PEST++ and PRMS

1. Files needed
   1. Template files (.tpl) – see section 2.3 of the [pestpp4.2.4 manual](https://github.com/jtwhite79/pestpp/tree/develop/documentation)
      1. These will look pretty identical to parameter files for PRMS and indicate which parameters to calibrate and their locations in the parameter file.
      2. Needs to have an indicator character, I suggest using %
   2. Instruction Files (.ins) – see section 2.4 of the [pestpp4.2.4 manual](https://github.com/jtwhite79/pestpp/tree/develop/documentation)
      1. These tell PEST how to read the model output files (e.g. temperature predictions). The output files must be in ASCII format
      2. Needs to have a marker delimitator, I suggest using @ for PRMS-SNTemp output
      3. In the same way that each parameter must have a unique name, so too must each observation be provided with a unique name. For PEST, observation names must be 20 characters or less in length; for programs of the PEST++ suite, they must be 200 characters or less in length.
      4. ‘ln’ tells the number of lines to advance (e.g. l1 = advance one line)
   3. Control files (.pst)
      1. .
2. PRMS-SNTemp specific issues
   1. Ss\_tau and gw\_tau need to be integers to be read into the model, however, PESTPP-GLM is a continuous algorithm and only modifies based on floating point numbers. Need to run a pre-processor to change on the floats to integers before running PRMS-SNTemp
   2. .
3. PESTPP-SEN
   1. Tests parameters sensitivity
   2. Defaults to adjusting each parameter 4 times, which creates a lot of model runs if we have a lot of parameters – e.g. 82 params x 4 = 328 runs (for just ss\_tau, gw\_tau sensitivity). We can group by parameter type which can speed things up.
      1. Tie\_by\_group = T in the write\_pestpp\_sen\_pst\_files() will tie the parameters by group type (e.g. ss\_tau, gw\_tau) rather than each segments’ parameter.
      2. I’m having some trouble with this currently (2020-08-12) when just setting tie\_by\_group = T; pestpp-sen works fine if the parameters aren’t grouped, but it seems that the input file (myparam.param) is being deleted when tie\_by\_group=T and then the model throws an error when it can write to a non-existent file
         1. This only seems to be an issue with lat\_temp\_adj parameters, not the segment parameters (e.g. ss\_tau)