5. Model File Distribution

5.1 General

Suppose that a modeller is calibrating (for example) a steady state groundwater model, or a natural state geothermal reservoir model. The execution speeds of models such as these are much greater if initial pressures provided to the model are close to final pressures. The latter depend, of course, on parameter values. It is PEST_HP's job to adjust these such that model-generated pressures and/or flows (calculated from pressures) match observed pressures and/or flows at measurement sites.

During each iteration of the inversion process through which a model such as this is calibrated, PEST_HP attempts to improve the fit between model-calculated pressures/flows and their measured counterparts. After finite-difference derivatives have been calculated, different parameter upgrades, calculated using different Marquardt lambdas, and different line search fractions along different upgrade directions, are tested on the different computing nodes on which PEST_HP agents are running. The model that is run on one of these nodes employs the best parameters calculated for that iteration. PEST_HP does not know in advance which parameter set will be the best; this becomes apparent after the model which employs this parameter set, and models which employ other parameter sets, have been run by their respective agents, and objective functions associated with all of these parameter sets have been calculated.

The best parameter set achieved during any iteration of the inversion process is used as the starting point for the next iteration. Finite difference derivatives are calculated by varying parameters individually and incrementally from base values comprising this parameter set. Ideally, the initial pressures used by the model for all of these parameter-increment runs should be those calculated on the basis of the best parameter set from the previous iteration. The model runs devoted to finite-difference derivatives calculation should then be fast because solution pressures calculated for incremented parameters should be only incrementally different from initial pressures calculated using base parameters.

One way of achieving this desirable outcome is to employ PEST_HP's observation re-referencing functionality. Thus at the beginning of each new iteration, a special model run is undertaken in which best parameters achieved during the previous iteration are employed. Pressures calculated using this parameter set are then saved as initial pressures for an immediately ensuing model run. This next model run should take very little time; its purpose is to create reference model outputs used in finite-difference derivatives calculation. Meanwhile these same initial pressures should be distributed to every agent at PEST_HP's disposal so that model runs that are undertaken by these agents are also fast. The speed of the finite-difference derivatives calculation process is thus greatly increased.

Note how, in the above procedure, the simulator is run twice when PEST-HP undertakes the re-referencing model run. During both of these runs it employs best parameters from the previous iteration. The first simulation calculates steady-state/natural state pressures based on these parameter values; the second simulation uses these pressures for its initial conditions. In doing so it reproduces the simulation context in which other model runs will be undertaken with parameters incrementally varied from their current values. As stated above, pressures and flows at observation

locations calculated during this second simulation become reference pressures used in finite difference derivatives calculation.

While this approach has been used successfully on many occasions, some problems may arise in its implementation – particularly in the cloud environment. One problem is that of distributing the file which contains initial pressures to agent nodes. The PEST_HP run manager node may not have permission to write to agent nodes. It may not even know where they are! Another problem is that this procedure requires that the model be run using best parameters achieved up to this point in the inversion process at the start of each new iteration (just before commencement of finite-difference derivatives calculation). However this same run has already been done during the previous iteration. Furthermore, all agents other than that which is undertaking the re-referencing run must stand idle while this repeated run proceeds. This does not constitute efficient usage of cloud resources.

These problems can be overcome by using PEST_HP's file distribution functionality. At the end of each iteration of the inversion process PEST_HP can optionally fetch one or a number of files from the node on which the best model run from the previous iteration was carried out. These files are first transferred to the PEST_HP manager's node using TCP/IP – the same protocol as that employed for all other PEST_HP manager-to-agent communication. Then, optionally, the manager can issue a user-specified system command to process these files; this processing is undertaken locally on its own node. Then calculation of finite-difference derivatives commences. However before PEST_HP commissions a model run on any particular agent, it transfers one or a number of files to that agent. These files can be the same as those previously transferred to the manager from the node on which the best model run was carried out. Alternatively, new files calculated by the manager node on the basis of these transferred files can be transferred to each agent node. As before, transfer between manager and agent (i.e. PEST_HP and AGENT_HP) is completed using TCP/IP. Thus write permission on agent nodes is not required. The model is then run by the respective incidence of AGENT_HP with a parameter incrementally varied in the usual manner.

This file distribution protocol eliminates the need for the manager to know where each agent is situated. It also eliminates the need for the manager to have read/write permission on each agent's drive. At the same time, this protocol reduces the computational burden of the re-referencing run. An expensive simulation with old starting pressures is no longer required. Instead, the re-referencing simulation can use pressures calculated during the parameter upgrade procedure of the previous iteration as its initial pressures. These same initial pressures are distributed to all agents for finite-difference derivatives calculation.

5.2 PEST Control File

To implement strategic file distribution as outlined above, a new section must be introduced to the PEST control file. This "distribution files" section must be the final section of the file. Figure 5.1 shows specifications of the "distribution files" section of the PEST control file while figure 5.2 shows an example.

```
* distribution files
DISTRIB_TYPE SOURCEFILE_AGENT TARGETFILE_MANAGER [SOURCEFILE_MANAGER TARGETFILE_AGENT]
The above line is repeated up to four times.
command = DISTRIB_MODEL_COMMAND
The above line is optional and can be inserted anywhere in the "distribution files" section.
```

Figure 5.1 Specifications for the "distribution files" section of a PEST control file.

```
* distribution files
2 model.out temp.dat init.dat init.dat
1 model.out init.dat
command = process_outputs.exe
```

Figure 5.2 Example of the "distribution files" section of a PEST control file.

As for any other section of a PEST control file, the "distribution files" section must begin with a header line which provides the name of the section. This line must begin with an asterisk character followed by a space.

Up to six lines of data can follow the "* distribution files" header. The section is considered to finish when a blank line, or the end of the PEST control file, is encountered. Only five lines can follow the section header if a model command is not supplied.

The optional command associated with the "distribution files" section (i.e. DISTRIB_MODEL_COMMAND) can be provided on any of the lines which follow the "* distribution files" section header. However only one such command can be supplied. The line which provides this command must commence with the string "command =". The actual command which the PEST_HP manager will deliver to the operating system must follow this string.

Note the following features pertaining to the protocol of the line which provides the optional system command associated with file distribution.

- The word "command" can be supplied in upper or lower case, or in a combination of the two.
- The "=" character which follows the "command" string can optionally be separated from this string by a space.
- The command which follows the "=" character can optionally be separated from it by a space.
- The text comprising the system command can optionally be surrounded by double quotes.

As for other commands cited in the PEST control file, the text comprising DSITRIB_MODEL_COMMAND of figure 5.1 will be provided to the operating system exactly as written by the user. This command can be the name of a batch file, or the name of an executable program; optionally it can also include command line arguments.

Lines other than that which provides the DISTRIB_MODEL_COMMAND which appear in the "distribution files" section of the PEST control file (there can be a maximum of five of these) must all begin with an integer. This integer must be 1 or 2. This is the value of the DISTRIB_TYPE variable associated with each distribution file.

If DISTRIB_TYPE is set to 1, then the nominated file on the node on which the best model run was achieved during the previous round of parameter upgrade testing is copied to the manager's node; however this file is not distributed to nodes used by other agents. The modeller must therefore organize file distribution to these nodes him/herself. Alternatively, in some parallelization contexts, all model incidences may read this file from the manager's working folder themselves, this obviating the need for distribution of the file to agent nodes. In contrast, if DISTRIB_TYPE is set to 2, the PEST_HP manager first copies the nominated file to its own node from the best-upgrade node, and

then copies it to each agent node, possibly after processing the file using the DISTRIB_MODEL_COMMAND system command.

If DISTRIB_TYPE is set to 1, then two filenames must follow the value of this variable. These are denoted as SOURCEFILE_AGENT and TARGETFILE_MANAGER in figure 5.1. The first is the name of the file that must be copied from the best-upgrade agent's node. The second is the name that will be given to this file on the manager's node. If either of these filenames contain spaces, the filenames should be enclosed in quotes. These names can include (relative) pathnames if desired. In many cases the names given to SOURCEFILE_AGENT and TARGETFILE_MANAGER will be the same, and contain no pathnames; hence the file is simply copied from the working directory (i.e. folder) of the best-upgrade agent to the working directory of the PEST_HP manager.

If DISTRIB_TYPE is set to 2, then four filenames must follow the value of this variable. The first two are identical to those required when DISTRIB_TYPE is set to 1. However the other two are required for copying the file from the manager's node to agent nodes. SOURCEFILE_MANAGER is the name of the file that must be copied from the manager node to all agent nodes. TARGETFILE_AGENT is the name given to this file on agent nodes once it has been copied. In many circumstances SOURCEFILE_MANAGER will be the same as TARGETFILE_MANAGER. However if a DISTRIB_MODEL_COMMAND is run by PEST_HP, then this command may process the TARGETFILE_MANAGER file to produce a new SOURCEFILE_MANAGER file that is appropriate for distribution to agents. While PEST_HP allows this complication, and allows a file copied from the best-upgrade agent's node to be named differently when it appears on the manager's node, there will be many occasions where such complexity is not warranted; there will also be many occasions where no local command is employed to process the copied file. In this case all of the filenames associated with a DISTRIB_TYPE of 2 will be the same.

5.3 Implementation Details

PEST_HP initiates file distribution at the end of each iteration of the inversion process, just before it commences the next iteration (and possibly at the end of the inversion process – see below). First, regardless of the DISTRIB_TYPE setting associated with each distribution file, PEST_HP copies the file from the best-upgrade agent's node to the manager's node. This is done for all distribution files cited in the "distribution files" section of the PEST control file. If this process fails for any of the distribution files, PEST_HP ceases execution with an appropriate error message. After it has transferred all distribution files to the manager's node (recall that there can be up to five of these), PEST_HP issues the optional system command associated with its file distribution functionality (i.e. DISTRIB_MODEL_COMMAND of figure 5.1). It then proceeds to the next iteration.

For distribution files with a DISTRIB_TYPE of 2, PEST_HP does not transfer the pertinent distribution file to an agent's node until just before the next occasion on which it employs that agent to carry out a model run. If there is a failure in this transfer, PEST_HP ceases execution with an appropriate error message.

Activation of file distribution functionality requires some small modifications to normal PEST_HP functionality in order to guarantee the integrity of the inversion process. These modifications are as follows.

- If, during any iteration of the inversion process, the objective function fails to fall, and if this failure triggers PEST_HP's initiation of higher order finite-difference derivatives calculation, PEST_HP will normally commence the new iteration using the best parameters that have been achieved so far during the inversion process, even though they were not achieved during the previous iteration. This strategy cannot be adopted if file distribution functionality is in place because model-generated files associated with the best parameter set will have been lost from the node which employed this parameter set for its model run. Instead, the next iteration employs the best parameter set from the previous iteration despite the fact that this is not the best overall parameter set achieved during the inversion process to date.
- There may be occasions when the PEST_HP run manager loses access to the node of the best-upgrade agent before it can copy files from that agent's node to its own node. (This can happen if the agent disappears from the network, or if PEST_HP has access to less than three nodes.) If this happens, PEST_HP records this state of affairs to the screen and to the run record file; it then continues to the next iteration without file distribution.
- All distribution file transfers are recorded on the run management record file.

Note also the following.

- If more than one file requires distribution, then PEST_HP will allow duplicate names for the SOURCEFILE_AGENT and SOURCEFILE_MANAGER files associated with these different distribution files (though it is unclear why a user would want to do this). However it will not allow duplicate names for the TARGETFILE_MANAGER or TARGETFILE_AGENT filenames associated with different distribution files.
- Even if no source and target files are provided in the "distribution files" section of the PEST control file, a DISTRIB_MODEL_COMMAND can still be supplied. If so, the PEST_HP manager will send this command to the operating system at the close of every iteration of the inversion process.
- PEST_HP cannot be restarted using the "/s" switch if file distribution functionality is activated, as it does not know the status of distribution files on either manager or agent nodes on recommencement of execution. Use the PARREP utility to create a new PEST control file based on optimised parameters from the previous run to restart it under these circumstances.
- PEST_HP cannot be started with the "/f" switch if file distribution functionality is activated as use of this functionality would support no gains in efficiency when PEST_HP is used in this way.
- If TCP/IP communications fail when an agent is receiving a distribution file from its manager, or if AGENT_HP cannot write the distribution file to its local drive, it ceases execution with an appropriate error message. The agent is therefore lost to the inversion process.

On termination of execution, PEST_HP copies distribution files from the folder of the best-upgrade agent to that of the manager

- if it can, and
- if parameters underwent improvement on the final iteration of the inversion process.

Success of this operation depends on whether the files that were written by the best-upgrade agent on the last parameter upgrade cycle are still present in that agent's folder. If PEST_HP is able to copy these files back to the manager, if notifies the user through its screen output, its run record file and its run management record file. If not, no message is recorded. In neither case is the DISTRIB_MODEL_COMMAND issued to the operating system.

5.4 File Distribution and Broyden Jacobian Updating

Notwithstanding the extra book-keeping that it requires, PEST_HP does not prevent Broyden updating of the Jacobian matrix if file distribution is activated. This is because conjunctive use of these two items of PEST_HP functionality may be warranted when calibrating large, highly parameterized, nonlinear models such as geothermal reservoir models.

Caution must be exercised in implementing both of these options together, however. In particular, it is advisable that the directory used by the PEST_HP manager should not also be employed by a PEST_HP agent.

During each iteration of the inversion process, pertinent files that are nominated in the "distribution files" section of the PEST control file are copied from the folder used by the agent that supervised the best model run, to the folder used by the PEST_HP run manager, at the end of the first parameter upgrade testing cycle, regardless of whether a second round of upgrade testing takes place following Broyden Jacobian updating. If a second round of upgrade testing does, indeed, take place, files earmarked for distribution are copied to the manager's folder only if the best parameter set obtained during that second testing cycle is an improvement over the best obtained during the previous cycle of parameter upgrade testing (i.e. the cycle of parameter upgrade testing undertaken without a Broyden-updated Jacobian matrix). File distribution from the manager's folder to agent folders (for files with a DISTRIB_TYPE of 2) then takes place as agent nodes are commissioned to undertake model runs at the beginning of the next iteration of the inversion process.

5.5 File Distribution and Randomized Jacobian

File distribution to agent node folders does not occur if PEST_HP is using a randomized Jacobian matrix (see section 9 of this document), and the covariance matrix from the previous iteration on which this Jacobian matrix is based is being supplemented with model runs from the present iteration in accordance with the setting of the NRANDREPEAT control variable. Under these circumstances, parameter reference values are unchanged for the next iteration. Hence files copied back to the manager's folder from the best-run agent's folder during the previous iteration cannot be used to provide initial conditions for model runs undertaken during the subsequent iteration of the inversion process as the upgraded parameters from the previous iteration have been rejected (because they did not result in objective function improvement). Instead, reference parameter values employed at the beginning of the previous iteration are retained for the new iteration.

Similar considerations apply to the optional DISTRIB_MODEL_COMMAND system command. Where PEST_HP is not using a randomized Jacobian matrix, this command is issued at the beginning of every iteration of the inversion process because upgraded parameters from the previous iteration are employed as reference values for the ensuring iteration, even if they did not lower the objective function. Where a randomized Jacobian matrix is employed, however, the DSITRIB_MODEL_COMMAND system command is not issued where a reference parameter set persists to the next iteration in accordance with the dictates of NRANDREPEAT functionality.