10. Jacobian Blanking and Simultaneous Increments

10.1 General

This section continues the theme of the previous section. It describes PEST_HP functionality that supports building of a Jacobian matrix using fewer model runs than there are adjustable parameters. In the present case, this is achieved by endowing more than one parameter with an incremental variation when undertaking Jacobian-filling model runs. The previous chapter described the use of random parameter increments to achieve model run efficiency. The present chapter focuses on the use of simultaneous parameter increments used for the purpose of finite difference derivatives calculation. It also discusses the forced zeroing of selected elements of a Jacobian matrix.

Simultaneous incrementation (as distinct from random incrementation) of more than a single parameter when undertaking a model run for the purpose of finite-difference derivatives calculation requires "blockiness" of the Jacobian matrix. That is, it requires that the set of model-generated counterparts to observations (these being simply referred to as "observations" herein) that are sensitive to some parameters are different from the set of observations that are sensitive to other parameters. If observation-to-parameter sensitivities are such that they are not completely block-isolated in this way, they can be rendered such through strategic zeroing of insensitive elements of a Jacobian matrix. Regardless of whether it occurs naturally, or through strategic zeroing of insensitive elements, use of simultaneous parameter increments requires concomitant use of a "blanking matrix" to ensure that variation of a particular model output in response to variation of multiple parameters is not misinterpreted as sensitivity of that output to more than a single one of the varied parameters. A blanking matrix can be supplied by the user, or can be calculated internally by PEST_HP.

Strategic blanking of Jacobian matrix elements can also raise the integrity of a Jacobian matrix computed using random parameter increments. As is described in the previous section of this manual, PEST_HP supports the use of random parameter increments through its "randomized Jacobian" functionality. Through the use of random parameter increments, the number of model runs required per iteration of an inversion process can be reduced to slightly greater than that of the dimensionality of the solution space of the inverse problem on which PEST_HP usage is based. However the dramatic decrease in model run requirements accrued through use of a randomized Jacobian matrix is accompanied by certain costs, particularly if parameter increments are calculated internally by PEST_HP under the assumption of stochastic independence. These costs are as follows.

- While progress in lowering an objective function may be high during early stages of an inversion process that is based on randomized Jacobian matrices, progress at later stages of this process may be compromised by the introduction of spurious correlations between some model outputs and some parameters, especially where true sensitivities of some model outputs to some parameters approach zero. The greater the number of random parameter increment vectors that are used to compute the Jacobian matrix, the smaller will be this effect. However use of a large number of random parameter increment vectors incurs a numerical cost. If the number of random parameter increments approaches the number of adjustable parameters, the benefits of using random parameter increments to fill a Jacobian matrix vanish.

- Parameter upgrades are calculated as linear combinations of parameter increment vectors. Where the number of random parameter increment vectors is equal to, or slightly greater than, the dimensionality of the inverse problem solution space, it is highly probable that they collectively span a space onto which any solution space vector has a non-zero projection; hence the objective function can be lowered. However it is most unlikely that an objective-function-reducing parameter upgrade vector, calculated as a linear combination of these random parameter increment vectors, does not have a substantial null space component. The upgrade vector is therefore likely to exhibit calibration-induced bias.

Strategic Jacobian matrix blanking may reduce both of these costs. In doing so, it performs a similar role to that of "localization" in improving the performance of an iterative ensemble smoother.

10.2 The JCOBLANK Utility

10.2.1 Simultaneous Parameter Increments

A utility program named JCOBLANK has been added to the standard PEST suite. As is described in Part II of the PEST manual, this program performs two tasks. They are

1. Construction of a blanking matrix;
2. Design of a model run schedule for finite-difference filling of a Jacobian matrix using simultaneous parameter increments.

The blanking matrix file and simultaneous increment file that are written by JCOBLANK can be read by PEST_HP. Hence PEST_HP can implement a simultaneous parameter increment strategy devised by JCOBLANK.

JCOBLANK can build a blanking matrix in two ways. One option is to employ a "zeroing file". Through this type of file, a user directly identifies Jacobian matrix elements that must be blanked. Alternatively, JCOBLANK can use an existing Jacobian matrix file (i.e. a JCO file) as a basis for blanking; under these circumstances it constructs a blanking matrix that zeroes elements of the Jacobian matrix file whose weighted sensitivities are low compared to other elements of this matrix. In either case, the greater the number of weighted Jacobian elements that are blanked, the greater is the chance that remaining elements of the Jacobian matrix have enough of a "blocky" structure to support design of a model-run-efficient simultaneous parameter increment strategy.

It is important to remember that gains in model run efficiency attained through use of simultaneous parameter increments may compromise the integrity of a Jacobian matrix. This may slow the progress of an inversion process, and may introduce bias to some estimated parameters. This effect is likely to be most significant for highly nonlinear models for which model output sensitivities to parameters are functions of the values of the parameters themselves.

While JCOBLANK is documented in part II of the PEST manual, some aspects of its functionality that are pertinent to PEST_HP's implementation of simultaneous parameter increments are now briefly discussed.

10.2.2 Observation Weights

Ideally, rows of a Jacobian matrix for which observation weights are zero should be blanked when this matrix is used as a basis for the design of a simultaneous parameter increment strategy. This is

because rows pertaining to observations with zero weight can be omitted from a Jacobian matrix with no adverse consequences for an inversion process. Blanking of zero-weighted rows prevents non-zero sensitivities in these rows from compromising potential blockiness of the Jacobian matrix, and hence development of an efficient simultaneous parameter increment strategy. JCOBLANK automatically blanks zero-weighted rows of a Jacobian matrix if it builds a blanking matrix from this matrix. However it does not automatically blank zero-weighted Jacobian matrix rows if it follows instructions provided in a zeroing file. It is thus the user's responsibility to ensure that zero-weighted rows are blanked through the instructions which he/she provides in this file if this is his/her desire. There may, indeed, be occasions where blanking of zero-weighted rows is not desirable. This occurs if an inversion process "carries" one or a number of model predictions for which linear uncertainty analysis will later be undertaken.

10.2.3 Simultaneous Increment Strategy

Design of a simultaneous increment strategy can be a numerically intensive procedure. Optimization of this strategy (i.e. reducing the number of model runs required for filling a blocky Jacobian matrix) can be even more numerically intensive. JCOBLANK uses a rather simple algorithm for development of a simultaneous increment strategy. It begins the process by inspecting the first column of the Jacobian matrix. It then determines whether another adjustable parameter can be incremented conjunctively with the parameter associated with this first column by visiting subsequent columns of the Jacobian matrix in order to ascertain whether all non-zero sensitivities in one of these columns are non-overlapping with those of the first column. If it finds such a column, the parameter that is associated with this column is co-assigned to the first Jacobian-building model run. JCOBLANK then continues its inspection of Jacobian matrix columns; however now it looks for a column of the Jacobian matrix whose non-blanked elements have no coincidence with non-blanked elements of columns associated with the two parameters that have been assigned to the first model run. If it finds such a column, then another parameter is designated as being simultaneously incremented during the first Jacobian-building model run. This column-search procedure continues until all Jacobian columns have been visited; the parameter composition of the first model run is thereby completely determined. JCOBLANK then repeats this process to determine the parameter composition of the second model run. And so on.

This strategy is reasonably efficient. However its outcomes may depend on the order in which Jacobian columns are visited. In particular, the parameter composition of different Jacobian filling model runs may be different if Jacobian columns are visited in a random order rather than in a sequential order. In contrast to JCOBLANK, PEST_HP employs a random visitation order of Jacobian matrix columns when it is asked to devise a simultaneous parameter increment strategy (see below). Hence the simultaneous increment strategy which it devises may be different from that devised by JCOBLANK, even if they are based on the same Jacobian matrix. Furthermore, neither of these strategies can be guaranteed to be that which minimizes the total number of simultaneous model runs required for finite-difference filling of a blocky Jacobian matrix.

10.2.4 Blanking Re-Visited

Before it devises a simultaneous parameter increment strategy, JCOBLANK actually builds a blanking matrix based on sensitivities that it finds in the Jacobian matrix. In accordance with user instructions, Jacobian matrix elements with low sensitivities are blanked in order to enhance blockiness of this matrix. The simultaneous increment strategy that it devises is based on this blanking matrix.

Once it has devised a simultaneous parameter increment strategy based on the blanking matrix, JCOBLANK can be asked to review the blanking matrix. The simultaneous parameter increment strategy that it devised using the above algorithm may not require that all blanked elements of a blanking matrix remain blanked. Hence, if asked to do so, JCOBLANK unblanks elements of the blanking matrix whose blanking is nonessential for support of the simultaneous parameter incrementation strategy which it has just devised. PEST_HP does this automatically when it devises its own simultaneous parameter increment strategy; see below.

10.2.5 Multiple Command Lines

If the NUMCOM variable in the "control data" section of a PEST control file is set to a number greater than 1, then PEST and PEST_HP employ different commands to run a model when calculating sensitivities with respect to different parameters. The index of the command that is used to run the model for a specific parameter is supplied through the DERCOM variable that is associated with each parameter in the "parameter data" section of the PEST control file.

If a model employs multiple command lines, the simultaneous parameter increment strategy devised by JCOBLANK and PEST_HP respect this. That is, only parameters that employ the same model command are incremented simultaneously.

10.2.6 Prior Information

Both JCOBLANK and PEST_HP ignore prior information when they build a blanking matrix, and then devise a simultaneous parameter increment strategy on the basis of that matrix. Because sensitivities of prior information equations are directly supplied through the prior information equations themselves, they do not require calculation through finite parameter differencing. Efficiency of their calculation does not therefore benefit from a simultaneous parameter increment strategy.

10.2.7 Covariance Matrices

If a covariance matrix is assigned to an observation group which does not exclusively pertain to prior information, then JCOBLANK does not perform Jacobian blanking and evaluation of a simultaneous parameter increment strategy if it is asked to base these on an existing Jacobian matrix. This is because JCOBLANK actually works with a weighted Jacobian matrix under these circumstances. The relationship between Jacobian rows and weights is invalidated by the use of a covariance matrix that applies to multiple observations.

10.3 PEST_HP and Simultaneous Parameter Increments

PEST_HP can read a blanking matrix from a so-called "blanking file". It can also read a file containing a simultaneous parameter increment strategy from a so-called "simultaneous increment file". These files must complement each other; they may both have been written by JCOBLANK. PEST_HP uses the contents of the simultaneous increment file to allocate parameter increments to Jacobian-filling model runs. Then, after the Jacobian matrix has been filled using these model runs, it blanks this matrix using information obtained from the blanking file; as stated above, this eliminates confusion in attribution of model output sensitivities to parameters.

Alternatively, PEST_HP can itself be asked to create a blanking matrix itself based on the latest Jacobian matrix that it has calculated, and to formulate a simultaneous parameter increment strategy based on this blanking matrix. These are used to assist it in building the next Jacobian

matrix. PEST_HP constructs the blanking matrix by identifying elements of low absolute value in the Jacobian matrix that it has just filled; in doing this, it employs an identical algorithm to that used by JCOBLANK when it is asked to build a blanking file based on an existing Jacobian matrix. After it has formulated a simultaneous parameter increment strategy based on this blanking matrix, PEST_HP saves a blanking file and a simultaneous increment file for use in the next iteration of the inversion process (and possibly in iterations following that).

Note the following aspects of PEST_HP's behaviour in building a blanking matrix, and in designing a simultaneous parameter increment strategy.

1. Rows of the Jacobian matrix corresponding to observations that are assigned weights of zero are blanked. If a PEST control file is "carrying" zero-weighted observations as predictions for later use in linear predictive uncertainty analysis, the blanking of these rows of the Jacobian matrix will invalidate this analysis.
2. PEST_HP's algorithm for construction of a simultaneous parameter increment scheme is similar to that of JCOBLANK. However in determining which parameters to increment on a particular model run, it visits columns of the Jacobian matrix in random order. Furthermore, the visitation order varies between the different occasions on which it devises a simultaneous parameter increment strategy.
3. The random number seed which governs PEST_HP's random column visitation order is set internally. It can be re-set by providing a seed to the RANDOMSEED control variable in the "randomized jacobian" section of a PEST control file. This does not require activation of randomized Jacobian functionality; RANDOMJAC can be set to zero in this section of the file, thereby de-activating this functionality.
4. Because PEST_HP visits columns of the Jacobian matrix in random order and JCOBLANK visits these columns in sequential order, a simultaneous parameter increment strategy devised by PEST_HP may differ from that devised by JCOBLANK if the latter program is provided with the same Jacobian matrix.
5. Once it has devised a simultaneous parameter increment strategy, PEST_HP alters the blanking matrix so that it conforms with this strategy (see above). Hence no blanking takes place beyond that which is required to complement the simultaneous parameter increment strategy.
6. If simultaneous parameter increments are employed in building a Jacobian matrix, PEST_HP disables sensitivity re-use and parameter bounds sticking if either of these have been enabled through settings supplied in a PEST control file.
7. As is usual practice, if the NOPTMAX control variable appearing in the "control data" section of a PEST control file is set to -1 or -2, PEST_HP ceases execution after filling the Jacobian matrix. If the SIMINCCALC control variable (see below) is set to 1, then PEST_HP also constructs a blanking matrix and devises a simultaneous parameter increment strategy before ceasing execution; as is described below, these are stored in appropriately-named blanking and simultaneous increment files. If desired, these files can be used on subsequent PEST_HP runs in which the SIMINC control variable is set to 1; however they must first be renamed (see below).

Note that it is not always necessary for Jacobian matrix blanking to be undertaken as part of the design of a simultaneous parameter increment strategy. As stated above, Jacobian matrix blanking may sometimes be undertaken to simply improve the effectiveness of randomized Jacobian matrix calculation; set JCOBLANK to -1, and both SIMINC and SIMINCCALC to 0 to achieve this outcome (see

below). With these settings PEST_HP does not record a blanking matrix in an external file; nor does it design a simultaneous increment strategy. It simply blanks any Jacobian matrix that it calculates, and reports to the screen the number of Jacobian elements that it blanks.

10.4 PEST_HP Control Variables

10.4.1 Section in PEST Control File

A new section has been introduced to the PEST_HP control file. This section is labelled "* simultaneous parameter increments". It must be placed between the "control data" and "parameter groups" sections of a PEST control file. Other sections (such as "singular value decomposition", "randomized jacobian" and "sensitivity reuse") can also appear between the "control data" and "parameter groups" sections of a PEST control file. There is no prescribed ordering for these sections if more than one of them is present.

Figure 10.1 shows variables which are featured in the "simultaneous parameter increments" section of a PEST_HP control file. Figure 10.2 shows an example.

Note that, at the time of writing, PESTCHEK tolerates the presence of this section, but does not check its contents. The PSTCLEAN utility removes this section from a PEST control file in order to render that file inoffensive to the normal versions of PEST and BEOPEST.

```
* simultaneous parameter increments
BLANKJCO    SIMINC
BLANKFILE
SIFILE
SIMINCCALC
FRACOBS FRACPAR FULLJCOITN SIMINCACCEL MINSIMINC MATCHAGENTS
```

Figure 10.1. Specifications of the "simultaneous parameter increments" section of a PEST control file.

```
* simultaneous parameter increments
1 1                             #BLANKJCO   SIMINC
blanking_file = blank.jcz       #BLANKFILE
si_file = siminc.dat            #SIFILE
1                               #SIMINCALC
.5  .5  5  1 95 1  #FRACOBS FRACPAR FULLJCOITN SIMINCACCEL MINSIMINC MATCHAGENTS
```

Figure 10.2. Example of a "simultaneous parameter increments" section of a PEST control file.

10.4.2 Variables

The role of each of the variables appearing in the "simultaneous parameter increments" section of a PEST control file is now discussed.

BLANKJCO

BLANKJCO is an integer variable which must be set to -1, 0 or 1.

If BLANKJCO is set to 1, PEST_HP reads a blanking file; it uses the contents of this file to blank every Jacobian matrix that it calculates. Note that Jacobian matrix blanking does not necessarily require that simultaneous parameter increments be employed for filling of the Jacobian matrix. As is stated above, Jacobian matrix blanking may raise the integrity of a Jacobian matrix calculated using random parameter increments.

Unless PEST_HP calculates its own blanking matrix (which occurs when the SIMINCCALC control variable is set to 1), the blanking file read by PEST_HP when BLANKJCO is set to 1 will normally have

been written by the JCOBLANK utility. The name of this file is assigned to the BLANKFILE control variable. PEST_HP reads, and then re-reads, this file during every iteration of the inversion process just after it has undertaken the model runs required for filling of the Jacobian matrix.

As is usual practice, if PEST_HP is started with the "/i" command-line switch, it reads an existing Jacobian matrix for use in the first iteration of the inversion process. This matrix is not blanked, even if BLANKJCO is set to 1.

If BLANKJCO is set to -1, PEST_HP performs Jacobian blanking during every iteration of the inversion process. However under these circumstances SIMINC must be set to 0, as should the SIMINCCALC control variable (see below). Under these circumstances PEST_HP simply blanks every Jacobian matrix that it calculates using the FRACOBS and FRACPAR settings that are listed after the SIMINCCALC control variable. Under these circumstances, it neither reads nor records a blanking file; nor does it devise a simultaneous parameter increment strategy. Blanking based on FRACOBS and FRACPAR is applied to the Jacobian matrix which it has just calculated, on every occasion that it fills a Jacobian matrix. This BLANKJCO setting can sometimes be useful where randomized parameter increments are employed for filling of the Jacobian matrix. Note also that when BLANKJCO is set to -1, Jacobian blanking takes place during the first iteration of the inversion process, even if PEST_HP execution is initiated using the "/i" switch.

SIMINC

SIMINC, an integer variable, must be set to 0 or 1. If it is set to 1, PEST_HP obtains a simultaneous parameter increment strategy from a user-nominated simultaneous increment file; it employs this strategy during every iteration of the inversion process. The name of this file is assigned to the SIFILE control variable. Normally this file will have been written by the JCOBLANK utility. PEST_HP reads, and then re-reads, this file during every iteration of the inversion process just before it commissions the model runs required for filling of that iteration's Jacobian matrix.

If SIMINC (or SIMINCCALC) is set to 1, PEST_HP requires that the FORCEN variable for all parameter groups be set to the same value, this being either *always_2*, *always_3*, *switch*, or *always_5*. If different parameter groups have different FORCEN settings, the number of model runs used for calculation of derivatives of model outputs with respect to different parameters differs between parameters. This makes design of a simultaneous parameter increment strategy difficult. Actually, a FORCEN setting of *always_2* may be suitable for most occasions. While this setting precludes the attainment of increased derivatives precision through use of higher order finite differences, this may not matter too much as use of simultaneous parameter increments together with concomitant Jacobian element blanking also reduces the precision of finite-difference derivatives.

Note the following.

- If SIMINC is set to 1, PEST_HP will object if BLANKJCO is not also set to 1.
- If SIMINC is set to 1, PEST_HP will object if randomized Jacobian matrix calculation is activated.

BLANKFILE

BLANKFILE is the name of the Jacobian blanking file which PEST_HP reads if BLANKJCO is set to 1. As is illustrated in figure 10.2, its name must follow the string "blanking_file =" on the third line of the "simultaneous parameter increments" section of a PEST control file. Spaces can optionally surround

the "=" sign. If BLANKJCO is set to 0, it is not necessary for the name of a file to follow the "blanking_file =" string, for its name is disregarded under these circumstances. If the name of the blanking file contains a space, its name should be enclosed by quotes.

The blanking file whose name is assigned to the BLANKFILE control variable must be a binary file; its name must have an extension of "*.jcz*". Normally this file will have been written by JCOBLANK. However it must not be named *case.jcz* where *case* is the filename base of the PEST control file, for this name is reserved for the blanking file that PEST_HP writes and reads if SIMINCCALC is set to 1.

SIFILE

SIFILE is the name of the simultaneous increment file which PEST_HP reads if SIMINC is set to 1. As is illustrated in figure 10.2, its name must follow the string "si_file =" on the fourth line of the "simultaneous parameter increments" section of a PEST control file. Spaces can optionally surround the "=" sign. If SIMINC is set to 0, it is not necessary for the name of a file to follow the "si_file =" string, for its name is disregarded under these circumstances. If the name of the simultaneous increments file contains a space, its name should be surrounded by quotes.

The simultaneous increment file whose name is supplied to the SIFILE control variable will normally have been written by JCOBLANK. This file must not be named *case.sif*, were *case* is the filename base of the PEST control file. This name is reserved for the simultaneous increment file which is written and read by PEST_HP if SIMINCCALC is set to 1.

SIMINCCALC

Set this integer variable to 1 to inform PEST_HP that it must periodically calculate a blanking matrix and devise a simultaneous parameter increment strategy itself. If PEST_HP devises a simultaneous parameter increment strategy during any particular iteration of the inversion process, it does so immediately after it has filled the Jacobian matrix pertaining to that iteration. First it determines a blanking matrix based on weighted sensitivities recorded in the Jacobian matrix that it has just calculated. This matrix is recorded in a file named *case.jcz*, where *case* is the filename base of the PEST control file. Then, based on this blanking matrix, PEST_HP devises a simultaneous parameter increment strategy for use during one or more subsequent iterations. It stores this strategy in a simultaneous increment file named *case.sif*, where *case* is the filename base of the PEST control file.

Note that just after PEST_HP has devised a simultaneous parameter increment strategy, it re-examines the blanking matrix on which basis this strategy was devised. If it is possible to unblank some elements of this matrix while maintaining the integrity of the simultaneous parameter increment strategy, then it does so. As was discussed above, this can increase the precision of (simultaneous) finite-difference derivatives calculation.

If SIMINCCALC is set to 1, PEST_HP does not use a PEST_HP-calculated simultaneous parameter increment strategy to fill the Jacobian matrix that it calculates during the first iteration of the inversion process. If SIMINC is set to 0, this matrix is filled in the normal way; that is, all parameters are incremented individually. Alternatively, if SIMINC is set to 1, the Jacobian matrix used in the first iteration is filled using a simultaneous parameter increment strategy supplied in a file whose name is ascribed to the SIFILE control variable; the Jacobian matrix is then blanked using the blanking matrix supplied in a file whose name is ascribed to the BLANKFILE control variable. The Jacobian matrix computed during all iterations identified by the value of the FULLJCOITN control variable (see below)

is computed in this same way (if SIMINC is set to 1). During all of these iterations, however, PEST_HP evaluates its own blanking matrix and simultaneous parameter increment strategy based on the thus-computed Jacobian matrix if SIMINCCALC is set to 1. These are stored in files *case.jcz* and *case.sif* for use in subsequent iterations; existing versions of these files are thus over-written.

If SIMINCCALC is set to 1, then PEST_HP will object if its randomized Jacobian functionality is activated.

FRACOBS and FRACPAR

PEST_HP calculates a blanking matrix from a Jacobian matrix in the same way that JCOBLANK does. (Unlike JCOBLANK, however, PEST_HP does not optionally read a zeroing file to obtain blanking information.) First PEST_HP blanks all rows of the Jacobian matrix for which observation weights are zero. Then, for each row, it ascertains the weighted sensitivity of highest absolute value. Any elements within this row whose absolute weighted sensitivities are less than FRACOBS times this maximum value are then blanked. The same operation is then performed for each column of the weighted Jacobian matrix using FRACPAR as the governing variable. FRACOBS and FRACPAR should both lie between 1.0 and 0.0 (inclusive); they are real variables.

Note that prior information sensitivities are not blanked. Note also that the blanking matrix and simultaneous parameter increment strategy devised by PEST_HP during any iteration of the inversion process are not used until the ensuing iteration.

FULLJCOITN

As stated above, if SIMINCCALC is set to 1, PEST_HP calculates a Jacobian matrix in the "normal" way during the first iteration of the inversion process. Hence it does not employ simultaneous parameter increments for filling of this matrix unless SIMINC is set to 1. On the basis of this Jacobian matrix it calculates a blanking matrix and devises a simultaneous parameter increment strategy for use in the next iteration (or more) of the inversion process.

PEST_HP also calculates a Jacobian matrix in the normal way at intervals of FULLJCOITN iterations. Thus, for example, if FULLJCOITN is set to 2, then PEST_HP calculates a Jacobian matrix in the normal manner during iterations 1, 3, 5, 7, etc of the inversion process; similarly, if FULLJCOITN is set to 3, PEST_HP calculates a Jacobian matrix in the normal way during iterations 1, 4, 7, 10, etc of the inversion process. During all other iterations of the inversion process, PEST_HP employs simultaneous increment and blanking strategies which it has calculated itself; it reads the blanking matrix and details of the simultaneous parameter increment strategy from files *case.jcz* and *case.sif* respectively, where *case* is the filename base of the PEST control file.

PEST_HP will cease execution with an error message if FULLJCOITN is set to 1 or less.

SIMINCACCEL

"SIMINCACCEL" stands for "simultaneous increment acceleration". If SIMINCACCEL is set to 1 then SIMINCCALC should also be set to 1.

If SIMINCACCEL is set to 0 and SIMINCCALC is set to 1, PEST_HP does not compute a blanking matrix or simultaneous parameter increment strategy during those iterations in which it actually uses a blanking matrix and simultaneous parameter increment strategy that it has previously calculated itself. This happens during all iterations except those for which the setting of the FULLJCOITN control

variable dictates that Jacobian matrix calculation takes place in the "normal" way. However if SIMINCACCEL is set to 1, then PEST_HP updates the blanking matrix and evaluates a new simultaneous parameter increment strategy even during those iterations in which it is using a blanking matrix and simultaneous parameter increment strategy that it has previously calculated for itself. This can result in a significant reduction in the number of model runs required for filling of the Jacobian matrix during the ensuing iteration (as long as this iteration has not been decreed by FULLJCOITN to be a "normal" Jacobian iteration). Unfortunately, this reduction in model run requirements may be accompanied by some loss of integrity of the Jacobian matrix that is calculated during that iteration. Hence if SIMINCACCEL is set to 1, it is good practice to set the MINSIMINC control variable (see below) to a sensible value so that the number of model runs employed for filling the Jacobian matrix does not become unworkably low.

If SIMINCACCEL is set to 1 then FULLJCOITN must be set to 3 or greater. If FULLJCOITN is not set to 3 or greater under these circumstances, then there can be no iterations to which a non-zero value of SIMINCACCEL actually pertains. This is because PEST_HP is either calculating a Jacobian matrix in the "normal" way during a particular iteration, or is employing a blanking strategy that it has devised itself on the basis of this "normally-calculated" Jacobian matrix.

MINSIMINC
If this integer variable is set to a number greater than 0, PEST_HP will never devise a parameter increment strategy which lowers the number of simultaneous parameter increments below this number.

If the value supplied for MINSIMINC is equal to or greater than the number of adjustable parameters, PEST_HP will cease execution with an error message.

MATCHAGENTS
If SIMINCCALC is set to 1, so that PEST_HP devises its own simultaneous parameter increment strategy during some iterations of the inversion process, PEST_HP will try to ensure that the number of simultaneous parameter increments that it adopts is an integral multiple of the number of agents available for carrying out model runs. First PEST_HP evaluates a simultaneous parameter increment strategy using the FRACOBS and FRACPAR control variables described above. Then, if necessary, it raises the number of simultaneous parameter increments calculated in this way until the number of increments is equal to a multiple of the number of available agents. This ensures that no agents are idle as the Jacobian matrix is being filled. Raising the number of simultaneous increments to achieve processor-to-increment parity also raises the integrity of the thus-calculated Jacobian matrix, as less blanking of this matrix is required to forestall the assignment of changes in model outputs to the wrong parameters.

10.5 Using SVD-Assist

PEST_HP allows the use of simultaneous parameter increments when undertaking SVD-assisted parameter estimation. However because the use of super parameters already constitutes a theoretically optimal run reduction strategy, little is likely to be gained by using both methods together.

The SVDAPREP utility which prepares a super-parameter PEST control file tolerates the presence of a "simultaneous parameter increments" section in the base parameter PEST control file from which

the super parameter PEST control file is derived. However it does not transfer this section to the super parameter PEST control file. It is the user's responsibility to add this section to the super parameter PEST control file him/herself.