The importance of modeling for Computational Science

L.O.Hertzberger

Informatics Institute
Faculty WINS
University of Amsterdam

Abstract

In this paper some developments in computational science and engineering are presented. It is observed that correct models are an essential prerequisite for this science and engineering discipline. In addition it is illustrated that the level of abstraction in these models is of crucial importance given the complex system study at hand. The influence of the computer technology on the models themselves is further discussed.

Contents

1 Introduction 2

2 Computational Science: a historical overview 2

3 Computational Science 3
  3.1 The role of Computational Science 3
  3.2 Models and Computational Science 4
  3.3 Impact of computer technology 6

4 Conclusion 7

1 Introduction

One of the various activities of Johan when he is at Stanford is the organization of the CSLI workshop . I was honored to receive an invitation to speak at the workshop in May 1997 on the subject of the developments in *computational science*. Modesty forces me to observe that the social activities we undertook, being a very good dinner together with Pieter Adriaans, turned out to have far more impact than my talk, because it was the start of Pieter becoming a professor in Johan's department in Amsterdam. Johan suggested on more than one occasion that we should give follow up on the ideas that were presented. That this has not happened until now is due to the sloppiness on my side and it is only fair to write something of it down on the occasion of this event.

2 Computational Science: a historical overview

Besides the argument whether computational science is a real science or a support science, a discussion I leave here for others to decide, it can be observed that it has had an impressive impact in the last two to three decades. The reason that a large part of the impact originates in this timeframe comes from the nature of computational science. It owes its existence to two major driving forces being the pull from challenging applications that cannot be realized without a computer and the push from computer technology itself. Concerning those applications it can be observed that already during the second world war, the Manhattan project aiming to produce the first atomic bomb as well as cryptography necessary to shield electronic message traffic during that time, were among the first to pull computational science developments. The importance of the technology is illustrated by the fact that when the first fully electronic computers like the Eniac became available they marked a big step forward in computational science.

Nuclear warfare applications, leaded by its major test laboratory Lawrence Livermore, has been one of the major technology pulls for generations of new supercomputers. Till the end of the seventies Seymour Cray was one of the major designers of these thoroughbreds in computer technology and consequently made a major contribution to the field. Numerical mathematics producing the key algorithms that exploited the vector pipelines in these machines can be considered to have produced the first stage of maturity in computational science.

However, because of its economic impact the industrial application pull in computer science and engineering rapidly shifted towards other industrial fields, such as the airplane and later the car and oil exploration industries. Because of that, Computer Fluid Dynamics (CFD) became one of the widely studied applications in computational science. In the mid seventies, a new computer technology emerged in the form of the microprocessor, which gradually in the eighties started to challenge the unassailable position of the supercomputer via the emergence of massively parallel machines. This development as well as the push for ever larger supercomputers was in the beginning of the nineties pulled by large application programs such as the Grand Challenge programs [1] aiming to get a breakthrough in a number of scientifically and economically important areas such as weather forecast, air and water pollution research, financial modeling, the study of 3D plasma's etc. Current computational science techniques are used to study complex systems and solve problems in a wide variety of areas such as water and air pollution (the hole in the Ozone layer), car crash design, reactor safety, the study of the human genome, the study of the phenomena in astronomy, light scattering on complex objects, the grow of sponges etc.

3 Computational Science

Computational science uses modeling and computer simulation to study the behavior of complex systems. In studying these complex systems by computational science techniques, three distinct levels of abstraction play an important role in the mapping process onto a computer. These are :

- A conceptual model of the system
- A computational model of the system
- An algorithm for mapping the computation onto the computer

Characteristic for these abstraction levels is that a number of design choices are made in the mapping process that are influenced by the possibilities of computer technology. As each of the abstraction levels in fact produces a different new model, extensive validation and verification are essential. When the burners of a chemical reactor are studied using computational science techniques CFD methods are applied. The conceptual model is represented by Navier Stokes equations producing a set of Partial Differential Equations, the computer specific model or solver are numerical methods for solving these equations, whereas a multi-grid algorithm is used to carry out the computation.

3.1 The role of Computational Science

There are various debates going on whether modeling and simulation is, besides theory and experimentation, a third methodology or whether it is an aid on towards the other two. It is the authors belief that it is more important to study the possibilities complementary to other scientific methodologies. Computational science methods can be used to understand the behavior of complex systems, e.g.:

- To study large amounts of parameters to be used for complex system design and construction. Examples could be an airplane or a car.
- To search for and understand phenomena that lay outside the domain of laboratory experimentation. Examples could be phenomena in astronomy or to study the pollution in water or the atmosphere.
- Model forming and validation to get insight in the behavior of complex physical systems. Examples could be light scattering on complex objects or radar scattering or the development of cracks in materials.

In some of the examples (for instance in astronomy) the power of the method can be enhanced by combining results from modeling and simulation (for instance about a star nebula) with real measured data. In all situations, the modeling step is the first essential activity in the mapping process.

3.2 Models and Computational Science

Making a conceptual model is one of the most important activities in computational science, because it is the start of every mapping chain. Before going into more detail, let us quickly evaluate the role of modeling in the scientific and engineering process. In science, models are used to understand the behavior of complex system such as the scattering of light on large objects with irregular shapes. Modeling and simulation is used for the analysis of a system to get a better understanding. Although analysis in order to better understand a complex system is also playing an important role in engineering, its essence is *design*. In the design process, an engineer wants to modify and optimize the system such that it is behaving according to his demands. In science and partly also in design, the models are based on some laws of nature (first principles), like in the application of CFD in the earlier example of the burners. In this case, these laws result in the Navier Stokes equations that are used to model the process. Similarly in the study of complex processes that govern the evolution of stars, hydrodynamics plays a dominant role. However, for a complete understanding of all processes active in star evolution, these models can become so complex, because so many different parameters have to be taken into account, that they can not be solved in all details by current computer capacity. Such problems are often called intractable. The only way to study such systems is to model them at different levels of abstraction and accept less information than is necessary to understand the operation of a full system in all details. Abstraction plays a dominant role in computational science. In fact each model is an abstraction from reality, and for that reason validation of the results is such an important aspect. With computational science applications spreading, also to areas that were not accessible before, the aspect of abstraction levels in models is becoming an increasingly dominant issue. This not at least because more and more computational science techniques are applied to systems where all first principles are not that well understood or cannot be applied. Here we will shortly illustrate this issue by looking at two engineering applications in relatively new areas; the first embedded systems design, the other electronic fee collection.

Embedded systems are systems where a computer or special electronic circuitry is used to provide the system's functionality. Examples are all types of mobile communication devices, TV's, set top boxes, medical equipment etc. With the increasing possibilities of microprocessor technology and its rapid decrease in price, it is becoming more attractive to integrate microprocessors with other electronic components (dedicated hardware) into one embedded system. This is even more important as the complexity of these systems increases because of the requests for more functionality and the impact of standardization to fulfill these demands. Designing such systems involves a tradeoff regarding which parts of the system functionality should be mapped on a processor and which ones on dedicated hardware. This so called hard- software co-design requires a design framework that can model various system components at different levels of abstraction. When exploring all design possibilities, more crude models at a higher level of abstraction are required than when the design is fixed and detailed performance optimization studies are necessary. The key issue here is to determine the right abstraction levels for the questions posed to the models, which can only be realized after extensive experimentation, testing and analysis. In addition, it illustrates that such behavioral simulation cannot be based on the first principles.

Another example of the previous problem in system design and evaluation is presented in the next case. Electronic Fee Collection (EFC) systems are systems that collect fees for the usage of roads without the need to install toll booths and consequently interfere with the free flow of traffic. In the Netherlands, it is planned to install these systems on major motorways and secondary roads in the western part of the country. Before installation on a large scale, a number of (design) questions had to be answered, such as: will it be possible to handle lane changes, what type of payment means (e.g. smart card or token card) is fast enough to handle the requested transactions, what is the detection system error rate under various weather conditions and what is its impact on the behavior of the total system. All these and many other questions had to be answered before a full-blown system will be installed on the roads. A modeling and simulation framework [2] was developed for the evaluation of various alternatives and later design (optimization) studies. An EFC system can be considered to be composed of five major sub-systems, of which the first four are:

- The communication sub-system responsible for the payment transaction between car and wall station it passes.
- The detection sub-system responsible for detecting the vehicle.
- The classification system identifying what type of vehicle is passing.
- The registration system for number plate identification.

Based on the observation of each individual sub-system, the fifth system being the coordination system decides on the overall action of the total system. Each of the sub-systems can be represented by a model. The overall simulation is driven by a traffic generator which has to generate representative traffic for the situation under study. In our case, also measurements from real world traffic were used to drive the simulation studies. Model abstraction has played a dominant role in this case study. Taking the communication between wall station and vehicle as an example, a variety of abstraction levels can be considered. One of the lowest is the physics of the electromagnetic communication. A higher one is that at the level of the communication protocol, next is the financial transaction protocol etc. The system will be prone to all types of errors that will be generated by these processes. The modeling approach we have taken was to use hierarchical decomposition and to apply discrete event modeling using stochastic allocation of errors to various system components and let the level of detail being determined by the type of questions that had to be answered by the total system [3]. It turns out that the majority of the questions that have to be answered can be done using high level models.

Another aspect that became apparent is this work is the fact that modeling itself becomes considerably more complex when human behavior also has to be taken into account. Although a considerable effort was devoted towards traffic generation in this project, it still can not be claimed that a fully validated traffic generator has been realized. Although global behavior of traffic jams (stop and go traffic and backwards traveling density waves) can be modeled relatively easy using one dimensional cellular automata, including more details inducing inherent noise, being the human interaction, is far more difficult.

3.3 Impact of computer technology

An illustration of the measure of interweaving of modeling and computation can be observed from the EFC example where the simulation methodology has resulted into importing discrete event methods into the modeling approach. However, most of the computer technology impact in modeling comes from the desire to exploit the parallelism of the hardware architecture. As the algorithm is the last step in the mapping chain, it is not surprising that in particular numerical algorithms were considerably influenced by the type of parallelism present in the hardware. The aspects of vectorization for numerical algorithms were mentioned already. Here we will shortly illustrate the impact of massively parallel computers that became available due to the advances in the price/performance ratio of commercial microprocessors. It resulted in a large renewed interest in numerical procedures (such LU factorization) to optimize the mapping on those new machines. It also revived a wide interest in modeling methods like cellular automata, genetic algorithms and particle methods that are intrinsically parallel and may often better model the physical process at hand. Here we will illustrate this via particle methods, more in particular Lattice Boltzmann methods. If we go back to the example of Fluid Dynamics, the conceptual model that is now applied is that of a lattice gas, being a population of particles following microscopic rules according to conservation laws. In this situation the conceptual and computer specific model are the same. They are mapped via a simple algorithm of propagation and collision steps into a computation. Because of the better preservation of the physical characteristics of the original complex system, they better preserve the inherent locality of the underlying physical laws and consequently the parallelism in the original system. This makes them more suitable for mapping on parallel computers. In addition they provide a simple way to handle complex boundary conditions allowing to model complex 3D geometries. Applications can be found in studying fluid flow through porous media like filtering paper [4] (illustrated in Figure 1) or ground water. Also, the effect of nutrient diffusion and flow on growth processes is a good example (as illustrated by the simulated coral in Figure 2). These examples also illustrate a next step in the maturity of computational science. In previous efforts the emphasis was on the study of 2D problems, but due to the rapid advances in compute power and memory the studies of complex systems in 3D are now becoming commonplace.

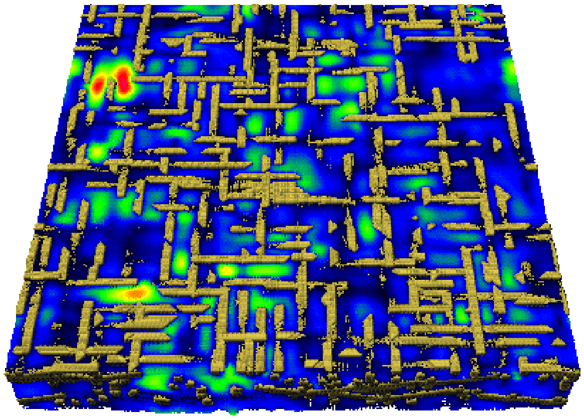

Figure 1: The flow velocity of a fluid through a fibrous medium.

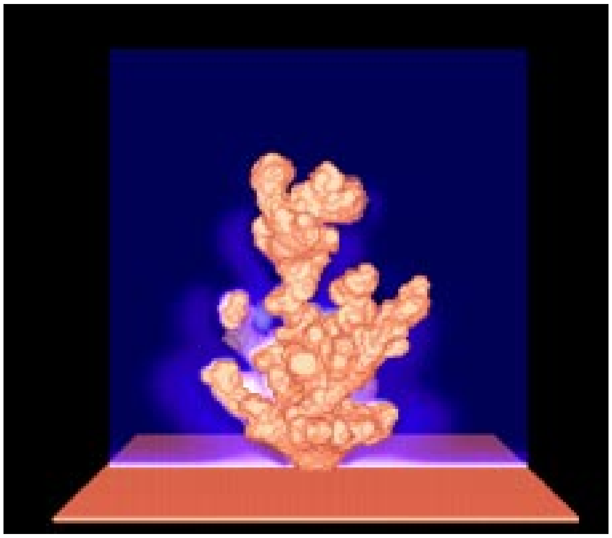

Figure 2: Simulated coral.

4 Conclusion

In this paper, some developments in computational science and engineering were presented. It was observed that correct models are an essential prerequisite for this science and engineering discipline. In addition it was illustrated that the level of abstraction in these models is of crucial importance given the complex system study at hand. The influence of the computer technology on the models themselves was further discussed.

Acknowledgement

This paper illustrates a number of examples of work from the author's group as well as the group of Peter Sloot. In particular discussions with Alfons Hoekstra and Andy Pimentel have to be acknowledged.

References

[1] “Grand Challenge: High Performance Computing and Communications”, NSF Report, Committee on Physical, Mathematical, and Engineering Sciences, Washington D.C.: US Office of Science and Technology Policy, National Science Foundation, 1992.

[2] “ADS-SIM: a generic simulation environment to evaluate and design Automatic Debiting Systems”, A.G. Hoekstra, G.R. Meijer and L.O. Hertzberger, in Proceedings of the 14th IPS Conference, Berlin, 1997.

[3] “Evaluating Automatic Debiting Systems by modelling and simulation of virtual sensors”, L. Dorst, A. Hoekstra, J.M. van den Akker, J. Breeman, F.C.A. Groen, J. Lagerberg, A. Visser, H. Yakali, L.O. Hertzberger, invited paper for IEEE Instrumentation and Measurement, 1998.

[4] “Permeability of three-dimensional random fibre webs”, A. Koponen, D. Kandhai, E. Hellen, M. Alava, A.G. Hoekstra, M. Kataja, K. Niskanen, P.M.A. Sloot and J. Timonen, in Physical Review Letters, 80:716–719, 1998.