Chapter 1

Introduction

This chapter introduces the use of partial differential equations to characterize the behavior of physical systems and the computational bottlenecks arising in the computer solution of these equations. Based upon this motivation, the use of parallel computers to provide suitable computational resources is proposed. The importance of these improved solution capabilities in the realm of semiconductor device characterization is also discussed. Finally, the organization of the remainder of this thesis is presented.

1.1 PDE Approximations and the Computational Bottlenecks

Many physical phenomena of interest to scientists and engineers can be described by sets of partial differential equations (PDEs). Although general closed-form solutions to these PDEs are often unobtainable, a discrete approximation to the solution can usually be obtained using a computer. Accordingly, computer modeling of physical systems is used in many scientific and engineering disciplines. (Examples include fluid dynamics, structural mechanics, and electronics.) Computer simulations based upon the approximate solutions of PDEs are used not only to verify the understanding of experimental results but also to reduce the amount of expensive and time-consuming physical experimentation. Unfortunately, PDE solvers generally demand large amounts of computational resources, and as model complexity grows, conventional uniprocessor architectures become overwhelmed by the computational demands.

Table 1.1 shows the runtimes from several typical PDE-based computer simulations. These state-of-the-art simulations from computational fluid dynamics (CFD) and semiconductor device physics were run on a collection of high-performance uniprocessor computer architectures. The CFD simulations were run using the commercial CENTRIC Spectrum solver [54]. The device simulations were run using PISCES [41] and FIESTA-HD [1], two semiconductor device modeling codes developed at Stanford University. The runtimes shown are for single simulations; a design optimization might require large numbers of such runs. Such unacceptably high execution-time requirements limit the utility of computer simulations. To realize the full potential of computer simulation, the computational bottlenecks must be overcome by employing high-performance computational resources which can significantly reduce the overall time-to-solution and provide complete solutions within an "engineering reasonable" amount of time. From this perspective, the most convenient and meaningful measure of the success of performance optimizations is the overall reduction in the time-to-solution.

Table 1.1 Survey of PDE-based simulation runtimes.

Problem	Solver	Computer	Time
Bluff Body Wind Noise	Spectrum	IBM RS6000/590	38.6h
Truck Cabin Wind Noise	Spectrum	HP900/755	80h
High-speed Train	Spectrum	Cray EL92	>100d
AC Airflow in Car	Spectrum	Convex C4	121.7h
CMOS Inverter	PISCES	IBM RS6000/590	4.5h
Photo-Emitter	PISCES	IBM RS6000/590	11.2h
Bipolar Transistor	FIESTA-HD	IBM RS6000/530H	45h

1.2 The Suitability of Parallel Computers for PDE-Based Simulations

As parallel computers built with high-performance microprocessors and supporting the message-passing programming paradigm become ubiquitous, using these architectures to further improve modeling capabilities becomes sensible. In fact, parallel machines now provide the highest levels of performance [3] and have the potential to open new frontiers in PDE-based simulation. Parallel grid-based PDE solvers have had measurable success at the research level as demonstrated by Zdenek [32], Wu [52], Fatoohi [15], Procassini [44], and others. However, large-scale application development has been slow because the programming models required to take advantage of these architectures are a radical departure from traditional programming paradigms. A parallelization methodology for large-scale PDE applications must address the following issues:

- It may be impractical to completely scrap existing uniprocessor applications. The reuse of existing application components is desirable in order to minimize the development effort and retain the value inherent in existing uniprocessor applications.
- PDE applications often rely upon libraries to perform some functions (e.g., Gaussian elimination). The shift to parallel architectures complicates the integration of library modules into applications.
- Applications should be easily portable across parallel hardware architectures.

A methodology exploiting the existing high-level algorithms and data structures in serial PDE applications to create fully parallel PDE applications would allow for the adaption of existing codes for parallel execution as well as simplify the creation of new parallel PDE codes. A suitable approach can be based upon the explicit message-passing programming model which is easily portable to any parallel computer supporting message-passing a library. Furthermore, this coarse-grain message-passing model provides a flexible and intuitive framework for the parallelization of grid-based PDE solvers without forcing significant changes to either the data structures or the code.

Finally, the resulting parallel applications provide excellent test-beds to aid the development of appropriate library modules for large-scale parallel applications.

1.3 Parallel Computing for Semiconductor Device Characterization

As time-to-market for integrated circuits diminishes and manufacturing technology becomes increasingly complex and costly, semiconductor device simulators become crucial in quantifying the electrical behavior of devices. Simulators are now used to perform not only design verification but also manufacturability and scalability studies. The interplay between adjacent devices as they are scaled to submicron sizes becomes more important and can dramatically increase the memory and execution time requirements of simulations. These two trends mean device simulators must realize significant performance gains to provide reasonable turnaround for device and process designers. The use of scalable, high-performance parallel architectures to deliver that performance will become strategically important in the continued success of large-scale device simulation. Accordingly, parallel applications must be judged on their ability to reduce the overall time-to-solution for meaningful device simulations.

1.4 Organization

The remainder of this thesis is organized in the following manner:

Chapter 2 summarizes the popular methods used to obtain approximate solutions to PDEs and highlights the underlying common structure. The exploitation of the parallelism inherent in these methods is then presented and the issues raised by this coarse-grain parallelization approach are discussed. Subsequently, an overview of applicable parallel hardware is given.

Chapter 3 presents the parallelization of a semiconductor device solver using the coarse-grain approach. The importance of using application knowledge to efficiently resolve data dependencies is emphasized. Also, the creation of a reusable application framework is discussed and the problems encountered are presented in-depth. Results which demonstrate the ability to dramatically reduce overall time-to-solution using the improved computational capabilities are presented.

Chapter 4 discusses approaches for the integration of direct linear solution modules into the parallel device solver presented in Chapter 3. An application-oriented approach is chosen and shown to provide good solution capabilities within the application.

Chapter 5 presents a parallel adaptive linear solution method developed to address some of the shortcomings of fully direct methods. This scheme seeks to take advantage of the complementary natures of direct and iterative linear solution methods. This method analyzes the data flow within the linear system and performs a partial factorization of the matrix. The remaining equations are then solved using an iterative method. Results demonstrate reduced memory requirements as well as improved parallel scalability.

Chapter 6 presents conclusions and suggests directions for future work.