# PEST++ Version 3.4 Input Instructions

The PEST++ Version 3 Visual Studio solution, as well as source code and executable as documented in this report, are available for download at http://wi.water.usgs.gov/models/pestplusplus/. More recent releases of PEST++, including any enhancements made since the publication of this report, will be available at <http://www.pestpp.org/>. The most current development version of the source code is maintained an online open-source version-control repository at https://github.com/dwelter/pestpp/.

In order to facilitate use by experienced PEST users, PEST++ adopts many of the conventions, variable names, and output formats of the original PEST (Doherty, 2010). The intent is to make PEST++ input and output compatible with the large number of existing PEST utilities (for example, Doherty, 2011a,b).

## The PEST++ Command Line

PEST++ supports various command line options that control run manager invocation as well as restart options. PEST++ Version 3 supports three run mangers to complete the forward model runs: (1) Yet Another Run ManageR (YAMR), (2) GENIE, and (3) a serial run manager. YAMR and GENIE are sophisticated parallel run managers capable of performing parallel runs on a single machine or over a TCP/IP-enabled network. YAMR is integrated into PEST++ and is invoked similarly to BeoPEST (Schreüder, 2009). Although PEST++ provides an interface to the GENIE run manager, this interface relies on the external GMAN and GSLAVE programs (Muffles and others, 2012) to manage and perform the actual model runs. The serial run manager provides a simple alternative that mimics the functionality currently in PEST. In addition to run manager specification, the command line also controls the restart functionality of PEST++.

The various options related to run manger and restart control are summarized in table 1–1, where /j and /r are optional commands; /j invokes Jacobian reuse for the first iteration, and /r invokes restart.

**Table 1–1.** Summary of PEST++ command line options.

|  |  |
| --- | --- |
| Run Manger / Mode | Command |
| Serial Run Manager / Master | pest++.exe <casename>.pst [/j] [/r] |
| YAMR / Master | pest++.exe <casename>.pst /H :<port> [/j] [/r] |
| YAMR / Worker Node | pest++.exe <casename>.pst /H <hostname>:<port>  or  pest++.exe <casename>.ymr /H <hostname>:<port> |
| GENIE / Master | pest++.exe <casename>.pst /G <GENIE Master hostname>:<port> [/j] [/r] |
| GENIE/Master | genie.exe /port <port> |
| GENIE / Worker Node | genie.exe /ip <GENIE Master IP address> /port <port> |

When PEST++ is run with the serial run manager or as the master node with a parallel run manager, it now supports the /j option to reuse an existing binary Jacobian file rather than computing the Jacobian for the first iteration. Note that PEST++ can be restarted by using a Jacobian computed by PEST as long as the PEST++ “autonorm” option is not invoked in the control file.

When PEST++ is used to involk the a YAMR worker node, the user has the option to specify an abbreviated “.ymr” control file which contains only the relevant information pertaining to the worker node rather than a full PEST++/PEST 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 shaded. PEST++ relies on the structure of the PEST control file (Doherty, 2010) to read the necessary algorithmic parameters and reads only those algorithmic parameters that are needed. For example, there is no need to read the NOBS variable because each line in the “observation data” section of the PEST 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 FACPARMAX FACORIG [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

PARGPN MEINCTYP 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 PARGPSCALE 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(20) SUPER\_EIGTHRES(1.0E-8)

++ N\_ITER\_BASE(1) N\_ITER\_SUPER(3)

++ SVD\_PACK(PROPACK) AUTO\_NORM(4)

++ LAMBDAS(0.1,1,10,100,1000)

++ MAX\_SUPER\_FRZ\_ITER(5)

++ MAX\_REG\_ITER(20)

++ MAT\_INV(inv\_type)

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

++ MAX\_RUN\_FAIL(3)

++ ITERATION\_SUMMARY(TRUE)

++ DER\_FORGIVE(TRUE)

++ UNCERTAINTY(TRUE)

++ FORECASTS(pred\_1,pred\_2,pred\_3)

++ PARAMETER\_COVARIANCE(prior\_parameter.cov)

++ OVERDUE\_RESCHED\_FAC(2.0)

++ OVERDUE\_GIVEUP\_FAC(10.0)

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 0 and 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 | 0 or greater, but less than 1 | Eigenvalue ratio threshold for truncation. |
| EIGWRITE | Integer | 0 or 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 | 0 or 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. PEST++ V3 only supports “parabolic” |

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 | 0 or 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 | 0 or 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; 0 or 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 inter-regularization group 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 file specific to PEST++ is marked by lines starting with “++”. Although the examples provided in this report place all 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 file. Lines starting with “++#” are considered comments and are ignored by PEST and PEST++.

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. Although one can use parentheses to more clearly delineate the values assigned to the variable (for example, ++N\_ITER\_BASE(1) specifies N\_ITER\_BASE=1), these could just as well be replaced by white spaces (for example, ++N\_ITER\_BASE 1 also specifies N\_ITER\_BASE=1). Table 1–2 includes a listing and explanation of the permissible PEST++ keywords.

**Table 1–2.**  PEST++ optional arguments.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| N\_ITER\_BASE | Integer | 1 or greater | Number of base parameter iterations performed for each superparameter iteration. |
| N\_ITER\_SUPER | Integer | 0 or greater | Number of superparameter 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 superparameters whose ratio with the largest superparameter is less than this ratio. This value can as small as zero if the user wants to specify the number of superparameters solely with MAX\_N\_SUPER. Because PEST++uses SVD on the superparameter 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 either of maximum number of parameters or the maximum number of observations | Maximum number of superparameters to use in the superparameter 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. Setting this value too large can result in appreciable slowdown, especially in early iterations. |
| MAX\_SUPER\_FRZ\_ITER | Integer | 1 or greater; default value is 5 | Maximum number of times a superparameter iteration will try to freeze any parameters that go out of bounds and 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; default is no scaling | Automatically normalizes the sensitivities by assuming there are X standard deviations between the upper and lower parameter bounds, where X is the value passed with the AUTO\_NORM variable (4 is shown). |
| SVD\_PACK(PROPACK) | String | “JACOBI” or “PROPACK; default is “JACOBI” | Flag to use PROPACK to compute SVD factorizations. “JACOBI” is the SVD provided by the EIGEN library; “PROPACK” is the iterative SVD factorization suitable for large problems. |
| MAT\_INV | String | “Q1/2J” or “JTQJ”; default is “JTQJ” | Flag to specify the formulation of the normal equation. This option is forced to “Q1/2J” when PROPACK is used. |
| SUPER\_RELPARMAX | Real | greater than 0; default is 0.1 | Parameter relative change limit for superparameters. |
| MAX\_RUN\_FAIL | Integer | greater than 0; default is 3 | Maximum times the run manager will try to rerun a failed run. |
| LAMBDAS | Comma-separated list of reals | greater than 0; default is (0.01,1,10,100,1000) | Specify the standard values of lambda to be used each iteration. |
| ITERATION\_SUMMARY | Boolean | “TRUE” or “FALSE”; default is “TRUE” | Setting this to “TRUE” will save a summary of each iteration to a series of comma-separated files for easy plotting. |
| DER\_FORGIVE | Boolean | “TRUE” or “FALSE”; default is “TRUE” | 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 |
| UNCERTAINTY | Boolean | “TRUE” or “FALSE”; default is “TRUE” | A flag to disable uncertainty analyses. |
| FORECASTS | Comma separated list of text | Observation names in the control file; default is none | The names of observations to treat as forecasts in the uncertainty analyses. |
| PARAMETER\_COVARIANCE | Text | Filename; default is none | The name of a PEST-compatible ASCII matrix or uncertainty file to use as the prior parameter covariance matrix. |
| OVERDUE\_RESCHED\_FAC | Real | greater than 1.0; default is 1.15 | YAMR specific command. If a model run takes longer than (OVERDUE\_RESCHED\_FAC \* the average runtime) it will rescheduled on another node if one is available |
| OVERDUE\_GIVEUP\_FAC | Real | greater than 1.0; default is 100.0 | YAMR specific command. If a model run has been running longer than (OVERDUE\_GIVEUP\_FAC \* the average runtime) it will canceled |
| REG\_FRAC | Real | greater than 0.0 | Regularization fraction. This option overrides PEST style regularization and allows the regularization component of the objective function to be specified as a fraction of the total objective function. REG\_FRAC is the fraction of the total objective function due to regulatization. |
| GLOBAL\_OPT | Text | “DE” | Turns on global optimization and specifies the global optimization technique. Differential Evolution (“DE”) is the only global optimization technique currently supported. |
| DE\_POP\_SIZE | Integer | default is 40 | Differential evolution population size. Generally ten times the number of estimated parameters is recommended, however Rainer Storn suggest that empirical experience shows anything above 40 rarely improves convergence. Perhaps the best particle advice is to use the larger of 40 and the number of available processors. |
| DE\_MAX\_GEN | Integer | greater than 0; default is 100 | The maximum number of differential evolution iterations or generations. |
| DE\_DITHER\_F | Boolean | “TRUE” or “FALSE”; default is “TRUE” | Switch to toggle differential evolution dithering. When this is set to “TRUE” dithering is enabled and the weighting factor, “DE\_F”, for each difference vector is drawn randomly from the interval [0.5 1.0]. Dithering is on by default as it can dramatically improve convergence. |
| DE\_F | Real | default is 0.8 | Differential evolution weighting factor. Used when “DE\_DITHER\_F” == “FALSE”. |
| DE\_CR | Real | default is 0.9 | Differential evolution crossover constant. |

## The YAMR Worker Control File (.ymr)

The variable within the optional YAMR worker control file are listed below. This list is followed by short explanation of each variable used by PEST++.

\* model command line

COMLINE

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

\* model input

TEMPFLE INFLE

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

\* model output

INSFLE OUTFLE

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

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

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

Variables in “model input” section of YAMR worker control file.

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

Variables in “model output” section of YAMR worker control file.

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

## References

Doherty, John, 2010, Addendum to the PEST manual: Brisbane, Australia, Watermark Numerical Computing.

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

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

Muffels, C.T., Schreüder, W.A., Doherty, J.E., 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, book 7, chap. C6, 26 p., <http://pubs.usgs.gov/tm/tm7c6/>.

Schreüder, W.A., 2009, Running BeoPEST, *in* Tonkin, M.J., ed. Proceedings, PEST Conference 2009, Potomac, Md., November 1–3, 2009: Bethesda, Md., S.S. Papadopulos and Associates, p. 228–240.

# GSA++ Implementation and Use

GSA++ shares a common command line with PEST++ as well as the input control, template files, and instruction file. Algorithmic variables that control the behavior of GSA++ are stored in a text file with a .gsa suffix. For example, control variables specific to the Method of Morris must be specified in a file that has the same base name the PEST control file, but with a .gsa extension. The variables in this file are shown in figure 6–1.

METHOD(MORRIS)

MORRIS\_R(4)

MORRIS\_P(4)

MORRIS\_DELTA(.666)

MORRIS\_POOLED\_OBS(FALSE)

**Figure 6–1.**  Example GSA++ input file for Method of Morris analysis.

**General GSA++ Options**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| METHOD | Text | “MORRIS”, “SOBOL” or “TORNADO” | Specifies type of analysis to be performed. |
| RAND\_SEED | Unsigned integer |  | Seed for the random number generator. |

**GSA++ Options Specific to Method of Morris**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| MORRIS\_R | Integer | positive integer | Sample size. The number of times the sensitivity will be computed for each parameter. |
| MORRIS\_P | Integer | positive integer | Number of levels or the number of points at which each parameter is sampled. |
| MORRIS\_DELTA | Real | multiple of  where p=MORRIS\_P | Size of the sampling step. This must be a multiple of and represent the size of the interval that will be used to calculate the sensitivities. |
| MORRIS\_POOLED\_OBS | Text | “TRUE” or “FALSE”; default is “FALSE” |  |
| MORRIS\_OBS\_SEN | Text | TRUE” or “FALSE”; default is “TRUE” | A value of “TRUE” instructs GSA++ to perform the Method of Method sensitivity for each observation. |

**GSA++ Options Specific to the Method of Sobol**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Values | Description |
| SOBOL\_SAMPLES | Long integer | positive integer | Size of the samples to be used in Sobol’s method when computing sample variances. This is “n” in the equation *s*2 = Σ ( xi − x )2 /  ( n − 1 ). |
| SOBOL\_PAR\_DIST | String | “NORM”  “UNIF” | Specifies whether the parameter samples should be drawn from a uniform or normal distribution. If the parameters are assumed to be uniformly distributed use “UNIF”; otherwise, if the parameters are normally distributed, use “NORM”. |

## GSA++ Output Files for the Method of Morris

The GSA++ implementation produces two output files summarizing the global sensitivity analysis. The Morris sensitivity file (.msn) is the primary output file which contains the metric associated with the Method of Morris analysis. The file contains a header line describing the information stored in the file, which consists of parameter\_name, sen\_mean(μ), sen\_mean\_abs(μ\*), and sen\_std\_dev(σ). Each subsequent line contains the metrics for one of the adjustable parameters. In addition to the .msn file, a raw sensitivity file (.raw) is also written which summarizes the raw model output that was used to compute the information stored in the .msn file. Each line stores a single sensitivity computed from a pair of model runs where phi\_0, phi\_1 are the values of the objective function used to compute the sensitivity; par\_0, par\_1 are the values of the adjustable parameter used to compute the sensitivity; and sen is the sensitivity.

parameter\_name, sen\_mean, sen\_mean\_abs, sen\_std\_dev

X1, -16.4665, 108.885, 138.542

X2, 53.5115, 72.4633, 98.2834

*.*

*.*

*.*

*X19, -2.32365, 5.56711, 6.42093*

*X20, -0.0338625, 2.83062, 3.88215*

**Figure 6–2.**  Example Morris sensitivity (.msn) file.

parameter\_name, phi\_0, phi\_1, par\_0, par\_1, sen

X1, 128.437, 84.7042, 0.999999, 0.333333, 65.5993

X2, 114.144, 128.437, 0.666666, 0, -21.4395

*.*

**Figure 6–3.**  Example raw sensitivity (.raw) file.

## GSA++ Output Files for Sobol’s Method

## GSA++ Output Files for Tornado Plots