PSUADE Reference Manual (Version 1.4)

Charles Tong
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA 94551-0808 [1]
Oct, 2011

[1]This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344.

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 just calling

```
[Linux] psuade
```

without any argument. This document describes the available commands for the interactive mode (most of the commands can be found by the 'help' command).

2 PSUADE Commands

The interactive 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. Examples of PSUADE input file can be found in the *Examples* directory.

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. You can edit the sample with commands given in later sections.

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). This 'another' sample should match the existing sample in the number of inputs and outputs.

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

This command writes a sample which has been loaded and manipulated previously to <file> in the 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.

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
```

Using this command with 'write' provides conversion between the standard and the PSUADE formats.

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.

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 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 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 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 psuadeSplit2, respectively. This command is useful, for example, to partition a sample into a training set and a test set. Note: the sample in PSUADE's local memory will be destroyed after this command, so you need to re-load.

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

The Morris design may require some inputs to be modified 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 modify 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.

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 analysis expert mode (by issuing 'ana_expert'), 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 : perform analysis of variation on the MARS response surface created from 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.

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 screening analysis based on fractional factorial sampling.

This command should be used after a sample is loaded into local memory. It displays first and second-order screening information based on a fractional factorial analysis. This method is suitable if monotonic input-output relationship is expected. This method assumes the use of FF4, FF5 or PBD sampling design. The second order analysis will be performed only for FF5 samples.

lsa : perform screening analysis based on local sensitivity analysis.

This command should be used after a sample is loaded into local memory. It displays first-order screening information based on a local sensitivity analysis. This method assumes the LSASampling method is used. It is suitable if linear input-output relationship is expected within the parameter ranges.

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.

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, which are also written to 'psuade_err_file.m' for visualization with MATLAB.

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' or 'rseval_m'.

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 : evaluate repeated a set of data points 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. This command is intended to be a response surface evaluation server to repeatedly evaluate new sample points from users especially when the response surface creation is time-consuming (so that once it is created, it can be evaluated repeatedly). Simply populate the 'rsevalDataIn.X' file with a set of new sample points, wait for the response surface evaluation to complete, and harvest the results in the 'rsevalDataOut.X' file. Once you are done with all evaluations, create an empty file called 'psComplete' in the current directory to terminate this command.

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

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.

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.

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.

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.

rs_qsa : perform variance-based sensitivity analysis repeatedly.

This command should be used after a PSUADE sample has been loaded into PSUADE's local memory. You can select first- or total-order analysis. The analysis will be repeated a number of times (since each analysis uses random samples) and statistics about the sensitivity measures will be calculated. Thus, this command computes not just the sensitivity indices, but also uncertainties about these indices. However, this analysis will take quite some computation time.

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 can perform outer-inner iterations to create cumulative distribution functions for mixed aleatoric-epistemic uncertainties. Additionally, this command can perform 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.

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 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 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 plot against a sample output. The output values are represented by a color chart. You can create a 'matlab' or 'scilab' file by toggling the 'scilab' command.

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 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 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 method used here for estimating the interpolation error is Gaussian process or MARS with bootstrap aggregation. 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 and then use the error standard deviations from response surface analysis to generate basic statistics such as mean and standard deviation. 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 plotted with 'matlab'.

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

This command is similar to rsi2 except that it creates a three-input intersection plot 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. The plot is currently available in 'matlab' format.

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.

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).

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 iplot2d except that the same operation is performed on all input pairs. 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.

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.

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.

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'.

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.

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 an input to a set of distinct values.

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.

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.

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 General 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.

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.

rsmax : change the maximum sample size for response surface.

ls : list the files in current directory.

This command is the same as the Linux 'ls' command (so you do not need to quit before listing files in the current directory).

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.

interactive : toggle the interactive mode.

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.

2.11 Advanced Commands

expert : toggle the expert mode.

io_expert : toggle the I/O expert mode.

rs_expert : toggle the response surface expert mode.

ana_expert : toggle the analysis expert mode.

sam_expert : toggle the sampling expert mode.

opt_expert : toggle the optimization expert mode.

scilab : toggle between the matlab/scilab mode.

Some functions in PSUADE only generates MATLAB plots and not scilab plots.

start_matlab : start MATLAB in the command line mode.

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.