13. Compatibility Issues

13.1 General

As has already been discussed, a control file for PEST_HP (and other HP suite programs) can cite a number of control variables that are not useable by standard versions of PEST. These are:

- RUN_ABANDON_FAC,
- WIN_MRUN_HOURS,
- SOFTSTOPHOURS and HARDSTOPHOURS,
- RRFSAVE,
- ZEROSENVAL,
- alternative LSQR control settings,
- UPTESTMIN and UPTESTLIM,
- ORR_NOT_FIRST,
- REG2MEASRAT,
- JCOWARNTHRESH and JCOZEROTHRESH.

A PEST_HP control file can also site secondary parameters.

The normal version of PEST is accompanied by a suite of utility programs that implement pre- and post-processing functionality of various types. Many of them read a PEST control file. At the time of writing, not all of these utility programs have been altered to recognize new variables and parameter types that are supported by members of the HP suite. Hence, if some of them are asked to read a PEST control file that contains variables and parameter types that are specific to the HP-suite, they may complain that the PEST control file contains an error, and/or that it should be checked with PESTCHEK. Some may offer a more obscure error message.

Eventually, all PEST-support programs will be altered to accommodate the presence of HP-specific variables and parameter types in a PEST control file. For the moment, however, use of utilities which do not accept these variables and parameter types requires that these variables and parameter types be removed from the PEST control file before these utilities are run. This can be accomplished automatically using the PSTCLEAN utility that is supplied with the PEST suite.

PEST-support utilities that have, at the time of writing, been modified to recognize HP-specific variables, and take appropriate action, include the following.

- Linear analysis utilities such as IDENTPAR and members of the PREDVAR* and PREDUNC* suites can happily read a PEST_HP control file. Where secondary parameters appear in a PEST control file that is read by these programs, they are ignored. This is because calculation of parameter and predictive error and uncertainty variance relies only on calibration-adjustable PEST parameters, and the sensitivities of model outputs to these parameters as recorded in a JCO file.
- The GENLINPRED program that runs many of these uncertainty and error analysis utilities can also accommodate a PEST_HP control file.

- PARREP reads a parameter value file and a PEST_HP control file. It creates a new PEST_HP control file in which the initial values of parameters are replaced by those recorded in the parameter value file. Lines in an existing PEST_HP control file which define secondary parameters and file-parameters are transferred directly to the new PEST_HP control file which PARREP writes.
- SVDAPREP will transfer the values of HP-specific control variables to the super parameter PEST_HP control file which it creates. However it will not write a super parameter PEST_HP control file if the base parameter PEST_HP control file features secondary parameters. Instead, it will cease execution with an appropriate error message. (It would make no sense to base secondary parameters on super parameters in a PEST_HP control file.)
- The RANDPAR* suite of utilities can read a PEST_HP control file that contains any of the HP-specific functionality discussed above. They can write a sequence of parameter value files which contain random values of primary (i.e. normal) PEST parameters that are featured in the control file. These utilities do not generate random values for secondary parameters; because secondary parameters are a derived parameter type, there is no need to cite them in parameter value files.
- JCO2JCO reads a PEST_HP control file and corresponding Jacobian matrix file. It can record a JCO file corresponding to another PEST_HP control file, notwithstanding the fact that either or both of these PEST_HP control files may contain HP-specific control variables and/or secondary (or file) parameters. Of course, a JCO file does not record sensitivities of model outputs to secondary (or file) parameters.
- JCOCHEK can check for compatibility between a PEST_HP control file and a JCO file, notwithstanding the presence of HP-specific control variables and/or file/secondary parameters in the PEST_HP control file.
- RRFCALCPSI can evaluate objective functions from parameter and model output values recorded in a run results file, regardless of the presence or otherwise of file/secondary parameters in the PEST_HP control file which it reads.
- ADDREG1, ADDREG2 and SUBREG1 can add/subtract regularisation to/from a PEST control file containing HP-specific control variables and/or file/secondary parameters.
- WTFACTOR can also accommodate all nuances of a PEST_HP control file.
- PESTCHEK can read and check the contents of a PEST_HP input dataset. This dataset can include HP-specific control variables, secondary parameters and file parameters. PESTCHEK issues a warning message where it encounters functionality which is specific only to programs of the HP suite.
- PEST and BEOPEST can read a PEST control file which features variables and parameters that are specific to the HP suite. They will then cease execution with an appropriate error message, drawing the user’s attention to functionality that they are incapable of supporting. If PEST is started with the “/hpstart” switch, however, it will not complain about the presence of HP-specific control variables in a PEST control file; instead it will undertake a single model run and record a hp starter file. It will not do this, however, if a PEST control file features secondary parameters or file parameters, for it has not been programmed to process these types of parameters. Instead it ceases execution with an appropriate message.

13.2 The PSTCLEAN Utility

The PSTCLEAN utility is a member of the PEST suite. It reads a PEST control file which can serve as a suitable input file for PEST_HP, or any other member of the HP suite. It writes a new PEST control file from which all HP-specific variables have been removed. The new PEST control file is readable by all PEST utility software.

As is explained in Part II of the PEST manual, PSTCLEAN also removes variables which are specific to PEST++, as well as comments, from a PEST control file. PEST++ variables are placed on lines that begin with the "++" string. Comments can be placed on any line of a PEST control file following a "#" character. PEST_HP tolerates both of these; hence there is no need to remove "++" lines or comments from a PEST control file before using PEST_HP or any other member of the HP suite.