ABOUT THE SUBJECT

The rapid and widespread proliferation of computing devices and technologies is producing significant changes to life, as ever more established activities go online, and as networking makes new activities possible. These developments have already affected most sectors of society, including the nature of scientific activity. This book looks at the character and potential of advances in computing technologies for scientific practice. The impacts that computing technologies have already had on science indicate that there will be a much greater expansion and transformation of the part computing plays in scientific research, with significant effects on the nature of scientific practice itself and of the social and organisational settings within which science is practiced.

At this time we are witnessing a take off for CSE. In the terms recently popularised by Malcolm Gladwell, CSE is approaching a tipping point, “the moment of critical mass, the threshold, the boiling point” (Gladwell, 2000, p12) when the spread of a minority or marginal activity moves into epidemic expansion.

Computational Science and Engineering (CSE) is the name we use here for the connected technological and practical advances that are being made to researchers’ situation and practice. CSE subsumes activities variously identified as Computational Science, Scientific Computing, and eScience. It is developing apace across the natural sciences, and is slowly spreading into the humanities and social sciences and from the developed world to the developing world, but many of these developments are independent and in parallel. Though there are initiatives to give a coordinated push to CSE’s growth throughout the disciplines, these initiatives are not as yet unified or coherent.

CSE is understood as providing a new means for the delivering of substantial scientific discovery (and not just a means of processing scientific information), in addition to the existing pair of observation/experimentation and theory, especially through the use of simulations. Simulation can be used to explore complex systems that are (relatively) inaccessible to experiment or observation because of cost or technological difficulties.

Computational models commonly involve mathematical models that embody a theoretical understanding from appropriate scientific disciplines and where possible results are validated against observational/experimental data. With valid computational models virtual experiments can generate new findings. The relentlessly expanding power and diminishing size of computing hardware means that computational resources are not only ever more prevalent and powerful, but also more easily accessible, so that very complex scientific computation can be done on desktop machines as well as on the High Powered Computers (HPC) that are required for the most massive jobs.
MISSION AND OBJECTIVES OF THIS BOOK

This handbook indentifies the fundamental elements making up CSE and showing their interdependence in a way that (a) reviews the state of the art and current achievements; (b) explores imminent developments advancing the state of play; (c) makes these accessible to as wide an audience of interested parties as possible. The handbook gives an organised survey of cogent topics. It is placed at a level that does not assume considerable prior familiarity with or facility in the heavily technical details of specific areas of CSE. However, be warned that the chapters do cover complex research, though the authors have attempted to make their work as accessible as possible and to guide readers to sources of more advanced understandings and discuss practical issues.

The handbook’s emphasis acts as a counterweight to the dominant focus on purely technical skills besting the literature by covering some practical aspects of developing and operating CSE inquiries, considering these from the point of view of those who might want to use CSE on a small scale as well as those conducting large scale projects, making this a useful resource for all budgets, as well as for those who are involved in managing the integrity and security of a CSE operation.

ORGANISATION OF THE BOOK

This book consists of 24 chapters that cover various aspect of CSE research. There is much overlap between the topics; thus, it was particularly difficult to organise the chapters. Initially traditional themes were picked as possible sections within the book; however it soon became clear that for example the numerical methods could not stand on their own, because they are a fundamental element of each CSE application discussed (even the visualization applications). Instead the chapters have been divided into 3 sections that cover broad areas of CSE research, and the chapters from each section have been ordered to reflect particular themes. This should make it easier for readers to find related topics.

- **Section 1: Hardware and Software:** This section explores the individual and interconnected elements of hardware and software. The section moves from hardware, to programming languages, to software development tools, to software development practices, and ends by connecting that to emerging scientific practices in the creation of research software.
- **Section 2: Applications:** This section explores the emerging trends in particular application areas, covering numerical methods, visualization, application areas new to CSE and the application of new techniques to established areas of CSE. The section tends to move from numerically intense research areas through visualization (although visualization can be numerically intense) to case studies, but the overlap is considerable.
- **Section 3: Organizational and Practical Issues:** This section covers issues less involved in the details of particular research areas and related instead to organisational issues, policy making, technology transfer, and research practices. This area is seldom discussed in CSE literatures, but its importance is becoming increasingly apparent.
MAIN CONTENTS OF THE BOOK

Section 1: Hardware and Software

The first chapter, Chapter 1, *Hardware Trends and Implications for Programming Models*, reviews the current changes to computing technology, especially hardware, and the ways in which those changes will, in turn, require considerable alteration to programming languages and practices and also covers software development tools. Chapter 2, *Multi-threaded Architectures: Evolution, Costs, Opportunities*, looks at a significant change in hardware i.e., the creation of massively multithread architectures. It considers the innovations necessary to create these architectures and asks how these developments may disrupt users’ practices.

Chapter 3, *High-Performance Customizable Computing*, examines another possible solution to the problem of increasing the speed of computations through an architecture that can dynamically change, e.g. to (re)introduce accelerators. The creation of hybrid systems allows the programmer to redistribute different computing operations such as calculating and data moving so that they can be carried out simultaneously on one system on different kinds of processors. Two kinds of accelerators - general purpose GPU’s (Graphics Processing Units) and customisable Field Programmable Gate Arrays (FPGA) – are compared.

Chapter 4, *High-Performance Computing for Theoretical Study of Nanoscale and Molecular Interconnects*, addresses one important development affecting hardware. Progress in nanotechnology means that not only are the design of semiconductor elements at a nanoscale possible, but also that the integration between hardware and wetware – the hybrid combination of biological and nanoelectric systems – is possible too. The difficulty and complexity of experiments at nanoscale presses researchers toward the use of simulations because experimentation is no longer feasible. Not only does this chapter review possible new hardware for HPC but it also does this in a novel way, using CSE, computational chemistry as an example, and so contributes a good case study of the CSE methodology.

Software development has been the Cinderella in CSE as it is a necessary but often unfunded element in research. Software development for scientific research purposes has commonly been for immediate research use only, with the emphasis on software research utility rather than achieving re-useable quality. Just as HPC offers much greater possibility of data re-use, so too does CSE invite the creation of re-usable software. Within the world of programming, a great deal has been learned about improving software quality, but much of this remains unknown to those developing scientific software (they are scientists, not software engineers). Chapter 5, *Effective Open-Source Performance Analysis Tools*, reviews open source tools (which are variable in quality) and will be of particular benefit to new adopters of these tools. Open source development has considerable appeal (not least because its tools are freely available to anyone at no cost), but there are still questions about how effective and dependable a method open source development is in a variety of environments such as HPC.

Chapter 6, *Pragmatic Software Engineering for Computational Science*, is designed to draw attention to quality processes for software development. This is done with an awareness of the risk of using top-heavy design requirements and attempts to avoid it by communicating the range of software development resources available. It aims to enable scientific software developers to select tools relevant to their purposes. Chapter 7, *A Framework for Testing Code in Computational Applications*, follows up on the previous chapter by focussing on one element of software development, the testing of the code.
Though code is being developed for scientific purposes, this does not eliminate the need to ensure that it works effectively as code. Similarly to its predecessor, this chapter recognises the difference between developing large software systems and designing code for scientific research purposes by attuning researchers to the issues involved in securing fully functional code rather than recommending any stock procedures for testing.

**Chapter 8, Developing Software for a Scientific Community: Some Challenges and Solutions**, also draws upon the practices of professional software developers in considering the problems in designing software for a wider community of users through transfer of tools from software engineering. This chapter approaches that same issue of adopting software engineering methods and tools from a different angle, making a more anthropological comparison of the situations of scientist end-users developing their own software and of commercial software developers in an attempt to assess the suitability of different models of software development for the specific demands of scientific communities.

The Enabling Grid for E-Science is a European Community supported initiative aimed at developing good infrastructure for CSE, and **Chapter 9, Opportunities and Challenges in Porting a Parallel Code from a Tightly-Coupled System to the Distributed EU Grid, Enabling Grids for E-sciencE**, reports on a challenging experience of porting codes between the different architectures of a local facility that was a tightly coupled systems and the E-Grid, which is a distributed system.

**Section 2: Applications**

Computer networks that carry valuable informational resources and security are a vital element in computer systems. Encoding is one means of creating security and **Chapter 10, Development of an Efficient and Secure Mobile Communication System with New Future Directions**, describes an encryption algorithm in terms of its organisation and its demands on processing as well as discussing its implementation in hardware, on FPGA chips. Like Chapter 4 it covers a hardware application area that is new to CSE methodology and so provides a case study of CSE methods as well as reporting a novel numerical method.

**Chapter 11, Parallel Quantum Chemistry At the Cross Roads**, considers how a scientific field already having extensive experience of parallel computing is to develop in the face of the much greater scale on which computing can nowadays be done. The crossroad is defined by the techniques preferred by two different groups in collaboration, the research and computational chemists having no understanding of the techniques – Directed Acyclic Graphs (DAG) – that the computer scientists favour.

The next two chapters address the simulation of biological systems. They are both strongly focused on the numerical methods that are necessary in the development of new scientific discoveries and have some reliance on “random” behaviour. **Chapter 11, Stochastic Simulations in Systems Biology**, addresses the issue of randomness (stochastics) in the biochemical reactions that occur in biological systems, while **Chapter 12, Information Geometry Applications in Biology and Physics**, is concerned with modelling biological processes and systems. It focuses on the numerical aspects of parametrized families of probability density functions and how these are can be applied to biology on different scales, from the molecule to the population.

In **Chapter 13, Some Illustrations of Information Geometry in Biology and Physics**, the authors illustrate some aspects of the methodology through case studies in: homogeneous statistical evolutionary rate processes for epidemics, amino acid spacings along protein chains, constrained disordering of crystals, distinguishing nearby signal distributions, and testing pseudorandom number generators.
Finite Elements Analysis (FEA) is well established in CSE as a way of modeling many physical processes, such as the spread of heat or the flow of liquids. There are many possible numerical methods that can be used for modelling these physical systems, and the next two chapters discuss two relatively new techniques. Chapter 14, *DQ Based Methods: Theory and Application to Engineering and Physical Sciences*, puts the case for the Differential Quadrature Method (DQM) as more computationally efficient in certain applications than the FEA method. While this chapter is concerned with the general application of the method across physical domains, it presents case studies where it is applied to problems from both science and engineering.

Chapter 15, *Mesh Morphing and Smoothing by Means of Radial Basis Function (RBF): A Practical Example, Using Fluent and RBF Morph*, compares an application of the Radial Base Function calculations with FEM; in doing so it presents the nature and application of the RBF method and persuasively demonstrates the high quality of RBF results. The structure of RBF’s software implementation is discussed and possible new uses are given, most importantly the coupling of simulations and the development of models with multi-physics. Although this is a technique that is very visual, it is not a visualization technique *per se*.

Visualization practices, too, are evolving with, and spurred by, developments in computational resources, facing a new set of possibilities that can facilitate scientific applications. Chapter 16, *Visualization: Future Technology and Practices for Computational Science and Engineering*, explores two main strands: developments giving users greater control over and increased interaction with simulations, and those that affect the quality of the user’s experience of the technology and understanding of the information being visualized. Chapter 17, *Visualizing Indicators of Debt Crises in a Lower Dimension: A Self-Organizing Maps Approach*, presents research into a CSE visualization application. The risk and occurrence of sovereign debt default – the failure of states to manage their national debt obligations – is of continuing and currently rather nerve-wracking interest, but the capacities for tracking the emerging signs of a sovereign debt crisis are underdeveloped as yet. An adaptation of a neural networks model, that of the Self Organising Map (SOM), is used to show how early indicators of impending sovereign debt default may be recognised and visually displayed.

Chapter 18, *Improving Computational Models and Practices: Scenario Testing and Forecasting the Spread of Infectious Disease*, reports on an attempt to add computational steering to a simulation of the spread of infectious disease. The complex model, which normally runs in batch mode, can now be run interactively with a visualization frontend, yielding considerable advantages.

The application of neural network structures to other phenomena has already been encountered in application to economic phenomena in the case of the sovereign debt crisis, and Chapter 19, *Artificial Neural Network Modelling of Sequencing Batch Reactor Performance*, considers how neural network models can be applied to biological processes in the context of engineering water management systems through the use of Sequencing Batch Reactors (SBR), involving a system of tanks, to recycle waste water. The application of simulation to SBR and similar biological processes is a new technique in waste water management, and here, where the results are compared to experimentation, a promising future is indicated.

**Section 3: Organizational and Practical Issues**

This section presents chapters discussing issues rarely raised in the CSE literature. Two chapters that could have been included in this section are: One, Chapter 8, *Developing Software for a Scientific
Community: Some Challenges and Solutions reviewed scientific research practices within CSE software development. The other, Chapter 11, Parallel Quantum Chemistry At the Cross Roads, considered which route computational chemists will take as new disruptive changes occur in petascale computing.

Given the background of the general changes in computational resources and CSE, Chapter 20, The State of Development of CSE, considers their impact upon the nature of scientific research, reviewing the current situation of CSE, and, especially, the changes in relations between collaborating scientific and computational specialists, as well as in the organisational structures that have framed scientific research for some time and have not, as yet, fully transformed to support the new forms of work that CSE enables.

Chapter 21, Integrating Data Management and Collaborative Sharing with Computational Science Research Processes, continues this theme, but against the broader environment of scientific data management, reviewing the way in which the lifetime of scientific data can be securely supported by adapting data management processes that have been successfully developed in other fields. The chapter sets out a structure for data management systems and through case studies discusses the ways in which these can be adopted into the scientific research process.

As CSE becomes increasingly complex, and as projects involve transitions across different domains of cyberspace, so the issue of data security becomes more prominent. Chapter 22, Security and Trust in a Global Research Infrastructure, looks at the impact of scaling up projects on authentication of access, which is a cornerstone of most security systems using real world examples.

All the chapters in this collection are concerned with advancing the collection of knowledge, but they have not involved much consideration of changing research practices. Chapter 23, CSE as Epistemic Technologies: Computer Modelling and Disciplinary Difference in the Humanities, raises these issues by exploring the impact of CSE and its nature on knowledge. It distinctively uses a case study from the humanities (rather than natural and social sciences) to ask about the way in which ideas driven by scientific and engineering needs will impact on disciplines that are scholarly in nature, and whether that will transform understandings of what knowledge and truth are.

Scientific knowledge has many applications outside the sciences and many consequences in the social world at the same time as it becomes ever more complex and, to outsiders, increasingly opaque. This has produced, among other effects, the development of a movement concerned with the public understanding of science, and the final chapter, Chapter 24, Science Communication with Dinosaurs, describes how CSE technologies have both shaped the practice of palaeontology research but also the communication of scientific results to the public, using a case in which the public has a strong interest, dinosaur research.

EXPECTATIONS

Teaching computational science has not yet taken off, and many of those starting out in the field will enter it with their first job after completing a MSc or PhD while others will move into it mid-career. This book would be an ideal text to support these researchers as well as those managers and policy makers who may find computational science important to them mid-career.

However, it is not an introductory text. It communicates current and active research from a variety of disciplines and application areas. We sincerely expect this book to will stimulate further research and developments in the technical computational and scientific research areas are discussed, but we
also hope that it might encourage more explicit discussion of organisational and practical issues, such as organisational structures to support collaborative work, policy making, technology transfer, research practices, and software development. These are all currently discussed in a diffuse way, but one of the key aims of this book is that by linking them together in a coherent way with active CSE research their interconnections can be more clearly seen.

Joanna Leng
Consultant, UK

Wes Sharrock
University of Manchester, UK

REFERENCES