Embedded Stochastic Galerkin Methods via Template-based Generic Programming

Mathematical and Computational Tools for Predictive Simulation of Complex Coupled Systems Under Uncertainty

Eric Phipps, Roger Pawlowski, Andy Salinger
Sandia National Laboratories, Albuquerque, New Mexico

Abstract

Predictive computational simulation promises to help solve many important problems throughout science and engineering. A critical task for predictive simulation is the quantification of uncertainties in simulation results arising from error or uncertainty in the input data. Thus there has been significant interest in developing computational methodologies for accurately and efficiently propagating representations of uncertainty in simulation input data to simulation output quantities of interest. While there are many approaches for representing uncertainty in simulations, in this work we are concerned with probabilistic representations of uncertain input data through random variables with prescribed probability distributions. Within this context, approaches fall into two categories: non-intrusive methods that sample the simulation code at various realizations of the input data, and intrusive or embedded methods that require more information from the simulation beyond quantity of interest realizations. In the latter case, a well-known class of such methods are intrusive stochastic Galerkin methods, which formulate new sets of equations governing the projection of simulation variables in spaces spanned by polynomials of the input random variables.

The study and use of stochastic Galerkin methods in large-scale computational environments is hampered by their inherent intrusiveness, a given simulation code must be extensively reengineered to formulate and solve the stochastic Galerkin equations. In this work, we present a general approach based on template-based generic programming for automating much of the process of incorporating stochastic Galerkin methods in large-scale simulation environments. Template-based generic programming is a generalization of the ideas underlying automatic differentiation whereby computer code for a given calculation is transformed into code to compute additional derived information (derivatives in the case of automatic differentiation and projection coefficients in the stochastic Galerkin case). Our approach leverages the template features of the C++ language by replacing the fundamental floating point scalar type with a template parameter and replacing this template parameter with carefully constructed classes that overload all of the floating point operations within the calculation. By combining this technique with traditional automatic differentiation, derivatives of stochastic Galerkin coefficients can be obtained for nonlinear solvers, implicit time-stepping, stability analysis and optimization.

In this poster we describe a set of software tools called Stokhos, which is a package within the Trilinos solver framework. In conjunction with Sacado, a Trilinos package for automatic differentiation of C++ codes, Stokhos provides a toolkit for incorporating stochastic Galerkin methods in large-scale computational environments. Leveraging the template-based generic programming approach, we contrast several approaches for propagating stochastic Galerkin coefficients in the manner described above. Additionally, we describe several tools in Stokhos for assembling the stochastic Galerkin equations in large-scale distributed memory environments, and interfaces that connect these equations to the numerous linear, nonlinear, transient, and optimization solvers within Trilinos. Finally we demonstrate how these techniques have been incorporated into several PDE simulation tools at Sandia, enabling ground-breaking research into large-scale embedded uncertainty quantification methods.