PSUADE: A Software Toolkit for Uncertainty Quantification

Charles Tong
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

April, 2016
What is PSUADE?

- **Problem Solving environment for Uncertainty Analysis and Design Exploration**
- **In development since 2005**
- **Designed mainly for non-intrusive UQ analysis**
  - Some support for semi-intrusive methods (e.g. derivatives)
  - Support moderate input dimensions ($O(1) - O(100)$)
  - For high dimension ➔ DASSI
  - For hybrid intrusive ➔ MEDUSA
  - For graphical user interfaces ➔ e.g. FOQUS
- **First public release (v. 1.0) in Oct 2007 (LGPL license)**
  - Current release v. 1.7.6a (partially parallelized)
- **Platform:** Linux, MacOS, Windows
- **Language:** C++, C, Fortran (external libraries)
- **Installation tools:** cmake, compilers
History of PSUADE

▪ Began as a library of sampling designs and simple response surfaces (spinoff from ‘doomsdace’)

▪ Since then, the PSUADE development goal (and challenges) has been for supporting the UQ of multiphysics applications
  • Moderate number of parameters (with PDFs/correlations)
  • Simulation models are nonlinear and may be computationally intensive
  • Simulation model may be imperfect ➔ model form uncertainties
  • Multiple levels: units, sub-systems, full system ➔ hierarchical UQ ➔ complex UQ workflow, diverse UQ methods, diagnostics
  ➔ Q: how to account for all errors in a UQ analysis (errors due to sample size and response surface interpolation/extrapolation)

▪ How has PSUADE been used?
  • As a standalone software (batch mode, command line interpreter)
  • Integrated as part of a larger software package (shared libraries)
  • Integrated into some graphical user interfaces
Relevant methods for multi-physics applications

- **Dimension reduction methods are needed**
  - Local sensitivity analysis generally not suitable
  - Classical methods such as SRC may not be sufficient
  - **Nonparametric methods needed for nonlinear problems**

- **Many runs may be needed to resolve nonlinearities/interaction**
  - Need parametric/nonparametric response surface methods
  - Adaptive response surface methods may help
  - Response surface errors are needed in analysis

- **Global sensitivity analysis are more suitable**
  - Variance decomposition: first order, second order, total order
  - Accommodate different input probability distributions and correlations

- **Hierarchical/multi-stage parameter inference methods may be needed**
  - Empty set (no feasible space) may be encountered

- Many data manipulation/visualization/diagnostics tools are needed

- **Other useful tools: optimization and optimization under uncertainty**
PSUADE contains a suite of typical UQ capabilities:

- Optimization under uncertainty
- Optimization (single/multi-fidelity)
- Bayesian inference
- Parameter Screening methods
- Response surface analysis and Visualization (static/adaptive)
- Uncertainty and global sensitivity analysis
- Sample data Management/Visualization
- Job setup/control
UQ Software Development Challenges

- How to present the many complex UQ tools to the users who have different levels of UQ expertise?
  - Q: What is the maximum level of UQ expertise that can be embedded in a software framework to enable engineers with minimal amount of statistics, probability theory knowledge employ UQ methods without being dangerous?

- Current solution
  - Command line interpreter (expert mode): most flexible, for expert users
  - Command line interpreter (wizard mode): limited in scope, for beginners
  - Graphical user interface: more limited

- Future?
**PSUADE’s Execution Model (Batch mode)**

---

**PSUADE**

**INPUT**
- dimension = 3
- variable 1 X1 = -3.1416  3.1416
- variable 2 X2 = -3.1416  3.1416
- variable 3 X3 = -3.1416  3.1416

**OUTPUT**
- dimension = 1
- variable 1 Y

**METHOD**
- sampling = LH
- num_samples = 1000

**APPLICATION**
- driver = ./simulator

**ANALYSIS**
- analyzer method = Moment
- printlevel 2

---

**Matlab/Scilab**

- .m/.sci files

---

**Sample Results**

---

**Problem Specification**
- create work directory
- read in parameter values
- substitute into input decks
- launch job runs
- Post-process output data
- write outputs to file

---

**Job execution**
- sequential
- asynchronous
- MPI (single node jobs)
Welcome to PSUADE (version 1.7.6)

psuade> load psd
load complete : nSamples = 1000
      nInputs = 3
      nOutputs = 2
psuade> rs_u
Enter output number (1 - 2) : 1
Sample size for generating distribution? 100000
Save the generated sample in a file? (y or n) n
Sample mean  = 3.502575e+00 (RS uncertainties included)
Sample std dev = 3.703165e+00 (RS uncertainties included)
Output distribution plots is in matlabrsua.m.

psuade> rssobol1
Enter output number (1 - 2) : 1
RS-based First Order Sobol' Indices
RSMSobol1: sample mean (std dev of mean) = 3.502e+00 (0.000e+00)
RSMSobol1: std dev (std dev of std dev) = 3.700e+00 (0.000e+00)
RSMSobol1: Normalized mean VCE for input 1 = 3.1463e-01
RSMSobol1: Normalized mean VCE for input 2 = 4.4724e-01
RSMSobol1: Normalized mean VCE for input 3 = 1.9060e-04

rssobol1 plot file = matlabrssobol1.m
psuade>
A Simple UQ may involve the following tasks

\[ \mu(Y) = \int_{\Omega} F(X)p(X)dX \]

Estimate prediction mean given data and model

- Parameter screening
- Response Surface analysis
- Inference/calibration
- Model prediction
- Sensitivity analysis

\[ \mu(Y) \approx \int_{\Omega} F(\tilde{X})p(\tilde{X})d\tilde{X} \]

Dimension reduction

\[ \mu(Y) \approx \int_{\Omega} \tilde{F}(\tilde{X})p(\tilde{X})d\tilde{X} \]

Validated surrogates

\[ \tilde{p}(\tilde{X}|D) \propto p(\tilde{X})L(D|\tilde{X}, \tilde{F}(\tilde{X})) \]

D – data

\[ \mu(Y) \approx \int_{\Omega} \tilde{F}(\tilde{X})\tilde{p}(\tilde{X}|D)d\tilde{X} \]

Many methods for integration

\[ \eta_i^2 = \int \left[ \int \cdots \int \tilde{F}(\tilde{X})\tilde{p}(\tilde{X}_{\sim i}|\tilde{X}_{i}, D)d\tilde{X}_{\sim i} - \mu \right]^2 \tilde{p}(\tilde{X}_i)d\tilde{X}_i / \sigma^2 \]
One PSUADEN design goal is to enable multi-stage UQ

Prior input distributions

P(X1) → Unit1 → P1 = P(X1|P(X1))
P(X2) → Unit2 → P2 = P(X2|P(X2))
P(X3)

P(X4) → Unit4 → P4 = P(X4|P(X4))
P(X5) → Unit5 → P5 = P(X5|P(X5))

Mini-UQ analysis

Sub1A

P(X1,X2,X3|P1,P2,P(X3))

Sub2B

P(X1,X2,X3|P1,P2,P(X3))

Sub3

P(X4,X5|P4,P5)

Full System Model
Parameter Screening Methodology

Let $Y \in \mathbb{R}^m$ be a design and evaluate $S = \{(X^i, Y^i), i=1,...,N\}$ such that $I(X, Y) \approx I(X_G, Y)$ where $X_G$ is a random information that brings about $Y$.

**Sample designs**
- Monte Carlo (quasi-MC)
- Latin hypercube
- Morris
- other space-filling designs

**Other tools**
- + tools to visualize the analysis results
- + tools to manipulate the data

**Methods**
- Average gradients
- Noise estimation
- Number of tree bisections
- + tools to visualize the analysis results
- + tools to manipulate the data

**Sample designs**
- Monte Carlo (quasi-MC)
- Latin hypercube
- Morris
- other space-filling designs

**Other tools**
- + tools to visualize the analysis results
- + tools to manipulate the data

**Design**
- PSUADE
  - INPUT
    - dimension = 20
    - variable 1 $X_1 = 0 \ 1$
    - variable 20 $X_{20} = 0 \ 1$
  - END
  - OUTPUT
    - dimension = 1
    - variable 1 $Y$
  - END
  - METHOD
    - sampling = MOAT
    - num_samples = 210
  - END
  - APPLICATION
    - driver = ./simulator
  - END
  - ANALYSIS
    - analyzer method = MOAT
    - printlevel 2
  - END
  - END

**Syntax**

```python
PSUADE
INPUT
dimension = 20
variable 1 $X_1 = 0 \ 1$
......
variable 20 $X_{20} = 0 \ 1$
END
OUTPUT
dimension = 1
variable 1 $Y$
END
METHOD
sampling = MOAT
num_samples = 210
END
APPLICATION
driver = ./simulator
END
ANALYSIS
analyzer method = MOAT
printlevel 2
END
END
```
Parameter Screening – graphical analysis

Modified Means

Scatter Plots of Gradients (log to hit P.g.h.n.e.)
Response Surface Methodology

- Sampling methods
  - Classical and space-filling designs
  - Special design (e.g. collocation, splines)
  - Adaptive sampling

- Curve fitting methods
  - Polynomial regression, sparse grid
  - Splines/multivariate adaptive splines
  - Gaussian process, Kriging, radial basis
  - Tree-based, nearest-neighbor, PLS.

- Response surface validation
  - Training set errors
  - Hold-out, adjusted R-squared
  - K-fold cross validation (parallel)
  - Extrapolation test

- Response surface visualization
  - 1D, 2D, 3D, …. (matlab/scilab)

- NEW: stand-alone surrogate (in C, Python)
Adaptive Response Surface Methodology

- Use spatial refinement technique for moderate dimensions (up to 20)
- Initial sample size = 20, 13 refinements, 10 points per refinements = 150 points
- Use bootstrapped spline variances as error indicators
- Available as a command the command line interpreter (load and a_refine)
Uncertainty and Sensitivity Analysis

- **Uncertainty analysis**
  - On raw data or response surfaces
  - **Mixed** epistemic/aleatory
  - Second order (uncertainties on priors)

- **Sensitivity analysis**
  - Local sensitivity analysis (linearity)
  - **Pearson** Correlation Coefficient
  - Spearman Coefficient (monotonicity)
  - Variance-based methods
    - First order, **Second order**, Total order
    - Raw sample data or response surfaces
    - Account for RS errors
    - Support correlated inputs & different PDFs
    - Multiple methods derived from mathematical eqns, e.g. 1st order
      - Based on numerical integration
      - Based on replicated LH (orthogonal)
Bayesian Inference Methodology

\[ Y^e(d) = Y^s(d, X) + \delta(d) + \epsilon \]

Experiment

Simulation

Model form uncertainty

Observation noise

Design parameters e.g. geometry

Uncertain parameters

\[ \pi(X|D) \propto p(X)L(D|X) \]

Prior

Likelihood

Posterior

Joint 2-input posterior

Measurement data D with Uncertainties (and surrogate model)

Best negative log likelihood gives information about systematic error

p(X)
Summary of Bayesian Inference Methodology

- Option to use response surface or actual simulation model
- Support various different types of input prior distributions
- Provide standard or user-defined likelihood function
  - Option to use response surface errors in the likelihood function
- Use parallel multiple chain MCMC to speed up inference
  - Option to use brute force method for low input dimension
- Option to use discrepancy model
  - Minimum negative log-likelihood as indicator
- A posterior sample will be created at the end
  - For feeding into the next stage of a hierarchical UQ analysis
  - For verifying convergence
- Prior and posterior distribution plots
Numerical Optimization

- Several derivative-free optimizers
- Linear and nonlinear constraints
- Derivative-based optimizer
- Multi-fidelity (or multi-level) optimizers
  - Level 2 simulator may be a low fidelity model or a RS
- Optimization under uncertainty
  - are design parameters
  - are discrete and continuous uncertain parameters
  - is some probabilistic metrics, e.g.
    - Linear combinations of mean and standard dev.
    - Value-at-risk (e.g. 5% probability below this function value)
- Response surface option to give more accurate estimations

```
PSUADE
INPUT
  dimension = 5
  variable 1 X1 = -2 2
  variable 2 X2 = -2 2
  variable 3 X3 = -2 2
  variable 4 X4 = -2 2
  variable 5 X5 = -2 2
END
OUTPUT
  dimension = 1
  variable 1 Y
END
METHOD
  sampling = MC
  num_samples = 10
END
APPLICATION
  driver = ./simulator
  opt_driver = ./simulator
END
ANALYSIS
  optimization method = bobyqa
  optimization max_feval = 10000
  optimization tolerance = 1.000000e-06
  optimization print_level = 1
END
END
```
A Graphical User Interface for UQ

Model Error Histogram
- Error mean: -0.000044
- Error std dev: 0.001447

Actual vs. Predicted Data
- Estimate = Actual
- Estimate +/- 1 std dev
- Actual +/- 0.3%

Probability Distribution for Y1
- Statistics for mean PDF:
  - Sample mean: 4.5390e+01
  - Sample std dev: 4.2762e+01
  - Sample skewness: 1.1721e+00
  - Sample kurtosis: 3.3548e+00

Cumulative Distribution for Y1

Surface/Contour Plots of "Y1 = RBF(X1, X2)"

Analysis
- Model: Expert (Click for Wizard Mode)
- Select Output under Analysis: Y1
- Qualitative Parameter Selection:
  - Choose Parameter Selection Method: MARS Ranking
  - Compute input importance
- Ensemble Data Analysis:
  - Choose UQ Analysis: Uncertainty Analysis
  - Visualize Data: None selected
- Response Surface (RS) Based Analysis:
  - Select RS: Radial Basis Function
  - Legendre Polynomial Order: 1
  - MARS Number of basis functions: 100
  - MARS Degree of Interaction: 2
  - Use best set for output Y1
  - Number of Cross-Validation Groups: 10
  - Visualize RS: X1, X2
  - Upper Threshold: Lower Threshold:
  - Choose UQ Analysis: Aleatory Only
- Bayesian Inference

Results

Save RS interpolation code to file...
PSUADE has other visualization/diagnostics tools

- 2D sample input scatter plot
- 3D sample scatter plot
- 2D response surface plot

- **Sample manipulation**
  - Split/merge samples
  - Add/delete/modify inputs/outputs
  - Split/merge samples
Summary

- PSUADE provides a number of useful UQ methodologies
  - Dimension reduction (for various types of models)
  - Response surface (with uniform and adaptive refinements)
    ✓ produce stand-alone response surface interpolation codes
  - Many uncertainty/sensitivity analysis tools
  - Parameter inferences (with model form corrections)
  - Model validation (hypothesis testing)
  - Numerical optimization and optimization under uncertainty

- PSUADE provides rich Matlab/Scilab graphics outputs and many data (sample) manipulation tools

- PSUADE provides a simple job control management tool
  - Sequential, asynchronous (fork/join), MPI (each job sequential), or
  - It can be hooked up with user-developed job submission system

- It is an open source code (current version: 1.7.6a)
  - http://computation.llnl.gov/casc/uncertainty_quantification
Thank You