AIAA AVIATION Forum
22-26 June 2015, Dallas, TX
22nd AIAA Computational Fluid Dynamics Conference

10.2514/6.2015-3052

Opportunities and Challenges for Multi-Disciplinary Analysis and Optimization at Exascale

Juan J. Alonso*
Stanford University, Stanford, CA 94305, USA

The advent of high-performance computers with vastly increased capabilities compared to todays supercomputers opens up tremendous possibilities for the calculation of physical problems without significant modeling assumptions. This is the route that most exascale approaches are being envisioned today, so as to harness the awesome capability that will be offered to the computational scientist and engineer alike. But a whole different set of uses of exascale computing can be envisioned if the applications are in design and Multi-Disciplinary Analysis and Optimization (MDAO): the complexity does not always come from ever-increasing mesh sizes and increasing fidelity in the physics but, rather, from the fact that very large numbers of simulations may need to be carried out in large-dimensional parameter spaces (design optimization an iterative process in nature), uncertainty quantification (where many samples may need to be accomplished), or even combinations of the two such as in design under uncertainty and reliability-based design. Whereas, when faced with leveraging exascale computers for very large individual calculations, the focus is on discovering parallel approaches to improve efficiency, scalability, and turnaround time of the simulations, when attempting MDAO at exascale, the difficulty of discovering and exploiting parallelism is significantly increased (with notable exceptions). This paper focuses on a very personal view, based on years of experience pursuing large-scale multi-physics simulations and MDAO, of the most likely uses of exascale computing platforms in the engineering design discipline, as well as the bottlenecks that may be faced and that require research efforts today so that we are prepared to leverage future computers.

I. Introduction & Motivation

In order to create the next generation of aircraft, rotorcraft, jet engines, rockets, or hypersonic propulsion systems, engineers must evaluate, with sufficient accuracy, the performance of a large number of candidate designs. As computing resources have improved, problem complexity has increased and predictions have become more accurate, reliable, and robust. There is still much more to be gained from computational and algorithmic improvements, however. The mesh resolution of todays models could be increased to improve the accuracy of the calculations and higher-fidelity models, such as LES and turbulent combustion, could be used to improve the accuracy on critical cases and even for design optimization purposes.

The advent of future exascale computing platforms has the potential to revolutionize the accuracy and turnaround times of these simulations for the engineering and scientific communities. We define *exascale computing* as computing that is performed on computers with an aggregate theoretical peak performance of over one exaFLOP (1 billion of operations per second), although and by extension, we also think of exascale computing as that which requires computers with hundreds of thousands to millions of computational elements / cores.

But the challenges abound. If one wants to not only analyze but also design/optimize a system or configuration of interest, the parallelization / scalability problem is only compounded: not only must the component codes scale, but the entire procedure that is used to complete a design, including the disciplinary analyses included, must scale as well simultaneously. This is a significant challenge, not only for scalability, but for appropriate load balancing and system software.

Unfortunately, the problem is further compounded when thinking about exascale: modern parallel machines as well as foreseeable exascale architectures are becoming increasingly complex, with deep, distributed

*Professor, Department of Aeronautics and Astronautics, AIAA Associate Fellow.

memory hierarchies and heterogeneous processing units. Because the costs of communication within these architectures vary by several orders of magnitude, the penalty for mistakes in the placement of data or computation is usually very poor performance/efficiency when measured as a fraction of the theoretical peak. Energy consumption is also dominated by the cost of data transfers and, therefore, scalability in energy efficiency (which is becoming a serious design consideration) can also suffer. For example, CFD solution algorithms (with which the author is most familiar) that were initially developed for serial computers or low-level parallelism (hundreds to a few thousands of processors) fail to provide expected convergence rates when used with more than 5,000-10,000 cores.

Although the basic concepts of MDAO are fairly well accepted in the community, the routine use of MDAO methods is not, by any means, pervasive. Their use, however is certainly being accelerated as computational capabilities continue to improve and disciplinary analyses are created that can be easily embedded into an MDAO environment. This begs the question of whether exascale computing can be an enabler and a catalyzer for further industrial deployment of MDAO. At moderate levels of fidelity (commensurate with analyses conducted during the conceptual design phase), it is common industrial practice to perform coupled multidisciplinary analyses (MDA) of the most tightly integrated disciplines in a design. But such calculations have very low arithmetic intensity and, therefore, are not well suited for exascale computing.

But the approaches to conceptual / preliminary design are changing rapidly: high-fidelity aero-structural analyses, conjugate heat transfer calculations, aero-acoustic simulations, reacting flows, aero-thermo-elastic problems, aero-servo-elasticiy, and even multi-phase flows with material interactions are beginning to be used more heavily. Such calculations do approach the level of complexity and arithmetic intensity that is required to ensure that the power of an exascale computer is brought to gear. A vision that exascale computing could facilitate is the routine use of such high-fidelity, multi-physics analyses in the inner loop of many future aircraft, spacecraft, jet engine, and rotorcraft analysis and design processes.

As part of the effort in the NASA Vision 2030 CFD report that the author participated in, significant effort was devoted to ensuring that a good representation of the needs for CFD in the 2030 time frame was put forward. While in this paper we are concerned with a much larger set of computational analyses (that pertain to MDAO), CFD is a great example to illustrate the challenges ahead. For that reason, this paper borrows some of the committee's thinking for exascale CFD simulations and expands to fill the void required by MDAO. As mentioned in the report, CFD in 2030 will be intimately related to the evolution of the computer platforms that will enable revolutionary advances in simulation capabilities. Any scalable, high-performance implementation of future analyses in an MDAO workflow must map well to the relevant future programming paradigms, which will enable portability to changing HPC environments and performance without major code rework. However, the specific architecture of the computing platforms that will be available is far from obvious. We can, however, speculate about the key attributes of such machines and identify key technology gaps and shortcomings so that, with appropriate development, CFD can perform at future exascale levels on the HPC environments in 2030.

Hybrid computers with multiple processors and accelerators are becoming widely available in scientific computing and, although the specific composition of a future exascale computer is not yet clear, it is certain that heterogeneity in the computing hardware, the memory architecture, and even the network interconnect will be prevalent. Future machines will be hierarchical, consisting of large clusters of shared-memory multiprocessors, themselves including hybrid-chip multiprocessors combining low-latency sequential cores with high-throughput data-parallel cores. Even the memory chips are expected to contain computational elements, which could provide significant speedups for irregular memory access algorithms, such as sparse matrix operations arising from unstructured datasets. With such a running target on the horizon, the descriptions of MDAO at exascale are inspired by a target supercomputer that incorporates all of the representative challenges that we envision in such a computing environment. These challenges are certainly related to heterogeneity, but more concretely, may include multicore CPU/GPU interactions, hierarchical and specialized networks, longer/variable vector lengths in the CPUs, shared memory between CPU and GPUs, and even a higher utilization of vector units in the CPUs.

An underlying assumption in all the comments made in this paper is that MDAO capabilities of the future will play a significant role in routine, multidisciplinary analysis (MDA) and optimization (MDAO) that will be typical of engineering and scientific practice. In fact, in 20-30 years from now, the author envisions that many of the aerospace engineering problems of interest will be of a multidisciplinary nature and that all the relevant disciplines will have to interface seamlessly with each other, including fluid mechanics / aerodynamics, acoustics, structures, heat transfer, reacting flow, radiation, dynamics and control, and even

ablation and catalytic reactions in thermal protection systems. With increasingly available computer power and the need to simulate complete aerospace systems, multidisciplinary simulations will become the norm rather than the exception.

How can exascale computing help realize this vision? There are some issues that we know will require exascale computing:

- Rapid turnaround (hours for MDA and less than a day for MDO).
- User-specified accuracy of coupled simulations, including sufficiently high fidelity in participating disciplines.
- Robustness of the solution methodology, requiring redundant computation.
- Ability to provide sensitivity and uncertainty information for possible embedding in design under uncertainty efforts.
- Effective leveraging of future HPC resources: the mapping of complex MDAO workflows to exascale computers.

As we attempt to discern the challenges of MDAO at exascale, the concept of an *MDAO workflow* becomes important. By *workflow* we mean a sequence of operations that typically involve multiple analyses of one or more of the participating disciplines and that results in an output of interest, typically a design with certain desired characteristics. It is clear that, by MDAO, different practitioners mean very different things and the workflows that would be used will also differ. Themselves, these various concepts present different challenges and opportunities and, therefore, this paper is structured around a number of representative workflows that we will investigate individually. In particular, in the following sections, we examine exascale challenges and opportunities for the following workflows:

1. In Section II we examine opportunities relating to the simulation of high-fidelity multi-disciplinary analyses that may require interactions between multiple disciplines, each represented with high fidelity.
2. In Section III we focus on multi-disciplinary (deterministic) optimization problems, that may involve multiple disciplines, in large-dimensional design spaces and with, possibly, large numbers of constraints.
3. In Section IV we look at the challenges and opportunities to create and adaptively improve surrogate models (including multi-fidelity-based ones) that may themselves be embedded in some of the other workflows considered here.
4. In Section V we look at uncertainty quantification (UQ) in multi-disciplinary systems and the embedding of such UQ methods in Design Under Uncertainty (DUU) approaches.

While these workflows are not all encompassing (there are many MDA/MDO approaches, for example, that attempt to partition a design problem or simultaneously solve the analysis and design problems) it is the case that more complicated workflows will have elements of these individual ones, and that the conclusions / suggestions presented in each of these sections can be extrapolated to more complicated situations.

II. Multi-Disciplinary Analyses

Multi-Disciplinary Analyses (MDAs) are the foundational workflow of MDAO techniques. In them, a number of disciplinary analyses are executed in sequence with, possibly, iterative loops if present. The completion of an MDA then requires the repeated evaluation of a set of disciplinary analyses, in addition to the communication of inputs and outputs among all disciplines. A typical example that only involves two disciplines is the aero-elastic analysis of an aircraft wing configuration. In such an MDA, we think of two main participating disciplines: the unsteady CFD of the flow around an aircraft wing, and the dynamic motion of the structural model (typically a FEM) that underlies it. An MDA in this context would involve (a) the execution of the CFD, (b) followed by the transfer of loads, (c) followed by an execution of the FEM, resulting in (d) displacements that are provided back to the CFD, and a repetition of the (a)-(d) sequence for a number of iterations that are required to simulate, for example, a limit-cycle oscillation. Note that, more complex workflows in aero-elastic analysis may be require (including inner iterations) if, for example, the CFD and FEM solutions need to be converged to zero residual simultaneously.

A typical assumption is that both disciplinary analyses will run simultaneously on the same exascale computer. If this were the case, the entire memory required for the computation can be stored, and the context switching between the execution of any of the disciplines is negligible in terms of computational cost. The challenges for exascale computing of this kind of simulation can be summarized as follows below. Suggestions for solutions are also included.

1. Scalability and performance of individual computations: The fundamental unit problem in leveraging exascale computing for MDAO is to be able to scale the individual discipline analysis so that it can efficiently utilize vast computational resources. This necessarily implies that the computational requirements for the analysis (mesh sizes, computational intensity / required numbers of FLOPS, number of iterations, etc) are large so that many cores can be employed simultaneously. However, it might not be necessary to employ an entire exascale computer effectively, when the required mesh size for a complete aircraft configuration in cruise may be in the neighborhood of 500 million cells / elements? If the computational requirements for the desired accuracy are fixed, then in a strong scaling sense, a limit to the scalability for such an analysis will appear. The only alternatives in these situations would be to further improve the parallel implementation of the disciplinary analysis in order to push the scalability of the solution to as many processing cores as possible, and to live with the limitation on the potentially-usable cores. Further computational resources can be leveraged if multiple disciplinary analyses could be conducted simultaneously as would be the case in surrogate model generation (see Section IV), UQ (see Section V), or even parallel approaches to optimization. It must be mentioned that the majority of efforts to reach exascale in aerospace applications are focused on these methods and approaches: significant amounts of work are being pursued in this domain and the result is likely to be improved disciplinary analyses that can indeed map efficiently onto exascale computers). Complex fluid flow simulations (LES, DNS, DES, ILES, etc) are leading the way.

2. Different scalability limits for different participating disciplines: The computational effort required to solve each of the participating disciplines in an MDA can vary dramatically and, therefore, the appropriate number of cores for a given discipline may exceed the number appropriate for another. This is typically the case since the mesh requirements and the arithmetic intensity of participating disciplines vary widely. For example, a CFD solution may be much more computationally expensive than a linear finite element simulation. If a number of processors in an exascale machine are reserved for an MDA and, during certain portions of the computation only a smaller number can be used effectively, the overall parallel efficiency of the calculation may suffer. Fast-switching of parallel contexts may be required to enable such computations. Advanced load-balancing techniques will also be necessary for MDA purposes.

3. Scalability of information exchange between disciplines: Although the exchange of information between disciplines (loads and displacements for the aero-elastic case used as an example workflow in this section) typically only consumes a small amount of computation (relative to the disciplinary analyses themselves) it is not uncommon for the details of the exchange algorithms to be based on distance searches, containment searches (for interpolation across overlapping meshes), and tree-based algorithms that are very hard to scale (in both memory and performance) to very large numbers of cores. With machines that will have limited amounts of memory per core, the challenges to scalability can be quite severe and need to be dealt with effectively. An example of a case in which the exchange between disciplinary analyses was a significant bottleneck for a large-scale parallel analysis are the full-engine simulations conducted at Stanford during the DoE ASCI program and depicted in Figure 1 below. Three separate scalable codes for the fan/compressor (SUmb, URANS, instance 1), the diffuser/combustor (CDP, LES), and the turbine (SUmb, URANS, instance 2) had to be coupled at every iteration through a mesh overlapping interpolation mechanism that was also parallelized with extreme scalability in mind. In large-scale computers such as the IBM Blue Gene/L, which at the time (2007-08) was the largest machine in the top 500 list with over 100,000 processors but with only 16 Mb of RAM for each computational core, the information exchange was the bottleneck for scalability of the entire computation.

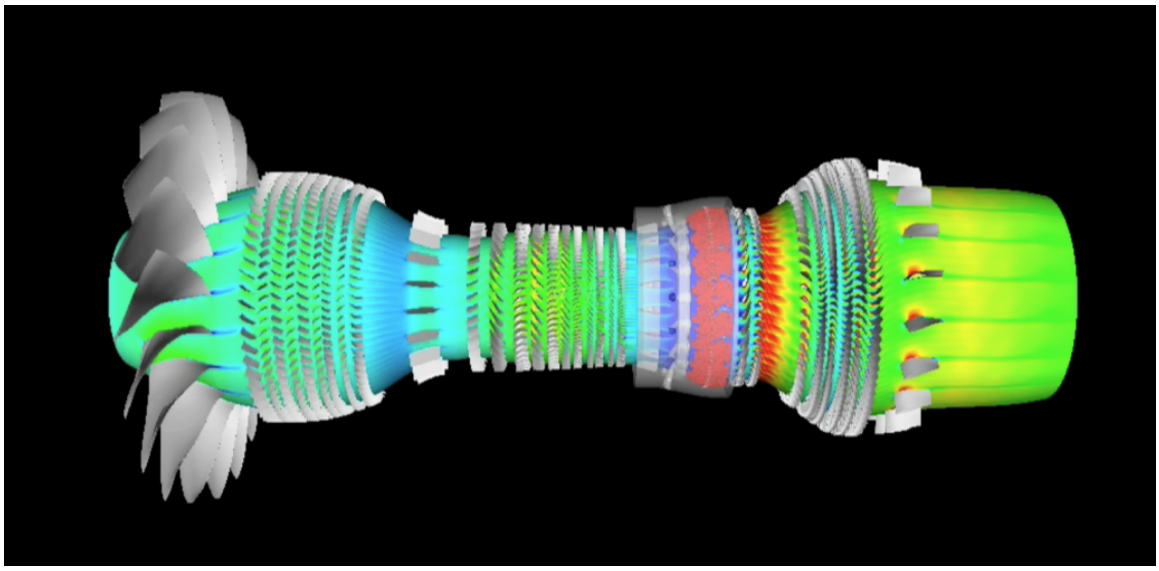

Figure 1: Sample full-engine simulation MDA that required the scalable interaction of three disciplinary codes.

III. Multi-Disciplinary Optimization

In this section, we discuss the challenges and opportunities of Multi-Disciplinary Optimization (MDO) workflows as they are mapped to exascale computers. It is difficult to abstract all of the possible MDO approaches into a single section, as one can consider all types of methods, including surrogate-based ones, under this umbrella. To be clear, in this section we discuss the exascale challenges and opportunities of optimization methodologies that do not use surrogate models in the following three modalities:

1. Gradient-based optimization of an MDA: in this kind of workflow, we assume a non-linear program problem formulation, where the cost function and constraints (and their gradients) are derived from the execution of an MDA. Such approaches are often referred to as *multi-discipline feasible* optimization approaches.

2. Non-gradient-based optimization of an MDA: where we assume that the same MDA is required to analyze the performance of the system (both objective and constraint functions), no gradients are needed, and population-based algorithms are used to evolve from an initial distribution of designs to the final one. Such approaches include genetic algorithms for single- and multi-objective optimization, and are based on the repeated evaluation of the underlying MDA.

3. MDO decomposition schemes allowing for parallel execution of the disciplines in the MDA: significant research efforts have been devoted over the past 15 years to develop MDO algorithms that permit the parallel execution (and possibly optimization) of the individual disciplines participating in the underlying MDA. This decomposition schemes have often been pursued to enable disciplinary autonomy and, thus, map better to existing industrial environments, but also to *permit parallel execution of the disciplinary analyses*. Whereas in the two examples above in the list the unit of work is a fully converged MDA, in the approaches considered under this heading (such as Simultaneous Analysis and Design, SAND, Collaborative Optimization, CO, etc) individual disciplinary analyses become the fundamental unit of computation.

In gradient-based optimization workflows, the execution of an MDA is the fundamental unit of work. For this reason, all of the challenges and opportunities described in Section II apply here as well. But gradient-based optimization workflows also result in a sequence of MDAs as the design space is searched. Typical gradient-based optimization algorithms require the execution of one MDA before the next one is run, thus imposing an inherently sequential nature that prevents parallel execution. It is true that, at every step of the

execution of the algorithm, additional MDAs may need to be run, for example for gradient calculations using either finite differencing or even adjoint methods, and even for line searches if the algorithm requires them. In this situations parallel execution of many simultaneous MDAs can take place. If finite differencing is the method of choice for the calculation of the gradients, as many MDAs as there are design parameters in the problem can be run simultaneously. With typical design optimization runs using 20-1,000 dimensional spaces, the opportunities for parallelism are very significant. But alas, the gradient vector is obtained with very low computational efficiency (order N calculations are needed for an N-dimensional design space). When adjoint methods are used for the computation of the gradient, a smaller degree of parallelism (measured in the number of simultaneous executions of the adjoint of the MDA problem that can be supported) can be found that is equal to the number of functional outputs that need to be differentiated, typically the total number of cost functions and constraints. Finally, during the line search portion of the optimization procedure, multiple MDAs can also be executed simultaneously. When expensive function evaluations are the norm, optimizers choose to only use a handful of MDA solutions for the line search, typically using the current point and two additional evaluations for the fitting of a quadratic function that leads to the choice of the optimal step. In this way, the level of parallelism can be increased by a typical factor of 2 x or 3 x.

Some non-gradient based optimization solution algorithms also operate on a similar principle: the design is evolved sequentially over a sequence of steps (or *generations*) that, in principle, leave very little room for parallelism. The nonlinear simplex method, for example, is one such non-gradient algorithm where the execution of MDAs is essentially sequential. The saving grace of some of these methods, inasmuch as exascale computing is concerned, is that each generation may require the evaluation of the *fitness* of a potentially large (typically 50-250) number of members of the population. When each fitness analysis is a complex MDA, exascale computing resources can be used quite effectively due to the embarrassingly-parallel nature of the process. Such approaches to optimization are often considered to be very expensive for realistic design, but the advent of more powerful computers is changing perceptions and is likely to continue to enhance the penetration of this methods into everyday design practices. This, together with surrogate modeling techniques presented in Section IV are becoming more common users of very large scale computing resources, owing in part to the inherent robustness of the process (when some calculations do not complete or converge, the process does not stall and can often continue to completion successfully), and the ability to successfully explore more complex and multimodal design spaces. It seems plausible that with exascale power readily available, such optimization algorithms may ind much more widespread use in the aerospace community in the future.

Up until now, the use of exascale computing in MDO has been restricted to the inherent parallelism that can be found within a single MDA, and embarrassingly-parallel options that allow for the simultaneous execution of multiple MDAs, as many as permitted by the specifics of the optimization algorithm being used. This situation begs the question of whether the MDO algorithm/process can be tailored to enable more parallel execution of the various steps required. Furthermore, in large-dimensional and highly-constrained optimization problems, is it possible to parallelize the optimization algorithm itself? This part of this Section deals with these kinds of parallelization opportunities.

Although not with exascale computing as a motivating factor, MDO hierarchical decomposition algorithms that permit the parallel execution of individual disciplinary analyses and, in some cases, disciplinary optimizations, have been developed over the years. The prime motivation for this class of algorithms could be found in the ability to execute disciplinary analysis in parallel, thus shortening the time to completion, and in the ability for various disciplines to retain control of their sub-discipline optimizations (rather than simply becoming analysis engines directed by a system-level optimizer). As far as exascale computing is concerned, however, these hierarchical decomposition algorithms enable new opportunities to further scale the computational requirements of an MDO problem. In Simultaneous Analysis and Design, for example, the addition of inter-disciplinary compatibility constraints does away with the sequential nature of the execution of disciplinary analyses within and MDA and permits multiple disciplinary analyses to proceed concurrently. When the disciplinary analyses are of sufficient high-fidelity, this can result in a very significant increase in the number of cores that can be used efficiently at any one time. In our aero-elastic analysis example, both the aero and structural disciplines may be evaluated concurrently, using more computational resources than would otherwise have been possible.

Moreover, if the MDO decomposition schemes permit the use of both gradient- and non-gradient-based optimization algorithms (or even the surrogate models that will be discussed in Section IV) the additional parallelism opportunities descried earlier also become available. In such approaches, parallelism at three

separate levels emerges:

1. At the level of the individual discipline analysis: typically corresponding to the use of multiple cores (as many as permitted by the efficient scaling of the solution procedure) and with parallelization approaches that rely on MPI, OpenMP, or a combination of both,

2. At the level of the multiple disciplines involved in the MDO process which now may be allowed to be computed simultaneously, and

3. At the level of the individual discipline optimizations which, through the computation of gradients or via population-based algorithms can result in additional parallelism opportunities.

In all of these situations, the opportunities for parallelism are quite significant and enabled by the innovative algorithms that the MDO community has been pursuing for years. Note that the granularity of this parallelism (the number of cores whose use is tightly coupled in any of the many analyses that can proceed concurrently) is rather coarse. This point must be made clear: exascale computations for MDO are likely to require a large machine for many concurrent simulations, rather than requiring a large machine that is completely used for each of the component analyses. For this reason, and for MDO purposes, one may view an exascale computer as a machine made up of many large clusters of nodes/cores, each of which is connected with a high-performance network interconnect, but with a low-performance network between such clusters. In this sense, exascale power can be made available to MDO workflows with significantly lower investments than for scientific discovery workflows.

It is worth mentioning one-shot methods.

IV. Surrogate Model Development

Surrogate modeling has a long history of use as an approximation methodology for analyses of various kinds. For example, a classic approach to approximating experimental data is least-squares polynomial regression. In the context of optimization, using polynomial regression to generate a surrogate model for optimization was often called Response Surface Modeling, or the Response Surface Method. Polynomial regression is part of a larger classification of parametric models, which have a fixed number of parameters that must be chosen, typically by minimizing the squared error of the model to the data. These techniques have a beneficial property of smoothing noisy data, but at the sacrifice of making strong assumptions about the shape of the data itself. This can prevent the model from identifying salient features, such as multiple local minima that can appear in complex design problems.

A significant transition, from parametric to non-parametric probabilistic modeling (such as Kriging and Gaussian Process Regression, GPR), occurred when Sacks identified the usefulness of uncertainty information to refine response surfaces of expensive simulations. Non-parametric methods make very general assumptions about the behavior of the fit, but effectively use the experimental data as parameters, yielding a valuable amount of generality when modeling new problems. Probabilistic regression methods can also provide an estimated variance, or measure of uncertainty, as a function of the inputs. Such information can be used to identify areas of the model that are inaccurate and could be re-sampled for improved surrogate models.

An extraordinary number of useful parametric and non-parametric regression methods exist. Major approaches include polynomial regression of various orders, Fourier-based fits, splines, neural networks, radial basis functions, and Kriging. All of these methods have in common that the MDA in question (or the component disciplinary analyses) needs to be sampled a number of times in order to obtain the fit (non-parametric methods) or the coefficients of the fit (parametric methods). Lately, our research group has also been leveraging various versions of Gaussian Process Regression, GPR. GPR is a probabilistic predictor derived using Bayesian statistics to condition a spatially varying mean and variance based on the training data. This posterior mean is used as an approximation of the data, and the posterior variance as an approximation of the uncertainty of the fit at a particular location. A nearly identical formulation can be arrived from a frequentist view by minimizing the square error of a predictor given spatially correlated samples, in a derivation proposed by Matheron which he called Kriging [66], named after his mentor.

Just as standard surrogate models require the repeated evaluation of the MDA of interest, so does GPR. However, GPRs are usually created in two steps. In a first step, a number of samples (parallel runs of the MDA) are created simultaneously. These can all be executed in parallel, and their number is likely to be high, leading to potential uses of exascale computing. Once a baseline GPR surrogate has been created, it is

often the case (whether for optimization of for other purposes) that the accuracy of the surrogate is improved through an iterative process of adaptive sampling, whereby additional samples of the MDA are executed to decrease the uncertainty of the fit in areas of the space that require it. Such adaptive sampling techniques for GPR refinement can take place one at a time, introducing a sequential bottleneck. Techniques such as expected improvement are often leverage to decide where to sample next. Of course, there is no inherent requirement for new samples to be drawn one at a time: the expected improvement technique, for example, can be used to determine the location (in design space) of the next k samples that would most improve the quality of the fit, thus enabling k-fold parallelism that could be exploited with an exascale computer.

As an approximation of an expensive function, surrogate models can be interrogated by an optimizer in what is known as Surrogate-Based Optimization (SBO) as a way to minimize expensive evaluations. Studies using surrogate-based optimization approaches grew substantially in the 1990s, especially in the area of Multi-Disciplinary Optimization of NASAs High Speed Civil Transport. This resulted in a Surrogate Modeling Framework formalizing the approach. Additional work was done in the 2000s in this area under the context of aircraft. Approaches for increasing the accuracy of surrogates in higher dimensions by using gradient information to enhance surrogate models of various kinds have been explored. Multiple fidelity information has also been used to enhance the accuracy of the fit while displacing the need for expensive simulations to numerous low-fidelity simulations.

All of these approaches result in further opportunities for parallelism that would allow surrogate-based models and surrogate-based optimization to leverage some very large computational capabilities. Whether it be the computation of gradients, the creation and update of multiple surrogate models for various functions of interest, or even the parallelism opportunities already explored Section III, the opportunities abound and can be fairly self explanatory.

V. Uncertainty Quantification and Design Under Uncertainty

The rapid advancement of both computer hardware and physical simulation capabilities has revolutionized science and engineering, placing computational simulation on an equal footing with theoretical analysis and physical experimentation. This rapidly-increasing reliance on the predictive capabilities of computational models has created the need for rigorous quantification of the effect that all types of uncertainties have on these predictions, in order to inform critical decisions based on quantified variability and risk. The field of uncertainty quantification (UQ) is also rapidly evolving, as researchers seek to address significant challenges in deploying UQ to ever more complex physical systems, while exploiting new advances in computing. With an effective foundation laid over the past 10–15 years in the constituent areas of uncertainty characterization, sensitivity analysis, error estimation, verification and validation, uncertainty propagation, and data assimilation and inference, new frontiers for UQ can now be pursued. Central to the success of our community in this area is the development of scalable UQ algorithmic techniques leveraging exascale computing architectures that allow predictions to be made with confidence in critical areas of engineering and science, including advanced new aircraft / spacecraft systems, renewable energy, climate evolution, and many issues of national security. Such endeavors, due to their computational intensity, are very likely to result in very effective use of exascale computing platforms, not just for the UQ processes themselves, but also for optimization under uncertainty approaches that wrap optimization frameworks around the basic UQ analyses.

Within this context, and using previously-mentioned foundational capabilities it is entirely possible to target a computational spectrum of solutions for UQ at scale, ranging from resource-constrained UQ with high-fidelity exascale simulations at one end, to exascale UQ using massive concurrency in reduced-fidelity simulations at the other. Typical applications will fall between these extremes, will be able to leverage models of varying fidelity, and will require a flexible architecture to balance UQ concurrency with variable levels of simulation parallelism, as well as provide the opportunity for concurrent execution of multiple balance points across the spectrum corresponding to multiple levels of fidelity. In general, ongoing efforts seek to make the most efficient use of limited simulations and are attempting to develop new approaches that leverage emerging architecture-aware solver technology to develop asynchronous, distributed UQ algorithms that are both latency tolerant and resilient.

More specifically, in the aerospace domain, advanced vehicle design must aggressively balance competing requirements of performance, durability, and life-cycle cost, while accounting for all sources of uncertainty present. Such problems are inherently multi-disciplinary and design performance is extracted from all par-

ticipating disciplines by (a) better understanding the individual disciplines through the use of high-fidelity tools, (b) reducing conservative safety margins through a strengthened technical basis, and (c) the use of more advanced technologies that carry intrinsic risks resulting from the lack of prior experience in new aerospace systems. Even though the proper formulation of such design problems is stochastic in nature, the aerospace industry is still focused on solving them using the same deterministic approaches that have been utilized for many years.

Modern design optimization problems, at their core, must properly account for the various sources of uncertainty and quantify their impact on the performance of the designed system. Such formulations are referred to as Design Under Uncertainty (DUU) problems. From a fundamental point of view, DUU can be represented as a traditional non-linear program (NLP), where elements of the objective and constraint functions include moments (means and variances, typically), tail probabilities, or other metrics related to the output distributions of the basic multidisciplinary analysis. An underlying UQ procedure is responsible for providing these statistics and, possibly, their gradients, and standard optimizers can be used to solve the resulting problem. In addition to these straightfoward formulations, fundamental advances in UQ and the formulation of DUU problems are being pursued, so that we can enable the solution of DUU problems with sufficient dimensionality to satisfy current and future industrial interest. We characterize these DUU problems by the following common elements:

- A significant number of aleatory uncertainties that are irreducible. The effect of this inherent variability is observed on the objective and constraint functions of interest via propagation through a set of coupled disciplinary analyses of high fidelity.
- The presence of lack of knowledge, manifesting itself either parametrically or through model form uncertainties, that casts doubt on the assessments of both expected performance and rare-event behavior. This epistemic uncertainty is reducible through the prudent use of external data (e.g., from measurements) or higher-fidelity reference simulations (e.g., DNS).
- The dimensionality of the design and UQ problems can be extensive, including cases where the presence of spatial random fields or temporal stochastic processes results in thousands to millions of additional dimensions. In these cases, correlations in space/time and lower-dimensional structures in general must be identified to break the curse of dimensionality.
- Inter- / multi-disciplinary interactions that compound the effects of both the aleatory and model-form uncertainty components.

An example of a representative DUU problem could be considered as follows. Consider the formulation and solution of DUU problems for the robust performance and reliability of nozzles / propulsion systems of advanced aircraft. Success in designing systems with such aero-structural-thermal interactions that are more robust and reliable can open the door for future vehicles of interest to the aerospace community. The problem of interest being described here, which has key elements of UQ and DUU methods is shown in Figure 2. In the figure, a representative nozzle shape for advanced vehicles, such as the Northrop Grumman UCAS X-47B, is shown alongside the critical analyses required to characterize the performance of such a system. Nozzles of this type must be buried well within the vehicle, leading to constraints on overall shape and size; must be lightweight, while sustaining pressure, shear, and thermal loads; must be tailored at the exit plane to be properly integrated with the aft shape of the aircraft configuration; and must ensure that it can withstand, without cracking, the expected number of cycles in the life of the system.

Additionally, nozzles for variable-cycle engines must accommodate the *confluence of multiple air streams* that can only be predicted with very high-fidelity tools. Multiple streams, such as those envisioned in future engines, can greatly amplify modeling uncertainties.

The design of such a nozzle can then be formulated through a number of different DUU problems where the weight of the nozzle is minimized subject to performance (aerodynamics), loads (thermal and pressure/shear), fatigue/cracking, and geometrical constraints (aft shape, S-bends, etc). Of relevance to the proper formulation is a detailed understanding of the uncertainties present in the problem, their dimensionality, and a careful process of triage so that a DUU parameterization can be created that incorporates the most critical deterministic and stochastic variables.

The opportunities for parallelism, at the exascale level are endless:

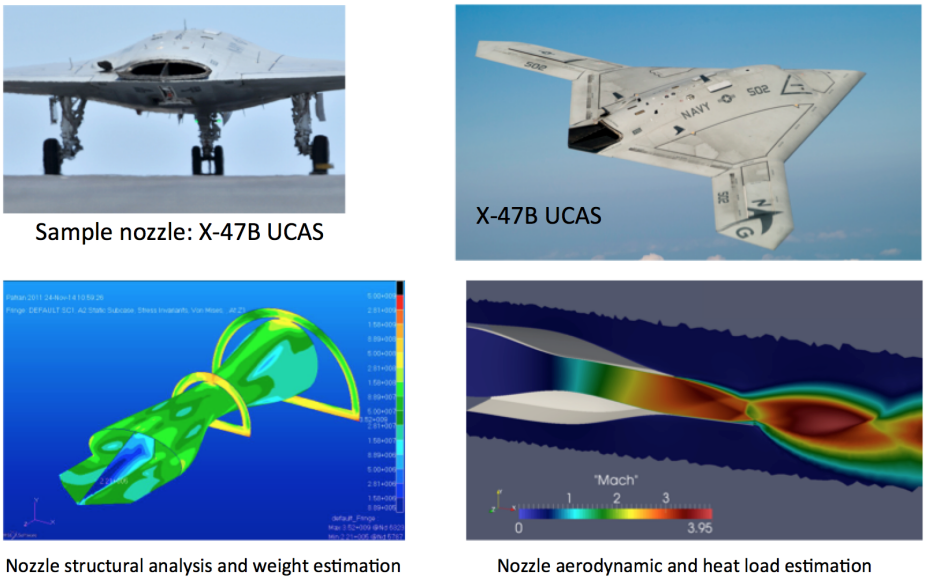

Figure 2: Sample nozzle calculation including aerodynamic, heat transfer and structural loads / weight estimation inspired by the X-47B UCAS

1. Let's begin with the computation of *aleatory* uncertainties. In order to understand the propagation of input uncertainties (to a given MDA of choice) to output uncertainties, a number of approaches can be used. Most common in the literature are methods based on Monte Carlo sampling that require the repeated evaluation of the functions of interest (the MDA) to allow for the extraction of moments (means, variances, etc) of the output quantities of interest. More sophisticated methods, such as polynomial chaos and/or stochastic collocation approaches can also be used to drastically reduce the computational requirements of such aleatory UQ procedures. Regardless of the method employed, sampling of the MDA at multiple points in the design space is needed. Such sampling (even in the case of the use of sparse grid formulations) can greatly improve the level of parallelism found in typical aerospace applications and easily scale to very large resources.

2. The computation of *epistemic* uncertainties also lends itself to the use of vast exascale computing resources as the only way to reduce this kind of uncertainty is by using better (and typically more expensive) models of the phenomena we are interested in. Imagine the full (wall resolved) Large-Eddy Simulation (LES) of an entire aircraft configuration. According to the consensus of the NASA Vision 2030 committee, that the author was a part of, such simulations will not yet be common practice in the year 2030. Even simplified approaches such as wall-modeled LES and/or Detached-Eddy Simulation (DES), while doable, are likely to be very expensive. There is no doubt in the author's mind that reducing epistemic uncertainty sources will lead to very effective use of exascale computing in aerospace-relevant applications.

3. At the simplest level, the formulation of DUU problems that include *both* aleatory and epistemic uncertainty sources can be viewed as a UQ problem with the ingredients of the first two bullets, turned into an iterative process that uses an optimizer to query the result of UQ processes. Since the opportunities for parallelism are pervasive in UQ by itself, DUU further enhances these opportunities in the many ways that have been discussed in Sections III and IV. The options are truly endless if the infrastructure is developed to take advantage of them.

VI. Conclusions

This paper has presented a personal view of the author's opinions on how future exascale computers might be effectively leveraged for the many component elements of MDAO. Specific thoughts regarding multi-

disciplinary analyses, multi-disciplinary analysis and optimization, the construction of surrogate models, and both uncertainty formulations and problems of design under uncertainty have been expressed. Opportunities for "many"-fold parallelism, at the level of repetition of disciplinary or multi-disciplinary analyses abound: from multiple sampling in surrogate models and UQ, to the computation of gradients, to population-based non-gradient optimization approaches, it is easy to see how relevant problems could effectively utilize exascale computing resources. The requirements (particularly for high-performance communication and data transfer) of future exascale computers derived from this type of computation / parallelism may be less severe than for situations where the entire exascale computer is needed for a single simulation. Furthermore, when attempting to do MDAO and / or DUU, the use of optimization processes, even when inherently sequential such as in gradient-based approaches, brings about additional exascale opportunities. The combination of both showcases even more compelling needs for exascale computers. In general, the author believes that the objective should not be how to fully utilize an exascale computer with a particular aerospace problem of interest but, rather, what new engineering processes can be enabled by the existence (and application) of exascale computing.

VII. Acknowledgments

The author would like to thank the following individuals and groups for their useful comments over the years that have shaped some of the impressions expresses in this paper: Dr. Michael Eldred (Sandia National Laboratories) for his input on UQ and DUU, particularly at exascale; Dr. Trent Lukaczyk, for his work in non-parametric (GPR) surrogate modeling and surrogate-based optimization: Profs. Pat Hanrahan and Alex Aiken of the Stanford CS department, for sharing their knowledge of advanced (exascale) computing architectures and programming models; and the members of the Aerospace Design Lab at Stanford University and the Intel Corporation (including the SU2 and the Stanford IPCC teams) for their commitment to efficiency and scalability of simulations of aerospace interest with very large numbers of procesors.