White Paper:
Science at Extreme Scales: Where Big Data Meets Large-Scale Computing
IPAM Long Program, Fall 2018

Author list (in alphabetical order)

Chris Anderson, Samuel Araki, Petter Bjørstad, Hans-Joachim Bungartz, Keiko Dow, Claudia Draxl, Marco Duarte, Ionut-Gabriel Farcas, Longfei Gao, Jochen Garcke, Pietro Grandinetti, Philipp Haehnel, Mojtaba Haghighatlari, Jeffrey Hittinger, René Jäkel, Frank Jenko, David Kim, Robert Martin, Erin Molloy, Saerom Park, Matthias Scheffler, Jannik Schürg, Eder Sousa, Justin Sunu, Chee Wei Tan

Contents

Executive Summary

Computing has revolutionized science: simulation, or model-based computing, has allowed us to investigate phenomena much more complex than theoretical analysis alone can access, and now data-based computing is allowing us to more effectively explore, understand, and use data resulting from experiments, observation, and simulation. The simulation community has for long driven High Performance Computing (HPC), and the data science community has driven the Big Data (BD) revolution. These two computing approaches have usually been addressed independently, but the need for HPC in data-based computing and the overwhelming availability of data in model-based computing indicates that the *integration of HPC and BD* can form even stronger links between theory and experiment or observation, a bridge enabling the fusion of the two communities' methodologies. By hosting this Long Program, IPAM took a lead in fostering fruitful conversations across a range of scientific disciplines, allowing the mathematical and neighboring sciences communities to consider this topic of groundbreaking potential both in a deeper and a broader manner.

This convergence comes at a critical time. We are entering a period where we have the capability to transition from interpretive qualitative models to truly predictive quantitative models of complex systems through computing. However, to realize this goal, we must deal with increasing complexity in models, algorithms, software, and hardware. The exponential growth in computing capability driven by Moore's law is stalling, and HPC hardware architectures are necessarily becoming more complex. Accordingly, to implement the more complex models as well as to make good use of these new architectures, algorithms are becoming more complex, too. The fusion of theory- and data-based computing concepts will help tame this complexity and access more of the available computing power.

The new computational science, which will advance by leveraging the best of simulation and data science, is not, however, a foregone conclusion. Much work remains to be done. To further advance scientific understanding through data analysis techniques such as Machine Learning (ML), methods must be devised to exploit domain knowledge and to enable the interpretation of ML-derived models. Information integration is a paramount step towards a predictive science, but existing approaches have limitations that become evident in large-scale applications. The logistics of data management are not to be overlooked as an area in need of advances; data must be findable, accessible, interoperable, and re-usable to enable this theory- and data-driven future. Data and information are not the same thing, and we must make judicious use of all forms of data reduction techniques to preserve the information content efficiently. Finally, both the changing hardware landscape and the increased availability and use of data will require the development of new algorithms. In this report, we summarize the ideas that came up during at the many discussions during the IPAM Long Program, describing in more detail these topics and the outlook for a new computational science paradigm.

1. Introduction and Background

The breathtaking progress in both computer technologies and advanced methods to effectively and efficiently exploit them opens the door for a "new kind of science" at the beginning of the 21st century. This paradigm change has been brought about by two waves of innovations. The first wave primarily focused on HPC. Simulations and optimizations enable breakthroughs in the fundamental understanding or improvement of processes and systems in various domains of science and engineering. The second wave, which started later, aims at the comprehensive modeling of natural science, engineering, and societal phenomena in a data-driven way, with an even broader impact, also reaching the social sciences and humanities. Datasets from observations, experiments, simulations, imaging, digitization, or social networks as well as business or patient data are collected, processed, and analyzed.

Currently, the themes of HPC and BD – including, in particular, data engineering and data analytics – are usually addressed rather separately. The interactions between the respective communities in scientific computing, statistics, machine learning, and data engineering are rather limited. However, it has become increasingly clear that HPC and BD are intrinsically linked. E.g., HPC needs input data of various kinds for simulations and produces output data to be analyzed, while BD needs HPC for managing and analyzing large datasets to extract highly complex, non-parametric models for inference. Data from simulations can be used to better design experiments, while data from experiments can provide better or better fitted models. Most importantly, only together can they pave the road towards a "predictive science" that has been the goal since the first days of simulation and is crucial for any reliable computational support of design and decision-making. This aim is at the very heart of the present Long Program: helping to advance the state-of-the-art in computational technology, i.e., HPC *and* BD, in a holistic and synergetic way in order to facilitate the creation of new paths to insight and prediction in various scientific domains.

In HPC, as both hardware and software are approaching the exascale, one of the key challenges is to overcome the communication bottleneck. Data motion tends to clearly limit the performance and determine the enormous energy consumption of future supercomputers; some even say "flops are for free." Thus, it is crucial to develop novel ways of efficiently representing, reducing, reconstructing, transferring, and maybe even avoiding huge amounts of data, including the design of new algorithms. Meanwhile, the analysis of very large datasets – originating from some kind of simulation, measurement, or tracking – requires increasingly sophisticated data analytics, which turns ever more compute-intense itself and thus becomes a major customer for HPC. In this sense, HPC and BD technologies are intrinsically linked, and the latest insights, methods, and algorithms have to be considered jointly in this context.

The fusion of HPC and BD is a new, emerging field of research with an endless number of applications and an enormous game changer potential. The present Long Program was aimed at being a catalyst at this frontier of science by bringing together leading innovators and pioneers from applied mathematics (scientific computing, optimization, data analytics, statistics etc.), computer science (HPC, data engineering, data analytics, visualization, imaging etc.), and various applications areas, allowing to expect a particularly high degree of cross-fertilization and breadth of impact.

The timeliness and importance of the present theme — the "convergence" of BD and HPC — is highlighted, among other things, by the fact that it has been identified as critical by the science agencies of the G-8 countries who have supported a series of workshops (see: www.exascale.org) since 2013 to examine the prospects of such a convergence. It is critical because the number of computer architectures that will be taken to the exascale is very limited, as are the resources to develop them. The feasibility of one architectural design to support the applications and workflows of both communities would greatly accelerate the path to computationally enabled scientific discovery for each. The workshop participants have identified conferences and workshops that foster a unification of the computationally intensive and data intensive research communities as a priority that could pay large dividends.

The communities are currently compartmentalized and do not easily recognize each other's vocabulary or scientific aesthetics; consider, e.g., physicists who deal with the data coming off the large hadron collider and physicists who solve the partial differential equations of lattice gauge theory in an attempt to understand precisely the same particles. Their software and the stresses they place on hardware are so different that separate computing facilities have grown around each, but this is not a scalable research model in a time of limited resources and talent. The same could be said for climate modelers integrating global observations and those running global simulators. Similar things are true is plasma physics, materials science, and many other domain sciences. Applied mathematicians and statisticians bring as vital a set of voices to this dialogue as computer scientists, yet they have not been engaged as much as

computer scientists to date. By hosting the present Long Program, IPAM took a lead in fostering fruitful conversations across a particularly wide range of scientific disciplines, pulling into the view of the mathematical sciences community a topic of groundbreaking potential.

2. Trends on the Path toward a Predictive Science

Science is fundamentally a process for understanding and shaping the world around us, ultimately with the goal of using this knowledge for reliable predictions to the benefit of mankind. The scientific process has been revolutionized through computing: simulation, or model-based computing, has allowed us to investigate phenomena much more complex than theoretical analysis alone allows, and now data-based computing is allowing us to more effectively explore, understand, and use data resulting from experiments, observation, and simulation. We now are entering a period where we have the capability to transition from interpretive qualitative models to truly predictive quantitative models of complex systems through computing. However, to realize this goal, we must deal with increasing complexity in simulations and hardware. Addressing this complexity will require a broader definition of computational science (and practitioners thereof) that embraces a more holistic approach to simulation, data management, and data analytics.

The Complexity of Hardware

The success of model-based computing has been driven significantly by advances in hardware for HPC. However, the hardware environment is entering a period of significant change and variation. Physical limitations have stalled increases in processor clock speed (the end of Dennard scaling) and quantum limits will soon limit further reductions in transistor size (the end of Moore's law). Traditional memory technologies (both capacity and bandwidth) are improving more slowly than microprocessor capabilities, shifting processing costs from operations to data motion and latency. The computer hardware industry has responded to these challenges by moving to increasing heterogeneity: multiple processors with different characteristics (e.g., GPUs and CPUs) and deeper memory hierarchies with high-latency non-volatile technology augmenting traditional low-latency volatile memory. Research is ongoing into new materials and structures for microelectronics to replace traditional CMOS technology, such as TFETs and spintronics, and again, these technologies will have different characteristics (e.g., clock speed) than traditional MOSFET technologies. New transistor technology is at least a decade away, so in the short term, some manufacturers are generalizing the concept of processors to processing packages, a combination of general-purpose CPU, specialized processors like GPUs, and closely integrated memory. It is likely that these packages will become even more diverse, including other technologies like FPGAs, phase change memory, processor-in-memory, and three-dimensional memory technologies. Making efficient use of such heterogeneous hardware will be a significant challenge. In the longer term, quantum and neuromorphic processors, which have the promise to revolutionize certain specialized calculations, may also appear as additional co-processors in HPC machines. It is likely that computational workflows will increasingly use stream (or dataflow) processing. Cloud computing, though high latency, provides additional specialized resources that could be confederated with traditional HPC resources and specialized database hardware to enable more complex simulation workflows.

The Complexity of Algorithms

The changes in computing hardware pose a challenge but also a tremendous opportunity. To leverage these capabilities, software and algorithms will necessarily become more complex. Mapping algorithms to specific hardware where they perform best will be necessary. Means to reduce synchronization and data motion will be needed to make good use of the large number of processing units while combating bandwidth and capacity limitations. More dynamic runtime systems, which identify and execute

concurrent operations, may play a more important role to achieve good machine utilization. In addition, one can envision combining model-driven and data-driven capabilities to more efficiently, accurately, and/or rapidly solve very challenging scientific problems. Simulations could be augmented by data-driven models of constitutive of "sub-grid" physics. Taking this a step further, ML techniques are already showing promise in scale-bridging, allowing detailed coupling between micro- and macro-scale physics. ML models may also be useful in dynamically controlling simulation algorithms, partitionings, preconditioners, etc. Conversely, simulation and theory may be useful in guiding learning techniques, augmenting datasets, and constraining training to be consistent with physical principles.

Computational Science 2.0

It should be clear from these observations that the future of computational science includes both data-driven and model-driven techniques, not as separate endeavors, but integrated to realize a more predictive science. Practitioners of this evolution in computational science must have at least an "appreciating awareness" of simulation, data analytics, data management, and computing architectures. This future therefore calls for more interdisciplinary training and more interdisciplinary collaboration, as it will be difficult to master all topics in great depth.

3. Scientific Understanding and Machine Learning

An exciting and relatively recent development is the uptake of ML in the sciences, where one aims for scientific discovery from data. Success of this so-called "Scientific ML" needs the integration of domain knowledge into ML models and coherent workflows, and the interpretation of the obtained ML models, which are actually two sides of the same coin.

Exploiting Domain Knowledge

Examples of domain knowledge include detailed mathematical descriptions of the particular phenomena, a dataset from simulations of a dynamical system, or the specification of a set of physical or chemical constraints for feasible solutions. The field of Scientific ML aims to incorporate this into existing or new ML models, in particular for the analysis and modeling of causal relationships.

Black-box data models face challenges in application areas that involve complex physical phenomena or lack labelled data. Incorporating knowledge of the underlying causal process can improve the consistency of the data model, e.g., when predicting surface stress coefficients from turbulence flow data, which can be described as the preservation of certain invariants or symmetries. Knowledge incorporation also finds relevance in classification tasks and natural language processing. Moreover, it improves the interpretability of the data model, which will be beneficial for subsequent scientific development, as will be discussed later. E.g., prior knowledge of the underlying physical process can help selecting relevant information to feed to the data model. Afterwards, it can help curating the outputs generated from it, also with the aim of data reduction, as outlined in the corresponding section below. Additionally, domain knowledge, often formalized by mathematical concepts, can be used to enhance the process of construction or validation of the data model.

In the following, we concentrate on approaches that incorporate domain knowledge in the data preparation and modeling workflows which have been discussed explicitly during the Long Program. The majority of the presented research has a focus on the design of Neural Networks (NNs), mainly due to the flexibility in designing deep learning architectures guided by scientific knowledge. For instance, convolutional long-short term memory can be used to incorporate time-varying domain characteristics in diffusion of contaminant in soil and weather pattern. In addition, plasma physics and machine learning were combined in the deep learning architectures for a more reliable prediction of disruptions in magnetically confined

fusion plasmas. In chemistry, incorporating the organic chemical reaction information into the attention mechanism of sequence-to-sequence models provides competitive prediction accuracy. Another important approach to incorporate scientific knowledge in data-driven models is through loss functions and regularization terms in the mathematical formulation of learning problem. For instance, a "physics-based loss function" was introduced to enhance the scientific consistency of predictive models for lake temperature or air pollution.

Domain knowledge has long been used in the data preparation steps of the modeling workflow such as data augmentation, sampling of data, or feature transformation and selection. For example, recent progress in the incorporation of chemical domain knowledge in the design of ML training datasets yields some important insights for the exploration of the vast chemical space. Furthermore, ML models trained on data subgroups make effective use of the sparse materials space. Additionally, in the field of molecular dynamics, image classification and object detection via Deep Neural Networks (DNNs) can be enhanced by removing redundant images corresponding to multiple rotations or illuminations of a given object. This removal can be achieved in real-time during the data generation process. Another example appears in mechanical engineering, where mathematical geometric-aware representations of data stemming from deformed surfaces enables an easier virtual product development process in the automotive industry.

Understanding Machine Learning Models

Recently, there has been a significant effort in trying to better understanding how a specific ML model operates and why certain decisions are produced, particularly in view of the empirical success of deep learning approaches. In the ML community, this is often referred to as interpretability, with varying concepts of what this actually implies, and the underlying hope of explaining the predictions of a model. The need for interpretability arises from several motivations.

In the sciences, one such motivation is scientific understanding, where the goal is to convert ML-derived explanations to knowledge: causal relationships might be inferred from observational data or, alternatively, useful information for scientific purposes can be obtained. Pursuing these motivations allows for an easier dissemination of the potential benefits of ML to the scientific community. Several examples of ML driven by this motivation were discussed in the Long Program. When considering an event controlled by a dynamical system, the data observed can be fed to nonlinear manifold learning algorithms to infer the underlying structure of the system. In phylogeny, correlations between genomes from a variety of animal species can be modeled using tree graph priors to infer evolutionary relationships. In genome-wide association studies, the ML model identifies correlations between genes and disease indicators. More generally, it was observed that a common motivation behind the application of unsupervised learning methods is the need for exploration, explanation, and understanding of the event described by the data.

For more practical purposes, i.e., when using deployed ML models for scientific purposes, one is interested in robust and reliable decisions. The interpretability of an ML model can determine whether the model will work in non-stationary environments, or even when its use might have an effect on the environment, altering the underlying assumption of having data from the same distribution while training and applying the model. For example, when generative adversarial networks modeling inertial confinement fusion are designed to be physically consistent, the underlying data model can predict key physics information, leading to improved generalizability to a variety of simulation datasets.

Another aspect where interpretability can help is the gap between the employed loss function used for model training and the actual quantity of interest in the task. Not all requirements can be framed by a suitable loss function; usually, the loss is a proxy for the actual goal, and in some cases one can consider

the case of multiple objectives. E.g., ML has been applied to functional magnetic resonance imaging (fMRI) data to design biomarkers that are predictive of psychiatric disorders. However, only "surrogate" labels are available (e.g., behavioral scores), and so the biomarkers themselves are also "surrogates" of the optimal descriptors.

Attaining Interpretability

While the need for interpretability is widely accepted, what interpretability actually implies does vary widely. According to the recent literature, there are two ways to confer interpretability.

On the one hand, a model is called *transparent*, if it becomes clear how the model operates. One can define this as follows: a transparent model provides a mapping of an abstract concept (e.g., a predicted class) into a domain that the human can make sense of. For example, the similarities between the multilayer perceptron architecture and the physical model for transmission electron microscopy (TEM) makes the application of the former architecture to the latter scientific domain transparent. Similarly, an application of ML for epidemiology leverages a networked dynamical system model for contagion dynamics, where nodes correspond to subjects with assigned states; thus, most properties of the ML model match the properties of the scientific domain considered.

On the other hand, post-hoc explanations can be made in order to extract information from learned models. Formally, an explanation is the collection of features of the interpretable transparent domain that have contributed for a given example to produce a decision (e.g., classification or regression). Regression, in particular, has been leveraged often in Scientific ML to explain phenomena. Examples include the design of property maps from few physical descriptors in materials discovery, and the formulation of reduced-order models for partial differential equations via lifting.

Often, direct approaches are undertaken to present this information via visualizations of learned representations, natural language representations, or the discussion of examples. Nonetheless, human interaction is still required to interpret this additional information, which has to be derived from the learned model during the post-hoc analysis. For example, in the previously mentioned application in fMRI, the biomarker design promotes spatially compact pixel selections, producing biomarkers for disease prediction that are focused on regions of the brain; these are then considered by expert physicians.

From the perspective of mathematics, a core contribution to the understanding of ML algorithms is their mathematical interpretation and derivation, which help to understand when and how to use these approaches. Classical examples are the Kalman filter or principal component analysis: several mathematical intuitions exists for each of these tools. In addition, numerical approximation and convergence theory are an important ingredient when understanding function approximation, as it arises for regression and classification tasks with for example deep learning algorithms or kernel based methods.

Future Directions

Looking forward, while interpretability can be defined in terms of transparency and explainability, a formal mathematical evaluation of these concepts remains elusive. Heuristics such as requiring neighboring data points to have very similar explanations provide criteria for the form of the explanation, but do not measure the quality of an explanation. In the end, it seems one will need to involve human evaluation of the quality of an explanation; however, the design of reasonable and efficient methods for human involvement in evaluation that avoid human bias is an open research question.

Concerning domain knowledge, one needs to further investigate which combinations of scientific knowledge and simulations will lead to the highest degree of generalization. While the discussion focused on supervised learning problems using manually labeled data, theory-guided data science can also be

applied to unsupervised learning and semi-supervised learning. For the deployment of a ML model, its robustness under changing environments is of utmost importance, also in view of quantifying the uncertainty of ML models. A still ongoing challenge is to develop further mathematical theorems for performance guarantees, and to evaluate and classify ML models through mathematical terms, such as topological or geometrical quantities.

The aforementioned aspects of Scientific ML (will continue to) involve a combination of domain science and ML expertise, where mathematics helps to build a bridge between the domains. We envision that in the future, theory and domain knowledge should guide ML towards a modular learning process, where the inclusion of human expert understanding of the domain science will be a module of the learning system that allows for knowledge discovery from data.

4. Information Integration

In order to gain insight into complex real-world systems, scientists perform physical and computational experiments, generating massive amounts of data. Extracting scientific knowledge from these datasets can be challenging for many reasons, including well-known limitations of experimental data. For example, in a physical experiment, measurements are collected using physical devices (e.g., sensors), resulting in noise or other sources of measurement error. Furthermore, it may not be possible to directly measure the quantity of interest in a physical experiment, and thus, the quantity may need to be observed indirectly by measuring a related quantity. Similarly, in a computational experiment, data are simulated using a mathematical model that may not fully characterize the complex dynamics of the system; thus, the resulting simulated data may fail to capture important physical properties of the system. In order to overcome the limitations in using a single information source, we need a *coherent framework for integrating information coming from multiple sources*. In this section, we describe several promising approaches for information integration and their potential for a predictive science.

Bayesian Inference

An important step towards predictive numerical simulations involves the fusion of numerical models with experimental data. To this end, Bayesian inference provides a natural framework to combine models with experimental data, because the solution of a Bayesian problem provides a quantification of the uncertainty stemming from measurement or other error sources. The inherent uncertainty associated to studying these systems necessitates a departure from the classical deterministic realm of modeling and simulation. Consequently, our main building blocks can no longer be deterministic, but instead we must operate within the context of probabilistic models. Bayesian inference provides a systematic framework to understand complex physical models in the context of the observed data. In the resulting Bayesian inverse problem, model parameters are inferred using the observed data and the underlying mathematical model, and the solution is a probability density function which characterizes the uncertainty in the model's input parameters given the observational data. By reducing this uncertainty in model parameters, we can begin to understand the underlying physical system described by the data.

Transfer Learning

Transfer Learning (TL) has emerged as a new learning technique for improving the learning result of one problem by integrating the learned knowledge from another related problem. In a broad sense, theory-guided data science is a good example of TL, because it enables knowledge transfer from different data types which are closely correlated. In TL, this correlation is important in order to avoid degraded performance of learning (called negative learning). In addition, introducing a deep learning framework

for scientific data analysis can seamlessly synthesize structured prior knowledge (e.g., differential equations, conservation laws, etc.) as well as heterogeneous data (e.g., model predictions at multiple scales/resolutions, images, time series, scattered measurements, etc.). TL, with the help of Bayesian inference, allows us to fill in the gaps that individual heterogeneous data sources present, and to combine data and scientific knowledge into a unified predictive model for science.

Uncertainty Quantification

Uncertainty Quantification (UQ) is a systematic approach to quantifying and reducing uncertainties in numerical simulations. These uncertainties can arise due to incomplete knowledge or intrinsic variability in the physical system. UQ is an example of an outer-loop scenario, in which a mathematical model (or another information source) is evaluated multiple times in order to assess statistical quantities of interest (e.g., mean values, standard deviations, etc.). Although UQ is paramount for predictive science, it can be computationally expensive and thus infeasible for complex systems. Advancements in hardware and computational resources are not sufficient to enable the application of techniques from UQ on complex physical systems. We also need advancements in algorithms, e.g., so to exploit the underlying problem structure in order to reduce computation and storage requirements. A recently developed UQ technique that can bridge the gap is the multi-fidelity methodology.

Multi-Fidelity Methods

The extensive body of research in model reduction or ML provides ways to obtain so-called low-fidelity models, i.e., information sources that are less accurate than the high-fidelity model, but that require less computational effort. However, reduced or ML models are traditionally designed to replace the high-fidelity model. Note that these models are generally obtained using high-fidelity model evaluations (e.g., interpolation). Thus, the standard approach of UQ in numerical simulations entails using only one model (or information source). Particularly in applications which require high-performance computers for a single simulation, creating accurate low-fidelity models can still be computationally too expensive. In addition many reduced or ML low-fidelity models of different accuracies are typically readily available in most applications.

This calls for a paradigm shift in addressing UQ, particularly in large-scale simulations. One of the most promising new approaches is the so-called multi-fidelity method. Instead of replacing the high-fidelity model with a low-fidelity information source, the multi-fidelity framework brings together high- and low-fidelity models such that the cheaper, less accurate models are leveraged for computational speedup while the computationally expensive model is kept in the loop to guarantee that accuracy and converge are preserved.

To guarantee that accuracy and convergence are preserved, and to distribute work among the underlying information sources, multi-fidelity methods employ a model management strategy such that the work division is proportional to the computational cost of the underlying models. It is important to note that not all low-fidelity models need to be accurate for the method to work well, but rather the interaction between the models is what drives the efficiency of the multi-fidelity computation. Therefore, a heterogeneous set of low-fidelity models which interact with each other is more beneficial than having very accurate but disconnected information sources.

Multi-fidelity approaches have been successfully employed in a number of applications, and the current extensive research efforts clearly show the interest in employing multi-fidelity methods in very complex, real-world applications. In addition, besides quantifying uncertainty in complex systems, multi-fidelity methods are also used in optimization or parameter inference. Especially on future exascale computers,

multi-fidelity approaches are among the few promising candidates which could realistically enable UQ in large-scale simulations.

The model management strategy in existing multi-fidelity approaches for UQ does a static work allocation, and it is furthermore assumed that the low-fidelity models are readily available. To extend this framework, current research focuses on transforming the static approach into an adaptive multi-fidelity method in which the low-fidelity models are created dynamically relying on approximation and cost rates. The long-term goal is to have fully adaptive multi-fidelity methods that dynamically create the underlying low-fidelity models by exploring and exploiting the structure of the problem at hand.

Future Directions

Information integration is a critical step on the path toward predictive science. While several promising approaches for information integration exist, their limitations become quickly visible in large-scale applications. Thus, to cope with the ever increasing complexity of these applications, enhancements in many computational and algorithmic aspects are necessary. Moreover, approaches for information integration must be validated in the context of specific applications, and thus, the advancement of such methods depends on scientific communities making large and heterogeneous datasets accessible to method developers.

5. Removing the Showstoppers of Big-Data Driven Science

Presently, science is often (still) characterized by the fact that BD are typically just useful for the research group that created them and maybe a few close collaborators. This is because data are usually not fully characterized. Taking the example of natural, engineering, or medical sciences, individual researchers and research groups argue (or act under the assumption) that details of sample preparation or the exact procedure of the investigation may give them a competitive advantage and therefore are better not fully revealed. Obviously, this culture has to change if we like to take full advantage of BD to advance science. Equally important is the fact that sharing is complicated because of a missing efficient data infrastructure. Organizations that run the latter in an non-bureaucratic, efficient, and user-driven manner are not yet in place. As noted in a Nature Editorial (June 2017): "Governments, funders, and scientific communities must move beyond lip-service and commit to data-sharing practices and platforms."

FAIR Principles

A change of scientific culture and application of the *FAIR principles* are the key issues, where the latter involve metadata, workflows of the data creation, data curation, and the underlying hardware. In this context, data curation aims at organizing the complete dataset by incorporating information from different data sources and constantly updating the database with new information (knowledge extraction). In FAIR, the *F* stands for *findable*. Keeping research data for at least ten years is now requested by many research organizations, and is useful to get research groups and individual researchers better organized. Avoiding duplication of work, it saves human and computational resources. Clearly, making data findable requires proper data infrastructures, including documentation, search engines, and hardware. The *A* stands for *accessible*. Accessibility has different facets, including the proper hardware that allows for swift access to data and the provision of application programming interfaces (APIs). Extremely useful for getting first insight into data is to provide the data not only in a machine-readable but also in a human-accessible form. The *I* stands for *interoperable*. This is the biggest challenge as we need to consider that data from different sources may be extremely heterogeneous. Consequently, the necessity arises to make this data comparable, which requires bringing them to a common format, having knowledge about the data quality, and being able to rely on a robust formal description of the data (metadata). We also recall that one quantity

may be named differently in different (sub-) communities or one and the same expression may have a different meaning in one or the other area. Thus, "dictionaries" are needed to translate between them. Finally, The *R* stands for *reusable*. It means that we can use data that has been created for some specific question in a different context: in other words, they are repurposable.

Despite the aim of keeping data according to the FAIR principles, communities need to consider the trade-off between communication cost (reading out data from the storage) and recreation. In particular, when data can be reproduced, it might be more efficient to store only the most relevant information (e.g., the input and major output files of a calculation; sample, workflow and information about the setup of an experiment; and alike).

Present Status

In the spirit of open and easily accessible datasets, a flexible management to connect to agile compute architectures is desirable. In the HPC community, data management is still organized merely by individual (sub-)communities or projects, and handled individually. Either data is produced on site (e.g., via simulations), or transferred before computation and stored in a parallel file system. So far, the data management aspect is focusing almost entirely on optimizing parallel access to the distributed storage system and on fast and concurrent access to the storage layer from compute elements. Outside HPC – driven mainly by the desire to use state-of-the-art servers (usually relying on standard internet connectivity rather than fast interconnects like in HPC) by companies to manage their datasets – a complementary paradigm has evolved to execute analytics tasks close to the data where it is located in a purely distributed manner. This was motivated by the fact that large datasets are hard to copy or move to a central computing system and hard to be updated if changes produced by analytics processes need to be incorporated. Nowadays, this simple MapReduce paradigm is applicable only for a limited class of analytics tasks and not flexible enough for iterative and learning scenarios. From the compute perspective, HPC architectures provide a much better environment for in-memory and ML applications. Nonetheless, especially for data-intensive applications, the fact that technology-wise the compute capacity is evolving faster than the ability to store and access data represents a limiting factor. Here, an intelligent data management and access scheme needs to be developed in order to speed up the Input/Output (I/O) phase of analytics applications, especially at scale.

Future Directions

For scalable analytics on HPC systems, the concept of burst buffers and data movers have been introduced to select and move data from the central data storage close to the actual compute element for analysis. This reduces the latency in the I/O stage during computing, but requires manual mapping of selected data subsets to the allocated compute resources and lacks advantages of a full data managements system as described above. This specific way of mapping of data and compute is hard or even impossible to transfer to other analytics scenarios; instead, it needs to be adapted to any given situation, and this burden is nowadays completely on the user side. Here, a curated data and metadata middleware can provide a transparent interface not only to describe and organize the data access, but also to define how data needs to be processed. To be able to describe the analytics workflow, the metadata description could to be extended towards a process description, rather than just describing the content and semantics of the dataset. Another approach might be to compress or reduce datasets, either by filtering techniques or efficient numerical representations, which is another attempt to reduce the data footprint during analysis (see next section). For data mining tasks, the dynamic selection of only parts of the dataset in a stream processing manner seems to be also and interesting attempt to reduce I/O efforts, but requires a detailed metadata description and data curation scheme.

6. Data Reduction

The capability to efficiently process large amounts of data has always been, and will always be, a key computational component of any workflow associated with engineering and scientific discovery. In some instances, the processing requires working with data that has been generated from external sources, e.g., the processing of data associated with the training of NNs. In other instances, such as in computer simulations of plasmas, data generation and data processing are intertwined; the data associated with the initial state of the plasma is generated or captured from experiment, and the simulation consists of repeatedly transforming this data to obtain the data representations of the state of the plasma at later times. In either instance, computations with the data require moving portions of the data into and out of an arithmetic unit where the requisite arithmetic operations are performed. This arithmetic unit can either be a portion of the CPU or, if current trends continue, some form of an attached computational accelerator, such as a GPU. It is a fact that the efficiency, or lack thereof, of processing data is increasingly being dominated by the time and energy expenditure required to just move data to and from these arithmetic units, and much less upon the time and energy required by the arithmetic units to carry out arithmetical operations. This fact, coupled with the fact that the amount of data being processed is growing exponentially, implies that efficient implementation of any data analysis procedure, ML activity, or simulation for engineering or scientific prediction requires reducing the costs arising from data movement. This is a critical aspect of exploiting the peak computational efficiency of today's and future HPC systems.

Present Status

Data reduction has been, and continues to be, one of the most useful ways to reduce the cost of data movement. The idea is to identify and use a representation of the data that is more compact, requiring fewer bits to store the data while preserving the information content in the data that is necessary for the success of a data analysis procedure or scientific simulation. The cost of data movement is reduced, because, quite simply, one is moving less data. Tremendous improvements in efficiency have been obtained through the use of data reduction techniques (codecs) in audio and video transmission. The challenge is to develop data reduction techniques that can be exploited to achieve similar dramatic reductions in the costs associated with data movement in the scientific discovery and prediction workflows.

The development of data reduction techniques is challenging because the reduced representation is problem dependent. Moreover, the use of an alternate representation of the data requires efficient procedures for encoding and decoding raw data; parameters associated with the data reduction process must be determined so that information loss is minimized. Also, there are significant challenges to developing adaptive compression techniques that are suitable for problems with dynamically changing data, such as data arising from real-time sensing, data associated with time dependent simulations, or iterative computational procedures.

There are techniques that already exploit the benefits of data compression. Decompositions using predetermined modes, such as those associated with truncated Fourier or Wavelet transformations, can significantly reduce the data representation. Modal decompositions obtained with principal component analysis and based upon simulation or experimental data can provide even more efficient representations. Taking these ideas a step further, hierarchical construction of low rank approximations of matrix sub-blocks has proven an effective technique for matrix compression or "sparsification." Any numerical discretization (h- or p-refinement) that adapts dynamically to the solution is another form of data reduction that focuses degrees of freedom where they provide the most benefit. For example, sparse grid techniques

where one maintains a prescribed level of accuracy in a discretization while reducing the necessary data to represent a solution.

New lossy floating point compression algorithms like ZFP and SZ are being used to compress simulation data not only for I/O, but also for compressing the solution state in memory during the simulation. Theoretical results are being developed to quantify the associated errors and their effect on stability and convergence. Floating point number representations other than IEEE 754 types, which make better use of a fixed bit budget, are also being devised and explored in the context of simulation.

Future Directions

There remain many problems that need to be addressed and many opportunities to be exploited to further take advantage of the benefits of data reduction.

Alternate floating point representations. With respect to data reduction that is implemented via alternate floating point representations, software tools and analytic results are needed that allow one to reliably assess, either before a computational process starts or as a computational process is being carried out, those portions of process that can tolerate errors introduced by reduced floating point precision. Hardware support for compressed or alternative floating point data representations, such as encoding and decoding in hardware are necessary. Additional efficiency could be obtained using specialized routines or hardware that carry out arithmetic operations on the compressed data directly, instead of decoding the data, performing arithmetic operations, and encoding the results. The benefits of data reduction will be made more widely accessible with the development of compiler support or programming languages that allow the declaration of compression and decompression at higher levels.

Working with reduced data. Improving efficiency by working upon reduced data directly could also be obtained with the identification and construction of high level operators that work directly with a reduced representation. A specific example of this is the construction of approximations of nonlinear differential operators when they are to be directly applied to the reduced basis associated with principle orthogonal decompositions. One of the possible uses of ML techniques is to identify surrogate models of these operators based upon simulation data rather than to determine these operators using classical numerical techniques.

Feature identification is becoming a key component for successful modeling using ML algorithms. There are, of course, techniques for determining features that are application agnostic, but it has been observed that significant improvements in the creation of features will be obtained by inclusion of domain knowledge as mentioned in Section 3.

Theoretical analysis. Mathematical analysis that determines upper and lower bounds for the accuracy that is obtainable when using alternate data representations needs to be developed. This is particularly challenging because standard methods of analysis typically assume infinite precision, and the effects of reduced precision need to be explicitly taken into account. Also needed is a better theoretical characterization of the information content of data as it is used in a specific computational process. A promising approach may be to make use of classical information theory. For example, determining how best to use a specified number of bits to implement (near) optimal data reduction. Perhaps most important are theoretical results and algorithmic techniques that can be used to deduce the stability of complex computations with respect to the errors involved in using reduced data representations.

7. New Algorithms

Emerging computer systems are branching into two distinct directions, heterogeneous and distributed, and both of these require the development or redesign of algorithms to make the best use of their capabilities.

Furthermore, many issues that seemed settled, now warrant broad reconsideration due to the change in hardware. At the same time, scientific communities aim to solve more ambitious problems that were traditionally believed to be too difficult. E.g., many domain sciences aim to solve their problems with DNNs. This optimization was believed to be too high-dimensional and non-convex to be tractable. However, recent successes seem to suggest that there are numerical methods that can solve these problems. A robust algorithm is able to arrive at the correct answer, and having a slower robust algorithm is usually better than a faster non-robust algorithm. Cost can refer to the electricity consumed by a system over the course of running an algorithm or to the time it takes an algorithm to complete.

New Hardware, New Algorithms

In recent years, there have been several areas of algorithm development which were strongly influenced by evolving hardware, including communication avoiding algorithms, asynchronous algorithms, and algorithms for mixed or low precision. Communication avoiding algorithms mitigate the amount of data transferred between memory hierarchy levels, which is the bottleneck of simulations in many applications running on a heterogeneous computer system. In particular, they practice reduction of data volume, elimination of metadata generation, reduction of data exchange frequency, homogenization of data access, data access hiding, and localization of data transfer. These general techniques can be applied to a wide range of algorithms. For simple math operations like matrix multiplication, communication can be reduced by changing the order of equations. There are also algorithms to reduce the number of writes, realizing that writing to memory is more expensive than reading. This technique cannot be applied to cache-oblivious algorithms but has been demonstrated for classical matrix multiplication. For more complex calculations, simple modifications to existing code may not be effective, and a more significant redesign of the algorithm is likely to be required. In some cases, such redesign has led to orders-of-magnitude improvement in computational speed.

Asynchronous algorithms replace the synchronous collective communication in a multi-process numerical scheme with asynchronous communication, in which processes that send data do not wait for the other processes to receive. This has an obvious benefit to the overall computational speed, while these methods can experience difficulty in convergence compared to synchronous methods. Finally, the change in computer architecture has motivated a redesign of algorithms using mixed or low precision without sacrificing much accuracy. Hierarchical, iterative methods where high precision is only achieved near convergence, the application of lower precision preconditioners and adaptive domain decomposition methods where different schemes are applied in different problem domains may serve as examples of directions that warrant more investigation. These trends will place increased demands to reliability and reproducibility of the computational results produced by the next generation of algorithms.

New Problems, New Algorithms

Successes of DNNs over the past decade have resulted in applications to many different problems. However, this push for rapid advancement may have resulted in development by trial-and-error, rather than sound theoretical reasoning. In particular, stochastic gradient descent and its variants, which are the dominant optimization algorithm for DNNs, are not effectively utilized due to their sensitivity to hyperparameters. Fortunately, recent developments in the theoretical understanding of stochastic algorithms provide new insights that explain their behavior, and suggest improvements to existing algorithms. Since DNNs are used in a wide range of communities, theoretical studies will have far-reaching implications.

Challenges related to difficult optimization problems – such as DNN optimization, hyperparameter tuning, and clustering – are believed to stem from the sensitivity of the optimization algorithm to the changes in

its parameters. Recent work suggests that this problem can be addressed by adaptive sampling algorithms exploring the solution space using a probability distribution which changes in the course of the optimization. However, the performance of these algorithms is sensitive to the probability distribution update, so special care is needed when developing new methods. Nonetheless, the difficulties in development may be outweighed by ease of parallelization and by insights from probabilistic interpretation such as measures of robustness.

8. Outlook

One of the main conclusions of the present Long Program was that much can be gained by changing the scientific community's mindset regarding Computational Science.

Thinking differently

An important theme in this context was the *relationship between data, information, and knowledge*. In particular, gathering more data is not an end in itself – the goal is to extract new *knowledge*. It is therefore important to ask the question: How much information is really contained in a given dataset, and what constitutes an adequate minimal representation of this data? This philosophy also applies to simulation data, of course, and one should actually be prepared to *improve efficiency by only using the accuracy required for the problem at hand*, e.g., by reducing the precision or by employing reduced models in the framework of a multi-fidelity approach. The basic idea of numerical analysis to control the error can thus be pushed towards larger – not smaller – errors, potentially relaxing the computational cost significantly. Another key theme was to *make better use of existing infrastructures***.** Experimental data is often not curated, which prevents it from being accessed directly via modern data science methods, and real-world applications running on present supercomputers often only reach a few percent of the theoretical peak performance, pointing to unrealized potential. Finally, instead of maximizing the number of floating point operations per second for monolithic applications, one should keep in mind that the goal is to make scientific discoveries, and this goal may be reached most efficiently, e.g., via *smart workflows combining simulation and AI/ML* that help to bridge different physical models and spatio-temporal scales while at the same time including experimental or observational data. All of these points emphasize that out-of-the-box thinking holds a great deal of promise - way beyond any brute-force approach along conventional paths.

Projects and Initiatives

During the present Long Program, several ideas for specific projects and initiatives emerged which exemplify this change of thinking. These include, but are not limited to:

- **Lossy compression of scientific data.** Preliminary studies have demonstrated that theory-driven models can tolerate substantial lossy floating-point compression, not only of I/O and tabular data, but also of the solution state in memory. Theory has been produced that provides justification that inline compression of the solution state does not alter the algorithmic stability under common assumptions. Application of this idea to a broader class of problems is necessary, and more guidance is needed to determine how much compression is allowable. Work has already begun applying ZFP to data from the gyrokinetic code GENE, which is an interesting application because it is used to calculate chaotic turbulent plasma flows. In this context, it is important to determine how aggressively the compression can applied before significantly altering the statistics of the computed solution.
- **Scientific ML competition.** An important strategic question that arose during the Long Program was how to motivate the ML community to work on Scientific ML problems. The observation was

made that other areas successfully drew attention to their problems by providing datasets and posing a competition. Conveniently, in the US, OSTP's Subcommittee on Machine Learning and Artificial Intelligence is currently collecting publicly available scientific datasets from NIST, DOD, DOE, IARPA, NSF, etc. that can currently leverage AI, ML, and large-scale data analysis capabilities. An effort should be made to work with this data to pose a Scientific ML Data Challenge, which will require defining meaningful metrics for each dataset besides standard ML metrics, including but not limited to preservation of physical properties, interpretability, and reproducibility.

- **Reproducibility and Workflow Mapping.** Scientific advances can only be achieved if the methods used are transparent and the results are reproducible. To improve the scientific process, especially in data-driven and exploratory analysis, not only the data, but also the algorithms and software tools need to be published and made available to the research community – for the sake of reproducibility, but also to identify errors or bias in results. New statistical methods that can establish transparency in algorithms and data usage at various stages of the workflow pipeline are needed. To address various reproducibility requirements, data dredging must be identified at all stages of the workflow in an automatic manner. Reliable and representative datasets should be obtained by data cleansing techniques with stringent statistics rigor. The use of verification software and data analysis under human intervention, e.g., data scientists, requires a new robust software infrastructure that can integrate flexible programming models and analytics tools and need to be interoperable on different architectures (HPC, cloud). Engineering such a scalable workflow for scientific computing is both an algorithmic and software challenge.
- **Towards multi-fidelity Monte Carlo sampling in large-scale problems.** The recent advances in multi-fidelity methods for uncertainty propagation, inference, and optimization show great potential for applying the aforementioned outer-loop scenarios in large-scale simulations. During this Long Program, a collaboration was started in which multi-fidelity sampling is applied to plasma microturbulence analysis. Important future research directions include the extension of the current framework, for example to an adaptive version in which the employed low-fidelity models are created dynamically using a quasi-optimal resource management, investigating higher order approaches, or to broaden the application domains. Moreover, although there is already a reasonable mathematical underpinning of the general Ansatz, there is need for more results, e.g., regarding adaptive model selection or assumptions for the applicability of the methodology. A clear interest in multi-fidelity methods is increasingly shown in other communities as well, e.g., from DoE labs or in the form of the Emukit toolkit from the UK Amazon machine learning lab.

It will be interesting to see how these (and many other) ideas developed during the present Long Program will play out in the months and years ahead.

Glossary

AI	Artificial Intelligence
BD	Big Data
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Processing Unit
DNN	Deep Neural Network

DOD	U.S. Department of Defense
DOE	U.S. Department of Energy
FAIR	Findable, Accessible, Interoperable, Reusable
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
HPC	High Performance Computing
IARPA	Intelligence Advanced Research Projects Activity
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input / Output
ML	Machine Learning
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NN	Neural Network
NIST	National Institute of Standards and Technology
NSF	National Science Foundation
OSTP	Office of Science and Technology Policy
TFET	Tunnel field-effect transistor
TL	Transfer Learning
UQ	Uncertainty Quantification