PSUADE Reference Manual
(Version 1.6)

Charles Tong

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

Livermore, CA 94551-0808

September, 2013

1   Introduction

PSUADE (Problem Solving environment for Uncertainty Analysis and Design Exploration) is a software package for various uncertainty quantification (UQ) activities such as uncertainty assessment (UA), sensitivity analysis (SA), parameter study, numerical optimization, etc. It consists of three major components: a suite of sampling methods, a job execution environment, and a collection of analysis tools. In addition, it provides a command line interpreter to allow users to interactively perform some of the analyses. 'Command line' mode is activated by calling PSUADE without argument; that is,

```
[Linux] psuade
```

This document describes some of the available commands for the command line (or interactive) mode (most of the commands can be found by issuing the 'help' command).

2   PSUADE Commands

The 'interactive mode' commands can be divided roughly into the following sections.

2.1   Commands for Read/Write from/to Files

run <**file**> : where <file> is a PSUADE input or data file.

> This command is equivalent to running PSUADE with <file> as an argument (which allows you to execute a batch script without leaving the PSUADE interactive session).

load <**file**> : where <file> is a PSUADE data file.

> After issuing a load, the sample data are in PSUADE's local memory and ready to be analyzed or manipulated.

loadmore <**file**> : where <file> is a PSUADE data file.

> This command should be used to add another sample to the existing one loaded previously by the load command (issuing a 'load' again will replace the current sample in PSUADE's local memory.

write <**file**> : where <file> is a PSUADE data file.

> This command writes a sample which has been loaded and manipulated previously to another file in the standard PSUADE data format. You will be asked to save the sample with just one output or all outputs.

nwrite <**file**> : where <file> is a data file.

> This command writes only the unevaluated sample points (those with sample state = 0 or if any of the outputs is undefined) to the specified file.

iwrite <file> : where <file> is a data file.

>This command writes ONLY the sample inputs to a file which will have the first line specifying the sample size and subsequent lines having the sample inputs.

owrite <file> : where <file> is a data file.

>This command writes ONLY the sample outputs to a file which will have the first line specifying the sample size and subsequent lines having the sample outputs.

read_std <file> : where <file> is a data file.

>This command reads in a sample which has been specified in a standard format which has the first line specifying the sample size, the number of inputs, and the number of outputs; and subsequent lines specifying each sample input and output values (one sample point per line). An example is:

```
4 2 1
0.0 0.0 0.0
1.0 0.0 1.0
0.0 1.0 1.0
1.0 1.0 2.0
```

write_std <file> : where <file> is a data file.

>This command writes in standard format (see read_std for the format) the sample previously loaded into the PSUADE memory.

write_matlab <file> : where <file> is a data file.

>This command writes in matlab format the sample previously loaded into the PSUADE memory (the sample inputs will be stored in the matrix X and sample outputs in the matrix Y).

read_xls <file> : where <file> is a data file.

>This command reads in a sample which has been specified in a format that can be used with the Excel spreadsheet. The data format has the first line specifying the sample size, the number of inputs, and the number of outputs; and subsequent lines specifying the sample number and sample input and output values (one sample point per line). An example is:

```
4 2 1
1 0.0 0.0 0.0
2 1.0 0.0 1.0
3 0.0 1.0 1.0
4 1.0 1.0 2.0
```

write_xls <file> : where <file> is a data file.

This command writes in xls format (see read_xls for the format) the sample previously loaded into the PSUADE memory.

write_ultra <file> : where <file> is a data file.

This command writes a sample previously loaded to a file in the 'ultra' format. It can only handle samples which have two inputs. If you do not know what 'ultra' format is, then mostly likely you do not need to use this command.

update <file> : where <file> is a standard PSUADE data file.

This command updates the sample previously loaded with the sample from the file <file>. The sample points that have been evaluated in <file> will be used to update the corresponding (same) unevaluated sample points in the local memory.

iadd <file> : where <file> is a standard PSUADE data file.

Suppose you have a PSUADE data file, and you would like to add a few more inputs to this file. You will have to prepare another PSUADE data file (<file>) which has the same sample size but different sample inputs. For example, if you have 2 inputs in local memory and 3 inputs in the data file, then after this command, the total number of inputs will be 5. You can write this new sample using write. This command is useful if you would like to add a few dummy input variables in your data file.

oadd <file> : where <file> is a standard PSUADE data file.

This command is analogous to iadd. This command only checks for the same sample size in both samples and it does not check that the two sets of sample inputs are exactly the same (so beware).

ireplace <file> : where <file> is a data file.

This command is similar to iadd except that instead of adding more inputs, it replaces ONE of the current inputs by another input in <file>. The format of the data file is: sample size in the first line and input values in subsequent lines. An example is:

```
4
0.0
1.0
2.0
3.0
```

oreplace <file> : where <file> is a data file.

This command is similar to ireplace except that it operates on ALL sample outputs. The format of the data file is: sample size and number of outputs in the first line and output values in subsequent lines (each line has all the outputs of one sample point). An example is:

```
4 2
0.0 1.0
1.0 4.0
2.0 3.0
3.0 2.0
```

This command may be useful if you run the sample yourself and you would like to update your PSUADE sample file with the simulation outputs.

splitdata : split the currently loaded sample into two subsets.

This command splits the sample in local memory into two subsets. It will ask for the sample size of the first subset. The two subsets are stored in psuadeSplit1 and psuade-Split2, respectively. Note: the sample in PSUADE's local memory will be destroyed after this command, so you need to re-load.

2.2   Commands for Basic Statistics

ua : uncertainty analysis on the sample in local memory.

This command should be used after a sample is loaded into local memory. It displays basic statistics information such as mean, standard deviation, skewness and kurtosis. By turning on interactive mode (by issuing 'interactive'), options are provided to generate MATLAB distribution plots.

ca : correlation analysis on the sample in local memory.

This command should be used after a sample is loaded into local memory. It displays correlation information such as Pearson, Spearman and Kendall coefficients.

anova : analysis of variation on the sample in local memory.

This command should be used after a sample is loaded into local memory. It performs and displays the results of an analysis of variation on the loaded sample.

1stest : perform one-sample tests.

This command should be used after a sample is loaded into local memory. It performs one-sample tests on the sample such as the Chi-squared test or the distribution test (search for mean and standard deviation for the sample).

2stest : perform two-sample tests.

This command performs two-sample tests on the sample such as the Student T-test (do two normally distributed populations differ?), the Mann-Whitney test (do the two sample have the same probability distribution?), and Kolgomorov Smirnov test (does the sample come from a population with a specific distribution with a given mean and standard deviation?). You will have to enter 2 data files in the format instructed by this command.

gendist : create a data set using the specified PDFs.

This command creates a ONE-INPUT sample based on the selected probability distribution. The result sample is written to a file 'sample1D' which has the first line specifying the sample size and subsequent lines specifying the sample input values.

pdfconvert : convert a sample based on its probability distribution.

This command should be used after a sample is loaded into local memory. It converts the sample inputs (assumed uniform to begin with) based on its probability distribution (defined in the INPUT section of the data file itself) and replaces the sample inputs with the converted values. The results can be written to a PSUADE data file using write.

rand_draw : draw a sample from the loaded sample with replacement.

This command should be used after a sample is loaded into local memory. It draws from the loaded sample a bootstrapped sample with replacement. The results can be written to a PSUADE data file using write.

rand_draw2 : draw a sample from two samples with two different sets of inputs.

When initiated, this command will request two sample files in PSUADE data format and then randomly draws a new sample of a given size.. The results can be written to another PSUADE data file using write.

2.3 Commands for Parameter Screening (Input Dimension Reduction)

moat : Morris screening analysis on the sample in local memory.

This command should be used after a sample is loaded into local memory. It displays screening information such as modified mean and standard deviations of the Morris gradients. Optionally, analysis results can be saved in a matlab/scilab file for graphical display. Use printlevel and/or interactive commands to turn on graphics generation.

delta_test : perform Delta test for screening input parameters.

This command should be used after a sample is loaded into local memory. During the optimization, the configurations and their corresponding objective values are displayed. Then the ranking results will be displayed. Finally, there is a final test that incrementally adds the the identified important parameters to the list and computes the objective values starting from the most important parameter. You can plot the objective values. If you observe a 'tub'-like curve, the recommended parameter partitioning (into important and unimportant ones) is around the bottom of the curve.

sot_sa : perform sum-of-trees screening analysis.

This command should be used after a sample is loaded into local memory. It displays screening information based on a sum-of-trees analysis.

ff : perform fractional factorial screening analysis.

This command should be used after a sample is loaded into local memory. It displays screening information based on a fractional factorial analysis. This command works only with fractional factorial samples.

lsa : perform local sensitivity screening analysis.

This command should be used after a sample is loaded into local memory. It displays screening information based on a gradient analysis. This command works only with LSA samples.

mars_sa : MARS screening analysis on the sample in local memory.

This command should be used after a sample is loaded into local memory. It display screening information based on the MARS importance ranking. This information will not be generated on some platform where MARS outputs are suppressed and is signaled by an error.

gp_sa : Gaussian process screening analysis on the sample in local memory.

This command should be used after a sample is loaded into local memory. It displays screening information based on importance ranking with Gaussian process (if the Gaussian process package has been installed).

pca : perform principal component analysis on the sample outputs.

This command should be used after a sample is loaded into local memory. PSUADE performs singular value decomposition on the sample outputs (normalized by mean and standard deviation of each output). Specifically, given the output matrix Y, the following decomposition is performed:

$$Y = USV^T$$

where Y is a matrix with N rows (sample size) and M columns (number of outputs), U is a matrix of size $M \times N$ (left singular vectors), S is a diagonal matrix of size $N \times N$, and V is a matrix of size $N \times N$ (right singular vectors). Upon exit from this command, the file 'psPCA.out' will contain the projection of the sample outputs onto the principal components V (that is, $Y \times V$ or $U \times S$). If you decide to use these principal components, you can use 'oreplace' to isert the file 'psPCA.out' to your PSUADE sample file.

2.4 Commands for Response Surface Analysis

svmfind : find the best parameter setting for SVM response surface.

This command should be used after a sample is loaded into local memory. The suppot vector machine (SVM) has two parameters. This command tunes these parameters to obtain the best training errors.

rscheck : check the quality of the response surface.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a response surface and then checks the training errors (to appear in the file 'psuade_rsfa_err.m' if print level has been set to 4). By turning on the 'interactive' mode, cross validation can also be activated. You will need to select how many groups (that is, the k in k-fold cross validation). At the end of cross validation, the file 'RSFA_CV_err.m' will be created for visualizing the distribution of cross validation errors. If the errors are not acceptable, you can refine the sample.

rstest : check interpolation errors of a response surface with a data set.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command builds a reponse surface for the loaded sample and asks for a test set (in standard PSUADE data format) to compute the interpolation errors. There are four different types of response surface test:

- rstest_ts: use the training test
- rstest_hs: hold out test set
- rstest_cv: k-fold cross validation
- rstest_gt: gneralization test

rscreate : create a response surface from the loaded sample.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command builds a reponse surface for the loaded sample. This command is to be used with 'rseval'.

ivec_create : create a single sample point in local memory.

You can create a local sample point in the local memory for use with 'rscreate'. Use 'ivec_modify' and 'ivec_show' to manipulate and read this internal sample point. The initial values are the mid points in each input.

ivec_modify : modify the content of the sample vector in local memory.

This command is to be used after an internal sample vector has been created using 'ivec_create'.

ivec_show : disply the content of the sample vector in local memory.

This command is to be used after an internal sample vector has been created using 'ivec_create'.

rseval : evaluate a data point from the response surface built from the loaded sample.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory and after recreate has been called. You can evaluate a single sample point from the local sample vector (created by 'ivec_create' or from a file. The file should

have PSUADE_BEGIN as its first line, the number of inputs as the second line, the input values in subsequent lines (each line begins with the input number followed by the input value), and PSUADE_END as the last line.

rseval_m : this command uses PSUADE as a response surface server. This command waits for users to put the new sample point in a file, evaluates the sample, and loops back.

rs_splot : create scatter plots from a response surface.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It is similar to splot except that the sample is taken from sampling the response surface instead of from the original sample. The scatter plots are created in the file 'matlabrssp1.m'/

rstgen : create a sample for response surface test.

Response surfaces usually give poor interpolations at the corners or edges or faces of the parameter space. This command creates a sample at the corners of the parameter space, and is to be evaluated and then used with 'rstest' for validating the response model at the corners. Depending on the number of inputs, we may choose full factorial or fractional factorial sample design.

rsvol : calculate volume of the constrained space.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command builds a reponse surface for the loaded sample, creates a large sample, and tabulates the number of sample points that satisfy a given constraint. The percentage of feasible sample points is the percentage of the feasible space in the entire parameter space.

int : perform numerical integration.

This command should be used after a sample is loaded into local memory. It uses the data to compute the integral under the response curve.

2.5 Commands for Quantitative Sensitivity Analysis

rs_ua : perform uncertainty analysis on a fuzzy response surface.

rs_uab : perform uncertainty analysis on a fuzzy response surface with bootstrapped aggregation and optional discrepancy function.

rs_uap : perform uncertainty analysis on a fuzzy response surface with a user-provided sample.

me : perform variance-based main effect analysis.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays first-order Sobol' indices for the sample. Ideally, this sample should be a replicated Latin hypercube sample. It will still work if it is not.

ie : perform variance-based two-way interaction effect analysis.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays second-order Sobol' indices for the sample. Ideally, this sample should be a replicated orthogonal array sample. It will still work if it is not.

tsi : perform variance-based total sensitivity analysis on the raw sample which has loaded in the local memory.

rssobol1 : perform variance-based main effect analysis on a RS.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays first-order Sobol' indices for the response surface built from the loaded sample (instead of from the raw sample in 'me'). Constraints can be imposed by using 'rs_constraint' in the ANALYSIS section of the PSUADE sample file. For a more thorough analysis, it is recommended that the more computationally expensive command 'rssobol1b' be used instead.

rssobol1b : perform variance-based main effect analysis on a RS with bootstrapping.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays first-order Sobol' indices for the response surface built from the loaded sample. These indices are computed by performing first-order Sobol' analysis multiple times with different bootstrapped samples. As thus, the analysis results will include standard deviations of the computed Sobol' indices.

rssobol2 : perform variance-based two-way interaction analysis on a RS.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays second-order Sobol' indices for the response surface built from the loaded sample (instead of from the raw sample in 'ie'). Constraints can be imposed by using 'rs_constraint' in the ANALYSIS section of the PSUADE sample file.

rssoboltsi : perform variance-based total sensitivity analysis on a RS.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays total-order Sobol' indices for the response surface built from the loaded sample. Constraints can be imposed by using 'rs_constraint' in the ANALYSIS section of the PSUADE sample file. For a more thorough analysis, it is recommended that the more computationally expensive command 'rssoboltsib' be used instead.

rssoboltsib : perform variance-based total sensitivity analysis on a RS with bootstrapping.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays total-order Sobol' indices for the response surface built from the loaded sample. These indices are computed by performing total-order Sobol' analysis multiple times with different bootstrapped samples. As thus, the analysis results will include standard deviations of the computed Sobol' indices.

rssobolg : perform variance-based group sensitivity analysis on a RS.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It displays group-order Sobol' indices for the response surface built from the loaded sample. You need to provide a file (instruction given when you run this command) to specify the groupings. Constraints can be imposed by using 'rs_constraint' in the ANALYSIS section of the PSUADE sample file.

ae_ua : response surface-based mixed aleatory-epistemic uncertainty analysis. This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command can perform outer-inner iterations to create cumulative distribution functions for mixed aleatoric-epistemic uncertainties.

so_ua : response surface-based second order uncertainty analysis.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command performs second level uncertainty analysis given uncertainties about the parameter lower and upper bounds.

2.6   Commands for Optimization/calibration

rsmcmc : perform MCMC on a response surface.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It performs a Bayesian inference using MCMC on the response surface built from the sample in the local memory. You will need to answer a few questions to set up the optimization problem. Upon completion, the file 'matlabmcmc.m' will have the posteriors of individual inputs, and 'matlabmcmc2.m' will have the pairwise posteriors.

mcmc : perform MCMC on a simulator.

This command should be used after a PSUADE input file (and no need for a generated sample, just the definition of inputs and outputs) has been loaded into PSUADE's local memory. It performs a Bayesian inference using MCMC on the simulator pointed to by the 'driver' (in ANALYSIS section) in the PSUADE input file. Since MCMC requires hundreds to thousands of simulations, it is time-consuming even when the simulators are fast to run (due to I/O operations, as PSUADE uses file-based protocol with the simulators). Hence, it is recommended that the simulator be compiled and linked with the PSUADE executable to greatly speed up this procedure. When launched, this command will prompt a few questions to set up the optimization problem. Upon completion, the file 'matlabmcmc.m' will have the posteriors of individual inputs, and 'matlabmcmc2.m' will have the pairwise posteriors.

mo_opt : response surface-based multi-objective optimization.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This function is useful if you have more than 1 output and the objective

function (for optimization) is formed by weighted combination of these outputs where the weights are uncertain. So given the ranges of these weights, this command generates optimal solution at the lattice points in this weight space.

2.7   Commands for Creating Visualization Plots

splot : generate scatter plots of an output against each input.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates scatter plots with each input on the X-axis and an output on the Y-axis. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

splot2 : generate scatter plots of an output against 2 inputs.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates scatter plots with 2 selected inputs on the X- and Y-axes and an output on the Z-axis.

rs1 : create an one-input response surface plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates an one-input response surface plot against a sample output. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

rs2 : create a two-input response surface plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates an two-input response surface plot against a sample output. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

rs3 : create a three-input response surface plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates an three-input response surface plot against a sample output. The output values are represented by a color chart. The output file is currently only available in 'matlab' format.

rs3m : create a three-input response surface plot (movie).

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates an three-input response surface plot against a sample output. The third input is represented by the time axis. The output file is currently only available in 'matlab' format.

rs4 : create a four-input response surface plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates an four-input response surface plot against a sample output. The fourth input is represented by the time axis and the output values are represented by a color chart. The output file is currently only available in 'matlab' format.

rssd : create a response surface error standard deviation plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. The response surface methods used here for estimating the interpolation error is Gaussian process, MARS with bootstrapped aggregation, Kriging, or polynomial regression. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

rssd_ua : perform uncertainty analysis of error standard deviation from response surface fit.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It uses the sample to create a response surface (Gaussian process, MARS with bagging, Kriging, or polynomial regression) and then use the error standard deviations from response surface analysis to generate basic statistics such as mean and standard deviation. This command is useful to estimate errors and distributions of errors induced by the use of response surfaces. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

rsi2 : create a two-input response surface intersection plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. The sample should have 2 or more outputs. Imposing constraints on the two selected outputs will result in a smaller feasible space which can be visualized in 'matlab'.

rsi3 : create a three-input response surface intersection plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command is similar to rsi2 except that it creates a three-input intersection plot. This plot capability is currently only available in 'matlab'.

rsi3m : create a three-input response surface intersection plot (movie).

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. This command is similar to rsi2 except that it creates a a three-input response surface plot against a sample output. The third input is represented by the time axis. This plot capability is currently only available in 'matlab'.

rawi2 : create a two-input intersection plot using raw data.

This command is similar to rsi2 except that it assumes the sample is already a factorial design and thus no response surface interpolation is required. This plot is currently available in 'matlab' format only.

rawi3 : create a three-input intersection plot using raw data.

This command is similar to rawi2 except that it creates a three-dimensional intersection plot. It assumes the sample is already a factorial design and thus no response surface interpolation is required. This plot is currently available in 'matlab' format only.

rspairs : create response surfaces for all 2-input pairs.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a number of sub-plots each is a 2-input plot for all pairs of the selected inputs. The plots are currently available in 'matlab' format.

rsipairs : create intersection plots for all 2-input pairs.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a number of sub-plots each is a 2-input plot for all pairs of the selected inputs. All the outputs will be used as constraints. The plots are currently available in 'matlab' format.

iplot1 : create a one-input sample input plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a sample input scatter plot for a selected input (sample number on the X-axis and input values on the Y-axis). You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

iplot2 : create a two-dimensional sample input plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a sample input scatter plot for two selected inputs (input 1 number on the X-axis and input 2 on the Y-axis). You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

iplot3 : create a three-dimensional sample input plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a sample input scatter plot for three selected inputs (on X-, Y-, and Z-axes). You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

iplot4m : create a three-input response surface plot (movie).

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a sample input scatter plot for four selected inputs (on X-, Y-, Z-, and t-axes; that is, the fourth input is represented by the time axis). The output file is currently only available in 'matlab' format.

iplot2_all : create two-dimensional sample input plots for all input pairs.

This command should be used after a sample is loaded into local memory. It is similar to iplot2 except that the same operation is performed on all input pairs. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

iplot_pdf : plot the sample PDF for selected inputs.

This command should be used after a sample is loaded into local memory. It creates histogram plots for each selected input of the loaded sample. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

oplot2 : create a two-dimensional sample output plot.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. It creates a sample output scatter plot for two selected outputs (output 1 number on the X-axis and output 2 on the Y-axis). You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

iotrace : plot inputs/outputs of each run.

This command should be used after a sample is loaded into local memory. It generates a plot with the number of lines equal to the sample size and with each line connecting all inputs and outputs of one sample point.

2.8 Commands for Setting Up PSUADE Runs

setupguide : display a short guide to help set up an application.

geninputfile : create a PSUADE input file.

genbatchfile : create a batch file.

Since many applications at LLNL use our Livermore Computing machines where jobs are submitted in batch mode, this command helps to set up the batch jobs.

gendriver : create a PSUADE driver program.

A driver program is needed in a PSUADE input file. This driver program takes the input parameter values from a file (argument 1), substitutes the data into the application input decks, runs the application, postprocesses the application outputs, and writes the output data to another specified file (argment 2). This command helps to create this driver program.

genexample : create a PSUADE example.

This command creates a simple PSUADE example. After the example has been created, simply run **psuade psuade.in** to see the results.

chkjobs : monitor the status of job runs.

2.9 Command for Editing the Sample in PSUADE's Local Memory

validate : change the states of selected sample points to be 'evaluated'.

This command should be used after a sample is loaded into local memory. The selected sample points will be considered evaluated in local memory and can be saved using 'write'.

invalidate : change the state of selected sample points to be 'evaluated'.

This command should be used after a sample is loaded into local memory. The selected sample points will be considered 'evaluated' in local memory and can be saved using 'write'.

srandomize : randomly change the order of the sample points.

ifilter : take out sample points that do not satisfy certain input conditions (bounds).

This command should be used after a sample is loaded into local memory. It removes all sample points having a selected input value falling outside the given upper and lower bounds.

ofilter : take out sample points that do not satisfy certain output conditions.

This command should be used after a sample is loaded into local memory. It removes all sample points having a selected output value falling outside the given upper and lower bounds.

idelete : remove an input from all sample points.

This command should be used after a sample is loaded into local memory.

odelete : remove an output from all sample points.

This command should be used after a sample is loaded into local memory.

sdelete : remove a sample point.

This command should be used after a sample is loaded into local memory.

ishuffle : re-arrange the orders of sample inputs.

This command should be used after a sample is loaded into local memory. You can use this command to re-arrange the sample inputs in a different order.

disp_sample : display one sample point.

This command should be used after a sample is loaded into local memory.

curr_sample : display information about the current sample.

list1 : list one sample output along with one sample input.

This command should be used after a sample is loaded into local memory. Options are provided for sorting the inputs or the outputs before displaying.

list2 : list one sample output along with two sample inputs.

This command should be used after a sample is loaded into local memory. Options are provided for sorting the inputs or the outputs before displaying.

max : display the sample point with the maximum output value.

This command should be used after a sample is loaded into local memory.

min : display the sample point with the minimum output value.

This command should be used after a sample is loaded into local memory.

irerange : re-generate a sample input using a different range.

This command should be used after a sample is loaded into local memory. It is intended for local sample refinement whereby it is desired to create another sample with has the same coordinates for most input parameters except a selected input which is shrunk to a smaller range.

ireset : map the range of an input to a distinct value.

This command should be used after a sample is loaded into local memory. It is intended for converting continuous variables into discrete (from ranges to discrete values).

oreset : map the range of an output to a distinct value.

This command should be used after a sample is loaded into local memory. It is intended for converting continuous variables into discrete (from ranges to discrete values).

ifloor : map an input to the nearest smaller integer.

This command should be used after a sample is loaded into local memory. It is intended for converting continuous variables into discrete values.

iceil : map an input to the nearest larger integer.

This command should be used after a sample is loaded into local memory. It is intended for converting continuous variables into discrete values.

itran : transform selected inputs to their logarithmic or exponential values.

otran : transform selected outputs to their logarithmic or exponential values.

itag : tag sample points with a selected input failling outside some bounds.

This command should be used after a sample is loaded into local memory. Sample points with selected input values that falls outside the specified bounds will be tagged and the untagged sample indices will be displayed. Calling this command or 'otag' again will accumulate the tagged sample points. The tagging can be reset after a 'load'.

otag : tag sample points with a selected output failling outside some bounds.

This command should be used after a sample is loaded into local memory. Sample points with selected output values that falls outside the specified bounds will be tagged and the untagged sample indices will be displayed. Calling this command or 'itag' again will accumulate the tagged sample points. The tagging can be reset after a 'load'.

purge : take out failed sample points.

This command should be used after a sample is loaded into local memory. It removes all sample points that have at least one output indicated as undefined ($1.0e35$) or the status flag turned off.

rm_duplicates : take out duplicate sample points.

This command should be used after a sample is loaded into local memory. It removes all duplicate sample points.

oop : perform a certain linear combination operation on the outputs.

This command should be used after a sample is loaded into local memory. Three outputs should be selected: O1, O2 and O3. Also, two scalar numbers are to be entered: a1 and a2. This command performs the following operations on all sample points: O1 = a1 * O2 + a2 * O3.

setdriver : set the driver field of the current PSUADE sample.

2.10 Other Miscellaneous Commands

run : run a PSUADE input file.

This command ('run psuade.in') is equivalent to running the batch mode 'psuade psuade.in'.

quit : exit the current PSUADE session.

rsmax : change the maximum sample size for response surface.

sys : initiate Linux shell command.

This command calls the external shell command with a C system call. The string attached to the end of this command (separated by a space) will be passed to the shell for execution. This general features replace the specific feature such as the previously support 'ls' command which is recognized only on Linux.

printlevel : change the diagnostics print level.

Depending on the diagnostics level, different types of information are displayed on the console. For example, setting print level to 4 will activate the cross validation in rscheck.

rename_file : rename a file.

> This command is similar to the Unix 'mv' command. It is provided here for convenience since the load command cannot take psuadeData as a valid file name so one may want to change the name of this file before loading.

sqc : check sample quality.

> This command should be used after a sample is loaded into local memory. It checks the quality of the sample by computing the max-min distances between all pairs of sample points. The minimum distance between any 2 sample points should be as large as possible.

> This command should be used after a sample is loaded into local memory. It finds the nearest neighbor for each sample point and then computes the distance to the nearest neighbor. The distance statistics are stored in a 'matlab/scilab' file for display. If the gradient of a sample point with respect to its nearest neighbor is large, it may indicate an 'outlier'.

nna : perform nearest neighbor analysis for outlier detection.

scilab : a toggle between Matlab or Scilab graphics outputs.

start_matlab : start Matlab without leaving the PSUADE command line interpreter.

show_format : disply different PSUADE file formats.

> User-provided information are often needed in design and analysis. These information are provided by users to PSUADE at various stages. This command lists several such file formats.

2.11   Advanced Commands

io_expert : toggle the I/O expert mode.

> This expert mode will affect the internal I/O operations by prompting the users more questions.

rs_expert : toggle the response surface expert mode.

> This expert mode will affect the internal response surface methods by prompting the users more detailed information.

ana_expert : toggle the analysis expert mode.

> This expert mode will affect the internal analysis methods by prompting the users more detailed information.

sam_expert : toggle the sampling expert mode.

> This expert mode will affect the internal sample generation methods by prompting the users more detailed information.

opt_expert : toggle the optimization expert mode.

This expert mode will affect the internal optimization methods by prompting the users more detailed information.

genconfigfile : create a PSUADE configure file.

Some functions in PSUADE makes use of configuration files. This command spits out an example configuration file. Features can be activated by uncommenting the lines.

use_configfile : tell PSUADE to use a configure file.

Some functions in PSUADE makes use of configuration files. This command gets a PSUADE configure file and lets the different internal modules to get their parameters from this file.

refine : refine a sample uniformly.

This command should be used after a sample is loaded into local memory. The loaded will be uniformly refined and the enlarged sample can be written to another PSUADE file for further processing. This command can operate only on a few sampling designs (LH, MC, LPTAU, METIS).

a_refine : refine a sample adaptively.

This command should be used after a sample is loaded into local memory. The loaded will be adaptively refined and the enlarged sample can be written to another PSUADE file for further processing. This command can operate only on METIS samples. Cubic splines (MARS) with bootstrapped aggregation will be used to estimate where refinement should be performed.

moatrepair <**file**> : where <file> is to store the Morris repair data.

The Morris design may require some inputs to be repaired due to the presence of constraints that cause the sample space to be non-hyper-rectangular. To create a Morris sample that stays inside the feasible parameter space, the moatgen command should be used first. The outputs from moatgen are to be used to repair a Morris sample. So the steps to create a Morris sample given some constraints are:

1. create a standard Morris sample (in the hyper-rectangular space)
2. use moatgen to create the repair file (say, repairData)
3. change the second line of the repair file to match up the input numbers
4. start PSUADE command line mode and load the Morris sample
5. use this command (moatrepair repairData)
6. write the repaired sample to a file (use write)

gmoatrepair <**file**> : where <file> is to store the Morris repair data.

This command is similar to moatrepair except that the original sample should be a generalized Morris (GMOAT) sample.

moatgen : generate a Morris repaired design.

This command creates a Morris repair file. More details is found in the description of moatrepair above. A PSUADE data file needs to be loaded first before moatgen can be used. The loaded sample is used to provide the constraints for creating the repair file. This command will ask for the desired resolution, which determines the width of the base where gradients will be computed (default is 4).

moatgen2 : generate a Morris repaired design for multiple constraints.

This command is similar to moatgen except that this command can handle multiple constraints. This command does not require a PSUADE data file be pre-loaded (but at least a PSUADE input file needs to be pre-loaded). Constraint files will be requested. The format for the constraint file can be found in the 'show_format' command.

moat_concat <**file**> : concatenate two Morris samples.

This command concatenates two Morris samples with two different sets of inputs. If the first sample has n_1 inputs and the second sample has n_2 inputs, then the final sample will have $n_1 + n_2$ inputs. The two samples should have the same number of replications. For example, if the sample in the PSUADE local memory has 10 inputs and 110 sample points and <file> has 5 inputs and 60 sample points (number of replications = 10), then the combined sample will have $10 \times (10 + 5 + 1) = 160$ sample points. The combined sample will be in local memory and needs to be stored.

interface_track : create a response surface for the interface defined by the output values (0 or 1).

This command may be useful to characterize a discontinuity in the parameter space based on the values of the given sample output. If the neighbors of a sample point with output '0' has a value of '1', this sample point is a candidate to be considered as on the interface.