## PEST++ ver. 3.0.0

## The PEST++ Command Line

PEST++ supports a number of different command line options which specify how to start a run using one of the supported run managers. Large problems (defined as having many parameters and/or observations) often require parallel computing. PEST++ relies on run mangers to complete the forward model runs and the current version provides the following four options: 1) **Y**et **A**nother Run **M**anage**R** (**YAMR**), 2) GENIE and 3) serial run manager 4) External run manager. YAMR and GENIE are sophisticated and capable of performing parallel runs on a single machine or over a TCP/IP-enabled network. YAMR duplicates the functionality of BEOPEST and is fully integrated in PEST++. Like BEOPEST, PEST++ can run as either a master or a compute node under the YAMR run manager. While PEST++ provides an interface to the GENIE run manager, this interface relies on the external GMAN and GSLAVE programs to manage and perform the actual model runs. The serial rule manager provides a simple alternative that duplicates the functionality currently in regular PEST. The external run manager was developed to terminate PEST++ execution when it comes time to make the forward model runs. It is designed to allow the user can take control over the process of making the forward model runs. The command lines required to start PEST++ are summarized in the following table where /j and /r are optional commands; /j to invokes jacobian reuse for the first iteration and /r invokes restart.

|  |  |
| --- | --- |
| **Run Manger / Mode** | **Command** |
| Serial Run Manager / Master | pest++ control\_file.pst [/j] [/r] |
| YAMR / Master | pest++ control\_file.pst /H :port [/j] [/r] |
| YAMR / Compute Node | pest++ /H hostname:port |
| Genie / Master | pest++ control\_file.pst /G hostname:port [/j] [/r] |
| External Run Manager | pest++ control\_file.pst /E [/j] [/r] |

When PEST++ is run in the serial or master mode it now supports the /j option to reuse an existing jacobian file rather than computing the jacobian for the first iteration. The functionality of this feature is identical to that in PEST. In fact PEST++ is compatible with jacobians computed by PEST as long as the PEST++ AUTONORM option is not used in the control file.

## The PEST++ Control File

For ease of reference, variables within the PEST control file are listed below, and the variables used by PEST++ are highlighted. PEST++ relies on the structure of the input file to deduce the algorithmic parameters and read only those algorithmic parameters that are absolutely necessary. For example, there is no need to read the NOBS variable because each line in the “observation data” section of the control file specifies an observation; however, it is necessary to read the NPAR variable to know where specification of parameters ends and information on tied parameters begins. This list is followed by short explanation of each variable used by PEST++.

pcf

\* control data

RSTFLE **PESTMODE**

**NPAR** NOBS NPARGP NPRIOR NOBSGP [MAXCOMPDIM]

**NTPLFLE** NINSFLE PRECIS DPOINT [NUMCOM JACFILE MESSFILE]

RLAMBDA1 RLAMFAC PHIRATSUF PHIREDLAM NUMLAM [JACUPDATE] [LAMFORGIVE]

**RELPARMAX FACPAR MAXFACORIG** [IBOUNDSTICK UPVECBEND] [ABSPARMAX]

**PHIREDSWH [NOPTSWITCH] [SPLITSWH]** [DOAUI] [DOSENREUSE]

**NOPTMAX PHIREDSTP NPHISTP NPHINORED RELPARSTP NRELPAR** [PHISTOPTHRESH] [LASTRUN] [PHIABANDON]

ICOV ICOR IEIG [IRES] [JCOSAVE] [VERBOSEREC] [JCOSAVEITN] [REISAVEITN] [PARSAVEITN]

\* automatic user intervention

MAXAUI AUISTARTOPT NOAUIPHIRAT AUIRESTITN

AUISENSRAT AUIHOLDMAXCHG AUINUMFREE

AUIPHIRATSUF AUIPHIRATACCEPT NAUINOACCEPT

\* singular value decomposition

SVDMODE

**MAXSING EIGTHRESH**

**EIGWRITE**

\* lsqr

LSQRMODE

LSQR\_ATOL LSQR\_BTOL LSQR\_CONLIM LSQR\_ITNLIM

LSQRWRITE

\* svd assist

BASEPESTFILE

BASEJACFILE

SVDA\_MULBPA SVDA\_SCALADJ SVDA\_EXTSUPER SVDA\_SUPDERCALC SVDA\_PAR\_EXCL

\* sensitivity reuse

SENRELTHRESH SENMAXREUSE

SENALLCALCINT SENPREDWEIGHT SENPIEXCLUDE

\* parameter groups

**PARGPNME INCTYP DERINC DERINCLB FORCEN DERINCMUL DERMTHD** [**SPLITTHRESH SPLITRELDIFF** SPLITACTION]

(*one such line for each of NPARGP parameter groups*)

\* parameter data

**PARNME PARTRANS PARCHGLIM PARVAL1 PARLBND PARUBND PARGP SCALE OFFSET** DERCOM

(*one such line for each of NPAR parameters*)

**PARNME PARTIED**

(*one such line for each tied parameter*)

\* observation groups

**OBGNME** [GTARG] [COVFLE]

(*one such line for each of NOBSGP observation group*)

\* observation data

**OBSNME OBSVAL WEIGHT OBGNME**

(*one such line for each of NOBS observations*)

\* derivatives command line

DERCOMLINE

EXTDERFLE

\* model command line

**COMLINE**

(*one such line for each of NUMCOM command lines*)

\* model input/output

**TEMPFLE INFLE**

(*one such line for each of NTPLFLE template files*)

**INSFLE OUTFLE**

(*one such line for each of NINSLFE instruction files*)

\* prior information

**PILBL PIFAC \* PARNME + PIFAC \* log(PARNME) ... = PIVAL WEIGHT OBGNME**

(*one such line for each of NPRIOR articles of prior information*)

\* predictive analysis

NPREDMAXMIN [PREDNOISE]

PD0 PD1 PD2

ABSPREDLAM RELPREDLAM INITSCHFAC MULSCHFAC NSEARCH

ABSPREDSWH RELPREDSWH

NPREDNORED ABSPREDSTP RELPREDSTP NPREDSTP

\* regularisation

**PHIMLIM PHIMACCEPT [FRACPHIM]** [MEMSAVE]

**WFINIT WFMIN WFMAX** [LINREG][REGCONTINUE]

**WFFAC WFTOL IREGADJ** [NOPTREGADJ REGWEIGHTRAT [REGSINGTHRESH]]

\* pareto

PARETO\_OBSGROUP

PARETO\_WTFAC\_START PARETO\_WTFAC\_FIN NUM\_WTFAC\_INC

NUM\_ITER\_START NUM\_ITER\_GEN NUM\_ITER\_FIN

ALT\_TERM

OBS\_TERM ABOVE\_OR\_BELOW OBS\_THRESH NUM\_ITER\_THRESH (*only if ALT\_TERM is non-zero*)

NOBS\_REPORT

OBS\_REPORT\_1 OBS\_REPORT\_2 OBS\_REPORT\_3..(*NOBS\_REPORT items*)

++# This line is a comment as are all lines that begin with “++#”

++# PEST++ input is parsed using key words that can be specified in any order

++ MAX\_N\_SUPER(max\_super) SUPER\_EIGTHRES(eig\_thres)

++ N\_ITER\_BASE(base\_iter) N\_ITER\_SUPER(super\_iter)

++ SVD\_PACK(PROPACK)

++ AUTO\_NORM(4)

++MAX\_SUPER\_FRZ\_ITER(max\_frz\_iter)

++MAX\_REG\_ITER(max\_reg\_iter)

++MAT\_INV(inv\_type)

++SUPER\_RELPARMAX(sup\_relpar\_max)

++MAX\_RUN\_FAIL(max\_fail)

++LAMBDAS(lambda list)

++ITERATION\_SUMMARY(TRUE)

++DER\_FORGIVE(TRUE)

Variables in “control data” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| RSTFLE | Text | “restart” or “norestart” | Instructs PEST whether to write restart data. |
| PESTMODE | Text | “estimation”, “prediction”, “regularisation”, “pareto” | PEST’s mode of operation. |
| NPAR | Integer | greater than 0 | Number of parameters. |
| NUMCOM | Integer | optional; greater than zero | Number of command lines used to run model. |
| RELPARMAX | Real | greater than 0 | Parameter relative change limit. |
| FACPARMAX | Real | greater than 1 | Parameter factor change limit. |
| FACORIG | Real | between 0 and 1 | Minimum fraction of original parameter value in evaluating relative change. |
| PHIREDSWH | Real | between 0and 1 | Sets objective function change for introduction of central derivatives. |
| NOPTMAX | Integer | −2, −1, 0, or any number greater than 0 | Number of optimization iterations. |
| PHIREDSTP | Real | greater than 0 | Relative objective function reduction triggering termination. |
| NPHISTP | Integer | greater than 0 | Number of successive iterations over which PHIREDSTP applies. |
| NPHINORED | Integer | greater than 0 | Number of iterations since last drop in objective function to trigger termination. |
| RELPARSTP | Real | greater than 0 | Maximum relative parameter change triggering termination. |
| NRELPAR | Integer | greater than 0 | Number of successive iterations over which RELPARSTP applies. |

Variables in optional “singular value decomposition” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| MAXSING | Integer | greater than 0 | Number of singular values at which truncation occurs. |
| EIGTHRESH | Real | 0or greater, but less than 1 | Eigenvalue ratio threshold for truncation. |
| EIGWRITE | Integer | 0or 1 | Determines content of SVD output file. |

Variables required for each parameter group in “parameter groups” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| PARGPNME | Text | 12 characters or less | Parameter group name. |
| INCTYP | Text | “relative”, “absolute”, “rel\_to\_max” | Method by which parameter increments are calculated. |
| DERINC | Real | greater than 0 | Absolute or relative parameter increment. |
| DERINCLB | Real | 0or greater | Absolute lower bound of relative parameter increment. |
| FORCEN | Text | “switch”, “always\_2”, “always\_3”, “switch\_5”, “always\_5” | Determines whether central derivatives calculation is undertaken and whether three points or four points are employed in central derivatives calculation. |
| DERINCMUL | Real | greater than 0 | Derivative increment multiplier when undertaking central derivatives calculation. |
| DERMTHD | Text | “parabolic”, “outside\_pts”, “best\_fit”, “minvar”, “maxprec” | Method of central derivatives calculation. |

Variables required for each parameter in “parameter data” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| PARNME | Text | 12 characters or less | Parameter name. |
| PARTRANS | Text | “log”, “none”, “fixed”, “tied” | Parameter transformation. |
| PARCHGLIM | Text | “relative”, “factor”, or absolute(n) | Type of parameter change limit. |
| PARVAL1 | Real | any real number | Initial parameter value. |
| PARLBND | Real | less than or equal to PARVAL1 | Parameter lower bound. |
| PARUBND | Real | greater than or equal to PARVAL1 | Parameter upper bound. |
| PARGP | Text | 12 characters or less | Parameter group name. |
| SCALE | Real | any number other than 0 | Multiplication factor for parameter. |
| OFFSET | Real | any number | Number to add to parameter. |
| DERCOM | Integer | 0or greater | Model command line used in computing parameter increments. |
| PARTIED | Text | 12 characters or less | The name of the parameter to which another parameter is tied. |

Variables required for each observation group in “observation groups” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| OBGNME | Text | 12 characters or less | Observation group name. |

Variables required for each observation in “observation data” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| OBSNME | Text | 20 characters or less | Observation name. |
| OBSVAL | Real | any number | Measured value of observation. |
| WEIGHT | Real | 0or greater | Observation weight. |
| OBGNME | Text | 12 characters or less | Observation group to which observation assigned. |

Variables in “model command line” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| COMLINE | Text | system command | Command to run model. |

Variables in “model input/output” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| TEMPFLE | Text | a filename | Template file. |
| INFLE | Text | a filename | Model input file. |
| INSFLE | Text | a filename | Instruction file. |
| OUTFLE | Text | a filename | Model output file. |

Variables in “prior information” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| PILBL | Text | 20 characters or less | Name of prior information equation. |
| PIFAC | Text | real number other than 0 | Parameter value factor. |
| PARNME | Text | 12 characters or less | Parameter name. |
| PIVAL | Real | any number | “Observed value” of prior information. |
| WEIGHT | Real | 0or greater | Prior information weight. |
| OBGNME | Text | 12 characters or less | Observation group name. |

Variables in optional “regularization” section of PEST control file.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| PHIMLIM | Real | greater than 0 | Target measurement objective function. |
| PHIMACCEPT | Real | greater than PHIMLIM | Acceptable measurement objective function. |
| FRACPHIM | Real | optional; 0or greater, but less than 1 | Set target measurement objective function at this fraction of current measurement objective function. |
| MEMSAVE | Text | “memsave” or “nomemsave” | Activate conservation of memory at cost of execution speed and quantity of model output. |
| WFINIT | Real | greater than 0 | Initial regularization weight factor. |
| WFMIN | Real | greater than 0 | Minimum regularization weight factor. |
| WFMAX | Real | greater than WFMAX | Maximum regularization weight factor. |
| LINREG | Text | “linreg” or “nonlinreg” | Informs PEST that all regularization constraints are linear. |
| REGCONTINUE | Text | “continue” or “nocontinue” | Instructs PEST to continue minimizing regularization objective function even if measurement objective function is less than PHIMLIM. |
| WFFAC | Real | Greater than 1 | Regularization weight factor adjustment factor. |
| WFTOL | Real | Greater than 0 | Convergence criterion for regularization weight factor. |
| IREGADJ | integer | 0, 1, 2, 3, 4 or 5 | Instructs PEST to perform interregularizationgroup weight factor adjustment, or to compute new relative weights for regularization observations and prior information equations. |
| NOPTREGADJ | integer | 1 or greater | The optimization iteration interval for recalculation of regularization weights if IREGADJ is 4 or 5. |
| REGWEIGHTRAT | Real | absolute value of 1 or greater | The ratio of highest to lowest regularization weight; spread is logarithmic with null space projection if set negative. |
| REGSINGTHRESH | Real | less than 1 and greater than 0 | Singular value of **x**t**qx** (as factor of highest singular value) at which use of higher regularization weights commences if IREGADJ is set to 5. |

## PEST++ Additions to the PEST Control File

Information in the PEST control specific to PEST++ is specified on lines starting with “++”. Although the previous example places all the PEST++ input in a single section at the end of the PEST control file, this is not a requirement. This information does not need to be contiguous and can reside anywhere in the PEST control file. Lines starting with “++#” are considered comments and are ignored.

Unlike the rest of the PEST control file, PEST++ uses keywords rather than location to specify variables. Lines are parsed using the space, tab, and parenthesis characters as separators. The example uses parentheses to more clearly delineate the values assigned to the variable, but these could just as well be replaced by white spaces. The following table includes a listing and explanation of the permissible PEST++ keywords.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| N\_ITER\_BASE | Integer | 1 or greater | Number of base parameter iterations performed for each super parameter iteration. |
| N\_ITER\_SUPER | Integer | 0 or greater | Number of super parameter iterations performed for each base parameter iteration. |
| SUPER\_EIGTHRES | Real | any positive number (typically should be greater than 1.0e−7) | PEST++ will not include any super parameters whose ratio with the largest super parameter is less than this ratio.This value can as small as zero if the user wants to specify the number of super parameters solely with SUPER\_NMAX. BecausePEST++uses SVD on the super parameter problem, a low value for this SUPER\_EIGTHRES will not adversely impact the stability of the solution. |
| MAX\_N\_SUPER | Integer | integer between 1 and the minimum of maximum number of parameters and the maximum number of observations | Maximum number of super parameters to use in the super parameter iterations. |
| MAX\_REG\_ITER | Integer | Integer greater than 1. Default is 20. | Provides a limit on the maximum the number of iterations used to compute dynamic regularization weights when PEST++ is run in regularization mode. |
| MAX\_SUPER\_FRZ\_ITER | Integer | 1 or greater, Default value is 5. | Maximum number of times a super parameter iteration will try to freeze any parameters that go out of bounds andl try to recompute a jacobian. If the jacobian cannot be computed in MAX\_SUPER\_FRZ\_ITER iterations PEST++ will switch to a base parameter iteration. |
| AUTO\_NORM(4) | Integer | 1 or greater | Automatically normalizes the parameters by assuming there are 4 standard deviations between the upper and lower parameter bounds |
| SVD\_PACK(PROPACK) | NA |  | Flag to use PROPACK to compute SVD factorizations |
| MAT\_INV | String | Q1/2J  JTQJ | Flag to specify whether to solve Q1/2J or JtQJ. This option is forced to “Q1/2J” when PROPACK is used |
| SUPER\_RELPARMAX | Real | Greater than 0 | Parameter relative change limit for super-parameters |
| MAX\_RUN\_FAIL | Integer |  | Maximum times the run manager will try to rerun a failed run |
| LAMBDAS | Comma separated list of real numbers |  | Specify the standard values of lambda to be used each iteration rather than use the default of (.01,1,10,100,1000) |
| ITERATION\_SUMMARY | “TRUE” or “FALSE” |  | Setting this to “TRUE” will save a summary of each iteration to a series of comma separated files for easy plotting. |
| DER\_FORGIVE | “TRUE” or “FALSE” |  | Setting this to “FALSE” will turn off derivative forgive and cause PEST++ to terminate if a run fails while computing the jacobian. |
| OVERDUE\_RESCHED\_FAC | Real |  | YAMR option to specify when an overdue run will be rescheduled. Runs are rescheduled when they are overdue by OVERDUE\_RESCHED\_FAC \* average run time |
| OVERDUE\_GIVEUP\_FAC | Real |  | YAMR option to specify when an overdue run will be aborted. Runs are aborted when they are overdue by OVERDUE\_GIVEUP\_FAC \* average run time |

## References

Doherty, J., 2011a, PEST surface water utilities: Brisbane, Australia,Watermark Numerical Computing.

Doherty, J., 2011b, Groundwater data utilities: Brisbane, Australia, Watermark Numerical Computing.

Muffels, C.T., Schreüder, W.A., Doherty, J., Karanovic, M., Tonkin, M.J., Hunt, R.J., and Welter, D.E., 2011, GENIE—A model independent TCP/IP run managerinMODFLOW and More 2011—Integrated Hydrologic Modeling, Proceedings of the 10th International Conference of the International Ground Water Modeling Center: Golden, Colo., Colorado School of Mines.

Muffels, C.T., Schreüder, W.A., Doherty, J., Karanovic, M., Tonkin, M.J., Hunt, R.J., and Welter, D.E., 2012,Approaches in Highly Parameterized Inversion: GENIE, A General Model Independent TCP/IP Run Manager. U.S. Geological Survey Techniques and Methods Report**.**