Technische Universität München
Lehrstuhl für Informatik V
Dr. Tobias Neckel
M. Sc. Friedrich Menhorn

SS 2018

July 11, 2018

Algorithms for Uncertainty Quantification

Tutorial 11: Software for Uncertainty Quantification

In this worksheet, we focus on UQ case studies and UQ software that could be used to approach those case studies.

Software for UQ

In the lecture we saw a list comprising libraries/frameworks/toolkits for UQ; the ones discussed in the lecture are marked in bold.

- **chaospy**
- **Dakota**
- **MUQ** library
- Mystic
- NASA UQTools
- OpenCossan (matlab)
- **Openturns**
- **Π4U**
- Promethee
- Psuade
- **Queso**
- **SG++**
- SmartUQ
- **TASMANIAN**
- UQLab (matlab)
- UQ Toolkit (**UQTk**)
- **Uranie**
- ...

In the following, we discuss a series of five case studies. Discuss what methodology and which library/framework/toolkit from the above list you would use for the propagation of uncertainty. Why?

Case study one

You are given a forward model, for which a single run takes about 10 minutes. You know what phenomena the given code models, but you can use it only as a legacy code, i.e. a black box. Moreover, you know that 8 input parameters are uncertain.
Solution
Given the "intermediate" stochastic dimensionality (8 uncertain inputs $\Rightarrow$ stochastic dimensionality = 8) and the relatively low runtime/simulation, a sparse grid-based approach, such as polynomial chaos expansion + sparse grid quadrature or sparse grid interpolation, would be fitted for this kind of problem. Software-wise, the required functionality is available in `chaospy`, `SG++`, or `TASMANIAN`.

Case study two

You are given a forward model, for which a single run takes about 30 seconds. You know what phenomena the given code models, but you can use it only as a legacy code. Moreover, you know that 20 of it input parameters are uncertain.
Solution
Given the relatively high stochastic dimensionality (20 uncertain inputs $\Rightarrow$ stochastic dimensionality = 20) and the low runtime/simulation, a sampling-based method (standard Monte Carlo or variants) would be fitted for this kind of problem. Software-wise, this is available e.g. in `chaospy` or `UQtk`.

Case study three

You are given a forward model, for which a single run takes about 2 hours. You know that the uncertainty resides in five input parameters.
Solution
Given the high runtime/simulation, the given stochastic dimension (five) is no longer "low". For example, a full grid approach with $N = 5$ points in 1D would require $N^5 = 3125$ points in 5D, therefore 3125 model runs. Hence, a sparse grid-based approach(standard or, if possible/available, adaptive), methods such as polynomial chaos expansion + sparse grid quadrature or sparse grid interpolation would be fitted for this kind of problem. Furthermore, we also need parallel computing. Therefore, software-wise, the required functionality could be achieved via `DAKOTA` or `QUESO`.

Case study four

You are given a forward model, for which a single run takes about 3 minutes; assume that you have access to the underlying model. You know that the uncertainty resides in a quantity that varies continuously, at every point in the entire domain of the problem.

Solution
Given that the uncertain quantity is a random field, we first need to resort to some kind of discretization technique (in the lecture, we saw the KL expansion). The discretization can involve a large number of terms, therefore a high stochastic dimensionality (for example, having a KL expansion with 20 terms $\Rightarrow$ stochastic dimensionality = 20). Thus, the choice of the UQ approach depends on the size of this discretization. If the number of terms is large, a sampling based approach might be the only one suitable (standard Monte Carlo or variants). If the dimensionality is not too large (e.g. ≤ 20, although this depends on several factors), sparse-grid based approaches might be more appropriate (given the runtime/simulation, in this situation, a sparse grid approach might be feasible). For the KL expansion approximation, `UQtk` or `MUQ` might be an option. Moreover, if sparse grids are used, an additional library, such as `chaospy`, `SG++`, or `TASMANIAN`, could be used, too.

Case study five

You are given a mathematical model and a corresponding numerical implementation. One run of the given code takes about five minutes. Furthermore, you know that two input parameters are uncertain. You are required to compute statistics of the output of interest with high accuracy.

Solution
Given the requirement to obtain accurate results and the availability of the underlying model and its numerical implementation, one option would be to employ stochastic Galerkin. Since this approach has little library-based support (some support in e.g. `UQtk`), an own implementation might be needed. However, a library suitable for fast prototyping such as `chaospy` could be used to smoothen things out. Another option would be to use polynomial chaos expansion + the pseudo-spectral approach, since the runtime/simulation is small; although this approach is theoretically less accurate than the Galerkin projection, it is non-intrusive and embarrassingly parallel.