A component framework for stochastic FSI problems

[*] **Joachim Rang,** [†]**Hermann G. Matthies**

[*] Technische Universität Braunschweig, Institute for Scientific Computing
Hans-Sommer-Strasse 65, 38106 Braunschweig, Germany
e-mail: j.rang@tu-braunschweig.de, web page: https://www.tu-braunschweig.de/wire/

[†] Technische Universität Braunschweig, Institute for Scientific Computing
Hans-Sommer-Strasse 65, 38106 Braunschweig, Germany
e-mail: wire@tu-braunschweig.de, web page: https://www.tu-braunschweig.de/wire/

ABSTRACT

Coupled problems appear in many areas of engineering. One such example are FSI problems, the coupling of fluid and structure. Often partitioned methods are used for the solution, i.e. each subproblem is solved by dedicated solver, and hence these solvers have to be coupled.

Here we consider FSI problems with uncertainties in the fluid and the structural part. A conceptionally straightforward scheme to solve such problems is the Monte Carlo method, where the coupled deterministic problems are solved many times with different realisations. But the convergence of the Monte Carlo method is rather slow, needing many solutions of the deterministic problems.

Therefore we consider stochastic methods with a functional approximation, sometimes also called spectral methods. In this case we need a different discretisation than sampling for the stochastic part, namely stochastic projection or stochastic Galerkin methods. Moreover the stochastic methods are implemented in a non-intrusive manner, i.e. the deterministic solvers are used as a block-box similarly as in the partitioned approach of deterministic FSI problems.

The stochastic FSI problem is thus solved in a similar way as the deterministic one by using and coupling the dedicated domain solvers. . The discretisation of the stochastical part can be

interpreted as a stochastical partitioning of the problem. For the communication and coupling of the stochastic and the deterministic software components we use the Component Template Library (CTL). The software architecture is explained in detail, as well as the mathematical background leading to a convergent solution in the framework of low-rank tensor approximations (e.g. Proper Generalized Decomposition). The approach is demonstrated on an example.