Future Directions in CSE Education and Research

Report from a Workshop Sponsored by the Society for Industrial and Applied Mathematics (SIAM) and the European Exascale Software Initiative (EESI-2)

August 4-6, 2014; Breckenridge, Colorado

[DRAFT: Compiled on Wednesday 11th March, 2015 at 10:06]

Workshop Organizers:

Officers of the SIAM Activity Group on Computational Science and Engineering (CSE), 2013-2014:

Ulrich Rüde, Universität Erlangen-Nürnberg, Chair
Karen Willcox, Massachusetts Institute of Technology, Vice Chair
Lois Curfman McInnes, Argonne National Laboratory, Program Director
Hans De Sterck, University of Waterloo, Secretary

Additional Contributors:

Hans Bungartz, Technische Universität München
James Corones, Krell Institute
Evin Cramer, Boeing
James Crowley, SIAM
Omar Ghattas, University of Texas at Austin
Max Gunzburger, Florida State University
Michael Hanke, KTH Stockholm
Robert Harrison, Brookhaven National Laboratory and Stonybrook University
Peter Jimack, University of Leeds
Chris Johnson, University of Utah
Kirk Jordan, IBM
David Keyes, KAUST
Rolf Krause, Università della Svizzera Italiana, Lugano
Vinay Kumar, University of Minnesota
Stefan Mayer, MSC Software, Munich
Juan Meza, University of California, Merced
Knut Martin Morken, University of Oslo
J. Tinsley Oden, University of Texas at Austin
Linda Petzold, University of California, Santa Barbara
Padma Raghavan, Penn State University
Anne Trefethen, University of Oxford
Peter Turner, Clarkson University
Vladimir Voevodin, Moscow State University
Barbara Wohlmuth, Technische Universität München
Carol Woodward, Lawrence Livermore National Laboratory
Abstract

Over the past two decades the field of computational science and engineering (CSE) has penetrated the academic and industrial world, with prominent roles in advancing research and innovation and providing interdisciplinary education. However, a combination of disruptive developments—including challenges in extreme-scale parallel computing, the emergence of big data and data-driven discovery, and a comprehensive broadening of the application fields of CSE—is redefining the scope and reach of the CSE endeavor. Mathematics-based advanced computing has become a prevalent means of discovery and innovation in almost all areas of science, engineering, technology, and society; and CSE takes a central role in this transformation. This report describes the rapid expansion of CSE and the challenges to the field. The document presents new strategies and directions for CSE in education and research for the next decade.
# Contents

1 Definition and Importance of CSE
   1.1 Definition of CSE ......................................................... 1
   1.2 Goal of this Document .................................................. 1
   1.3 Importance of CSE ........................................................ 2
   1.4 CSE as an Emerging Discipline ......................................... 2
   1.5 The Broad CSE Community .............................................. 5
   1.6 Organization of this Document ....................................... 6

2 CSE Success Stories .......................................................... 6
   2.1 Application Highlights .................................................. 7
   2.2 Advances in CSE Methods, Algorithms, and Software .......... 9
   2.3 The Emergence of Predictive Computational Science and Engineering .................................................. 10

3 CSE and High-Performance Computing—Ubiquitous Parallelism .... 10
   3.1 The Symbiotic Relationship between CSE and HPC ............... 10
   3.2 Ubiquitous Parallelism: A Phase Change for CSE Research and Education ................................................. 11
   3.3 Extending the Scope of CSE through HPC Technology ........ 12
   3.4 Emergent Topics in HPC-Related Research and Education .... 13

4 CSE and the Data Revolution: The Synergy between Computational Science and Data Science ........................................... 15
   4.1 Relation of CSE to Big Data Analytics and Data Science ...... 15
   4.2 Role of Massive Data in CSE ............................................ 16

5 CSE Education and Workforce Development ........................... 18
   5.1 Growing Demand for CSE Graduates in Industry, National Labs, and Academic Research ......................... 18
   5.2 Future Landscape of CSE Educational Programs ................. 19
   5.3 Graduate Program Learning Outcomes ............................... 20
   5.4 Education in Parallel Computing and Extreme-Scale Computing .................................................. 22
   5.5 CSE Education in Uncertainty Quantification and Big Data .................................................. 23
   5.6 Changing Educational Infrastructure .................................. 24

6 Recommendations and Conclusions ....................................... 24
Acknowledgments ............................................................... 25
References ........................................................................ 26
1 Definition and Importance of CSE

1.1 Definition of CSE

Computational science and engineering (CSE) is a multidisciplinary field of research and education lying at the intersection of applied mathematics, computer science, and core disciplines of science and engineering (Figure 1). CSE is devoted to the development and use of computational methods for scientific discovery in all branches of the sciences and for the advancement of innovation in engineering and technology. It is a broad and vitally important field encompassing methods of high-performance computing (HPC) and playing a central role in the data revolution. It is affecting every aspect of science and technology and is changing higher education worldwide.

Figure 1: CSE at the intersection of mathematics and statistics, computer science, and target disciplines from the sciences and engineering. This combination gives rise to a new field whose character is different from its original constituents.

CSE is rooted in the mathematics and physical sciences and engineering, where it is now widely recognized as an essential cornerstone that drives scientific and technological progress in conjunction with theory and experiment. Since the late 20th century CSE has helped revolutionize the life sciences and medicine, and in the 21st century its pivotal role is further expanding to increasingly broad areas that now include the social sciences and humanities, business and finance, and government policy.

1.2 Goal of this Document

The goal of this document is to present the expanding role of CSE in the 21st-century landscape of research and education. It follows in the footsteps of the 2001 report on graduate education in CSE by the SIAM Working Group on CSE Education [31], which was instrumental in setting directions for the nascent CSE field 15 years ago. It also relates to a more recent report on undergraduate CSE education [36]. The current document describes and investigates the rapid expansion of CSE since the beginning of the 21st century and the challenges the CSE field is encountering in the context of recent disruptive developments resulting from emerging extreme-scale systems, big data, and data-driven discovery, and from a comprehensive broadening of the application fields of CSE. The document presents new strategies and directions for CSE in education and research for the next decade. Among the many exciting challenges and opportunities that arise for CSE, this report focuses in particular on fundamental
evolutions in two areas: the emerging ubiquitous parallelism in high-end computing, and the sea change provoked by the data revolution. The implications for CSE education and workforce development are also investigated.

1.3 Importance of CSE

In its core science and engineering applications, CSE closes the centuries-old gap between theory and experiment, by providing technologies that can convert theoretical models into predictive simulations and by devising systematic methods to integrate experimental data with algorithmic models. CSE increasingly becomes the driver for progress in those applications where classical experiments or theory alone have reached their limitations or where experimental approaches are costly, slow, or dangerous. Examples include automobile crash tests, nuclear test explosions, emergency flight maneuvers, and operator emergency response training. Experiments in fundamental science may even be impossible when the systems under study span microscopic or macroscopic scales in space or time that are beyond human reach. Although a traditional theoretical analysis would not suffer from these limitations, theory alone lacks the predictive power that is often needed. For example, the well-established mathematical models for fluid flow provide an accurate theoretical description of the atmosphere. Only when combined with the power of numerical simulation, however, do these equations become useful for predicting tomorrow’s weather or designing more energy-efficient airplane wings. Another example is the use of simulation models to conduct systematic virtual experiments of exploding supernovae: CSE technology serves as a virtual telescope reaching farther than any real telescope, expanding human reach into outer space. And computational techniques can equally well serve as a virtual microscope, being used to understand quantum phenomena at scales so small that no real microscope could resolve them.

The growing importance of CSE in increasingly many application areas has paralleled the tremendous, exponential growth that has occurred in computing power over the past five decades according to Moore’s law. Less well known but crucially important for the success of the CSE paradigm is the fact that these advances in computing power have been matched or exceeded by equivalent advances over the past five decades in the efficiency of the mathematics-based computational algorithms that lie at the heart of CSE (Figure 2). Indeed, the development of efficient new algorithms continues to be a core focus of CSE, which is crucial to the effective use of advanced computing capabilities to address the pressing problems of humankind.

The emergence and growing importance of massive data sets in many areas of science, technology, and society, in conjunction with the availability of ever-increasing parallel computing power, are transforming the world. Data-driven approaches enable novel ways of scientific discovery and quantifying uncertainties in science and engineering applications. At the same time, relying on new forms of massive data, we can now use the quantitative and model-based approaches that characterize “the scientific method” to drive progress in many areas of society where qualitative forms of analysis, understanding, and decision making were the norm until recently. The CSE paradigm contributes as a cornerstone technology to the data revolution. In these and many other ways, CSE is becoming essential for increasingly broad areas of science, engineering, and technology. It expands human capability beyond its classical limitations.

The enormous magnitude of the CSE challenge can be appreciated when considering the dramatic growth of disciplinary complexity alone. Dominating impediments in many sciences are model complexity and the uncertainty in parameters and data, as well as the huge space and time scales that must be resolved. In computational mathematics, the traditional abstract view of optimality becomes more and more obsolete and must be replaced by new performance-oriented metrics that reflect the true computational cost. In computing, the disruptive transition from modest concurrency to ubiquitous parallelism leaves software development trailing behind. In CSE all these challenges converge and then often pile up to obstructive roadblocks. In research and education these barriers must be overcome by cutting new lanes across the disciplinary borders and by creating a new paradigm of coalescence among the classical disciplines.

1.4 CSE as an Emerging Discipline

CSE is unique in that it enables progress in virtually all other disciplines by providing windows of discovery when traditional means of research and development reach their limits.

To apply the CSE approach to a specific problem in the physical sciences or engineering, researchers generally first design a simulator that can correctly represent what has been experimentally observed. The creation of a
In addition to application domain knowledge, CSE research requires expertise in advanced computing. This includes elements of mathematics and statistics, scientific computing, and computer science. Additionally, many CSE tasks involve data analysis to provide suitable input for the computations, and visualizations are required to help humans interpret the simulation outcome. Thus, many CSE problems can be characterized by a pipeline approach.
that includes mathematical modeling techniques (based on physical or other principles), simulation techniques (discretizations of equations, solution algorithms, data structures, software frameworks, and problem solving environments), and analysis techniques (data mining, data management, and visualization, as well as the analysis of error, sensitivity, stability, and uncertainty). In practice the CSE pipeline is a loop connected through multiple feedbacks, as illustrated in Figure 3.

CSE research often creates much wider impact than just developing a method that is applicable to one area of science or engineering, or by just delivering a single specific insight in a science or engineering discipline. CSE methods and findings tend to have broad applicability, and the abstract concepts of algorithm and method development that form an integral part of the common core of CSE generally apply to a wide range of disciplinary problems where they then lead to breakthroughs that could not be achieved otherwise. For example, a method to simulate large ensembles of interacting particles may be useful to an astrophysicist studying galaxy formation as well as to a nanotechnology researcher exploring molecular dynamics. Ultimately, CSE aims at developing a universal set of simulation methods and tools for the scientists and engineers of the future. This universality of CSE must be reflected in education, in institutional structures, and in funding programs.

Nevertheless, high-level CSE research differs from research in mathematics or computer science in that it is typically oriented to one or more realistic applications of a target discipline in the sciences, engineering, or technology. Such problems often involve complicated three-dimensional geometries, multiple interacting scales, heterogeneities, anisotropies, and multiphysical or biological descriptions; or they may involve complex networks or systems with many components. Thus, the methods developed often thwart rigorous proofs of accuracy or efficiency, and so CSE research must regularly address validation and verification by means other than traditional mathematical analysis. Additionally, CSE research and education cannot be limited to any single step of the simulation pipeline. A mathematical simulation algorithm is useful in CSE only when it is also implemented on a real computer, and a new parallel processing paradigm becomes relevant for CSE only when it is demonstrated to be applicable for a large-scale numerical computation. Of course, research on an individual piece of the CSE
pipeline, such as the development of a new finite-element method, may also be useful for progress in CSE, even though taken by itself its main significance would be as a work in numerical analysis.

Invention and development of new computational algorithms
User-friendly presentation of solutions
Mathematical modeling
Analysis of mathematical models
Invention and development of new computational algorithms
Analysis of computational algorithms
Development of CSE software with efficient algorithm implementation
Computational solution of applications problems
User-friendly presentation of solutions

Figure 4: Interaction among different CSE components. The development of new algorithms and software is at the core of this view of CSE.

In other words, CSE is not just a new name for existing computational disciplines, nor—least of all—is it just mathematics in which the proofs are half-finished. In fact, as opposed to other mathematical sciences, CSE often achieves its progress through a clever combination of techniques and methods employed for the different stages of the CSE pipeline. In such a case, the innovation that characterizes excellent research may consist in the creativity needed to synthesize a computational solution for a complex problem from the right building blocks. The structure of CSE research and education is illustrated in Figure 4. In essence, CSE research and practice are profoundly interdisciplinary, so that the synthesis achieves a quality that is fundamentally new and unique. CSE researchers and practitioners require a diverse skill set with a focus and drive that is different from what any of the constituent disciplines of mathematics, computer science, and science and engineering can offer by themselves.

1.5 The Broad CSE Community

The past two decades have seen tremendous growth in the CSE community, including a dramatic increase in both size and breadth of intellectual perspectives and interests. The growth in community size can be seen, for example, through the membership of the SIAM Activity Group on CSE, which has grown from approximately 1,000 to 2,000 since 2005. The biennial SIAM CSE Conference [34] has grown from about 400 attendees in 2000 to more than 1,300 attendees at the 2013 conference and is now SIAM’s largest conference. The increased breadth of the community is evidenced in many ways: by the diversity of minisymposium topics at SIAM CSE conferences; through a new broader structure for SIAM’s Journal on Scientific Computing, including a new journal section that focuses on computational methods in specific problems across science and engineering; and by the sharply increased use of CSE approaches in industry over the past decade [29, 14].

As we envision the future of CSE, and in particular as we consider educational programs, we must keep in mind that such a large and broad intellectual community has a correspondingly broad set of needs. Figure 5 presents one way to view the different aspects of the broad CSE community: CSE Core Researchers and Developers—those engaged in the conception, analysis, development, and testing of CSE algorithms and software; and CSE Domain Scientists and Engineers—those primarily engaged in applying core CSE algorithms and software in particular science and engineering campaigns. The latter community can usefully be further subcategorized into those who interact with the core technologies at a developer level within their own applications, creating their own implementations and contributing to methodological/algorithmic improvements, and those who are happy to
use state-of-the-art CSE technologies as products, combining them with their expert knowledge of an application area to push the boundaries of a particular application. Within the CSE Core Researchers and Developers group in Figure 5, we further identify two groups: those focused on broadly applicable methods and algorithms and those focused on methods and algorithms motivated by a specific domain of application. We make this distinction because it is a useful way to cast differences in desired outcomes for different types of CSE educational programs, discussed in detail in Section 5. As with any categorization, the dividing lines in Figure 5 are fuzzy, and in fact any single researcher might span multiple categories.

![Figure 5: One view of the different aspects of the broad CSE community. The part of the CSE community that focuses on developing new methods and algorithms is labeled CSE Core Researchers and Developers. They may be driven by generally applicable methods, or by methods developed for a specific application domain. CSE Domain Scientists and Engineers focus their work primarily in their scientific or engineering domain, but make extensive use of CSE methods in their research or development work. The subcategory of ‘method developers’ interacts with computational methods and algorithms at a deep level through implementation and method improvements. The subcategory of ‘method users’ are end users who often combine the use of advanced CSE methods with a strong focus on applications.](image)

1.6 Organization of this Document

The remainder of this document is organized as follows. Section 2 gives an overview of key progress in CSE over the past two decades, invoking success stories of CSE in a variety of application domains and crosscutting advances in methods, algorithms, and software. Section 3 discusses a crucial challenge that CSE is addressing for the coming decade: the ever-increasing parallelism in computing hardware and the drive toward extreme-scale applications require fundamentally new ways to consider high-performance computing for CSE. Section 4 discusses a second major game changer in CSE: the ongoing data revolution offers tremendous opportunities for breakthrough advances in science and engineering, exploiting new techniques and approaches that are significantly expanding computational science in synergy with data science. Section 5 discusses how these changes are affecting the needs and goals of CSE education and workforce development. Section 6 formulates recommendations and conclusions for the future of CSE over the next decade.

2 CSE Success Stories

CSE has become indispensable for leading-edge scientific investigation and engineering design that increasingly rely on advanced modeling and simulation as a complement to theory and experiment. As stated by the 2005 PITAC Report [4], “Computational science is one of the most important technical fields of the 21st century because it is essential to advances throughout society ... [it] provides a unique window through which researchers can investigate problems that are otherwise impractical or impossible to address.” For example, CSE addresses questions such as the following:
2 CSE SUCCESS STORIES

- What is the best therapy for a patient with a specific disease that minimizes risk and maximizes success probability?
- What are the likely results of hurricanes, tornados, storm surges on coastal regions and what plans can be implemented to minimize losses of human life and property?
- What are the intricate functions of the human brain, the human nervous system, and the cardiovascular system, and how can we better understand them so as to prolong or improve our quality of life?
- How will our climate evolve and how can we predict the outcomes of climate change?
- How quickly could a region recover if part of the power grid became inoperable?

2.1 Application Highlights

We highlight a few examples to illustrate how combined advances in CSE theory, analysis, algorithms, and software have made CSE technology indispensable for applications throughout science and industry. A rich variety of CSE application advances, in areas such as astrophysics, biology, chemistry, climate modeling, combustion-energy science, fusion-energy science, gamma ray bursts, geodynamics, hazard analysis, high-temperature superconductor material design, human sciences and policy, hydrology, materials science, management of greenhouse gases, nuclear energy, particle accelerator design, and virtual product design, is discussed in [22, 6, 27, 7, 29, 23] and highlighted in diverse presentations in the SIAM CSE conference series [34].

**Computational medicine.** Computational medicine has always been at the frontier of CSE; the virtual design and testing of new drugs and therapies accelerates medical progress and reduces cost for development and treatment. The complex processes within the human body lead to elaborate multiscale models, as shown for the macro-or organ-scale in Figure 2.1. For example, cardiac function builds on a complicated interplay between different temporal and spatial scales (i.e., body, organ, cellular level, and molecular levels), as well as different physical models (i.e., mechanics, electrophysiology, fluid mechanics, and their interaction). At the micro-scale, computational drug design and computational biology employ methods such as molecular dynamics and coarse graining, where processes such as protein binding or the action of a virus or an antigen can be investigated through numerical simulation. Another prominent example of CSE in medicine is computational neuroscience, which is of major importance in ongoing efforts to understand the human brain.

![Computational mechanics and fluid-structure interaction models](image.png)

**Figure 6:** (Left): Computational mechanics and fluid-structure interaction models are routinely used for the design and optimization of implants (e.g., for the spine, knee, hip, and stents); shown here is a percutaneous interspinous spacer implantation. (Right): Simulation of the electrical-mechanical activity of the human heart. We refer to [10] for parallel and adaptive techniques and to [2] for a description of a coupled elector-mechanical model.

CSE advances in computational medicine, which have thus far focused primarily on the classical “forward problem” introduced in Section 1, already are helping in, for example, placing electrodes for pacemakers or studying diseases such as atrial fibrillation. And opportunities abound for next-generation CSE advances. The solution of inverse problems can help identify correct values for material parameters, for example, to detect scars or infarctions. Using uncertainty quantification, we can estimate the influence of varying material parameters (fiber directions, sheets, scars, etc.) or varying geometry. Indeed, the inclusion of stochastic effects is of particular
importance in computational medicine, since data often is unknown and as many (micro-)scale processes are best modeled by using stochastic approaches. In bioinformatics, these simulation tools are accompanied by methods for analyzing biological data, in particular in genetics and genomics. New scientific fields such as systems biology and systems medicine aim at combining and integrating the huge amount of data and information available through measurements, simulation, and analysis in order to get a better understanding of the complex and coupled biological processes in our body.

**Automotive industry.** CSE-based simulation, namely, computer-aided design (CAD) and computer-aided engineering (CAE) methods and tools, has become an indispensable component of developing advanced products in industry. Based on appropriate mathematical models, such as differential equations and variational principles, CAE methods such as multibody simulation, finite elements, and computational fluid dynamics are essential for assessing the functional behavior of products during development. Besides the many obvious advantages of virtual testing compared with physical testing, such as flexibility, speed, and cost, another important benefit is that CAE has enabled manufacturers to test the functional behavior early in the design cycle of new products, when physical prototypes are not yet available. As an example, Figure 2.1 shows selected application areas of CAE in the automotive industries. Note that in accordance with the intention of CSE to provide widely applicable methods and tools, similar CAE methods and tools are used in different engineering applications. For example, drop tests of mobile phones are investigated by applying simulation methods that are also used in automotive crash analysis.

![Figure 7: Examples of CAE in the automotive industry (courtesy of AUDI AG).](image)

**Simulation-based optimization of a 3D printing process.** Complex manufacturing systems depend increasingly on CSE-based simulation and optimization. For example, Figure 2.1 illustrates a 3D printing application [2] or additive manufacturing process, where thin layers of metal powder are molten by a high-energy electron beam that welds the powder selectively to create complex 3D metal structures with an almost arbitrary geometry by repeating the process layer by layer.

Simulation can be used for designing the electron beam gun, developing the control system, and generating the powder layer, thereby accelerating the printing process in commercial manufacturing, for example, of patient-specific medical implants. Possibly the greatest simulation challenge, however, is to develop numerical models for the welding process itself—a complex 3D multiphysics problem that involves the modeling of thermal energy transfer from the electron beam to the powder bed, the melting of the powder particles, and the flow of the melt.
with a free surface and that is determined largely by surface tension and wetting effects. A realistic simulation with physical resolution of a few microns requires millions of mesh cells and several hundreds of thousands of time steps—computational complexity that can be tackled only with parallel supercomputers and sophisticated software. Figure 2.1 visualizes the results of a simulation performed by large-scale computing frameworks as described in [24].

2.2 Advances in CSE Methods, Algorithms, and Software

These compelling CSE success stories stem from breakthroughs in applied mathematics and computer science that have dramatically advanced simulation capabilities. These advances include fundamental results in computational methods and algorithms—for example, the rich body of theory that underlies modern differential equation solvers; advanced discretization methods for PDEs; scalable solvers for the core problems in numerical (non)linear algebra that underlie large-scale applications across science, engineering, and business, from image recognition to fusion-energy science to page rank algorithms; the mathematical formulations and algorithms of computational geometry that together with differential equation solvers have revolutionized engineering design through CAD/CAE tools; the graph and combinatorial algorithms that have changed the way we model and operate large-scale networked systems; and large-scale simulation-based optimization. Equally important is tremendous progress in high-performance computing systems, including development-related and production-related computing technologies, such as operating systems, programming models, programming languages, compilers, configuration systems, debuggers, profilers, source-to-source translators, messaging systems, I/O systems, dynamic resource management, performance optimization, data handling, workflow controllers, and visualization.

An implicit theme throughout CSE advances in applications, applied mathematics, and computer science is that well-designed and flexible CSE software is a cornerstone for long-term interdisciplinary collaboration. As shown in Figure 4, the CSE research challenges at the heart of many applications lie in combining sophisticated CSE methods and algorithms as expressed in software (for example, to address nonlinearly coupled multiphysics and multiscale scenarios in complex geometries). That is, CSE software is a key crosscutting technology that connects advances in mathematics, algorithms, and domain-specific science and engineering to achieve robust and efficient simulations on advanced computing systems.

As CSE applications increase in sophistication, with goals of predictive science, design, and control, no single research team possesses the expertise and resources to address all aspects of a simulation. Interdisciplinary collaboration using software developed by independent groups becomes essential, bringing to the forefront issues of
CSE software interoperability. Added complexities arise for applications that need to run efficiently in parallel with many thousands of cores (discussed further in Section 3), where each algorithmic component and each data structure must be suitable for parallel execution and scalable performance. Formidable challenges in algorithm design and software engineering thus arise, especially since CSE software must be flexible and extensible to permit future changes. Thus, the role of software as a foundation for continuing CSE advances is driving attention to issues of software sustainability, interoperability, and productivity [21, 17], sorely needed in the quest for ever-increasing sophistication of CSE simulations in an era of disruptive computing architecture changes.

### 2.3 The Emergence of Predictive Computational Science and Engineering.

A perhaps unanticipated byproduct of advances in CSE has been the realization that every step in the process of scientific discovery, particularly through computer modeling, simulation, big data analytics, and HPC, confronts inherent uncertainties that strongly affect the outcomes of computer predictions. This uncertainty is not a feature of just computer models, although discretization errors do corrupt predictions, but is a fundamental attribute of the scientific method, which relies on observational data, always corrupted by experimental error, and the models themselves, which not only are imperfect abstractions of reality, but which are characterized by unknown or imperfectly known and usually random parameters. The new predictive science that has emerged within the last decade and that addresses these issues can be referred to as **predictive CSE**. It embraces the mathematics, statistics, philosophy, and basic science underlying model selection, model calibration, model validation, model and code verification, all in the presence of uncertainties, and ultimately, the propagation of uncertainties through the forward problem or the inverse problem, to the quantification of uncertainties of outputs, the target goals of the simulation. When actual computer predictions are used for critical life and death decisions, all of these sources of uncertainty must be addressed. Ironically, many of the algorithms called upon today to cope with these issues have their roots in the mathematics and statistics of the last century and beyond, but their extensions and performance on today’s computer architectures are often unsatisfactory. New algorithms, models, and methodologies that move this area forward will require significant breakthroughs. The need for new algorithms that cope with all of the complexities of predictive modeling stands as a challenging and fundamentally important goal for CSE research. The result of understanding the issues affecting predictability has been a gradual transformation of thinking on how computer models are perceived, designed, tested, and used. We believe this new look at modeling will gradually change the very definition of “solutions” to forward and inverse predictions problems; it will completely change the nature of algorithms, data management protocols, and, importantly, the very definition of what a prediction means—with, for example, probability distributions displacing deterministic predictions. The emergence of predictive CSE will have a profound impact on education in CSE, on how software is developed, and possibly on the architecture of tomorrow’s computers.

### 3 CSE and High-Performance Computing—Ubiquitous Parallelism

The development of CSE and high-performance computing are closely interlinked, since the availability of sufficient computational power is a prerequisite for many CSE research topics. Today’s CSE research is enabled by computer technology and driven into ever more complex simulations, in ever more disciplines, thanks to the rapid growth of available compute power. The most challenging supercomputer applications generally arise from simulations in science and engineering.

#### 3.1 The Symbiotic Relationship between CSE and HPC

These so-called grand challenge applications exercise computational technology at its limits (and beyond). They arise, for example, when complex coupled multiphysics problems require the solution of multiple physical fields over many time steps. Such simulations become especially compute intensive when the models require extremely fine temporal and spatial resolution, as is regularly the case when phenomena on different scales in space and time interact and when this multiscale coupling is studied computationally. The emergence of CSE as a fundamental pillar of science has occurred because computer technology is finally beginning to deliver sufficient compute power to create effective computational models and because tremendous algorithmic advances over the past decades have...
Table 1: Snapshot of architectural aspects of current supercomputing and their impact on designing efficient algorithms and high performance software.

<table>
<thead>
<tr>
<th>Core</th>
<th>Node</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Instruction pipelining</td>
<td>• Several CPU chips (currently often 2 or 4) may be</td>
<td>• Thousands of nodes connected by a fast network</td>
</tr>
<tr>
<td>• Superscalar execution</td>
<td>combined with local memory to become a node</td>
<td>• Different network topologies (tree, hypercube, etc.) are common</td>
</tr>
<tr>
<td>• Vectorization (i.e., vectors of typically 2-8 floating-point numbers must be treated in blocks)</td>
<td>• Several cores (currently typically 8) are on a CPU chip</td>
<td>• Often network has hierarchical structure itself</td>
</tr>
<tr>
<td>• Branches and branch prediction affect performance (e.g., in tight loops)</td>
<td>• Within a node shared-memory parallelism can be used</td>
<td>• Message passing must be used between nodes</td>
</tr>
<tr>
<td>• Cores may have their own cache memories</td>
<td>• E.g. programming with OpenMP</td>
<td>• Programming, e.g., is done with MPI</td>
</tr>
<tr>
<td>• Access to local (cache) memory is fast</td>
<td>• Several cores may share second/third level caches</td>
<td>• High latencies require message aggregation</td>
</tr>
<tr>
<td>• Access to remote memory is slow</td>
<td>• Memory access bottlenecks may occur</td>
<td>• Low bandwidth requires a careful algorithm design</td>
</tr>
<tr>
<td>• Prefetching favors uniform memory access</td>
<td>• Memory pinning may be essential</td>
<td>• Often programming is done with domain partitioning</td>
</tr>
<tr>
<td>• Multithreading: each core may need several threads to hide memory access latency</td>
<td>• Nodes may be equipped with accelerators (i.e., graphics cards)</td>
<td></td>
</tr>
</tbody>
</table>

delivered algorithmic speedup that is at least on par with the increase in raw computational power provided by the hardware (see also Figure 2).

On the other hand, the computational needs of such CSE applications are a main driver for HPC research. CSE applications often require closely interlinked systems, where not only the aggregate power but also the tight interconnection between systems is essential. This requirement often differentiates scientific computing in CSE from other uses of information processing systems. In particular, many CSE applications cannot be served efficiently by weakly coupled networks of nodes, as available in grid computing or generic cloud computing services. Consequently, the CSE-specific requirements for HPC systems often include high-bandwidth and low-latency interconnects. These specific prerequisites have been a main motivation for the development of high-end computer systems and software.

Thus, HPC and CSE are intertwined in a symbiotic relationship: HPC technology enables breakthroughs in CSE research, and leading-edge CSE applications are the main drivers for the evolution of supercomputer systems (see also [32, 38, 39, 33, 30]).

3.2 Ubiquitous Parallelism: A Phase Change for CSE Research and Education

Parallelism for CSE is fundamental for extreme-scale computing, but the significance of parallelism goes much beyond the topics arising from supercomputing. All modern computer architectures are parallel, even those of moderate-size systems or on the desktop. Since single-processor clock speeds have stagnated, any further increase of computational power can be achieved only by a further increase in parallelism. High-performance architectures will involve an ever increasing number of parallel threads, possibly reaching a billion by 2020.

Significantly, future mainstream computers for science and engineering will not be accelerated versions of current architectures but smaller versions of today’s (and tomorrow’s) extreme-scale machines. In particular, they will inherit the node and core architecture from these systems; and thus the programming methodology must be adapted for these systems, just as it must be adapted for the extreme scale. Consequently, parallel computing in its full breadth has become a central and critical issue for CSE research. The inevitable trend to parallel computing
is complicated by the hierarchical and heterogeneous architecture of modern parallel systems. Although computer science research is making progress in developing techniques to make these architectural features transparent to the application developer, we have reached a state where, for high-end applications, the specifics of an architecture must be explicitly exploited in the algorithm design and the software development. Table 1 provides a snapshot of features in current architectures that may be relevant for designing efficient CSE solutions and hence should be addressed in a CSE curriculum.

Current technological trends indicate that the degree of parallelism will increase rapidly in the future. This situation is already reflected in the emergence of multicore architectures, including the continued success of inexpensive accelerator devices. These trends are increasing, moving parallel computing into the CSE mainstream. Therefore, current and future CSE research will have to address parallel computing in much more detail than in the past. Inevitably, all CSE methods and algorithms will have to be considered in the context of such parallel settings.

### 3.3 Extending the Scope of CSE through HPC Technology

**Real-time and embedded supercomputing.** Low-cost (or moderate-cost) computational power, becoming available through accelerator hardware (such as GPUs), will increasingly enable nontraditional uses of HPC-enabled CSE technology. Figure 9 illustrates a selection of possible future development paths, many of which involve advanced interactive computational steering applications and/or real-time simulation.

Opportunities thus arise to use simulation in the development loop, for example, when a real-time-capable virtual model of a power plant is used to design, develop, and test the respective control systems. Once simulation software can be used in real time, it can be used for training and education. The classical application is the flight simulator, but the methodology can be extended analogously to all situations where humans operate complex technical artifacts and where systematic training on a simulator may save time and money. In safety-critical applications, real-time simulators permit the training of personnel for situations that the real systems will (one hopes) never experience. Further uses of fast, embedded CSE systems include the development of simulators for modeling of predictive control systems (which base automatic control and optimization on realistic physical descriptions of the devices and processes) and for patient-specific biomedical diagnosis (where CSE-based evaluation of patient specific data may support diagnosis and therapy planning). CSE may now be extended beyond its traditional role...
as a research tool, whose direct relevance is limited to researchers and engineers designing new products, and into a vast array of new application domains.

**Mesoscopic-scale modeling enabled by extreme-scale computing.** Extreme-scale computing will enable mesoscale simulation, for example, to model the collection of cells that make up a human organ or the collection of particles that form a pile of sand directly, without resorting to averaging approaches. The potential can be seen when realizing that human has around $10^{11}$ neurons and $10^{13}$ red blood cells and a pile of sand may have $10^{10}$ grains. These numbers are halfway between the atomic scale of Angstroms to the human scale of meters. Direct simulation at the mesoscale requires dealing with large ensembles of objects, a task that will increasingly become feasible with more powerful computers. Note that with $10^{18}$ Flops/s, an extreme-scale computer can still execute $O(10^5)$ Flops/s for each human blood cell per second. Thus, extreme-scale computation may open the route to new modeling techniques and research directions that are out of reach on conventional computer systems.

In order to exploit this power, however, new methods must be devised, new algorithms invented, and new modeling paradigms formulated. New techniques for validation and verification are needed. Fascinating opportunities in fundamental research arise that go far beyond extending conventional continuum models and simply increasing the mesh resolution.

**Data science and HPC.** The paradigm of scalable algorithms and implementations that is central to HPC and CSE is also relevant to emerging trends in big data analytics and data science. Big data analytics is quickly moving in the direction of mathematically more sophisticated analysis algorithms and parallel implementations. CSE technology has the potential to become the foundation of sophisticated model-based approaches to data analytics. These in turn hold the promise of extracting valuable insight from the data that goes beyond what can be recovered by statistical modeling alone. Furthermore, the CSE tradition of developing and implementing efficient parallel algorithms for compute-intensive and data-intensive problems will contribute significantly to modern data science.

HPC supercomputers and cloud data centers serve different needs and are optimized for applications that have fundamentally different characteristics. Nevertheless, they face challenges that have many commonalities in terms of extreme scalability, fault tolerance, cost of data movement, and power management. Extensive potential exists for cross-fertilization of ideas and approaches between extreme-scale HPC and large-scale computing for data analysis. Economy-of-scale pressures will contribute to a convergence of technologies for computing at large scale. For these reasons, HPC and CSE education and research must foster synergies with big data analytics and data science.

### 3.4 Emergent Topics in HPC-Related Research and Education

We currently observe a broad and comprehensive infiltration of HPC technology into mainstream CSE. Parallel computing and HPC competence have become essential for a wide class of current and future CSE uses, not only those at the high end. In particular, many of the new CSE development areas illustrated in Figure 9 will not be accessible without more investment in HPC and parallel computing research and education.

**Toward a quantitative analysis of efficient methods and algorithms.** The advent of exascale and other performance-critical applications requires that CSE research and education increasingly address the performance abyss that widens between traditional mathematical theory and the practical use of modern computer systems. For example, cost metrics that are based on counting floating-point operations increasingly fail to correlate with the truly relevant cost factors, such as time to solution or energy consumption. Research will be necessary in order to quantify much more complex algorithmic characteristics, such as memory footprint and memory access structure (i.e., cache reuse, uniformity of access, utilization of block transfers, etc.), processor utilization, communication, and synchronization requirements. These effects must be built into better cost and complexity models—models that are simple enough that they can be used but that capture the true nature of all computational costs much better than just counting the flops.

Just as fundamental as the changes caused by the shifts in technology, we note that the classical approach to convergence theory in traditional-style numerical mathematics exhibits an inherent fundamental deficit when used to assess computational cost. Conventional analysis often provides an insufficient basis to quantify the efficiency of algorithms and software, since the theorems often leave the constants unspecified. Although by themselves rigorously proven, such results of mathematics permit only heuristic—and often misleading—predictions of real
computational performance. While frequently mathematically elegant, such analysis is fundamentally limited as a guiding principle for the design of efficient computational algorithms and simulation methods. In the context of CSE, new interdisciplinary methods to design algorithms and software must be developed. Here CSE as a science requires that the traditions of numerical analysis be fundamentally expanded whenever the existing theoretical results are found inadequate to design and select efficient parallel algorithms. As the cost of computing increases for many high-end applications, systematic performance engineering for CSE applications will require new forms of analysis, including the development of systematic benchmarking and heuristic algorithm engineering methodologies.

**Ultrascalability and asynchronous algorithms.** Since, for the foreseeable future, all additional compute power will be delivered through increased parallelism, we must expect that high-end applications will reach degrees of parallelism of up to $10^9$ within a decade. This situation poses a formidable challenge to the design and implementation of algorithms and software. Traditional paradigms of bulk-synchronous operation are likely to face significant performance obstacles. In particular, it cannot be acceptable for a single slow processor to determine the execution time for a given phase of the computation and thus force millions of sibling processors to remain idle. Many algorithms permit variants that are more asynchronous, enabling processing to continue even if a small number of processors stay behind; but this is a wide open area of research, since it requires a new look at data dependencies and possibly also nondeterministic execution schedules. Additionally, the available systems software must be extended to permit the efficient and robust implementation of such asynchronous algorithms.

**Power wall.** Currently, moving a word of data in a large-scale system requires approximately 10 Njoule ($= 10^{-8}$ Joule) of energy, a value that is expected to improve by a modest factor within the next decade [13]. In the following we assume that a reduction to just 1 Njoule can be achieved for each elementary data transfer. Thus, transferring $N \times N$ data items for an all-to-all interaction between $N$ inputs to create $N$ outputs (as, e.g., in a conventional matrix-vector multiplication) will cause an energy dissipation of 277 kWh when $N = 10^9$. This figure disregards any computation; it is just the energy consumption for the naively implemented $N$ by $N$ data communication. Thus, treating gigascale problems with $N = 10^9$ will become very expensive, but the power consumption for a single such algorithmic step may still be considered manageable. This situation changes drastically, however, when we transition to terascale problems with $N = 10^{12}$. The same all-to-all data exchange will now dissipate 277 GWh, which is equivalent to the energy output of a medium-size nuclear fusion explosion. Clearly on exascale systems where we may want to operate on terascale data, even $O(N^2)$ algorithms are no longer feasible.

The same conclusion also holds for “smaller” systems. For example, returning to $N = 10^9$, if we assume that petaflop computers are widely available by the time our energy to move each word of data is down to 1 Njoule, then in theory we can execute our $N^2$ operation in 1000 seconds. But this will still be at a cost of 277 kWh, thus requiring over 1 MW of sustained power just for the data movement. Clearly such power levels are neither feasible nor affordable in a standard office environment. Consequently, only hierarchical algorithms whose complexity is significantly below $O(N^2)$ can be used. This situation must be addressed in research and education.

In the context of CSE research, the power wall is thus primarily a question of designing the most efficient (in terms of operations and data movement) algorithms, but of course the technology development is also challenged to develop more energy-efficient systems as quickly as possible.

**Fault tolerance and resilience.** With increasing numbers of functional units and cores and with continued miniaturization, the potential for hardware failures rises. Fault tolerance on the level of a system can be reached only by redundancy, which drives energy and investment cost. At this time many algorithms used in CSE are believed to have a good potential for a so-called algorithm-based fault tolerance. That is, the algorithm either is tolerant against faults (e.g., still converges to the correct answer, but perhaps a little more slowly) or can be extended or augmented to compensate for different types of failure (by exploiting specific features of the data structures and the algorithms, for example). At present, many open research questions arise from these considerations, especially when the systems, algorithms, and applications are studied in combination.

**Performance engineering and co-design.** Currently, the design and implementation of CSE software are essentially detached from research into systematic performance analysis and the modeling of computer performance. Modeling is restricted mostly to the system level and is typically performed for existing applications, rather than as a systematic tool for designing, developing, and implementing CSE applications. Consequently a new potential
exists to make performance considerations a primary design objective for CSE software since using systematic approaches to performance prediction and evaluation, as a software design methodology, is currently not developed. This is related to the often-inadequate mathematical analysis of computational algorithms that does not permit rigorous a priori predictions of performance. Bringing these aspects together—by developing models, algorithms, and software, and by including performance analysis for the target platforms is sometimes called co-design, a term that we use here in a holistic sense. We believe this to be an essential paradigm for developing next-generation CSE tools.

**Programmability of heterogeneous architectures and sustainability of CSE software.** All modern supercomputers are hierarchically structured, as summarized in Table 1. This structure, in turn, creates the need to program a supercomputer with the hierarchy and architecture in mind, often using a hybrid combination of different languages and tools. For example, a given application may utilize MPI on the system level, OpenMP on the node level, and special libraries or low-level intrinsics to exploit core-level vectorization. Newer techniques from computer science, such as automatic program generation and domain-specific languages, may eventually help reduce the gap between real-life hardware structures and model complexity.

This complexity will need to be managed, and to some extent alleviated, in the future. For example, the development of new and improved unifying languages, combined with the tools to select appropriate algorithms for target architectures and to implement these algorithms automatically, may ease the burden on CSE software developers. Such tools are topics of current research and are therefore far from reaching the level of maturity required to support large-scale development. Consequently, CSE developers must currently rely on an approach that combines hardware-optimized libraries and compose their application built upon them, or they must master the complexity—typically in larger teams where members can specialize—by undertaking explicitly hardware-aware development. This task is made even more complex when accelerators, such as GPUs, are to be used.

Designing and developing CSE software to be sustainable are challenging software engineering tasks, not only in the extreme scale, but also in conventional applications that run on standard hardware. Classical decomposition and structuring techniques often lead to intolerable performance penalties, since the natural numerical objects are too lightweight to be efficiently manageable. Thus the best software architecture is often determined by performance considerations, and it is a high art to identify kernel routines that can be used an internal interface for a software performance layer that can be optimized for various architectures.

### 4 CSE and the Data Revolution: The Synergy between Computational Science and Data Science

The world is experiencing an explosion of data. Indeed, since 2003, new data has been growing at an annual rate that exceeds the data contained in all previously created documents. The coming of extreme-scale computing and data acquisition from high-bandwidth experiments across the sciences is creating a phase change. The rapid development of networks of sensors and the increasing reach of the Internet and other digital networks in our connected society create entirely new data-centric analysis applications in broad areas of science, commerce, and technology [3, 29, 16]. These massive amounts of data offer tremendous potential for generating new knowledge, not only in the natural sciences and engineering, where they enable new approaches such as data-driven scientific discovery and data-enabled uncertainty quantification, but also in almost all areas of human activity. For example, biology and medicine have increasingly become quantitative sciences over the past two or three decades, aided by the generation of large data sets. The advent of massive data is also starting to change the social sciences, which are rapidly becoming more quantitative.

#### 4.1 Relation of CSE to Big Data Analytics and Data Science

Big data is transforming the fabric of society, in areas that go beyond research in the physical sciences and engineering [16]. Big data analytics aims at extracting information from massive amounts of data in areas as diverse as business intelligence, cybersecurity, social network recommendation, and government policy. Analysis of the data is often based on statistical and machine learning methods from data science. Similar to CSE, data science
combines aspects of mathematics and statistics, computer science, and domain knowledge; and hence it possesses an important synergy with CSE.

Similarly, in core application areas of CSE, our ability to produce data is rapidly outstripping our ability to use it. With exascale data sets, we will be creating far more data than we can explore in a lifetime with current tools. Yet, exploring these data sets is the essence of new paradigms of scientific discovery. Thus, one of the greatest challenges is to create new theories, techniques, and software that can be used to effectively understand and make use of this rapidly growing data and to make new discoveries and advances in science and engineering.

In order to meet this challenge, a broad range of new activities has resulted that significantly extend CSE methodology, for example, incorporating advanced statistical techniques, model reduction approaches, and network-based analysis.

CSE is expected to play an important role in developing the next generation of parallel high-performance data analytics approaches that employ descriptions of the data based on physical or phenomenological models informed by first principles, with the promise of extracting valuable insight from the data that crucially goes beyond what can be recovered by statistical modeling alone. In classical science, observational data is exploited by scientists to derive theoretical models (in a process of induction) or to confirm postulated theoretical models (in the deductive approach). Modern computational techniques open the possibility of using data in many more powerful ways. Important examples where massive data are interpreted through computational models are data assimilation for weather forecasting and model-predictive control in engineering applications. Enormous potential lies, for example, in the emerging model-based interpretation of patient-specific data from medical imaging for diagnosis and therapy planning.

The analysis of big data requires efficient and scalable mathematics-based algorithms executed on high-end computing infrastructure, which are core CSE competencies that translate directly to big data applications. CSE education and research must foster the important synergies with big data analytics and data science that are apparent in a variety of emerging application areas.

4.2 Role of Massive Data in CSE

In the core CSE research areas of the physical sciences and engineering [29], important new CSE techniques and approaches are emerging to exploit big data in conjunction with existing modeling techniques. For example, the CSE focus area of uncertainty quantification aims at characterizing the uncertainties inherent in the use of CSE models and data. To this end, new methods are being developed building on statistical techniques that include Markov chain Monte Carlo methods, Bayesian inference, maximum likelihood, expectation maximization, hidden Markov processes, and Markov decision processes. While these underlying techniques have broad applications in many areas of data science, CSE efforts tend to have a special focus on developing efficient computational techniques at scale, with potential for broad applicability in other areas of big data analytics and data science. Massive data is also a crucial component in other CSE focus areas, such as validation and verification, reduced-order modeling, data assimilation, optimal control, data-driven scientific discovery, and analysis of graphs and networks. CSE techniques to address the challenges of working with massive data sets include large-scale optimization and linear and nonlinear solvers, inverse problems, stochastic methods, scalable techniques for scientific visualization, and high-performance parallel implementation.

Exploiting massive amounts of data is having a profound influence in many areas of CSE applications. The following paragraphs describe some striking examples.

Data-driven scientific discovery is revolutionizing the related fields of chemistry and materials science, in a transformation that is illustrative of those sweeping all of science, leading to successful transition of basic science into practical tools for applied research and early engineering design. Chemistry and materials science are both mature computational disciplines that through advances in theory, algorithms, and computer technology are now capable of increasingly accurate predictions of the physical, chemical, and electronic properties of materials and systems. The equations of quantum mechanics (including Schrödinger’s, Dirac’s, and density functional representations) describe the electronic structure of solids and molecules that controls many properties of interest, and statistical mechanics must be employed to incorporate the effects of finite temperature and entropy. These are forward methods—given a chemical composition and approximate structure, one can determine a nearby stable structure and compute its properties. To design new materials or chemical systems, however, one must solve the inverse problem—what is the system that has specific or optimal properties? And, of course, the system must
be readily synthesized, inexpensive, and thermally and chemically stable under expected operating conditions. Breakthrough progress has recently been made in developing effective constrained search and optimization algorithms for precisely this purpose \[41, 8, 12\], with this process recognized in large funding initiatives such as the multiagency U.S. Materials Genome Initiative \[11\]. This success has radically changed the nature of computation in the field. Less than ten years ago most computations were generated and analyzed by a human, whereas now 99.9% of computations are machine generated and processed as part of automated searches that are generating vast databases with results of millions of calculations to correlate structure and function \[25, 28\]. In addition to opening important new challenges in robust and reliable computation, the tools and workflows of big data are now crucial to further progress.

Computation and big data also meet in **characterization of physical material samples** using techniques such as X-ray diffraction and adsorption, neutron scattering, pychography, transmission electron, and atomic microscopes. Only for essentially perfect crystals or simple systems can one directly invert the experimental data and determine the structure from measurements. Most real systems, typically with nanoscale features and no long range order, are highly underdetermined \[5\]. Reliable structure determination requires fusion of multiple experimental data sources (now reaching multiple terabytes in size) and computational approaches. Computation provides a forward simulation (e.g., given a structure, what spectrum or diffraction pattern results), and techniques from uncertainty quantification are among those proving successful in making progress.

![Figure 10: Visualization of a topological analysis and volume rendering of one time step in a large-scale, multi-terabyte combustion simulation. The topological analysis identifies important physical features (ignition and extinction events) within the simulation, while the volume rendering allows viewing the features within the spatial context of the combustion simulation. Simulation by Jackie Chen, Sandia National Laboratories; Visualization by the Scientific Computing and Imaging Institute.](image)

**In scientific visualization**, new techniques are being developed to give visual insight in the deluge of data that is transforming scientific research. For example, Figure 10 displays novel topological analysis techniques that allow combustion researchers to automatically identify important physical features in multi-terabyte simulation results. Data analysis and visualization are key technologies for enabling future advances in simulation and data-intensive based science, as well as in several domains beyond the sciences. Specific big data visual analysis challenges and opportunities include in-situ interactive analysis; user-driven data reduction; scalable and multilevel hierarchical algorithms; representing evidence and uncertainty; heterogeneous data fusion; data summarization and triage for interactive queries; and analysis of temporally evolved features \[18, 19, 40\].

**In high-energy particle physics**, the Large Hadron Collider at the European Organization for Nuclear Research (CERN) is producing more than 50 petabytes of experimental data annually. Breakthrough scientific re-
sults, such as the discovery of the Higgs boson, critically rely on analysis and interpretation of this massive data using advanced statistical and computational tools to quantify the agreement between observation and theoretical models.

Many geoscience systems are characterized by complex, large-scale behavior that couples multiple physical, chemical, and/or biological processes over a wide range of length and time scales. Examples include earthquake rupture dynamics, climate change, multiphase reactive subsurface flows, long term crustal deformation, severe weather, and mantle convection. The uncertainties prevalent in the mathematical and computational models characterizing these systems have made high-reliability predictive modeling a challenge. However, the geosciences are at the cusp of a transformation from a largely descriptive to a largely predictive science. This is driven by continuing trends: the rapid expansion of our ability to instrument and observe the Earth system at high resolution; sustained improvements in computational models and algorithms for complex geoscience systems; and the tremendous growth in computing power.

Three basic stages are involved in exploiting the confluence of big data, advanced models and algorithms, and high-performance computing to facilitate model-based decision-making with quantified uncertainties. First, large observational data sets can be integrated into complex geoscience models to quantify and reduce the uncertainties in unknown model parameters (e.g., initial conditions, boundary conditions, coefficients, sources). This is the statistical inverse problem. Second, once the model parameters and their uncertainty have been estimated from the data via statistical inversion, the resulting stochastic model can be used to make predictions by propagating probability distributions of uncertain inputs through the model to yield predictions with quantified uncertainties. This is the stochastic forward problem. Third, given stochastic predictions of outputs of interest and decision variables (design or control) that can be manipulated to influence the outputs, a stochastic optimization problem can be solved to produce optimal decisions. One example of this pipeline is a subsurface contaminant mitigation problem: the model is given by PDEs describing flow and transport through a porous medium; the data are well measurements of pressure and contaminant concentration; the uncertain parameters are the permeability and initial contaminant concentration fields; the decision variables are the water injection and extraction rates at control wells; and the decision objective is the contaminant concentration at water supply wells.

Unfortunately, the pipeline for inverse, forward, and optimization problems under uncertainty is prohibitive using current methods for large-scale, complex geoscience models characterized by high-dimensional parameter spaces. However, advances in large-scale UQ algorithms in recent years [1] are beginning to make feasible the use of Bayesian inversion to infer parameters and their uncertainty in large-scale complex geoscience systems from large-scale satellite observational data. Two examples are global ocean modeling [20] and continental ice sheet modeling [15]. Continued advances in UQ algorithms, Earth observational systems, computational modeling, and HPC systems over the coming decades will lead to successful instantiations of the data-driven predictive modeling and decision-making pipeline described above to an expanding set of complex geoscience systems models. This in turn will lead to a better understanding of Earth dynamics as well as improved tools for simulation-based decision-making for critical Earth systems.

5 CSE Education and Workforce Development

With all these opportunities on the horizon for the CSE field, there is a growing demand for CSE graduates and a need to expand CSE educational offerings. This need includes CSE programs at both the undergraduate and graduate levels, as well as continuing education and professional development programs. In addition, the increased presence of digital educational technologies provides an exciting opportunity to rethink CSE pedagogy and modes of educational delivery.

5.1 Growing Demand for CSE Graduates in Industry, National Labs, and Academic Research

Industry, national laboratories, government, and broad areas of academic research are making more use of simulations, high-end computing, and simulation-based decision making than ever before. This trend is apparent broadly across domains—for example, energy, manufacturing, finance, and transportation are all areas in which CSE is playing an increasingly significant role, with many more examples across science, engineering, business,
and government. Research and innovation, both in academia and in the private sector, are increasingly driven by large-scale computational approaches. A National Council on Competitiveness report points out that high-end computing plays a “vital role in driving private-sector competitiveness … all businesses that adopt HPC consider it indispensable for their ability to compete and survive” [9]. With this significant and increased use comes a demand for a workforce versed in technologies necessary for effective and efficient mathematics-based computational modeling and simulation. There is high demand for graduates with the interdisciplinary expertise needed to develop and/or utilize computational techniques and methods in order to advance the understanding of physical phenomena in a particular scientific, engineering, or business field and to support better decision making.

As stated in a recent report on workforce development by the US Department of Energy (DOE) Advanced Scientific Computing Advisory Committee [37], “All large DOE national laboratories face workforce recruitment and retention challenges in the fields within Computing Sciences that are relevant to their mission. … There is a growing national demand for graduates in Advanced Scientific Computing Research-related Computing Sciences that far exceeds the supply from academic institutions. Future projections indicate an increasing workforce gap.” This finding was based on a number of reports, including one from the High End Computing Interagency Working Group [14] stating: “High end computing (HEC) plays an important role in the development and advanced capabilities of many of the products, services, and technologies that are part of our everyday life. The impact of HEC on the agencies of the federal government, the quality of academic research, and on industrial competitiveness is substantial and well-documented. However, adoption of HEC is not uniform, and to fully realize its potential benefits we must address one of the most often cited barriers: lack of HEC skills in the workforce.”

In order to take advantage of the transformation that high-performance and data-centric computing offer to industry, the critical factor is a workforce versed in CSE and capable of developing the algorithms, exploiting the compute platforms, and designing the analytics that turns data with its associated information into knowledge to act. The tasks to exploit the emerging tools require the critical thinking and the interdisciplinary background that is prevalent in CSE training [29]. The CSE practitioner has both the expertise to apply computing tools and the analytical skills to tease out the problems that often are encountered when commercial enterprises seek to design new products, develop new services, and create novel approaches from the wealth of data available. The CSE practitioner knows how to use computational tools and analytics in uncharted areas, often applying previous domain-specific understanding to these new areas. The CSE practitioner, while often a member of a team of others from varying disciplines, is the catalyst driving the change that industry seeks in order not only to remain competitive but also to be first to market, providing the necessary advantage to thrive in a rapidly evolving technological ecosystem.

5.2 Future Landscape of CSE Educational Programs

CSE educational programs are needed in order to create young professionals who satisfy this growing demand and who support the growing CSE research field. These include CSE programs at both the undergraduate and graduate levels, as well as continuing education and professional development programs. These also include programs that are “CSE focused” and those that follow more of a “CSE infusion” model. The former includes programs that have CSE as their primary focus (e.g., a B.S., M.S., or Ph.D. in computational science and engineering), while the latter includes programs that infuse some CSE training within another degree structure (e.g., a minor, emphasis, or concentration in CSE complementing a major in mathematics, science or engineering or a degree in a specific computational discipline such as computational finance or computational geosciences).

At the undergraduate level, the breadth and depth of topics covered will depend on the specific degree focus. However, the following high-level topics are important content for an undergraduate program:

1. Foundations in mathematics and statistics, including calculus, linear algebra, differential equations, applied probability, and discrete mathematics.
2. Simulation and modeling, including conceptual models, accuracy, use of modeling tools, assessment of computational models, data-based models, and physics-based models.
3. Computational methods and numerical analysis, including errors, nonlinear equations, solution of systems of linear equations, interpolation, curve fitting, optimization, Monte Carlo, numerical methods for ODEs, and numerical methods for PDEs.
4. Computing skills, including compiled high-level languages, algorithms (numerical and nonnumerical), elementary data structures, analysis, parallel programming, scientific visualization, and awareness of computational complexity and cost.

At the graduate level, again the breadth and depth of topics covered will depend on the specific degree focus. In the next section, we make specific recommendations in terms of a set of learning outcomes desired for a CSE graduate program. We also note the growing importance of and demand for terminal master’s degrees, which can play a large role in fulfilling the industry and national laboratory demand for graduates with advanced CSE skills.

All CSE graduates should possess the attributes of having a solid foundation in modern mathematics; an understanding of probability and statistics; a grasp of modern computing, computer science, programming languages, principles of software engineering, and high-performance computing; and an understanding of foundations of modern science and engineering, including biology. These foundations should be complemented by deep knowledge in a specific area of science, engineering, mathematics and statistics, or computer science. CSE graduates should also possess skills in teamwork, multidisciplinary collaboration, and leadership. A valuable community project would be to collect resources to assist early-career researchers in advancing skills to support CSE collaboration and leadership.

A third area of educational programs is that of continuing and professional education. Opportunities exist for SIAM or other institutions to engage with industry to create and offer short courses, including those that target general CSE skills for the non-CSE specialist as well as those that target more advanced skills in timely opportunity areas (such as parallel and extreme-scale computing, and computing with massive data). Often one assumes that much of the workforce for industry in CSE will come at the postgraduate level; increasingly, however, industry needs people who have an understanding of CSE even at the undergraduate level in order to realize the full potential growth in a rapidly expanding technological workplace. Future managers and leaders in business and industry must be able to appreciate requirements to practice CSE and the benefits that accrue from CSE. Continuing education can play a role in fulfilling this need. The demand for training in CSE-related topics exists more broadly among graduate students and researchers in academic institutions and national laboratories, as evidenced by the growing number of summer schools worldwide, as well as short courses aimed at the research community. For example, the Argonne Training Program for Extreme-Scale Computing [26] covers key topics that CSE researchers must master in order to develop and use leading-edge applications on extreme-scale computers. The program targets early-career researchers to fill a gap in the training that most computational scientists receive and provides a more comprehensive program than do typical short courses. The recent creation of the SIAM Activity Group on Applied Mathematics Education represents another opportunity for collaboration to pursue some of these ideas in continuing and professional education.

5.3 Graduate Program Learning Outcomes

A learning outcome is defined as what a student is expected to be able to do as a result of a learning activity. In this section, we describe a set of learning outcomes desired of a student graduating from a CSE Ph.D. program. We focus on outcomes because they describe the set of desirable competencies without attempting to prescribe any specific degree structure. These outcomes can be used as a guide to define a Ph.D. program that meets the needs of the modern CSE graduate; they can also play an important role in defining and distinguishing the CSE identity and in helping employers understand the skills and potential of CSE graduates.

In Table 5.3, we focus on the “CSE Core Researchers and Developers” category in Figure 5. We distinguish between a CSE Ph.D. with a broadly applicable CSE focus and a CSE Ph.D. with a domain-driven focus. An example of the former is a “Ph.D. in computational science and engineering” while an example of the latter is a “Ph.D. in computational geosciences.” The listed outcomes relate primarily to those CSE-specific competencies that will be acquired through classes. Of course, the full competencies of the Ph.D. graduate must also include the more general Ph.D.-level skills, such as engaging deeply in a research question, demonstrating awareness of research context and related work, and producing novel research contributions, many of which will be acquired through the doctoral dissertation. We also note that it would be desirable for graduates of a CSE master’s degree program to also achieve most (if not all) of the outcomes in Table 5.3. In particular, in educational systems where

---

1Videos and slides of lectures are available via the ATPESC website [http://extremecomputingtraining.anl.gov](http://extremecomputingtraining.anl.gov).
Table 2: Learning outcomes desired of a student graduating from a CSE PhD program. Italicized text denotes differences in learning outcomes for programs with a broadly applicable CSE focus (left) and a domain-driven focus in a particular field of science or engineering (right). Learning outcomes that are common to both types of PhD programs span left and right columns.

<table>
<thead>
<tr>
<th>CSE PhD with broadly applicable CSE focus</th>
<th>CSE PhD with domain-driven focus in field X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine mathematical modeling, physical principles, and data to derive, analyze, and assess a model across a range of systems (e.g., statistical mechanics, continuum mechanics, quantum mechanics, molecular biology).</td>
<td>Combine mathematical modeling, physical principles and data to derive, analyze, and assess a range of models within field X.</td>
</tr>
<tr>
<td>Demonstrate graduate-level depth in devising, analyzing, and evaluating new methods and algorithms for computational solution of mathematical models (including parallel, discrete, numerical, statistical approaches, and mathematical analysis).</td>
<td>Demonstrate mastery in code development to exploit parallel and distributed computing and other emerging modes of computation in algorithm implementation.</td>
</tr>
<tr>
<td>Demonstrate proficiency in code development to exploit parallel and distributed computing and other emerging modes of computation in algorithm implementation.</td>
<td></td>
</tr>
<tr>
<td>Select and apply techniques and tools from software engineering to build robust, reliable, and maintainable software.</td>
<td></td>
</tr>
<tr>
<td>Develop, select, and use tools and methods to represent and visualize computational results.</td>
<td></td>
</tr>
<tr>
<td>Critically analyze and evaluate results using mathematical and data analysis, physical reasoning, and algorithm analysis, and understand the implications on models, algorithms and implementations.</td>
<td></td>
</tr>
<tr>
<td>Identify the sources of errors in a CSE simulation (such as modeling errors, code bugs, premature termination of solvers, discretization errors, roundoff errors), and understand how to diagnose them and work to reduce or eliminate them.</td>
<td></td>
</tr>
<tr>
<td>Appreciate and explain the context of decision-making as the end use of many CSE simulations, and as appropriate be able to formulate, analyze, and solve CSE problems in control, design, optimization, or inverse problems.</td>
<td>Appreciate and explain the context of decision-making as the end use of many CSE simulations, and as appropriate be able to formulate, analyze and solve CSE problems in control, design, optimization or inverse problems as relevant to field X.</td>
</tr>
<tr>
<td>Understand data as a core asset in computational research and demonstrate appropriate proficiencies in processing, managing, mining, and analyzing data throughout the CSE/simulation loop.</td>
<td></td>
</tr>
<tr>
<td>Demonstrate the ability to develop, use, and analyze sophisticated computational algorithms in data science and engineering, and understand data science and engineering as a novel field of application of CSE.</td>
<td></td>
</tr>
<tr>
<td>Demonstrate graduate-level proficiency in one domain in science or engineering.</td>
<td>Demonstrate graduate-level depth in domain knowledge in field X.</td>
</tr>
<tr>
<td>Communicate across disciplines and collaborate in a team.</td>
<td></td>
</tr>
</tbody>
</table>
there is no substantial classwork component for the Ph.D., the learning outcomes of Table 5.3 would also apply to the master’s or honors degree that may precede the Ph.D.

In the next two subsections, we elaborate more on the interaction between CSE education and two areas that have seen considerable change since the design of many existing CSE programs: extreme-scale computing and computing with massive data.

5.4 Education in Parallel Computing and Extreme-Scale Computing

Extreme-scale computing poses new challenges to education, but there is also the broader and fundamental need to educate a wide spectrum of engineers and scientists to be better prepared for the age of ubiquitous parallelism, as addressed in Section 3; see also [32, 39]. Parallelism has become the basis for all computing technology, which necessitates a shift in teaching even the basic concepts. Simulation algorithms and their properties have been in the core of CSE education, but now we must emphasize parallel algorithms. The focus used to be on abstract notions of accuracy of methods and the complexity of algorithms; today it must be shifted to the complexity of parallel algorithms and the real-life cost to solve a computational problem, which is a completely different notion. Additionally, the asymptotic complexity and thus algorithmic scalability become more important when the machines grow larger. At the same time, the traditional complexity metrics increasingly fail to give guidance about which methods, algorithms, and implementations are truly efficient. As elaborated in Section 3, designing simulation software has become an extremely complex, multifaceted art. The education of future computational scientists must address these topics that arise from the disruptive technology that is dramatically changing the landscape of computing.

Today’s extreme scale is tomorrow’s desktop. An analogous statement holds for the size of the data that must be processed and that is generated through simulations. In education we need to distinguish between those whose research aims to simulate computationally demanding problems (see Section 3 and Figure 9) and the wider class of people who are less driven by performance considerations. For example, many computational engineering problems exist in which either the models are not extremely demanding computationally or in which model reduction techniques are used to create cheap models.

Education in programming techniques needs to be augmented with parallel programming elements and a distinctive awareness of performance and computational cost. Additionally the current trends are characterized by a growing complexity in the design of the computer architectures. They are hierarchical and heterogeneous, as illustrated in Table 1. These architectures are reflected by complex and evolving programming models that should be addressed in a modern CSE education. Programming such systems will typically require using different languages and programming paradigms explicitly, having to interface them with the heterogeneous architecture on the one side and the algorithmic requirements on the other side, leading to a complex, interdependent software architecture that is not easy to make flexible, efficient, and highly performant. Overall this goal can be achieved only through a co-design process where all aspects and design decisions are balanced and weighed against each other. Graduates of CSE programs should be aware of these issues and should be capable of collaborating with specialists in these topics.

In defining the educational needs in parallel and high performance computing for CSE, we must distinguish between different intensities. Any broad education in CSE will benefit from an understanding of parallel computing, simply because sequential computers have ceased to exist. All students must be trained to understand concepts such as concurrency, algorithmic complexity, and its relation to scalability, elementary performance metrics, and systematic benchmarking methodologies.

In more demanding applications, parallel computing expertise and performance awareness are necessary and must go significantly beyond the content of most current curricula. This requirement is equally true in those applications that may be only of moderate scale but that nevertheless have high-performance requirements, such as those in real-time applications or that require interactivity; see Figure 9. Here, CSE education must include a fundamental understanding of computer architectures and the programming models that are necessary to exploit these architectures.

Besides classification according to scientific content and HPC intensity, educational structures in CSE must also address the wide spectrum of the CSE community that was described and analyzed in Section 1.5 (see also Figure 5).
CSE Domain Scientists and Engineers – Method Users. Users of CSE technology will typically use dedicated supercomputer systems and specific software on these computers; they will usually not program HPC systems from scratch. Nevertheless, they need to understand the systems and the software they use, in order to achieve leading-edge scientific results. If needed, they must be capable to extend the existing applications, if necessary in collaboration with CSE and HPC specialists.

An appropriate educational program for CSE Users in HPC can be organized in courses and tutorials on specific topics such as are regularly offered by computing centers and other institutions. These courses are often taught in compact format (ranging from a few hours to a week) and are aimed at enabling participants to use specific methods and software or specific systems and tools. They naturally reach only limited depth, but a wide spectrum of such courses is essential in order to widen the scope of CSE and HPC technology and to bring it to bear fruit as widely as possible.

CSE Domain Scientists and Engineers – Method Developers. These are often domain scientists or engineers who have specialized in using computational techniques in their original field. They often have decades of experience in computing and using HPC, and thus, historically, they are mostly self-taught. Regarding the next generation of scientists, students of the classical fields (such as physics, chemistry, or engineering) will increasingly want to put stronger emphasis on computing within their fields.

The more fundamental knowledge that will be needed to competently use the next generation of HPC systems thus can then not be adequately addressed by compact courses as described above. A better integration of these topics into the university curriculum is necessary, by teaching the use of computational methods as part of existing courses or by offering dedicated HPC- and simulation-oriented courses (as electives) in the curriculum. An emphasis on CSE and HPC within a classical discipline may be taught in the form of a selection of courses that are offered as electives by CSE or HPC specialists, or—potentially especially attractive—by co-teaching of a CSE specialist jointly with a domain scientist.

CSE Core Researchers and Developers. Scientists who work at the core of CSE are classified in two groups according to Figure 5. Domain-driven CSE students as well as those focusing on broadly applicable methods should be expected to spend a significant amount of time learning about HPC and parallel computing topics. These elements must be well integrated into the CSE curriculum. Core courses from computer science (such as parallel programming, software engineering, and computer architecture) may present the knowledge that is needed also in CSE, and they can be integrated into a CSE curriculum. Often, however, dedicated courses that are especially designed for students in CSE will be significantly more effective, since they can be adapted to the special prerequisites of the student group and can better focus on the issues that are relevant for CSE. Often again co-teaching such courses, labs, or projects may be fruitful, especially when such courses cover several stages of the CSE pipeline (see Section 1.4).

These three levels of CSE education are naturally interdependent, but we emphasize that all three levels are relevant and important. In particular, the problem of educating the future generation of scientists in the competent use of computational techniques cannot be addressed solely by offering one-day courses on how to use the latest machine in the computing center.

5.5 CSE Education in Uncertainty Quantification and Big Data

The rising importance of massive data sets in application areas of science and engineering and beyond has broadened the skillset that CSE graduates may require. For example, data-driven uncertainty quantification requires statistical approaches that may include tools such as Markov chain Monte Carlo methods and Bayesian inference. Analysis of large networks requires skills in discrete mathematics, graph theory, and combinatorial scientific computing. Similarly, many data-intensive problems require approaches from inverse problems, large-scale optimization, machine learning, and data stream and randomized algorithms.

The broad synergies between computational science and data science offer opportunities for educational programs. Many CSE competencies translate directly to the analysis of massive data sets at scale using high-end computing infrastructure. Computational science and data science are both rooted in solid foundations of mathematics and statistics, computer science, and domain knowledge, and this common core may be exploited in educational programs that can prepare the computational and data scientist of the future.
5.6 Changing Educational Infrastructure

As we think about CSE educational programs, we must also consider the changing external context of education, particularly with regard to the advent of digital educational technologies and their associated impact on the delivery of education programs.

One clear impact is an increased presence of online digital materials, including digital textbooks, open educational resources, and massive open online courses (MOOCs). Recent years have already seen the development of online digital CSE resources, as well as widespread availability of material in fields relevant to CSE, such as HPC, machine learning, and mathematical methods. An opportunity exists to make better community use of current materials, as well as to create new materials. There is also an opportunity to leverage other resources, such as CSGF essay contest winners (https://www.krellinst.org/csgf/outreach/cyse-contest) and archived SIAM plenaries and other high-profile lectures. It would be timely to create a SIAM focus group to create and curate a central repository linking to CSE digital materials and to coordinate community development of new CSE online modules. This effort could also be coordinated with an effort to pursue opportunities in continuing education.

Digital educational technologies are also having an impact on the way residential courses are structured and offered. For example, many universities are taking advantage of digital technologies and blended learning models to create “flipped classrooms,” where students watch video lectures or read interactive online lecture notes individually, and then spend their face-to-face class time engaged in active learning activities and problem solving. Digital technologies are also offering opportunities to unbundle a traditional educational model—introducing more flexibility and more modularity to degree structures. Many of these opportunities are well suited for tackling the challenges of building educational programs for the highly interdisciplinary field of CSE.

6 Recommendations and Conclusions

[Placeholder text:] We welcome all input on report recommendations/conclusions.
Acknowledgments

This report is an outcome of a workshop in August 2014 on *Future Directions in CSE Education and Research*, sponsored by the Society for Industrial and Applied Mathematics [35] and the European Exascale Software Initiative [32]. We gratefully acknowledge La Joyce Clark and Cheryl Zidel of Argonne National Laboratory for workshop support. We also thank Jenny Radeck of Technische Universität München for assistance with graphics and Gail Pieper of Argonne National Laboratory for editing this document. The first author gratefully acknowledges support by the Institute of Mathematical Sciences at the National University of Singapore during the work on this report.
REFERENCES

References


REFERENCES


