in the PEST Utility Support software. They can help add the prior information you will need to the PEST control file.

To include pilot points in your simulation, you need to do several things.

1. Set the pilot point spacing in the Model|PEST Properties dialog box to a value greater than zero. Normally, this will be a value that is larger than the grid or mesh spacing and small enough so that there aren’t too many pilot points. 
2. For any parameters for which you wish to use pilot points check the “Pilot Points” check box in the “Model|Manage Parameters” dialog box.
3. In the Data|Edit Data Sets dialog box, check the check box for “PEST Parameters used” for a data set that you wish to calibrate. 
4. You must assign the parameter names to particular cells. If there is more than one parameter for a data set, the easiest way to do this is with objects. In the first illustration below, the global variable Hk1 is used in an object to assign the Hk1 parameter to some cells for the Kx data set. In the second illustration below, you can see that there are two global variables named Hk1 and Hk2. These correspond to the Array parameters named Hk1 and Hk2.  
     
   

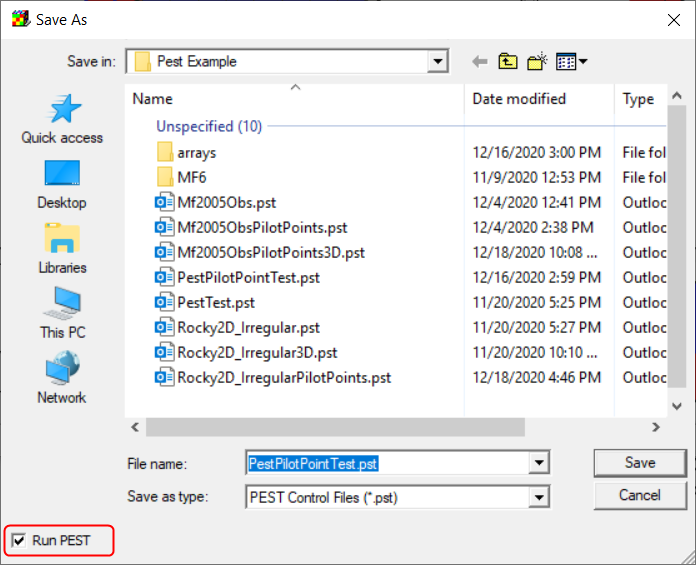
When you use pilot points with a particular parameter, the parameter will no longer be included in the PEST Control File. Instead it will be replaced by the pilot points associated with that parameter.

The pilot point spacing determines the locations and number of the candidate pilot points. ModelMuse will select some of these candidate pilot points for use with each parameter. Here’s how it does that. First it looks on a layer of a data set and determines which parameters that use pilot points are used on that layer. Then it selects the candidate pilot points that are either inside an active cell where the parameter is used or are no further than the pilot-point spacing times the square root of 2 from such a cell. This is illustrated below. In the illustration, the small cross marks indicate the candidate pilot points and the cell colors indicate which parameter is associated with a particular cell. The ones inside the dotted line will be used with the cells colored in blue. The initial value assigned to each pilot point depends on whether or not a pilot point is inside a cell associated with the corresponding parameter. If the pilot point is inside a cell associated with the parameter, the initial value of the pilot point will be the value for the cell. If the pilot point is not inside such a cell, the value will be the value of the nearest active cell on that layer associated with the parameter.



For each combination of data set, layer, and array parameter, a separate set of pilot points will be selected from among the candidate pilot points. When you run the model outside of PEST (using the RunModflow.Bat or RunSutra.bat batch files) the first thing that will be down is to calculate the “kriging factors” associated with each set of pilot points. These kriging factors will be reused when running the model from PEST (using the RunModel.Bat batch file). The kriging factors are calculated by PLPROC, a program that you can download from the PEST home page. (<https://pesthomepage.org/>). ModelMuse will look for PLPROC in three places: 1) the PEST directory specified in the Model|PEST Properties dialog box, 2) the ModelMuse directory, and 3) the directory where the model is being run.

PLPROC will also be used to perform the pilot point interpolation. This is done in the RunModflow.Bat, RunSutra.bat, and RunModel.Bat batch files. If boundary condition parameters are used, EnhancedTemplateProcessor may be used to incorporate the parameters in the boundary input files. Next MODFLOW or SUTRA will be run. Finally a program for extracting simulated values will be run. The program will be either Mf6ObsExtractor, ObsSeriesExtractor, or SutraObsExtractor depending on whether the modeling program is MODFLOW 6, MODFLOW-2005, or SUTRA respectively. These program can be run in two modes. In one mode, the simulated values themselves are extracted. In the other model an instruction file for PEST is created. The first mode is used in RunModel.Bat which PEST runs. The second mode is used in RunModflow.Bat or RunSutra.bat which ModelMuse runs. Thus to create the proper instruction file, you first have to have your model run correctly outside of PEST.

When you run your model from ModelMuse, in addition to actually running the model, it will also create two additional batch files. RunPest.bat and RunPestChek.bat. These two batch files run PEST and PESTCHEK respectively. Once you can get your model running properly, you can use RunPestChek.bat to check whether there are any errors in your PEST control file. If PESTCHEK reports any errors, You can correct them in the “Model|Manage Parameters” or “Model|PEST Properties” dialog boxes. Then select File|Export|Run PEST again. However, you should uncheck the “Run PEST” check box. After the new PEST control file has been created, run PESTCHEK again to see if all the errors have been created. Once there are no more errors, you can start running PEST.  


# Visualization

Typically, the user will want to be able to see how the parameter estimation process has affected the distribution of model properties in exactly the same way as they are seen by the model. ModelMuse has several options that can be used for this purpose for MODFLOW models. Post-calibration visualization for SUTRA models is still under development.

1. For MODFLOW models, the array data generated by parameter substitution and pilot point interpolation is written to files in the “arrays” subdirectory of the directory where the model is being run. You can import these by selecting “File|Import|Gridded Data Files”. The files will all have “.txt” as their extension. The files will be used to create new data sets which you can then visualize in ModelMuse using the normal methods.
2. You can also import the PVAL file modified by PEST into ModelMuse. The PVAL file is a model input file for MODFLOW-2005 but not for SUTRA or MODFLOW 6. Nevertheless, ModelMuse makes use of the PVAL file for both SUTRA and MODFLOW 6 by using it as an input to EnhancedTemplateProcessor which is then used to create model input files. The Manage Parameters dialog box has a button labeled “Import Pval file”. After importing the PVAL file, you can visualize the results in ModelMuse using the normal methods. However, ModelMuse does not currently import Array parameters from the PVAL file and doesn’t perform pilot point interpolation. However, it does substitute parameter values during visualization.

# Known Issues

There a number of issues that are yet to be resolved. Here are the most prominent ones of which you should be aware.

## Intermediate Files not Deleted

The RunModel.Bat file should include instructions to delete intermediate files so that if something goes wrong with one of the programs run in the batch file, PEST won’t assume that the model ran properly when it really didn’t.

## No Prior Information for Pilot Points

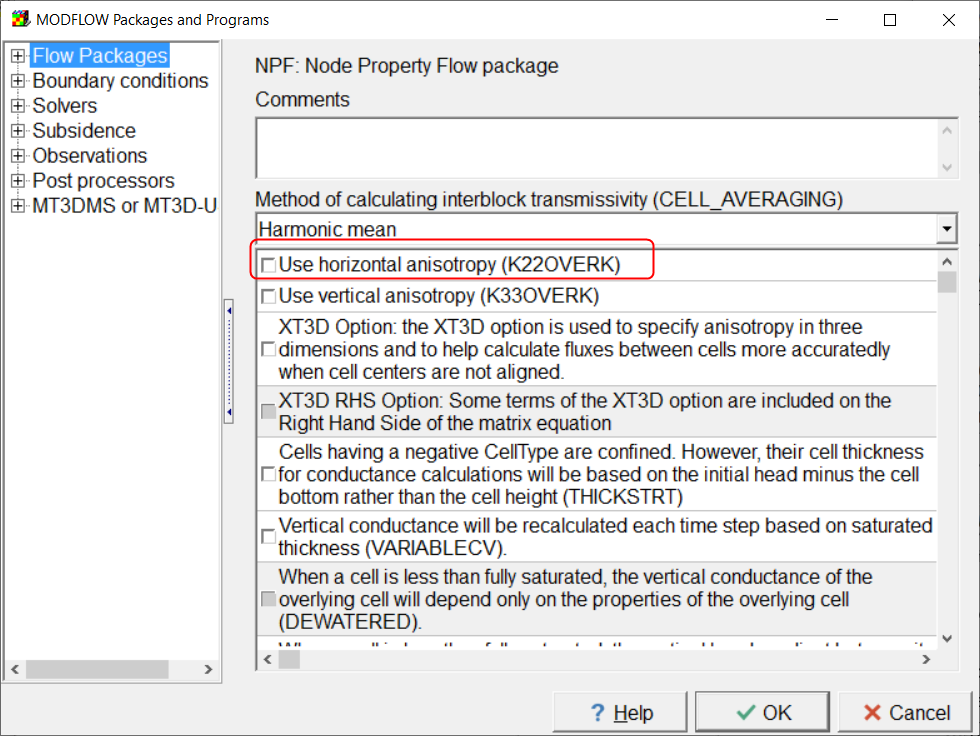
ModelMuse does not include any prior information for Pilot Points as would typically be required. You may wish to use GENREG in the PEST groundwater utilities. To add such information. There are other utilities in the PEST Utility Support Software that you might also consider.

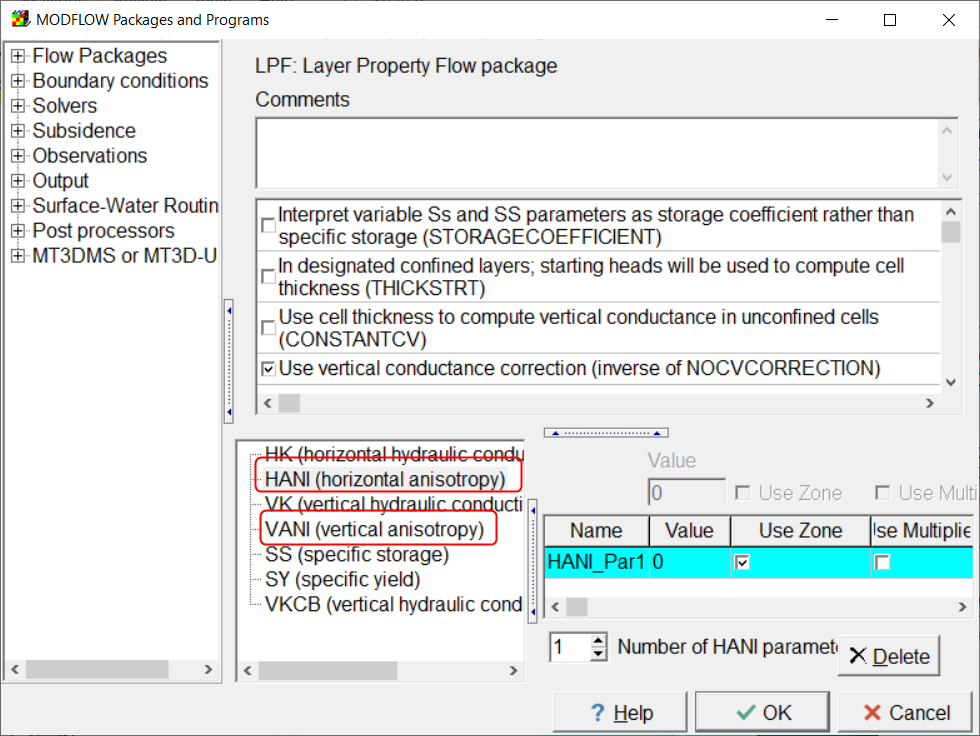
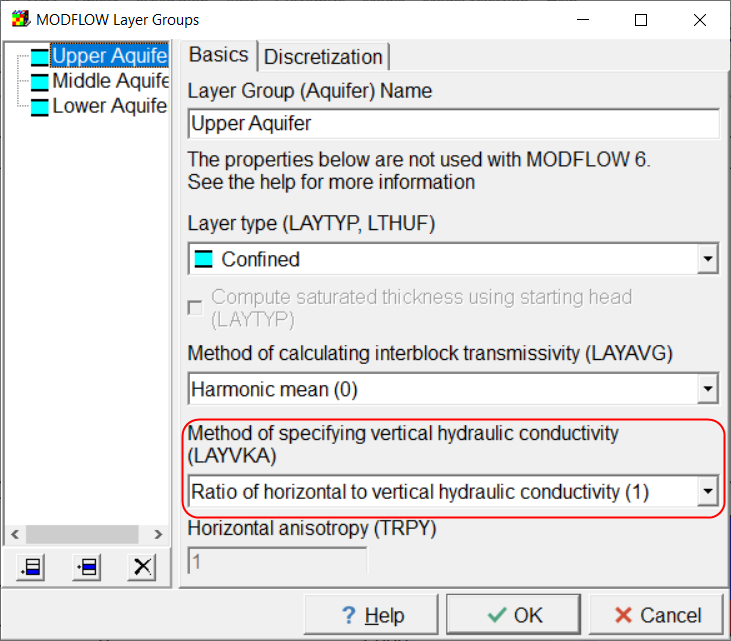
## Linked Data Sets and Anisotropy.

In ModelMuse, there are a number of data sets whose default formulas link them with other data sets. The most prominent of these are Kx, Ky, and Kz. The default formulas for Ky is Kx and the default formula for Kz is Kx/10. Now consider the case where you want Ky to have the same value as Kx but you also want to calibrate Kx. One option would be to define one or more parameters for Kx but do nothing with Ky. PEST will then modify Kx but the value of Ky was set by ModelMuse and doesn’t instruct PEST to modify it so nothing happens to Ky so you don’t achieve your goal of having Ky be the same as Kx. Another option would be to have both Kx and Ky be estimated and to use the same parameters for both and in the same locations. This doesn’t work either so long as the formula for Ky is set to Kx. Suppose the parameter value was 1E-4 m/s. Kx then gets a value of 1E-4 times whatever value was assigned to Kx by the default formula or objects. Let’s assume that the default formula for Kx is 1 so the final value for Kx is 1E-4. The value of Ky set by the default formula is 1E-4. This is multiplied by the parameter value to get a final value of 1E-8. That is very different from your goal of having Kx equal to Ky.

The best way to handle this to meet the goal is to specify horizontal anisotropy as the model input rather than specifying Ky directly. In MODFLOW 6, this is done using an option in the NPF package. There is a similar option for vertical anisotropy. In MODFLOW-2005, horizontal anisotropy is part of the model input by default and you can also have vertical anisotropy be part of the model input. You can also have horizontal anisotropy and vertical anisotropy parameters. The parameters are specified in the “Model|MODFLOW Packages and Programs” dialog box. The vertical anisotropy option is specified in the “Model|MODFLOW Layers” dialog box.

There are no similar options for SUTRA. Your best option is probably to use tied parameters for Ky and Kz. However, this isn’t supported right now if pilot points are used.



## Tied Parameters and Pilot Points.

At present, if Pilot Points is selected for a parameter in the “Model|Manage Parameters” dialog box, the parameter is replaced by a series of pilot points. Therefore you can not have such a parameter involved in tied parameters either by being tied to another parameter or by having another parameter being tied to it. However, ModelMuse doesn’t prevent you from tying such parameters in either direction. There might be some way of handling this at least in some cases but, at present, ModelMuse will just create a defective PEST control file.

## SUTRA Visualization

ModelMuse does not yet have a way to import the SUTRA input data for visualization purposes the way it can import MODFLOW data.

## SUTRA Boundary Condition Parameters

There isn’t yet a way a specifying boundary condition parameters for SUTRA.

## Pilot Points for Boundary Conditions

ModelMuse does not yet provide a way to utilize pilot points for boundary conditions.

## Import PVAL with Array Parameters

When writing the PVAL file, Array parameters are written after the non-Array parameters and are preceded by “#-- “. This allows the PVAL file to be used with MODFLOW-2005. However, ModelMuse does not yet have a way to import the Array parameter values from such a PVAL file.

## Better or Additional Methods for Designating Pilot Points.

ModelMuse doesn’t provide a way for the user to have much control over the pilot point locations.

## Bugs in SUTRA

There are some bugs in the released version of SUTRA that inhibit it from being used with PEST. Alden Provost has provided a fixed version of SUTRA but we are still awaiting the official release of a fixed version.

## User Control over Pilot Point Locations

ModelMuse does not currently let the user specify pilot point locations directly.