

UC Berkeley

UC Berkeley Previously Published Works

Title

Randomized Algorithms for Scientific Computing (RASC)

Permalink

<https://escholarship.org/uc/item/2d43x0c7>

Authors

Buluc, Aydin
Kolda, Tamara G
Wild, Stefan M
[et al.](#)

Publication Date

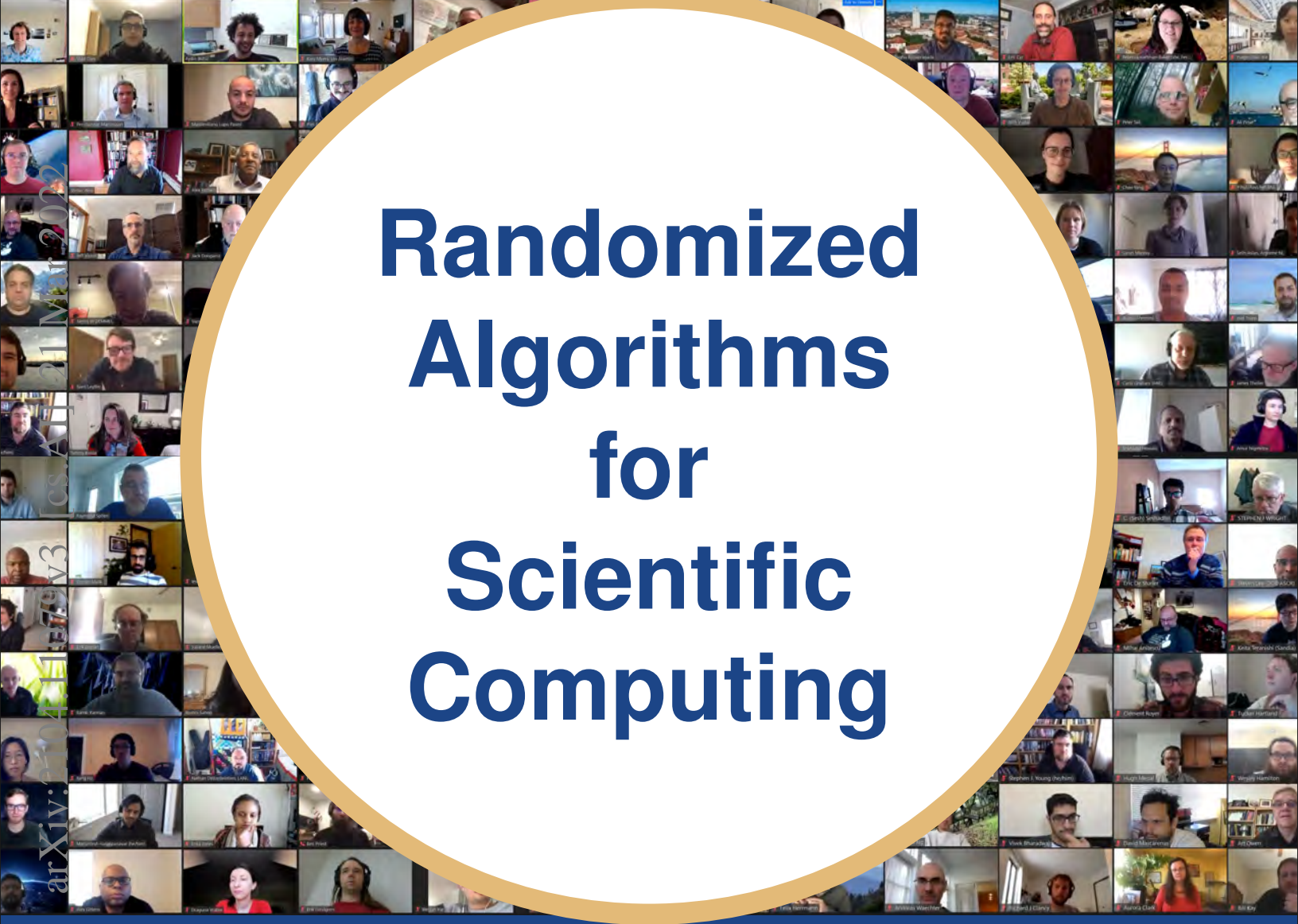
2021-04-19

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

CASA CASR ARSC **ASCR** ASRC ARSC ACSR RASC CASR SARC ACSR SCAR
CRSA CRSA SRAC RSCA CASR RCAS SARC CSRA RACS CASR CSAR ACSR
CASR RASC SACR SRCA ARCS CSRA **RASC** ARSC ACSR RCSA CASR RAS
CASR CARS ACSR SACR SRCA CSRA SARC CRSA RCSA SCAR CSRA RCSA
CASR CRAS ASCR ARCS SARC CRSA SACR RCSA ARSC CRSA SCAR RCSA



Randomized Algorithms for Scientific Computing

Report from the DOE ASCR Workshop
held December 2–3, 2020, and January 6–7, 2021



U.S. DEPARTMENT OF
ENERGY

Office of
Science

Cover

Cover art adapted from group photo of attendees of the virtual workshop.

Aydin Buluc, Tamara G. Kolda, Stefan M. Wild, Mihai Anitescu, Anthony DeGennaro, John Jakeman, Chandrika Kamath, Ramakrishnan Kannan, Miles E. Lopes, Per-Gunnar Martinsson, Kary Myers, Jelani Nelson, Juan M. Restrepo, C. Seshadhri, Draguna Vrabie, Brendt Wohlberg, Stephen J. Wright, Chao Yang, Peter Zwart. “Randomized Algorithms for Scientific Computing (RASC).” United States Department of Energy, Advanced Scientific Computing Research, 2021. doi:10.2172/1807223.

Availability of This Report

This report is available, at no cost, at <https://www.osti.gov/servlets/purl/1807223>.

Disclaimer

This document was prepared as an account of a workshop sponsored by the U.S. Department of Energy. Neither the United States Government nor any agency thereof, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof and shall not be used for advertising or product endorsement purposes.

Randomized Algorithms for Scientific Computing (RASC)

Report from the DOE ASCR Workshop
held December 2–3, 2020, and January 6–7, 2021

Prepared for
U.S. Department of Energy
Office of Science
Advanced Scientific Computing Research Program

RASC Report Authors

Aydın Buluç (co-chair), Lawrence Berkeley National Laboratory
Tamara G. Kolda (co-chair), Sandia National Laboratories
Stefan M. Wild (co-chair), Argonne National Laboratory

Mihai Anitescu, Argonne National Laboratory
Anthony DeGennaro, Brookhaven National Laboratory
John Jakeman, Sandia National Laboratories
Chandrika Kamath, Lawrence Livermore National Laboratory
Ramakrishnan (Ramki) Kannan, Oak Ridge National Laboratory
Miles E. Lopes, University of California, Davis
Per-Gunnar Martinsson, University of Texas, Austin
Kary Myers, Los Alamos National Laboratory
Jelani Nelson, University of California, Berkeley
Juan M. Restrepo, Oak Ridge National Laboratory
C. Seshadhri, University of California, Santa Cruz
Draguna Vrabie, Pacific Northwest National Laboratory
Brendt Wohlberg, Los Alamos National Laboratory
Stephen J. Wright, University of Wisconsin, Madison
Chao Yang, Lawrence Berkeley National Laboratory
Peter Zwart, Lawrence Berkeley National Laboratory

DOE/ASCR Point of Contact

Steven Lee

Additional Report Authors

This report was a community effort, collecting inputs in a variety of formats. We have taken many of the ideas in this report from the thesis statements collected from registrants of “Part 2” of the workshop. We further solicited input from community members with specific expertise. Much of the input came from discussions at the workshop, so we must credit the entire set of participants listed in full in Appendix B. Although we cannot do justice to the origination of every idea contained herein, we highlight those whose contributions were explicitly solicited or whose submissions were used nearly verbatim.

Solicited and thesis statement inputs from

Rick Archibald, Oak Ridge National Laboratory
David Barajas-Solano, Pacific Northwest National Laboratory
Andrew Barker, Lawrence Livermore National Laboratory
Charles Bouman, Purdue University
Moses Charikar, Stanford University
Jong Choi, Oak Ridge National Laboratory
Aurora Clark, Washington State University
Victor DeCaria, Oak Ridge National Laboratory
Zichao Wendy Di, Argonne National Laboratory
Jack Dongarra, University of Tennessee, Knoxville
Jed Duersch, Sandia National Laboratories
Ethan Epperly, California Institute of Technology
Benjamin Erichson, University of California, Berkeley
Maryam Fazel, University of Washington, Seattle
Andrew Glaws, National Renewable Energy Laboratory
Carlo Graziani, Argonne National Laboratory
Cory Hauck, Oak Ridge National Laboratory
Paul Hovland, Argonne National Laboratory
William Kay, Oak Ridge National Laboratory
Nathan Lemons, Los Alamos National Laboratory
Ying Wai Li, Los Alamos National Laboratory
Dmitriy Morozov, Lawrence Berkeley National Laboratory
Cynthia Phillips, Sandia National Laboratories
Eric Phipps, Sandia National Laboratories
Benjamin Priest, Lawrence Livermore National Laboratory
Ruslan Shaydulin, Argonne National Laboratory
Katarzyna Swirydowicz, Pacific Northwest National Laboratory
Ray Tuminaro, Sandia National Laboratories

Contents

Executive Summary	iv
1 Introduction	1
2 Application Needs and Drivers	4
2.1 Massive Data from Experiments, Observations, and Simulations	4
2.2 Forward Models	7
2.3 Inverse Problems	12
2.4 Applications with Discrete Structure	15
2.5 Experimental Designs	18
2.6 Software and Libraries for Scientific Computing	20
2.7 Emerging Hardware	23
2.8 Scientific Machine Learning	24
3 Current and Future Research Directions	27
3.1 Opportunities for Random Sampling Schemes	27
3.2 Sketching for Linear and Nonlinear Problems	30
3.3 Algorithms for Discrete and Combinatorial Problems	33
3.4 Streaming Algorithms and Data	35
3.5 Complexity Analysis	37
3.6 Verification and Validation	40
3.7 Software Abstractions	41
4 Themes and Recommendations	45
4.1 Themes	45
4.2 Recommendations	47
References	49
Appendices	61
A Workshop Information	61
B Workshop Participants	63
C Acknowledgments	74

Executive Summary

Randomized algorithms have propelled advances in artificial intelligence (AI) and represent a foundational research area in advancing AI for Science. Future advancements in DOE Office of Science priority areas such as climate science, astrophysics, fusion, advanced materials, combustion, and quantum computing all require randomized algorithms for surmounting challenges of complexity, robustness, and scalability.

Advances in data collection and numerical simulation have changed the dynamics of scientific research and motivate the need for randomized algorithms. For instance, advances in imaging technologies such as X-ray ptychography, electron microscopy, electron energy loss spectroscopy, or adaptive optics lattice light-sheet microscopy collect hyperspectral imaging and scattering data in terabytes, at breakneck speed enabled by state-of-the-art detectors. The data collection is exceptionally fast compared with its analysis. Likewise, advances in high-performance architectures have made exascale computing a reality and changed the economies of scientific computing in the process. Floating-point operations that create data are essentially free in comparison with data movement. Thus far, most approaches have focused on creating faster hardware. Ironically, this faster hardware has exacerbated the problem by making data still easier to create. Under such an onslaught, scientists often resort to heuristic deterministic sampling schemes (e.g., low-precision arithmetic, sampling every n th element) and sacrifice potentially valuable accuracy.

Dramatically better results can be achieved via randomized algorithms, reducing the data size as much as or more than naive deterministic subsampling can achieve, while retaining the high accuracy of computing on the full data set. By randomized algorithms we mean those algorithms that employ some form of randomness in internal algorithmic decisions to accelerate time to solution, increase scalability, or improve reliability. Examples include matrix sketching for solving large-scale least-squares problems (see Figure 1) and stochastic gradient descent for training machine learning models. We are not recommending heuristic methods but rather randomized algorithms that have certificates of correctness and probabilistic guarantees of optimality and near-optimality. Such approaches can be useful beyond acceleration, for example, in understanding how to avoid measure zero worst-case scenarios that plague methods such as QR matrix factorization.

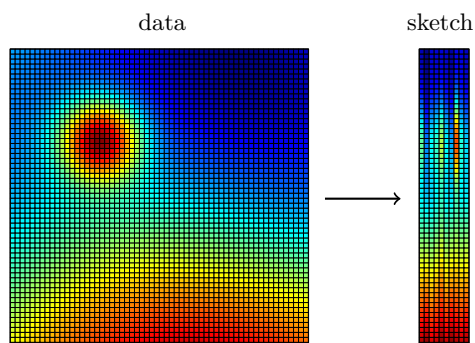


Figure 1: Randomized sketching can dramatically reduce the size of massive data sets.

Randomized algorithms have a long history. The Markov chain Monte Carlo method was central to the earliest computing efforts of DOE's precursor (the Atomic Energy Commission). By the 1990s, randomized algorithms were deployed in regimes such as randomized routing in Internet protocols, the well-known quicksort algorithm, and polynomial factoring for cryptography. Starting in the mid-1990s, random forests and other ensemble classifiers have shown how randomization in machine

learning improve the bias-variance tradeoff. In the early 2000s, compressed sensing, based on random matrices and sketching of signals, dramatically changed signal processing. The 2010s saw a flurry of novel randomization results in linear algebra and optimization, accelerated by the pursuit of problems of growing scale in AI and bringing non-asymptotic guarantees for many problems.

The accelerating evolution of randomized algorithms and the unremitting tsunami of data from experiments, observations, and simulations have combined to motivate research in randomized algorithms focused on problems specific to DOE. The new results in artificial intelligence and elsewhere are just the tip of the iceberg in foundational research, not to mention specialization of the methods for distinctive applications. Deploying randomized algorithms to advance AI for Science within DOE requires new skill sets, new analysis, and new software, expanding with each new application. To that end, the DOE convened a workshop to discuss such issues.

This report summarizes the outcomes of that workshop, “Randomized Algorithms for Scientific Computing (RASC),” held virtually across four days in December 2020 and January 2021. Participants were invited to provide input, which formed the basis of the workshop discussions as well as this report, compiled by the workshop writing committee. The report contains a summary of drivers and motivation for pursuing this line of inquiry, a discussion of possible research directions (summarized in Table 1), and themes and recommendations summarized in the next two pages.

Table 1: A sampling of possible research directions in randomized algorithms for scientific computing

Analysis of randomized algorithms for production conditions
Randomized algorithms cost and error models for emerging hardware
Incorporation of sketching for solving subproblems
Specialized randomization for structured problems
Overcoming of parallel computational bottlenecks with probabilistic estimates
Randomized optimization for DOE applications
Computationally efficient sampling
Stratified and topologically aware sampling
Scientifically informed sampling
Reproducibility
Randomized algorithms for solving well-defined problems on networks
Universal sketching and sampling on discrete data
Randomized algorithms for machine learning on networks
Randomized algorithms for combinatorial and discrete optimization
Randomized algorithms for discrete problems that are not networks
Going beyond worst-case error analysis
Bridging of computational and statistical perspectives
Integration of randomized algorithms into coupled workflows
Mergeable summaries
In situ and real-time data analysis
Privacy
Composable and interoperable randomized abstractions for computation
Use cases for randomized communication
Broker abstractions for randomized input/output

Overarching Themes

Theme 1: Randomized algorithms essential to future computational capacity

The rate of growth in the computational capacity of integrated circuits is expected to slow while data collection is expected to grow exponentially, making randomized algorithms—which depend on sketching, sampling, and streaming computations—essential to the future of computational science and AI for Science.

Theme 2: Novel approaches by reframing long-standing challenges

The potential for randomized algorithms goes beyond keeping up with the onslaught of data: it involves opening the door to novel approaches to long-standing challenges. These include scenarios where some uncertainty is unavoidable, such as in real-time control and experimental steering; design under uncertainty; and mitigating stochastic failures in novel materials, software stacks, or grid infrastructure.

Theme 3: Randomness intrinsic to next-generation hardware

Computing efficiencies can be realized by purposely allowing random imprecision in computations. Imprecision is inherent in emerging architectures such as quantum and neuromorphic computers. Randomized algorithms are a natural fit for these environments, and future computing systems will benefit from the co-design of randomized algorithms alongside hardware that favors certain instantiations of randomness.

Theme 4: Technical hurdles requiring theoretical and practical advances

Crafting sophisticated approaches that break the “curse of dimensionality” via sublinear sampling, sketching, and online algorithms requires sophisticated analysis, which has been tackled thus far only in a small subset of scientific computing problems. Foundational research in theory and algorithms needs to be multiplied many times over in order to cover the breadth of DOE applications.

Theme 5: Reconciliation of randomness with user expectations

Users are conditioned to certain expectations, such as viewing machine precision as sacrosanct, even when fundamental uncertainties make such precision ludicrous. New metrics for success can expand opportunities for scientific breakthroughs by accounting for tradeoffs among speed, energy consumption, accuracy, reliability, and communication.

Theme 6: Need for expanded expertise in statistics and other areas

Establishing randomized algorithms in scientific computing necessitates integrating statistics, theoretical computer science, data science, signal processing, and emerging hardware expertise alongside the traditional domains of applied mathematics, computer science, and engineering and science domain expertise.

Recommendations

Based on community-wide input and workshop discussions, this report recommends pursuing a research program comprising six complementary thrusts.

Recommendation 1: Theoretical foundations

Foundational research in the theory of randomized algorithms to (among other issues) understand existing methods, tighten theoretical bounds, and tackle problems of propagating theory into coupled environments. The output of this research will be theorems and proofs to uncover new techniques and guarantees and to address new problem settings.

Recommendation 2: Algorithmic foundations

Foundational development of sophisticated algorithms that leverage the theoretical underpinnings in practice, identifying and mending any gaps in theory, and establishing performance for idealized and simulated problems. The output here will be advances in algorithm analysis and understanding, prototype software, and reproducible experiments.

Recommendation 3: Application integration

Deployment in scientific applications in concert with domain experts. This will often require extending existing theory and algorithms to the special cases of relevance for each application, as well as application-informed sampling designs. The output here will be software alongside benchmarks and best practices for specific applications, focused on enabling novel scientific and engineering advances.

Recommendation 4: Performance on next-generation hardware

Adaptation of randomized algorithms to take advantage of best-in-class computing hardware, from current architectures to quantum, neuromorphic, and other emerging platforms. The output here will be high-performance open-source software for next-generation computing hardware, including enabling efficient utilization of nondeterministic hardware and maximizing performance of deterministic hardware.

Recommendation 5: Outreach

Outreach to a broader community to facilitate engagement outside the traditional computational science community, including experts in statistics, applied probability, signal processing, and emerging hardware. The output of this effort will be community-building workshops and research efforts with topically diverse teams that break new frontiers.

Recommendation 6: Workflow standardization

Standardization of workflow, including debugging and test frameworks for methods with only probabilistic guarantees, software frameworks that both integrate randomized algorithms and provide new primitives for sampling and sketching, and modular frameworks for incorporating the methods into large-scale codes and deploying to new architectures. The output here will be community best practices and reduced barriers to contributing to scientific advances.

1 Introduction

Randomized algorithms have propelled advances in artificial intelligence (AI) and represent a foundational research area in advancing AI for Science. Future advancements in DOE Office of Science priority areas such as climate science, astrophysics, fusion, advanced materials, combustion, and quantum computing all require randomized algorithms for surmounting challenges of complexity, robustness, and scalability.

Advances in data collection and numerical simulation have changed the dynamics of scientific research and motivate the need for randomized algorithms. For instance, advances in imaging technologies such as X-ray ptychography, electron microscopy, electron energy loss spectroscopy, and adaptive optics lattice light-sheet microscopy collect hyperspectral imaging and scattering data in terabytes at breakneck speed enabled by state-of-the-art detectors. The data collection is exceptionally fast compared with its analysis. Likewise, advances in high-performance architectures have made exascale computing a reality and changed the economies of scientific computing in the process. Floating-point operations that create data are essentially free in comparison with data movement. Thus far, most approaches have focused on creating faster hardware. Ironically, this faster hardware has exacerbated the problem by making data still easier to create. Under such an onslaught, scientists often resort to heuristic deterministic sampling schemes (e.g., low-precision arithmetic, sampling every n th element) and sacrifice potentially valuable accuracy.

Dramatically better results can be achieved via randomized algorithms, reducing the data size as much as or more than naive deterministic subsampling while retaining the high accuracy of computing on the full data set. By randomized algorithms we mean those algorithms that employ some form of randomness in internal algorithmic decisions to accelerate time to solution, increase scalability, or improve reliability. Examples include matrix sketching for solving large-scale least-squares problems and stochastic gradient descent for training machine learning models. We are not recommending heuristic methods but rather randomized algorithms that have certificates of correctness and probabilistic guarantees of optimality and near-optimality. Such approaches can be useful beyond acceleration, for example, in understanding how to avoid measure zero worst-case scenarios that plague methods such as the QR matrix factorization.

Randomized algorithms have a storied history in computing. Monte Carlo methods were at the forefront of early Atomic Energy Commission (AEC) developments by Enrico Fermi, Nicholas Metropolis, and Stanislaw Ulam [110, 111] and inspired John von Neumann to consider early automated generation of pseudorandom numbers to avoid latency costs of relying on state-of-the-art tables [126]. Ulam's line of inquiry was rooted in solitaire card games but aimed at practical efficiency [57]:

After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than abstract thinking might not be to lay it out say one hundred times and simply observe and count the number of successful plays.

In the 1950s, Arianna Rosenbluth programmed the first Markov chain Monte Carlo implementation, which was for an equation-of-state computation on AEC's groundbreaking MANIAC I (Mathematical Analyzer Numerical Integrator and Automatic Computer Model I) computer [112]. In subsequent years, the consideration of systems at equilibrium and study of game theory have resulted in many randomized algorithms for resolving mixed strategies for Nash equilibria [117]. By the 1990s, randomized algorithms were deployed in regimes such as randomized routing in Internet protocols,

the well-known quicksort algorithm, and polynomial factoring for cryptography [118, 86]. In the mid-1990s, methods such as random forests and other randomized ensemble classifiers improved accuracy in machine learning, demonstrating that ensembles built from independent random observations can yield superior generalization [36, 37]. In the early 2000s, compressed sensing, based on random matrices and sketching of signals, dramatically changed signal processing [46]. The National Academies’ Mathematical Sciences in 2025 report [119] stated:

It revealed a protocol for acquiring information, all kinds of information, in the most efficient way. This research addresses a colossal paradox in contemporary science, in that many protocols acquire massive amounts of data and then discard much of it, without much or any loss of information, through a subsequent compression stage, which is usually necessary for storage, transmission, or processing purposes.

The work showed that the traditional Shannon bounds of information theory can be overturned whenever the underlying signal has structure and that randomized algorithms are key to this development.

The 2010s saw a flurry of novel results in linear algebra and optimization, accelerated by problems of increasing scale in artificial intelligence.

The accelerating evolution of randomized algorithms and the unrelenting tsunami of data from experiments, observations, and simulations have combined to motivate research in randomized algorithms focused on problems specific to DOE. The new results in AI and elsewhere are just the tip of the iceberg in foundational research, not to mention specialization of the methods for distinctive applications. Deploying randomized algorithms to advance AI for Science within DOE requires new skill sets, new analysis, and new software, expanding with each new application. To that end, DOE convened a workshop to discuss such issues.

This report summarizes the outcomes of that workshop, “Randomized Algorithms for Scientific Computing (RASC),” held virtually across four days in December 2020 and January 2021.¹ The first two days of the workshop, the “boot camp,” focused on highly interactive technical presentations from experts and had 453 participants. The second part of the workshop, held one month later, focused on community input and had 204 fully engaged participants. Participants in both parts were invited to provide inputs during, in between, and after the sessions. These inputs have formed the basis of this report, which was compiled by the workshop writing committee.

The report is organized as follows. [Section 2](#) describes the need for a colossal leap in computational capacity, across the board, motivated by ever-larger and more heterogeneous data collection, larger-scale and higher-resolution simulations, bounding uncertainty in high-dimensional inverse problems, higher-complexity real-time control scenarios, inherent randomness in emerging computational hardware itself, and scientific applications of AI. Ideas for foundational and applied research in randomized algorithms that address these needs are described in [Section 3](#). These ideas range from linear and nonlinear systems, to algorithms for discrete and combinatorial problems, to random sampling strategies and streaming computations, to software abstractions. Much attention is focused on providing a combination of theoretical robustness (i.e., assurances of correctness for model problems), efficiency, practicality, and relevance for problems of interest to DOE. We conclude with high-level themes and recommendations in [Section 4](#), not least of which is the need

¹In contrast to past workshops, such as Scientific Machine Learning [18] and AI for Science [137], this workshop was held virtually because of the COVID-19 pandemic, which prevented travel in the winter of 2020/2021. The virtual format had the benefit of enabling much broader engagement than past in-person workshops.

for reconciling user expectations with the results of randomized algorithms and the need to engage broader expertise (e.g., from statistics) than has historically been needed in the ASCR program. The appendices give further details of the workshop: see [Appendix A](#) for the workshop agenda, [Appendix B](#) for a full list of participants, and [Appendix C](#) for acknowledgments.

2 Application Needs and Drivers

Over the next decade, DOE anticipates scientific breakthroughs that depend on modeling more complex chemical interactions yielding higher-capacity batteries, computing on emerging hardware platforms such as quantum computers with inherent randomness, analyzing petabytes of data per day from experimental facilities to understand the subatomic structure of biomolecules, and discovering rare and previously undetected isotopes. Achieving these science breakthroughs requires a colossal leap in DOE’s capacity for massive data analysis and exascale simulation science.

Algorithmic advances have always been the key to such progress, but the scale of the challenge over the next decade will oblige DOE to forge into the domain of *randomized algorithms*. Supporting this new effort will mean recruiting experts outside the traditional areas of computational science and engineering, high-performance computing, and mathematical modeling by attracting and involving experts in applied probability, statistics, signal processing, and theoretical computer science.

Advances in randomized algorithms have been accelerating in the past decade, but there is a high barrier to integration into DOE science. This is in large part because DOE has *unique needs that require domain-specific approaches*. In this section we highlight specific DOE applications, the challenges of the coming decade, and the potential of randomized algorithms to overcome these hurdles.

2.1 Massive Data from Experiments, Observations, and Simulations

Subsection lead: C. Kamath

The DOE Office of Science operates several national science user facilities, including accelerators, colliders, supercomputers, light sources, and neutron sources [30]. Spanning many different disciplines, these facilities generate massive amounts of complex scientific data through experiments, observations, and simulations. For example, by the year 2035, ITER, the world’s largest fusion experiment [1], will produce two petabytes of data every day, with 60 instruments measuring 101 parameters (Figure 2) during each experiment or “shot.” The data will be processed subject to a wide range of time and computing constraints, such as analysis in near-real time, during a shot, between shots, and overnight, as well as remote analysis and campaign-wide long-term analysis [42].

These constraints, as well as the volume and complexity of the data, require new advances in data-processing techniques. Similar requirements are echoed by scientists as they prepare their simulations for the exascale era [49]. The nanoscale facilities at DOE also provide challenges as scientists aim to control matter at the atomic scale through the Atomic Forge [83]. Manipulating atoms by using a scanning transmission electron microscope involves real-time monitoring, feedback, and beam control. Scalable randomized algorithms will be essential for achieving success in this new field of atom-by-atom fabrication (Figure 3).

In order to fully realize the benefits of the science enabled by DOE facilities, the techniques currently used for processing the data must be enhanced to keep pace with the ever-increasing rate, size, and complexity of the data.

In order to fully realize the benefits of the science enabled by these DOE facilities, the techniques currently used for processing the data must be enhanced to keep pace with the ever-increasing rate, size, and complexity of the data. For simulation data generated on exascale systems, these techniques include compression, in situ analysis, and computational steering, while experimental

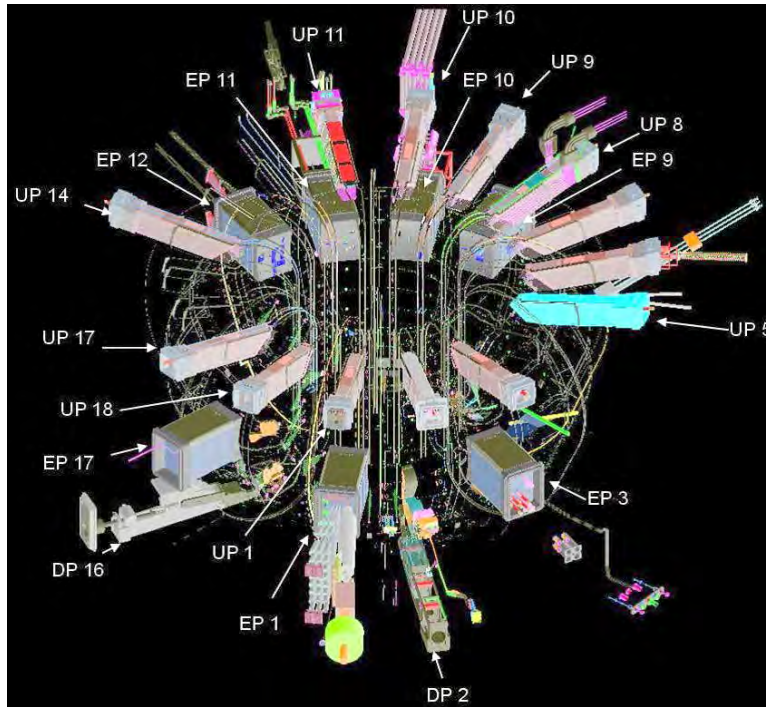
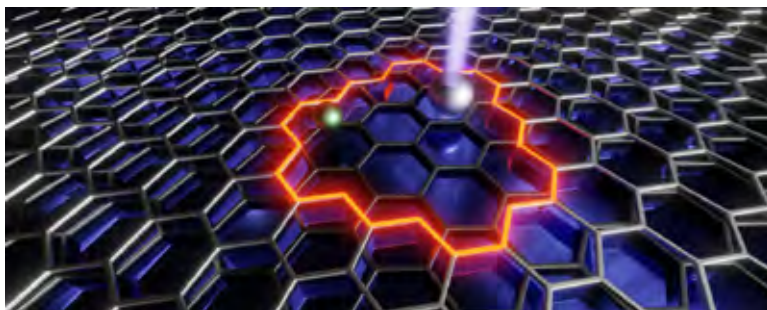


Figure 2: Schematic of ITER diagnostics [34] illustrating some of the instruments that will generate two petabytes of data per day at the ITER Scientific Data Centre [35]. Randomized algorithms offer a potential solution to the challenge of processing of these massive, complex data sets.

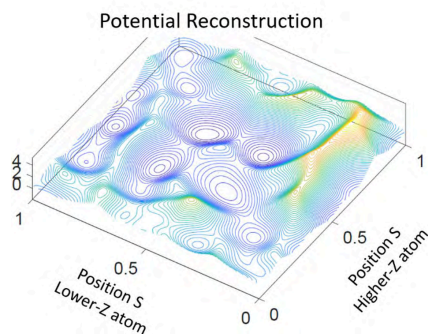
and observational data require robust, real-time techniques to identify outliers, to fit machine learning models, or to plan subsequent experiments. In contrast to simulations that can be paused to reduce the data overload, experimental and observational data often must be processed as it arrives in an unrelenting stream, or else risk losing parts of the data.

Randomized algorithms offer a solution to this challenge of processing massive data sets in near-real time by introducing concepts of sparsity and randomness into the algorithms currently in use. However, several technical issues must be addressed before such algorithms are broadly accepted and used by scientists. Most important is a need to understand the uncertainties in the reliability and reproducibility of the results from these algorithms, especially given their “random” nature. In experiments where not all the data being generated can be stored, the critical information to be saved must be correctly identified, even though the randomized algorithms sample only a subset of the data stream; and predicting whether a sample is useful or not can be difficult. Randomized algorithms must also be able to process data at different levels of precision, as well as data stored in different data structures, such as images and time series, including structured and unstructured data from simulations.

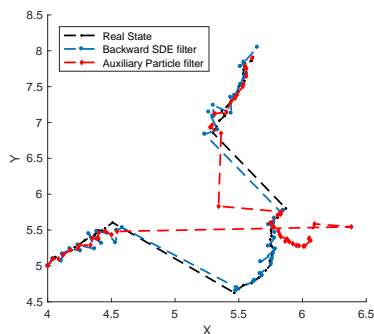
Addressing some of these roadblocks to the use of randomized algorithms in processing massive data sets requires longer-term research, but several near-term opportunities exist. Two areas where randomized algorithms could prove useful are data reduction through various compression techniques and acceleration of data analysis. These algorithms could also be more accurate than periodic sampling in some problems and more efficient than the use of dense linear algebra, lead to better communication efficiency in high-performance computing, and allow more sophisticated analysis to be performed in real time. In problems involving both simulations and experiments/observations,



(a) Visualization of STEM interacting with a silicon atom in a graphene hole.



(b) Reconstruction of potential on a torus expanded coordinate system.



(c) Tracking of atomic position using auxiliary particle and backward doubly stochastic differential equation filters.

Figure 3: A scanning transmission electron microscope is capable of measuring and modifying the location of a silicon atom in a graphene hole. Current mathematical methods cannot reconstruct the energetic landscape from the sparse and noisy measurements from the microscope or track the atoms accurately. However, newly developed randomized algorithms are providing fundamental tools to achieve the goal of real-time atomic control [56, 19]

randomized algorithms could enable data assimilation of larger data sets, improve data-driven simulations by allowing faster use of experimental data, perform better parameter estimation by exploring larger domains, and open new venues for improvement as ideas for use of these algorithms in simulations are transferred to experiments/observations and vice versa. The insight and compression capabilities provided by randomized algorithms could also be used for improved long-term storage of data. The specific areas where additional research is required to accomplish these improvements are outlined in [Section 3](#).

Technical issues are not the only roadblocks to the use of randomized algorithms in processing massive data sets. From a societal viewpoint, these algorithms are currently not well understood enough for domain scientists to be comfortable incorporating them into their work. Often lacking is an awareness of opportunities where such algorithms may make processing massive data sets tractable, as well as a lack of robust software that is scalable to massive data sets. Addressing both the technical and societal concerns would help scientific communities processing data from experiments, observations, and simulations to accept and benefit from randomized algorithms. If successful, this would result in reduced time to science, more effective science, and a greater return on the investment DOE has made in its facilities.

2.2 Forward Models

Subsection lead: C. Yang

As we gain a deeper understanding of a wide range of physical phenomena, forward models become more complex. A scientific inquiry often begins with a hypothesis in the form of a forward model that describes what we believe to be the fundamental laws of nature and how different physical processes interact with each other. Mathematically, these models are represented by algebraic, integral, or differential equations. They contain many degrees of freedom to account for the multiscale and multiphysics nature of the underlying physical processes. For example, to model the interaction of fluids and structures, we need to include velocity, pressure for the fluid, and displacement of the structure. To simulate a photovoltaic system, we need take into account electron excitation, charge separation, and transport processes, as well as interface problems at a device level. To understand electrochemistry in a Lithium-ion battery, we need to simulate the dynamic interface between the electrode and electrolytes during the charging and discharging cycles ([Figure 4](#)). To perform a whole-device modeling of a tokamak fusion reactor, we need to combine the simulation of core transport, plasma materials interaction at the wall, and global MHD stability analysis ([Figure 5](#)). To model the fully coupled Earth system, we need to consider the interactions among the atmospheric, terrestrial/subsurface, and ocean cryosphere components ([Figure 6](#)).

Such complexity challenges our ability to perform computer simulations with sufficient resolution to validate our hypotheses by comparing with scientific observations and experimental results. A high-fidelity simulation to predict extreme events in a climate model at a spatial resolution of 1 kilometer yields an extremely large number of coupled algebraic, integral, and differential equations with the number of variables, n , in the billions. The complexity of existing numerical methods for solving these problems is often $O(n^p)$ for some integer power $p > 1$, and the number of degrees of freedom n can be millions or billions. For example, the complexity of a density-functional-theory-based electronic structure calculation for weakly correlated quantum many-body systems is $O(n^3)$, with n as large as a million. More accurate models for strongly correlated systems such as the coupled cluster model may require $O(n^7)$ floating-point operations, with n in the thousands. The computational bottleneck is often in the solution of large-scale linear algebra problems. Because of the nonlinearity of many forward models, these equations need to be solved repeatedly in an

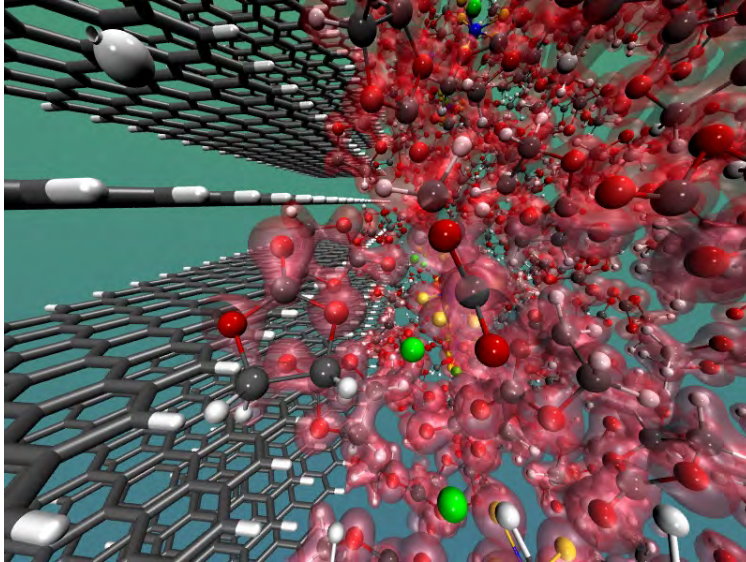


Figure 4: Snapshot of an ab initio molecular dynamics simulation of the solid-electrolyte interphase (SEI) of a lithium-ion battery. A nonlinear eigenvalue with millions of degrees of freedom needs to be solved at each time step. Thousands of time steps are required to generate a trajectory from which properties of the SEI can be characterized. Randomized trace estimation may be used to reduce the computational complexity in each time step from $O(n^3)$ to $O(n)$ and significantly shorten the simulation time. (provided by J. Pask; a similar figure appeared in [145])

iterative procedure. Even with the availability of the exascale computing resources, performing these types of simulations is extremely challenging .

A high-fidelity simulation to predict extreme events in a climate model at a spatial resolution of 1 kilometer yields an extremely large number of coupled algebraic, integral, and differential equations, with the number of variables, n , in the billions.

Furthermore, because of model uncertainties, such as fluctuation and noise, multiple simulations need to be performed in order to obtain ensemble averages.

State of the Art

Randomized algorithms have proven effective at overcoming some of the challenges discussed above. In particular, randomized projection methods have been used to reduce the dimension of some problems by projecting linear and nonlinear operators in the model onto a randomly generated low-dimensional subspace and solving a smaller problem in this subspace [17, 9]. Although projection methods have been used in a variety of applications, the traditional approach often requires a judicious construction (e.g., through a truncated singular value decomposition or the application of a Krylov subspace method) of a desired subspace that captures the main characteristics of the solution to the original problem. This step can be costly. For problems that exhibit fast singular value decay, randomized projection works equally well but at much lower cost. Fast sketching strategies based on structured random maps can potentially accelerate computations dramatically.

In addition to being used as an efficient technique for dimension reduction, randomized algorithms have played an important role in reducing the complexity of linear solvers for discretized elliptic partial differential equations (PDEs) and Helmholtz equations [102]. By taking advantage of the

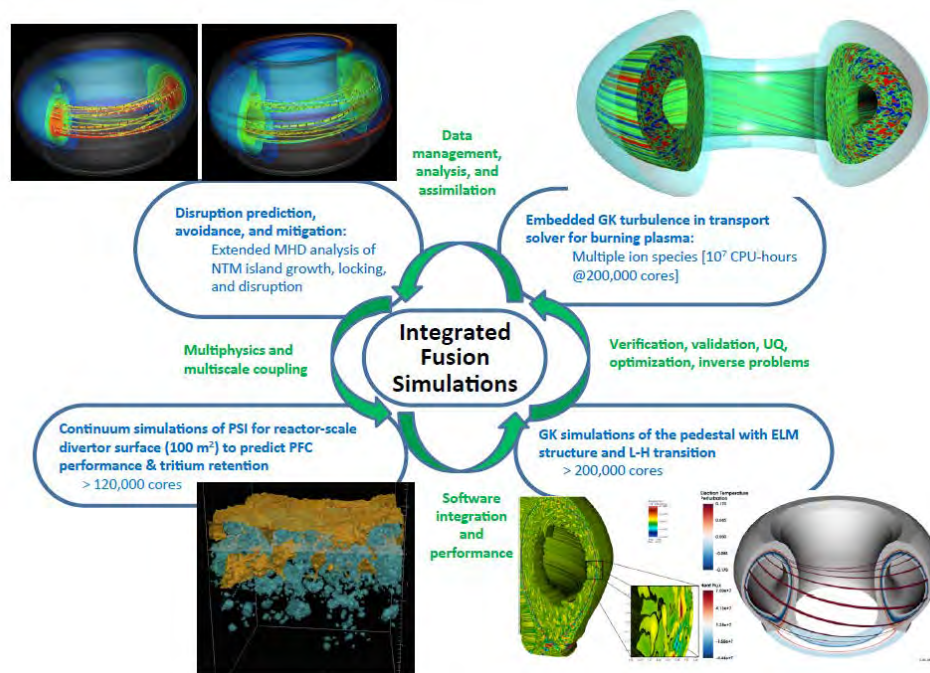


Figure 5: A whole-device model of a tokamak when using the highest-fidelity physics models available for core transport, the pedestal and scrape-off layer, plasma-materials interactions at the wall, and global MHD stability. Simulating such a model is extremely challenging and requires a tremendous amount of computational resources. Randomized projection methods can be used to significantly reduce the computational cost. (from [50])

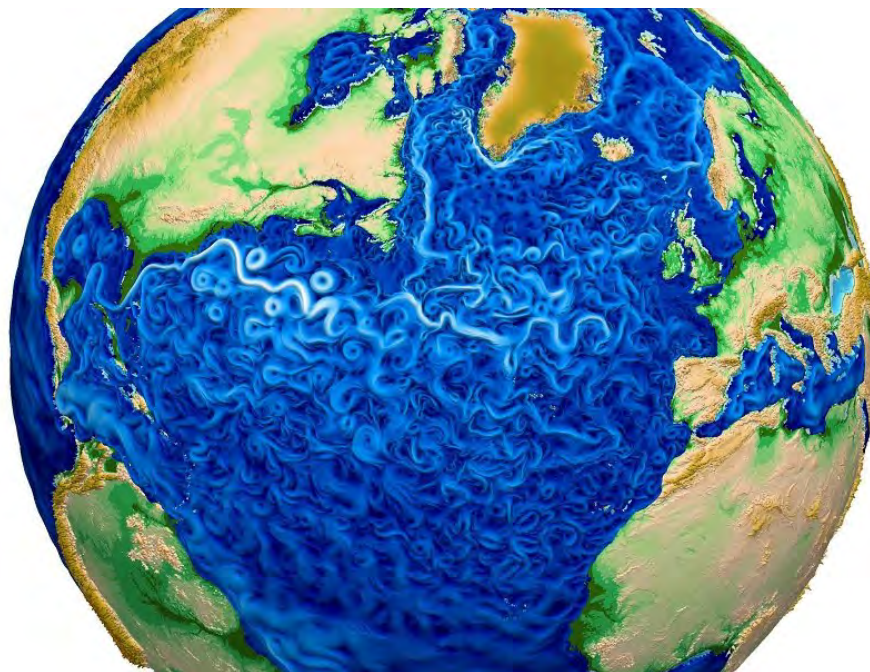


Figure 6: DOE’s Energy Exascale Earth System Model(E3SM) simulation showing global eddy activity. Performing this simulation, which combines atmosphere, ocean, and land models, plus sea- and land-ice models, requires DOE exascale computing resources. Randomized algorithms can significantly accelerate the computation and are well suited for exascale platforms. (from [e3sm.org](https://ascr-discovery.org/2020/11/climate-on-a-new-scale/) and the article <https://ascr-discovery.org/2020/11/climate-on-a-new-scale/>)

low-rank nature of the long-range interaction in the Green’s function, randomized algorithms allow us to construct compact representations of approximate factors of linear operators [104, 72, 153], preconditioners for iterative solvers [58, 67, 87], or direct solvers that construct data sparse representations of the inverse of the coefficient matrix for many linear systems [155, 103]. They are also used in constructing low-rank approximations to tensors, for example, the two-electron integral tensors that appear in Hartree–Fock or hybrid functional density-functional-theory-based electronic structure calculations [77, 78, 94].

In addition to being used as an efficient technique for dimension reduction, randomized algorithms have played an important role in reducing the complexity of linear solvers for discretized elliptic PDEs and Helmholtz equations.

Randomized algorithms have also been used to compute physical observables such as energy density through randomized trace estimation [80, 15, 144]. This technique has been used successfully in ground- and excited-state electronic structure calculations for molecules and solids [61, 149]. The use of this type of randomized algorithm often leads to linear complexity scaling with respect to the number of atoms, which is a major improvement compared with methods that require solving a large-scale eigenvalue problem with $O(n^3)$ complexity.

For high-dimension problems, Monte Carlo methods have been used extensively to generate random samples of desired quantities to be averaged over (as an approximation to a high-dimensional integral) [14, 151, 90].

The random samples must be generated according to an underlying distribution function that may

be unknown. A widely used technique to achieve this goal is the Metropolis–Hasting algorithm, also known as the Markov chain Monte Carlo. This algorithm has been successfully used in kinetic models to study rare events and chemical reactions in large-scale molecular systems [150], quantum many-body models to approximate the ground state energies of strongly correlated systems [14, 39], and turbulent flow models [90] used to study atmosphere-ocean dynamics, combustion, and other engineering applications.

Opportunities

Although randomized algorithms have been developed and used in several applications, many more applications can potentially benefit. This is particularly true in a multifidelity or multiscale framework in which we need to couple simulations across multiple spatial and temporal scales, for example, a wind farm modeled at $\mathcal{O}(10m)$ resolution coupled with climate simulations run at $\mathcal{O}(10km)$. Randomized algorithms might be used to provide avenues for enhancing the information shared in such a coupling and open the door to questions related to uncertainty. One example is the use of stochastic subgrid process models, that is, computational models that provide source terms from processes with spatial and time scales below mesh resolution, to represent missing information (introduced by grid-level filtering) probabilistically, rather than deterministically, by sampling subgrid source term from appropriate distributions conditioned nonlocally on grid-level evolution. Probabilistic models learn such distributions from direct numerical simulation data and deploy learned samplers in large-scale simulations.

Although randomized algorithms have been developed and used in several applications, many more applications can potentially benefit from randomized algorithms. This is particularly true in a multifidelity or multiscale framework in which we need to couple simulations across multiple spatial and temporal scales.

In general, sampling methods play an important role in the convergence of randomized algorithms. A good sampling method can lead to a significant reduction in variance and, consequently, a reduced number of samples required to achieve high accuracy. Although importance sampling and umbrella sampling methods have been developed in several applications in quantum and statistical mechanics, room for improvement still remains. These techniques can potentially be applied to a much broader class of problems.

Despite the tremendous success randomized linear algebra has enjoyed in accelerating key matrix computation kernels of many simulations, much is yet to be done to extend these techniques to multilinear algebra (tensor) and nonlinear problems.

In order to achieve both high accuracy and efficiency, a hybrid method may be desirable in which a low-complexity randomized algorithm is used to provide a sufficiently good approximation that can be refined or postprocessed by a deterministic method.

Integrating randomized algorithms in the existing deterministic algorithms-based simulation pipeline would require a careful examination of practical issues including data structure and parallelization. Randomized or hybrid randomized and deterministic algorithms have the potential to reduce the complexity of many of the most demanding simulation problems to $\mathcal{O}(n^2)$ or $\mathcal{O}(n)$. They can be much more scalable than existing methods and are suitable for exascale machines.

With the help of randomized algorithms we can continue to push the envelope of large-scale simulation in many scientific disciplines and significantly improve the fidelity of models needed to

provide accurate descriptions of a broad range of natural phenomena and engineered processes. Randomized algorithms are particularly powerful in tackling high-dimensional problems that are not amenable to conventional deterministic approaches. They are game changers for solving some of the most challenging computational problems in many applications.

2.3 Inverse Problems

Subsection lead: B. Wohlberg

The analysis of experimental measurements plays a fundamental role in basic science and mission areas within DOE. In many cases the quantities of interest cannot be directly measured and must instead be inferred from related quantities that are amenable to measurement. This inference is typically posed as an *inverse problem*, where the corresponding *forward problem* describes the physics of the measurement process that maps the quantities of interest to the measurements.

Large-Scale Inverse Problems for Science Data

A classical example of an inverse problem is X-ray computed tomography (CT), in which a map of the internal density of an object is inferred from a sequence of radiographs taken from different view directions.

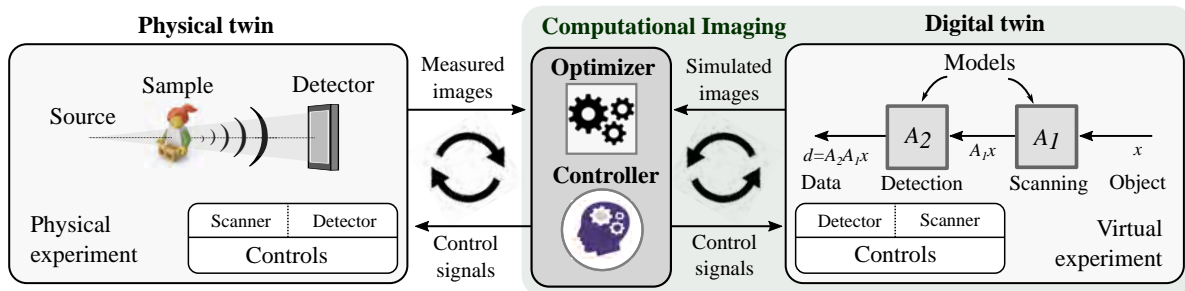


Figure 7: Illustration of the main components of a computational imaging system, including a sensing system and an inverse problem solver consisting of a “digital twin” or “forward model” of the imaging system and an optimizer. (courtesy of D. Gürsoy, Argonne National Laboratory.)

In addition to CT, numerous other imaging techniques involving inverse problems (Figure 7) are relevant to DOE applications including materials science, parameter estimation for complex models related to global climate change, oil and gas exploration, groundwater hydrology and geochemistry, and calibration of computational cosmology models. Many inverse problems arise in experiments at DOE imaging facilities, which can produce large volumes of data (e.g., up to 10 GB/s at the current Linac Coherent Light Source, with 100 GB/s predicted for next-generation instruments [30, Sec. 15.1.4]). Examples of such problems include CT [76] and CT reconstruction from a set of coherent imaging experiments [21]. Such data sets are rapidly growing in size as imaging technology improves and new experimental facilities are constructed, leading to an urgent need for improved computational capabilities in this area.

The primary challenges to be addressed are the following.

- Many of these problems are of a sufficient scale that they can be solved only by using advanced high-performance computing resources (e.g., see [30, p. 158]) that are in limited supply. While computing power is important, the most significant constraint is usually the need to keep the entire reconstruction and measured data set in working memory. Online or streaming

algorithms that avoid this need would allow large-scale problems to be solved on a much broader range of computing hardware.

- DOE imaging facilities are heavily oversubscribed, with the result that experiments have to be conducted within a limited time window. Thus, calibration or other issues that might degrade the utility of the experiments are often discovered only when reconstructions are computed after the experiment has been completed. A capability for real-time or progressive reconstructions would be of enormous value in addressing this difficulty [146, Sec. PRO 1].

Inverse problems at DOE facilities produce very large volumes of data, for example, 10 GB/s at the current Linac Coherent Light Source, with 100 GB/s predicted for planned instruments.

While careful design of massively parallel algorithms can provide close to real-time reconstructions of relatively large problems given a sufficient number of compute nodes [76], not all inverse problems are amenable to this type of problem decomposition, and such large-scale computing facilities (256,000 cores in [76]) are not widely available.

Randomization methods offer a number of different approaches to address these challenges:

1. Application of inherently randomized algorithms such as stochastic gradient descent, randomized Levenberg–Marquardt, and derived methods for solving the optimization problems associated with the inverse problems [32, 28, 138, 141, 140]
2. Use of randomized methods for solution of subproblems of non-randomized algorithms (e.g., use of sketching for solving the linear subproblem related to the data fidelity term within an alternating direction method of multipliers (ADMM) algorithm [33])
3. Use of randomized algorithms for efficiently training machine learning surrogates of physics models, with efficient use of queries of expensive physics models, and for efficiently using large, complex data sets

Quantifying Uncertainty in Inverse Problems through Randomized Algorithms

Quantifying uncertainty in inverse problems and parameter estimation problems is an important step toward providing confidence in estimation results (see, e.g., Figure 8), but emerging sensors and novel applications pose significant challenges. For example, LIDAR sensors, for which the usage is rapidly increasing, provide much higher resolution information about wind profiles. While the instrument measurement error is well understood, little alternative information exists that can be used to assess the accuracy of a reconstructed 3D wind field at that resolution (tens of meters in space and seconds in time). The increased expectations of capabilities lead researchers to consider applications with parameters of high spatiotemporal heterogeneity. Thus, uncertainty must be expressed over parameter spaces of vastly increased dimensionality with respect to problems currently approachable via state-of-the-art uncertainty quantification techniques. Moreover, this parameter heterogeneity, the complexity of the physics of these novel applications, and the advancement of sensing techniques result in experimental data sets composed of various data sources with vastly different data collection protocols in different regions of their state space and errors of different probabilistic characteristics. Despite the significant advances in methods for Bayesian inference, efficiently leveraging the physical constraints and laws characterizing applications of interest in conjunction with data remains a significant computational challenge. This is particularly true where the encoding of physical constraints and laws is in the form of expensive computational simulations

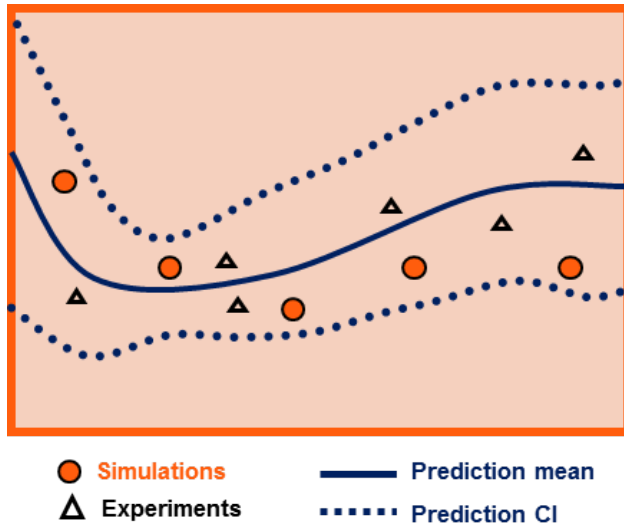


Figure 8: Illustration of the estimation of mean and confidence interval from both simulations and experiments. (diagram by “Jiangzhenjerry,” CC BY-SA 3.0 license, source <https://commons.wikimedia.org/w/index.php?curid=19797809>)

with their own complexity drivers; this challenge is in turn compounded by the heterogeneity of data sources described above.

Quantifying uncertainty in inverse problems and parameter estimation problems is an important step toward providing confidence in estimation results.

Randomization methods offer an unprecedented promise for tackling the challenges in uncertainty quantification. Specifically, randomization techniques may lead to gains in computational efficiency in various places along the probabilistic modeling pipeline (e.g., accelerated solution of subproblems, the training of machine-learning-based surrogate models). Furthermore, randomization can support the solution of stochastic programs associated with approximate Bayesian inference such as variational inference. Other examples of research challenges and opportunities include the following:

1. Leveraging randomization methods for probabilistic inference with streaming data. Integrating online data assimilation algorithms (e.g., particle filters) together with randomization techniques such as sketching, approximate factorizations, and randomized calculations with hierarchical matrices may lead to improvements in scalability and efficiency. Furthermore, the impact of compression via randomization of streaming data on the inference process needs to be explored.
2. Analysis of the effect of randomization in sketching, data compression, and other techniques on the convergence and bias of probabilistic reconstructions. Such an analysis would distinguish, in the probabilistic setting, the uncertainty stemming from randomized methods and the uncertainty inherent in the inverse problem due to observation errors, for example.

Carrying out these advances will result not only in faster solution to the analysis problems but also in increased predictive capabilities of the resulting computational tools.

2.4 Applications with Discrete Structure

Subsection lead: J. Restrepo

We next consider problems with discrete structure, notably networks and graphs. In the context of network science, specific application drivers are familiar: critical infrastructures such as the Internet and power grids, as well as biological, social, and economic structures.

Faster and better ways to analyze, sample, manage, and sort discrete events, graphs, and connected data streams will have dramatic impact on networked applications. In what follows, we highlight two application areas that demonstrate the mathematical and algorithmic challenges.

Community detection is arguably one of the most widely used network analysis procedures. The basic purpose is to detect and categorize aggregations of activity in a network. Many algorithms have been developed for this purpose. Practitioners often run these algorithms on their data, but their data is actually a snapshot of the whole (e.g., a Twitter snapshot from the entire stream). These methods yield no guarantee on the structure of the original, not the observed, network. Current sampling methods include stratified sampling techniques along with topological/dimensionality reduction techniques (i.e., the Johnson–Lindenstrauss lemma [6]). Some algorithms reduce quadratic time to near-linear time complexity for specific classes of problems, but the scope is frequently narrow.

Researchers have little understanding of the mathematics in downsampling such a complex structure, and the current state-of-the-art approaches are based on unproven heuristics (see surveys [95, 101, 4]). Associated with sampling and searching is a general class of algorithms called streaming. A rich history of streaming algorithms exists in the theoretical computer science and algorithms community. While some of these methods have had significant success in practice (e.g., the HyperLogLog sketch [64]), much of this field has remained purely mathematical. Advances in graph sampling would provide methods to subsample a graph so that community detection algorithms on the subsample would provide guarantees on the entire structure.

New graph search challenges appear in the context of black-box optimization techniques, as a result of its relevance to many machine learning and reinforcement learning contexts. For graphs that are larger to manage or store than the resources available on a single machine, the search requirements on the discrete space are impractical or expensive.

Randomized algorithms may have an impact on streaming, and in general on sampling and searching, and thus an impact on community detection and other analysis needs in network science. Further, new uses for searches in connection with tasks associated with machine learning will also benefit, should randomization lead to efficiencies associated with informing neural nets with data associated with graph structures.

Power Grid A critical national security challenge is the maintenance of the integrity of national power grids (Figure 9) under adverse conditions caused by nature or humanity.

With more renewable power sources on the grid, such as solar and wind, uncertainty on the power supply side increases, caused by variations in weather; and potential disruptions become even more difficult to manage. Thus, grid planners and operators need to assess grid management and response strategies to best maintain the integrity of the national power grid under extremely complex and uncertain operating conditions. Currently, the Exascale Computing Project (ECP) subproject ExaSGD (Optimizing Stochastic Grid Dynamics at ExaScale) is developing methods to optimize the grid’s response to a large number of potential disruption events under different weather

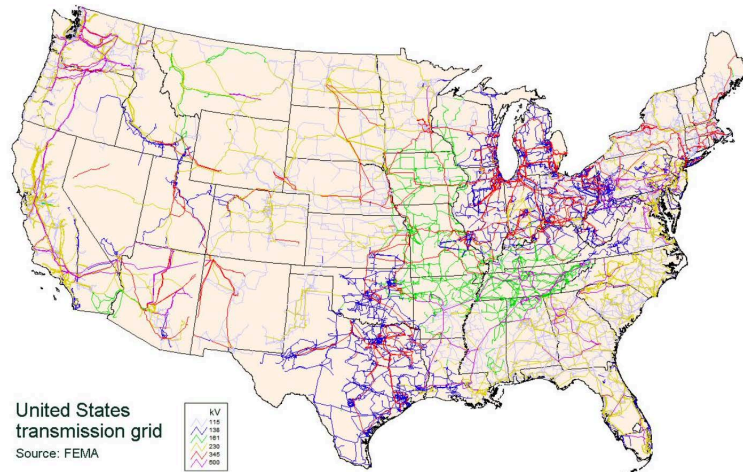


Figure 9: The U.S. electric power transmission grid is over 120,000 miles long. The figure suggests that complexities in the analysis and management of the network result from node heterogeneity, the federated nature of the management of the network, and the complexity of sources and users. Inherent stochasticity in this and other networks could be exploited by proposing ways to speed up sampling, searching, and analysis of the network via randomized algorithms. (source: FEMA).

scenarios [79]. For example, in the Eastern Interconnection grid one must consider thousands of contingencies taking place under thousands of possible (and impactful) weather scenarios. ExaSGD is developing massively parallel algorithms for security-constrained optimal power flow analysis that could involve simultaneous optimization of millions of power grid realizations. Future research will need to address other power grid analysis questions that require discrete optimization. For example, the unit commitment problem, selecting how much steady-generation power (e.g., coal-based or nuclear-based) to buy and from where, is a large mixed-integer program for given demand and generation estimates. Finding the worst-case placement of k outages is a bilevel discrete optimization problem since the network can reoptimize to mitigate the damage.

Grid planners and operators need to assess grid management and response strategies to best maintain the integrity of the national power grid under extremely complex and uncertain operating conditions.

Randomized algorithms could help in a number of places in the near term. Progress in these will also have an impact on other networked infrastructure systems and beyond:

1. Randomized rounding to find feasible solutions for, for example, unit commitment given a fractional solution to a relaxation. Finding a provably good solution, or even finding a feasible solution with reasonable probability, is valuable. Using approximations for the DC optimal power flow (DCOPF) may speed up interdiction problems.
2. Fast approximation of DCOPF. Randomized approximation schemes for network flow exist [26]. Can these be made faster, if less accurate? Can these network-flow approximation

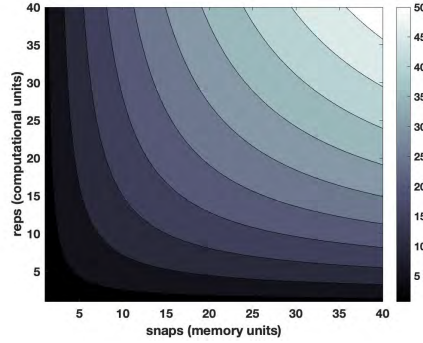


Figure 10: Logarithm of the total operations required to compute a derivative via checkpointing as a function of storage (*snaps*) and time or stage steps (*reps*); see [129]. Randomization could bring down the computational costs when the differentiation products do not have to be exact in structure and/or in value.

algorithms be extended to include the phase-angle constraints from DCOPF?

3. Randomized selection of scenarios. Is there a way to select a finite set of scenarios that are representative of the full set? There has been some experimental success on finding average damage estimates on stochastic versions of network-interdiction problems (e.g., [81]). The largest current supercomputers tend to have GPUs on the nodes, and GPUs are particularly strong when used for randomized algorithms. One might even use metaheuristics, even without provable performance guarantees, to find better solutions in practice.

Automatic Differentiation Automatic differentiation (i.e., algorithmic differentiation) is a methodology for computing derivatives of functions defined by algorithms [69, 92]. Automatic differentiation is the key technique underlying backpropagation for computing gradients of neural networks [25, 133], but automatic differentiation can also account for data-dependent control flow. Combinatorial problems abound in automatic differentiation. A fundamental method is so-called checkpointing [129], wherein derivatives are found with an awareness of finite storage and run-time resources on a given machine, trading one for the other, depending on the resource limitations. Figure 10 shows curves of constant effort required in obtaining a derivative by exchanging storage resources and run times. Automatic differentiation often models computation using directed acyclic graphs, and many automatic differentiation algorithms can be interpreted as graph transformations. Sparsity structure in Jacobians and Hessians is detected by using Bayesian probing [68] or related techniques and exploited by using graph coloring techniques [65].

Randomized algorithms have the potential to address many of the combinatorial challenges in automatic differentiation. For example, in checkpointing, randomization could be exploited to overcome computational resource challenges associated with storage and/or run time by exchanging fidelity in obtaining derivatives. Such an exchange is an acceptable tradeoff in the context of many optimization applications as well as in sensitivity analysis.

In many applications an approximate gradient is adequate. Randomized algorithms could lead to lower computational costs in differentiation-related computations by harnessing randomized linear algebra and randomized changes in the automatic differentiation algorithms.

2.5 Experimental Designs

Subsection lead: D. Vrabie & P.H. Zwart

Experimentation in real and computational environments forms the basis for hypothesis-driven scientific discovery. When planning laboratory experiments, changing control parameters in an accelerator or nuclear reactor, or performing any task associated with planning a new course of action to collect new data or in response to new information just collected, in combination with models for forecasting dynamic response and all other knowledge available, experimental design theory can aid in making optimal choices.

Advances in technology, such as brightness improvements in light sources or robots for automated chemical synthesis, are rapidly pushing the complexity of scientific experiments to a level where human intuition can no longer keep up with the high dimensionality of the decision landscape that needs to be explored in order to select the best possible next actions under uncertainty.

While experimental design approaches, such as Gaussian process-based strategies, alleviate some of these problems, we are rapidly encountering bottlenecks due to the computational cost associated with some of the underlying algorithms that, when implemented naively, can have computational overheads that exceed the time required to perform the experiment without advanced decision-making algorithms. Furthermore, the traditional scientist-guided approaches for selection of critical parameters that rely on human intuition could introduce bias in the experimental design and the end results.

State of the Art

The vast majority of experimental design approaches require some sort of uncertainty quantification at their core, since this uncertainty or functions thereof are the driving force that guides experimental design choices [127]. Approaches such as Gaussian processes and Bayesian neural networks provide easy access to uncertainty quantification but can incur significant overhead depending on the problem. The underlying computational bottlenecks within experimental design or autonomous experiments are typically related to matrix inversion, often recast as solving a large linear system of equations, and the global optimization of some utility function that provides guidance on the next set of actions to take. Randomized algorithms will have a major impact on these types of problems, enabling a computationally efficient exploration of decision space by balancing utility and uncertainty reduction.

Experimental design questions that are encountered can be roughly grouped in two categories: (1) *one-shot designs*, in which a data collection strategy is determined up front and cannot be changed, or only at great cost, once data acquisition is initiated, and (2) *adaptive designs*, in which new measurements have the ability to influence the data acquisition schemes.

The theory of how to design one-shot experimental designs is well established [132], with examples ranging from the design of where to place wireless 5G transmitters, traffic monitoring sensors, temperature or pH sensors in reactors, to the design of clinical trials, or to the placement of fixed-position direct radiation monitoring systems.

Autonomous decision-making systems are replacing the intuition of the scientist and can scan through the data and make smart decisions about how the experiment should proceed. This capability is critical in contexts characterized by high-complexity dynamics and high-dimensional decision spaces. In the experimental sciences, for instance, the use of adaptive experimental designs is rapidly becoming commonplace (Figure 11). Beamlines at DOE large-scale scientific user

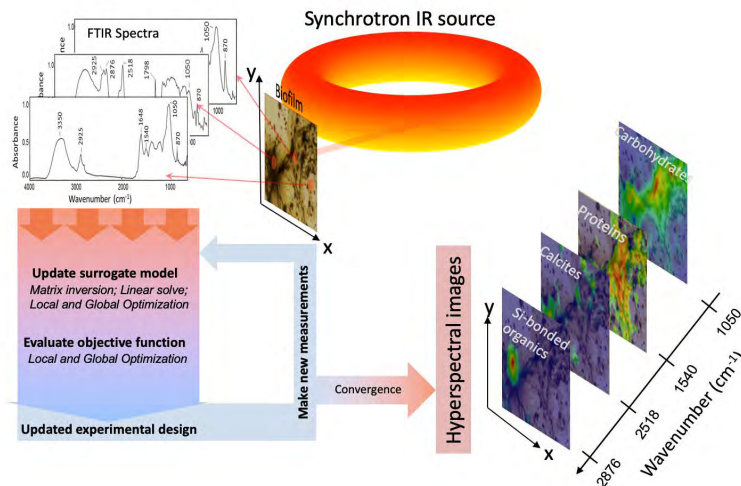


Figure 11: At the DOE large-scale scientific user facilities, real-time feedback loops provide continuous updates on experimental design, enabling extremely efficient autonomous data acquisition. Some subtasks in this workflow are expensive when using deterministic approaches and can cause a bottleneck in decision-making. Randomized algorithms can alleviate these problems, providing rapid feedback at timescales compatible with experiments that require fast response. (figure courtesy of BSISB & CAMERA, LBNL).

facilities such as the National Synchrotron Light Source, Advanced Photon Source, and Advanced Light Source are utilizing adaptive design approaches to improve the throughput and usage of scientific instruments [120, 108]. These approaches build, interrogate, and update a surrogate while an experiment is running and provide rapid feedback on how to perform measurements. In so-called self-driving autonomous laboratories this notion is pushed even further, where artificial intelligence/machine learning approaches provide suggestions on which samples need to be synthesized [128, 124]. A similar situation is encountered in running large-scale simulations, where choices of system parameters that can change the outcome of the results need to be tuned, for instance, to ensure reproducing related observational data. In all these approaches, the underlying computational complexity can rapidly escalate such that approximation methods are required to train the hyperparameters of surrogate models or efficiently interrogate these models in order to obtain new, optimal experimental design parameters.

Opportunities for Randomized Algorithms

While the majority of applications in experimental design have focused on controlling and providing feedback on continuous state variables, challenges in materials design and bioengineering require the handling of discrete and combinatorial decision spaces as well [96]. The integration of randomized algorithms approaches in these spaces will enable the deployment and integration of fast, scalable decision-making frameworks to a diverse application space. Randomized algorithms can potentially be used to construct data-driven surrogate models or to hybridize multifidelity data-driven and physical models for expensive/complex systems.

When developing new randomized algorithm approaches, the availability of formal verification of stability and probabilistic performance characteristics will be of great importance because it will provide the end user of these algorithms with correct expectations and the ability to obtain the right tradeoff between precision and run time. A thorough understanding of the error properties of randomized algorithm approaches is especially important when the cost of making a mistake is

very expensive, for instance in the control of accelerators or fusion reactors, such as tokamaks and stellarators, or when not exploring a parameter space can be costly, for instance in materials or molecular design.

The use of randomized sketching algorithms, for instance, can simultaneously regularize and reduce computational complexity without impacting the accuracy of the overall procedure. Alternatively, randomized algorithms that provide well-understood tuning options that can balance accuracy and computational complexity in a predictable fashion could be the optimal approach for cases where medium- or even low-accuracy answers are useful, as long as these answers come with an associated uncertainty estimate.

Data is the new frontier for enhancing the predictive capabilities of global-scale models of the Earth and environmental systems, understanding water cycles, and predicting extreme events. Real-time feedback from dynamic systems provides the opportunity to explore the complex decision landscapes more agilely and more effectively. Computational models powered by the world’s fastest computers must guide the experimental data collection, aid in interpreting the data, ultimately inform follow-up actions such as the design of new experiments, or serve as a basis for public policy. Randomized algorithms play a critical role in advancing the application and use of autonomous experimental designs: as the range of applications and the availability of data increase, the need for general-purpose, high-performance, well-tested, plug-and-playable algorithms and software is of paramount importance.

Randomized algorithms play a critical role in advancing the application and use of autonomous experimental designs: as the range of applications and availability of data increases, the need for general-purpose, high-performance, well-tested, plug-and-playable algorithms and software is of paramount importance.

2.6 Software and Libraries for Scientific Computing

Subsection lead: S. Wild

Numerical software and libraries are a cornerstone of scientific computing and continue to transform science [123]. Such libraries can enhance productivity of computational scientists in many ways, including by reducing development time and enabling performance portability. Libraries that include implementations of randomized algorithms would allow scientists to focus on the use and application of these algorithms for addressing grand-challenge domain-science problems. Extending the reach, reliability, and understanding of randomized algorithms for DOE’s complex software and system stacks would help realize performance gains on problems and architectures not yet imagined.

Current Status

In recent years, development of open-source numerical libraries for scientific computing has focused on addressing the challenges associated with emerging exascale computing architectures and enabling the solution of big data problems. Emerging hardware and special-purpose accelerators are also a driver and are discussed in [Section 2.7](#).

DOE’s Exascale Computing Project [75]) and SciDAC programs have centralized much of the development of production software. Community-driven efforts such as the Extreme-scale Scientific Software Development Kit (xSDK [20]) have transformed the state of interoperability among math libraries used on DOE’s leadership-class computing facilities ([Figure 12](#)).

xSDK Version 0.6.0: November 2020

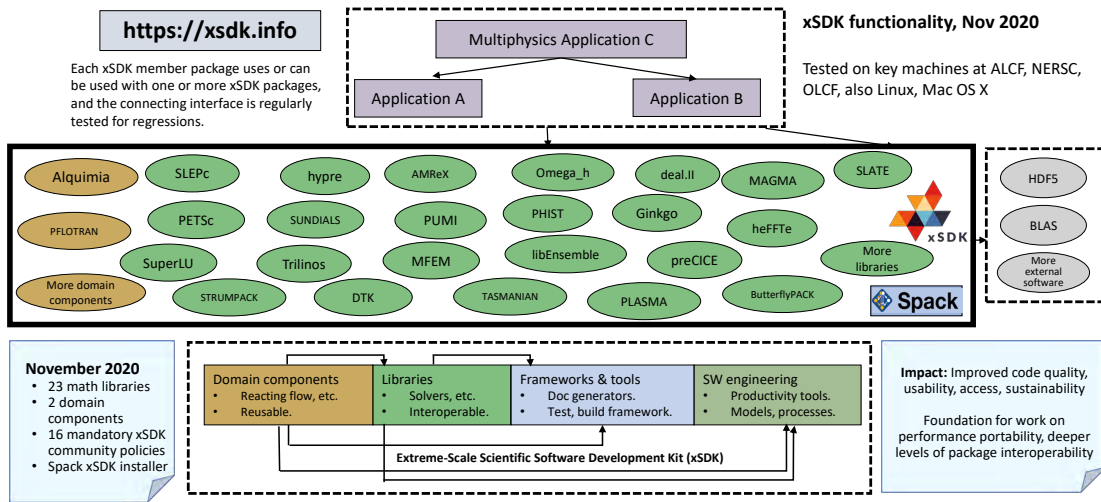


Figure 12: *xSDK* is an example of a component of the current DOE ecosystem of exascale math libraries [20]. The diverse DOE hardware and software stacks represent both consumers and providers of randomized algorithms for scientific computing.

With few exceptions (e.g., Monte Carlo-based software), production libraries in use on DOE compute systems have addressed challenges distinct from those arising in randomized algorithms. Traditionally such libraries have focused on deterministic algorithms that produce highly accurate results with provable guarantees. One expects that the results are reproducible in the sense that the output is bitwise identical each time the algorithm runs under the same software and hardware conditions. These guarantees are generally established based on a specified precision level for the underlying elementary operations, all of which are assumed deterministic. An enduring example of such a numerical setting is the LINPACK benchmark [51], which has been used to measure performance of the top supercomputers in the world for nearly three decades.

At the same time, deterministic precision levels have received significant attention. For example, recent years have seen the introduction of a veritable zoo of floating-point conventions (e.g., bfloat16, TensorFloat, fp24, PXR24) beyond traditional IEEE standards. A significant driver of such developments has been data-intensive computing and special-purpose and commodity hardware and accelerators. Similarly, mixed- and variable-precision techniques are of increasing interest [2, 71]. Exploration and adoption of these techniques are a recognition of performance gains realizable by allowing libraries and software to exploit multiple (deterministic) precision levels.

Randomized techniques have been used extensively for empirical performance optimization and software testing to find bugs [8]. The basis for these approaches is a recognition of the tradeoffs among the applicability, accuracy, and expense relative to deterministic techniques such as formal verification or analytic performance optimization.

Challenges and Opportunities

The development of libraries of randomized algorithms shares many of the challenges of producing production machine learning software frameworks for high-performance computing [29] and using randomized testing and performance optimization techniques. For example, shared modeling challenges include considering metrics and solution characteristics beyond simplistic floating-point operation-based and machine-precision-based quantities. For some problems, randomized algorithms offer the potential for accuracy levels beyond machine precision or attainable by classical deterministic methods.

For some problems, randomized algorithms offer the potential for accuracy levels beyond machine precision or attainable by classical deterministic methods.

Similarly, the trends driving hardware technology ([148, Sec. 2]) have resulted in increasingly heterogeneous and nondeterministic computing paradigms in order to extend performance gains. For example, as math libraries and software seek to avoid synchronization and mitigate faults whenever possible, the costs of preserving bitwise reproducibility has begun to outweigh the benefits for many scientific use cases. In other cases, the process of pushing the hardware and software layers to the computing fabric also results in a loss of such reproducibility. Hybrid classical and post-Moore computing workflows (Section 2.7) are further contributing to nondeterminism in computing environments.

The growing recognition of the tradeoffs between performance and achieving traditional notions of reproducibility are only a first step in leveraging nondeterminism for scientific advances and efficiency. The subtle distinction between random data (e.g., from nondeterministic hardware) and intentionally randomized operations (from a randomized algorithm) poses challenges for software development and debugging as well as validation and verification.

Standardized benchmarks for randomized algorithms are lacking, despite the fact that they have been repeatedly identified as a need for advancing progress and co-design for DOE scientific computing [137, Sec. 16]. The benchmarks would come with well-defined notions of convergence/correctness. Convergence for deterministic iterative solvers is typically described in terms of an iteration budget or residual tolerance. Bringing randomized algorithms and their associated benchmarks to a similar status would be a breakthrough in terms of enabling software-hardware co-design and facilitating optimization for diverse architectures and computing environments. Since randomized algorithms offer an approach for addressing problems where data is too large to fit in memory, benchmarks would further understanding about which algorithms perform best for given input sizes on specific systems.

The use of AI-inspired and other automated techniques to improve programmer productivity is also recognized as a DOE grand-challenge computing problem ([148, Sec. 4.1], [137, Sec. 9]). An example of such an approach is to automatically synthesize software programs based on a scientific user's intent [62]. Other uses include the automation of compilation, testing, and debugging of numerical software. The search spaces, both discrete and continuous, that arise in such problems are prohibitively large. Randomized algorithms offer a means to navigate such spaces for design goals within defined resource requirements.

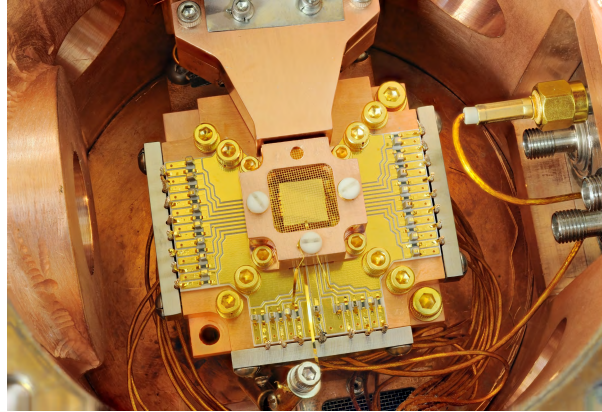


Figure 13: NIST device used for ion trapping in quantum computing (source: [https://commons.wikimedia.org/wiki/File:Quantum_Computing;_Ion_Trapping_\(5941055642\).jpg](https://commons.wikimedia.org/wiki/File:Quantum_Computing;_Ion_Trapping_(5941055642).jpg), public domain CC-PD-Mark license).

2.7 Emerging Hardware

Subsection lead: A. DeGennaro

Many of the problems of interest to applied mathematics are computationally intensive. Modern problems in optimization, uncertainty quantification, and engineering design involve evaluating the output of sophisticated computer models over high-dimensional parameter spaces. As maintenance of Moore’s law slows due to physical constraints in hardware manufacturing, it is imperative to pursue research that expands the capabilities of new computational paradigms and hardware. Quantum computing and neuromorphic computing are particular examples of such emerging paradigms. At the same time, randomization has already proven to be a powerful technique for computational acceleration, and its role in these computing schemes should be researched.

A central question for future progress will be how to co-design emerging hardware and randomized algorithms in a way that is optimized for particular tasks (e.g., speed, error-proneness).

Quantum computing [27, 125] is an emerging computational paradigm that could benefit from randomization. Quantum computers (Figure 13) currently can solve only small problems, chiefly because of noise in qubit states and quantum decoherence. Co-design of quantum hardware with randomized algorithms might help expand the size of problems that could be computed. Opportunities exist to discover and apply quantum-informed downsampling as an algorithm to load a small but statistically representative sample of a given data set onto a quantum computer. This hints at a more general motivation that quantum computing can inform classical algorithms. It also suggests that in the quantum realm, and in the classical realm as well, randomization should be used to find novel initialization schemes and more efficient methods for optimization and exploration. Randomization can help optimize quantum algorithms, such as quantum annealing and quantum Monte Carlo, and can help with the efficient solution of problems in optimization (e.g., quadratic unconstrained binary optimization) and linear algebra (e.g., eigenvalue decomposition). If successful, the integration of randomized algorithms with quantum hardware could result in significantly faster time to solution, as well as privacy preservation. Quantum supremacy could also potentially aid the solution of machine learning problems that require large data sets.

Neuromorphic computing [109, 47] is another computational paradigm that could benefit from

randomization. Graph algorithms as currently implemented on neuromorphic computers are deterministic in nature [73, 134] and could benefit from the usual speed and efficiency of randomization. Neuromorphic sensors are plagued by a significant amount of noise and variations that must be reduced. A good noise model of these sensors is clearly needed. Further research could potentially reveal a better understanding of the tradeoffs of randomized algorithms with noise and provide an opportunity to balance computation robustness with error tolerance. Randomization might also help us understand the computational capabilities of neuromorphic computers. Co-design of randomized algorithms with neuromorphic computers could lead to a novel random programming model or help researchers understand and evaluate the parameter space for neuromorphics (e.g., spike thresholds, synaptic delays). If successful, neuromorphic computers could lead to better algorithms for emulating random processes and solving stochastic/partial differential equations. They could also lead to fast computing algorithms with low memory requirements associated with data.

2.8 Scientific Machine Learning

Subsection lead: S. Wright

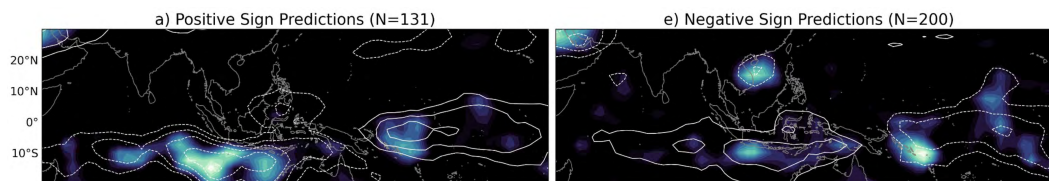


Figure 14: Interpreting a subseasonal forecast: Light areas show the geographical regions most relevant to the forecast. (from [106, Figure 4].)

The use of machine learning techniques in scientific computing has a long history dating back to the 1990s and earlier. The SIAM Conference on Data Mining, held every year since 2001, has always had a strong focus on science and engineering applications and on connections to high-performance computing [91]. In January 2018, a DOE ASCR workshop and report [18] identified six Priority Research Directions for Scientific Machine Learning (SciML) that highlight basic research challenges such as:

1. Science and engineering applications often make use of detailed models, based on laws of physics, chemistry, and biology, that enable detailed simulations to be performed and useful predictions to be made. The “model-free” ethos that pervades machine learning—the idea of “letting the data speak for itself”—is not naturally compatible with the use of physical models. However, there is increasing interest in making current machine learning techniques “play well” with physical models—augmenting, enhancing, and complementing physical models in ways that potentially reduce computational requirements while maintaining adequate fidelity to scientific laws.
2. Even in their most successful applications, including speech and image recognition, machine learning models are susceptible to perturbations in input data and parameters. That is, their predictions can be affected strongly by minute changes to the data. *Robustness* of these models reduces such sensitivity and is essential to scientific applications where model outputs that are obviously invalid would reduce the credibility of machine learning methodology.
3. It is particularly important in scientific applications for models to be *interpretable*—for their simulations and predictions to accord with prior knowledge (see, e.g., Figure 14). Since the

applications can be mission-critical, *trustworthiness* is another essential property; the outputs and predictions must be reliable. An example of the latter phenomenon is that when a machine learning model is presented with data that is outside the scope of its training data, it is able to flag that data as being “out of distribution” and issue a warning that the predictions may not be trustworthy.

Machine learning enhancements to physical and biological models can be useful in “plugging gaps” in existing composite models, using data-driven machine learning models in those parts of the system for which the physics is not adequately known. But even in cases where the physics *is* known, machine learning can still play a role in surrogate models that can be executed more cheaply as part of optimization, control, and inversion processes. Such surrogates can be trained using data generated by high-resolution physical models—an expensive process, but one that can be done “offline” and in a way that makes use of massively parallel computing. Surrogates of this type already have been used in fields such as Earth science [24]. Better understanding is needed at an abstract level of how physical and machine learning model components can be composed in ways that are efficient and serve the uses of the overall model. Randomness plays an important role in generating data to train machine learning surrogates from physical models, accounting for the uses to which the overall model will be put (for example, optimization, control, inversion). Potential benefits of this improved machine-learning-enhanced modeling methodology are vast and include accelerated scientific discovery in transistor design, materials discovery, and aerospace engineering.

In scientific applications, the outputs of machine learning models must be valid and trustworthy, with quantified sensitivity and uncertainty and with interpretable behavior. The scientific computing community includes generations of computational scientists with wide and deep experience in modeling important processes. Machine learning models whose outputs conflict with this experience are unlikely to be trusted by these scientists. Techniques for improving interpretability and quantifying uncertainty are active areas of research in the machine learning community, but the existing community of scientific computing people must be engaged in order to ensure that the results of this work are meeting their standards of quality. The challenges include high dimensionality, nonlinearity, and nonconvexity of the models, all of which make it difficult to sample the uncertainty in ways that are both theoretically and practically valid. Randomization can drive sampling strategies and help reduce the effective problem dimension. Techniques for exploring high-dimensional parameter spaces (based, for example, on Bayesian neural nets) are under investigation.

Robustness of the outputs of models to perturbations is essential in many mission-critical applications (e.g., reactor control). Machine learning models are known to be sensitive to perturbations in their inputs and to their learned parameters. An area of active investigation for the past decade has been on improving the robustness of machine learning models to such perturbations. Various approaches are being investigated, including dropout in neural network training, bagging, bootstrapping, and adversarial training. Randomization can help by generating augmented training sets, and also in the form of stochastic differential equation-based analysis leading to model outcomes that are distributional rather than point estimates.

Randomness already plays an essential role in machine learning (the optimization algorithms used to train neural nets incorporate randomness, for example). It plays an important role, too, in resolving the issues cited above, in ways that bring the benefits of modern advances in machine learning to scientific computing.

Randomness will be essential to the design of the next generation of machine learning models, facilitating robustness and reliability with respect to perturbations in model parameters and data.

Potential research directions include the design of stable network architectures [70, 60], novel noise injection methods for training robust machine learning models, randomness as a resource to introduce implicit regularization, randomness for data augmentation and robust training [45], and randomness as a strategy for computing distributional estimates rather than just point estimates [143]. Such innovations are key to enabling deployment of machine learning models in mission-critical scientific applications.

Better understanding of the randomized optimization algorithms that are at the heart of scientific machine learning (and in fact all of machine learning) will be vital to future progress. The basic analysis of such algorithms makes assumptions that do not hold true in practical situations. Scientific machine learning requires solution of nonconvex, nonsmooth problems in which the randomized gradients do not satisfy an “independent identically distributed” (i.i.d.) property. Although progress has been made in understanding the algorithms under these conditions and in analyzing scaled stochastic gradient approaches such as ADAM, much foundational work remains to be done.

3 Current and Future Research Directions

Having reviewed the wide range of application drivers for randomization in scientific computing, we now cover the research directions that must be pursued in order to enable randomization as a first-class tool and a driver of progress in scientific computing. One of the recurrent themes throughout this section is increased emphasis on practicality, whether it is sharpening complexity bounds by reducing the constants and other lower-order terms, determining precise sketch size or sampling rates based on application needs, or integrating randomized algorithms into coupled workflows.

In many cases, the existing theory beyond popular randomization techniques, such as sketching or sampling, is too general and does not provide tight enough bounds for scientific computing tasks. By restricting the problem domain or structure, greater statistical and computational efficiencies can be gained, which will greatly broaden the applicability of randomized methods to scientific computing.

Several subsections reiterate the importance of making the randomization itself computationally efficient in practice, beyond the big- O notation. Since most scientific computing tasks are truly large scale and require massively parallel computing, emphasis is also placed on the coupling of randomization and parallelization. Moreover, the community suggests that these computational aspects of randomization be captured through software abstractions for wide adoption, availability, portability, and high performance.

3.1 Opportunities for Random Sampling Schemes

Subsection lead: K. Myers

As we see throughout this report, random sampling undergirds and enables many other types of randomized algorithms. From the Monte Carlo methods used to generate random samples in many forward models (Section 2.2) to stratified sampling and graph sampling approaches for discrete processes (Section 2.4) to the desire for data reduction or compression in the context of massive data generators and quantum computers (Sections 2.1 and 2.7), random sampling is the key component of many scientific computing advances.

Likewise, this report showcases several cutting-edge scientific areas that are faced with higher volumes or rates of data than ever before (see Section 2.1). Experts need answers more quickly than is possible with the current state of the art. Random sampling offers the promise of tractable analysis “downstream” from data-generating mechanisms, whether they be exascale simulations, experimental data from high-throughput user facilities, or opportunistic measurements from sensors with ever-increasing data rates. At the same time, we must have assurance that the random sample will retain the relevant characteristics of the original data sets. Without this assurance, we cannot trust the conclusions.

Random sampling offers the promise of tractable analysis “downstream” from data-generating mechanisms, whether they be exascale simulations, experimental data from high-throughput user facilities, or opportunistic measurements from sensors with ever-increasing data rates.

Furthermore, we need sampling schemes that are themselves computationally tractable in the presence of large and/or streaming data. Ideally we want efficient sampling methodologies that ensure accuracy in the solution with minimal computational effort. An added challenge in the context of complex simulations is that we may need to control the computational cost of the sampling scheme before we know the available computational resources, which could change as the simulation evolves.

To illustrate some of these concepts in the context of a scientific data set, Figure 15 presents sampling schemes explored and developed under the ECP [75]. Here the focus was the Deep Water Impact Ensemble data set [122], a set of simulations produced on a regular grid and used to study asteroid-generated tsunamis. Panel (a) shows a volume-rendered visualization of the simulation’s water fraction variable, showing the plume of water generated after an asteroid has hit the surface of the ocean.

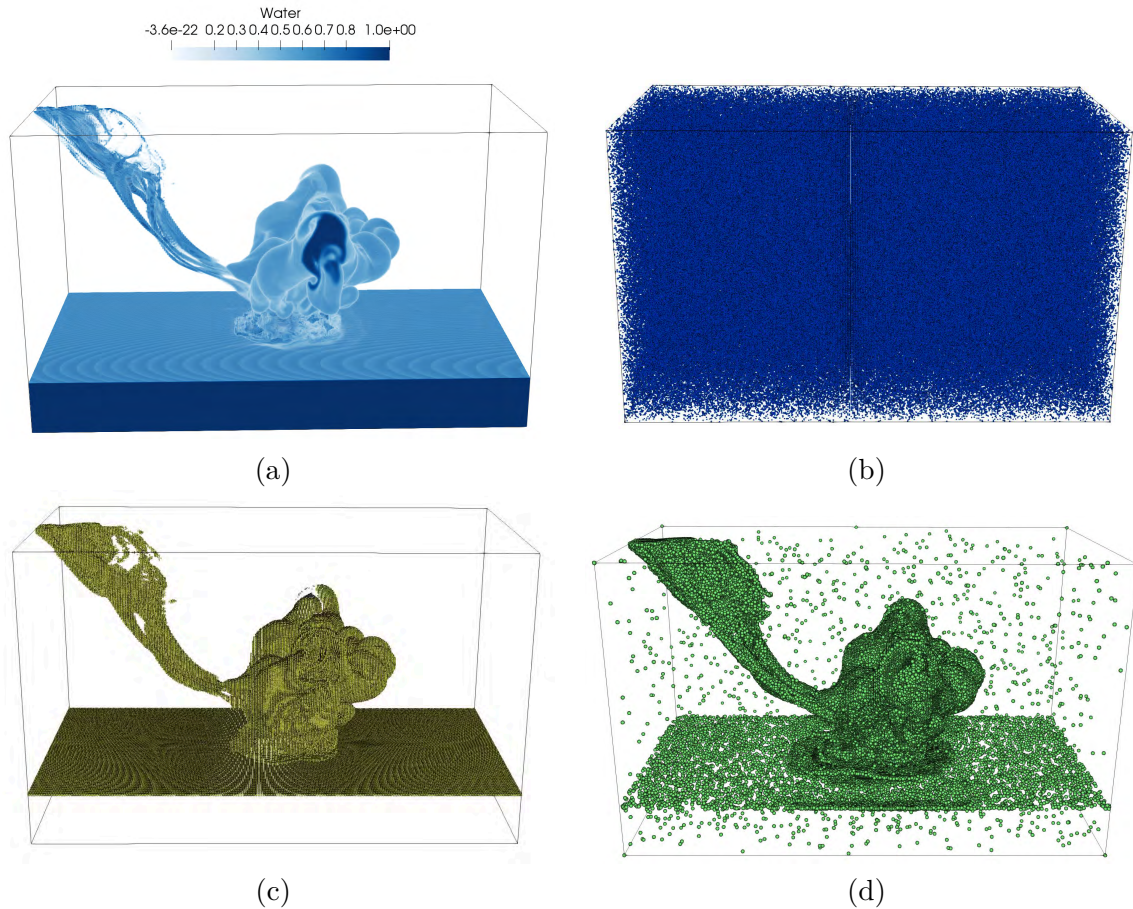


Figure 15: Sampling schemes applied to an ensemble of simulations run on a regular grid to explore asteroid-generated tsunamis. (a) Volume-rendered visualization of the simulation’s water fraction variable, illustrating the plume of water generated after an asteroid has hit the surface of the ocean. (b) Simple random sampling scheme in which every simulation grid point has an equal probability of being included in the sample regardless of the underlying scientific content. (c) Local-smoothness sampling. (d) Joint multicriteria sampling. Both (c) and (d) are data-driven sampling approaches that reveal the features of scientific interest. All three sampling schemes use a sampling ratio of 2%. The color schemes in each panel are used to indicate that each panel presents a different sampling method. (adapted from [31]).

Suppose our storage budget allows us to save only 2% of the grid points of the simulation. We need to generate a sample of the simulation that will support post hoc analysis. Panel (b) of Figure 15 shows an example of a simple random sampling scheme, where each simulation grid point has the same probability of being included in the sample regardless of what underlying scientific content may be conveyed by that grid point. This is easy to compute but yields a uniformly distributed collection of points that loses the scientific features of interest. In contrast, panels (c) and (d) show two different data-driven schemes for selecting 2% of the original grid points that were explicitly

designed to capture salient features of a specific scientific data set in a computationally tractable framework [31].

More developments are needed in these directions, especially as more data-intensive applications come online and data rates continue to accelerate. Here we discuss several promising research directions.

Computationally efficient sampling Efficient sampling of data offers many important research opportunities, particularly in the context of massive and/or streaming data sets. These include how to adapt to the geometry of data, achieve robust nonlinear dimension reduction, identify sparse representations (e.g., intrinsic low-dimensional structures), filter noise, and identify anomalies. An exciting research direction along these lines is adaptive sampling. In the context of user facilities and other experiments, this methodscan address the question of where to sample next in order to gain the most information, allowing scientists to find the “needle in the haystack” under highly dynamic conditions. In optimization, some adaptive sampling approaches use varying sample sizes to gradually reduce the variance in the stochastic gradient. These enable optimal balance between the computational burden and the accuracy of approximated information.

Stratified and topologically aware sampling A concern is that naive sampling techniques can miss small but important subsets of data. Stratified sampling—strategically partitioning data into classes to which we assign a sampling distribution—can address this. However, many partitioning techniques assume that data points that are geometrically close to one another are similar, a situation that is not always true. Topological data analysis tools allow us to construct graphs from data where relationships are informed by more than distance.

Scientifically informed sampling For assuring that the sampled data set retains the salient characteristics of the original data set, an interesting challenge is that the salient characteristics could differ depending on the scientific questions of interest. For instance, if the interest is in recognizing the occurrence of rare events in a massive data set, a Monte Carlo sampling scheme might produce some samples with no instances of that event, leading to an underestimate of their occurrence, and other samples with one or more instances, leading to an overestimate. On average the Monte Carlo samples will have the correct proportion, but any given sample could be far from the truth. In that situation, importance sampling [121] may be more appropriate because of the particular interest in rare events. This sort of situational responsiveness demands the development of sampling methods that are informed by the scientists.

Reproducibility The idea that different samples could have different characteristics even when generated from the same sampling scheme leads to important questions about reproducibility. Will the scientific results be different if a different set of samples is used? Scientists are unlikely to use analysis algorithms that give widely different answers for different random samples. To address this concern and to evaluate whether data samples are useful to the scientist could require research in information theory or theoretical computer science. Success here would lead to greater acceptance of randomized algorithms, more confidence in science results, and fewer false discoveries.

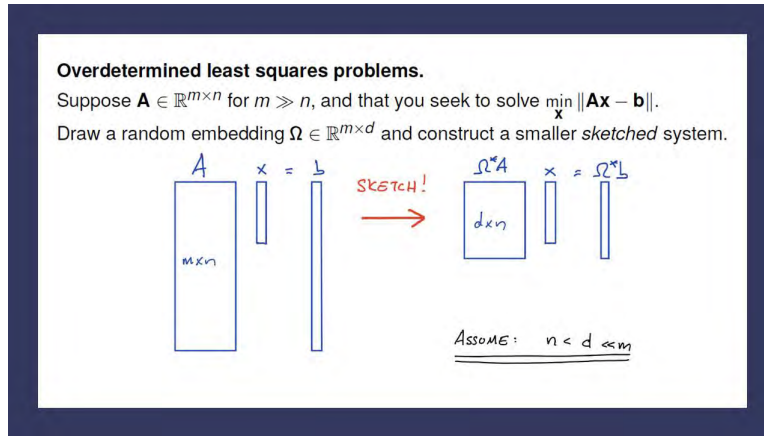


Figure 16: Matrix sketching in the context of regression, using a random embedding to create a smaller problem. (image from the RASC workshop presentation of Per-Gunnar Martinsson, December 2, 2020.)

3.2 Sketching for Linear and Nonlinear Problems

Subsection leads: T. Kolda & P.G. Martinsson

Sketching is a mathematical technique wherein a large problem or data set is replaced by a much smaller “sketch” that retains essential properties. Counterintuitively, the size of the sketch can be *independent* of the size of the original problem, meaning that the cost savings can be better than exponential. Linear sketching has been applied successfully in many scenarios, including regression (Figure 16) and low-rank factorization [72, 100, 153, 104]. As a relatively new technique, the most convincing DOE applications of sketching have been in numerical linear algebra. More broadly speaking, sketching is widely used in industry for counting unique elements, estimating quantiles, or detecting frequent items in massive data sets or data streams (e.g., [130]). Looking forward, sketching promises to be a key tool in areas including solution of large-scale nonlinear inverse problems, PDE-constrained optimization, solution of linear and semidefinite programmer solvers, and quantum chemistry.

Counterintuitively, the size of the sketch can be *independent* of the size of the original problem, meaning that the cost savings can be better than exponential.

State of the Art

A prototypical problem is linear regression, fitting an n -dimensional linear model to a set of m observations where the number of observations is orders of magnitude larger than the number of dimensions ($m \gg n$). We let $\mathbf{A} \in \mathbb{R}^{m \times n}$ denote the matrix of m observations, \mathbf{b} be the corresponding right-hand side and \mathbf{x} be the solution. Since the problem is overdetermined, we cannot in general find a solution that solves the problem exactly. Instead, we seek a *least-squares* solution that yields the minimum square error, namely, $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2$. When the coefficient matrix \mathbf{A} is large, the problem of finding the minimizer can be accelerated by forming a much smaller “sketch” of the full system, produced via a sketching matrix $\Omega \in \mathbb{R}^{m \times d}$, so that the sketched system $\min_{\mathbf{x}} \|\Omega^T \mathbf{Ax} - \Omega^T \mathbf{b}\|^2$ has only $d \ll m$ rows and is much more efficient to solve (Figure 16).

Many ways can be used to create the linear sketch Ω . A common approach is to choose Ω to have random entries drawn from a standard normal distribution. Such Gaussian sketches are fast and

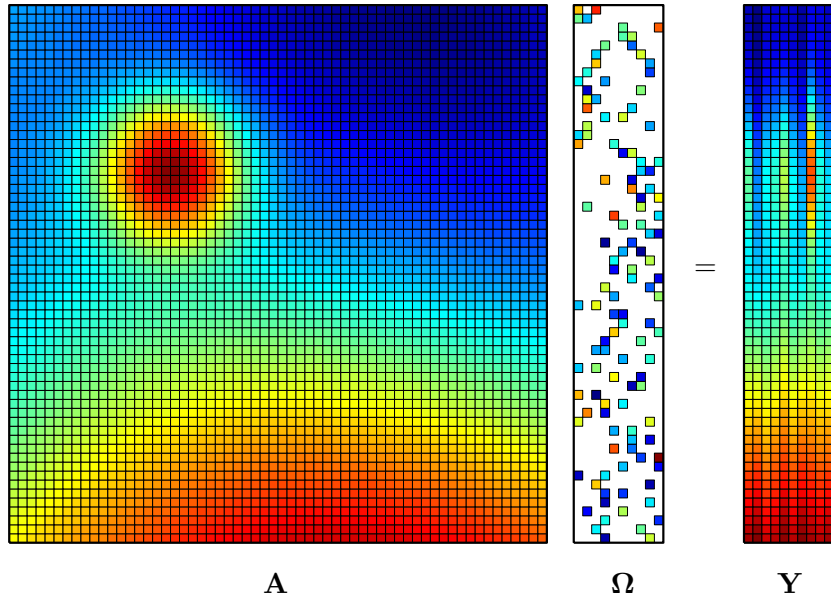


Figure 17: Randomized sketching in linear algebra: Given a matrix \mathbf{A} , a compressed sketch \mathbf{Y} is formed by applying \mathbf{A} to a tall, thin random matrix $\mathbf{\Omega}$. When $\mathbf{\Omega}$ is drawn from a “good” distribution, the sample matrix \mathbf{Y} contains all the information required to compute an approximate basis for the column space, to find a set of rows of the matrix that approximately spans its row space, and to accomplish many other tasks.

reliable and yield the smallest possible sketch: the size of d is as small as theoretically possible. For the price of a slightly larger sketch, further acceleration is possible by using random maps that are sparse (Figure 17) or have other internal structure that enables their application using highly efficient algorithms such as fast Fourier transforms [44, 72]. For instance, fast Johnson–Lindenstrass transforms [7, 97] employ this strategy. Specialized sketches have been developed for the case that \mathbf{A} is sparse, using specialized sampling strategies, such as leverage-score sampling, that interact with only a subset of the data [44, 100].

In some environments, the minimizer of the sketched system can serve as a good approximation to the minimizer of the original problem, referred to as the “sketch-to-solve” regime. Using the solution to the sketched system directly can lead to dramatic acceleration, but the error can be bounded only when the properties of the original system are a priori well understood. Alternatively, the sketched system can be used as a preconditioner that ensures rapid convergence in an iterative solver for the original problem. Such a “sketch-to-precondition” approach has proven to be powerful in accelerating practical computations and has a particular advantage in that this solver is 100% reliable, since the computed solution is *guaranteed* to fit the data well [131, 16].

Another successful application of matrix sketching concerns low-rank approximations of matrices. The idea is to use linear sketches to compute approximate bases for the row and/or column spaces, (cf. Figure 17). Once these have been constructed, all further computations can be executed on the small sketches. Algorithms of this type have proven to be highly communication and storage efficient and excel in severely communication-constrained environments such as GPU computing or when data is stored out of core [154, 104].

Other recent examples of sketching include a two-stage Gauss–Seidel preconditioner with a randomized and asynchronous version of the Gauss–Seidel preconditioner developed by Avron et al. using a graph Laplacian problem as a probe, randomized pivot selection in QR [53, 54, 105], accel-

erated tensor decomposition [139, 22], and even very recent and potentially groundbreaking work in semidefinite program solvers [156].

Training of large-scale machine learning methods has led to many new and popular randomized methods for optimization, such as AdaGrad [52] (with approximately 9,000 citations per Google Scholar as of this writing) and ADAM [89] (with over 60,000 citations). Despite its popularity, the convergence of ADAM is not yet well understood.

Future Challenges

Many open research problems remain. Here we mention a few exemplars. In each case the problem requires domain expertise to understand the accuracy requirements and special structure. Theoretical and statistical analyses are needed to, for instance, determine the required size to obtain the required accuracy or to develop appropriate sampling strategies. Implementations need to be adapted or rewritten, especially to realize reduced communication costs.

Incorporating sketching for solving subproblems Most large-scale simulations solve a sequence of linear systems to find a solution, and these linear solvers are the primary bottleneck. Consider applications in high-frequency electromagnetic scattering and strongly advective advection-diffusion-reaction systems. Sketching is a promising tool, but foundational questions need to be answered. What is the size of the sketch that is required in order to guarantee the needed accuracy in the overall simulation? Can smaller sketches be used in earlier iterations where less accuracy is needed? Alternatively, consider a problem with multiple subsystems as in multiscale problems. Can smart sketching yield improved approximate solutions at some scales? In most cases, the answers will be application specific and perhaps even problem specific. In the context of ill-conditioned inverse problems, the modes associated with small singular values are important because these are actually large when viewed from the perspective of the inverse. Thus, some randomized algorithm ideas associated with ignoring small singular values are not directly applicable. We do know, however, that many subblocks of the matrix inverse can often be approximated by low-rank operators. Some hierarchical basis methods can already exploit this property, but further research into these types of algorithms should be expanded. An interesting and challenging question is how one can detect subblocks where it is appropriate to employ low-rank approximations via randomized algorithms. This problem has been solved in some specific cases, but for more general matrices this is significantly less understood.

Specialized randomization for structured problems Greater efficiencies (i.e., in the form of smaller or easier-to-compute sketches) can be realized by exploiting problem structure. For instance, can we exploit the dependency grid structure of PDE solvers to come up with randomized variants of multilevel preconditioners? Can we exploit Kronecker structure in quantum structure calculations? How does doing so impact efficiency and robustness? This could potentially speed up the setup phase of an algebraic multigrid solver considerably in the context of extreme parallelism.

Overcoming parallel computational bottlenecks with probabilistic estimates In parallel computing, the cost of floating-point operations is negligible compared with communication costs. Sketching can greatly reduce communication costs, and it should be feasible for certain applications to ameliorate the loss in accuracy with inexpensive extra iterations. As another application, global reduction operations are bottlenecks. However, one may be able to distribute the data such that responses from only a subset of the compute nodes are adequate to guarantee sufficient accuracy.

Randomized optimization for DOE applications In machine learning, stochastic optimization is standard practice. Such methods compute an inexpensive stochastic gradient by using only partial information. Can such methods be employed in the context of DOE applications based on large-scale simulations? For instance, perhaps the stochastic gradient employs only a subset of the grid points. This situation is not dissimilar to multigrid methods, except that those are deterministic and used only in the context of linear solves. Can the computational burden of large-scale partial differential equation optimization be reduced while providing the same kinds of guarantees on accuracy or uncertainty quantification? A potential advantage of randomized approaches is removing dependencies on outliers in data integration optimization tasks. While randomized algorithms are powerful, many popular ones (e.g., ADAM) are not well understood: they work, but it is not always clear why, how, or under what circumstances. As computing resources and applications compel more use of such algorithms in high-performance scientific computing, it is vital that these algorithms are understood from a theoretical and quantitative standpoint. Significant efforts in the theoretical foundations are necessary in order to characterize, measure, and understand those computational outputs, which are distributional in nature.

3.3 Algorithms for Discrete and Combinatorial Problems

Subsection lead: A. Buluç

We organize the major research themes in randomized algorithms for discrete problems in five distinct themes. The overarching goal of randomization here is finding scalable ways to sample, organize, search, or analyze very large data streams and discrete structures on finite resource machinery. In this section, all connected structures such as graphs and their high-level counterparts (e.g., hypergraphs and simplicial complexes) are collectively referred to as “networks.”

Randomized algorithms for discrete problems that cannot be modeled as networks

Many important discrete problems cannot, or need not, be represented as graphs or their generalizations. For example, randomized techniques have been successfully used for routing in modern supercomputers [88] and load balancing in various settings [114]. As the concurrency increases to extreme scales, these methods will find more and more use in scientific computations. Another area where discrete non-graph problems arise is the analysis of sequencing data. For example, randomized algorithms are used to find compact index structures [59], which are crucial for efficiently comparing large sequencing (DNA, RNA, or protein) data sets. Exponential rise in sequencing data that has been outpacing Moore’s law is a pressing reason to adopt these randomized algorithms widely in practice.

Randomized algorithms for solving well-defined problems on networks

Randomized algorithms are used to provide approximate solutions to many subgraph counting problems with errors diminishing with sample size [82]. Various modifications to the celebrated color-coding technique of Alon et al. [10] have been used for this purpose. Some randomized graph algorithms are known for problems that require exact optimality, such as the min-cut [84] and minimum spanning tree [85] problems. However, several open issues impede the adoption of these clever techniques. While randomized algorithms often match or exceed the complexity bounds of the best deterministic problems in the worst case, they sometimes fail to match the performance of the best deterministic problems on real inputs in practice. A common reason is that existing randomized algorithms are designed to perform the same number of operations regardless of the input, making the common case as slow as the worst case. This situation is exemplified in Karger’s algorithm for

minimum cuts, whose “primary misfortune is that it always runs in its worst-case $O(n^2 \lg n)$ time bound” [40].

Adoption of more realistic complexity measures by the community (Section 3.5) will help close the gap between theory and practice. An obstacle to adoption of randomized methods is scalability to large concurrencies, especially on distributed-memory architectures where most big science computations are performed. New research demonstrating scalability of parallel randomized algorithms for important discrete problems will ignite the interest of domain scientists on randomized methods. Furthermore, the ability to run on a streaming setting is crucial for processing data coming from experimental facilities.

Universal sketching and sampling on discrete data The technique of “graph sketching” refers to working on a subset of either nodes or edges from a much larger graph to draw a conclusion. Sketching can be achieved by using various forms of graph sampling methods. A popular graph sampling strategy is based on random walks, as shown in Figure 18. The theory of sketching and sampling, especially in the streaming setting, is often phrased in terms of specific algorithms that solve prespecified questions, such as frequent elements. In practice, the questions are often determined *after* generating a sketch of the data or data stream. A theory of universal sketches, where streaming and data analysis algorithms can answer a large variety of questions, needs to be developed.

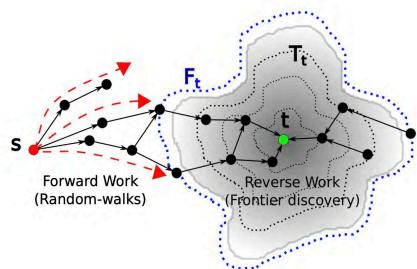


Figure 18: Example of randomization on graph traversal. The FAST-PPR algorithm, a fast personalized PageRank algorithm, uses careful random sampling to find relevant vertices in a massive network [98].

With the explosion of high-volume and high-velocity data, the extraction of information has become a serious challenge. Ample research opportunities exist for exploring, determining, and optimizing how randomization may prove fruitful in achieving scalability and high performance in network and topological data analysis. Furthermore, the practicality of these algorithms is of paramount importance for their adoption in scientific computing. The theory of graph sampling, particularly as it concerns statistically nonstationary and time-dependent graphs, is rich with questions that have implications on the practical side of proposing randomized algorithms to search, sample, explore, and reduce networks.

Much of the existing literature provides bounds of the following form: Given accuracy and confidence parameters, necessary mathematical bounds exist on the sketch size and memory footprint. In practice, the situation is inverted. There is a fixed memory budget, and users need to get the “best possible” answer. A research opportunity exists to develop bounds of the latter form. Moreover, existing theory focuses on asymptotic results ignoring constant factors. For practical applications of the theory we need a more precise theory that tackles the constant factors involved. The purview of challenges in streaming algorithms extends to edge computing as well as distributed

computing.

Various forms of subgraph sampling are key to performing efficient training for graph representation learning. The methods currently used often lack generality, and their computational complexity is poorly understood.

Randomized algorithms for combinatorial and discrete optimization A significant portion of problems in combinatorial optimization suffer from exponential (or worse) complexity using traditional methods [152]. In practice, the situation is often exacerbated by the presence of nonlinear and nonconvex constraints, resulting in a lack of algorithmic scalability that even the most advanced high-performance computing platforms fail to overcome. Randomized algorithms can help overcome the scalability challenges in combinatorial optimization, especially if provable approximation guarantees are provided. A specific technique is randomized rounding for stylized combinatorial optimization problems. Randomization can also help solve PDE-constrained optimization problems arising, for example, in the control of additive manufacturing processes for a given control trajectory (combinatorial component).

Randomized algorithms for machine learning on networks Geometric deep learning [38] is often the umbrella term for various machine learning techniques on unstructured connected data, with the prime example being graph neural networks. Various forms of subgraph sampling are key to performing efficient training for graph representation learning [74]. The methods currently used for this purpose often lack generality, and their computational complexity is poorly understood. Developing a robust theory of sampling graphs, hypergraphs, and other discrete structures is a key research direction. Furthermore, we need to understand how scientific goals relate to existing discrete and graph sampling techniques that are being employed and develop methods to quantify the effect of these sampling techniques on the scientific goal. Going beyond simple graphs that are characterized by pairwise interactions and generalizing these methods to higher-order structures [23] are another key research direction.

3.4 Streaming Algorithms and Data

Subsection lead: J. Nelson

A sketch of a data set \mathcal{D} is simply a low-memory data structure to support answering any query from some given family of queries (see Section 3.2). A primary goal is to achieve a sketch size that is sublinear in $|\mathcal{D}|$, the size of \mathcal{D} . A *streaming algorithm* is simply a sketch that supports dynamic data; in other words, the data structure should be able to process a stream of updates to \mathcal{D} , during which the sketch should be updated on the fly.

The earliest and perhaps simplest streaming algorithm is the probabilistic counter of Morris [115] to count the number of events in a data stream using very few bits. This can be useful for sensors or edge computing where there are extremely limited hardware or energy resources on device. The Morris algorithm maintains a counter of up to N values subject to a single operation: *increment* (Figure 19). Whereas a counter that is exact must use $\Omega(\log N)$ bits of memory, Morris leverages randomization to develop his approximate counter (which reports the counter value N up to approximately 1% multiplicative error with at most 1% failure probability) using only $O(\log \log N)$ bits of memory—an exponential improvement. Indeed, for many streaming problems both randomization and approximation are necessary in order to obtain sublinear memory [11].

Goal:	Count up to $N = 100000$ (streaming) events using an 8-bit counter
Method:	Initialize $\eta = 0$ to be the 8-bit value. Increment η probabilistically according to the following procedure: <ul style="list-style-type: none"> • Let ξ be a uniform random value in $(0, 1)$ • If $\xi < (a/(a+1))^\eta$, set $\eta = \eta + 1$
Result:	$\hat{n} \equiv a((1 + 1/a)^\eta - 1) \approx n$ with error $\sigma^2 = n(n-1)/2a$

Figure 19: Morris’s Probabilistic Counter [115]

In early literature on streaming algorithms in the late 1970s to the mid-1990s, motivations ranged from wanting to study a crisp algorithmic model out of intellectual curiosity, without regard to practice, to wanting to use low-memory data analytics in applications such as network traffic monitoring and databases [115, 116, 113, 63, 11, 12]. More recently, streaming algorithms have found their way into computational linear algebra [153], machine learning [66], and state-of-the-art optimization algorithms for problems as fundamental as linear programming [147].

A remarkable illustration of how randomization enables streaming algorithms has been the discovery of techniques for computing an approximate low-rank factorization of a given matrix or tensor in a single pass over its entries [104]. Traditional techniques for computing such a factorization, for example, Krylov methods or Gram–Schmidt orthogonalization, require multiple interactions with the matrix and cannot be deployed to matrices that are too large to be stored. In contrast, randomized sketches (as illustrated in Figure 17) of the row and column spaces of a matrix can be extracted in a single pass, and one can reconstruct the matrix using only the information contained in these sketches. These new algorithms have the potential to dramatically enhance our ability to store and analyze gigantic data sets arising in applications such as turbulence modeling and molecular dynamics.

More specific to current DOE interests, with rapid increases in computing power, modern scientific simulations generate high-fidelity data that is outpacing our ability to write this data to disk for later analysis. Similarly, rapid advancements in sensor technologies create storage and analysis bottlenecks for DOE scientific facilities. Thus, a high-priority research area for the DOE Advanced Scientific Computing Research program is the development of robust, efficient, and scalable algorithms for dimensionality reduction and/or data compression of streaming scientific data. Challenges that must be overcome include accurately quantifying uncertainties in such representations arising from both the data and stochastic approximations inherent in randomized algorithms, making the algorithms robust to hyperparameter tuning to ensure their effectiveness for real DOE scientific problems, making the algorithms efficient enough in terms of computational complexity and software implementation for in situ and/or online application, and porting the algorithms to emerging computing architectures that emphasize high-bandwidth streaming computations over random accesses that are common in many randomized algorithms. If successful, such techniques would dramatically increase the throughput of DOE’s simulation and data acquisition workflows, thereby quickening the pace of scientific breakthroughs.

Also of interest is the analysis of large structured data such as graphs or matrices, which are a key component of data science workflows that becomes expensive in distributed memory, usually because of memory and especially communication overhead.

Randomized streaming and especially sketching algorithms can approximately summarize, e.g., vertex neighborhoods [107] or matrix rows [43] in small memory for subsequent communication and analysis on other compute nodes or client hardware—affording approximation of quantities of interest with reduced latency by dramatically limiting communication overhead.

High-performance computing algorithmic pipelines can use sketches to perform numerical linear algebra, local structure approximations in graphs, dimensionality reduction, and nearest-neighbor computations, among other key data science tasks. Such approximations and latency improvements, along with performant and user-friendly software, are necessary in order to make high-performance computing resources accessible and useful to nonexpert data scientists across many scientific domains of interest.

Mergeable summaries One useful technique for distributed processing of streaming data is that of using *mergeable summaries* [3]. A fully mergeable streaming algorithm is one that allows several different streams to be processed separately so that the resulting sketches can be merged in an arbitrary merge tree with no degradation in accuracy or confidence to obtain a sketch for the union of all datasets. Such algorithms are important for minimizing communication (only sketches need to be communicated) when data is naturally distributed across a network. They are useful even when data is not distributed because they allow for a divide-and-conquer approach to obtain parallel algorithms.

In situ and real-time data analysis As discussed in Section 2.1, in situ and real-time data analysis are necessary in order to keep pace with increasing rate, size, and complexity of streaming data in national science user facilities. Streaming algorithms are also needed to train or update models in an online fashion as more data is seen.

Privacy In several applications, data is streamed in from multiple sources that would like to maintain privacy against the central server processing the data. For example, consider Apple wanting to automatically learn words for its spellchecker dictionary by monitoring words typed by iPhone users, while guaranteeing user privacy so that Apple itself cannot determine which users typed which texts [142]. A formal definition of privacy preservation is given by *differential privacy* [55] in which a database is preprocessed into a randomized output that is statistically nearly indistinguishable from what would be output had any single user’s data been removed. In traditional differential privacy this data randomization is performed by a trusted central server, but the example given here shows the necessity of development of solutions in the so-called local model, where data is distributed and the central processing server is untrusted. Several recent works have given solutions to specific tasks in this local model, as well as a newer shuffle model [41], but this direction is still in an early stage of development.

3.5 Complexity Analysis

Subsection lead: M. Aniteșcu

One of the main drivers of this document is the strikingly reduced complexity (e.g., dependent primarily on the rank rather than the dimension, in the case of singular value decomposition) that randomized algorithms achieve in some notable circumstances. Nevertheless, many advances are still required in order to understand when and how to use such algorithms and to sharply quantify their performance and limitations. For instance, even for the fairly basic cases of linear systems and

Randomized algorithms come with different degrees of acceleration and reliability

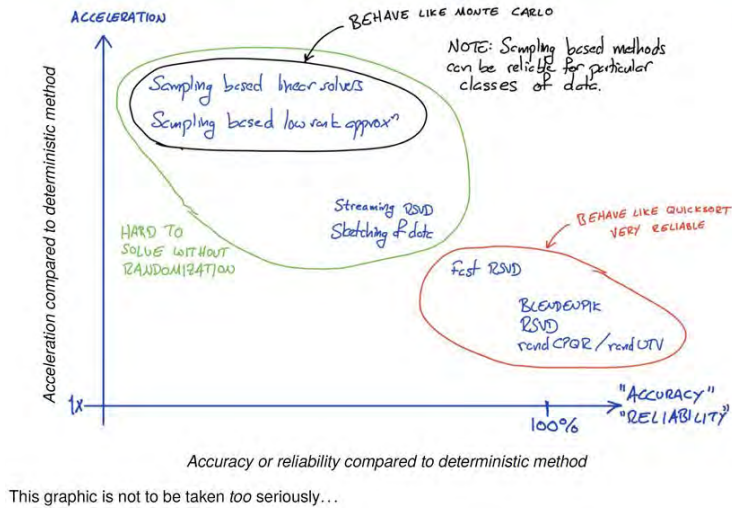


Figure 20: Very rough sketch of the landscape of accuracy and acceleration for existing randomized algorithms for linear algebraic problems. Some algorithms can be 100% accurate and essentially risk-free, even improving robustness. For massive problems, small sacrifices in accuracy may yield orders of magnitude acceleration. (image from the RASC workshop presentation of Per-Gunnar Martinsson, December 2, 2020.)

matrix approximations, a spectrum of randomized algorithms exists, some that are very accurate but only 2–10 times faster than deterministic methods and others that are less accurate but many orders of magnitude faster than deterministic methods. As a rough sketch, Figure 20 shows that the boundaries of the respective domains are not sharply understood in practice. A persistent challenge in randomized algorithms is identifying the constants in the big- O promises of theory for randomized algorithms and thus obtaining a sharp characterization of the problem size at which randomized approaches start to be competitive with, or better than, deterministic ones.

Alternatively, understanding the boundary of such regimes sometimes offers the opportunity for hybridizing discrete and randomized algorithms to obtain an even better complexity/accuracy boundary. An example of the “best of both worlds” is provided by randomized quasi-Monte Carlo approaches [93]. Such approaches are deterministic approaches to high-dimensional integration that reduce integration error from Monte Carlo’s $O(n^{-1/2})$ to “almost” $O(n^{-1})$. Two problems with quasi-Monte Carlo exist: it is deterministic, with no computable a posteriori error bounds, and the “almost” qualification about error reduction hides worst-case powers of $\log(n)$ that are not negligible. The randomized version of quasi-Monte Carlo supports error estimation by replication, has finite sample variance no worse than a constant multiple of Monte Carlo’s, and effectively circumvents the worst case. Moreover, for some sufficiently smooth integrands, the error is much better than either Monte Carlo’s or quasi-Monte Carlo’s. Composing or combining deterministic and randomized methods and understanding the properties of the resulting algorithms would bring about both novel mathematics and improved capabilities for DOE’s applications.

In addition to such general challenges and opportunities concerning the complexity of randomized algorithms, this workshop identified two notable priority research directions related to the complexity of randomized algorithms.

Analysis of Randomized Algorithms for Production Conditions

The promise of reduced complexity of randomized algorithms is an exceptional opportunity to advance the state of the art in mathematics and computer science while addressing critical questions facing the applications in the space of DOE. Randomization has recently been demonstrated to vastly improve both the theoretical and practical complexity of ubiquitous computational kernels, and it is a key enabler for approaching complex tasks that are deterministically intractable.

Randomization has recently been demonstrated to vastly improve both the theoretical and practical complexity of ubiquitous computational kernels, and it is a key enabler for approaching complex tasks that are deterministically intractable.

Achieving such lofty goals faces a set of challenges, such as quantifying and predicting performance of randomized algorithms under realistic production conditions. Moreover, we need to develop analytical tools and frameworks for different classes of randomized methods for paradigms beyond linear algebra. Furthermore, accelerating scientific adoption requires the ability to crisply communicate the algorithmic options and their properties to help users choose the best algorithm for any given task.

To these ends, the community will build on existing tools and results from theoretical computer science, for example by developing rigorous and practical metrics to monitor convergence of randomized algorithms. Novel efforts are needed not only to achieve sharper a priori complexity and run-time estimates but also practical a posteriori error estimates (which are typically significantly less conservative than a priori estimates [104]) for realistic usage environments. A sustained effort is required in the analysis of tradeoffs between cost and accuracy, with a particular focus on characterizing environments and algorithmic templates where less accuracy is sufficient. In relation to hardware models, an interesting opportunity occurs in randomized techniques to avoid worst-case aggregation of errors (randomized rounding or truncation) and, more generally, in the integration of probabilistic error estimates with floating-point error estimates. Furthermore, since the optimal hardware world is likely to be hybrid, an important research direction is the analysis of coupled (hybrid) deterministic/randomized algorithms that combine the best of both worlds.

Randomized Algorithms' Cost and Error Models for Emerging Hardware

The performance of algorithms is affected not only by their mathematical formulation but also by the nature of the hardware system, which can exhibit radically different costs for different primitives. A recent striking example concerns the vastly different computational speeds of integer programming in the classical versus quantum model. This can be seen in Shor's algorithm [136] that achieves polynomial time factorization of an integer in the quantum model, whereas the running times of the best-known deterministic algorithms is exponential (all statements with respect to the number of bits required to represent the number). In this regard, urgent needs have emerged caused by the ending of Moore's law and data federation in combination with novel hardware contexts that include an increased emphasis on streaming application and heterogeneous architectures. Important conceptual challenges include the following: How do algorithms need to change to mirror the hardware evolution? How does error analysis incorporate not only algorithmic errors due to randomization but also hardware originated errors?

The performance of algorithms is affected not only by their mathematical formulation but also by the nature of the hardware system, which can exhibit radically different costs for different primitives.

Addressing such challenges requires progress on multiple research fronts at the intersection of mathematics and computer science. An important first step is a proper abstraction of the problem, which needs interaction between mathematicians, computer scientists, statisticians, and hardware architects to develop concise cost models for the underlying hardware to aid algorithm design. Such descriptions should lead to novel error models and estimators for randomized algorithms for heterogeneous architectures. Moreover, such a focus would broaden the scope of algorithmic innovation and error analysis itself by creating opportunities for new optimized randomized algorithms for evolving hardware cost models that now include, for example, bandwidth and latency limitations.

Such a holistic approach to analysis and algorithmic design will not only increase confidence in randomized algorithms but will produce better randomized algorithm infrastructure for underlying software that benefits many applications; see [Section 3.7](#).

3.6 Verification and Validation

Subsection lead: J. Jakeman

Verification and validation are processes for checking the accuracy and reliability of algorithms or models. Broadly speaking, verification and validation involve the use of systematic tools to study how well computational results agree with known solutions or reference data. DOE has a strong history of supporting verification and validation research for *deterministic* physics-based computations; and verification and validation standards are well established for many science drivers [48, 5, 13]. In the context of *randomized algorithms*, however, the processes for verification and validation have not yet reached the same level of development. Accordingly, future efforts in verification and validation are needed to ensure that new technologies based on randomized algorithms can be used safely and with high confidence.

In order to develop tools and systems for verification and validation of randomized algorithms, challenges and opportunities need to be addressed in several directions.

Going beyond worst-case error analysis Traditionally, error analysis of deterministic algorithms is studied from a *worst-case perspective*. In other words, the goal of this type of analysis is to measure the largest error that may arise among all possible inputs. A common limitation of this approach is that the error bounds tend to be overly conservative for typical inputs, and consequently they may not provide a realistic guidance about an algorithm’s performance in practice. As a way of overcoming this issue, randomized algorithms naturally lend themselves to other types of error analysis. In particular, randomized algorithms are suited to probabilistic analyses that are flexible enough to handle average-case error, as well as a posteriori error estimates to be discussed below. For instance, statements of the form “the error is no worse than ϵ with probability exceeding $(1-\delta)$ ” are common in theory for randomized algorithms, and such *probabilistic bounds* could help in going beyond worst-case analyses for verification and validation.

Bridging computational and statistical perspectives A long-term challenge for verification and validation is bridging the perspectives of computational and statistical research. From the viewpoint of statistics, the output of a randomized algorithm can be considered an “estimate,” while

an exact solution can be considered an “unknown parameter.” When randomized algorithms are viewed from this standpoint, the potential exists to apply a variety of classical statistical methods in the service of error estimation for verification and validation (Figure 21). Some of the most well-established methods in this class are bootstrap, jackknife, and cross-validation. Although these methods have been applied in statistics for decades, their potential uses for verification and validation of randomized algorithms has yet to be fully realized; a recent overview of these connections may be found in [104, Secs. 4.5–4.6 and 12.1–12.2]. From the viewpoint of computer science or applied mathematics, these tools often are referred to as methods for *a posteriori error estimation*, because they are designed to quantify error *after* a randomized solution has been computed. Furthermore, this approach to error estimation offers an interesting contrast to the worst-case error analysis because the *a posteriori* approach tends to be less pessimistic and more adaptive to a given input. Also important is that these connections with statistical methods are generally not available for deterministic algorithms and they represent a unique opportunity that is directly enabled by the use of randomized algorithms.

From the viewpoint of statistics, the output of a randomized algorithm can be regarded as an “estimate,” while an exact solution can be regarded as an “unknown parameter.” When randomized algorithms are viewed from this standpoint, there is a potential to apply a variety of classical statistical methods in the service of error estimation for verification and validation.

Integrating randomized algorithms into coupled workflows Growing evidence suggests that randomized algorithms can dramatically reduce the cost of solving problems that require only moderate accuracy. Research is needed to develop error estimates for a wide range of accuracy requirements, from modest accuracy to machine precision. Often, randomized algorithms are part of a larger workflow; for example, a randomized linear algebra solver is used within a finite element model. Little attention has been given, however, to quantifying the effects of randomization for predictions in coupled workflows. Algorithms that can delineate between the role and effects of noise, errors, data-set distribution, and randomness on overall performance or accuracy would be of great value. Such information could be used to identify the largest sources of error and efficiently focus resources to reduce error in downstream prediction goals.

Successful efforts in this area will be transformative. Verification and validation of randomized algorithms can impact numerous tasks, from distinguishing failure of algorithms versus failure of code (e.g., an insufficient number of iterations versus a bug), to assessing the accuracy of quantum simulations, to providing error estimates needed to estimate risk in decision-making in natural sciences, engineering, and public policy.

3.7 Software Abstractions

Subsection lead: R. Kannan

Scientific data is growing exponentially. Scientists are using faster hardware without any changes on their existing computational algorithm to process the growing data. Because of increasing challenges of miniaturization of chip design, Moore’s law is becoming obsolete, and we can no longer expect computing power to scale as it has in the past. Beyond exascale computing, fundamentally novel techniques are required, such as randomized algorithms to accelerate scientific discoveries on ever-growing data. In this section we consider software abstractions for randomized algorithms of three important components in von Neumann architectures: computation, communication, and input/output. These randomized abstractions for software should enable increased productivity for

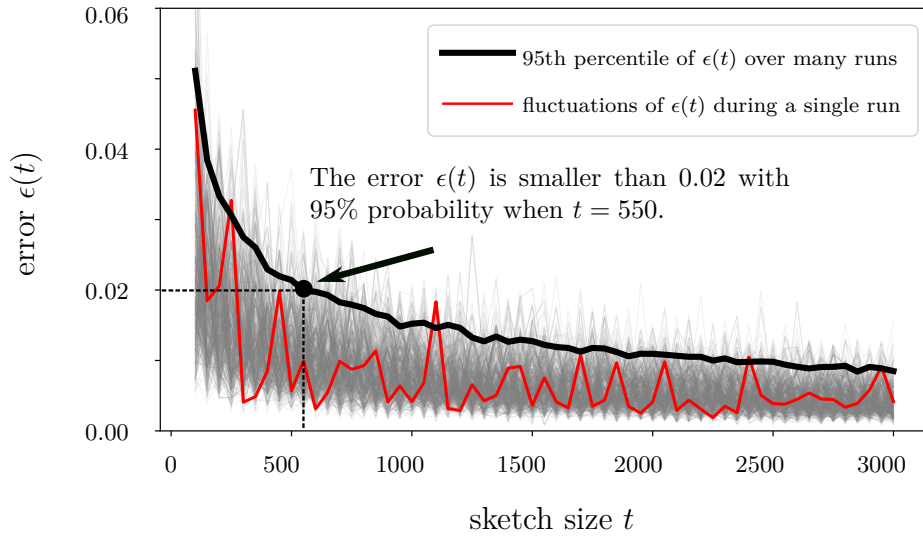


Figure 21: Understanding how the errors of a randomized sketching algorithm fluctuate, with a view toward verification and validation. The light gray curves correspond to different runs of the randomized algorithm as the sketch size t is increased. (Larger t corresponds to more computation.) One such curve for a single run is highlighted in red. The black curve represents the 95th percentile of the error $\epsilon(t)$ over many runs. (That is, for any fixed sketch size t , the error $\epsilon(t)$ will fall below the black curve with 95% probability.) Although the black curve is unknown in practice, it can be approximated by using bootstrap methods for error estimation. Once an approximation to the black curve is available, it can guide the user in selecting an appropriate sketch size for a desired error tolerance. (adapted from the article [99] with permission from the authors.)

developers and make it easy to port existing applications. While any specification or standardization would be a community effort, we lay out key principles and challenges for designing randomized software abstraction.

Composable and Interoperable Randomized Abstractions for Computation

Most randomized algorithms are based on sketching and sampling. Applying these techniques to existing numerical and scientific libraries (e.g., BLAS, LAPACK, FFT, GraphBLAS) and parameterizing them appropriately are challenging. In the near future, most scientific libraries should have some randomization support, such as the sampled dense-dense matrix multiplication operation in the BLAS. Generally speaking, randomized solvers and algorithms are heavily parameterized. Determining the externalization of these parameters and the default values for broader application classes will be an area of further investigation. Many solvers and scientific libraries leverage structures. Randomization techniques must be designed to retain local properties such as symmetric, hierarchical, block, and neighborhood relations, while also preserving global properties of the data (e.g., norm, density). We envision different software layers, from fundamental core operations, solvers, and scientific libraries, to applications that will support various modes of randomization. Real-world applications should deal with the combinatorial search problem of composing the varied randomized layers of the software stack for ideal performance. That is, once the different layers of software (Figure 22) start supporting various randomized abstractions, each of the component parts will be optimized, but a workflow with multiple randomized parts will suffer from differences in abstractions.

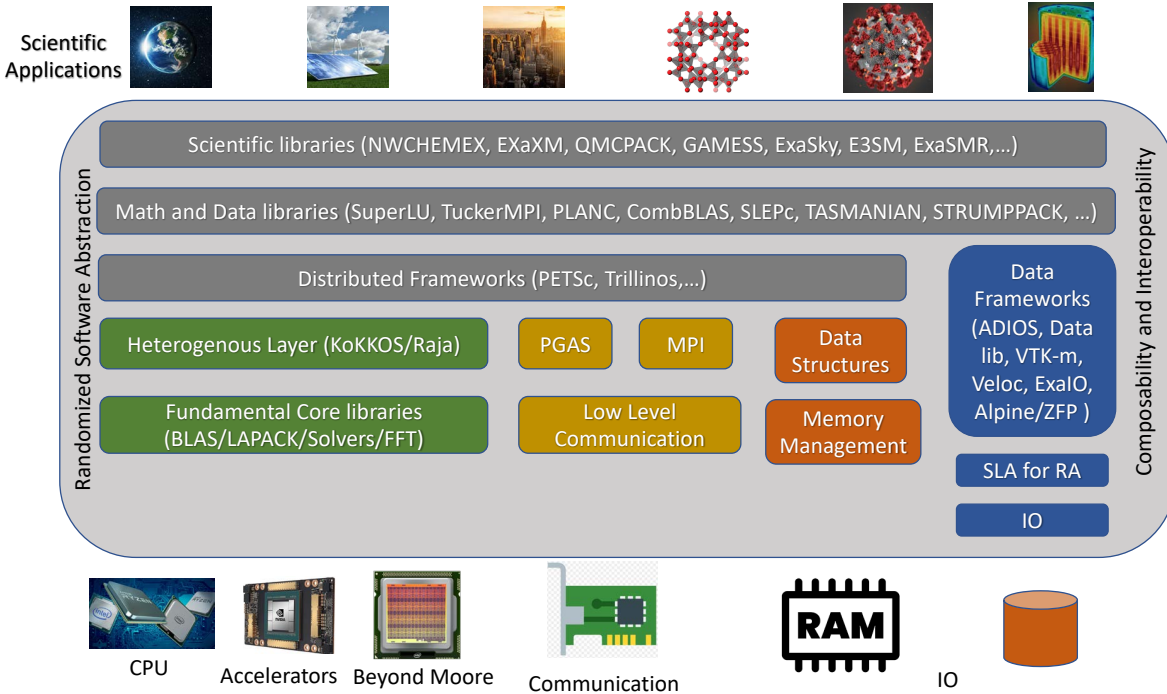


Figure 22: Interoperable randomized abstractions for the DOE software ecosystem.

Interoperable randomized software abstractions across multiple libraries from different entities (academia, labs, and industry) will require a coordinated effort.

Use Cases for Randomized Communication

Communication is important when data is too large to fit in local memory. Many distributed software environments have evolved over different communication libraries, such as MapReduce, MPI, and SHMEM. Sparsification of data has long been used to reduce communication. Recently, researchers in deep learning algorithms have considered sparsification of gradients to minimize *All_Reduce* time. These randomization strategies for communication will impact the realization of a communication operation. The higher-level communication layers, such as MPI and PGAS, and the lower levels, such as Mellanox SHARP, will support different randomization of communication. In theory, similar to computation, algorithms designed for randomized communication will scale to a larger number of processors and data. The existing partitioners that consider balancing load versus minimizing the communication volume must also include randomization as an additional constraint. A potential use case for randomization in communication is addressing missing information from faulty nodes. That is, instead of communication randomization schemes that are local to every node, randomized communication for collective calls can give some rigorous approximations.

Broker Abstractions for Randomized Input/Output

There is no free lunch for input/output in randomized algorithms. In von Neumann-based computer architectures, the memory access is block by block; and in the case of slower storage, it is also sequential. Hence, low compute-intensive randomized algorithms that involve memory and storage access, such as graph applications, cannot show significant advantages for overall time to solution.

A potential approach is to have a broker-based input/output abstraction for expressing the randomization requirements as service-level agreements to address memory access issues. Randomization algorithms also require investigation of fundamental data structures to offer near-real-time random access to the data.

Apart from these three important directions, another important topic is education outreach of randomized thinking and programming. Most existing programming abstractions are based on sequential and deterministic approaches. The computing world faced a significant challenge when educating programmers on distributed and parallel techniques, and history will repeat itself with regard to randomized programming. Some of the hurdles that must be faced include educating researchers on randomized equivalents of existing traditional deterministic solutions using well-defined techniques and novel algorithm design when ground truths are unavailable. Aside from the critical aspects of computation, communication, and input/output, other areas that require new abstractions include reproducibility, debugging, fault tolerance, journals and logs for reversible computation, instrumentation, and performance evaluation, including metrics and measurements for randomized algorithms.

4 Themes and Recommendations

We surveyed the driving application needs in [Section 2](#) and proposed research directions in [Section 3](#). We close with identification of the overarching research themes in randomized algorithms and recommendations for moving forward.

4.1 Themes

Increased computational capacity is required on multiple fronts, including ever-higher resolution in simulations for designing more efficient batteries, processing of massive data from ITER’s nuclear fusion experiment, real-time control of scanning transmission electron microscopy, and integrating ever more data for numerical weather prediction and long-term climate modeling. Traditional algorithms, software, and hardware can no longer keep pace, and randomized algorithms offer the potential for exponential increases in computational efficiency, leading to Theme 1.

Theme 1: Randomized algorithms essential to future computational capacity

The rate of growth in the computational capacity of integrated circuits is expected to slow while data collection is expected to grow exponentially, making randomized algorithms—which depend on sketching, sampling, and streaming computations—essential to the future of computational science and AI for Science.

As we begin to take a fresh look at long-standing problems from a new perspective, novel approaches emerge, as has taken place in every revolution in science. Consider the algorithmic advances that have occurred in *serial* algorithms simply from revisiting them to enable more parallelism. Just considering how to incorporate randomized algorithms has already inspired a fresh look at verification and validation and the hope of incorporating ideas such as bootstrapping from statistics. This inspires Theme 2.

Theme 2: Novel approaches by reframing long-standing challenges

The potential for randomized algorithms goes beyond keeping up with the onslaught of data: it involves opening the door to novel approaches to long-standing challenges. These include scenarios where some uncertainty is unavoidable, such as in real-time control and experimental steering; design under uncertainty; and mitigating stochastic failures in novel materials, software stacks, or grid infrastructure.

In signal processing, the rate at which a continuous signal is sampled is usually selected according to the Nyquist—Shannon sampling theorem, depending on the highest frequency present in the signal to be sampled. In the 2000s, a great deal of excitement was generated by the concept of *compressed sensing*, which allows sampling at a potentially much lower rate by exploiting randomized sampling and the common signal properties of *sparsity* or *compressibility*. For magnetic resonance imaging in health care, this has had significant real-world impacts such as scan times reduced by 50% and increased accuracies for the same scan times.² In computing, we know that we can realize huge gains in efficiency if we allow for the occasional numerical error. In emerging computing regimes such as quantum and neuromorphic computing, imprecision is inherent. This brings us to Theme 3.

²See, e.g., [How Compressed SENSE makes MRI Faster from Philips](#).

Theme 3: Randomness intrinsic to next-generation hardware

Computing efficiencies can be realized by purposely allowing random imprecision in computations. Imprecision is inherent in emerging architectures such as quantum and neuromorphic computers. Randomized algorithms are a natural fit for these environments, and future computing systems will benefit from the co-design of randomized algorithms alongside hardware that favors certain instantiations of randomness.

Indiscriminate use of randomness in algorithms is not what is proposed. For instance, sampling has long been popular in handling large-scale graphs, but the errors cannot be bounded when naive approaches are employed. Instead, methods have been devised that use more sophisticated sampling strategies and provide probabilistic error guarantees (e.g., [135]). Rather, we propose the integration of domain-informed sampling techniques, sketching with theoretical guarantees, and online computations achieving almost the same accuracy as static computations. In turn, this approach will require substantial efforts in overcoming both seen and unseen technical hurdles in order to achieve the deployment of randomized algorithms to key DOE science and national security applications, as in Theme 4.

Theme 4: Technical hurdles requiring theoretical and practical advances

Crafting sophisticated approaches that break the “curse of dimensionality” via sublinear sampling, sketching, and online algorithms requires sophisticated analysis, which has been tackled thus far only in a small subset of scientific computing problems. Foundational research in theory and algorithms needs to be multiplied many times over in order to cover the breadth of DOE applications.

Even though inputs to our calculations have a degree of uncertainty, skepticism about randomized algorithms remains. Various discretizations are accepted as the cost of doing business, and faulty computations due to errant cosmic particles are all but certain in exascale computers. Indeed, poorly crafted randomized algorithms can be terribly inaccurate, but the same is true of any numerical method. The answer is not only to develop better algorithms but also to work on educating our colleagues and users about the advantages and even necessity of randomized algorithms per Theme 5.

Theme 5: Reconciliation of randomness with user expectations

Users are conditioned to certain expectations, such as viewing machine precision as sacrosanct, even when fundamental uncertainties make such precision ludicrous. New metrics for success can expand opportunities for scientific breakthroughs by accounting for tradeoffs among speed, energy consumption, accuracy, reliability, and communication.

DOE has funded decades of world-class research in computational science at the national labs and at universities. Nevertheless, barriers to randomized algorithms persist because naive strategies are rarely competitive with well-understood and optimized deterministic approaches. We need *new expertise* to ably surmount the technical hurdles outlined above, which means outreach to a broader constituency of researchers and is the motivation for Theme 6. One could argue that this is analogous to the integration of mathematics, computer science, and domain expertise in the founding of computational science and engineering.

Theme 6: Need for expanded expertise in statistics and other areas

Establishing randomized algorithms in scientific computing necessitates integrating statistics, theoretical computer science, data science, signal processing, and emerging hardware expertise alongside the traditional domains of applied mathematics, computer science, and engineering and science domain expertise.

4.2 Recommendations

A concerted research program in randomized algorithms will require a mixture of efforts for success. Pursuing research programs in one area at the expense of other areas will slow progress along all fronts. Here we present six recommended priorities for research efforts.

The importance of basic research, promoted in Recommendation 1, cannot be overstated. Such research may be in smaller stand-alone projects or part of joint efforts. Regardless of how it takes place, the researchers engaged in foundational research will need to commit to engaging with algorithmic researchers to bring the theory into practice.

Recommendation 1: Theoretical foundations

Foundational research in the theory of randomized algorithms to (among other issues) understand existing methods, tighten theoretical bounds, and tackle problems of propagating theory into coupled environments. The output of this research will be theorems and proofs to uncover new techniques and guarantees and to address new problem settings.

Development of randomized algorithms is the cornerstone of the proposed effort, per Recommendation 2. The role of algorithm researchers is to unravel the theory into working prototypes of methods, tested on idealized problems that reflect real-world applications. Algorithmic researchers will need to engage with applications and possibly emerging hardware.

Recommendation 2: Algorithmic foundations

Foundational development of sophisticated algorithms that leverage the theoretical underpinnings in practice, identifying and mending any gaps in theory, and establishing performance for idealized and simulated problems. The output here will be advances in algorithm analysis and understanding, prototype software, and reproducible experiments.

While one may imagine application-agnostic randomized algorithms, the reality is that most applications will need tailoring of approaches to the domain. This might be in the form of sampling strategies, such as appropriate stratified sampling, or it might go so far as requiring new specific theory. The goal of Recommendation 3 is to focus on customizing solutions to specific applications and their individual needs.

Recommendation 3: Application integration

Deployment in scientific applications in concert with domain experts. This will often require extending existing theory and algorithms to the special cases of relevance for each application, as well as application-informed sampling designs. The output here will be software alongside benchmarks and best practices for specific applications, focused on enabling novel scientific and engineering advances.

Randomly distributed data, as in sparse matrices, has always bedeviled computational efficiency. One might conclude that introducing more randomness could be detrimental to computational efficiency. However, the next-generation hardware will have inherent randomness. The goal of Recommendation 4 is to develop and co-design randomized algorithms that are scalable.

Recommendation 4: Performance on next-generation hardware

Adaptation of randomized algorithms to take advantage of best-in-class computing hardware, from current architectures to quantum, neuromorphic, and other emerging platforms. The output here will be high-performance open-source software for next-generation computing hardware, including enabling efficient utilization of nondeterministic hardware and maximizing performance of deterministic hardware.

The shift to randomized algorithms represents a fundamentally new direction for DOE; thus, it requires new specializations that are not currently represented in its research program. Recommendation 5 has to do with broadening the teams of researchers that are engaged with DOE via this new effort. Pursuing this effort will accelerate the success of deploying randomized algorithms over the next decade and bring a broader perspective to DOE's problems overall. Without a specific push in the direction of diversification, it will be too easy to fall back on the known and trusted personnel in the current program.

Recommendation 5: Outreach

Outreach to a broader community to facilitate engagement outside the traditional computational science community, including experts in statistics, applied probability, signal processing, and emerging hardware. The output of this effort will be community-building workshops and research efforts with topically diverse teams that break new frontiers.

In concert with efforts in computer science and elsewhere, we also need to consider the standardization of randomized algorithms, which is the focus of Recommendation 6. It is difficult to think of standardization in a topic that is still so new in computational science, yet the next decade should bring a wealth of advances. For these advances to have the greatest impact, they will need to be incorporated into software frameworks, which will require standards in how to do so.

Recommendation 6: Workflow standardization

Standardization of workflow, including debugging and test frameworks for methods with only probabilistic guarantees, software frameworks that both integrate randomized algorithms and provide new primitives for sampling and sketching, and modular frameworks for incorporating the methods into large-scale codes and deploying to new architectures. The output here will be community best practices and reduced barriers to contributing to scientific advances.

References

- [1] *ITER Home Page*. <https://www.iter.org>. Accessed: 2021-02-02.
- [2] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, M. Gates, T. Grützmacher, N. J. Higham, S. Li, N. Lindquist, Y. Liu, J. Loe, P. Luszczek, P. Nayak, S. Pranesh, S. Rajamanickam, T. Ribizel, B. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, I. Yamazaki, and U. M. Yang. *A Survey of Numerical Methods Utilizing Mixed Precision Arithmetic*, 2020, [arXiv:2007.06674](https://arxiv.org/abs/2007.06674).
- [3] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems*, 38(4):26:1–26:28, 2013, [doi:10.1145/2500128](https://doi.org/10.1145/2500128).
- [4] N. K. Ahmed, J. Neville, and R. Kompella. Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data*, 8(2):1–56, 2014, [doi:10.1145/2601438](https://doi.org/10.1145/2601438).
- [5] AIAA. *Guide for the Verification and Validation of Computational Fluid dynamics simulations*, 1998, [doi:10.2514/4.472855](https://doi.org/10.2514/4.472855).
- [6] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson–Lindenstrauss transform. *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, p. 557–563, 2006, [doi:10.1145/1132516.1132597](https://doi.org/10.1145/1132516.1132597).
- [7] N. Ailon and B. Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009, [doi:10.1137/060673096](https://doi.org/10.1137/060673096).
- [8] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege. A systematic review of the application and empirical investigation of search-based test case generation. *IEEE Transactions on Software Engineering*, 36(6):742–762, 2010, [doi:10.1109/tse.2009.52](https://doi.org/10.1109/tse.2009.52).
- [9] A. Alla and J. N. Kutz. Randomized model order reduction. *Advances in Computational Mathematics*, 45(3):1251–1271, 2019, [doi:10.1007/s10444-018-09655-9](https://doi.org/10.1007/s10444-018-09655-9).
- [10] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995, [doi:10.1145/210332.210337](https://doi.org/10.1145/210332.210337).
- [11] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999, [doi:10.1006/jcss.1997.1545](https://doi.org/10.1006/jcss.1997.1545).
- [12] N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. Tracking join and self-join sizes in limited storage. *Journal of Computer and System Sciences*, 64(3):719–747, 2002, [doi:10.1006/jcss.2001.1813](https://doi.org/10.1006/jcss.2001.1813).
- [13] AMSE. *Guide for Verification and Validation in Computational Solid Mechanics*, 2006, <https://www.asme.org/codes-standards/find-codes-standards/v-v-10-guide-verification-validation-computational-solid-mechanics>.
- [14] B. M. Austin, D. Y. Zubarev, and W. A. Lester. Quantum Monte Carlo and related approaches. *Chemical Reviews*, 112(1):263–288, 2012, [doi:10.1021/cr2001564](https://doi.org/10.1021/cr2001564).

-
- [15] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, 58(2):1–34, 2011, doi:10.1145/1944345.1944349.
- [16] H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK’s least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010, doi:10.1137/090767911.
- [17] C. Bach, D. Ceglia, L. Song, and F. Duddeck. Randomized low-rank approximation methods for projection-based model order reduction of large nonlinear dynamical problems. *International Journal for Numerical Methods in Engineering*, 118(4):209–241, 2019, doi:10.1002/nme.6009.
- [18] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. M. Wild, and K. Willcox. *Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence*, 2019, doi:10.2172/1478744.
- [19] F. Bao, R. Archibald, and P. Maksymovych. Lévy backward SDE filter for jump diffusion processes and its applications in material sciences. *Communications in Computational Physics*, 27(2):589–618, 2020, doi:10.4208/cicp.oa-2018-0238.
- [20] R. Bartlett, I. Demeshko, T. Gamblin, G. Hammond, M. Heroux, J. Johnson, A. Klinvex, X. Li, L. C. McInnes, J. D. Moulton, D. Osei-Kuffuor, J. Sarich, B. Smith, J. Willenbring, and U. M. Yang. xSDK foundations: Toward an extreme-scale scientific software development kit. *Supercomputing Frontiers and Innovations*, 4(1):69–82, 2017, doi:10.14529/jsfi170104.
- [21] S. Barutcu, P. Ruiz, F. Schiffrers, S. Aslan, D. Gursoy, O. Cossairt, and A. K. Katsaggelos. Simultaneous 3D X-ray ptycho-tomography with gradient descent. In *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, doi:10.1109/icip40778.2020.9190775.
- [22] C. Battaglino, G. Ballard, and T. G. Kolda. A practical randomized CP tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018, doi:10.1137/17M1112303.
- [23] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020, doi:10.1016/j.physrep.2020.05.004.
- [24] P. Bauer, P. D. Dueben, T. Hoefler, T. Quintino, T. C. Schulthess, and N. P. Wedi. The digital revolution of Earth-system science. *Nature Computational Science*, 1(2):104–113, 2021, doi:10.1038/s43588-021-00023-0.
- [25] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018, <http://jmlr.org/papers/v18/17-468.html>.
- [26] A. A. Benczúr and D. R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015, doi:10.1137/070705970.
- [27] C. H. Bennett and D. P. DiVincenzo. Quantum information and computation. *Nature*, 404(6775):247–255, 2000, doi:10.1038/35005001.

-
- [28] E. Bergou, Y. Diouane, V. Kungurtsev, and C. W. Royer. *A stochastic Levenberg–Marquardt method using random models with complexity results and application to data assimilation*, 2018, [arXiv:1807.02176](https://arxiv.org/abs/1807.02176).
- [29] M. Berry, T. E. Potok, P. Balaprakash, H. Hoffmann, R. Vatsavai, and Prabhat. *Machine Learning and Understanding for Intelligent Extreme Scale Scientific Computing and Discovery. DOE Workshop Report, January 7–9, 2015, Rockville, MD*, 2015, [doi:10.2172/1471083](https://doi.org/10.2172/1471083).
- [30] E. W. Bethel and M. Greenwald. *Report of the DOE Workshop on Management, Analysis, and Visualization of Experimental and Observational Data – The Convergence of Data and Computing*, 2016, [doi:10.2172/1525145](https://doi.org/10.2172/1525145).
- [31] A. Biswas, S. Dutta, E. Lawrence, J. Patchett, J. C. Calhoun, and J. Ahrens. Probabilistic data-driven sampling via multi-criteria importance analysis. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2020, [doi:10.1109/TVCG.2020.3006426](https://doi.org/10.1109/TVCG.2020.3006426).
- [32] E. K. Bjarkason, O. J. Maclaren, J. P. O’Sullivan, and M. J. O’Sullivan. Randomized truncated SVD Levenberg–Marquardt approach to geothermal natural state and history matching. *Water Resources Research*, 54(3):2376–2404, 2018, [doi:10.1002/2017WR021870](https://doi.org/10.1002/2017WR021870).
- [33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2010, [doi:10.1561/22000000016](https://doi.org/10.1561/22000000016).
- [34] P. Brans. The many wonders of ITER Diagnostics. *ITER Newslines*, 2019, <https://www.iter.org/newsline/-/3360>. Accessed: 2021-02-02.
- [35] P. Brans. How to manage 2 petabytes of data everyday. *ITER Newslines*, 2020, <https://www.iter.org/newsline/-/3534>. Accessed: 2021-02-02.
- [36] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996, [doi:10.1007/bf00058655](https://doi.org/10.1007/bf00058655).
- [37] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001, [doi:10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [38] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017, [doi:10.1109/msp.2017.2693418](https://doi.org/10.1109/msp.2017.2693418).
- [39] D. Ceperley and B. Alder. Quantum Monte Carlo. *Science*, 231(4738):555–560, 1986, [doi:10.1126/science.231.4738.555](https://doi.org/10.1126/science.231.4738.555).
- [40] C. Chekuri, A. V. Goldberg, D. R. Karger, M. S. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *SODA*, volume 97, pp. 324–333, 1997.
- [41] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology – EUROCRYPT 2019*, pp. 375–403. Springer International Publishing, 2019, [doi:10.1007/978-3-030-17653-2_13](https://doi.org/10.1007/978-3-030-17653-2_13).
- [42] J. Choi, R. Wang, R. M. Churchill, R. Kube, M. Choi, J. Park, J. Logan, K. Mehta, G. Eisenhauer, N. Podorszki, M. Wolf, C. S. Chang, and S. Klasky. Data federation challenges in

-
- remote near-real-time fusion experiment data processing. In *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI*, pp. 285–299. Springer International Publishing, 2020, doi:10.1007/978-3-030-63393-6_19.
- [43] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing - STOC '09*, pp. 205–214. ACM Press, 2009, doi:10.1145/1536414.1536445.
- [44] K. L. Clarkson and D. P. Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63(6):1–45, 2017, doi:10.1145/3019134.
- [45] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019, <http://proceedings.mlr.press/v97/cohen19c.html>.
- [46] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to compressed sensing. In *Compressed Sensing: Theory and Applications*, chapter 1. Cambridge University Press, 2013.
- [47] M. Davies, N. Srinivasa, T.-H. Lin, G. China, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018, doi:10.1109/mm.2018.112130359.
- [48] Department of Defense. *VV&A Recommended Practices Guide*, 2011, <https://vva.msc.mil/>.
- [49] DOE ASCR. *Exascale Requirements Review Crosscut Report*, 2018, <https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2018/DOE-ExascaleReport-CrossCut.pdf>.
- [50] DOE-FES and DOE-ASCR. *Report of the Workshop on Integrated Simulations for Magnetic Fusion Energy Sciences*, 2015, https://science.osti.gov/-/media/fes/pdf/workshop-reports/2016/ISFusionWorkshopReport_11-12-2015.pdf.
- [51] J. Dongarra, P. Luszczek, and A. Petitet. The LINPACK benchmark: Past, present and future. *Concurrency and Computation: Practice and Experience*, 15(9):803–820, 2003, doi:10.1002/cpe.728.
- [52] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011, <https://jmlr.org/papers/v12/duchi11a.html>.
- [53] J. A. Duersch and M. Gu. Randomized QR with column pivoting. *SIAM Journal on Scientific Computing*, 39(4):C263–C291, 2017, doi:10.1137/15m1044680.
- [54] J. A. Duersch and M. Gu. Randomized projection for rank-revealing matrix factorizations and low-rank approximations. *SIAM Review*, 62(3):661–682, 2020, doi:10.1137/20m1335571.
- [55] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pp. 265–284. Springer Berlin Heidelberg, 2006, doi:10.1007/11681878_14.

-
- [56] O. Dyck, M. Ziatdinov, S. Jesse, F. Bao, A. Y. Nobakht, A. Maksov, B. Sumpter, R. Archibald, K. Law, and S. Kalinin. Probing potential energy landscapes via electron-beam-induced single atom dynamics. *Acta Materialia*, 203:116508, 2021, doi:10.1016/j.actamat.2020.116508.
- [57] R. Eckhardt. Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, 15:131–137, 1987, <https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9068>.
- [58] K. S. Eikrem, G. Nævdal, and M. Jakobsen. Iterative solution of the Lippmann–Schwinger equation in strongly scattering acoustic media by randomized construction of preconditioners. *Geophysical Journal International*, 224(3):2121–2130, 2020, doi:10.1093/gji/ggaa503.
- [59] B. Ekim, B. Berger, and Y. Orenstein. A randomized parallel algorithm for efficiently finding near-optimal universal hitting sets. In *Lecture Notes in Computer Science*, pp. 37–53. Springer International Publishing, 2020, doi:10.1007/978-3-030-45257-5_3.
- [60] N. B. Erichson, O. Azencot, A. Queiruga, L. Hodgkinson, and M. W. Mahoney. *Lipschitz Recurrent Neural Networks*, 2021, arXiv:2006.12070.
- [61] M. D. Fabian, B. Shpiro, E. Rabani, D. Neuhauser, and R. Baer. Stochastic density functional theory. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 9(6):e1412, 2019, doi:10.1002/wcms.1412.
- [62] H. Finkel and I. Laguna. *Report of the Workshop on Program Synthesis for Scientific Computing*, 2021, arXiv:2102.01687.
- [63] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985, doi:10.1016/0022-0000(85)90041-8.
- [64] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. *Discrete Mathematics and Theoretical Computer Science Proceedings*, pp. 127–146, 2007, <https://dmtcs.episciences.org/3545>.
- [65] A. H. Gebremedhin, F. Manne, and A. Pothen. What color is your Jacobian? Graph coloring for computing derivatives. *SIAM Review*, 47(4):629–705, 2005, doi:10.1137/S0036144504444711.
- [66] B. Ghazi, R. Panigrahy, and J. R. Wang. Recursive sketches for modular deep learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2211–2220, 2019, <http://proceedings.mlr.press/v97/ghazi19a.html>.
- [67] P. Ghysels, X. S. Li, F.-H. Rouet, S. Williams, and A. Napov. An efficient multicore implementation of a novel HSS-structured multifrontal solver using randomized sampling. *SIAM Journal on Scientific Computing*, 38(5):S358–S384, 2016, doi:10.1137/15M1010117.
- [68] A. Griewank and C. Mitev. Detecting Jacobian sparsity patterns by Bayesian probing. *Mathematical Programming*, 93(1):1–25, 2002, doi:10.1007/s101070100281.
- [69] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.

-
- [70] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017, doi:10.1088/1361-6420/aa9a90.
- [71] A. Haidar, H. Bayraktar, S. Tomov, J. Dongarra, and N. J. Higham. Mixed-precision iterative refinement using tensor cores on GPUs to accelerate solution of linear systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2243):20200110, 2020, doi:10.1098/rspa.2020.0110.
- [72] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011, doi:10.1137/090771806.
- [73] K. E. Hamilton, N. Imam, and T. S. Humble. Community detection with spiking neural networks for neuromorphic hardware. In *Proceedings of the Neuromorphic Computing Symposium*. ACM, 2017, doi:10.1145/3183584.3183621.
- [74] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.
- [75] M. A. Heroux, J. Carter, R. Thakur, J. S. Vetter, L. C. McInnes, J. Ahrens, T. Munson, and J. R. Neely. *ECP Software Technology Capability Assessment Report*, 2020, doi:10.2172/1597433.
- [76] M. Hidayetoğlu, T. Biçer, S. G. de Gonzalo, B. Ren, D. Gürsoy, R. Kettimuthu, I. T. Foster, and W. W. Hwu. MemXCT: memory-centric X-ray CT reconstruction with massive parallelization. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2019, doi:10.1145/3295500.3356220.
- [77] W. Hu, L. Lin, and C. Yang. Interpolative separable density fitting decomposition for accelerating hybrid density functional calculations with applications to defects in silicon. *Journal of Chemical Theory and Computation*, 13(11):5420–5431, 2017, doi:10.1021/acs.jctc.7b00807.
- [78] W. Hu, J. Liu, Y. Li, Z. Ding, C. Yang, and J. Yang. Accelerating excitation energy computation in molecules and solids within linear-response time-dependent density functional theory via interpolative separable density fitting decomposition. *Journal of Chemical Theory and Computation*, 16(2):964–973, 2020, doi:10.1021/acs.jctc.9b01019.
- [79] Z. Huang. *ExaSGD: Optimizing Stochastic Grid Dynamics at ExaScale*. interim report, 2020, https://www.exascaleproject.org/wp-content/uploads/2020/01/ECP_AD_Power-Grids.pdf.
- [80] M. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990, doi:10.1080/03610919008812866.
- [81] U. Janjarassuk and J. Linderoth. Reformulation and sampling to solve a stochastic network interdiction problem. *Networks*, 52(3):120–132, 2008, doi:10.1002/net.20237.
- [82] M. Jha, C. Seshadhri, and A. Pinar. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 495–505, 2015.

-
- [83] S. Kalinin, A. Borisevich, and S. Jesse. Fire up the atom forge. *Nature*, 539(7630):485–487, 2016, doi:10.1038/539485a.
- [84] D. R. Karger and C. Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, 1996, doi:10.1145/234533.234534.
- [85] D. R. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2):321–328, 1995, doi:10.1145/201019.201022.
- [86] R. M. Karp. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34(1-3):165–201, 1991, doi:10.1016/0166-218X(91)90086-C.
- [87] D. E. Keyes, H. Ltaief, and G. Turkiyyah. Hierarchical algorithms on hierarchical architectures. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2166):20190055, 2020, doi:10.1098/rsta.2019.0055.
- [88] J. Kim, W. J. Dally, S. Scott, and D. Abts. Technology-driven, highly-scalable dragonfly topology. In *2008 International Symposium on Computer Architecture*. IEEE, 2008, doi:10.1109/isca.2008.19.
- [89] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*, 2015, arXiv:1412.6980.
- [90] P. R. Kramer. A review of some Monte Carlo simulation methods for turbulent systems. *Monte Carlo Methods and Applications*, 7(3-4):229 – 244, 2001, doi:10.1515/mcma.2001.7.3-4.229.
- [91] V. Kumar and R. Grossman, editors. *Proceedings of the 2001 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2001, doi:10.1137/1.9781611972719.
- [92] A. Leal et al. *autodif: automatic differentiation in C++ couldn't be simpler*. GitHub repository, 2020, github.com/autodiff/autodiff.
- [93] P. L'Ecuyer. Randomized quasi-Monte Carlo: An introduction for practitioners. In *Springer Proceedings in Mathematics & Statistics*, pp. 29–52. Springer International Publishing, 2018, doi:10.1007/978-3-319-91436-7_2.
- [94] J. Lee, L. Lin, and M. Head-Gordon. Systematically improvable tensor hypercontraction: Interpolative separable density-fitting for molecules applied to exact exchange, second- and third-order Moller—Plesset perturbation theory. *Journal of Chemical Theory and Computation*, 16(1):243–263, 2020, doi:10.1021/acs.jctc.9b00820.
- [95] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '06*, pp. 631–636. ACM, ACM Press, 2006, doi:10.1145/1150402.1150479.
- [96] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal Control*. John Wiley & Sons, 2012.
- [97] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007, doi:10.1073/pnas.0709640104.

-
- [98] P. Lofgren, S. Banerjee, A. Goel, and C. Seshadhri. FAST-PPR: Scaling personalized pagerank estimation for large graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1436–1445. ACM, 2014, doi:10.1145/2623330.2623745.
- [99] M. E. Lopes, N. B. Erichson, and M. W. Mahoney. Error estimation for sketched SVD via the bootstrap. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6382–6392, 2020, <http://proceedings.mlr.press/v119/lopes20a.html>.
- [100] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011, doi:10.1561/22000000035.
- [101] A. S. Maiya and T. Y. Berger-Wolf. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, pp. 105–113. ACM, ACM Press, 2011, doi:10.1145/2020408.2020431.
- [102] P. G. Martinsson. A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1251–1274, 2011, doi:10.1137/100786617.
- [103] P.-G. Martinsson. *Fast Direct Solvers for Elliptic PDEs*, volume CB96 of *CBMS-NSF conference series*. SIAM, 2019. ISBN 978-1-611976-03-8.
- [104] P.-G. Martinsson and J. A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020, doi:10.1017/S0962492920000021.
- [105] P.-G. Martinsson, G. Q. Ortíz, N. Heavner, and R. van de Geijn. Householder QR factorization with randomization for column pivoting (HQRRP). *SIAM Journal on Scientific Computing*, 39(2):C96–C115, 2017, doi:10.1137/16m1081270.
- [106] K. J. Mayer and E. A. Barnes. Subseasonal forecasts of opportunity identified by an interpretable neural network. *Earth and Space Science Open Archive*, 2020, doi:10.1002/essoar.10505448.1.
- [107] A. McGregor. Graph sketching. In *Encyclopedia of Algorithms*, pp. 879–882. Springer New York, 2016, doi:10.1007/978-1-4939-2864-4_796.
- [108] C. N. Melton, M. M. Noack, T. Ohta, T. E. Beechem, J. Robinson, X. Zhang, A. Bostwick, C. Jozwiak, R. J. Koch, P. H. Zwart, A. Hexemer, and E. Rotenberg. K-means-driven Gaussian process data collection for angle-resolved photoemission spectroscopy. *Machine Learning: Science and Technology*, 1(4):045015, 2020, doi:10.1088/2632-2153/abab61.
- [109] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014, doi:10.1126/science.1254642.
- [110] N. Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science*, 15:125–130, 1987, <http://library.lanl.gov/cgi-bin/getfile?15-12.pdf>.
- [111] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949, doi:10.1080/01621459.1949.10483310.

-
- [112] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6): 1087–1092, 1953, doi:10.1063/1.1699114.
- [113] J. Misra and D. Gries. Finding repeated elements. *Science of Computer Programming*, 2(2): 143–152, 1982, doi:10.1016/0167-6423(82)90012-0.
- [114] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001, doi:10.1109/71.963420.
- [115] R. Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978, doi:10.1145/359619.359627.
- [116] J. Munro and M. Paterson. Selection and sorting with limited storage. *Theoretical Computer Science*, 12(3):315–323, 1980, doi:10.1016/0304-3975(80)90061-4.
- [117] J. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286, 1951, doi:10.2307/1969529.
- [118] National Research Council. *Probability and Algorithms*, 1992, doi:10.17226/2026.
- [119] National Research Council. *The Mathematical Sciences in 2025*, 2013, doi:10.17226/15269.
- [120] M. M. Noack, G. S. Doerk, R. Li, J. K. Streit, R. A. Vaia, K. G. Yager, and M. Fukuto. Autonomous materials discovery driven by gaussian process regression with inhomogeneous measurement noise and anisotropic kernels. *Scientific Reports*, 10(1):17663, 2020, doi:10.1038/s41598-020-74394-1.
- [121] A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000, doi:10.1080/01621459.2000.10473909.
- [122] J. Patchett and J. Ahrens. Optimizing scientist time through in situ visualization and analysis. *IEEE Computer Graphics and Applications*, 38(1):119–127, 2018, doi:10.1109/MCG.2018.011461533.
- [123] J. M. Perkel. Ten computer codes that transformed science. *Nature*, 589(7842):344–348, 2021, doi:10.1038/d41586-021-00075-2.
- [124] R. Pollice, G. Dos Passos Gomes, M. Aldeghi, R. J. Hickman, M. Krenn, C. Lavigne, M. Lindner-D’Addario, A. Nigam, C. T. Ser, Z. Yao, and A. Aspuru-Guzik. Data-driven strategies for accelerated materials design. *Accounts of Chemical Research*, 54(4):849–860, 2021, doi:10.1021/acs.accounts.0c00785.
- [125] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018, doi:10.22331/q-2018-08-06-79.
- [126] RAND Corporation. *A Million Random Digits with 100,000 Normal Deviates*. RAND Corporation, Santa Monica, CA, 2001, doi:10.7249/MR1418.
- [127] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, pp. 63–71. Springer Berlin Heidelberg, 2004, doi:10.1007/978-3-540-28650-9_4.

-
- [128] F. Ren, L. Ward, T. Williams, K. J. Laws, C. Wolverton, J. Hattrick-Simpers, and A. Mehta. Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Science Advances*, 4(4):eaq1566, 2018, doi:10.1126/sciadv.aaq1566.
- [129] J. M. Restrepo, G. K. Leaf, and A. Griewank. Circumventing storage limitations in variational data assimilation studies. *SIAM Journal on Scientific Computing*, 19(5):1586–1605, 1998, doi:10.1137/s1064827595285500.
- [130] L. Rhodes. *DataSketches.GitHub.io: A Required Toolkit for the Analysis of Big Data*, 2018, https://datasketches.apache.org/docs/pdf/DataSketches_deck.pdf.
- [131] V. Rokhlin and M. Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36):13212–13217, 2008, doi:10.1073/pnas.0804869105.
- [132] T. P. Ryan and J. P. Morgan. Modern experimental design. *J. Stat. Theory Pract.*, 1(3-4): 501–506, 2007.
- [133] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015, doi:10.1016/j.neunet.2014.09.003.
- [134] C. D. Schuman, K. Hamilton, T. Mintz, M. M. Adnan, B. W. Ku, S.-K. Lim, and G. S. Rose. Shortest path and neighborhood subgraph extraction on a spiking memristive neuromorphic implementation. In *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop NICE '19*. ACM Press, 2019, doi:10.1145/3320288.3320290.
- [135] C. Seshadhri, A. Pinar, and T. G. Kolda. Triadic measures on graphs: The power of wedge sampling. In *SDM13: Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 10–18, 2013, doi:10.1137/1.9781611972832.2.
- [136] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. IEEE, 1994.
- [137] R. Stevens, V. Taylor, J. Nichols, A. B. Maccabe, K. Yelick, and D. Brown. *AI for Science*, 2020, doi:10.2172/1604756.
- [138] Y. Sun, B. Wohlberg, and U. S. Kamilov. An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408, 2019, doi:10.1109/tci.2019.2893568.
- [139] Y. Sun, Y. Guo, C. Luo, J. Tropp, and M. Udell. Low-rank Tucker approximation of a tensor from streaming data. *SIAM Journal on Mathematics of Data Science*, 2(4):1123–1150, 2020, doi:10.1137/19m1257718.
- [140] Y. Sun, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov. *Async-RED: A Provably Convergent Asynchronous Block Parallel Stochastic Method using Deep Denoising Priors*, 2020, arXiv:2010.01446. To appear in International Conference on Learning Representations.
- [141] J. Tang and M. Davies. *A Fast Stochastic Plug-and-Play ADMM for Imaging Inverse Problems*, 2020, arXiv:2006.11630.

-
- [142] A. Thakurta, A. Vyrros, U. Vaishampayan, G. Kapoor, J. Freudiger, V. Sridhar, and D. Davidson. US Patent 9594741, 2017.
- [143] D. Tran, M. D. Hoffman, R. A. Saurous, E. Brevdo, K. Murphy, and D. M. Blei. *Deep probabilistic programming*, 2017, [arXiv:1701.03757](https://arxiv.org/abs/1701.03757).
- [144] S. Ubaru, J. Chen, and Y. Saad. Fast Estimation of $\text{tr}(f(A))$ via Stochastic Lanczos Quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017, [doi:10.1137/16M1104974](https://doi.org/10.1137/16M1104974).
- [145] U.S. Department of Energy Basic Energy Sciences. *Basic Energy Sciences Exascale Requirement Review*, 2015, <https://www.osti.gov/servlets/purl/1341721>.
- [146] U.S. Department of Energy Basic Energy Sciences. *Roundtable on Producing and Managing Large Scientific Data with Artificial Intelligence and Machine Learning*, 2020, https://science.osti.gov/-/media/bes/pdf/reports/2020/AI-ML_Report.pdf.
- [147] J. van den Brand, Y. T. Lee, A. Sidford, and Z. Song. Solving tall dense linear programs in nearly linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020, [doi:10.1145/3357713.3384309](https://doi.org/10.1145/3357713.3384309).
- [148] J. S. Vetter, R. Brightwell, M. Gokhale, P. McCormick, R. Ross, J. Shalf, K. Antypas, D. Donofrio, T. Humble, C. Schuman, B. Van Essen, S. Yoo, A. Aiken, D. Bernholdt, S. Byna, K. Cameron, F. Cappello, B. Chapman, A. Chien, M. Hall, R. Hartman-Baker, Z. Lan, M. Lang, J. Leidel, S. Li, R. Lucas, J. Mellor-Crummey, P. Peltz Jr., T. Peterka, M. Strout, and J. Wilke. *Extreme Heterogeneity 2018 - Productive Computational Science in the Era of Extreme Heterogeneity: Report for DOE ASCR Workshop on Extreme Heterogeneity*, 2018, [doi:10.2172/1473756](https://doi.org/10.2172/1473756).
- [149] V. Vlček, E. Rabani, D. Neuhauser, and R. Baer. Stochastic GW calculations for molecules. *Journal of Chemical Theory and Computation*, 13(10):4997–5003, 2017, [doi:10.1021/acs.jctc.7b00770](https://doi.org/10.1021/acs.jctc.7b00770).
- [150] A. F. Voter. Introduction to the kinetic Monte Carlo method. In K. E. Sickafus, E. A. Kotomin, and B. P. Uberuaga, editors, *Radiation Effects in Solids*, volume 235 of *NATO Science Series*, pp. 1–23. Springer Netherlands, Dordrecht, 2007, [doi:10.1007/978-1-4020-5295-8_1](https://doi.org/10.1007/978-1-4020-5295-8_1).
- [151] A. F. Voter and J. D. Doll. Surface self-diffusion constants at low temperature: Monte Carlo transition state theory with importance sampling. *The Journal of Chemical Physics*, 80(11):5814–5817, 1984, [doi:10.1063/1.446606](https://doi.org/10.1063/1.446606).
- [152] L. A. Wolsey. *Integer Programming*, volume 42. Wiley Online Library, 1998.
- [153] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10:1–157, 2014, [doi:10.1561/04000000060](https://doi.org/10.1561/04000000060).
- [154] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008, [doi:10.1016/j.acha.2007.12.002](https://doi.org/10.1016/j.acha.2007.12.002).
- [155] J. Xia. Randomized sparse direct solvers. *SIAM Journal on Matrix Analysis and Applications*, 34(1):197–227, 2013, [doi:10.1137/12087116X](https://doi.org/10.1137/12087116X).

-
- [156] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher. Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200, 2021, doi:[10.1137/19M1305045](https://doi.org/10.1137/19M1305045).

A Workshop Information

Workshop Agenda

All times Eastern

Part 1 Bootcamp, Randomized Algorithms Tutorials: December 2, 2020

11:00 AM–11:30 AM	Introduction and Charge Steven Lee, US Department of Energy, ASCR
11:30 AM–11:55 AM	<i>Probability for Randomized Algorithms</i> Miles Lopes, University of California, Davis
11:55 AM–12:15 PM	Discussion
12:15 PM–12:40 PM	<i>(Practical) Randomized Algorithms for Graph Problems</i> C. Seshadhri, University of California, Santa Cruz
12:40 PM–1:00 PM	Discussion
1:00 PM–1:15 PM	Break
1:15 PM–1:40 PM	<i>Randomized Algorithms in Linear Algebra and Scientific Computing</i> Per Gunnar Martinsson, University of Texas, Austin
1:40 PM–2:00 PM	Discussion
2:00 PM–2:25 PM	<i>A Whirlwind Tour of Stochastic Optimization</i> John Duchi, Stanford University
2:25 PM–2:45 PM	Discussion
2:45 PM–3:00 PM	Wrap-up Part 1, Day 1

Part 1 Bootcamp, Science and Engineering Opportunities: December 3, 2020

11:00 AM–11:15 AM	Agenda and Randomized Breakout Structure
11:15 AM–11:30 AM	<i>Random Thoughts on Randomized Algorithms</i> Julia Ling, Citrine Informatics
11:30 AM–11:45 AM	<i>Random Thoughts on Randomized Algorithms</i> Salman Habib, Computational Science Division, Argonne National Laboratory
11:45 AM–12:00 PM	<i>The Intersection of Discrete Math and Chemistry: Opportunities for Randomized Algorithms</i> Aurora Clark, Department of Chemistry, Washington State University
12:00 PM–12:30 PM	Application Drivers Panel #1 Discussion
12:30 PM–12:45 PM	Break
12:45 PM–1:00 PM	<i>Thoughts on Applications of RASC for Inverse Problems in Imaging</i> Charles Bouman, School of Electrical and Computer Engineering, Purdue University
1:00 PM–1:15 PM	<i>Imaging Biology</i> Peter Zwart, MBIB / BSISB / CAMERA, Lawrence Berkeley National Laboratory
1:15 PM–1:35 PM	Application Drivers Panel #2 Discussion
1:35 PM–1:45 PM	Breakout Group Instructions
1:45 PM–2:45 PM	Breakout Groups on Priority Research Opportunities
2:45 PM–3:00 PM	Wrap-up Part 1

Part 2: January 6, 2021

11:00 AM–11:15 AM	Agenda and Part 2 Aims
11:15 AM–12:15 PM	<i>Federal Agency Panel</i> Fariba Fahroo, AFOSR Steven Lee, DOE Reza Malek-Madani, ONR Grace Peng, NIH
12:15 PM–12:30 PM	Instructions for Day 1 Breakouts
12:30 PM–1:30 PM	Breakouts: Application Needs and Drivers
1:30 PM–1:45 PM	Break
1:45 PM–2:30 PM	Application Needs and Drivers Breakout Report Backs
2:30 PM–2:45 PM	Wrap-up Part 2, Day 1
2:45 PM–3:00 PM	Writing Committee Meeting

Part 2: January 7, 2021

11:00 AM–11:15 AM	Welcome to Day 2 and Summary of Day 1
11:15 AM–11:30 AM	Instructions for Day 2 Breakouts
11:30 AM–12:45 PM	Breakouts: Foundational/Computational Research
12:45 PM–1:00 PM	Break
1:00 PM–1:45 PM	Foundational/Computational Research Breakout Report Backs
1:45 PM–2:00 PM	Workshop Wrap-up
2:00 PM–3:00 PM	Writing Committee Meeting

Workshop Format

Part 1 of the workshop took advantage of expanded capacity enabled by a virtual format. Attendees pre-registered for the workshop and participated in discussion and breakouts. Additional attendees were able to follow the bootcamp (all of Part 1 except for the breakouts) in real time through a youtube stream.

Registration to Part 2 required submission of a 200-word thesis statement on long-term research needs. These statements were used to structure the breakouts in Part 2 and seed the report content.

B Workshop Participants

Name	Institution
Karthik Aadithya	Sandia National Laboratories
Erin Acquesta	Sandia National Laboratories
Nagesh Adluru	University of Wisconsin-Madison
Bulbul Ahmmed	Los Alamos National Laboratory
James Ahrens	Los Alamos National Laboratory
Sinan Aksoy	Pacific Northwest National Laboratory
H. Metin Aktulga	Michigan State University
Srinivas Aluru	Georgia Institute of Technology
Vinay Amatya	Pacific Northwest National Laboratory
Oluwamayowa Amusat	Lawrence Berkeley National Laboratory
Marian Anghel	Los Alamos National Laboratory
Mihai Anitescu	Argonne National Laboratory
Rick Archibald	Oak Ridge National Laboratory
Yeva F. Ashari	Institut Teknologi Bandung (ITB)
Selin Aslan	Argonne National Laboratory
Ahmed Attia	Argonne National Laboratory
Alan Ayala	University of Tennessee, Knoxville
Jacob Badger	University of Texas at Austin
Siwar Badreddine	INRIA
Zhe Bai	Lawrence Berkeley National Laboratory
Craig Bakker	Pacific Northwest National Laboratory
Prasanna Balaprakash	Argonne National Laboratory
Grey Ballard	Wake Forest University
David Barajas-Solano	Pacific Northwest National Laboratory
Andrew Barker	Lawrence Livermore National Laboratory
Andreas Bärtschi	Los Alamos National Laboratory
William Beckner	University of Texas at Austin
Getachew Befekadu	Morgan State University
Julie Bessac	Argonne National Laboratory
Vivek Bharadwaj	University of California, Berkeley
Noah Birge	Los Alamos National Laboratory
Simon Bolding	Los Alamos National Laboratory
Raghu Bollapragada	University of Texas at Austin
Erik Boman	Sandia National Laboratories
Brian Borchers	New Mexico Institute of Mining & Technology
Tyler Borgwardt	Los Alamos National Laboratory
Kerry Bossler	Sandia National Laboratories
Nicolas Boumal	Swiss Federal Institute of Technology Lausanne
Charles Bouman	Purdue University
Robert Bridges	Oak Ridge National Laboratory
Christopher Brislawn	Los Alamos National Laboratory
David Brown	Lawrence Berkeley National Laboratory
Johannes Brust	Argonne National Laboratory

Continued on next page

Name	Institution
Aydın Buluç	Lawrence Berkeley National Laboratory
Dmitry Burov	California Institute of Technology
Geunyeong Byeon	Argonne National Laboratory
Daan Camps	Lawrence Berkeley National Laboratory
Suma Cardwell	Sandia National Laboratories
Erin Carson	Charles University
Umit Catalyurek	Georgia Institute of Technology
Luis Chacon	Los Alamos National Laboratory
Moses Charikar	Stanford University
Sridhar Chellappa	Max Planck Institute, Magdeburg, Germany
Chao Chen	University of Texas at Austin
Xiao Chen	Lawrence Livermore National Laboratory
Changqing Cheng	Binghamton University
Hari Chhetri	Oak Ridge National Laboratory
Eric Chi	North Carolina State University
Jocelyn Chi	North Carolina State University
Jong Choi	Oak Ridge National Laboratory
Youngsoo Choi	Lawrence Livermore National Laboratory
Pinghan Chu	Los Alamos National Laboratory
Julianne Chung	Virginia Tech
Matthias Chung	Virginia Tech
Richard Clancy	University of Colorado
Aurora Clark	Washington State University
Lisa Claus	Lawrence Berkeley National Laboratory
Mark Coletti	Oak Ridge National Laboratory
Emil Constantinescu	Argonne National Laboratory
Alice Cortinovis	Ecole Polytechnique Federale de Lausanne
Martin Crawford	Sandia National Laboratories
Jody Crisp	Oak Ridge Institute for Science and Education
Eric Cyr	Sandia National Laboratories
Prasanna Date	Oak Ridge National Laboratory
Ieva Dauzickaite	University of Reading
Mike Davis	Cray/HPE
Eduardo D'Azevedo	Oak Ridge National Laboratory
Saibal De	University of Michigan, Ann Arbor
Eric de Sturler	Virginia Tech
Nathan DeBardeleben	Los Alamos National Laboratory
Suman Debnath	Oak Ridge National Laboratory
Victor DeCaria	Oak Ridge National Laboratory
Anthony DeGennaro	Brookhaven National Laboratory
Marta D'Elia	Sandia National Laboratories
James Demmel	University of California, Berkeley
Aditya Devarakonda	Johns Hopkins University
Som Dhulipala	Idaho National Laboratory

Continued on next page

Name	Institution
Zichao Wendy Di	Argonne National Laboratory
Wen Ding	Columbia University
Zhiyan Ding	University of Wisconsin, Madison
Edgar Dobriban	University of Pennsylvania
Jin Dong	Oak Ridge National Laboratory
Jack Dongarra	University of Tennessee
Ján Drgoňa	Pacific Northwest National Laboratory
John Duchi	Stanford University
Jed Duersch	Sandia National Laboratories
Alan Edelman	Massachusetts Institute of Technology
Karl Elbakian	Los Alamos National Laboratory
Amr El-Bakry	ExxonMobil Upstream Integrated Solutions
Eirik Endeve	Oak Ridge National Laboratory
Ethan Epperly	California Institute of Technology
N. Benjamin Erichson	University of California, Berkeley
Laureano Escudero	Universidad Rey Juan Carlos
Jennifer Estrada Lupianez	Los Alamos National Laboratory
Barry Fadness	Lawrence Livermore National Laboratory
Fariba Fahroo	AFOSR
Doreen Fan	Lawrence Berkeley National Laboratory
Yilin Fang	Pacific Northwest National Laboratory
Ionut Farcas	University of Texas at Austin
Ali Farghadan	University of Michigan
Maryam Fazel	University of Washington
Adrian Febre	MacLeod Ale Brewing Company
S. M. Ferdous	Purdue University
Gregory Fiechtner	DOE-BES
Thomas Flynn	Brookhaven National Laboratory
Chris Frazier	Sandia National Laboratories
Kasimir Gabert	Sandia National Laboratories
Jim Gaffney	Lawrence Livermore National Laboratory
Shreyas Gaikwad	University of Texas at Austin
Sayan Ghosh	Pacific Northwest National Laboratory
Sujit Ghosh	North Carolina State University
Pieter Ghysels	Lawrence Berkeley National Laboratory
Jay Gibble	Unaffiliated
Anna Gilbert	Yale University
Alex Gittens	Rensselaer Polytechnic Institute
Andrew Glaws	National Renewable Energy Laboratory
Abeynaya Gnanasekaran	Stanford University
Johnny Goett	Los Alamos National Laboratory
Emily Gottry	Azusa Pacific University
Carlo Graziani	Argonne National Laboratory
Andre Green	Los Alamos National Laboratory

Continued on next page

Name	Institution
Jared Greenwald	DOE-NNSA
Laura Grigori	INRIA
Sidharth GS	Los Alamos National Laboratory
Qiang Guan	Kent State University
Dan Gunter	Lawrence Berkeley National Laboratory
Castro Guzmán	University of São Paulo
Salman Habib	Argonne National Laboratory
Hamed Haddadi	Corning Incorporated
Aric Hagberg	Los Alamos National Laboratory
Mahantesh Halappanavar	Pacific Northwest National Laboratory
Kathleen Hamilton	Oak Ridge National Laboratory
Wesley Hamilton	University of North Carolina at Chapel Hill
David Han	University of Texas at San Antonio
Songfeng Han	Corning Incorporated
Joseph Hart	Sandia National Laboratories
Tucker Hartland	University of California, Merced
Rebecca Hartman-Baker	Lawrence Berkeley National Laboratory
Tamir Hasan	Los Alamos National Laboratory
Cory Hauck	Oak Ridge National Laboratory
Koby Hayashi	Georgia Institute of Technology
Barbara Helland	DOE-ASCR
Bruce Hendrickson	Lawrence Livermore National Laboratory
Stefan Henneking	University of Texas at Austin
Benjamin Hernandez	Oak Ridge National Laboratory
Felix Herrmann	Georgia Institute of Technology
Judy Hill	Oak Ridge National Laboratory
Jeffrey Hittinger	Lawrence Livermore National Laboratory
Yang Ho	Sandia National Laboratories
Thuc Hoang	DOE-NNSA
Aaron Holder	DOE-BES
David Hong	University of Pennsylvania
Shahadat Hossain	University of Lethbridge
Paul Hovland	Argonne National Laboratory
Yicheng Hu	University of Wisconsin, Madison
Zixi Hu	Lawrence Berkeley National Laboratory
Kuang Huang	Columbia University
Yehan Huang	Stanford University
Ryan Humble	Stanford University
Travis Humble	Oak Ridge National Laboratory
Xiaoming Huo	Georgia Institute of Technology
Hoon Hwangbo	University of Tennessee, Knoxville
Jeffery Hyman	Los Alamos National Laboratory
Mac Hyman	Tulane University
Mahdi Imani	George Washington University

Continued on next page

Name	Institution
Alexander Infanger	Stanford University
Vighnesh Iyer	University of California, Berkeley
Elchin Jafarov	Los Alamos National Laboratory
John Jakeman	Sandia National Laboratories
Jacek Jakowski	Oak Ridge National Laboratory
Carter Jameson	North Carolina State University
Ruhui Jin	University of Texas at Austin
Hans Johansen	Lawrence Berkeley National Laboratory
Erik Johnson	University of Colorado, Boulder
Patrick Johnstone	Brookhaven National Laboratory
Erika Jones	Federal Contractor
Piet Jones	Oak Ridge National Laboratory
Caleb Ju	Georgia Institute of Technology
Vesa Kaarnioja	LUT University, Finland
Gokberk Kabacaoglu	Flatiron Institute, Simons Foundation
Chandrika Kamath	Lawrence Livermore National Laboratory
Ulugbek Kamilov	Washington University in St. Louis
Raghavendra Kanakagiri	Indian Institute Of Technology Tirupati
Lulu Kang	Illinois Institute of Technology
Shinhoo Kang	Argonne National Laboratory
Ramakrishnan Kannan	Oak Ridge National Laboratory
Nikita Kapur	University of Kansas
Kolja Kauder	Brookhaven National Laboratory
William Kay	Oak Ridge National Laboratory
Jeffrey Keithley	Los Alamos National Laboratory
Carl Kelley	North Carolina State University
Luke Kersting	Sandia National Laboratories
Arif Khan	Pacific Northwest National Laboratory
Ratna Khatri	George Mason University
Joe Kileel	University of Texas at Austin
Aleksandra Kim	ETH Zurich
Kibaek Kim	Argonne National Laboratory
Kyungjoo Kim	Sandia National Laboratories
Tamara Kolda	Sandia National Laboratories
Tzanio Kolev	Lawrence Livermore National Laboratory
Hemanth Kolla	Sandia National Laboratories
Olivera Kotevska	Oak Ridge National Laboratory
Aditi Krishnapriyan	Lawrence Berkeley National Laboratory
Simge Kucukyavuz	Northwestern University
Paul Laiu	Oak Ridge National Laboratory
Kevin Lamb	Los Alamos National Laboratory
Brett Larsen	Stanford University
Jeffrey Larson	Argonne National Laboratory
Randall Laviolette	DOE-ASCR

Continued on next page

Name	Institution
Earl Lawrence	Los Alamos National Laboratory
Joel LeBlanc	Michigan Tech Research Institute
Steven Lee	DOE-ASCR
Rich Lehoucq	Sandia National Laboratories
Ryan Lekivetz	JMP
Nathan Lemons	Los Alamos National Laboratory
Andrew Leong	Los Alamos National Laboratory
Eitan Levin	California Institute of Technology
Matthew Levine	California Institute of Technology
Cannada Lewis	Sandia National Laboratories
Sven Leyffer	Argonne National Laboratory
Sherry Li	Lawrence Berkeley National Laboratory
Wenting Li	Los Alamos National Laboratory
Ying Wai Li	Los Alamos National Laboratory
Florence Lin	Retired
Meifeng Lin	Brookhaven National Laboratory
Neil Lindquist	University of Tennessee
Julia Ling	Citrine Informatics
Bowen Liu	North Carolina State University
Frank Liu	Oak Ridge National Laboratory
Jia Liu	University of West Florida
Jinsong Liu	University of Texas at Austin
Xinchao Liu	University of Arkansas
Yan Liu	Oak Ridge National Laboratory
Yang Liu	Lawrence Berkeley National Laboratory
Li-Ta Lo	Los Alamos National Laboratory
Lena Lopatina	Los Alamos National Laboratory
Miles Lopes	University of California, Davis
Vanessa Lopez-Marrero	Brookhaven National Laboratory
Vincenzo Lordi	Lawrence Livermore National Laboratory
Kathryn Lund	Charles University
Erik Lundgren	Los Alamos National Laboratory
Dalton Lunga	Oak Ridge National Laboratory
Hengrui Luo	Lawrence Berkeley National Laboratory
Xihaier Luo	Brookhaven National Laboratory
Massimiliano Lupo Pasini	Oak Ridge National Laboratory
Piotr Luszczek	University of Tennessee
Jonathan Maack	National Renewable Energy Laboratory
Arthur Maccabe	Oak Ridge National Laboratory
Sarah Mackay	Lawrence Livermore National Laboratory
Sandeep Madireddy	Argonne National Laboratory
Michael Mahoney	University of California, Berkeley
Dhairya Malhotra	Flatiron Institute, Simons Foundation
Osman Asif Malik	University of Colorado, Boulder

Continued on next page

Name	Institution
Indu Manickam	Sandia National Laboratories
Aniruddha Marathe	Lawrence Livermore National Laboratory
Oana Marin	Argonne National Laboratory
Osni Marques	Lawrence Berkeley National Laboratory
Zach Marshall	Lawrence Berkeley National Laboratory
Per-Gunnar Martinsson	University of Texas at Austin
David Dennis Mascarenas	Los Alamos National Laboratory
Estelle Massart	University of Oxford
Anna Matsekh	Los Alamos National Laboratory
Yury Maximov	Los Alamos National Laboratory
Lennon McCartney	Corning Incorporated
Ryan McClarren	University of Notre Dame
Hugh Medal	University of Tennessee
Maksim Melnichenko	University of Tennessee
Dean Menezes	University of California, Los Angeles
Matt Menickelly	Argonne National Laboratory
Agnieszka Miedlar	University of Kansas
Benjamin Miller	University of Texas at Austin
Richard Mills	Argonne National Laboratory
Michael Minion	Lawrence Berkeley National Laboratory
Narasinga Rao Miniskar	Oak Ridge National Laboratory
Raul Miranda	DOE
Laura Monroe	Los Alamos National Laboratory
Leslie Moore	Los Alamos National Laboratory
Jose Morales	University of Texas at San Antonio
Dmitriy Morozov	Lawrence Berkeley National Laboratory
Juliane Mueller	Lawrence Berkeley National Laboratory
Aliasgar Musani	Indian Institute of Technology Tirupati
Jeremy Myers	William & Mary, Sandia National Laboratories
Kary Myers	Los Alamos National Laboratory
Adan Myers y Gutierrez	Los Alamos National Laboratory
Samuel Myren	Los Alamos National Laboratory
Nicholas Nelsen	California Institute of Technology
Jelani Nelson	University of California, Berkeley
Esmond Ng	Lawrence Berkeley National Laboratory
Kam Ng	Corning Incorporated
Arnur Nigmatov	Lawrence Berkeley National Laboratory
Israt Nisa	Lawrence Berkeley National Laboratory
Marcus Noack	Lawrence Berkeley National Laboratory
Sewoong Oh	University of Washington
Thomas O'Leary-Roseberry	University of Texas at Austin
Vladyslav Oles	Oak Ridge National Laboratory
Eliza O'Reilly	California Institute of Technology
George Ostrouchov	Oak Ridge National Laboratory

Continued on next page

Name	Institution
Art Owen	Stanford University
Jonathan Ozik	Argonne National Laboratory
Mirjeta Pasha	Arizona State University
Vivak Patel	University of Wisconsin, Madison
Abani Patra	Tufts University
Jagabandhu Paul	California Institute of Technology
Grace Peng	NIH
Richard Peng	Georgia Institute of Technology
Talita Perciano	Lawrence Berkeley National Laboratory
Mauro Perego	Sandia National Laboratories
Ronan Perry	Johns Hopkins University
Elliott Perryman	Lawrence Berkeley National Laboratory
Kalyan Perumalla	Oak Ridge National Laboratory
Matt Peterson	Sandia National Laboratories
Russell Philley	University of Texas at Austin
Cynthia Phillips	Sandia National Laboratories
Eric Phipps	Sandia National Laboratories
Ali Pinar	Sandia National Laboratories
Maria Pinilla-Orjuela	Los Alamos National Laboratory
Robinson Pino	DOE-ASCR
Harun Pirim	Mississippi State University
Todd Plantenga	FireEye, Inc.
Nick Polydorides	University of Edinburgh
Ravi Ponmalai	University of California, Irvine
Gabriel Popoola	Sandia National Laboratories
Alex Pothén	Purdue University
Kumaraguru Prabakar	National Renewable Energy Laboratory
Benjamin Priest	Lawrence Livermore National Laboratory
Andrey Prokopenko	Oak Ridge National Laboratory
Constantin Puiu	Oxford University
Mihaela Quirk	DOE-NNSA
Robert Rallo	Pacific Northwest National Laboratory
Teresa Ranadive	University of Maryland
Vishwas Rao	Argonne National Laboratory
Navamita Ray	Los Alamos National Laboratory
Thomas Reeves	Cornell University
Lothar Reichel	Kent State University
Yihui Ren	Brookhaven National Laboratory
Rosemary renaut	Arizona State University
Ashwin Renganathan	Argonne National Laboratory
Viktor Reshniak	Oak Ridge National Laboratory
Juan Restrepo	Oak Ridge National Laboratory
Elina Robeva	University of British Columbia
Heinrich Roder	Biodesix

Continued on next page

Name	Institution
Theron Rodgers	Sandia National Laboratories
Anna Rörich	University of Stuttgart
Clément Royer	Université Paris Dauphine-PSL
Johann Rudi	Argonne National Laboratory
Minseok Ryu	Argonne National Laboratory
Anna Sabin	Pacific Northwest National Laboratory
Sonia Sachs	DOE-ASCR
Ilya Safro	University of Delaware
Cosmin Safta	Sandia National Laboratories
Roshni Sahoo	Stanford University
Arvind Saibaba	North Carolina State University
Jacob Salazar Solano	University of Texas at Austin
Zain Saleem	Argonne National Laboratory
Samitha Samaranyake	Cornell University
Arun Sathanur	Pacific Northwest National Laboratory
Florian Schaefer	California Institute of Technology
Drew Schmidt	Oak Ridge National Laboratory
Sudip Seal	Oak Ridge National Laboratory
Tom Seidl	Sandia National Laboratories
Oguz Selvitopi	Lawrence Berkeley National Laboratory
C. (Sesh) Seshadhri	University of California, Santa Cruz
John Shalf	Lawrence Berkeley National Laboratory
Nek Sharan	Los Alamos National Laboratory
Ruslan Shaydulin	Argonne National Laboratory
Deborah Shutt	Los Alamos National Laboratory
Erotokritos Skordilis	National Renewable Energy Laboratory
George Slota	Rensselaer Polytechnic Institute
Emily Smith	DOE-BES
J. Darby Smith	Sandia National Laboratories
Youngseok Song	Colorado State University
Braden Soper	Lawrence Livermore National Laboratory
Matthew Sottile	Lawrence Livermore National Laboratory
Raymond Spiteri	University of Saskatchewan
Prateek Srivastava	Rochester Institute of Technology
Andreas Stathopoulos	William & Mary
Gilbert Strang	Massachusetts Institute of Technology
Roy Streit	Metron
Nicholas Stull	Los Alamos National Laboratory
Omer Subasi	Pacific Northwest National Laboratory
Anirudh Subramanyam	Argonne National Laboratory
Aditya Sundararajan	Oak Ridge National Laboratory
Kelly Swanson	Lawrence Livermore National Laboratory
Katarzyna Swirydowicz	Pacific Northwest National Laboratory
Peter Tait	McMaster University

Continued on next page

Name	Institution
Yu-Hang Tang	Lawrence Berkeley National Laboratory
Valerie Taylor	Argonne National Laboratory
Keita Teranishi	Sandia National Laboratories
Satya Tetala	Colorado State University
James Theiler	Los Alamos National Laboratory
Yifeng Tian	Los Alamos National Laboratory
David Tralli	The Aerospace Corporation
Anh Tran	Sandia National Laboratories
Hoang Tran	Oak Ridge National Laboratory
Benjamin Treweek	Sandia National Laboratories
Joel Tropp	California Institute of Technology
Antonino Tumeo	Pacific Northwest National Laboratory
Raymond Tuminaro	Sandia National Laboratories
Bora Ucar	CNRS and LIP ENS de Lyon
Madeleine Udell	Cornell University
Juliette Ugirumurera	National Renewable Energy Laboratory
Ugochukwu Ugwu	Kent State University
Jiblal Upadhya	Middle Tennessee State University
Nathan Urban	Brookhaven National Laboratory
Roel Van Beeumen	Lawrence Berkeley National Laboratory
Nick Vannieuwenhoven	KU Leuven
Heidy Vega	University of California, San Diego
Nicolas Venkovic	Cerfacs
Jeff Victor	General Electric
Draguna Vrabie	Pacific Northwest National Laboratory
Richard Vuduc	Georgia Institute of Technology
Andreas Waechter	Northwestern University
Ji Wang	University of California, Davis
Zhehui (Jeph) Wang	Los Alamos National Laboratory
Ziteng Wang	Northern Illinois University
Bei Wang Phillips	University of Utah
Ben Whitney	Oak Ridge National Laboratory
Stefan Wild	Argonne National Laboratory
Don Willcox	Lawrence Berkeley National Laboratory
David Williams-Young	Lawrence Berkeley National Laboratory
Vince Winstead	Minnesota State University, Mankato
Jonathan Wittmer	University of Texas at Austin
Brendt Wohlberg	Los Alamos National Laboratory
David Womble	Oak Ridge National Laboratory
Stephen Wright	University of Wisconsin, Madison
John Wu	Lawrence Berkeley National Laboratory
Penghao Xiao	Lawrence Livermore National Laboratory
Yu Xiao	Corning Incorporated
Miaolan Xie	Cornell University

Continued on next page

Name	Institution
Weijun Xie	Virginia Tech
Yuege Xie	University of Texas at Austin
Ricky Yan	Unaffiliated
Chao Yang	Lawrence Berkeley National Laboratory
Yunan Yang	New York University
Jia Yin	Lawrence Berkeley National Laboratory
Srikanth Yoginath	Oak Ridge National Laboratory
Stephen Young	Pacific Northwest National Laboratory
Roozbeh Yousefzadeh	Yale University
Kwangmin Yu	Brookhaven National Laboratory
Alfi Zakiyyah	Bina Nusantara University
Rohit Zambre	University of California, Irvine
Boya Zhang	Lawrence Livermore National Laboratory
Guannan Zhang	Oak Ridge National Laboratory
Jiaxin Zhang	Oak Ridge National Laboratory
John Zhang	University of California, Los Angeles
Pei Zhang	Oak Ridge National Laboratory
Xinyu Zhang	North Carolina State University
Yifan Zhang	University of Texas at Austin
Ziyun Zhang	California Institute of Technology
Lili Zheng	University of Wisconsin, Madison
Dekun Zhou	University of Wisconsin, Madison
Yanhui Zhu	University of West Florida
Yue Zhu	Pacific Northwest National Laboratory
Roger Zoh	Indiana University
Zihui Zou	University of Texas at Austin
Petrus Zwart	Lawrence Berkeley National Laboratory

C Acknowledgments

We are grateful to many contributors to the process that led to the composition of this report on an ambitious timeline.

We are grateful to the staff at ORISE (Jody Crisp, Andrew Fowler, and Paul Hudson, in particular) for managing the logistics of a large, multipart virtual workshop.

Special thanks are due to the presenters at the bootcamp. Miles Lopes, C. (Sesh) Seshadhri, Per Gunnar Martinsson, and John Duchi coordinated to provide a cohesive set of tutorials. Julia Ling, Salman Habib, Aurora Clark, Charles Bouman, and Peter Zwart shared valuable perspectives on challenges and opportunities in their respective scientific areas.

We are grateful to the breakout leads for the bootcamp, who were invaluable in facilitating discussion among randomly assigned breakout participants. These leads included Jim Ahrens, Rick Archibald, Eric Cyr, Anthony DeGennaro, Jed Duersch, Aric Hagberg, Joey Hart, Rebecca Hartman-Baker, Jeff Hittinger, Paul Hovland, John Jakeman, Chandrika Kamath, Ramki Kannan, Hemanth Kolla, Rich Lehoucq, Sven Leyffer, Sherry Li, Osni Marques, Kary Myers, Jelani Nelson, Esmond Ng, Cindy Phillips, Eric Phipps, Ali Pinar, Vishwas Rao, Juan Restrepo, Ray Tuminaro, Draguna Vrabie, David Womble, Steve Wright, Chao Yang, and Stephen Young.

We thank the federal agency panelists Fariba Fahroo (AFOSR), Steven Lee (DOE), Reza Malek-Madani (ONR), and Grace Peng (NIH) for sharing insight on the status and opportunities for randomized algorithms in each of their agencies.

Breakout leads on both days of Part 2 distilled significant attendee input. We are grateful to Mihai Anitescu, Charlie Bouman, Aurora Clark, Anthony DeGennaro, Salman Habib, Chandrika Kamath, Ramki Kannan, Miles Lopes, Per Gunnar Martinsson, Kary Myers, Jelani Nelson, Cindy Phillips, Juan Restrepo, C. Seshadhri, Joel Tropp, Draguna Vrabie, Brendt Wohlberg, Steve Wright, Chao Yang, and Peter Zwart for serving in this role.

We thank Tiffani Conner and Gail Pieper for their technical editing of the report.

We thank Steven Lee for the charge to identify research needs in randomized algorithms for scientific computing to advance the mission of DOE's Office of Science.