

UNCERTAINTY QUANTIFICATION IN COMPUTATIONAL PREDICTIVE MODELS FOR FLUID DYNAMICS USING A WORKFLOW MANAGEMENT ENGINE

**Gabriel Guerra,¹ Fernando A. Rochinha,¹ Renato Elias,²
Daniel de Oliveira,³ Eduardo Ogasawara,³ Jonas Furtado Dias,³
Marta Mattoso,³ & Alvaro L. G. A. Coutinho^{2,*}**

¹Mechanical Engineering Department, Federal University of Rio de Janeiro, Brazil

²High Performance Computing Center, Federal University of Rio de Janeiro, Brazil

³Systems Engineering and Computer Science Department, Federal University of Rio de Janeiro, Brazil

Original Manuscript Submitted: 06/10/2011; Final Draft Received: 10/11/2011

Computational simulation of complex engineered systems requires intensive computation and a significant amount of data management. Today, this management is often carried out on a case-by-case basis and requires great effort to track it. This is due to the complexity of controlling a large amount of data flowing along a chain of simulations. Moreover, many times there is a need to explore parameter variability for the same set of data. On a case-by-case basis, there is no register of data involved in the simulation, making this process prone to errors. In addition, if the user wants to analyze the behavior of a simulation sample, then he/she must wait until the end of the whole simulation. In this context, techniques and methodologies of scientific workflows can improve the management of simulations. Parameter variability can be put in the general context of uncertainty quantification (UQ), which provides a rational perspective for analysts and decision makers. The objective of this work is to use scientific workflows to provide a systematic approach in: (i) modeling UQ numerical experiments as scientific workflows, (ii) offering query tools to evaluate UQ processes at runtime, (iii) managing the UQ analysis, and (iv) managing UQ in parallel executions. When using scientific workflow engines, one can collect data in a transparent manner, allowing execution steering, the postassessment of results, and providing the information for reexecuting the experiment, thereby ensuring reproducibility, an essential characteristic in a scientific or engineering computational experiment.

KEY WORDS: sparse grid stochastic collocation method, scientific workflows, provenance, computational fluid dynamics, parallelization, adaptive sparse grid

1. INTRODUCTION

The complexity involved in engineering systems has been frequently tackled with the use of sophisticated models on computational engineering experiments. That, from the decision maker's standpoint, requires the use of robust and reliable numerical simulators. Often, the reliability of those simulations is disrupted by the inexorable presence of uncertainty in the model data, such as inexact knowledge of system forcing, initial and boundary conditions, physical properties of the medium, as well as parameters in constitutive equations. This enforces the need for efficient uncertainty quantification (UQ) methods for the computation of confidence intervals in computed predictions, the

*Correspond to Alvaro L. G. A. Coutinho, E-mail: alvaro@nacad.ufrj.br, URL: <http://www.nacad.ufrj.br/>

assessment of the suitability of model formulations, and/or the support of decision making during the experiment analysis.

Computational UQ experiments traditionally rely on the Monte Carlo method [1]. This method requires the generation of an ensemble of random realizations associated to the uncertain data and then employs deterministic solvers repeatedly to obtain the ensemble of data that should be processed to estimate final results mean and standard deviation. The implementation of the Monte Carlo method is straightforward, but its convergence rate is very slow (proportional to the inverse of the square root of the number of realizations) and often infeasible due to the large amount of CPU time needed.

Another technique that has been applied recently is the so-called stochastic Galerkin (SG) method, which employs polynomial chaos expansions to represent the solution and inputs leading to stochastic differential equations [2]. The Galerkin projection minimizes the truncated expansion error. The main SG drawback relies on its need for solving a system of coupled equations, which requires efficient and robust solvers and, most importantly, the modification of an existing deterministic code. This last issue entails difficulties on using commercial or already-in-use codes. The nonintrusive stochastic collocation (SC) method [3], addresses this point. SC methods are built on the combination of interpolation methods and deterministic solvers. SC methods require the solution of deterministic problems corresponding to points of an abstract random space. Similarly to SG, SC methods achieve fast convergence when the solution possesses sufficient smoothness in random space [4].

In any case, running intensive computational UQ experiments often need special software frameworks. They are usually composed by several programs that implement UQ methods. These programs are computationally intensive and consume and produce a large amount of data. As the computational experiment complexity increases, running these simulations becomes a challenge. One available solution is provided by the design analysis kit for optimization and terascale applications (DAKOTA) project [5]. DAKOTA offers an extensible framework to integrate UQ into existing simulation codes. Specifically for UQ analysis, DAKOTA contains a number of important stochastic tools that can be linked to existing numerical codes. Here, we present a different solution for handling the challenges posed by UQ when applied to high demanding computational scientific and engineering experiments.

To help scientists and engineers in managing resources involved in computational experiments, scientific workflows are gaining much interest. A workflow may be defined as a model of a process, which consists in a series of activities and its dependencies [6]. A scientific workflow structures the processing of a simulation as a graph of activities, in which vertices correspond to data processing activities and edges represent the dataflow between them. Workflow activities are associated to scientific or engineering applications that prepare, process, and analyze data. Scientific workflow management systems (SWfMS) [7] are software systems that support the definition, execution and monitoring of scientific workflows. While DAKOTA is focused on supporting UQ, SWfMS are generic tools for scientific experiments, such as structural analysis, computer fluid dynamics (CFD), astronomy, bioinformatics, among others [7]. Therefore, by adopting this generic tool all scientific/engineering experiments can integrate its execution data (i.e., provenance data) and obtain a uniform register of all computational experiments.

Although a computational experiment may be modeled as a scientific workflow, controlling its large number of variations and executions is far from being trivial. It can be very complex to manage the execution of thousands of scientific activities and to control all data generated. A key concept to help this management is the provenance registration of data generated by the workflow. Provenance data give the history of the ownership or location of data and are seen as a way to help reproducing the experiment [8]. Provenance data provide information to answer complex queries regarding the experiment. These answers can be used to refine and enhance the obtained results. Another important aspect is that provenance databases accumulate experiment data along several years. It is a valuable knowledge that can serve for long-term analysis and assist decisions regarding new experiments. Provenance data can also be used to identify bottlenecks in the execution and to assess performance data [9]. This is particularly important because computational UQ experiments often demands high-performance computing (HPC) environments for producing results in a feasible time. This poses yet another challenge: managing a large amount of runs on a remote HPC machine and gathering provenance data from these runs.

One of the emerging workflow execution engines that supports parallel execution of scientific workflows in HPC environments is Chiron [10]. Chiron aims at reducing the complexity involved in designing and managing parallel executions within computational experiments. In this paper, we have used Chiron to model and evaluate two UQ

workflows. The first is a fluid-structure interaction workflow, related to the problem of vortex-induced vibrations in risers. We also employed Chiron to perform a sensitivity analysis workflow for computing model coefficients in large eddy simulation (LES) of turbulence. These workflows fit well as many-task-computing (MTC) applications [11]. Computational results show that a systematic approach for distributing parallel scientific and engineering applications is viable, saving time and reducing operational errors, with the additional benefits of querying provenance data during runtime.

The paper is organized as follows: Section 2 briefly presents the mathematical background for the UQ methods used in this paper. Section 3 presents Chiron and discusses how it can be used as an enabling technology to support UQ analyses. Section 4 presents results using the UQ workflows. Finally, Section 5 offers conclusions.

2. MATHEMATICAL BACKGROUND AND STOCHASTIC COLLOCATION

Often, predictive models are based on either partial or ordinary differential equations representing the physics of the systems to be simulated. Here, the uncertainty on those models is introduced through random variables (or fields) within a probabilistic perspective yielding a mathematical problem consisting of stochastic partial differential equations (SPDEs) to be solved. The formal framework is built with a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$, where Ω is the event space, $\mathcal{F} \subset 2^\Omega$ is the σ -algebra of subsets in Ω and $\mathcal{P} : \mathcal{F} \rightarrow [0, 1]$ is the probability measure.

Considering a general differential operator \mathcal{L} defined on a d -dimensional bounded spatial domain $\mathcal{D} \subset \mathbb{R}^d$, ($d = 1, 2, 3$) with boundary $\partial\mathcal{D}$ and a time interval $\mathcal{I} = [0, T]$, the mathematical model leads to the following set of SPDEs having as solution, for \mathcal{P} almost everywhere $\omega \in \Omega$, the stochastic field (or process) $\mathbf{u}(\mathbf{x}, t, \omega) : \mathcal{D} \times \mathcal{I} \times \Omega \rightarrow \mathbb{R}$:

$$\mathcal{L}(\mathbf{x}, t, \omega; \mathbf{u}) = f(\mathbf{x}, t, \omega) \quad \mathbf{x} \in \mathcal{D} \quad (1)$$

$$\mathcal{B}(\mathbf{x}, t, \omega; \mathbf{u}) = g(\mathbf{x}, t, \omega) \quad \mathbf{x} \in \partial\mathcal{D} \quad (2)$$

$$\mathbf{u}(\mathbf{x}, 0, \omega) = \mathbf{u}_0(\mathbf{x}, \omega) \quad \mathbf{x} \in \mathcal{D}, \quad (3)$$

where $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$, t is time coordinate, and $\mathbf{u}(\omega) = (u_1(\omega), \dots, u_i(\omega)) \in \mathbb{R}^i$, $i \geq 1$. Besides, \mathcal{B} represents a boundary operator, the fields f and g are source terms, and \mathbf{u}_0 is the initial condition, possible random to account uncertainties at the start of the process. The notation used in this section is similar to the one adopted in [12].

To deal with the probability space in a computational simulation, we need to set the basis for building discrete approximations. This can be accomplished by reducing the original infinite dimensional space to one characterized by a finite number of random variables $\mathbf{Y} = [Y_1(\omega), \dots, Y_N(\omega)]$. Often, this dimension reduction is attained by using a truncated spectral representation of the input data obtained through a Karhunen-Loève expansion [13].

Therefore, following [14], the solution of Eqs. (1)–(3) can be represented by the same set of random variables $\{Y_i(\omega)\}_{i=1}^N$, e.g.,

$$\mathbf{u}(\mathbf{x}, t, \omega) = \mathbf{u}[\mathbf{x}, t, Y_1(\omega), \dots, Y_N(\omega)]. \quad (4)$$

Now, assuming that $\{Y_i\}_{i=1}^N$ are independent random variables with probability density functions $\rho_i : \Gamma_i \rightarrow \mathbb{R}^+$, and their images $\Gamma_i \equiv Y_i(\Omega)$ bounded intervals in \mathbb{R} for $i = 1, \dots, N$, the joint probability density of \mathbf{Y} holds

$$\rho(\mathbf{Y}) = \prod_{i=1}^N \rho_i(Y_i) \quad \forall \mathbf{Y} \in \Gamma. \quad (5)$$

Thus, for a fixed \mathbf{Y} on the discrete random domain, the original infinite dimensional problem corresponds to a deterministic problem defined on $\mathcal{D} \times \mathcal{I}$ that can be solved by standard numerical techniques. Indeed, classical stochastic simulations methods, such as Monte Carlo, explores this view by sampling over the random domain, following the probability distribution function (PDF) of the input data and solving the resulting deterministic problems. The results are postprocessed in order to describe the sought solution statistics. Commonly, these statistics are restricted to low-order moments.

Stochastic collocation also relies on the same mathematical background, but unlike direct Monte Carlo sampling, the $\{Y_i\}_{i=1}^M$ points of the multidimensional stochastic space are chosen, such as to construct a polynomial interpolation

of the problem solution in the stochastic space $\Gamma \subset \mathbb{R}^N$, the support of the random variables coming from a tensor product among the supports of each $\{Y_i\}$. Because collocation requires only the solution of M of deterministic problems, its computational implementation consists only of a stochastic wrapper around a deterministic kernel. This nonintrusive implementation makes collocation quite attractive, as reliable existing deterministic codes can be used without almost any extra effort. Moreover, any tools already in place, such as the scientific workflows described hereafter, can be adapted in order to optimize the simulations from a computational standpoint.

Full-tensor products of the corresponding 1D interpolation rule does not represent a viable alternative as the stochastic dimension increases. The resulting number of support points would lead to prohibitively computational costs. In order to diminish this burden and make viable the stochastic simulations using collocation, sparse grids have been employed with success (see [12, 15, 16]). The Smolyak algorithm [17] leads to a very efficient manner to construct interpolation functions specially due to the reduced number of support nodes required.

Here, in order to make the present text self-contained, we briefly reproduce the presentation of the Smolyak algorithm in [12]. A 1-D function $f : \Gamma_j \rightarrow \mathbb{R}$, can be interpolated as follows:

$$\mathcal{U}^i(f) = \sum_{j=1}^{m_i} f(Y_j^i) \cdot a_j^i, \quad (6)$$

where $a_i(\mathbf{Y})$ are interpolation functions indexed by i , which denote the approximation level, and computed on the support nodes,

$$X^i = \{Y_j^i | Y_j^i \in \Gamma \text{ for } j = 1, \dots, m_i\}, \quad (7)$$

with m_i as the number of elements of the set X^i . The passage from 1D interpolation to the multivariate case is carried out with tensor products that lead to

$$(\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_N})(f) = \sum_{j_1=1}^{m_1} \dots \sum_{j_N=1}^{m_N} f(Y_{j_1}^{i_1} \dots Y_{j_N}^{i_N}) \cdot (a_{j_1}^{i_1} \otimes \dots \otimes a_{j_N}^{i_N}), \quad (8)$$

which provides a basis for the application of the Smolyak algorithm. The Smolyak algorithm results in the interpolant $\mathcal{A}_{q,N}(f)$ built on a sparse set of the original grid; that is,

$$\mathcal{A}_{q,N}(f) = \sum_{q-N+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-|\mathbf{i}|} \cdot (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_N}), \quad (9)$$

where $\mathbf{i} = (i_1, \dots, i_N) \in \mathbb{N}^N$, $q \geq N$, $\mathcal{A}_{N-1,N} = 0$, $|\mathbf{i}| = i_1 + \dots + i_N$, and i_k is the interpolation level along k th random dimension. Thus, considering $\Delta^i = \mathcal{U}^i - \mathcal{U}^{i-1}$, the resulting algorithm can be recast in a hierarchical way as follows:

$$\mathcal{A}_{q,N}(f) = \sum_{|\mathbf{i}| \leq q} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_N})(f) = \mathcal{A}_{q-1,N}(f) + \sum_{|\mathbf{i}|=q} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_N})(f). \quad (10)$$

Equation (10) suggests an implementation based on the use of the results obtained in previous approximation levels if additional precision is required. Besides, if convenient support nodes are chosen, this implementation can explore nested sets of points (e.g., $X^i \subset X^{i+1}$). Combining these two ingredients, the computation of the interpolant in a higher level only demands the computation of the function value on the nodes that are added to the grid when moving up in the hierarchical construction. Moreover, the surpluses for each point of the stochastic grid can be computed in parallel, which increases the computational performance considerably [12].

Therefore, the Smolyak algorithm becomes a rational way of building approximations of any stochastic $u \in \mathcal{D} \times \mathcal{I} \times \Omega$ as follows:

$$u(x, t, \mathbf{Y}) = \sum_{|\mathbf{z}| \leq q} \sum_{j \in B_i} w_j^i(x, t) \cdot a_j^i(\mathbf{Y}). \quad (11)$$

The above formal representation reveals the core of the stochastic collocation method used to solve Eqs. (1)–(3). As mentioned earlier, a deterministic problem is solved for each point of the sparse grid and the solution is then extended

to any other point of the random space. From this equation, it is also possible to extract the statistics, such as low-order moments, by either sampling or performing numerical quadratures in a multidimensional space. Besides, even the PDF of the random variable can be estimated from histograms produced with a large number of samples [12].

The choice of the support nodes and the interpolation functions lead to different approximation schemes; one is based on Clenshaw-Curtis points and Lagrangian interpolation and will be referred to as a conventional sparse grid collocation method (CSGC). The second method explores piecewise linear hat functions and equidistant Newton-Cotes support nodes, which are suitable for the adaptive sparse grid collocation method (ASGC) proposed in [12].

3. CHIRON: ENABLING TECHNOLOGY FOR UNCERTAINTY QUANTIFICATION

Realistic computational experiments often explore large numerical models to be simulated over large sets of parameters. Executing a sequence of programs exploring a varied set of parameter values is commonly known as a parameter sweep [18]. In particular, a number of stochastic simulation techniques, such as discussed here, can be implemented with the aid of a parameter sweep. Modeling the chaining of programs on a parameter sweep that involves a scientific or engineering simulation is a complex task and currently may be done with the help of a scientific workflow system. SWfMS is a software that supports scientific workflow modeling and execution, collecting provenance data during the design and execution of the workflow. These are tools that offer a graphical view of the problem, making it easier to control the versions of the experiment and to gather provenance data for later analysis. SWfMS also flexibilizes the computational experiment facilitating its maintenance [19]. However, large-scale experiments often require the use of a parallel computing environment. Few SWfMS are ready for parallel execution in current HPC environments. Thus, to enable the efficient use of such resources, we have used Chiron [10], a datacentric scientific workflow engine that executes, in parallel, scientific applications.

Chiron focuses on the parallel execution of the workflow. Figure 1 presents a simplified view of how Chiron works on an HPC cluster parallel environment where A, B, and C represent the activities of the workflow. Hence, the users design their workflows on their personal computers. Then, they put their workflow with the input parameters they want to be consumed by the workflow on the cluster submitting their job to a HPC scheduler, such as Condor [20] or PBS [21]. The scheduler calls Chiron to handle the execution of the workflow. Basically, Chiron coordinates the workflow parallel execution, assigning different input parameter sets to computing nodes. Chiron uses a data-centric approach using a scientific workflow algebra to handle the workflow parallel execution efficiently. The algebra

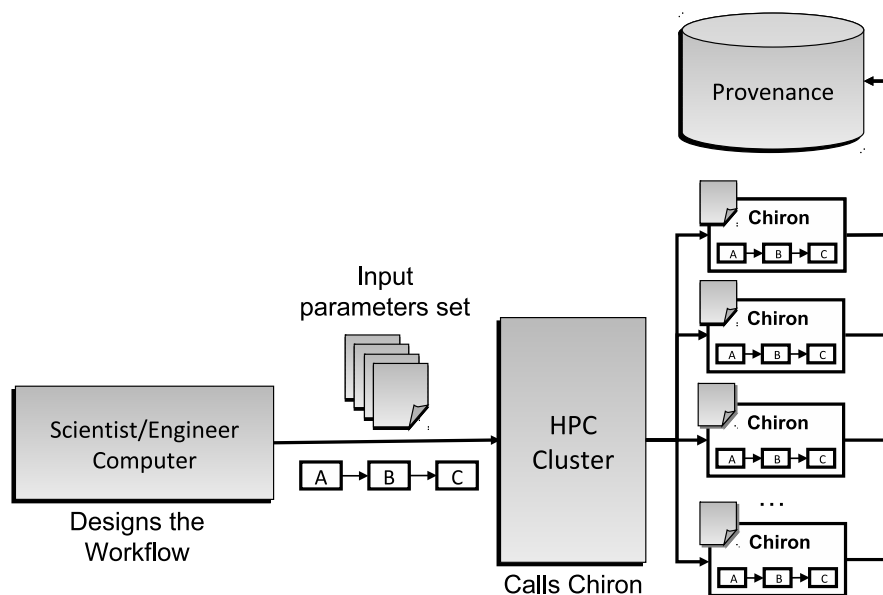


FIG. 1: Chiron execution architecture.

standardizes data consumption and production and also opens horizons for workflow optimizations [10]. Workflows in Chiron are defined through a declarative language (XML), which transforms into an algebraic expression that enables workflow automatic optimization. Chiron also establishes an optimized parallel execution plan for the workflow. The parallel execution of a workflow introduces several difficulties for collecting provenance data, because these data are also distributed across the HPC environment in different nodes in a cluster or even in different virtual machines in a cloud environment [22]. On the Chiron-generated database, only provenance data are stored, such as metadata and several extracted results. Intermediate application data, such as huge files and other complex computation results, are only referenced in the database. These files are kept in the application storage area. Provenance is essential for scientific and engineering experiments and ensures that the experiment can be reproduced over different conditions. Most parallel approaches for scientific workflow execution provide weak support for distributed provenance gathering [23]. Chiron provides native support for distributed provenance by storing provenance data during the execution of all activations. Thus, it is possible to monitor the status of the experiment and available results through runtime provenance queries. Users can also perform provenance queries to check whether execution failures have occurred. Monitoring some specific attributes, results or checking the elapsed time of a given task may indicate that a failure happened. Such information can be used to refine the task (to prevent it from failing again) and resubmit it. Depending on the gathered results, the scientist/engineer may decide to add extra sets of parameters on the parameter sweep or make other decisions regarding the experiment.

Chiron requires additional software such as PostGres [24], Java, and additional libraries like MPJ [25] and HSQLDB [26]. They are open source softwares that can be made available in HPC centers. The libraries are included on Chiron distribution code. Chiron is available on clusters of the High-Performance Computing Center at the Federal University of Rio de Janeiro for any project or scientist that desires to use it. There are other SWfMS available for use that are publicly distributable, such as VisTrails [27] or Swift [11]. However, differently from these systems, Chiron supports users in complex exploratory processes involving dynamic workflows by providing provenance queries services during runtime parallel executions.

4. RESULTS

In this section, we evaluate SC with Chiron for carrying out a UQ analysis in two different situations. Both situations present challenges concerning high-performance simulations and uncertainty propagation through predictive models. We intend to present viable and efficient ways for supporting UQ with a scientific workflow. The simulation codes are built in a nonintrusive way such that the collocation implementation is wrapped around deterministic kernels. In both examples, we explore features of the computational implementation and aspects of the uncertainty analysis.

Traditionally, CFD is one of the most demanding areas of scientific computing and always on the leading edge of the available supercomputer technology. Recently, there is growing interest on complex computational mechanics studies involving verification, validation and uncertainty quantification, which usually require a massive amount of simulations and generate huge data [28]. This scenario increases the complexity in managing such analysis. Scientific workflows are promising solutions to automatically support such exploratory demands, in addition to providing less error-prone environments. In particular, because the amount of generated data increases dramatically, provenance plays a key role in this scenario.

The first example deals with fluid-structure interaction (FSI) in the context of vortex-induced vibrations (VIV), which is central to the design of risers and floating structures in offshore engineering. Here, the VIV phenomenon is described by a simple model, often used by engineers in the initial design stages, that, despite its simplicity, is capable of tracking important aspects of the dynamics. A long-term response is a key ingredient for understanding the fatigue failure mechanisms of riser structures. Because the long-term statistics lead to a great amount of correlated data, these can be obtained and handled with the help of our enabling computational infrastructure.

The second example is devoted to a critical assessment of LES models. The inherent complexity of turbulent flows demands the use of refined grids in time and space for representing the multiscale character of the involved phenomena, especially the dissipation mechanisms. The use of direct numerical simulation (DNS) often leads to prohibitive computational costs that scales with the Reynolds number, in that case, inversely proportional to the smallest scale to be captured within the simulation. Despite some recent improvements on DNS schemes [29], in many practical

engineering applications, LES models, which rely on adding extra dissipation, are often used. Here, a sensitivity analysis with respect to a LES parameter, similar to the one proposed in [30], is pursued in a benchmark problem. In this example, emphasis is placed on UQ performed within a stabilized finite element HPC code coordinated by a scientific workflow.

4.1 Vortex-Induced Vibrations

Vortex-induced vibration is caused by the vortex shedding behind bluff bodies and may lead to degradation of structural performance or even structural failure. This is a particularly important issue for offshore structures, such as pipes, risers, and mooring lines. Several ways have been pursued to predict the dynamic response of structures undergoing large-amplitude vibrations induced by the surrounding flow. One of the most effective prediction methods consists of solving the coupled fluid-structure system. The flow is modeled by Navier-Stokes equations, and the structure, due to its rigidity, is often characterized by a simple oscillator with one or two degrees of freedom. However, the application of this approach to real problems would lead to prohibitive costs in the preliminary design phase. An alternative, considering the trade-off between costs and precision, is to use phenomenological models based on wake oscillators [31] that replace the vortex-shedding mechanisms of the flow by simple models. Here, we use the model proposed in [32] that captures important features of the VIV dynamics after a calibration of its parameters, taking into consideration the available experimental data. This calibration is thus a source of modeling uncertainties. Consider the elastically supported rigid circular cylinder shown schematically in Fig. 2. The coupling wake and structure oscillators are described by the following equations:

$$\ddot{y} + \left(2\zeta\delta + \frac{\gamma}{\mu}\right) \dot{y} + \delta^2 y = Mq \quad (12)$$

$$\ddot{q} + \varepsilon(q^2 - 1)\dot{q} + q = A\ddot{y}, \quad (13)$$

with

$$M = \frac{C_{L_0}}{4\pi^3 St^2 \mu}, \quad \delta = \frac{\Omega_s}{\Omega_f} = \frac{1}{St U_r}, \quad \gamma = \frac{C_D}{\pi^2 St}, \quad U_r = \frac{2\pi U}{\Omega_s D}, \quad St = \frac{\Omega_f D}{2\pi U}, \quad \mu = \frac{m}{\rho D^2 (\pi/4)},$$

where y is the dimensionless in-plane cross-flow displacement of the structure, q is the dimensionless wake variable associated with the fluctuating lift coefficient of the structure. A and ε are empirical parameters, ζ and δ are the structure-reduced damping and angular frequency, respectively, γ is a stall parameter, μ is the mass ratio, M is the mass parameter that simulates the effect of the wake on the structure, C_{L_0} is the reference lift coefficient of

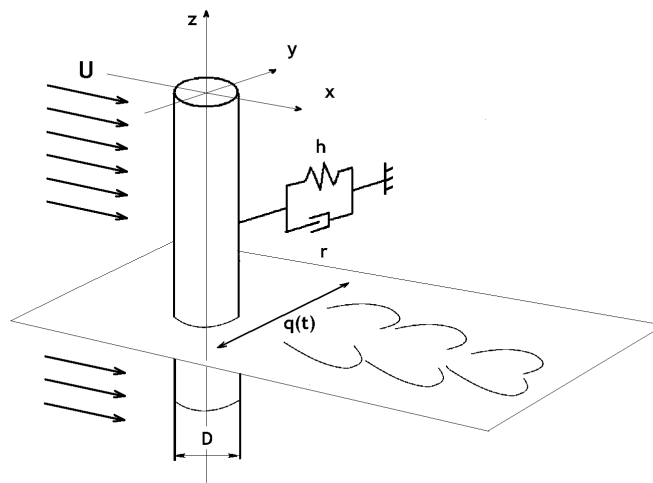


FIG. 2: Wake oscillator model for vortex-induced vibration.

the structure, D is the diameter of the cylinder, U is the free-stream velocity of the uniform flow, St is the Strouhal number, Ω_s and Ω_f are respectively the vortex shedding and structural angular frequencies, U_r is the reduced velocity, C_D is the drag coefficient of the structure, m is the structural mass, including the fluid-added mass, and ρ is the fluid density. The coupling empirical parameter is such that, $A = 10$, if $0 \leq \delta \leq 1$, or $A = 4$, if $\delta > 1$. The dependency of A on the frequency ratio δ , including a jump, might entail abrupt changes on the structure response. Whenever the shedding frequency approaches the natural frequency of the structure, the coupled system enters in the lock-in regime. The amplitude of the structural response during lock-in achieves a maximum and, thus, this is considered a critical mode of vibration. A schematic view of the lock-in phenomenon is depicted in Fig. 3, where the maximum amplitude in the steady-state regime is plotted as a function of the reduced velocity U_r in the range $[0, 16]$. In order to avoid the danger of failure, computational models are built aiming at identifying the narrowband of the reduced velocity corresponding to lock-in. Bearing that in mind, we intend to investigate the impact of uncertainties on the predictions drawn from those computations. Figure 3 compares the differences, which are more pronounced along the lock-in, among the system response obtained with the models in [32, 33] and experimental data. The models differ only in the way parameters are determined. As we can see in Fig. 3, the lock-in response shows a significant sensitivity with respect to those parameters, which motivates the following analysis, which is carried out using the model in [32], as its predictions are closer to the available experimental data.

Two parameters are considered uncertain, the coupling parameter A and the lift coefficient C_{L_0} and, thus, modeled as random variables. The former is parametrized as $A = \bar{A} c_f$, with \bar{A} the mean value and c_f a random variable to be described in the sequence. The mean \bar{A} assumes the deterministic values assigned before (e.g., $\bar{A} = 10$, if $0 \leq \delta \leq 1$, or $\bar{A} = 4$, if $\delta > 1$). The choice of those two parameters relies on a sensitivity analysis about the model parameters' influence on the system response, which is not reported here for the sake of brevity. Moreover, both parameters are to be determined through experiments and then subject to unavoidable variability. In order to illustrate the effect of uncertainties propagation on the computed results, we assume the following probabilistic models:

$$C_{L_0} = \bar{C}_{L_0} + \sigma_{C_{L_0}} \xi_1 \quad (14)$$

$$c_f = \bar{c}_f + \sigma_{c_f} \xi_2, \quad (15)$$

where mean and variance are given by $(\bar{C}_{L_0}, \sigma_{C_{L_0}}) = (0.3, 0.01)$ and $(\bar{c}_f, \sigma_{c_f}) = (1, 0.2)$. The random variables $\xi_i (i = 1, 2)$ with support $[-1, 1]$ are assumed independent and uniformly distributed. Global polynomial interpolations, such as those employed in CSGC, might not be able to handle the presence of discontinuities or sharp gradients in

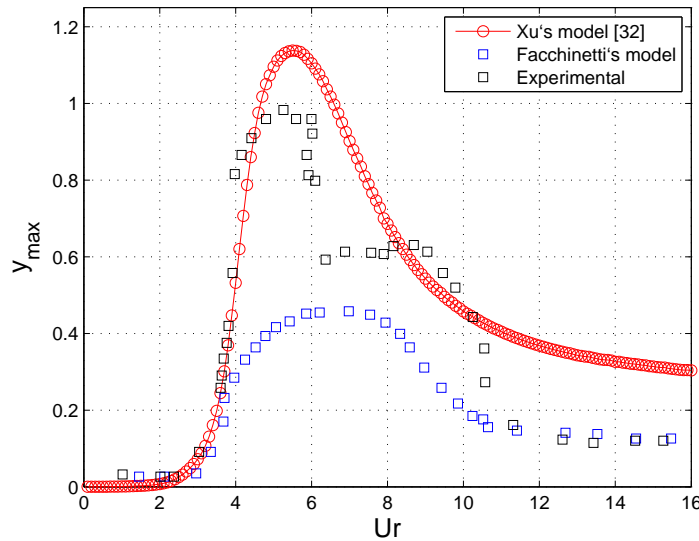


FIG. 3: Maximum amplitude of oscillation as a function of the reduced velocity U_r .

the solution with respect to the random space. More efficient and robust methods have been designed in order to cope with those drawbacks [see [12] and references therein]. Motivated by the abrupt change on the response within the lock-in regime, we adopted here the ASGC introduced in [12]. This method introduces a local refinement based on the hierarchical sparse grid collocation discussed earlier. Note that the discontinuity on the uncertain parameter A might contribute to the existence of sharp gradients on the response. We performed a parameter sweep over the deterministic reduced velocity U_r in the range $[1, 16]$ with constant step 0.1. This yields 160 stochastic problems to be solved using both Monte Carlo and ASGC methods.

The Monte Carlo analysis was performed as a reference solution. Chiron automatically stores provenance data, such as files consumed and produced, standard output and standard error, execution time, and the computer node that executed the program. Additionally, for both Monte Carlo and ASGC analyses, we also stored important provenance data (i.e., number of points, converged error, average, and variance of energy that can be queried during runtime). In the Monte Carlo analysis, the number of samples is fixed on 30,000 and we used the provenance data to check whether the simulation converged in all cases. In the ASGC analysis, the number of interpolation points (interpolation level) is left free to vary according to an assigned error [12]. Monitoring the number of points, stored in the provenance database, is important because each case has two stopping criteria: the first is related to the acceptable error being reached and the second registers whether the program exceeds the number of points used. However, it is not acceptable if the execution is stopped by the second criterion. When this occurs, the case is marked to be adjusted and reexecuted with new input parameters.

Because of the large amount of data and complexity of the analysis to be carried out, we used Chiron to coordinate the parallel stochastic simulations. In order to explore the parallel environment and the flexibility provided by Chiron, we use 32 cores to execute simultaneous simulations. For Monte Carlo, each processor is allocated to a fixed reduced velocity, which handles the corresponding stochastic simulation. As the ASGC computational kernel is implemented as a parallel code, a different configuration is employed. Each reduced velocity leading to an adaptive stochastic analysis is allocated to a group of eight processors running in parallel. Each one of these processors is made responsible for computing the surplus associate to a grid point.

There are significant changes in the response of the flow-structure-coupled system, characterized by larger amplitude responses and synchronization of the oscillating frequency within the lock-in regime. Thus, we expect to observe distinct convergence behavior with respect to the random dimension for a fixed reduced velocity. Besides, for time-dependent problems, the error of the approximation of the stochastic response might grow substantially during long-term integration, as described in [34]. Employing adaptivity has represented a viable alternative in order to cope with those drawbacks inherent to the lock-in regime. No instabilities along the time integrations were detected, and a fixed-error threshold was previously assigned such that a minimum accuracy was ensured automatically by increasing the interpolation level when required. Indeed, the level of refinement was used as the provenance target.

Now, we turn our attention toward the uncertainty analysis of the lock-in phenomena. We select $U_r = 5.7$ as the reference reduced velocity because it corresponds to a response amplitude peak for the deterministic analysis in Fig. 3. Thus we perform a UQ analysis using both uncertain parameters equal to their mean value. Figures 4 and 5 show the evolution of the mean μ and standard deviation σ of the structure response $y(t, \omega)$. In both cases, we also plot a reference solution computed with Monte Carlo. At this point, it is interesting to note the contrast between the deterministic solution (Fig. 3) and the stochastic mean response shown in Fig. 6. Extreme response of structures, particularly within the offshore industry, are often characterized by means of the probability of trespassing a certain safe threshold. Along with that, the first time in which the structure attains such a threshold is considered a critical point. Accordingly, we adopted as a quantity of interest to be tracked along the simulations $\mu + 2\sigma$, which is marked with a dot in Figs. 4 and 5. That represents a rough estimation of the maximum amplitude to be achieved by the structure with nonzero probability to occur. Its dependence with the reduced velocity is presented in Fig. 6. A more refined analysis, involving reliability of the structure, which often is linked with events of low probability, is enabled by obtaining the statistics tails contained in a PDF, such as the one plotted in Fig. 7 for the fixed time $t = 149.6$ and $U_r = 5.7$. The PDF was approximated by sampling the interpolation function calculated by the ASGC method. Replacing an exact solution, not available for this example, by the aforementioned MC simulation, we estimate the time evolution of the error approximation. Figures 8 and 9 show the time-dependent pointwise relative error between the MC and ASGC solutions, for the mean μ and the standard deviation σ , respectively. From Figs. 8 and 9, we can

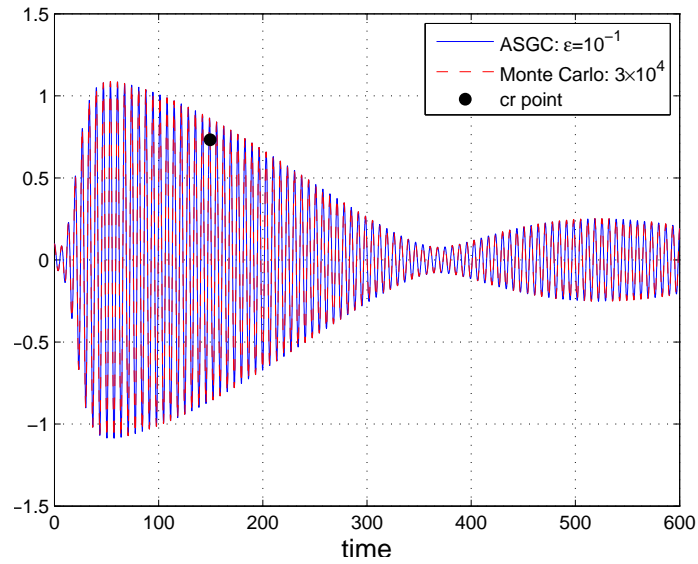


FIG. 4: Evolution of average for reduced velocity $U_r = 5.7$. A dot was plotted to label the critical point (cr) at $t = 149.6$.

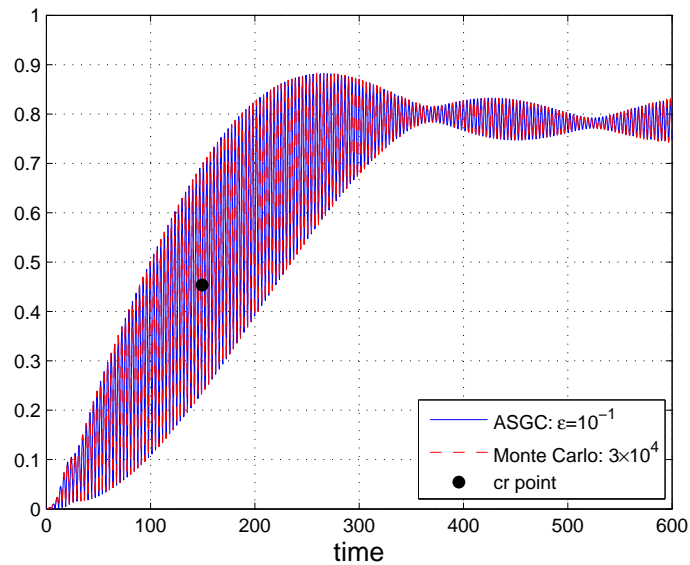


FIG. 5: Evolution of standard deviation for reduced velocity $U_r = 5.7$. A dot was plotted to label the critical point (cr) at $t = 149.6$.

observe that the adaptive scheme is able to handle potential numerical instabilities triggered along time [34] and keep the approximation error within acceptable levels.

4.2 Lid-Driven Cavity Flow

Large-eddy computational models have become popular as a simulation tool for complex turbulent flows. LES aims at reducing computational costs by introducing a spatial low-pass filter, often implicitly defined through the numerical

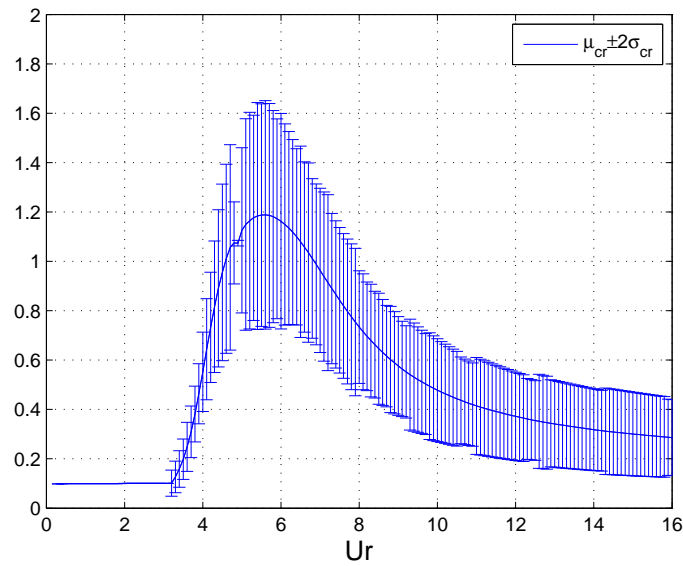


FIG. 6: Mean μ_{cr} and standard deviation $2\sigma_{cr}$ of the cross-flow structure displacement computed at the critical point versus the reduced velocity U_r .

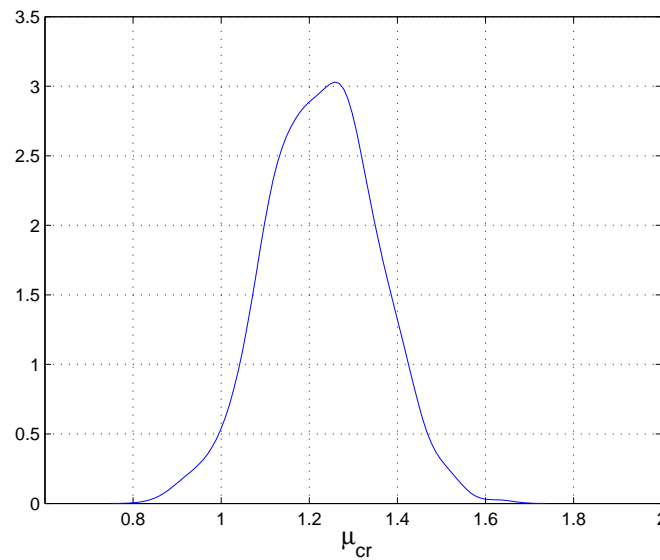


FIG. 7: PDF approximation of the response $y(t = 149.6, \omega)$ computed for $U_r = 5.7$ at the respectively critical point (cr).

grid. The filtering leads to subgrid-scale stresses that must be modeled. The use of such approach has motivated a large body of research. A critical review, with particular emphasis on the multiscale character of turbulence can be found in [30]. LES methods are built using different discretization schemes, encompassing finite differences, finite volumes, finite elements, and spectral methods. The complexity of the resulting numerical models makes analysis troublesome, and therefore, to get a deeper understanding of LES applicability, numerical investigation emerges as a viable way. Here, we perform a preliminary assessment of LES turbulent simulations using a stabilized finite element method and a static Smagorinsky model for representing the subgrid stresses dissipation. This analysis is carried out by evaluating

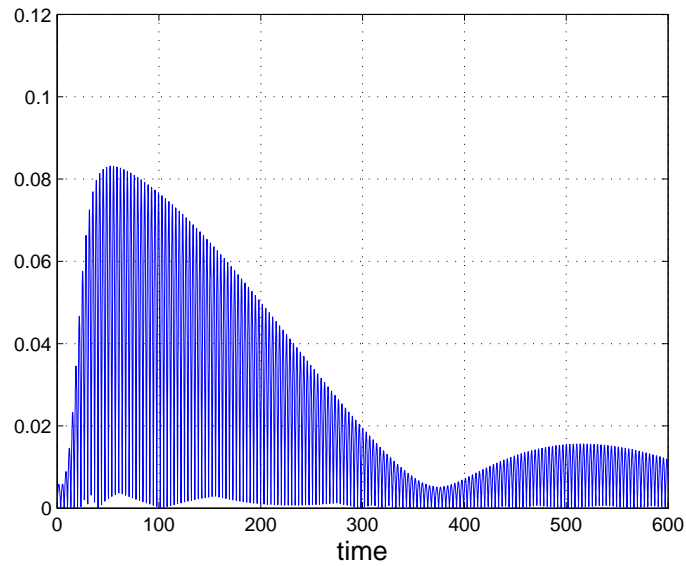


FIG. 8: Evolution of the pointwise difference between MC and ASGC of the mean μ .

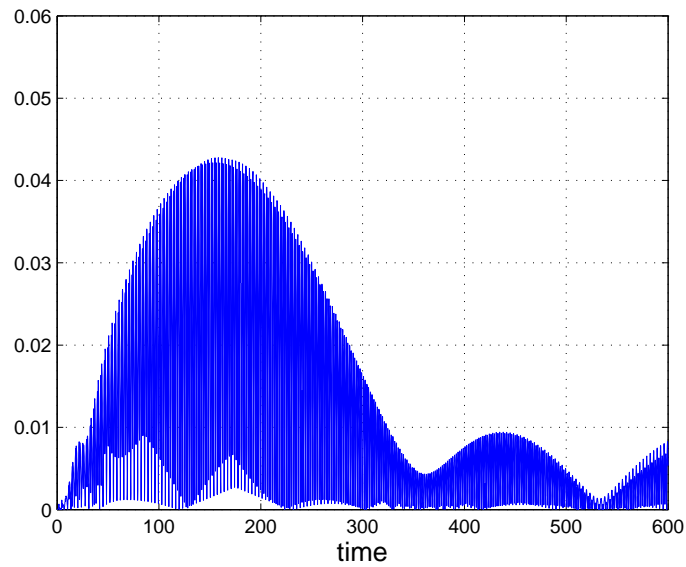


FIG. 9: Evolution pointwise difference between MC and ASGC of the standard deviation σ .

the sensitivity of the computed response to the Smagorinsky parameter, which will be considered a random variable aiming at estimating the variability of LES predictions with respect to uncertainty in the model coefficient. We have chosen the 3D lid-driven cavity flow at Reynolds number 12×10^3 [35] as a benchmark problem for this analysis. That is a confined flow within a cubic region submitted to no-slip boundary conditions on the walls and such that the flow is driven by the imposed lid motion. At that point, it is worthwhile to make clear that the stochastic approach used here is not related to the classical Reynolds average Navier-Stokes (RANS) equations, in which the velocity fluctuations associated to turbulence are viewed as random variables.

EdgeCFD [36, 37], a deterministic flow solver for Navier-Stokes equations, is employed in this study. The main characteristics of this incompressible flow solver are streamline-upwind/Petrov-Galerkin (SUPG) and pressure-stabili-

zing/PetrovGalerkin (PSPG) [38] stabilized finite element formulation; implicit time-marching scheme with adaptive time-stepping control; advanced inexact Newton solvers; edge-based data structures to save memory and improve performance; support to message passing, and hybrid parallel programming models; LES extensions using a classical Smagorinsky model or the variational multiscale method [39]. As is well known, finite elements might suffer from numerical instabilities in the presence of the convective term and also due to the use of equal-order interpolations for velocity and pressure. That can be remedied by a stabilized formulation such as the one adopted here.

As stated earlier, we adopted the modeling of turbulence by LES. In this model, the Navier-Stokes equations are solved for larger scales, whereas the effects of smaller scales are approximated through an augmented viscous dissipation that is included in the stress tensor. Hence, in the static Smagorinsky model, these scales are modeled by adding a term called turbulent viscosity that is given by

$$\nu_T = (C_s L_e)^2 |S|, \quad (16)$$

where S stands for the symmetric part of the local strain rate tensor and L_e to a filter width, usually defined as the cubic root of the element volume, and C_s is the Smagorinsky constant. For this reason, the sensitivity analysis is carried out considering the Smagorinsky parameter as a random variable described by

$$C_s = \overline{C}_s + \sigma_{C_s} \xi, \quad (17)$$

where $(\overline{C}_s, \sigma_{C_s}) = (0.35, 0.35)$, and ξ an independent random variable with uniform distribution taking values on $(-1, 1)$; therefore, C_s varies in the range $(0, 0.7)$. Those intervals, encompassing the standard choice for the Smagorinsky constant $C_s = 0.1$, are chosen such that comparisons with the results in [30] would be possible. The computational domain of the cavity is discretized with a nonuniform mesh of 32^3 tetrahedral elements. Figures 10 and 11 show, respectively, the coarse and the fine computational meshes and their respective velocity fields at the end of simulation time. These solutions were calculated with $C_s = 0.1$ and only used as a reference.

Despite the simple domain, turbulence develops complex flow patterns inside the cubic region compatible with the high Reynolds number of this problem [35]. This makes this problem quite attractive for the present purposes. Because the LES models directly impart the energy dissipation mechanisms along the several flow scales, the total kinetic energy is adopted as output variable on the present analysis. Each simulation is carried out within an interval corresponding to 360 time steps, insuring the development of the main flow features. A deterministic analysis regarding the convergence of the employed finite element formulation is presented in [40].

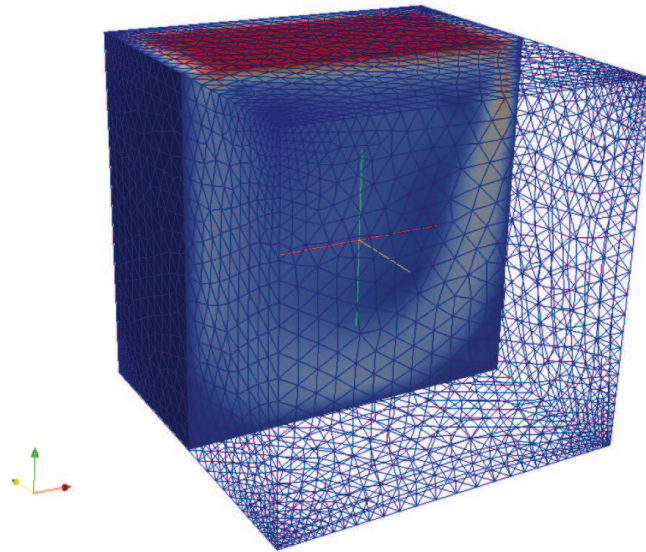


FIG. 10: Velocity field: coarse mesh 32^3 .

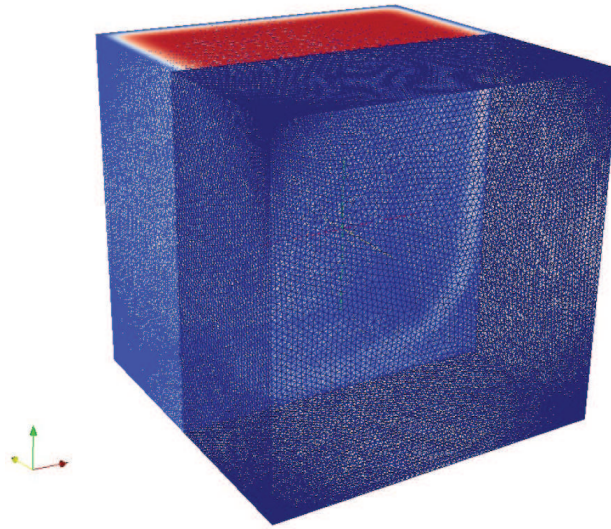


FIG. 11: Velocity field: reference mesh 128^3 .

The stochastic collocation method is implemented in its classical form using Lagrange polynomials as interpolation functions and Clenshaw-Curtis as support nodes. Again, Monte Carlo simulation was used to provide a reference solution. After a preliminary analysis taking into the consideration the convergence of key flow variable mean values, the number of samples for Monte Carlo is fixed in 1000. The MC reference solution is achieved using CSGC with eight approximation levels, which means that 257 support nodes were used. For these values, the difference between the mean of the solutions is on the order of 1.0×10^{-2} , which represents a good approximation for the purposes of this test.

Here, the UQ capabilities are embodied in the workflow generated by Chiron. The provenance information registered is then used to evaluate the statistical moments. This is important to insure reliability of the computed statistics because provenance data could be lost without a structured and systematic controlled approach such as Chiron. Thus, in this case, we could track the difference between MC and CSGC solutions and determine an adequate interpolation level to ensure a minimum error. In this way, the system could, if necessary, automatically increase the interpolation level and resubmit the experiment to obtain the statistical moments of the solution with a prefixed error level.

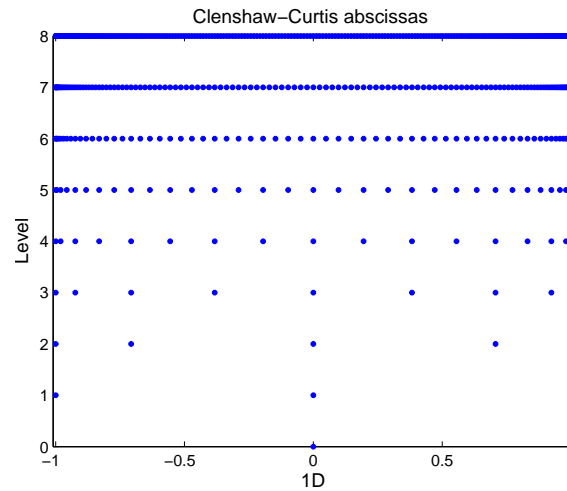
The computational implementation in Chiron starts with allocating a number of cores, which in the present case is 32, each one running the deterministic EdgeCFD finite element code. After that, going through the workflow, the output data are postprocessed and the kinetic energy computed. Finally, the statistical moments are computed and made available for analysis, including the stochastic convergence evaluation for both MC and CSGC methods.

Again, provenance data, such as files consumed and produced, standard output and standard error, execution time, and the computer node that executed the program, are used in real time to verify if any deterministic solver failed. Thus far, none have failed. Figure 12 presents the hierarchical evolution of the grid on the random domain for the CSGC solution, and Table 1 summarizes the computational data associated with the stochastic simulations. Computational performance expressed by the runtime analysis for both MC and CSGC corresponds to parallel executions. That leads to a significant speed up of the analysis when compared to sequential sweep over the input points on the random domain. It is worth mentioning that the computational costs of both simulation methods are strongly related to the number of deterministic problems to be solved until convergence is reached. CSGC is way more efficient than MC on those experiments.

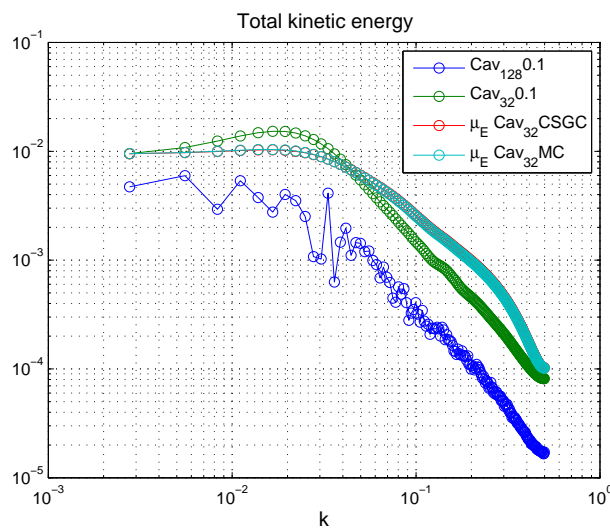
Note by the data in Table 1 that, for instance, the total serial MC simulation time would be more than 4000 h to be completed. Thus, we observe a significant speed-up improvement for this kind of analysis using Chiron. After evaluating the performance of the stochastic collocation as a stochastic simulation tool in a complex turbulent flow, we now summarize the sensitivity of the LES model to parametric uncertainty linked to the subgrid model. We assume that the

TABLE 1: Number of simulations and total time for MC and CSGC methods

Number max of simulations MC	1000
Number max of simulations CSGC	257
Single representative serial simulation time	4.4 h
Parallel MC total time analysis	126.1 h
Parallel CSGC total time analysis	39.2 h

**FIG. 12:** Unidimensional sparse grid for increasing interpolation level.

kinetic energy is an ergodic process, inasmuch the mean and variance achieve constant values after a little while. That enables the use of Fourier transform to obtain the energy spectra. Figure 13 depicts those energy spectra corresponding to the computed results, for the coarse mesh of 32^3 elements, with stochastic collocation ($\mu_E \text{Cav}_{32} \text{CSGC}$ curve in the figure) and Monte Carlo ($\mu_E \text{Cav}_{32} \text{MC}$ curve in the figure), which are not distinguished in the scale of Fig. 13.

**FIG. 13:** FFT of total kinetic energy mean calculated with Monte Carlo and CSGC.

Moreover, we also plot the spectrum of the kinetic energy obtained using a nominal model corresponding to $C_s = 0.1$ ($\text{Cav}_{32}0.1$ curve in the figure), a typical value for the Smagorinsky coefficient. Those results are compared to the ones obtained with the fine mesh simulation and the reference model ($\text{Cav}_{128}0.1$ curve in the figure). Figure 14 shows the mean value of the global kinetic energy evolution with time and its variability is plotted with bars centered at the mean and of length twice the standard deviation. Those results are labeled in Fig. 14 as $(\mu_E \pm 2\sigma_E \text{Cav}_{32} \text{CSGC})$ and $(\mu_E \pm 2\sigma_E \text{Cav}_{32} \text{MC})$ for results computed with Monte Carlo and CSGC, respectively. It must also be noted from this plot, that due to the nonlinearity of Navier-Stokes equations, the difference between the mean kinetic energy obtained with the stochastic model and the kinetic energy computed using the reference deterministic model, the last labeled as ($\text{Cav}_{32}0.1$). Figure 14 also contains, as a check on the capability of the LES model to reproduce key characteristics of the turbulent flow, the plot of the energy evolution computed with the fine mesh ($\text{Cav}_{128}0.1$). Although this solution cannot be seen as a DNS computation, we expect that the role of the Smagorinsky model significantly decreases because the filter length parameter is much smaller. Besides, in [40] the same fine mesh was used and its energy decay rate compares reasonably with theoretical results.

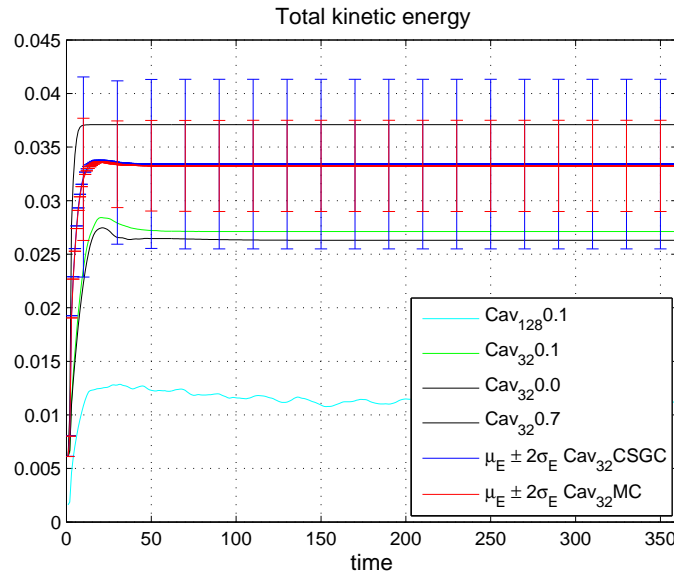


FIG. 14: Evolution of total kinetic energy mean calculated with Monte Carlo and CSGC.

5. CONCLUSION

We have explored the capabilities of Chiron, a scientific workflow engine especially designed to manage and support simulation-based analysis involving large-scale engineering applications using high-performance parallel computers, in the context of uncertainty quantification. Two different problems, corresponding to two different computational implementations, demonstrated the flexibility and effectiveness of Chiron for supporting UQ analyses. Gathering and managing data through provenance control helped in the computation of the probabilistic description of the solutions. The chosen application problems also bear interesting characteristics in terms of UQ analysis, which were also pursued here. The first example deals with the vibrational response of submerged structures excited by vortex detachments in the surrounding flow. We adopted a popular engineering model to predict this dynamics, with special attention to the lock-in phenomena. Two of the parameters used by this modeling were considered uncertain, inasmuch as they often are obtained by means of model calibration based on field observations or physical experiments.

The second situation studied a CFD benchmark problem, the lid-driven cavity in turbulent regime. We presented a preliminary analysis of the performance of a LES model to reproduce the fine scale behavior, translated into energy

dissipation of the flow, within a probabilistic perspective. We considered the Smagorinsky coefficient a uniform random variable and studied the impact of this uncertainty on the predictions. We intend, in the near future, to further study this critical evaluation of Smagorinsky model and the present UQ analysis helps on providing a rational basis for comparisons to available experimental results.

We also checked the performance of stochastic collocation as tool for UQ. Convergence on the random dimension assessed in those two nonlinear problems by comparisons with Monte Carlo simulation. In our numerical experiments, the two stochastic collocation methods have outperformed the classical stochastic simulation technique.

ACKNOWLEDGMENTS

We express our gratitude to Dr. Xiang Ma and Professor Nicholas Zabaras for insightful discussions on stochastic simulation and for providing us with the adaptivity collocation computational kernel. Computer time on a SGI Altix 8200 is provided by the High Performance Computer Center of COPPE/Federal University of Rio de Janeiro. This work is partially supported by CNPq, CAPES, FAPERJ, and Petrobras.

REFERENCES

1. Elishakoff, I., Notes on philosophy of the monte carlo method, *Int. Appl. Mech.*, 39:753–762, 2003.
2. Babuska, I., Tempone, R., and Zouraris, G. E., Galerkin finite element approximations of stochastic elliptic partial differential equations, *SIAM J. Numer. Anal.*, 42(2):800–825, 2004.
3. Xiu, D. and Hesthaven, J. S., High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.
4. Xiu, D. and Karniadakis, G. E., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24:619–644, 2002.
5. Adams, B. M., Dalbey, K. R., Eldred, M. S., Gay, D. M., Swiler, L. P., Bohnhoff, W. J., Eddy, J. P., and Hough, P. D., DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, Sandia Tech. Rep. No. SAND2010–2183, 2010.
6. van der Aalst, W. and van Hee, K. M., *Workflow Management: Models, Methods and Systems*, MIT Press, Cambridge, MA, 2002.
7. Deelman, E., Gannon, D., Shields, M., and Taylor, I., Workflows and e-science: An overview of workflow system features and capabilities, *Future Gen. Comput. Syst.*, 25(5):528–540, 2009.
8. Raicu, I., Zhao, Y., Dumitrescu, C., Foster, I., and Wilde, M., Falcon: a fast and light-weight task execution framework, *IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC07)*, 2007.
9. Shoshani, A. and Rotem, D., *Scientific Data Management: Challenges, Technology, and Deployment*, 1st ed., Chapman and Hall/CRC, 2009.
10. Ogasawara, E., Dias, J., Oliveira, D., Porto, F., Valduriez, P., and Mattoso, M., An algebraic approach for Data-Centric scientific workflows, *VLDB*, Seattle, 2011.
11. Zhao, Y., Hategan, M., Clifford, B., Foster, I. T., von Laszewski, G., Nefedova, V., Raicu, I., Stef-Praun, T., and Wilde, M., Swift: Fast, reliable, loosely coupled parallel computation, *Proc. of IEEE SCW'07*, pp. 199–206, 2007.
12. Ma, X. and Zabaras, N., An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, *J. Comput. Phys.*, 228:3084–3113, May 2009.
13. Ghanem, R. G. and Spanos, P. D., *Stochastic Finite Elements: A Spectral Approach*, 5th ed., Springer, Berlin, 1991.
14. Oskendal, B., *Stochastic Differential Equations. An Introduction with Applications*, 5th ed., Springer, Berlin, 1998.
15. Xiu, D., Fast numerical methods for stochastic computations: A review, *Commun. Comput. Phys.*, 5(2-4):242–272, 2009.
16. Nobile, F., Tempone, R., and Webster, C. G., A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, 46:2309–2345, May 2008.

17. Griebel, M., Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences, *Computing*, 61:151–179, 1998.
18. Walker, E. and Guiang, C., Challenges in executing large parameter sweep studies across widely distributed computing environments, *Proc. of 5th IEEE Workshop on Challenges of Large Applications in Distributed Environments*, pp. 11–18, New York, 2007.
19. Deelman, E., Mehta, G., Singh, G., Su, M.-H., and Vahi, K., Pegasus: Mapping large-scale workflows to distributed resources, *Workflows for e-Science*, Taylor, I. J., Deelman, E., Gannon, D. B., and Shields, M., eds., pp. 376–394, Springer, London, 2007.
20. Thain, D., Tannenbaum, T., and Livny, M., Condor and the grid, *Grid Computing: Making the Global Infrastructure a Reality*, Berman, F., Fox, G., and Hey, T., eds., Wiley, Hoboken, NJ, 2002.
21. Bayucan, A., Henderson, R. L., and Jones, J. P., *Portable Batch System Administration Guide*, Veridian System, 2000.
22. Oliveira, D., Ogasawara, E., Baiao, F., and Mattoso, M., Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows, *Proc. of IEEE International Conference on Cloud Computing*, pp. 378–385, 2010.
23. Mattoso, M., Werner, C., Travassos, G. H., Braganholo, V., Ogasawara, E., Oliveira, D., Cruz, S. M., Martinho, W., and Murta, L., Towards supporting the life cycle of large scale scientific experiments, *Int. J. Business Process Integr. Man.*, 5(1):79–92, 2010.
24. Global Development Group, PostgreSQL 9.0.4 documentation, <http://www.postgresql.org/>, 2009.
25. Carpenter, B., Getov, V., Judd, G., Skjellum, A., and Fox, G., MPJ: MPI-like message passing for java, *Concurrency: Practice Exp.*, 12(11):1019–1038, 2000.
26. Simpson, B. and Toussi, F., HSQLDB User Guide, <http://hsqldb.org/doc/guide/guide.pdf>, 2007.
27. Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T., VisTrails: visualization meets data management, *Proc. of 2006 ACM SIGMOD International Conference on Management of Data*, Chicago, Illinois, pp. 745–747, 2006.
28. Cheung, S., Oliver, T., Prudencio, E. E., Prudhomme, S., and Moser, R., Bayesian uncertainty analysis with applications to turbulence modeling, *Reliab. Eng. Syst. Safety*, 96(9):1137–1149, 2011.
29. Dong, S. and Shen, J., An unconditionally stable rotational velocity-correction scheme for incompressible flows, *J. Comput. Phys.*, 229:7013–7029, 2010.
30. Lucor, D., Meyers, J., and Sagaut, P., Sensitivity analysis of large-eddy simulations to subgrid-scale-model parametric uncertainty using polynomial chaos, *J. Fluid Mech.*, 585(1):255–279, 2007.
31. Gabbai, R. and Benaroya, H., An overview of modeling and experiments of vortex-induced vibration of circular cylinders, *J. Sound Vib.*, 282(3-5):575–616, 2005.
32. Xu, W. H., Wu, Y. X., Zeng, X. H., Zhong, X. F., and Yu, J. X., A new wake oscillator model for predicting vortex induced vibration of a circular cylinder, *J. Hydrodyn.*, 22(3):381–386, 2010.
33. Facchinetti, M. L., de Langre, E., and Biotley, F., Coupling of structure and wake oscillators in vortex-induced vibrations, *J. Fluids Struct.*, 19(2):123–140, 2004.
34. Gerritsma, M., van der Steen, J.-B., Vos, P., and Karniadakis, G. E., Time-dependent generalized polynomial chaos, *J. Comput. Phys.*, 229(22):8333–8363, 2010.
35. Bouffanais, R., Deville, M. O., and Leriche, E., Large-eddy simulation of the flow in a lid-driven cubical cavity, *Phys. Fluids*, 19(5):055108, 2007.
36. Elias, R. N. and Coutinho, A. L. G. A., Stabilized edge-based finite element simulation of free-surface flows, *Int. J. Numer. Methods Fluids*, 54(6-8):965–993, 2007.
37. Elias, R. N., Paraizo, P. L. B., and Coutinho, A. L. G. A., Stabilized edge-based finite element computation of gravity currents in lock-exchange configurations, *Int. J. Numer. Methods Fluids*, 57:1137–1152, 2008.
38. Tezduyar, T., Stabilized finite element formulations for incompressible flow computations, *Adv. Appl. Mech.*, 28:1–44, 1992.
39. Gravemeier, V., Gee, M. W., Kronbichler, M., and Wall, W. A., An algebraic variational multiscale-multigrid method for large

- eddy simulation of turbulent flow, *Comput. Methods Appl. Mech. Eng.*, 199(13-16):853–864, 2010.
40. Lins, E. F., Elias, R. N., Guerra, G. M., Rochinha, F. A., and Coutinho, A. L. G. A., Edge-based finite element implementation of the residual-based variational multiscale method, *Int. J. Numer. Methods Fluids*, 61(1):1–22, 2009.