SANDIA REPORT

SAND2013-9165
Unlimited Release
Printed Oct, 2013

UQTk Version 2.0 User Manual

Bert Debusschere, Khachik Sargsyan, Cosmin Safta

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-9165
Unlimited Release
Printed Oct, 2013

UQTk Version 2.0 User Manual

Bert Debusschere, Khachik Sargsyan, Cosmin Safta

Abstract

The UQ Toolkit (UQTk) is a collection of libraries and tools for the quantification of uncertainty in numerical model predictions. Version 2.0 offers intrusive and non-intrusive methods for propagating input uncertainties through computational models, tools for sensitivity analysis, methods for sparse surrogate construction, and Bayesian inference tools for inferring parameters from experimental data. This manual discusses the download and installation process for UQTk, provides pointers to the UQ methods used in the toolkit, and describes some of the examples provided with the toolkit.

Contents

Chapter 1

Overview

The UQ Toolkit (UQTk) is a collection of libraries and tools for the quantification of uncertainty in numerical model predictions. In general, uncertainty quantification (UQ) pertains to all aspects that affect the predictive fidelity of a numerical simulation, from the uncertainty in the experimental data that was used to inform the parameters of a chosen model, and the propagation of uncertain parameters and boundary conditions through that model, to the choice of the model itself.

In particular, UQTk provides implementations of many probabilistic approaches for UQ in this general context. Version 2.0 offers intrusive and non-intrusive methods for propagating input uncertainties through computational models, tools for sensitivity analysis, methods for sparse surrogate construction, and Bayesian inference tools for inferring parameters from experimental data.

The main objective of UQTk is to make these methods available to the broader scientific community for the purposes of algorithmic development in UQ or educational use. The most direct way to use the libraries is to link to them directly from C++ programs. Alternatively, in the examples section, many scripts for common UQ operations are provided, which can be modified to fit the users' purposes using existing numerical simulation codes as a black-box.

The next chapter in this manual discusses the download and installation process for UQTk, followed by some pointers to the UQ methods used in the toolkit, and a description of some of the examples provided with the toolkit.

Chapter 2

Download and Installation

Requirements

The core UQTk libraries are written in C++, with some dependencies on FORTRAN numerical libraries. As such, to use UQTk, a compatible C++ and FORTRAN compiler will be needed. UQTk is installed and built most naturally on a Unix-like platform, and has been tested on Mac OS X and Linux. Installation and use on Windows machines under Cygwin is possible, but has not been tested.

Many of the examples rely on Python and matplotlib for postprocessing and graphing. As such, Python version 2.7.x with compatible NumPy, SciPy, and matplotlib are recommended. Further the use of XML for input files requires the Expat XML parser library to be installed on your system. Note, if you will be linking the core UQTk libraries directly to your own codes, and do not plan on using the UQTk examples, then those additional dependencies are not required.

Download

The most recent version of UQTk, currently 2.0, can be downloaded from the following location:
`http://www.sandia.gov/UQToolkit`

After download, extract the tar file into the directory where you want to install UQTk.

```
% tar -xzvf uqtk_src_v2.0_Oct-14-2013.tgz
```

Make sure to replace the name of the tar file in this command with the name of the most recent tar file you just downloaded.

Directory Structure

After extraction, there will be a new directory `UQTk_v2.0` (version number may be different). Inside this top level directory are the following directories:

`config`	Configuration files
`doc_cpp`	Documentation for C++ libraries
`src_cpp`	C++ source code
`examples_cpp`	Examples with C++ libraries
`src_matlab`	Matlab toolbox
`examples_matlab`	Matlab examples

Compilation

Before compiling, some configuration settings need to specified, such as the location of your compilers. To do so, change your directory to the configuration directory (again change the version number in the directory name below to the current version):

```
% cd UQTk_v2.0/config
```

In this directory, create a file named `config.site`, based on one of the templates provided. *E.g.*, if you are working on a Mac or Linux machine with the GNU compilers installed, the file `config.gnu` may be a good place to start from. Copy the file over to config.site, and edit the paths to point to the compiler locations on your system.

Additionally, to use some of the scripts in the examples provided with the distribution, the path to the upper level directory of UQTk needs to be set in the environment variable `UQTK_SRC`. If you are using the tcsh shell *e.g.*:

```
% setenv UQTK_SRC ~/software/UQTk_v2.0
```

or the bash shell:

```
% UQTK_SRC=~/software/UQTk_v2.0
% export UQTK_SRC
```

Test the value with `echo $UQTK_SRC`.

When this is done, go back up to the main level directory and compile with the `make` command.

```
% cd ..
% make
```

If all goes well, there should be no errors. After compilation ends, there will be three new directories in the `src_cpp` directory:

`src_cpp/lib`	Compiled library files
`src_cpp/include`	Include files for all libraries
`src_cpp/bin`	Binary executables for the apps that come with UQTk

To use the UQTk libraries, your program should link in the libraries in `src_cpp/lib` and add the `src_cpp/include` directory to the compiler include path. The apps are standalone programs that perform UQ operations, such as response surface construction, or sampling from random variables. For more details, see the Examples section.

Chapter 3

Source Code Description

UQTk implements many probabilistic methods found in the literature. For more details on the methods, please refer to the following papers and books on Polynomial Chaos methods for uncertainty propagation [2, 3], Bayesian inference [4], Bayesian compressive sensing [1], and the Rosenblatt transformation [5].

For more details on the actual source code in UQTk, HTML documentation is also available in the `doc_cpp/html` folder.

Chapter 4

Examples

The primary intended use for UQTk is as a library that provides UQ functionality to numerical simulations. To aid the development of UQ-enabled simulation codes, some examples of programs that perform common UQ operations with UQTk are provided with the distribution. These examples can serve as a template to be modified for the user's purposes. In some cases, *e.g.* in sampling-based approaches where the simulation code is used as a black-box entity, the examples may provide enough functionality to be used directly, with only minor adjustments. Below is a brief description of the main examples that are currently in the UQTk distribution. For all of these, make sure the environment variable `UQTK_SRC` is set and points to the UQTk upper level directory (*i.e.* the one that has `src_cpp` and `examples_cpp` as subdirectories), as described in the compilation section.

Elementary Operations

- Located in `examples_cpp/ops`
- Illustrates the use of UQTk for elementary operations on random variables that are represented with Polynomial Chaos (PC) expansions.
- To run an example, type `make examples` in `examples_cpp/ops` or run `./prob1.py`
- For more documentation, see `examples_cpp/ops/prob1.pdf`

Forward Propagation of Uncertainty

- Located in `examples_cpp/surf_rxn`
- Several examples of propagating uncertainty in input parameters through a model for surface reactions, consisting of three Ordinary Differential Equations (ODEs). Two approaches are illustrated:
  - Direct linking to the C++ UQTk libraries from a C++ simulation code:

    * Propagation of input uncertainties with Intrusive Spectral Projection (ISP), Non Intrusive Spectral Projection (NISP) via quadrature , and NISP via Monte Carlo (MC) sampling.
    * For more documentation, see `examples_cpp/surf_rxn/prob2.pdf`
    * An example can be run with `./prob2.py`
  - Using simulation code as a black box forward model:
    * Propagation of uncertainty in one input parameter with NISP quadrature approach.
    * For more documentation, see `examples_cpp/surf_rxn/prob3.pdf`
    * An example can be run with `./prob3.py`

- Both examples can be run with `make examples` in `examples_cpp/surf_rxn`

Bayesian Inference of a Line

- Located in `examples_cpp/line_infer`

- Infers the slope and intercept of a line from noisy data using Bayes' rule. The C++ libraries are called directly from the driver program. By changing the likelihood function and the input data, this program can be tailored to other inference problems.

- To run an example, type `make examples` in `examples_cpp/line_infer` or run `./prob5.py` directly.

- More documentation in `examples_cpp/line_infer/prob5.pdf`

Surrogate Construction and Sensitivity Analysis

- Located in `examples_cpp/uq_surr`

- A collection of scripts that construct a PC surrogate for a computational model that is specified as a black box simulation code. Also provides tools for sensitivity analysis of the outputs of this black box model with respect to its input parameters.

- For more information, see `examples_cpp/uq_surr/uq_surr.pdf`

- A full example can be run with `./example.x` in `examples_cpp/uq_surr`

Karhunen-Loève Expansion of a Stochastic Process

- Located in `examples_cpp/kl_sample`
- Some examples of the construction of 1D and 2D Karhunen-Loève (KL) expansions of a Gaussian stochastic process, based on sample realizations of this stochastic process.
- For more information and examples, see `examples_cpp/kl_sample/kl_example.pdf`

Chapter 5

Support

UQTk is the subject of continual development and improvement. If you have questions about or suggestions for UQTk, feel free to e-mail Bert Debusschere, at `mailto:bjdebus@sandia.gov`.

References

[1] S. Babacan, R. Molina, and A. Katsaggelos. Bayesian compressive sensing using Laplace priors. *IEEE Transactions on Image Processing*, 19(1):53–63, 2010.

[2] B.J. Debusschere, H.N. Najm, P.P. Pébay, O.M. Knio, R.G. Ghanem, and O.P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comp.*, 26(2):698–719, 2004.

[3] O P Le Maître and Omar M Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics (Scientific Computation)*. Springer, 1st edition. edition, April 2010.

[4] Y. M. Marzouk and H. N. Najm. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862–1902, 2009.

[5] M. Rosenblatt. Remarks on a multivariate transformation. *Annals of Mathematical Statistics*, 23(3):470 – 472, 1952.

DISTRIBUTION:

1	MS 0899	Technical Library, 8944 (electronic copy)

Sandia National Laboratories