4. Run Results File

4.1 File Contents

As the name implies, a "run results file" records the outcomes of a sequence of model runs. The first line of the file provides the number of parameters and number of model outputs (i.e. observations) which are featured in the file. These coincide with the number of parameters and observations featured in the PEST control file on which a PEST_HP run is based. The run results file then records the names of all of these. Then, for a sequence of model runs, it provides parameter values used for those runs together with model outcomes (i.e. the model-generated counterparts to observations featured in the PEST control file) calculated on the basis of these parameters. (Note that the outcomes of prior information equations are not reported.) The run results file is an ASCII file. Its format is obvious from an inspection of it.

A run results file can serve many purposes. These include uncertainty analysis (where parameter values are randomly selected, or semi-randomly selected using methodologies such as Latin hypercube), global sensitivity analysis (where parameter values follow a Sobol sequence or express trajectories defined by sequential variation of individual parameters), and construction of a proxy model that can partially replace a large, complex model in calibration of the latter.

4.2 The RRFSAVE Option

There are two means by which PEST_HP can be instructed to write a run results file. The first is through use of the RRFSAVE variable in the "control data" section of the PEST control file. This variable appears on the tenth line of this file; it must follow the values of the ICOV, ICOR, IEIG and optional IRES variables (these variables are ignored by PEST_HP as it does not write the corresponding files). It can be supplied anywhere following these variables, intermixed with other variables that can also optionally appear on this line (namely VERBOSEREC, JCOSAVEITN, REISAVEITN, PARSAVEITN and PARSAVERUN); see appendix 1 for an example. If the value of the RRFSAVE variable is supplied as "rrfsave", then PEST_HP will record a run results file. Alternatively, if RRFSAVE is supplied as "norrfsave", or omitted altogether, then PEST_HP will not record a run results file.

If recording of a run results file is instigated in this manner, then PEST_HP does not record run results for every single run which it commissions. In particular, results are not saved for model runs which are undertaken for the purpose of finite-difference derivatives calculation because

1. Too many of these runs are undertaken; and
2. Parameter values are very similar for these runs.

Instead it records outcomes of the following runs.

1. The initial model run;
2. All model runs undertaken for testing of parameter upgrades.

If the name of the PEST control file is *case.pst*, then the run results file saved by PEST_HP is named *case.rrf*.

In complying with the protocol of a run results file, PEST_HP records the reason for which each recorded model run was carried out. Where applicable, this includes the value of the Marquardt lambda and line search factor, and whether or not Broyden Jacobian improvement was being implemented. This is followed by the values of all parameters employed for that model run, and the values of model-calculated observations read from model output files recorded on that model run.

In the event of a failed model run, all model output values for that run are recorded as -1.11E35. In contrast, model output values of -1.22E35 indicate an abandoned model run.

4.3 The "/f" Command Line Option

If started with the "/f" command line switch, PEST_HP does not undertake parameter estimation at all. Instead it undertakes a series of model runs specifically for the purpose of calculating model outputs using different sets of parameters. These parameter sets must be supplied in a sequence of parameter value files recorded prior to running PEST_HP. They can be constructed using PEST-suite utility programs such as RANDPAR, RANDPAR1 and LHS2PEST; a Sobol sequence parameter set generator is available on request. The protocol for parameter value files is such that a single set of parameters is stored in each file. (See PEST documentation for specifications of this type of file.) PEST_HP expects that these files are named as a sequence with a common filename base. Suppose that the filename base is *sample*. Then the parameter value files must be named *sample1.par*, *sample2.par*, *sample3.par*, etc. The sequence does not need to begin at 1; however there must be no gaps in the sequence. The names of the parameters which appear in each of these files must be the same. They must correspond to parameters that appear in the PEST control file on which basis PEST_HP is run.

Suppose that PEST_HP is started using the following command.

```
pest_hp case /f /h :4004
```

Then, upon commencement of execution, PEST_HP reads the PEST control file *case.pst*. It then issues a series of prompts. These prompts, and possible responses to these prompts, are presented below.

```
PEST_HP will run the model repeatedly, using parameter values recorded
 in a sequence of parameter value files. It will record all model-calculated
 observations in a run results file.

 Enter filename base of parameter value files: parval
 Enter first index to use: 101
 Enter last index to use: 200

 Enter parallel run packet size: 50

 Enter name for run results file: record.rrf
```

Responses to the above prompts instruct PEST_HP to read files *parval101.par* to *parval200.par* to obtain 100 sets of parameter values. (If the names of parameters read from these files do not correspond to parameter names recorded in the PEST control file *case.pst*, PEST_HP will cease execution with an appropriate error message.) The PEST_HP run manager will then tell its agents to carry out model runs based on these parameters. In the above example, parallel model runs will be carried out in packets of 50. (It is the user's responsibility to choose the run packet size in a way that

minimizes the chances of agents being idle; the number of agents available for carrying out model runs in this example should thus be 5, 10, 25 or 50.) After each packet of model runs has been completed, PEST_HP calculates objective function components and writes these to the screen and to its run record file; it also records parameter values and model outputs for all runs comprising the packet in the run results file. It then initiates the next packet of parallel model runs.

There is no reason why the run packet size cannot be equal to the total number of parameter sets for which model outputs must be calculated. This is a matter of convenience. Sometimes a useful choice for run packet size is a number which is a small multiple of the number of available run agents. This allows you to view objective functions before the complete set of model runs is finished, thus allowing verification that the model's behaviour is in accordance with expectations. On the other hand, if model run times are sensitive to parameter values, some agents may intermittently become idle while waiting for a packet of model runs to finish because of slow model run times on other agents. Maximum efficiency is thereby gained if the run packet size is equal to the number of runs which must be undertaken.

In the event of model run failure, PEST_HP does not cease execution with an error message if started with the "/f" switch. Nor does it try to repeat a failed model run using another agent. Instead it assumes that the model did not like the parameter set with which it was provided. The objective function is recorded as -1.11E35. All model outputs associated with the offending parameter set are recorded in the run results file as -1.11E35. Model run failure is thus easily recognized.

If an appropriate value is supplied for the RUN_ABANDON_FAC variable in the "control data" section of the PEST_HP control file (see section 7.2.3), PEST_HP will abandon model runs that are requested in any run packet if these runs are unduly late. The role of the RUN_ABANDON_FAC variable in defining "late" when PEST_HP is run with the "/f" command line option is the same as its role in defining "late" where model runs are undertaken for parameter upgrade testing. Model outputs of abandoned runs are all assigned a value of -1.22E35 prior to recording these values in the run results file. The abandoned status of the model run is recorded in the header to the subsection of the run results file in which the results of the abandoned model run are recorded. Meanwhile values of -1.22E35 are recorded for the total objective function, and components of the objective function pertaining to the abandoned model run, in the PEST run record file.

A non-zero WIN_MRUN_HOURS setting (see section 7.2.4) is also respected if PEST is run with the "/f" switch. Agent-terminated model runs are treated as failed model runs. Objective function and model output values recorded in the run record and run results files reflect this status.

As is documented in the PEST manual, a parameter value file supplies values for the PRECIS and DPOINT variables used by PEST for writing numbers to model input files; it also provides the SCALE and OFFSET for each parameter. PEST_HP ignores all of these in reading the sequence of parameter value files to which it is directed when run with the "/f" command line option. Instead, values for all of these variables are obtained from the PEST control file. This frees the parameter value generator which writes the parameter value files from needing to know the values of these control variables.

When the PEST_HP manager is run with the "/f" switch, each agent writes a file named *pest####.pfl* in its working directory on each occasion that it runs the model. This file contains a single entry, namely the name of the parameter value file from which the manager obtained parameter values

that pertain to the current model run. If the parameter value filename contains a blank, its name is enclosed in quotes; otherwise it is not. This file is overwritten on each occasion that the agent runs the model.

4.4 Reading a Run Results File

The RRFCALCPSI utility supplied with the standard PEST suite can read a run results file calculated by PEST_HP. It calculates objective function components associated with each observation group. However it adds value to the contents of the run results file by allowing a different PEST control file to be used in calculation of objective function values from that which was used by PEST_HP in the production of this file.

See Part II of the PEST manual for details.