**GSA++**

**Introduction**

GSA++ is the part of the PEST++ suite of tools designed to facilitate global sensitivity analysis (GSA). It leverages previous PEST++ work and uses the PEST/PEST++ input control file (.pst) as well as the PEST++ runs mangers. This makes performing GSA analysis on a problem already set-up setup to run PEST and or PEST++ trivial and extremely efficient when using one of the parallel run managers is employed.

GSA techniques are more robust than traditional derivative based local sensitivity analysis methods because they capture the sensitivities of model parameters over the entire input parameter space rather than just at a single point. There are many different global sensitivity methods. Some provide detailed information about how interactions between individual parameters affect the sensitivity and have large computational burdens, while others only provide general information on the variability on the sensitivities but have much lower computational requirements. For an in-depth introduction to GSA methods the reader is referred to Saltelli et el (2004) and (2008). GSA++ currently provides support for the Method of Morris Morris(1991) with extensions to this method proposed by Campolongo et al (2003) and Sin ((2009). There is an effort underway to also incorporate the Sobol’s method in to GSA++. Sobol’s Method use the decomposition of variance to provide information effect of parameter interactions on the sensitivity but it has much higher computational requirements than the method of Morris. Typically the Method of Morris of Morris is used in screening level analysis to find the most important parameters, and then Sobol’s method is performed on the most important parameters to analyze parameter interactions.

**Implementation of the Method of Morris**

The Method of Morris, Morris (1991) and Satelli (2004 and 2008) is a global sensitivity analysis method because it provides estimates of parameter sensitivities over their entire ranges. It is a “One at a Time” or OAT method as it only varies a single parameter at a time. The method originally proposed by Morris in his 1991 paper provides two measurements of parameter sensitivities: the mean (μ) and the standard deviation (σ). The mean measures the central tendency and captures the impact of a parameter where as the standard derivation represents the spread of the individual sensitivities around the mean and is an indicator of nonlinearities and parameter interactions. Compololongo et al. (2003) extended the method of Morris by adding the absolute mean (μ\*) as a metric. It provides a better estimate of the overall impact of parameters than the mean because it is not subject to canceling out of positive and negative values.

The method of Morris is based on elementary effects which are the sensitivities to parameters scaled to the interval [0,1]. The k dimensional vector x is defined to hold the scaled parameters where each member of x is drawn from the set 0, 1/(p-1), 2/(p-1),….,1}. Here p is a variable that defines how many intervals should be used for each parameter. Morris (1991), defines elementary effects as:

Where x is the k dimensional vector of scale model inputs as describe above, and is a predetermined multiple of 1/(p-1) and represents the size of the perturbation of the scaled parameters used in the analysis. Morris (1991) recommends choosing an even p and setting.

The method of Morris perturbs one parameter at a time and uses equation 1to compute the sensitivity. A detailed account of how this is performed can be found in Morris’ original paper Morris (1991) as well as Saltelli (2004). Briefly the parameters are perturbed one at a time and to provide a trajectory through parameter space which can be used to compute a single sensitivity with respect to each parameter. The sample size, r, is the number of trajectories used in the analysis is represents the number of time the sensitivity is computed for each parameter. Table XX summarizes the important variables in the method of Morris.

|  |  |  |
| --- | --- | --- |
| Variable | input/output | Description |
| p | input | Number of levels or the number of point each parameter is sampled at. |
|  | input | 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. |
| r | input | Sample size. The number of times the sensitivity will be computed for each parameter |
| μ | output | Mean sensitivity of the output with respect to each parameter. |
| μ\* | output | Mean sensitivity of the absolute value of the output with respect to each parameter. |
| σ | output | Standard deviation of the sensitivity of the output with respect to each parameter. |

The method of Morris summaries the sensitivity of a single model output with respect to a number of input parameters and GSA++ uses the PEST/PEST++ object phi as this single output.

**Running GSA++**

GSA++ shares a common command line with PEST++ and a .pst input control, template files and instruction file developed for PEST and used by PEST++. The command line specifies the name of the control file as well as the run manger. The control file (.pst) file specifies the parameters and observation to be used in the analysis and provides the information GSA++ needs make the forward runs. The command line for GSA++ as well as the \*.pst input file are briefly summarized below and the reader is referred to the PEST++ and PEST documentation for additional information. Variables specific to Global Sensitivity Analysis are stored in a new file with a .gsa suffix. This file is covered in “Implementation of the Method of Morris” section below.

GSA++ provides a number of command line options to 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. GSA++ relies on run mangers to complete the forward model runs and the current version provides the following three options: 1) **Y**et **A**nother Run **M**anage**R** (**YAMR**), 2) GENIE and 3) serial 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++. When using the YAMR or GENIE parallel run managers GSA++ relies on PEST++ run in slave mode to perform model runs for YAMR and the external GMAN and GSLAVE programs to manage and perform the model runs for GENIE. The serial rule manager provides a simple alternative that duplicates the functionality currently in regular PEST. The command lines required to start GSA++ are summarized in the following table.

|  |  |
| --- | --- |
| **Run Manger / Mode** | **Command** |
| Serial Run Manager / Master | GSA++ control\_file.pst |
| YAMR / Master | GSA++ control\_file.pst /H :port |
| YAMR / Compute Node | pest++ /H hostname:port |
| Genie / Master | GSA++ control\_file.pst /G hostname:port |

GSA++ provides support for resuming a run that failed due to network or system problems through the /r command line option.

**Implementation of the Method of Morris**

As discussed in the previous section, GSA++ reuses the PEST/PEST++

.gsa file

METHOD(MORRIS)

MORRIS\_R(4)

MORRIS\_P(4)

MORRIS\_DELTA(.666666)

MORRIS\_POOLED\_OBS(FALSE)

PHI\_NORM(1)