Applying Active Learning to Adaptive Mesh Refinement Simulations

Dmitry Duplyakin
Department of Computer Science, University of Colorado – Boulder, USA
dmitry.duplyakin@colorado.edu

Advisors: Jed Brown (University of Colorado), Donna Calhoun (Boise State University)

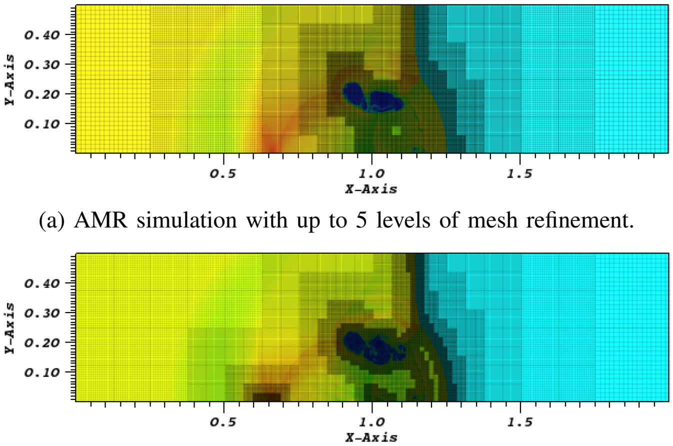
I. Background and Problem Statement

First introduced in the 1980s, Adaptive Mesh Refinement (AMR) is a popular technique in many areas of modern science and engineering. AMR refers to the methods which increase the resolution of the simulation in the regions of the computational domain where the solution exhibits features of interest. In a common approach to AMR, the computational domain is decomposed into logically Cartesian regions consisting of coarse (lower resolution), or fine (higher resolution) meshes. Depending on the implementation, finer regions may cover portions of the coarser grids (patch-based AMR). Alternatively, the decomposition may be a true partitioning of the domain (tree-based AMR). Simulations based on AMR typically see a significant reduction in computational and storage requirements when compared with solutions obtained on uniformly high-resolution meshes.

In AMR computations, it is difficult to predict how much refinement will be applied in a particular simulation. User-defined criteria for refinement may result in drastically varying simulation runtimes. Even experienced users are likely to struggle with choosing the sufficient amount of computational resources that will allow them to meet their computing deadlines. For instance, as shown in Fig. 1, allowing one more level of refinement in shock-bubble simulations with AMR reveals fine features of the studying phenomenon at the expense of a 3.8x increase in the simulation time. It is certain that this growth factor will be different when we enable yet another refinement level, but it is difficult to predict its magnitude. Moreover, when we vary one of many physical parameters – in shock-bubble interactions: viscosity, size and density of the bubble, change of pressure at the shock front, among others, we also observe drastic changes in performance characteristics.

Fig. 2 shows the level of detail we can observe in simulations with high degree of refinement. Such simulations typically have high memory requirements and can take a significant amount of time even on supercomputers and computing clusters. Libraries such as MPI [1] and OpenMP [2] power many AMR libraries and packages by accelerating these computations via distributed-memory and shared-memory parallelism, respectively. However, as the number of machines simultaneously used for these computations increases, the cost of these computational experiments, typically expressed in node-hours or core-hours, also grows. For large real-world applications, the total cost of a desired series of AMR simulations may exceed even large allocations on supercomputers and can become impractical. In our modeling and the proposed analysis of Design of Experiments (DOE), we demonstrate how the magnitude and the variability of performance characteristics can be modeled and describe the practical approach to selecting the most informative and cost-efficient simulation configurations. Our approach aims to avoid running overly expensive computations if the performance characteristics of interest can be predicted with sufficient accuracy. This optimization is vital in applications with modest time-to-solution and total compute budget constraints.

Fig. 1: Visualization of a 2D shock-bubble interaction simulated using ForestClaw package. Switching from (a) to (b) allows to resolve finer features but increases the simulation wall clock time by the factor of 3.8.



(a) AMR simulation with up to 5 levels of mesh refinement.

(b) AMR simulation with up to 6 levels of mesh refinement.

II. Proposed Analysis

We extend our previous work described in [4]. We use a combination of Gaussian Process Regression (GPR) [5] and Active Learning (AL) [6] as a method that allows us to obtain high-confidence predictions across a large input space without the need for a static, and most likely inefficient, experiment

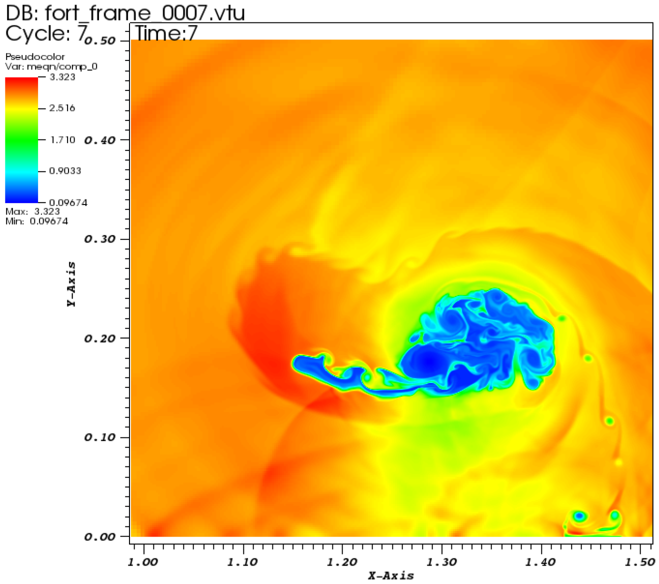


Fig. 2: Visualization of fine features in the simulated shock-bubble interaction produced using VisIt [3].

design. In the current work, we describe how we apply this approach and use the tools we have developed to model a more challenging problem than the HPGMG benchmark [7] studied previously. We switch from GPR modeling with 2 parameters (problem size and the CPU frequency) to the modeling with 5 parameters. These input parameters, or *features*, are the number of processors used, discretization box size, maximum level of refinement, bubble size, and bubble density. The last two are physical properties of the simulated shock-bubble interaction phenomenon; the others represent numerical and machine properties. In our analysis, we focus on AMR in FORESTCLAW [8], the block-structured adaptive finite volume library for solving hyperbolic partial differential equations on mapped, logically Cartesian meshes.

We analyze the simulation cost function that exhibits rapid, unpredictable growth along multiple dimensions in the aforementioned input space. This growth occurs only at particular combinations of the selected parameters. In the rest of the input parameter space, the cost function only changes gradually or remains flat and so can be predicted accurately with little experimental data. With the cost-aware AL algorithms, we cautiously select candidate experiments that explore these growth areas. We analyze the behavior of the proposed experimentation based on AL in situations that we could not investigate previously with the 2-feature dataset and reason about implementation trade-offs. Among the proposed Active Learning algorithms, we identify the most cost-efficient algorithm and demonstrate that its advantages can be significant.

III. Presented Results

We obtain additional evidence that supports our argument made in the previous work for using Active Learning in combination with Gaussian Process Regressions to optimize sequences of parallel computer experiments. We evaluate the analysis techniques in the context of learning and predicting the computational experiments for Adaptive Mesh Refinement simulations. We demonstrate how the proposed Active Learning algorithms that treat model predictions and uncertainty in different ways perform in a 5-feature input space of machine-specific, numerical, and physical parameters of these simulations. In the set of four evaluated Active Learning algorithms, we determine that the proposed exploratory algorithm which we refer to as *RandGoodness* outperforms the competing algorithms by up to 23.0-34.2% with respect to the aggregate prediction errors. We observe this relative advantage with the median values at 9.9-12.6% in the analysis of the complete dataset and its subsets. We conclude that the candidate selection in *RandGoodness* is the most cost-efficient Active Learning algorithm for the type of performance datasets where studied characteristics exhibit rapid growth. AMR users can benefit from employing this Active Learning algorithm to guide their decisions when they run series of simulations from large sets of candidates in parameter sweeps, optimization problems, uncertainty quantification studies, among other types of analysis.

References

[1] M. P. Forum, “MPI: A Message-Passing Interface Standard,” Knoxville, TN, USA, Tech. Rep., 1994.

[2] L. Dagum and R. Menon, “OpenMP: an industry standard API for shared-memory programming,” *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[3] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil, “VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data,” in *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, Oct 2012, pp. 357–372.

[4] D. Duplyakin, J. Brown, and R. Ricci, “Active learning in performance analysis,” in *Proceedings of the IEEE Cluster Conference*, Sep. 2016. [Online]. Available: http://www.flux.utah.edu/paper/duplyakin-cluster16

[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[6] B. Settles, “Active learning literature survey,” *Computer Sciences Technical Report*, vol. 1648, 2009.

[7] M. Adams, J. Brown, J. Shalf, B. V. Straalen, E. Strohmaier, and S. Williams, *HPGMG 1.0: A Benchmark for Ranking High Performance Computing Systems*, May 2014. [Online]. Available: http://www.osti.gov/scitech/servlets/purl/1131029

[8] C. Burstedde, D. Calhoun, K. Mandli, and A. R. Terrel, “ForestClaw: Hybrid forest-of-octrees AMR for hyperbolic conservation laws,” *arXiv preprint arXiv:1308.1472*, 2013.