Chapter 6

Summary and Future Work

This chapter summarizes the work presented in this thesis. Based upon the results, directions for future work are then suggested.

6.1 Summary

The need to reduce the overall time-to-solution in PDE-based computer simulations by alleviating the computational bottlenecks motivated this work. Although parallel hardware architectures have the ability to provide sufficient computational resources, a suitable parallel software architecture is also needed. In complex large-scale applications, considerations such as code reuse, portability, and non-local data referencing can greatly complicate software development. Despite these concerns, a conservative coarse-grain software parallelization approach can be used to dramatically reduce solution times. The coarse-grain methodology exploits the existing high-level algorithms and data structures in serial PDE applications.

The parallelization of PISCES using the coarse-grain model has demonstrated its efficacy for complex applications. The value of this "dusty deck" simulator was preserved by minimizing and localizing the modifications to the existing code. The resulting coarse-grain application was easily portable across a wide range of parallel hardware architectures. Solution capabilities were enhanced by greatly reducing overall solution times for a wide range problems. In fact, the behavior of the parallel code as problem sizes increase suggests a vast potential for simulating large and ultra-large structures. In the semiconductor industry, the competitive advantage gained could have vast benefits. The subsequent parallelization of a second semiconductor device simulator, FIESTA-HD, demonstrates the generality of the flexible communication framework and application-area-specific domain decomposition tools created during the parallelization of PISCES.

The coarse-grain parallelization of PISCES also demonstrates the feasibility of using parallel direct methods within realistic large-scale parallel applications. The parallel multi-frontal method used provides acceptable performance for both small and large PISCES-MP simulations. For small simulations, matrix factorization times have been reduced to fractions of a second using eight to sixteen processors. As the number of processors increases, the necessarily small subdomains and the application-imposed load imbalances limit the improvements in execution times. Despite this behavior, the linear solver is sufficient to allow good improvements in overall solution times when viewed from an application perspective. For large-scale problems, the linear solver does provide excellent performance. Good scalability has been demonstrated on up to 128 processors for problems of up to 1.5 million equations. Moderately parallel computers (8-16 processors) are suited can provide substantially improved execution times for moderately sized problems. Highly parallel computers (64-128 processors) are generally intended to solve “grand-challenge” problems. Results have demonstrated that these machines provide excellent solution capabilities when appropriately utilized.

An adaptive direct/iterative solution method was subsequently developed to address some of the shortcomings inherent in parallel direct methods. This approach is based upon the existing multi-frontal direct solver and provides greater flexibility than traditional linear solution methods. The symbolic factorization step adaptively chooses the breakdown between equations to be directly eliminated and equations to be solved iteratively. Straightforward modifications to the numeric steps are then made to work with the symbolic representation of the partially factored matrix and the reduced linear system.

Any appropriate iterative equation solver may be integrated into this framework, making the adaptive solver suitable to many application areas. The resulting “meta-solver” allows any combination of direct and iterative solution and provides the ability to efficiently exploit computational resources by benefiting from the strengths of both direct and iterative methods while minimizing their weaknesses. In a serial environment, this solver efficiently solves ill-conditioned problems by balancing the memory demands of direct methods against the convergence properties of iterative methods. In a parallel environment, the adaptive solver can reduce both the memory requirements and the solution times with respect to a fully direct solution. As the number of processors increases, both the superior scalability of iterative methods coupled with the more orderly distribution of directly eliminated equations resulting from the parallel partitioning provide excellent improvements over the fully direct solver.

6.2 Future Work

Many of the directions for future work proposed in Lucas' dissertation [35] have been addressed by the work summarized above. Similarly, based upon this thesis, several issues present themselves as topics for future study and development.

PDE-based computer simulation is important in the characterization of the behavior of physical systems. The new parallel semiconductor device simulator, PISCES-MP, provides users with significantly enhanced computational capabilities. By employing this computational power to perform simulations heretofore computationally infeasible, the understanding of these complex physical systems can be improved.

Although this work has focused on two-dimensional simulations, the coarse-grain methodology should be extensible to three-dimensional simulations. However, verification of this extensibility is needed to further explore the issues arising from the addition of the third dimension.

Further testing and development of domain decomposition algorithms for irregular meshes is needed.

The integration of diverse application modules with differing domain decomposition criteria warrants further study. Although a single, application-oriented decomposition was sufficient in PISCES-MP, a simple and efficient method for maintaining multiple decompositions within a single application may prove useful in other applications.

The adaptive direct/iterative linear solver encompasses both direct and iterative methods as well as their interactions. This “meta-solver” framework presents many directions for future study.

- Improvements in parallel direct techniques will improve overall performance.
- Improved preconditioning in the iterative solution step could be used to improve iterative convergence while performing less direct elimination.
- The addition of other iterative algorithms into the “meta-solver” framework would make the adaptive solver useful in a wider range of application areas.
- The higher workload in three-dimensional simulations should dictate the use of improved iterative schemes to solve a larger portion of the linear system. In this case, the trade-offs between the direct elimination and iterative solution will require more study.