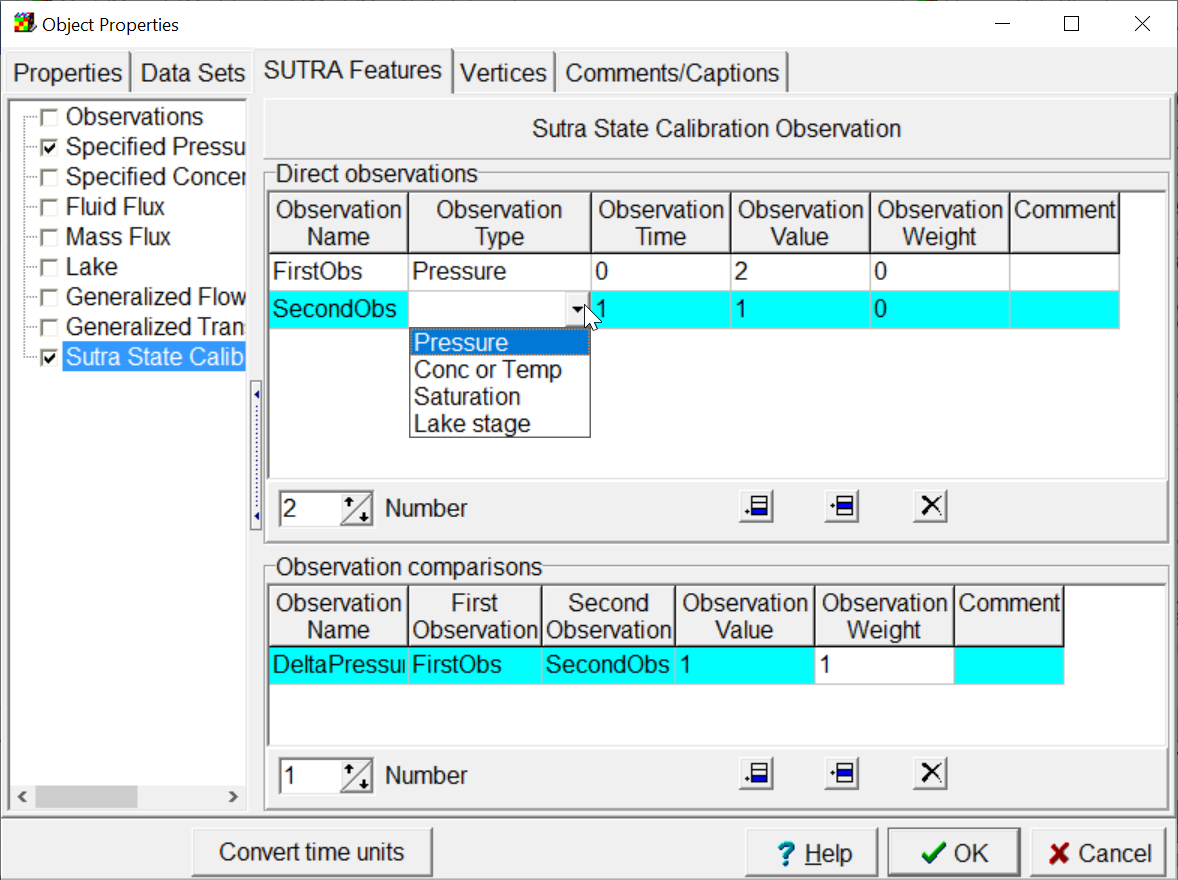
# ModelMuse with support for PEST – Beta 1

I have modified ModelMuse to begin support for PEST. This version supports creating defining observations for PEST in MODFLOW and SUTRA models. It does yet not support defining parameters for use with PEST or performing parameter estimation with PESt.

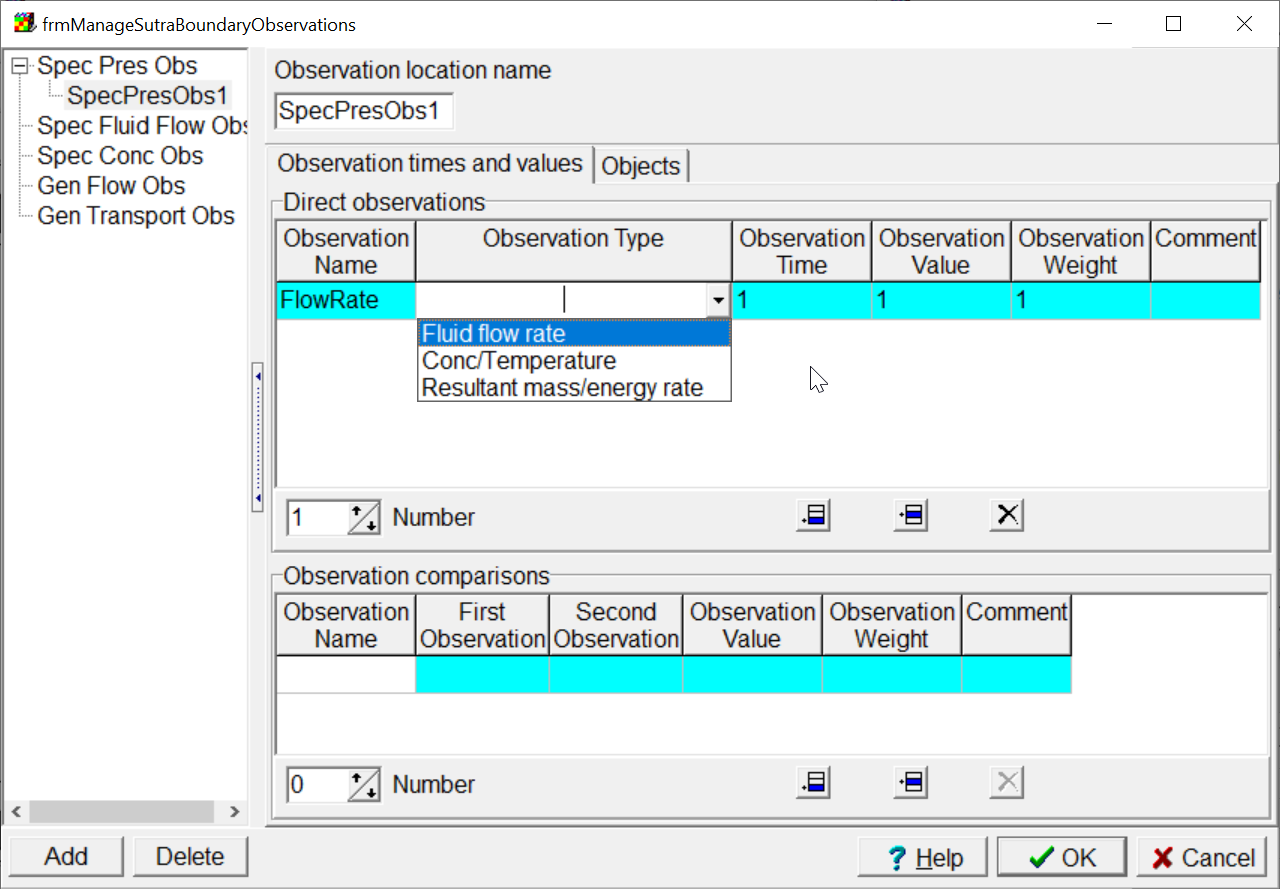
Because the values printed by the modeling programs may not correspond directly to values observed in the field, I have also developed three other programs that perform various manipulations of the printed values. These manipulations include interpolating in time and space to the observation location, adding together several values to compute a single composite value, and calculating the difference between two values to allow for observations of gradients. The programs are written in Free Pascal and can be compiled on multiple operating systems. I am supplying Windows executables. If you need an executable for another operating system, you can contact me for the source code and instructions.

## SUTRA

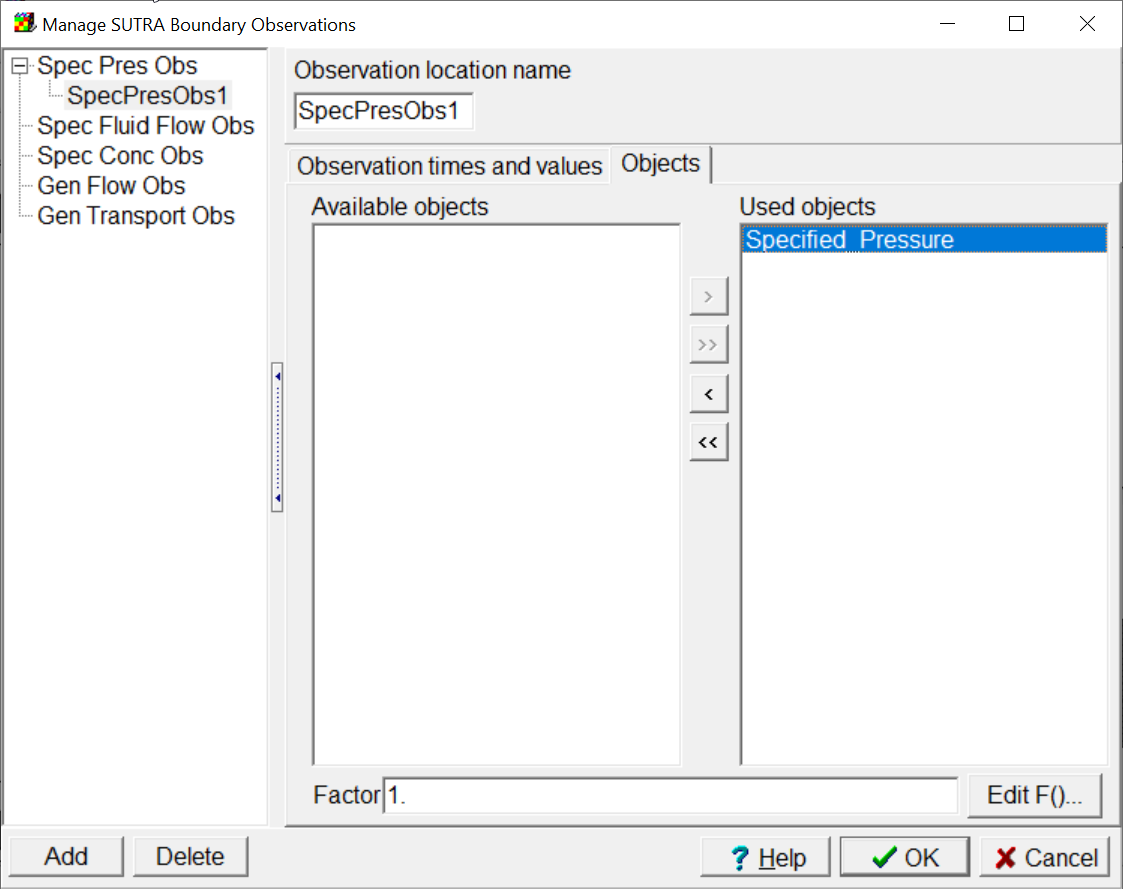
With SUTRA, you specify a state observation for calibration using point objects. State observations are specified on the upper half of the Sutra State Calibration, Observation pane in the Object Properties dialog box. State observations can be of pressure, concentration or temperature, saturation, or lake stage. If you specify two different state observations with the same object, you can specify the difference between those two observations as another observation in the lower half of the Sutra State Calibration, Observation pane. Note that ModelMuse does not enforce that the two observations that are compared are of the same type. It is up to the user to ensure that any comparisons are between commensurate observations.



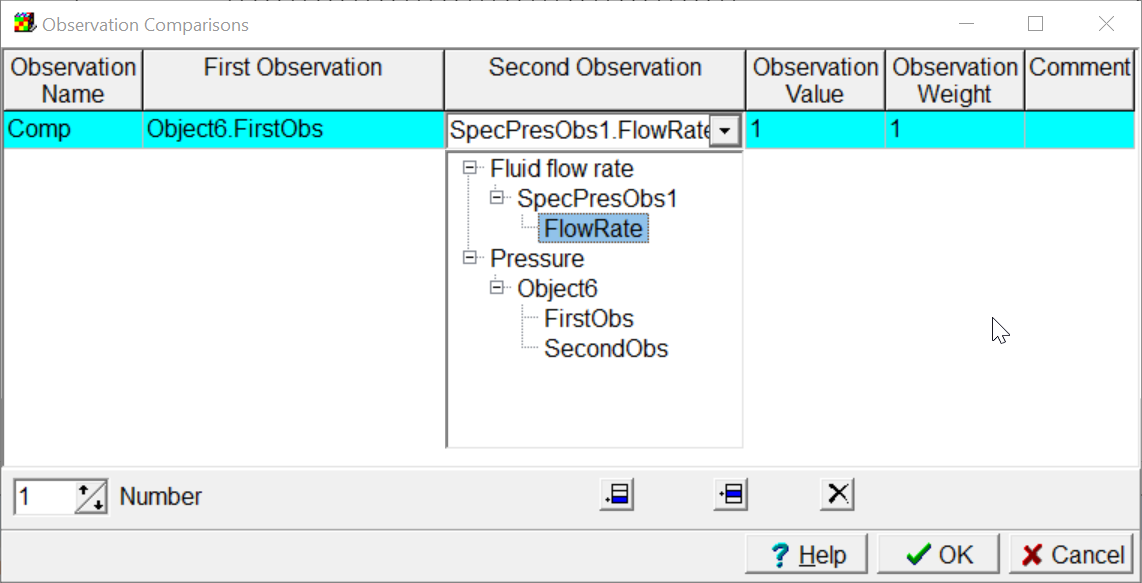
To specify a calibration observation related to boundary conditions, select “Model|Edit SUTRA Boundary Observations” to display the Manage SUTRA Boundary Observations dialog box. On the left side of the dialog box is a list of types of boundary conditions for which observations can be made. To define a new observation, select one of the types on the left and click the **Add** button. On the upper half of the **Observation times and values tab** on the right, specify one or more observations. The type of boundary condition available will be determined by the type of boundary condition. The lower half can be used to define comparisons between two different observations defined in the upper half.



Each observation location must be associated with one or more objects that define boundary conditions of the appropriate type. The objects are selected on the Objects tab. Objects in the **Used objects** list will be included in the observation. Objects in the **Available objects** list will not be included in the observation. Objects can be moved between the lists by selecting them and then using the buttons between the lists. Each object is associated with a **Factor** formula. The **Factor** formula is specified in the edit box below the lists. The formula will apply to the selected object in the **Used objects** list.. The boundary object may define a boundary that extends over several nodes. The **Factor** formula determines how much of the simulated value for the boundary at each node contributes to the observation. For example, if only half the simulated flow at a node is part of the observation, the **Factor** formula should evaluate to 0.5 at that node. The Factor formula should evaluate to a value between 0 and 1 inclusive at all locations.



Finally, you may wish to make comparisons between observations defined with different objects. For example, if you are using a head gradient as an observation, you might define head observations with two different objects and then compare the difference between them to the difference in head along the observed head gradient. Such comparisons can be defined by selecting “Model|Edit Observation Comparisons” to display the Observation Comparisons dialog box.



When ModelMuse generates the input files for SUTRA, it will also generate two additional files that are input files for the program SUTRA Observation Extractor. These files will have the extensions, .soe\_i and .soe\_ev. The former is used to create an instruction file for PEST. The later is used to extract values from the SUTRA output files in a form that can be used by PEST. When used for parameter estimation, the following operations should be performed. First run SUTRA to generate the SUTRA output files. Next run SUTRA Observation Extractor with the .soe\_i file to create the instruction file for PEST. Then when actually performing the calibration run SUTRA and SUTRA Observation Extractor in sequence using the .soe\_ev input file.

The input requirements of SUTRA Observation Extractor are described in a separate word document.

# MODFLOW-2005

The features described her apply to all versions of MODFLOW directly based on MODFLOW-2005 such as MODFLOW-NWT. They do not apply to MODFLOW 6.

MODFLOW-2005 has built in methods for defining head observations and flow through some types of boundaries. These methods are already supported in ModelMuse. However, there are additional types of data generated by MODFLOW-2005 in the output files of the GAGE, MNWI, SUB and SWT package. These data could potentially be used in parameter estimation. This section describes how observations for model calibration can be defined in ModelMuse for those packages.