

Multi-Level Monte Carlo sampling with Parallel-in-Time Integration for Uncertainty Quantification in Electric Machine Simulation

Robert Hahn ¹, Sebastian Schöps ¹

¹Technical University of Darmstadt, Darmstadt, Germany, robert.hahn@tu-darmstadt.de

While generally considered computationally expensive, Uncertainty Quantification using Monte Carlo sampling remains beneficial for applications with uncertainties of high dimension. The Multi-Level Monte Carlo method reduces the overall computational effort, but is unable to reduce the time to solution in a sufficiently parallel computing environment. In this work, we propose a Uncertainty Quantification method combining Multi-Level Monte Carlo sampling and Parallel-in-Time integration for select samples, exploiting remaining parallel computing capacity to accelerate the computation. As a first numerical test, the method is applied to an electric induction machine model, where a speedup of around 27.2% compared to Multi-Level Monte Carlo sampling is observed.

Index Terms—Electric machines, Uncertainty, Monte Carlo methods, Parallel algorithms, Numerical simulation

I. INTRODUCTION

Uncertainty quantification (UQ) has become popular in electrical engineering applications such as electrical machine design. In practice, the number of random variables can be substantial, arising from variations in material properties, geometry, boundary conditions and sources [1], [2], [3]. A common objective is to propagate these uncertainties forward in order to evaluate failure rates, stochastic moments (e.g. mean values) and sensitivities of quantities of interest. Several strategies have been developed for this purpose. Among these, stochastic collocation and stochastic Galerkin methods, often combined with polynomial chaos expansions [4], are efficient when the number of uncertain input parameters is small. However, as the dimensionality of the uncertainty increases, these methods eventually become infeasible due to the curse of dimensionality. While sparse grid techniques [5] can alleviate this problem to some extent, they cannot eliminate it completely. On the other hand, the Monte Carlo (MC) method is not affected by dimensionality, but is considered too expensive in its naive implementation. Therefore, multilevel (ML) sampling has been developed [6] and has been shown to perform well when hierarchically combining different spatial resolutions [7]. While MLMC reduces the computational effort, the time to solution remains unaffected in a computing environment with sufficient parallel capacity.

In this paper we propose to use MLMC with hierarchical refinement in the time domain (as opposed to the space domain as in [7]). This is a significant advance in cases where, for example, long transient simulations are required to obtain the steady state. We combine this with parallel-in-time integration to reduce the overall computational cost of UQ in a high performance computing (HPC) environment.

II. METHODOLOGY

Let's assume that the model, e.g. an electric machine, depends on input parameters ξ and performing the simulation produces some output Quantity of Interest (QoI) $Q = Q(\xi)$, e.g. torque. The inputs are seen as realizations of some underlying random distribution and thus Q is random as well. Its

stochastic moments, e.g., expected value, are then determined by evaluating samples. The Monte Carlo (MC) method draws N independent samples $\xi^{(k)}, k \in \{1, \dots, N\}$ from the input parameter distribution, and the numerical simulation required for each input parameter sample can then be performed in parallel on different processor cores or computing nodes without communication overhead, making the method well suited for HPC. The MC estimation for the expected value of the output quantity Q is computed via

$$\mathbb{E}_{\text{MC}}[Q(\xi)] = \frac{1}{N} \sum_{k=1}^N Q^{(k)}, \quad (1)$$

where $Q^{(k)} = Q(\xi^{(k)})$ denotes the simulation result for input parameters $\xi^{(k)}$. The quality of this estimate depends on the number of samples N . As the MC estimator is unbiased, i.e. $\mathbb{E}[\mathbb{E}_{\text{MC}}[Q]] = \mathbb{E}[Q]$, we can expect to recover the true value of $\mathbb{E}[Q]$ for $N \rightarrow \infty$, with a Root Mean Square Error (RMSE) of $\mathcal{O}(N^{-0.5})$ which is, as mentioned previously, rather slow but independent of the dimension of ξ .

To reduce the computational cost of MC sampling, the Multi-Level Monte Carlo (MLMC) method introduces different discretization levels $\ell \in \{0, \dots, L\}$ where $\ell = L$ corresponds to the original model and $\ell \in \{0, \dots, L-1\}$ correspond to coarser discretizations (in time), which produce less accurate solutions but are cheaper to evaluate. In the MLMC method, N_ℓ samples are drawn for each discretization level ℓ . Each sample is then evaluated twice, on discretization levels ℓ and $\ell-1$. The MLMC estimate of the expected value is given by

$$\mathbb{E}_{\text{MLMC}}[Q(\xi)] = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{k=1}^{N_\ell} [Q_\ell^{(k)} - Q_{\ell-1}^{(k)}]. \quad (2)$$

As shown in [6], the samples evaluated on coarser discretization levels do not increase the bias error, i.e.

$$\mathbb{E}[\mathbb{E}_{\text{MLMC}}[Q]] = \mathbb{E}[Q_L]. \quad (3)$$

They do however contribute to a reduction of the variance error, and the optimal number of samples N_ℓ per level ℓ follows from an optimization problem if the evaluation cost and variance

on each level are known. Otherwise, these quantities can be estimated at run-time.

To reduce the time-to-solution for numerically expensive models, methods that can make use of parallel computing hardware are desirable. For models which require small timesteps or the simulation over large timescales, Parallel-in-Time (PinT) methods can become attractive. The Parareal algorithm, introduced in [8] and applied to machines in [9], divides the simulation interval $[T_0, T_M]$ into M sub-intervals $[T_n, T_{n+1}]$ and then iteratively employs a coarse and a fine propagator. After the coarse propagator $\mathcal{C}$ – a cheaper and less accurate timestepping method – is used to sequentially determine initial values $\mathbf{U}_n^{(0)} = \mathcal{C}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(0)})$ for each interval, the resulting initial value problem (IVP) for each sub-interval is solved in parallel using the fine propagator $\mathcal{F}$, a more expensive and more accurate timestepping method.

To alleviate the discontinuities at the interval boundaries this procedure introduces, the Parareal algorithm uses another sequential pass over the intervals, in which one new coarse solution per interval has to be computed. Combining the fine solution, the new coarse solution and the previous coarse solution, the Parareal update equation

$$\mathbf{U}_n^{(k+1)} = \mathcal{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) + \mathcal{C}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k+1)}) - \mathcal{C}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{(k)}) \quad (4)$$

generates new initial values for the next iteration. This process is repeated iteratively, until the discontinuities at the interval boundaries fall below a prescribed tolerance. After k iterations, the solution on the first k intervals is exact up to the accuracy of the fine propagator, and the discontinuities at the first k interval boundaries vanish. Typically, the coarse propagator is chosen to be much cheaper than the fine propagator, s.t. the sequential update overhead can be neglected. If the algorithm converges after sufficiently few iterations, the overall time-to-solution is reduced.

The combination works as follows: we use Parareal only on the finest level L , which is responsible for the longest computation times but requires the fewest sample evaluations. Here, leftover parallel computing capacity not saturated with sample parallelization is exploited to achieve a considerable speedup. Under optimal conditions, the time to solution for samples on level L can be driven down to the time needed on level $L - 1$.

III. NUMERICAL EXAMPLE

As a first numerical test, we consider the Steinmetz model of the induction machine shown in [10, p. 108]. The nominal values of the machine parameters are shown in Tab. I. For each parameter, we assume a uniform distribution between 95% and 105% of the nominal value. We perform a transient simulation for $t \in [0 \text{ s}, 1 \text{ s}]$ using the implicit Euler method to analyze the start-up behavior of the machine, with the electrical energy consumed in this timeframe used as the QoI. The discretization level ℓ determines the time step size, with step sizes $\Delta t \in \{1 \times 10^{-3} \text{ s}, 1 \times 10^{-4} \text{ s}, 1 \times 10^{-5} \text{ s}\}$ for levels $\ell \in \{0, 1, 2\}$, respectively. For a single evaluation on the finest level, benchmarks show a reduction in time-to-solution

TABLE I
NOMINAL VALUES OF MACHINE PARAMETERS.

Name	Unit	Value
Stator resistance	Ω	$1.111\,140 \times 10^{-1}$
Squirrel cage resistance	Ω	$7.158\,602 \times 10^{-2}$
Iron loss resistance	Ω	$1.736\,354 \times 10^6$
Leakage inductance of the stator windings	H	$1.649\,983 \times 10^{-3}$
Leakage inductance of the squirrel cage	H	$1.063\,014 \times 10^{-3}$
Main inductance	H	$6.407\,774 \times 10^{-2}$

of 32.5% when using the Parareal algorithm. Taking the cost of the coarser samples into account, the overall time-to-solution is reduced by 27.2% compared to MLMC without Parareal.

IV. CONCLUSION AND OUTLOOK

In this work, we presented a method combining Multi-Level Monte Carlo sampling with Parareal. Since the evaluation of individual samples is trivially parallel, the available parallel computing capability should first be leveraged by parallelizing over samples. The viability of the method is restricted to parallel computing environments, in which remaining parallel capacity can be exploited by the Parareal algorithm.

First benchmarks indicate a speedup of around 27% compared to plain Multi-Level Monte Carlo sampling. In the full paper, we will investigate the method's applicability to more complex machine models.

ACKNOWLEDGEMENT

This work is supported by the Graduate School for CE at TU Darmstadt and the European HPC Joint Undertaking (JU) under grant agreement No. 101118139. The JU receives support from the European Union's Horizon Europe Programme.

REFERENCES

- [1] S. Cl  net, "Uncertainty quantification in computational electromagnetics: The stochastic approach," *ICS Newslett.*, vol. 13, pp. 3–13, 03 2013.
- [2] P. Offermann and K. Hameyer, "A polynomial chaos meta-model for non-linear stochastic magnet variations," *COMPEL*, vol. 32, no. 4, pp. 1211–1218, 2013.
- [3] U. R  mer, S. Sch  ps, and T. Weiland, "Stochastic modeling and regularity of the nonlinear elliptic curl-curl equation," *SIAM/ASA J. UQ*, vol. 4, pp. 952–979, 08 2016. arxiv:1606.07643.
- [4] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton, NJ, USA: Princeton University Press, 2010.
- [5] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta. Num.*, vol. 13, pp. 147–269, 05 2004.
- [6] M. B. Giles, "Multilevel Monte Carlo methods," *Acta. Num.*, vol. 24, pp. 259–328, 2015.
- [7] A. Galetzka, Z. Bontinck, U. R  mer, and S. Sch  ps, "A multilevel Monte Carlo method for high-dimensional uncertainty quantification of low-frequency electromagnetic devices," *IEEE Trans. Magn.*, vol. 55, pp. 1–12, 08 2019. arxiv:1803.09122.
- [8] J.-L. Lions, Y. Maday, and G. Turinici, "A parareal in time discretization of PDEs," *Comptes Rendus de l'Acad  mie des Sciences – Series I – Mathematics*, vol. 332, no. 7, pp. 661–668, 2001.
- [9] S. Sch  ps, I. Niyonzima, and M. Clemens, "Parallel-in-time simulation of eddy current problems using parareal," *IEEE Trans. Magn.*, vol. 54, pp. 1–4, 03 2018. arxiv:1706.05750.
- [10] R. De Weerd and K. F. toegepaste wetenschappen. Departement elektrotechniek ESAT other, "Eindige elementen modellering van kooianker inductiemotoren," 1995.