

AD-A284 848

DOCUMENTATION PAGE

Form Approved

OMB No 0704-0188



ation is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Avenue, 2nd Floor, Washington, DC 20540-6001, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

2. REPORT DATE

July 28, 1994

3. REPORT TYPE AND DATES COVERED

Final Report/July 1, 1991-May 31, 1994

4. TITLE AND SUBTITLE

"Mega-Scale Simulation of Multi-Layer Devices--
Formulation, Kinetics, and Visualization"

5. FUNDING NUMBERS

DAAL03-91-G-0152

6. AUTHOR(S)

Robert W. Dutton

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stanford University
AEL 203
Stanford, CA 94305-4055

8. PERFORMING ORGANIZATION
REPORT NUMBER

P-28297-EL

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING
AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

94-30706



12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A new energy transport model including both carrier and lattice temperatures has been developed and implemented in PISCES 2ET. Major capabilities in physical models for compound semiconductor devices include: heterojunction interfaces, deep level trapping and new mobility models. Applications of PISCES 2ET in the modeling of GaAs MESFET sidegating and electronic effects in light-emitting structures have been achieved. The GaAs MESFET modeling of dc and ac effects have been confirmed experimentally at Stanford and in collaboration with industry. The LED and vertical cavity laser modeling is being applied by Hewlett-Packard in both their research laboratories and product divisions. Algorithms developed for improved accuracy and efficiency in device modeling include: ac analysis for microwave devices, multi-processor direct solvers and massively parallel iterative solvers. Supported under the high-performance computing (HPC) initiative, a prototype version of PISCES-MP running on Intel, Thinking Machines and IBM parallel machines has demonstrated order-of-magnitude speed enhancements compared to the single processor version. The parallel iterative solver in the STRIDE 3D code has solved device problems with 15 million equations in 20 minutes on a 520 processor Intel machine. A mixed-mode analysis capability that couples PISCES and SPICE has been demonstrated and applied to SRAM and OEIC applications.

14. SUBJECT TERMS SPICE; silicon-based technology; compound semiconductors; multi-layer, multi-material devices; heterostructures; energy transport (ET) model; hydrodynamic; drift-diffusion; mixed-mode device/circuit simulation; SOI; substrate current; device simulations; PISCES; OEIC; deep-level traps; MMIC; GaAs/AlGaAs; parallel computing; high-frequency ac analysis

15. NUMBER OF PAGES

89

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION
OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION
OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

UL

ARO Final Report

Accession For	NTIS ☒ CRA&I ☐ DTIC ☐ TAB ☐ Unannounced ☐ Justification ☐	By	Distribution /	Availability Codes	Dist	Avail and/or Special
						A-1

INTRODUCTION

The PISCES program developed by Stanford University has become a world-wide resource for both academic research and industrial applications in device modeling. The PISCES code supports two-dimensional (2D) analysis of electronic devices for dc, ac and time-transient operating conditions. Over more than a decade there have been three major new releases of the code (versions 2, 2B and 2ET) as well as commercial vendors that now support proprietary versions based on the original Stanford code. Over the past half-dozen years the Semiconductor Research Corporation (SRC) has become a partner with ARO in the research support for PISCES developments, their major focus being on the developments that support silicon-based technology. There have also been ARPA supported contracts to create parallel computing versions of both 2D and 3D device analysis based on PISCES and a 3D prototype code STRIDE, also initially developed under ARO support. The focus of the ARO supported research activities has been in the areas of multi-material modeling for advanced physical models and in the underlying techniques and algorithms that support efficient computation.

OBJECTIVES

The goal of this research is targeted at developing new methods and prototype design tools to support modeling of advanced multi-material devices that are required for high-speed circuits. The simulation of such devices has become an essential stage in the design and manufacturing of new components. With the rapid evolution of synthesized heterojunction materials, it is of major benefit to have models to provide physical underpinning to the very expensive and empirical technology development cycle.

In order to support the overall modeling goal, the research objectives center on developing new, flexible simulation capabilities and on providing advanced physical models in order to support the revolutionary developments in device technology. From a computational perspective, the development of framework utilities to reduce the user burden and the exploitation of parallel computers to make the simulations more efficient are key areas of research activity in this project. A major focus in this research has been to develop physical models and numerical techniques that support analysis of compound and heterojunction material devices.

RESEARCH RESULTS

The scope of the efforts supported by this ARO is centered on models, algorithms and prototype 2D design tools that support multi-layer device simulation. Nonetheless, over the course of previous ARO contracts that supported PISCES and related algorithmic developments, key concepts have been developed that have now matured and are bearing fruit in the context of other projects. As a result, this report includes both the immediate outcome of the presently funded work as well as the spin-off results that had their foundations in earlier ARO supported contributions.

The report is organized into three main sub-sections and a set of supporting appendices. First, the key contributions to **physical modeling** and the supporting formulations are presented. Second, there are a number of **technology applications** where the modeling activities have been bench-

marked and the industrial benefits can be measured. Third, the underlying framework and computational aspects of the modeling work are presented. Here too there are a number of applications resulting from the growing industrial use of the tools and the national project in High Performance Computing (HPC). Since a majority of the research supported by this contract has been published at conferences and in other archival journals, the use of appendices that draw from this body of work will be the main approach for summarizing technical details. Finally, it should be noted that other supporting materials including both software and Ph.D. thesis documentation is available directly from Stanford by means of internet access using the following world-wide web URL address at <http://www-ee.stanford.edu/tcad.html>.

1. ADVANCED DEVICE MODELS---There have been three major developments in transport modeling of multi-layer materials: a) formulation and implementation of energy transport (ET) models that include fully coupled carrier and lattice temperatures, b) implementation of heterojunction interfaces in support of a variety of material systems and c) complete implementation and testing of deep level trapping for GaAs devices. The highlights of these activities are now reviewed and further details are found in Appendix 1.

1a. Energy Transport Models: Over this contract period we have developed both the basic concepts of the energy transport (ET) model [1] and a fully coupled version, called Dual ET (i.e. DUET or 2ET)[2], that models both carrier and lattice temperatures in a coupled and consistent manner. In coordination with SRC that supports the silicon applications, this research has not only developed the basic formulation but also mounted an intense effort to calibrate and check the accuracy of the physical approach. Of special interest is the calibration of mobility models [3] as well as impact ionization [4]. Quantitative agreement in the modeling with experimental substrate current results for submicron silicon devices is most encouraging. Moreover, deep submicron behavior of 0.12 micron SOI devices measured by Professor Hu's group at U.C. Berkeley shows a good match with the simulated results obtained using the DUET model [4].

1b. Heterojunction Interfaces: In order to support the modeling of high-speed and optoelectronic devices that utilize heterojunctions, major enhancements to the model formulation and implementation for device simulation were required. There are a variety of changes related to materials and interface properties, including consideration of energy band offsets. Owing to spatial variation of composition there is also a need to interpolate data across the various regions. Details of the models supported are given in the PISCES-2ET manual [5].

1c. Deep Level Trapping: The role of deep level traps on changing both the dc and ac behavior of GaAs devices has been recognized for many years but a quantitative model of the effect has been lacking. Under ARPA support of an experimental effort, including extensive process modeling and fabrication of test structures, the ARO efforts in device analysis have provided an essential compliment. Based on implementation of a general rate equation for deep level impurities, including the ac and time transient electronic trapping behavior, it is now possible to achieve quantitative agreement of side gating effects in bulk GaAs [6]. Two major mechanisms are revealed to be responsible for sidegating: 1) the formation of a Gunn domain at the channel/substrate interface (requiring a negative differential mobility model) and 2) substrate conduction between a parasitic Schottky contact and the sidegate due to carrier injection. Both effects are modeled quantitatively for the first time in this work.

2. TECHNOLOGY DRIVERS---By means of a number of specific applications, the advanced modeling capabilities developed in this research have been tested. The support for the PISCES development efforts have come from several sources. For the sake of completeness in this report, that diversity of applications will also be reflected. A representative set of technologies, both silicon-based as well as those using compound materials, will be used for illustration. In the area of silicon technologies, co-sponsored by SRC, the application to power devices is given as the primary example. For compound materials, two examples are reviewed, the first one involves GaAs MESFETs with co-sponsorship coming from ARPA. The second example is solely sponsored by ARO and considers AlGaAs used for opto-electronic (OEIC) devices. As done in the previous section, Appendix 2 is used to provide further details on each of these application domains.

2a. Silicon-based Applications: The previous section has considered advances in physical models developed in this research. The research and development of the DUET model [1] for considering the coupled temperature solution is a major outcome of this project. A large number of the applications related to silicon technology come from the commercial semiconductor (SRC) areas. The calibration of substrate current models and deep submicron SOI characteristics have already been cited. The case of SOI modeling has broad implications for both high speed communications and for radiation resistant applications. Another application currently under investigation is electrostatic discharge (ESD). Preliminary results show that the coupling of transient current pulses and joule heating is essential. It is expected that follow-on work in this area under SRC support will more completely demonstrate the full impact of the DUET model representing physical phenomena involved in ESD.

The primary silicon application reviewed here is that of modeling power devices. While Stanford has had only small, primarily industrially funded, contracts in this area, there are a sustained set of contributions over more than a decade. Of special interest here is the need to support a variety of physical models and complex geometries for power devices. Specifically, the need for mobility models that capture high field and carrier-carrier scattering mechanics as well as various other surface phenomena that occur in power MOS devices are critical [7]. The issues of complex geometry are also important factors for power devices. The use of trench isolation and cylindrical geometries are both aspects of PISCES enhancements that have received significant attention owing to unique factors associated with these devices. There are many trends in power devices, not covered specifically in this report which are expected to leverage from the newest version of PISCES. For example, the growing interest in SiC as well as other compound materials is one key trend. The analysis capabilities, coupled carrier and lattice temperatures, have already been mentioned in the context of SOI devices.

2b. GaAs Device Simulation: The modeling of GaAs technology continues to be a significant application area owing to the importance of circuits and devices for high speed communications. Under ARPA support, Stanford has developed versions of SUPREM to model GaAs fabrication processes in both 1D (version 3.5) and 2D (version 4GS). The device modeling of GaAs has been supported through both the ARPA contract and the ARO efforts reported here [5]. There are several aspects of the GaAs device modeling that are of special interest and importance: 1) the mobility model [3], 2) the trapping effects [6], and 3) the ac high frequency behavior[8].

Modeling of the negative differential mobility (NDM) and trapping have been critical in providing quantitative prediction of the sidegating effects observed in GaAs [6]. The trapping of carriers, combined with the surface and bulk boundary conditions result in the creation of a stationary Gunn domain. In contrast to earlier models that suggested other coupling between physical models such as trapping and impact ionization, the present simulations have achieved quantitative agreement with experimental studies. Further details are summarized in Appendix 1 and in the thesis contribution of Liu [9].

Another key aspect of GaAs MESFET modeling is the correct prediction of high frequency behavior. In this work, two key factors have contributed to the successful matching of experimental results up to the frequency regime of 20 GHz [8]. First, the development of robust numerical techniques to be considered in the following section, now allow ac analysis to frequencies beyond the cutoff of the device. Since measurement accuracy and resolution become questionable in this regime, having an accurate modeling capability is especially important. Second, the parasitics show critical effects at the highest frequencies. For example, the package parasitic inductance is observed to play a major role in these devices above 5 GHz. The development of both geometry modeling and mixed-mode simulation support for the distributed device effects such as GaAs FET high frequency modeling have provided major benefits as demonstrated in this project.

2c. Optoelectronic (OEIC) Devices: The simulation and modeling of OEIC device effects is a final application area stressed in the research supported through this ARO contract. In collaboration with Hewlett-Packard and Los Alamos National Labs (LANL) in their joint CRADA for OEIC modeling, the PISCES code has been extended and enhanced for laser modeling as well as for simpler LED structures being developed by the optical components division (OCD). Several aspects of the cylindrical geometry and mixed-mode analysis capabilities discussed above are also critical for OEIC modeling [10]. In the case of data communications transmitter circuits, the complexity of electrical boundary conditions on final optical output is especially important. Appendix 2 shows in some detail the interplay of geometry modeling for device analysis specification and the unique capabilities provided through mixed-mode modeling of OEICs. The key point to be reiterated here is the fact that extracted SPICE models for the optoelectronic side of these circuits is not only difficult but provides no means to directly link the physical effects inside the devices with the observed behavior. The TCAD approach being developed in this work takes a major step in the direction of solving such problems.

3. COMPUTATIONAL METHODS---Numerical techniques and advanced algorithmic approaches to device modeling have been the final area of major achievement during this contract. Over nearly a decade, under ARO support, a number of advanced algorithms for parallel computing have been developed [11]. As a result of these key algorithmic demonstrations, both the State of California through its Competitive Technology (CompTech) office and ARPA's equipment grant (siting of a 32 node Intel iPSC/860 at Stanford) provided the necessary leverage to carry forward the initial ARO work into full-scale prototype demonstrations. Namely, the development of PISCES-MP (multi-processor version) [12] and the benchmarking of STRIDE [13] on the experimental 520 node "Delta" machine sited at CalTech are key highlights of the CompTech/ARPA supported follow-on work which derives from the pioneering efforts at Stanford that were initially supported by ARO. Further details of these efforts as well as the most recent numerical results in ac and mixed-mode analysis are summarized in this subsection as well as in Appendix 3.

3a. Multi-Processor PISCES: The original demonstrations of the distributed multi-frontal (DMF) algorithm developed by Lucas [14] utilized matrix stencils created by PISCES. Nonetheless, to achieve the full-scale parallelization of the code many other changes had to be implemented. The Aladdin-CAD (CompTech) project mentioned above provided the motivation and partial support for this effort. Demonstration of PISCES-MP on the Intel iPSC platform showed excellent speed up factors up to 8 processors but node memory and interprocessor communications limits of that machine reduced the efficiency for further scaling [11]. The follow-on efforts under the HPC initiative and supported by ARPA have demonstrated further scaling of PISCES-MP on a TMC CM-5 Machine with 32 nodes [15]. Now in progress is an effort to move PISCES-MP to the IBM SP/1 and SP/2 machines which will lead to understanding of both problem size and algorithmic limits of the direct solver technology used. The sustained efforts on the PISCES project, including the development of a multi-processor (MP) version, provides a major demonstration with long term industrial payoff potential.

3b. Parallel Iterative Solvers: The interest in and needs for fully 3D TCAD applications has grown steadily over the past decade. The previous ARO contract resulted in the demonstration of the STRIDE [13] prototype 3D device solver. Under CompTech and now ARPA support that prototype has been used to generate 3D bipolar benchmark examples with 4.9 million grid that achieve convergent results over the entire operating region [16]. In fact, these results were honored in Intel Grand Challenges Computing benchmarks in May, 1992. This work has also received national recognition at the HPC Symposium (March 15, 1994) and the results are shown in the HPC "Blue Book" literature illustrating applications where HPC technology is expected to have major payoffs. Again, it should be emphasized that the core DMF algorithm and initial demonstration of STRIDE came under previous ARO-sponsored projects. DMF solver technology as well as parallelization of preconditioners required for iterative solutions have resulted in excellent overall efficiency and scalability of computation time with processors and problem size. In addition to the very exciting benchmark results discussed above, there has also been more detailed theoretical work in understanding the overall capabilities of pre-conditioning for block iterative solvers [17]. This aspect of the work is quite basic and has been primarily supported by ARO.

3c. Algorithms for ac Analysis: As noted in the previous section on use of ac analysis in the design of GaAs MESFETs, such capabilities are of major value and benefit for high frequency applications. From an algorithmic point of view, the development of robust iterative techniques for ac analysis has been a major challenge. First, previous methods that exploit direct matrix solution techniques become highly memory and CPU intensive [18]. Second, the numerical nature of the matrix changes with frequency and robustness becomes a serious challenge at frequencies above the device cut-off. The key features of the new algorithm used in this work is summarized in Appendix 2 and further details are presented in the PISCES 2ET documentation[5]. By means of careful control of the matrix conditioning, based on operating frequency and other information about device operating conditions, excellent convergence rates and overall numerical stability have been achieved through research supported in this contract.

3d. Mixed-Mode (PISCES-SPICE) Simulations: There has been a number of developments in the area of mixed-mode simulation, as well as sustained motivation of the need for such modeling. In the previous section, two important applications for mixed-mode have been illustrated.

Namely, both power device and OEIC modeling are cases where the interactions of complex device physics and circuit boundary conditions lead to requirements that mandate the use of mixed-mode techniques.

During this contract, our efforts have focused on creating a loosely coupled mixed mode capability that utilizes standard versions of PISCES and SPICE with modular changes to each of the respective codes. This has required not only careful understanding of the internal algorithms used at each level but how to interface and control the coupled operation of the two tools. The boundary conditions and developing a consistent representation, especially on the device analysis side has been a major challenge. On the circuit analysis side, the use of SPICE 3 has been of major benefit owing to the improved code structure and documentation. Finally, there has been an effort to parallelize the mixed-mode capabilities in the context of PVM, running on both networked workstations and the IBM SP/1. The potential for major speeding factors has been demonstrated based on results for ring oscillator circuits and six transistor SRAMS.

The mixed-mode aspect of this project received AASERT funding beyond the present ARO contract and this research will be continued in support of OEIC applications. In fact, the conclusions of this report outline both the major findings of this project and the key opportunities for further research, including technology transfer to industry.

CONCLUSIONS AND RECOMMENDATIONS

The development of physical models and algorithms for multi-layer, multi-material device simulation has resulted in major accomplishments in all aspects of the project. Moreover, there has been ongoing as well as new opportunities for technology transfer and industrial impact. The highlights of this work include:

- *Implementation of the 2ET model in PISCES and demonstration for both advanced silicon and compound material heterojunction devices.*
- *Application of the advanced PISCES 2ET modeling capabilities to:*
 - *GaAs MESFET sidegating with quantitative experimental confirmation,*
 - *Laser diode modeling in support of Hewlett-Packard applications.*
- *Algorithmic advances in support of ac high frequency analysis with mixed-mode boundary conditions.*

In addition to the direct output of this contract, there have been a set of spin-off results from previous ARO contracts that deserve special comment. In the area of silicon devices, there are ongoing developments and applications supported by SRC. For example, advanced MOS mobility model development and characterization of ESD are two primary focus areas at present under the SRC contract. The work on mixed-mode simulation has received support from both ARO and SRC. With the ending of this contract, the underlying support associated with the AASERT Fellowship for Francis Rotella now comes within the scope of the SRC contract on PISCES. Table 1 gives a summary of the various staff members funded over this contract and their employment status, including those still involved in graduate education.

The ongoing work on parallel algorithms and their demonstration in the context of the HPC initiative now falls under the support of ARPA contracts. Over this contract period, there have been several demonstrations of major significance that have drawn directly on efforts started under previous ARO contracts. Specifically, the benchmarking of STRIDE on the Caltech "Delta" machine with 520 processors demonstrated simulation results for a bipolar transistor with nearly 5 million grid---probably the world's largest device simulation. The parallelization of PISCES has also resulted in important benchmark results on both Intel (iPSC/860) and TMC (CM/5) machines with up to 64 processors. Ongoing work in this area is in fact expected to have a direct impact on commercial application codes both in the TCAD area as well as other engineering applications that use direct solvers.

There are several aspects of the present work that deserve special attention and are attractive for follow-on efforts. The application domain of OEIC device and circuit modeling is considered to be the prime candidate for further research effort. While there have been very promising results in both the transport modeling of the devices as well as in supporting mixed-mode simulations demonstrated during this contract period, there are a number of major advances still required in order to support full prototype applications. First, the quantum confinement/transport, including detailed consideration of the role of phonons in the equilibration process, requires the development of a physically consistent formulation. Second, the complete linking of the electrical and optical effects needs further investigation as well. In support of this system-level modeling there are associated needs for better framework integration tools and methodology. Finally, the overall issue of packaging of OEICs offers a very fertile ground for new efforts. Since nearly half the system cost of OEICs comes from the packaging, there is a real opportunity to impact both performance and cost issues for this key new technology.

There are certainly other opportunities to further advance the TCAD support for advanced device technologies. The area of wireless communications is now of major commercial interest and has ongoing impact on defense systems. The present contract has resulted in key demonstrations of improved parasitic modeling and ac high frequency characterization. The opportunities to further develop mixed-mode TCAD for high frequency capabilities in support of MMIC applications including both silicon- and GaAs-based technologies is thus strongly recommended. In fact, some of the demonstrations of new physical modeling capabilities for GaAs (i.e. deep level trap modeling) now open the way to more completely characterize the analog/digital trade-offs for this relatively mature but often troublesome technology base. On the silicon side, the recent advances in SOI technology appears to offer great promise for further advances into the microwave operating regime. Yet the trade-offs in parasitic effects and overall technology integration path need to be further scrutinized. The range of tools and techniques demonstrated in this contract indeed provide the cornerstones for further application and development in the area of MMIC analysis and design.

References:

- [1] D. Chen, E. Kan, U. Ravaioli, C-W. Shu, and R. W. Dutton, "An Improved Energy Transport Model Including Nonparabolicity and Non-Maxwellian Distribution Effects," *IEEE Electron Device Letters*, vol. 13, no. 1, pp. 26-28, Jan. 1992.

- [2] L. So, D. Chen, Z. Yu, and R. W. Dutton, "Dual Energy Transport Model for Coupled Lattice and Carrier Systems," *TECHCON'93 Proceedings*, Sept. 1993.
- [3] L. So, D. Chen, Z. Yu, and R. W. Dutton, "Robust Simulations of GaAs Devices Using Energy Transport Model," *Proceedings of 1993 Int'l Workshop on VLSI Process and Device Modeling (VPAD 1993)*, May 1993.
- [4] Z. Yu, L. So, E. Kan, and R. W. Dutton, "Dual Energy Transport Model for Advanced Device Simulation," *Proceedings of the International Workshop on Computational Electronics (IWCE'94)*, May 1994.
- [5] Z. Yu, D. Chen, L. So, and R. W. Dutton, "PISCES-2ET -- Two Dimensional Device Simulation for Silicon and Heterostructures," Stanford University, August 1994.
- [6] Y. Liu, Z. Yu, R. W. Dutton, and M. Deal, "Accurate Modeling of GaAs MESFET Sidegating Effects by Trapping Simulation," submitted to IEDM'94.
- [7] R. W. Dutton and J. D. Plummer, "Tool Integration for Power Device Modeling Including 3D Aspects," *Power Semiconductor Devices and Circuits*, Asea Brown Boveri Symposia Series, Edited by A.A. Jaecklin.
- [8] K-C. Wu, Z. Yu, L. So, R. W. Dutton, and J. Sato-Iwanaga, "Robust and Efficient ac Analysis of High Speed Devices," *Proceedings of International Electron Devices Meeting (IEDM 1992)*, pp. 935-938.
- [9] Y. Liu, "A Study of Sidegating Effects of Gallium Arsenide MESFETs: Experiments and Simulations," Stanford Ph.D. Thesis, September 1994.
- [10] R. W. Dutton, F. Rotella, Z. Sahul, L. So, and Z. Yu, "Integrated TCAD for OEIC Applications," *Proceedings of Lasers and Applications / Optoelectronics for Information and Microwave Systems (OE/LASE 1994)*, SPIE Workshop, January 1994.
- [11] R. W. Dutton, "Algorithms and TCAD Software Using Parallel Computation," *Proceedings of 1993 International Workshop on VLSI Process and Device Modeling (VPAD 1993)*, May 1993.
- [12] B. Herndon, A. Raefsky, R. Goossens, and R. W. Dutton, "PISCES MP--Adaptation of a Dusty Deck for Multiprocessing," *Proceedings of NASECODE-VII*, May 1992.
- [13] K-C. Wu, G. Chin, and R. W. Dutton, "A STRIDE Towards Practical 3-D Device Simulation--Numerical and Visualization Considerations," *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 9, pp. 1132-1140, September 1991.
- [14] R. Lucas, "Solving Planar Systems of Equations on Distributed Memory Multiprocessors," Stanford Ph.D. Thesis, December 1987.
- [15] B. Herndon, A. Raefsky, R. Goossens, and R. W. Dutton, "A Methodology for Parallelizing PDE Solvers: Applications to PISCES," submitted to the *Journal of Computer and Software Engineering, Special Issue on Parallel Computing*.
- [16] K-C. Wu, "Experience of Running Large-Scale 3-D Semiconductor Simulation on the Delta," *Proceedings of the First Intel Delta Applications Workshop*, pp. 313-319, February 1992.
- [17] Z-Y. Wang, K-C. Wu, and R. W. Dutton, "An Approach to Construct Pre-Conditioning Matrices for Block Iteration of Linear Equations," *IEEE Transactions on Computer-Aided Design*, vol. 11, no. 11, pp. 1334-1343, November 1992.
- [18] D.R. Apte, and M.E. Law, *IEEE Trans. on CAD*, vol. 11, no. 5, pp. 671-673, 1992.

Table 1: Staff/Students Supported Under Contract DAAL03-91-G-0152

Staff / Student Name	Date Degrees Granted	Educational / Employment Status (degree, funding / company, location)
Greg Anderson	Ph.D., pending	Intel, Oregon
Stephen Beebe	MS, January 1992	Ph.D. student (SRC funded)
Datong Chen	Postdoc	National Semiconductor, California
Goodwin Chin	Ph.D., May 1992	IBM, New York
Edwin Kan	Research Associate	New ARPA Contract
Aon Mujtaba	MS, June 1993	Ph.D. candidate (SRC funded)
Francis Rotella	MS, June 1993	Ph.D. candidate (AASERT funded)
Zak Sahul	MS, June 1992	Ph.D. candidate (SRC funded)
Lydia So	Postdoc	HP/LANL CRADA, Palo Alto
Hui Wang	Visiting Professor	Tsinghua University, Beijing
Zhiping Yu	Sr. Research Asso.	SRC and NSF contracts

An Improved Energy Transport Model Including Nonparabolicity and Non-Maxwellian Distribution Effects ^[1]

Datong Chen, *Member, IEEE*, Edwin C. Kan, *Student Member, IEEE*, Umberto Ravaioli, *Member, IEEE*, Chi-Wang Shu, and Robert W. Dutton, *Fellow, IEEE*

Abstract—An improved energy transport model for device simulation is derived from the zeroth and second moments of the Boltzmann transport equation (BTE) and from the presumed functional form of the even part of the carrier distribution in momentum space. Energy-band nonparabolicity and non-Maxwellian distribution effects are included to first order. The model is amenable to an efficient self-consistent discretization taking advantage of the similarity between current and energy flow equations. Numerical results for ballistic diodes and MOSFET's are presented. Typical spurious velocity overshoot spikes, obtained in conventional hydrodynamic (HD) simulations of ballistic diodes, are virtually eliminated.

THE drift-diffusion (DD) model, which assumes a local equilibrium of mean carrier energy, is usually not accurate enough for submicrometer device modeling, owing to the rapidly changing electric fields (see, for example, [1], [2] and references therein). The conventional hydrodynamic (HD) model, which also incorporates momentum and energy balance equations, and in the more complete form even includes the convective term for the carrier velocity, is known to yield unphysical solutions in some cases, e.g., the spurious velocity overshoot spike in n^+-n-n^+ ballistic diodes. Numerical stability is also a difficult problem, due to the hyperbolic nature of the model equations. An improved energy transport (ET) model, which overcomes the physical limitations and preserves the excellent numerical stability and efficiency of the DD model, is presented in this paper as an alternative to more computationally expensive methods, such as the Monte Carlo (MC) model.

Manuscript received October 4, 1991. This work was supported by the Joint Services Electronics Program under Grant N0001490-J-1270 (U. Ravaioli), the National Science Foundation through the National Center for Computational Electronics (NCCE) under Grant NSF ECS 88-09023 (E. C. Kan and U. Ravaioli), the Semiconductor Research Corporation under Contract 90-DJ116 (D. Chen and R. W. Dutton), and by the Army Research Office under Grant DAAL03-91-G-0123 (C.-W. Shu).

D. Chen and R. W. Dutton are with the Center for Integrated Systems, Stanford University, Stanford, CA 94305.

E. C. Kan and U. Ravaioli are with the Beckman Institute and the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

C.-W. Shu is with the Division of Applied Mathematics, Brown University, Providence, RI 02912.

IEEE Log Number 9105370.

For electrons, the zeroth and second-order moments of the steady-state Boltzmann transport equation (BTE) give the carrier continuity and energy balance equations:

$$\nabla \cdot J = -q(G - R) \quad (1)$$

$$\nabla \cdot S = F \cdot J - n \left(\frac{\partial E}{\partial t} \right)_{\text{coll}} \quad (2)$$

where the current J and the energy flow S are defined by $J = -qn\langle v \rangle = -q \int d^3k v f$ and $S = n\langle Ev \rangle = \int d^3k E v f$. Here, $G - R$ stands for the carrier generation-recombination rate, F is the electric field, q is the elementary charge, n is the carrier concentration, v is the carrier velocity, k is the crystal momentum, f is the carrier distribution function, and E is the carrier energy. The angular brackets denote averages over the distribution function, i.e., $\langle A \rangle \triangleq \int d^3k A f / \int d^3k f$. The last term in (2) is the energy loss rate due to collisions. By decomposing f into its even and odd parts, f_0 and f_1 , and using the microscopic relaxation time approximation $(\partial f_1 / \partial t)_{\text{coll}} = -f_1 / \tau$ [3], the odd part of the BTE becomes

$$f_1 = \frac{q\tau F}{h} \nabla_k f_0 - \tau v \nabla_r f_0. \quad (3)$$

Substitution of (3) into the expressions for J and S yields

$$J = -q \int d^3k v f_1 = q(n\hat{\mu}F + \nabla \cdot (n\hat{D})) \quad (4)$$

$$S = \int d^3k E v f_1 = -n\hat{\mu}^E F - \nabla \cdot (n\hat{D}^E) \quad (5)$$

where all the transport coefficients $\hat{\mu}$, $\hat{D}$, $\hat{\mu}^E$, and $\hat{D}^E$ are tensors and their components are defined by

$$\mu_{i,j} = \frac{q}{h} \left\langle \tau v_i \frac{\partial f_0}{\partial k_j} \right\rangle_0, \quad D_{i,j} = \langle \tau v_i v_j \rangle_0 \quad (6)$$

$$\mu_{i,j}^E = \frac{q}{h} \left\langle \tau E v_i \frac{\partial f_0}{\partial k_j} \right\rangle_0, \quad D_{i,j}^E = \langle \tau E v_i v_j \rangle_0. \quad (7)$$

Here $\langle \rangle_0$ denotes average over f_0 . The divergence of the tensors in (4) and (5) means to take the divergence of each

row of the tensor as a vector element. It is worth noticing that the constraint $f_1 \ll f_0$, which is assumed in the Legendre polynomial expansion approach in [3] and the perturbed distribution approach in [4], is not necessary in this model.

For materials like Si, where the mean kinetic energy $m\langle v \rangle^2/2$ is much smaller than the mean thermal energy $m\langle v^2 \rangle/2$, the correction from the field-induced anisotropy is generally not important [5]. Therefore, f_0 can be approximated by an isotropic distribution $f_0(E)$ and $\hat{\mu}$, $\hat{D}$, $\hat{\mu}^E$, and $\hat{D}^E$ become scalar functions of the mean carrier energy $\langle E \rangle$. We assume in the following $\tau = \tau(E) \propto E^{-p}$, where p is determined by the dominant scattering mechanism and ranges from 0.5 to 1 according to our preliminary bulk MC simulations. Additional hot-carrier effects are included by incorporating the following first-order approximations: a) non-parabolic bands: $\hbar^2 k^2/2m^* = E(1 + \alpha E)$; and b) non-Maxwellian effects: $f_0 = (1 + \gamma E/k_B T_e) f_m(E)$. Here, $f_m(E)$ is the Maxwellian distribution at an elevated temperature T_e and γ is the non-Maxwellian parameter. With these assumptions and from (6) and (7), the modified Einstein relations are valid not only for μ and D but also for μ^E and D^E , i.e., $D = k_B T_m \mu / q$ and $D^E = k_B T_m \mu^E / q$, where the equivalent carrier temperature $T_m = (1 + \gamma) T_e$ is related to the mean carrier energy by

$$\begin{aligned} \langle E \rangle &= \left(1 + \gamma + \frac{5}{2} \alpha k_B T_e \right) \frac{3}{2} k_B T_e \\ &= \left(1 + \frac{5}{2} \alpha k_B T_m \right) \frac{3}{2} k_B T_m. \end{aligned} \quad (8)$$

Then, we can recast (4) and (5) into

$$J = q(-\mu n \nabla \psi + k_B \nabla(n \mu T_m)) \quad (9)$$

$$S = -C_e \left(-\mu n k_B T_m \nabla \psi + \frac{k_B^2}{q} \nabla(n \mu T_m^2) \right) \quad (10)$$

where ψ is the electrical potential and $C_e \triangleq \mu^E/(k_B T_m \mu) = (5/2 - p)(1 - \alpha k_B T_m/2)$ is the energy flow factor C_e . C_e is in general a smoothly varying function of the mean carrier energy and for nearly parabolic bands $C_e \approx 5/2 - p$. It is important to notice that in our approach the Wiedmann-Franz law for heat flow is *never* invoked, as it is usually necessary in the conventional HD method to obtain a closure of the moment equations.

Based on the similarity of the functional expressions in (9) and (10), an extended Scharfetter-Gummel method [6] was developed for self-consistent discretization of both the carrier continuity equation and the energy balance equation. By assuming that along each mesh line $\bar{J}$, $\bar{S}$, and $(1/T)(d\psi/dl)$ are constant, we can solve 1-D forms of both (9) and (10) on the mesh lines and perform the standard box integration around each node [7]. With the following normalized notation: $\bar{J} = J/q$, $\bar{S} = -S/qC_e$, $\bar{N} = n\mu$, $\bar{T} = k_B T_m/q$, (9) and (10) are evaluated on the mesh line connecting the nodes i and j by

$$\bar{J}_i = \frac{1}{L_{ij}} (B(\bar{u}_{ij}) \bar{N}_j \bar{T}_j - B(-\bar{u}_{ij}) \bar{N}_i \bar{T}_i) \quad (11)$$

$$\bar{S}_i = \frac{1}{L_{ij}} (B(\bar{u}_{ij}) \bar{N}_j \bar{T}_j^2 - B(-\bar{u}_{ij}) \bar{N}_i \bar{T}_i^2). \quad (12)$$

L_{ij} is the distance between nodes i and j , $B(x) = x/\exp(x) - 1$ is the Bernoulli function, and $\bar{u}_{ij} = (\psi_j - \psi_i)/\bar{T}$, with $\bar{T} \triangleq \int_{x_i}^{x_j} \bar{T} dx / L_{ij}$. Usually, the carrier temperature varies much more slowly than the potential does. Therefore, the linear approximation $\bar{T} = (\bar{T}_i + \bar{T}_j)/2$ is reasonable in practice. This also implies a quadratic approximation in space for the potential according to the above discretization assumptions.

Since $F = -\nabla \psi$, by substituting $F \cdot J$ with $-\nabla \cdot (\psi J) + \psi \nabla \cdot J$ and employing a macroscopic relaxation time approximation for $\langle (\partial E / \partial t)_{coll} \rangle$, we can rewrite (2) as

$$\nabla \cdot H = \psi \cdot (G - R) - n \frac{\langle E \rangle - E_0}{\tau_e} \quad (13)$$

where $E_0 = 3k_B T_0/2$, with T_0 being the lattice temperature, τ_e is the energy relaxation time, and $H = S + \psi J$ is the total energy flow, consisting of both the thermal energy flow S and the electrical flow ψJ . From (11) and (12), and using $\bar{\psi} = (\psi_i + \psi_j)/2$, we can express the total energy flow on the mesh line as

$$\begin{aligned} H &= q(\psi \bar{J} - C_e \bar{S}_i) \\ &= \frac{q}{L_{ij}} (B(\bar{u}_{ij}) \bar{N}_j \bar{T}_j (\bar{\psi} - C_e \bar{T}_j) \\ &\quad - B(-\bar{u}_{ij}) \bar{N}_i \bar{T}_i (\bar{\psi} - C_e \bar{T}_i)). \end{aligned} \quad (14)$$

In this improved ET model, the necessary transport parameters $\mu(T_m)$, $C_e(T_m)$, and $\tau_e(T_m)$ can all be determined, for instance, from bulk MC simulations by evaluating the following ensemble averages:

$$\mu = -\frac{\langle v \rangle}{F}, \quad C_e = \frac{3}{2} \frac{\langle E v \rangle}{\langle E \rangle \langle v \rangle}, \quad \tau_e = -\frac{\langle E \rangle - E_0}{q F \langle v \rangle}. \quad (15)$$

We have tested the model by simulating typical Si n^+n-n^+ ballistic diodes and MOSFET's. For simplicity, we assume $\mu(T_m) = \mu_0 T_0 / T_m$, $C_e = 3/2$, and $\tau_e = 0.4$ ps. Also, we have used the modified Caughey-Thomas model [7] for the low-field mobility μ_0 and the surface scattering model given in [8]. A full Newton method is employed and quadratic convergence can be reached within 7-8 iterations in all cases, from a good initial guess. A comparison of results for the drift velocity in an n^+n-n^+ structure, using different models, is given in Fig. 1. It is clear that the ET model can describe velocity overshoot with reasonable accuracy, when compared to the Monte Carlo result. The spurious velocity overshoot spike at the anode junction, which appears in solutions with the HD models [9], is virtually absent in our results. We believe that the momentum relaxation time approximation, used in the formulation of the HD model but not invoked by the ET model, is responsible for the spurious velocity peak. For a detailed explanation, the reader is referred to a forthcoming publication [10].

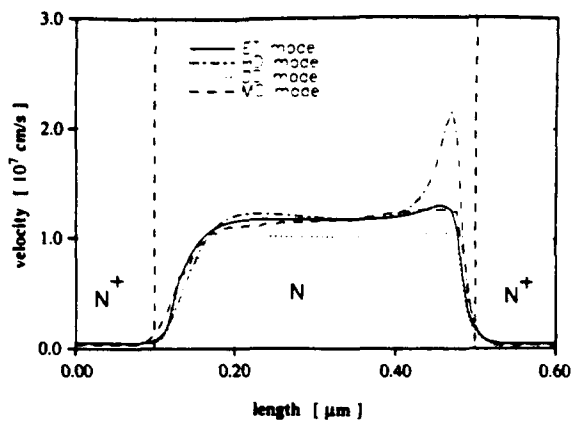


Fig. 1. Drift velocity profiles for a Si $n^+n\text{-}n^+$ structure, with abrupt junctions, $n^+ = 5 \times 10^{17} \text{ cm}^{-3}$, and $n = 2 \times 10^{15} \text{ cm}^{-3}$. The length of the n -region is $0.4 \mu\text{m}$ and bias applied to the collector n^+ region on the right is 1.5 V . Results are obtained using the energy transport (solid line), drift-diffusion (dotted line), hydrodynamic (dot-dashed line) and Monte Carlo (dashed line) models.

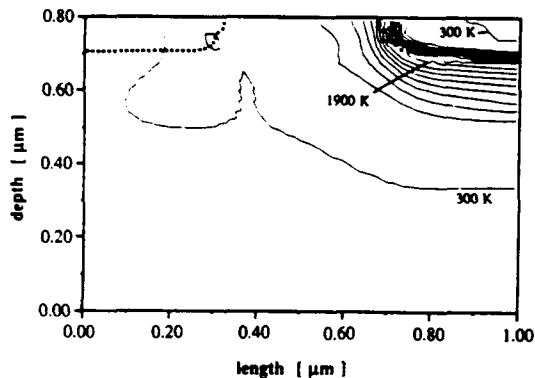


Fig. 2. Electron temperature (T_m) contours obtained for a typical $0.4\text{-}\mu\text{m}$ implanted channel n -MOSFET, solved with the ET model on a 40×40 uniform rectangular grid. Maximum doping in the implanted channel is $2 \times 10^{17} \text{ cm}^{-3}$. The applied biases are $V_{GS} = 2.0 \text{ V}$, $V_{SB} = 0.0 \text{ V}$, and $V_{DS} = 2.0 \text{ V}$. The step between temperature contour lines is 200 K . The bold dotted lines indicate the position of the source (left) and drain (right) junctions.

To further test the new model and the new discretization scheme in 2-D, a typical $0.4\text{-}\mu\text{m}$ implanted channel n -MOSFET has been simulated. The model was implemented in the PISCES II simulation code. For simplicity, although this does not reflect a limitation of the approach, the discretization nodes are distributed on a uniform rectangular

pattern. In Fig. 2 we show the temperature contours for a 40×40 grid. The introduction of surface scattering in the model did not appear to alter the convergence properties of the scheme. The stability of the solution was still excellent when we simulated the same structure on a rather coarse 10×10 grid. This is a good indication that the discretization scheme is well-posed. Our results are very encouraging and show that this ET model should be suitable for the simulation of general 2-D structures.

ACKNOWLEDGMENT

The authors would like to thank Prof. K. Hess and Prof. T. Kerkhoven of the University of Illinois, Prof. J. W. Jerome of Northwestern University, and Dr. Z. Yu of Stanford University for many fruitful discussions.

REFERENCES

- [1] B. Meinerzhagen and W. L. Engl, "The influence of the thermal equilibrium approximation on the accuracy of classical two-dimensional numerical modeling of silicon submicrometer MOS transistors," *IEEE Trans. Electron Devices*, vol. 35, no. 5, pp. 689-697, May 1988.
- [2] P. A. Sandborn, A. Rao, and P. A. Blakey, "An assessment of approximate nonstationary charge transport models used for GaAs device modeling," *IEEE Trans. Electron Devices*, vol. 36, no. 7, pp. 1244-1253, July 1989.
- [3] R. Stratton, "Diffusion of hot and cold electrons in semiconductor barriers," *Phys. Rev.*, vol. 126, no. 6, pp. 2002-2014, June 1962.
- [4] C. C. McAndrew, E. L. Heasell, and K. Singhal, "A comprehensive transport model for semiconductor device simulation," *Semicond. Sci. Tech.*, vol. 2, pp. 643-648, 1987.
- [5] D. Chen, E. C. Kan, and K. Hess, "Field-induced anisotropic distribution functions and semiconductor transport equations with tensor-form coefficients," *J. Appl. Phys.*, vol. 68, no. 10, pp. 5360-5362, Nov. 1990.
- [6] D. Chen, E. C. Kan, U. Ravaioli, Z. Yu, and R. W. Dutton, "A self-consistent discretization scheme for current and energy transport equations," in *Proc. IV Int. Conf. Simulation of Semiconductor Devices and Processes (SISDEP)* (Zürich), Sept. 12-14, 1991, pp. 235-239.
- [7] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. Vienna: Springer, 1984, ch. 4.
- [8] H. Shin, A. Tasch, C. Maziar, and S. Banerjee, "A new approach to verify and derive a transverse-field-dependent mobility model for electrons in MOS inversion layers," *IEEE Trans. Electron Devices*, vol. 36, no. 6, pp. 1117-1123, June 1989.
- [9] E. Fatemi and J. Jerome, "Solution of the hydrodynamic device model using high-order nonoscillatory shock capturing algorithms," *IEEE Trans. Computer-Aided Design*, vol. 10, no. 2, pp. 232-244, Feb. 1991.
- [10] D. Chen *et al.*, "Analysis of spurious velocity overshoot in hydrodynamic simulations of ballistic diodes," submitted to NUPAD '92.

Dual Energy Transport Model for Advanced Device Simulation [2]

Zhiping Yu, Lydia So*, Edwin Kan, and Robert W. Dutton
Center for Integrated Systems, Stanford University, CA 94305

Abstract Dual energy transport (DUET) model in semiconductor devices including heterostructures has been developed to simulate the distribution of carrier and lattice temperatures in addition to profiles of the electrostatic potential and carrier concentrations. The modeling approach is in consistency with the conventional drift-diffusion (DD) model, making it easy to implement in the existing code. Carrier energy dependent mobility and impact ionization models have been examined and are used for simulation of various velocity overshoot and hot electron effects. Two simulation examples, one for submicron MOSFET and one for deep-submicron SOI, are presented through comparison with the measurement data to demonstrate the improvement of the new model over DD model in predicting the device characteristics for modern (submicron) structures.

DUET Transport Model DUET model uses six state variables – potential (ψ), electron and hole concentrations and temperatures (n , p , T_n , and T_p), and lattice temperature (T_L) to describe the status of a semiconductor device. Except of Shockley semiconductor equations for ψ , n , and p and thermal diffusion equation for T_L , the governing equations for carrier temperatures are derived from the energy balance principle which in the form of the Fick's second law can be written as follows:

$$\frac{\partial w_n}{\partial t} = -\nabla \cdot s_n + j_n \cdot \mathcal{E}_n - u_{wn} \quad (1)$$

where electrons are used as the example and w_n is the electron kinetic energy density, s_n is the energy flux, $\mathcal{E}_n$ is the electric field acting on electrons, and u_{wn} is the net energy loss rate. The other symbols have conventional meanings. The key issues in the model development are to obtain the expressions for carrier/energy fluxes in terms of the state variables and to identify the mechanisms for carrier energy gain and loss. In DUET the distribution function in the real and crystal momentum space is constructed from the Fermi-Dirac statistics for the quasi-thermal equilibrium using the relaxation time approximation [1], so all the transport coefficients and flux expressions can rigorously be derived. For example, the electron current density and energy flux are computed from the following expressions:

$$j_n = n\mu_n \nabla E_{Fn} + qn\mu_n Q_n \nabla T_n \quad (2)$$

$$s_n = -P_n T_n j_n - \kappa_n \nabla T_n \quad (3)$$

where Q , P , and κ are thermopower, thermoelectric power, and thermal conductivity, respectively, and all of which can be related to the mobility μ . The energy exchange mechanisms considered in the model include SRH, Auger, and radiative recombinations, and impact ionization. Together with proper boundary conditions and carrier energy dependent models for mobility and impact ionization, the model has been implemented in PISCES (version 2Et) [2] and the initial test of the code showed very encouraging results as presented below.

Simulation Examples The following two examples show that DD model is no longer accurate in predicting $I-V$ characteristics for submicron devices when the non-stationary phenomena such as the velocity overshoot and nonlocal field dependence of physical parameters such as the impact ionization rate become important. While both DD and DUET models provide good simulation results compared to the measurement for devices with relatively long channel length, DD model starts to break for output characteristics of SOI at $L_{eff} = 0.12 \mu$ (Fig. 1) and substrate current of MOSFET at $L_{eff} = 0.8 \mu$ (Fig. 2). On the other hand DUET can consistently model the device characteristics well even when the device size is scaled down to the deep submicron range. Although we are now still calibrating the model parameters for DUET in order to improve the simulation accuracy, the examples presented clearly indicate that the DUET modeling approach is justified.

Appendix Complete set of DUET equations and expressions for the energy dependent mobility and impact ionization rate.

References

- [1] R. Stratton, "Semiconductor current-flow equations (diffusion and degeneracy)," *IEEE Trans. Electron Devices*, vol. ED-19, pp. 1288-1292, Dec. 1972.
- [2] Z. Yu, D. Chen, L. So, and R. W. Dutton, "PISCES-2ET document – models and user's manual," *Technical Rep.*, Stanford University, to be printed in March 1994.

*also with Los Alamos National Lab.

Appendix

Poisson's equation

$$\nabla \cdot (-\epsilon \nabla \psi) = q(p - n + N_D^+ - N_A^-) \quad (1)$$

Carrier continuity equations:

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{j}_n - u \quad (2)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{j}_p - u \quad (3)$$

Energy balance equations for carriers:

$$\frac{\partial w_n}{\partial t} = -\nabla \cdot \mathbf{s}_n + \mathbf{j}_n \cdot \mathbf{E}_n - u_{wn}$$

$$\frac{\partial w_p}{\partial t} = -\nabla \cdot \mathbf{s}_p + \mathbf{j}_p \cdot \mathbf{E}_p - u_{wp}$$

where

$$\begin{aligned} u_{wn} &= (u_{srh} + u_{rad}) \frac{3}{2} k_B T_n - \\ &\quad (u_{n, Auger} - g_{n, imp}) \left[E_g(T_L) + \frac{3}{2} k_B T_p \right] \\ &\quad - g_{p, imp} \frac{3}{2} k_B T_n + \frac{w_n(T_n) - w_n(T_L)}{\tau_{wn}} \\ u_{wp} &= (u_{srh} + u_{rad}) \frac{3}{2} k_B T_p - \\ &\quad (u_{p, Auger} - g_{p, imp}) \left[E_g(T_L) + \frac{3}{2} k_B T_n \right] \\ &\quad - g_{n, imp} \frac{3}{2} k_B T_p + \frac{w_p(T_p) - w_p(T_L)}{\tau_{wp}} \end{aligned}$$

Thermal diffusion equation for lattice:

$$\begin{aligned} c_L \frac{\partial T_L}{\partial t} &= \nabla \cdot (\kappa_L \nabla T_L) \\ &\quad + u_{srh} \left[\frac{3}{2} k_B T_n + E_g(T_L) + \frac{3}{2} k_B T_p \right] \\ &\quad + \frac{w_n(T_n) - w_n(T_L)}{\tau_{wn}} + \frac{w_p(T_p) - w_p(T_L)}{\tau_{wp}} \end{aligned}$$

Mobility (Haensch's) and impact ionization rate (Slotboom) models:

$$\begin{aligned} \mu(N, T_L, E_{\perp}, T_c) &= \frac{\mu_0(N, T_L, E_{\perp})}{1 + \gamma(N, T_L, E_{\perp})[w(T_c) - w(T_L)]} \\ \gamma(N, T_L, E_{\perp}) &= \frac{\mu_0(N, T_L, E_{\perp})}{q \tau_w v_{sat}^2(T_L)} \end{aligned}$$

$$\alpha = A \exp[-(b/\mathcal{E}_{eff})^m]$$

$$\mathcal{E}_{eff} = \frac{3}{2} \frac{k_B}{q} \frac{T_c - T_L}{\tau_w v_{sat}}$$

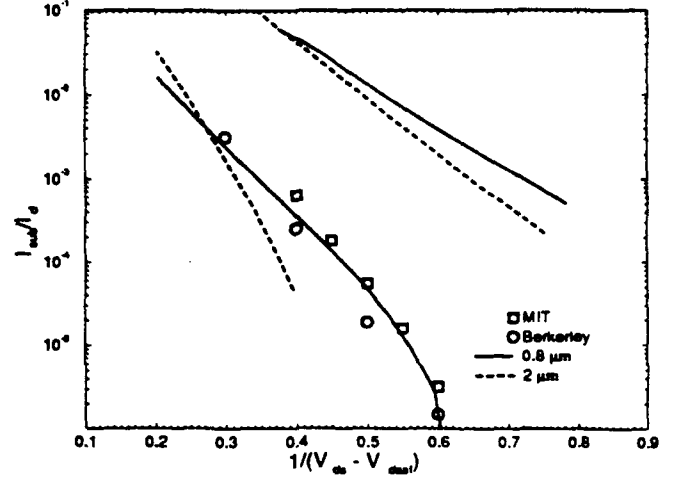


Figure 1: Simulation results for the substrate current in MOSFET with two different channel length (2 and 0.8 μm) and the comparison is made for 0.8 μm case between the ET-simulated and measured results (from MIT and UC Berkeley, respectively). The upper curves are simulated using DD model while the lower curves are obtained from ET simulation.

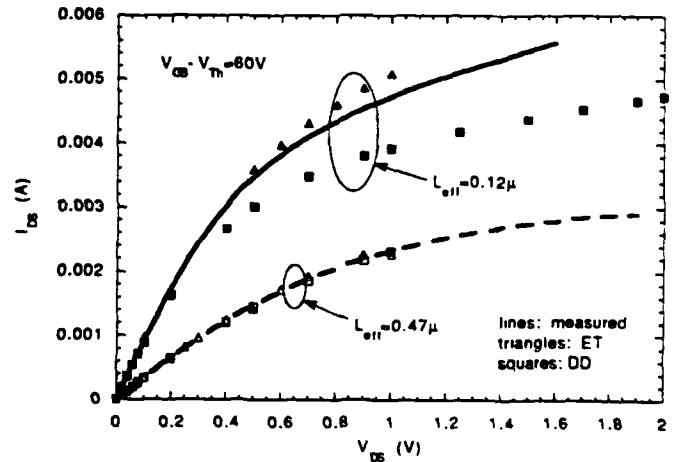


Figure 2: Simulated and measured data for SOI structure with different channel length.

Robust Simulation of GaAs Devices Using Energy Transport Model [3]

Lydia L. So, Datong Chen, Zhiping Yu, and Robert W. Dutton

Integrated Circuits Laboratory, Stanford University, Stanford, CA 94305

The carrier transport mechanism in GaAs which includes a negative differential mobility in the carrier velocity-field curve when implemented in the 2D device simulator, causes grid-dependent numerical instability and poor convergence behavior. It has well been established in the literature [1, 2] that the conventional drift-diffusion (DD) approximation which assumes a local thermal equilibrium among the same type of carriers (electrons or holes) has significant shortcomings in the description of GaAs device operation. In this paper we report an extension of the energy transport (ET) model [3, 4] in which carrier energy conservation equations are solved to GaAs device simulation. A technique whereby the empirical form of the field-dependent mobility is transformed to an energy-dependent one used in the ET model is presented.

The empirical formula describing the dependence of mobility on electric field implemented in the 2D device simulator PISCES [6] has the form $\mu(N_T, E) = \frac{\mu_o(N_T) + \frac{v_{sat}}{E} (\frac{E}{E_o})^4}{1 + (\frac{E}{E_o})^4}$, where μ_o is the low-field mobility, E is the magnitude of the electric field, E_o is the critical electric field, and v_{sat} is the saturation drift velocity. In applying the ET model to GaAs, a formulation describing the dependence of carrier velocity on local temperature needs to be determined. From the second-order moment of the Boltzmann transport equation, the energy conservation equation at steady state is given by $\nabla \cdot \mathbf{S} = \mathbf{E} \cdot \mathbf{J} - n \langle \frac{\partial E}{\partial t} |_{coll} \rangle$. The macroscopic quantities are: energy flux $\mathbf{S}$, electric field vector $\mathbf{E}$, electron current density $\mathbf{J}$, electron concentration n , and the average electron energy loss due to collision $\langle \frac{\partial E}{\partial t} |_{coll} \rangle$. Since the empirical formula for $\mu(\mathbf{E})$ were obtained experimentally under uniform field condition where $\nabla \cdot \mathbf{S} = 0$ and $\mathbf{J} = qn\mu(E)\mathbf{E}$, by using the assumptions in that the average energy loss term $\langle \frac{\partial E}{\partial t} |_{coll} \rangle = \frac{3}{2} \frac{k(T_e - T_L)}{\tau_w}$, we arrive at a transcendental relation of local field and carrier energy,

$$\mu(E)E^2 = \frac{3}{2} \frac{k(T_e - T_L)}{q\tau_w}. \quad (1)$$

Here, k is the Boltzmann constant, τ_w † denotes the energy relaxation time, and T_e and T_L are the electron and lattice temperatures respectively. Hence, the carrier temperature can be solved for a given field strength, and the dependence of mobility on local carrier energy can be obtained.

During actual implementation in the device simulator PISCES, it is more convenient to have an analytical form in which the mobility dependence on local carrier temperature can be solved directly. At low fields we assume $J = qn\mu_o E$, hence

$$E = \sqrt{\frac{3}{2} \frac{kT_L}{q} \frac{(T_e/T_L - 1)}{\mu_o \tau_w}}; \quad \text{and} \quad \mu_L(T_e) = \frac{\mu_o + \frac{v_{sat}}{E_o} [\frac{3}{2} \frac{kT_L}{q\mu_o \tau_w E_o^2} (\frac{T_e}{T_L} - 1)]^{3/2}}{1 + [\frac{3}{2} \frac{kT_L}{q\mu_o \tau_w E_o^2} (\frac{T_e}{T_L} - 1)]^2}. \quad (2)$$

† Monte-Carlo Simulation results reported in Ref. [5] were used to extract the energy relaxation time, $\tau_w = 0.8 \times 10^{-12}$ second.

At high fields we make the assumption that $J = qnv_{sat}$, hence

$$E = \frac{3}{2} \frac{kT_L}{qv_{sat}\tau_w} \left(\frac{T_e}{T_L} - 1 \right); \quad \text{and} \quad \mu_H(T_e) = \frac{\mu_o + \frac{v_{sat}}{E_o} \left[\frac{3}{2} \frac{kT_L}{qv_{sat}\tau_w E_o} \left(\frac{T_e}{T_L} - 1 \right) \right]^3}{1 + \left[\frac{3}{2} \frac{kT_L}{qv_{sat}\tau_w E_o} \left(\frac{T_e}{T_L} - 1 \right) \right]^4} \quad (3)$$

The mobility relations can thus be formulated approximately as a weighted average of

$$\mu(N_T, T_e) = 0.5 \cdot (1 - \mathcal{F}) \cdot \mu_L(N_T, T_e) + 0.5 \cdot (1 + \mathcal{F}) \cdot \mu_H(N_T, T_e) \quad (4)$$

where $\mathcal{F} = \tanh[\alpha(T_e - T_c)]$ is a fitting function in which α and T_c are fitting parameters. The average carrier velocities as a function of carrier temperature, $v = \mu(T_e)E(T_e)$, by using the transcendental relation of Eq. (1) and fitting function of Eq. (4), are shown in Fig. 1.

We simulated a non-homogeneous GaAs $n^+ - n - n^+$ diode featuring a 5- μm n -region doped to $1 \times 10^{17} \text{ cm}^{-3}$ and 0.1- μm n^+ -region doped $5 \times 10^{19} \text{ cm}^{-3}$ at two ends. The corresponding current-voltage characteristics using different grids of uniform spacing are shown in Fig. 2. The DD model which uses the local field formulation tends to converge very slowly. It can be seen that near electric field strengths at which intervalley carrier transfer occurs and Gunn-domain forms, simulated currents display non-physical 'zig-zags', and their periodicity is extremely grid sensitive. Simulated net charge distribution for a 1- μm GaAs $n^+ - n - n^+$ diode at $V_{dd}=1.0$ volt is shown in Fig. 3.

We also simulated a GaAs MESFET structure featuring a 1 μm gate and 0.8 μm spacers. The drain-source current-voltage relationship is compared using both the DD and ET model in Fig. 4. The DD model exhibits similar non-physical 'zig-zags' in the drain current - an indication that the present grid density is insufficient to resolve the Gunn-domain. On the other hand, the ET model yields numerically stable solutions with excellent convergency rate without incurring an additional computation cost.

In summary, the energy transport formulation that accounts for the principle non-stationary effect has been applied to GaAs MESFET device simulation. The excellent convergence behavior of the ET model as demonstrated in the GaAs-based devices permits a wide application to III-V material based devices.

References

- [1] W. R. Curtice, and Y.-H. Yun, "A temperature model for the GaAs MESFET," *IEEE Tran. Electron Devices* vol. ED-28, no. 8, pp. 914-962, Aug. 1981.
- [2] C. M. Snowden, and D. Loret, "Two-dimensional hot electron models for short-gate length GaAs MESFETS" *IEEE Tran. Electron Devices* vol. ED-34, no. 2, pp. 212-223, Feb. 1987.
- [3] D. Chen, E. C. Kan, U. Ravaioli, C. Shu and R. W. Dutton, "An improved energy transport model including non-parabolic and non-Maxwellian distribution effects," *IEEE Elec. Dev. Lett.*, vol. 13, no. 1, pp. 26-28, Jan. 1992.
- [4] D. Chen, E. Sangiorgi, M. R. Pinto, E. C. Kan, U. Ravaioli and R. Dutton, "Analysis of spurious velocity overshoot in hydrodynamic simulations of ballistic diodes," *Proceeding of NUPAD V (Numerical Process and Device Modeling Workshop)*, Seattle, May 31 - June 1, 1992.
- [5] M. V. Fischetti, "Monte Carlo simulation of transport in technologically significant semiconductors of diamond and zinc-blende structures. Part I: Homogeneous Transport," *IEEE Trans. Electron Devices*, vol. 38, no. 3, pp. 634-649, Mar. 1991.
- [6] M. R. Pinto, C. S. Rafferty, and R. W. Dutton, *PISCES-II User's Manual*, Stanford University, CA, 1984.

Figures:

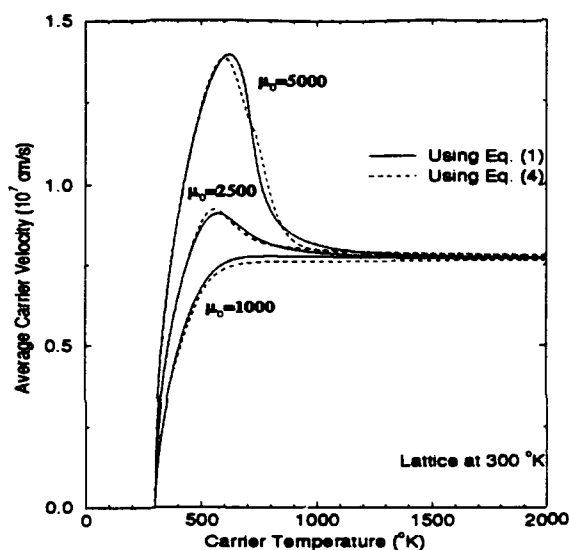


Fig. 1: GaAs average carrier drift velocity as a function of electron temperature for different low-field mobility values.

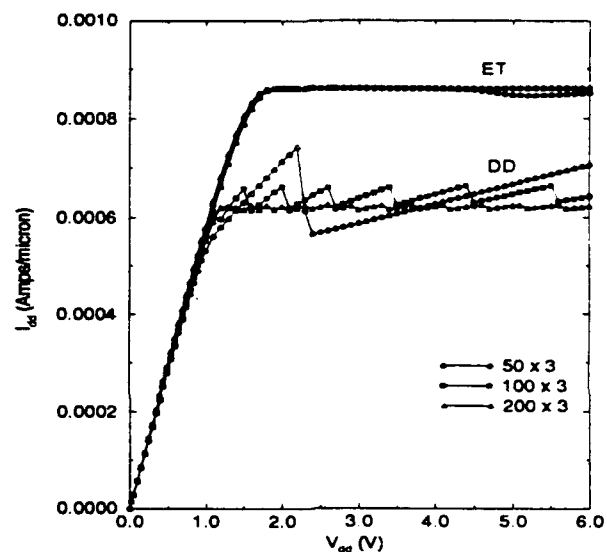


Fig. 2: Current-voltage characteristics of a 5- μ m GaAs $n^+ - n - n^+$ diode using different grids.

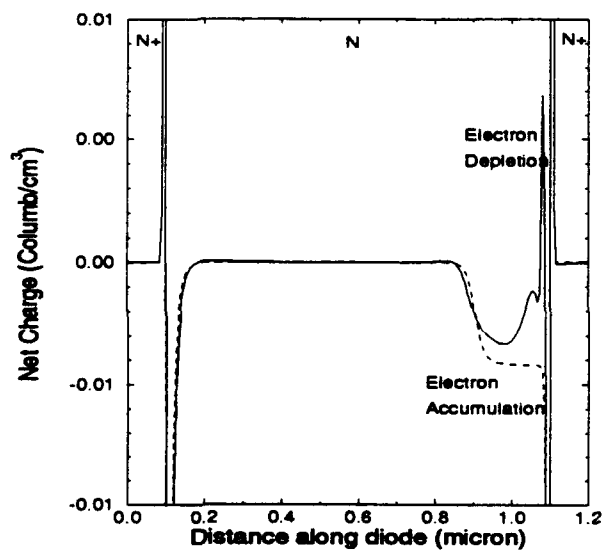


Fig. 3: Simulated net charge distribution along a 1 diode using a 200 $\times$ 3 uniform grid. — ET and - - - DD model.

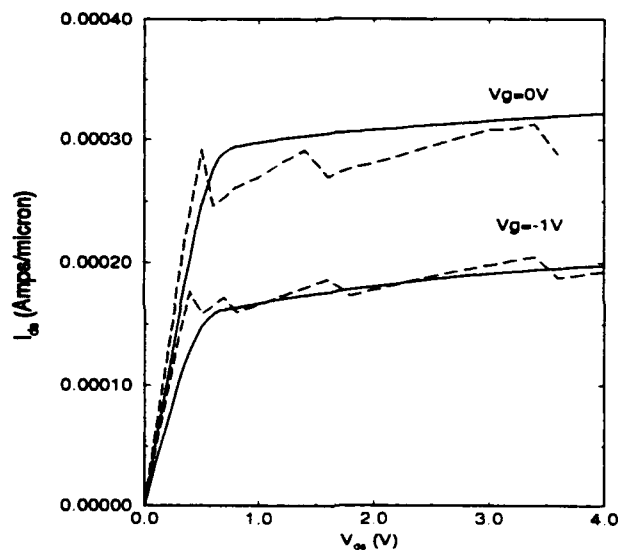


Fig. 4: Comparison of simulated drain current of GaAs MESFET using — ET model and - - - DD model.

Accurate Modeling of GaAs MESFET Sidegating Effects By Trapping Simulation ^[6]

Yi Liu, Zhiping Yu, Robert W. Dutton, and Michael D. Deal

AEL 204, IC Lab, Stanford University, CA 94305. Phone: (415) 725-3644

Quantum Electronics and Compound Semiconductor Devices

Abstract GaAs MESFETs play an increasingly critical role in digital mobile communications owing to their low power dissipation [1]. However, sidegating effects which are unique for devices built on semi-insulating (SI) GaAs substrates cause cross-talk of neighboring MESFETs, thus thwarting increased IC density. In this paper, a detailed analysis of carrier trapping processes in the substrate is conducted using an efficient numerical trapping model. Two major mechanisms are revealed to be responsible for sidegating phenomena. One is the formation of a Gunn domain at the channel/substrate interface. The other is the substrate conduction between a parasitic Schottky contact and sidegate due to carrier injection. For the first time, quantitative agreement between the measured and simulated device characteristics is obtained. Among key findings of this study is the confirmation of correct electron mobility model used in FET simulation. Design issues for improving circuit performance will also be discussed.

Test Structures and Measured Sidegating Characteristics The fabricated test structure for studying sidegating effects is schematically shown in Figure 1. The purpose of using an ungated device and a separate Schottky contact is to identify and isolate the effects of the parasitic Schottky contact. With the Schottky contact left floating, the measured I-V characteristics of the drain current vs. sidegate bias exhibit a distinct threshold behavior (Figure 2(a)). When the Schottky contact is grounded, the drain current reduction due to the sidegating effect is much more severe (Figure 2(b)).

Simulation and Explanation In order to analyze the carrier trapping effect on the device characteristics, a two-level numerical trapping model is implemented in device simulator PISCES [2]. The flowchart for this model is outlined in Figure 3. For dc analysis, a deep-impurity model which assumes a single quasi-Fermi level for each type of carrier in both band and trap level(s) is sufficient, while trap rate equation(s) must be included in basic semiconductor equations for ac and transient analyses. The trapping model is implemented in such a way that the number of independent variables is not increased in the solution process, thus keeping the memory requirements and code modifications to a minimum. Figure 4(a) shows the vertical structure for simulation. The simulated drain current decreases with the increasing negative sidegate voltage with a threshold behavior only when the negative-differential-mobility (NDM) model is used in the simulation (Figure 5). The simulation shows that a negative space charge region appears on the substrate side at the channel/substrate interface when the electric field in the substrate reaches a critical value (Figure 6), explaining the threshold behavior in Figure 2(a). The formation of the Gunn domain because of the NDM has previously been studied analytically by Böer for Gunn diodes in [3]. This negative space charge region is due to the enhanced trapping of electrons by deep-level electron-traps because of the electron accumulation in the Gunn domain on the substrate side. Electrons are depleted on the channel side, leading to a decrease in the drain current. The simulated drain current vs. the sidegate voltage using more realistic lateral structure in Figure 4(b) shows an excellent agreement with the measured data (Figure 2(a)). Figure 7 shows the simulation structures that include Schottky contacts. The charge distribution simulated using the vertical structure in Figure 7(a) shows that negative space charge is created in substrate region at the channel/substrate interface (Figure 8), which again is due to the enhanced trapping of electrons. But in this case, the electron trapping is a consequence of the process of electron-injection into the substrate from the sidegate and hole-injection from the Schottky contact; the electron concentration in the region at the channel/substrate interface is increased due to the electron-injection. The drain current reduction in this case is more severe because the on-set of the conduction in the Schottky-substrate-sidegate structure starts at a lower sidegate voltage value than that for the Gunn domain to form. The simulated drain current using the lateral structure in Figure 7(b) shows a good agreement with the measured data (Figure 2(b)).

Discussion of Design Issues Based on results from both experiments and simulations, direct Schottky contact on substrate should be avoided. Methods are being developed to reduce or eliminate the contact on the substrate in the overlapping portion of the gate contact of the MESFET. The supply voltage and spacing between devices must be optimized to reduce the chance of the Gunn domain formation.

References

- [1] K. Miyatsuji, *et al.*: ISSCC Digest, 34 (1994)
- [2] Z.Yu, D.Chen, L.So, R.W.Dutton, "PISCES-2ET document," *Tech. Rep.*, Stanford University (1994)
- [3] K.W.Böer, G.Döhler, *Phys. Rev.*, **186**, 793 (15 Oct. 1969)

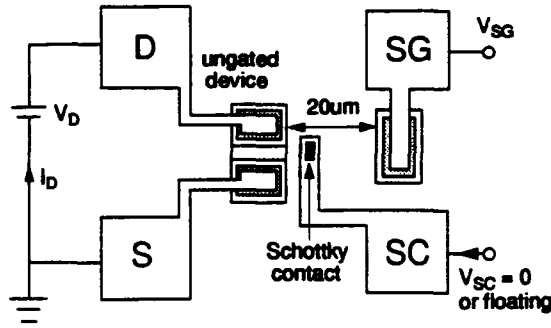


Figure 1. Test Structure for measurements.

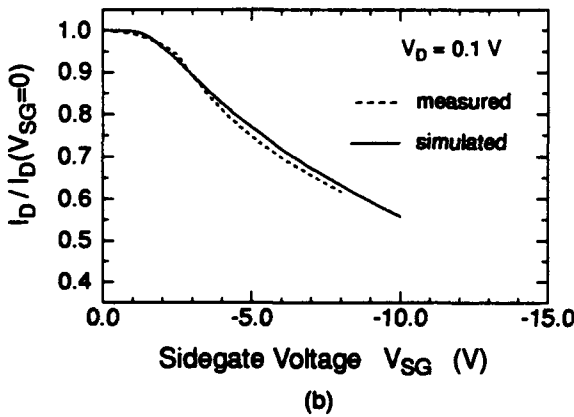
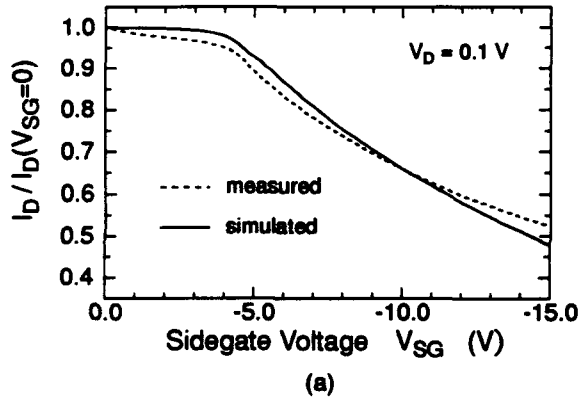


Figure 2. Measured and simulated sidegating effects due Gunn domain (a) and Schottky contact (b).

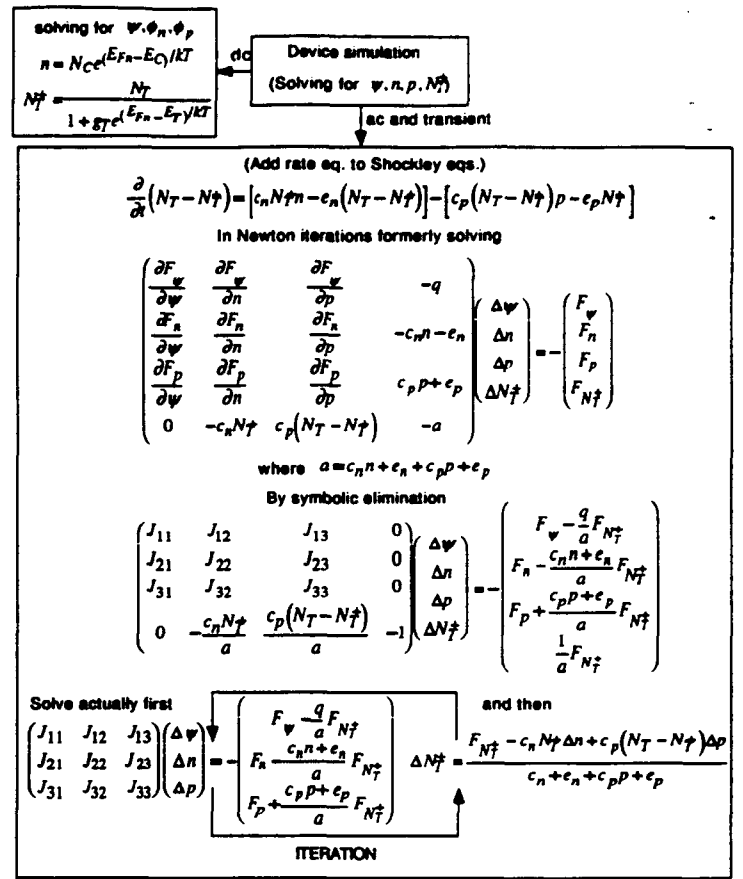


Figure 3. Flowchart for the trap model

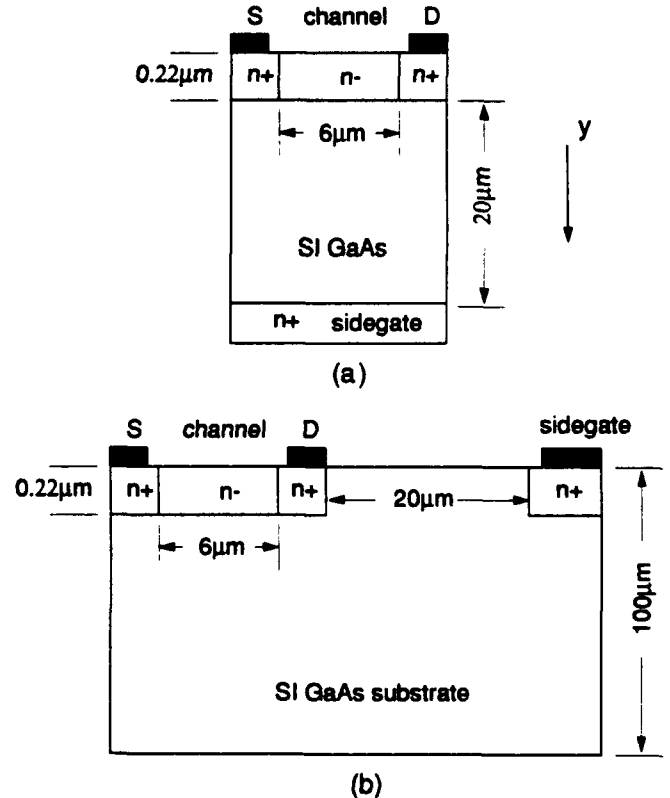


Figure 4. Vertical (a) and lateral (b) structures for simulations.

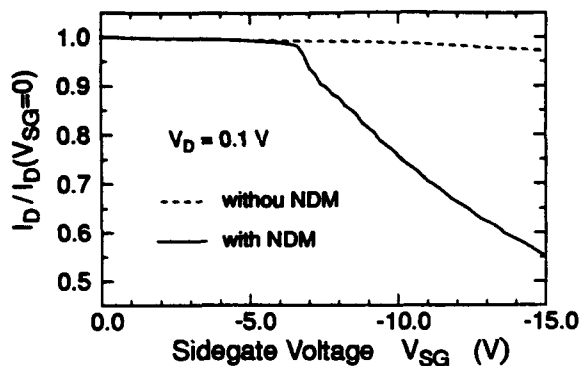


Figure 5. Simulation results for the vertical structure in Figure 4(a) using different mobility models

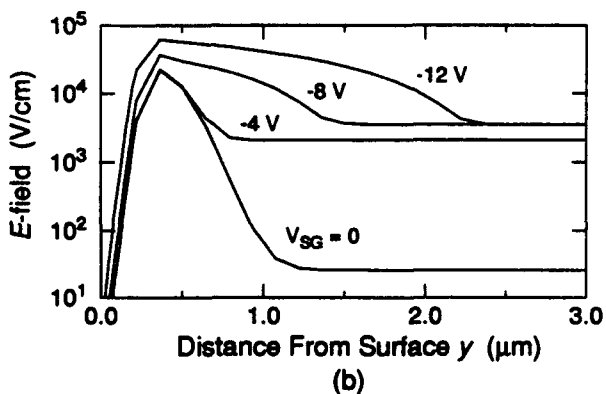
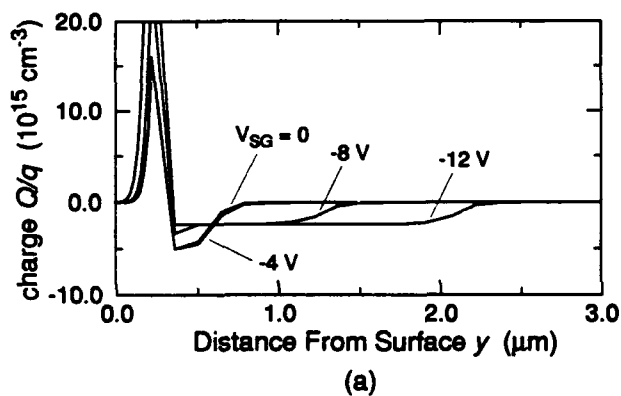


Figure 6. Charge (a) and field (b) distributions simulated using the vertical structure in Figure 4(a).

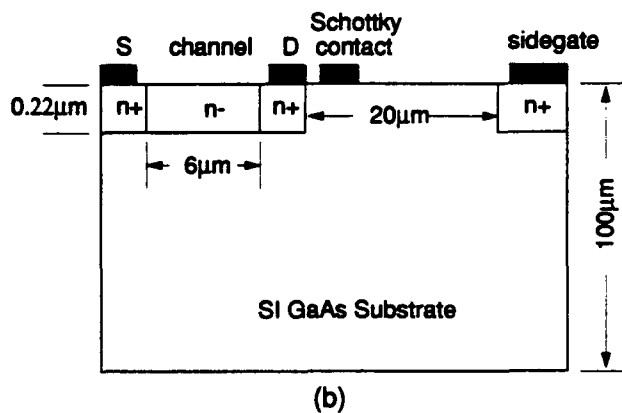
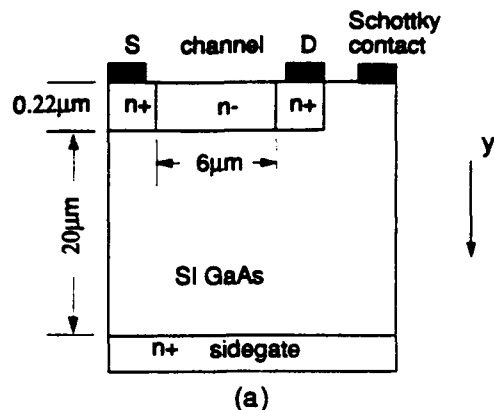


Figure 7. Vertical (a) and lateral (b) structures for simulations that include effects of the Schottky contact.

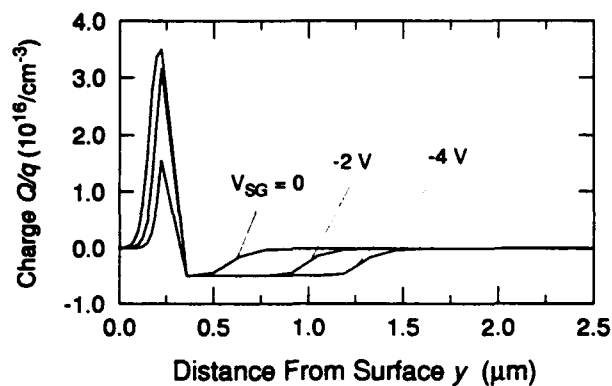


Figure 8. Charge distribution simulated using the vertical structure in Figure 7(a).

auspices of the International CAD Framework Initiative (CFI) is reviewed. The paper concludes with an evaluation of progress to date in integrated modeling of power devices and a look into the future of requirements such as full 3D modeling.

1. INTRODUCTION

The evolution of silicon Integrated Circuits (IC) technology over the past two decades has been truly remarkable. In the late sixties and early seventies, the bipolar IC dominated both digital and analog domains. Throughout the eighties, the Metal Oxide Semiconductor (MOS) technology seemed destined to dominate in virtually all applications except high speed analog. As we enter the nineties, a variety of bipolar technologies have re-emerged, both in high density applications such as Bipolar Complementary MOS (BiCMOS) and new high speed devices such as Heterojunction Bipolar Transistor (HBT) structures. Over this same time period, the landscape of technologies for high voltage and power IC applications has changed as well. Pure bipolar thyristor and Gate Turn-Off (GTO) discrete devices gave way to integrated digital/analog structures. In parallel with the emergence of high density MOS came Double Diffused MOS (DMOS), V-Groove MOS (VMOS) and other novel power devices and IC structures. Finally, the further development of merged structures has led to Insulated Gate Bipolar Transistors (IGBT) and other MOS gate-controlled bipolar structures. Clearly, the power device field has kept pace with and benefitted directly from the evolution of the IC industry as a whole.

From a circuits point of view, the power IC industry has paralleled the evolution of the analog IC industry. Starting with small-scale IC's, the level of chip complexity for power circuits tracked the development and complexity of Medium Scale Integration (MSI) bipolar IC's up to about the late seventies. The more recent developments in MOS-based power devices has opened an era of more complex digital circuitry. Hence, through the eighties we have seen a dramatic growth in functional integration involving advanced control logic as well as new gate-controlled power devices.

The above two themes of technology and circuit evolution for power devices provide an important background for this contribution. Namely, there is a long historical connection of power IC's with the bipolar world yet a steady shift towards and influence by MOS technology and circuit concepts as well. In looking at the needs for Technology CAD (TCAD) tools to address the diverse needs involved in modeling power devices, it is essential that we look at the technology and modeling requirements that come from each domain. In addition there are broader systems issues to be considered, for example thermal effects and the influence of packaging.

TOOL INTEGRATION FOR POWER DEVICE MODELING INCLUDING

3D ASPECTS [7]

Robert W. Dutton and James D. Plummer

Stanford University,
Stanford, CA, USA

Dedicated to Christiaan Abbas, July 22, 1956 - January 17, 1992

ABSTRACT

The past decade has witnessed an all explosive growth in Integrated Circuits (IC) capabilities, both for high density information processing applications and industrial purposes including sensors, actuators and power devices. The growing possibilities to create "smart" devices of unique capabilities for sensing and controlling diverse electro-mechanical (and optical) systems offer almost unlimited potential. Yet to realize these potentials, heterogeneous constraints must be quantified and used effectively in design. This talk will cover the specific issues of process, device and circuit modeling and the integration issues necessary to enable full exploitation of Technology Computer Aided Design (TCAD), tools for Power Device Modeling. The Process and Device Modeling discussions build on experience with the 2D tools SUPREM and PISCES developed at Stanford and quasi-3D applications for power devices. Both simulation and experimental results are used to illustrate key points. The growing importance of combined integrated circuit and power device applications are illustrated using several approaches to mixed mode simulation. Here the examples are broadened to include both university-based tools such as MEDUSA and SPICE/PISCES as well as commercial circuit/system level simulators. The final aspect of this contribution centers on a framework for tool integration across these various levels of design. Recent progress in establishing a TCAD framework under the

This paper is organized as follows. First, the early efforts in 1D TCAD are put in the context of DMOS technology development that occurred at Stanford in the seventies. Then recent trends in 2D process and device modeling and other power device structures are discussed, including selected work in quasi 3D simulations. Next the importance of mixed-mode (circuit/device) simulation is discussed along with examples from several groups working in this area. The concepts of TCAD framework are then presented; the growing importance of heterogeneous tool integration is emphasized. Finally, recent trends in 3D TCAD discussed.

2. EVOLUTION OF 1D PROCESS MODELING

Despite the rapid evolution of 2D and even 3D modeling tools for complex device structures, the 1D process simulator SUPREM 3 continues to be used broadly in industry. The reason for this persistent use, even after development of robust 2D tools such as SUPREM 4, goes to the base of problems involved in modeling power devices. Namely, the complexity of modeling bipolar device structures in terms of double-diffused profiles has continued to be a major research challenge, even after more than three decades of investigation. Moreover, the calibration of impurity diffusion effects, especially in the high concentration regime, is complex and more data is needed. Fortunately, a majority of the controlling device physics and critical parameters for power devices occur in the bulk regions as opposed to at the surfaces. Nonetheless, it is instructive to revisit our earliest efforts with DMOS process modeling to understand the underlying problems.

Shortly after the introduction of the DMOS structure¹, the Stanford group was actively developing DMOS high voltage multiplexing switches for ultrasonic imaging applications². As part of that development effort we considered the issues of threshold-voltage control in double diffused MOS structures³. Fig. 1 shows the results of theoretical and experimental variations in threshold voltage as a function of the boron predeposition parameter $\sqrt{Dt}$. The models exploited the typical analytical forms used prior to the development of SUPREM. In fact, it was out of this work that it became clear that more accurate process models were needed for devices such as DMOS. The first efforts in 1D process modeling (SUPREM) were discussed in the context of understanding the parameter variations of double diffused devices⁴. Despite encouraging progress in modeling the redistribution effects in double-diffused profiles, the characterization technology still lagged the theory. In an effort to quantify the device effects, the use of the Spreading Resistance Probe (SRP) was further developed and exploited for those early DMOS structures⁵. Fig. 2 shows typical results obtained using the SRP for DMOS de-

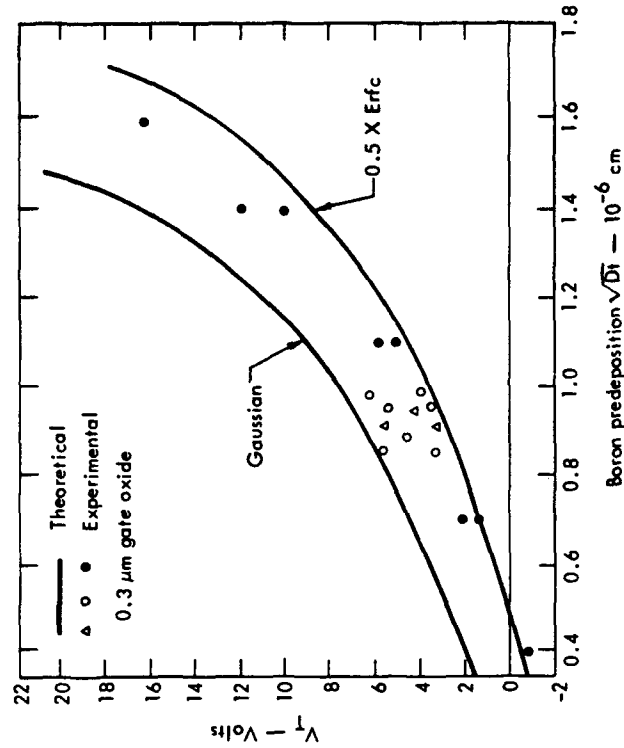


Fig. 1. Comparison of experimental and theoretical values of threshold voltage for DMOS devices with varying boron thermal cycle³.

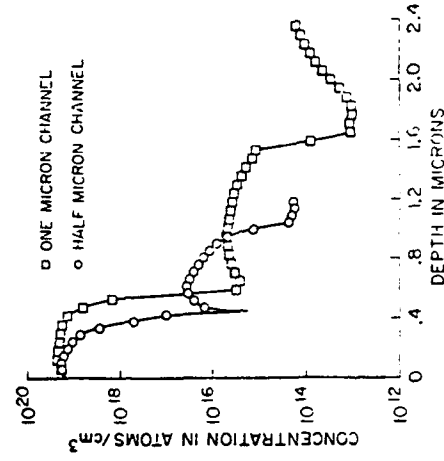


Fig. 2. Measured spreading resistance profiles for channel lengths of 0.5 μm and 1.0 μm ⁵.

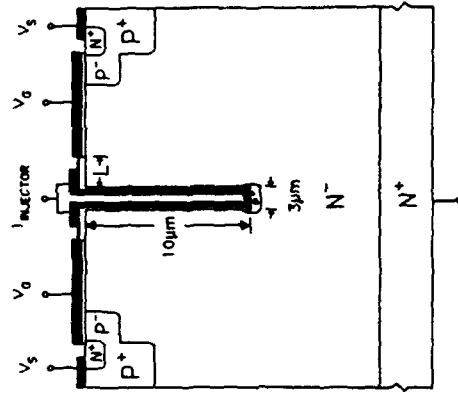


Fig. 3. Basic structure of the CMDMOS device¹⁰.

jector below the surface, it was anticipated that the minority carriers would originate from it and be distributed over a greater (or deeper) portion of the drift region¹⁰. During ON-state operation, a small hole current is introduced between the p+ and the source in the CMDMOS device. A controlled number of minority carriers are injected into the path of the DMOS electron current to modulate the conductivity of the epitaxial drift region. This causes the various components of the ON-resistance to be reduced significantly. Fig. 4 shows the effects of the current injected at the p+ diffusion on the hole concentrations in the CMDMOS. If a small amount of minority carrier current, 1.0×10^{-7} A/ μ m of hole current is injected from the p+ diffusion into the drift region of a CMDMOS device, then, according to Liu's analysis¹⁰, the minority carriers will set up a conductivity modulated area in the drift region. This is indeed confirmed by the PISCES simulation for a CMDMOS device operating under a drain bias of 0.5 Volt and a gate drive of 2.0 Volts, which is illustrated in Fig. 4, showing the simulated hole concentration contours of the device. From the figure it is seen that, contrary to assumptions made in first-order analyses, the holes are not distributed throughout the entire drift region. Instead, the holes have a limited distribution due to the drain bias on the substrate which tends to repel holes toward the upper portion of the drift region. This upper portion is considered to be the "modulated" region since the hole concentration in this region is high enough to significantly change the majority carrier concentration as

vices with channel length of 1 μ m and 0.5 μ m. Although many factors, such as probe calibration and surface preparation, can affect results significantly, it was based on the combined results of 1D SUPREM simulations and extensive use of SRP diffusion parameter extraction that subsequent versions of SUPREM evolved and demonstrated utility in the industry. However, as stated in the introduction to this section, the many difficulties of double-diffused bipolar and DMOS structures continue to complicate the development of truly robust and accurate 2D process models. One of the key challenges in the modeling of high concentration profiles is to understand the role of implant damage on both diffusion and impurity activation. Currently this is one of the most active areas of effort in process modeling and a physically based model is still under development⁶.

Having taken a brief journey back to the earliest 1D process modeling efforts with SUPREM and identified an ongoing problem of process characterization and the inherent difficulties of modeling double-diffused impurity profiles, now let us turn to the more recent efforts in 2D modeling. Here we find many significant successes, despite the profile limitations cited above.

3. RECENT TRENDS IN 2D PROCESS AND DEVICE MODELING

It was out of the same DMOS research efforts described above that both SEDAN and PISCES (1D and 2D device analysis) tools evolved. SEDAN was used in a quasi-2D mode to evaluate the DMOS threshold behavior⁷. Based on the obvious 2D effects observed in the DMOS, we in turn developed the PISCES program⁸. Using PISCES, we have subsequently evaluated a variety of power devices including: Field Assisted Turn-Off thyristor (FATO)⁹, Conductivity Modulated DMOS (CMDMOS)¹⁰ and GBT¹¹. In this section, we will first demonstrate the advances in 2D power device modeling by way of example. Next we discuss several key device physics effects that are necessary to achieve accurate modeling. Finally we return to the crucial role played by 2D process modeling - especially for new nonplanar devices - which exploits trenches.

Conductivity modulation is one key parameter in the operation of virtually all power devices - it controls the ON-resistance of the structure. As stated in the introduction, the evolution of MOS technology has strongly influenced the development of new power devices. Fig. 3 shows the cross-section of a conductivity modulated DMOS structure that uses trench technology and a novel injector structure to enhance the modulation effects. As can be seen, without the p+ diffusion in the middle of the structure, the device is simply a vertical power MOSFET structure. The p+ region, shown as buried below the surface, is the source of the minority carriers that are used to modulate the conductivity of the lightly doped epitaxial drift region. In general, this p+ injector can be placed at any depth below the surface or even at the surface. By having the p+ in-

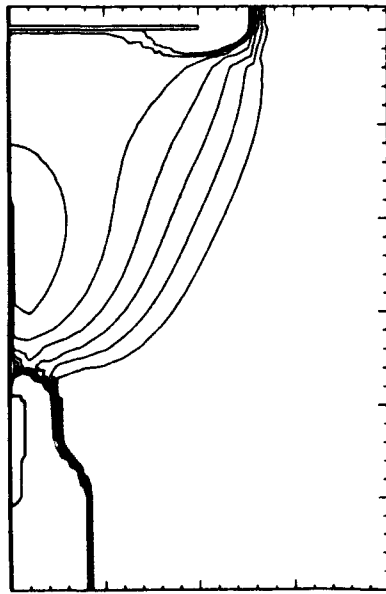


Fig. 4. Hole concentration contours from PISCES simulation for CMDMOS device with buried injector.¹⁰

dictated by charge neutrality requirements. In addition to these static plots of carrier distributions, PISCES has been used to characterize switching events¹⁰. Owing to the favorable geometric effects of the CMDMOS, minimum switching times on the order of 10 ns are possible.

Another structure that uses the benefits of a trench to achieve improved turn-off characteristics is the FATO⁹. The gated region in the trench serves as a shorting path for the current flow as shown in Fig. 5b. As with the analysis of the CMDMOS, PISCES was used extensively to characterize the ON- and OFF-state charge distributions as well as the transient effects. Fig. 5 shows that the trench structure can be thought of as inducing a second p-type collector for the thyristor's PNP transistor. Current flows through this collector and does not cause the device to latch. This current is analogous to the OFF-current in a GTO device. The area of this induced "turn-off" collector, relative to the area of the primary PNP, will determine its conduction capability and hence the maximum current that can be switched. As might be expected for this MOS gating device, the role of the surface mobility along the trench wall is crucial.

The surface mobility on the sidewalls of dry etched trenches were compared for two processes to that obtained on the wafer surface⁹. It has been determined that electron mobility is substantially reduced whereas mobility for holes is relatively unchanged. Two etching process variations were investigated in this study. Process A used SF_6 and C_2F_5 gases in a symmetrical, equal-electrode-area Radio Frequency (RF) reactor with the sample at ground. Anisotropy in etching was induced by polymer deposition on the sidewalls. Process B used a magnetic field enhanced plasma to increase the path length of electrons and hence the

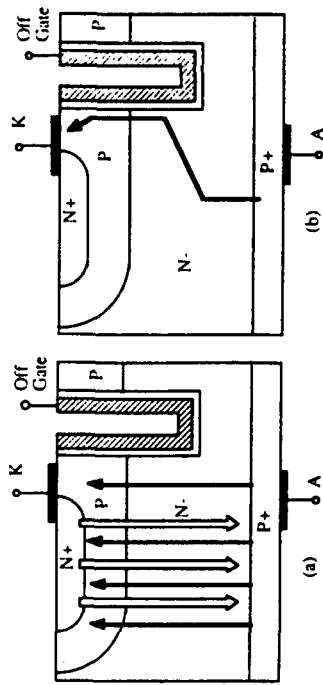


Fig. 5. Current flow paths in the FATO thyristor (a) during the on-state and (b) during the turn-off process. Filled arrows represent hole current; open arrows represent electron current⁹.

plasma density, giving a two-times higher etch rate ($7000 \text{ \AA}/\text{min}$). Anisotropy was induced by the formation of an oxide layer on the sidewall that was later removed. There were small but perceptible differences between surface mobilities resulting from the processes as illustrated in Fig. 6⁹. Neither of the etch processes investigated produced n-channel devices with surface quality mobilities. Post-etch treatments such as oxidation and stripping had little effect. It is believed that atomic scale surface roughness is the cause of this mobility reduction. Holes seem to be more immune to these surface roughness effects, thus their mobility is not significantly reduced by the dry etch process.

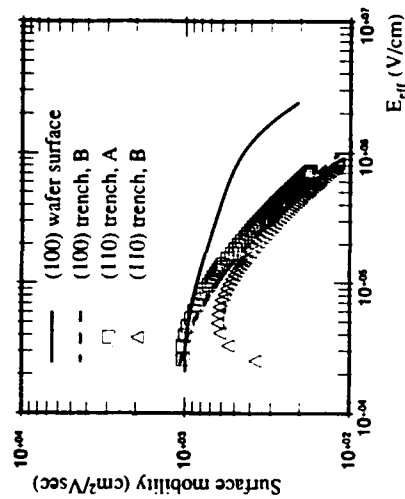


Fig. 6. Electron surface mobility vs. effective normal field for different surface orientations. All trenches underwent a sacrificial oxide post-etch treatment⁹.

The above examples have illustrated several trends in the design of power ICs. The effects associated with device ON-resistance and OFF-voltage each bring with them important physical effects as well as possibilities for innovative technology solutions in terms of new fabrication techniques and geometries. Each of these points is now discussed further based on specific examples using PISCES and SUPREM 4.

Aspects of the device physics to be considered here include ON-resistance and OFF-voltage. There are a range of parameters that control ON-resistance. The carrier mobility is influenced by factors such as surface scattering⁹ as well as bulk phenomena such as carrier-carrier scattering - both these effects are considered in PISCES.

The implementation of carrier-carrier scattering becomes important when both hole and electron concentrations are high which is an essential factor in lowering the ON-resistance of power devices. Under these conditions, the scattering events become much more probable. Dorkel and Leturcq¹² have proposed a semi-empirical mobility model which includes not only the effects of lattice and ionized impurity scattering but also the carrier-carrier scattering events. The equations that are implemented in PISCES are as follows¹³:

$$\mu_{ccs} = \frac{2 \cdot 10^{17} T^2}{\sqrt{(pn)}} \left\{ \ln \left[1 + 8.28 \cdot 10^8 T^2 (pn)^{-\frac{1}{3}} \right] \right\}^{-1} \quad (1)$$

lattice mobility:

$$\mu_L = \mu_{Lo} \left(\frac{T}{300} \right)^{-\alpha} \quad (2)$$

ionized impurity mobility:

$$\mu_I = \frac{AT^2}{N} \left[\ln \left(1 + \frac{BT^2}{N} \right) - \frac{BT^2}{N + BT^2} \right]^{-1} \quad (3)$$

where N is the concentration of ionized impurities, and A and B are parameters which depend on the nature of the carriers.

And finally, the mixed-scattering mobility expression which combines the above terms:

$$X = \sqrt{\frac{6 \mu_L (\mu_I + \mu_{ccs})}{\mu_I \mu_{ccs}}} \quad \mu = \mu_I \left[\frac{1.025}{1 + \left(\frac{X}{1.68} \right)^{1.43}} - 0.025 \right] \quad (4)$$

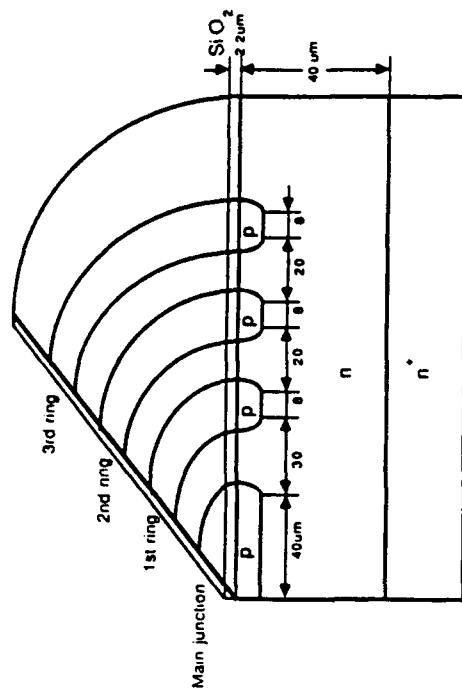


Fig. 7. Floating-field limiting structure with cylindrical symmetry¹⁴.

All parameters used in this implementation are taken from Dorkel and Leturcq.

In addition to the physical parameters such as mobility that affect ON-resistance, geometry plays a key factor as well. For many cases, the 3D influences on device behavior can be emulated using quasi-3D analysis where cylindrical symmetry is used to replicate the behavior in the third-dimension. Such a formulation has been added to PISCES and details of the implementation are described in 14. Fig. 7 shows a high voltage structure with three guard rings and various geometric parameters as indicated. The device is designed to break down in the punchthrough mode when the depletion region reaches the end of the lightly doped substrate. The doping concentration of the lightly doped n-epitaxial region is $1.5 \cdot 10^{14} \text{ cm}^{-3}$ with the $\langle 100 \rangle$ surface orientation and doping concentration of the n^+ region on the backside substrate is $1.3 \cdot 10^{18} \text{ cm}^{-3}$.

Figure 8 shows the potential of the reverse-biased floating-field limiting rings. The breakdown voltages are also indicated. Here, the breakdown voltage is determined at the voltage where the current becomes about 100 times larger than the current at the middle voltage range. The breakdown voltage obtained using quasi-three-dimensional simulation with interface charge agrees well with the experimental data. Two-dimensional simulation predicts about a 30-percent higher breakdown voltage than the experimental data. The difference of potential between the main junction and the first ring is crucial in determining the breakdown voltage because this difference is larger than

the difference of potential between any of the adjacent rings. Other details of the differences are discussed in 14.

Impact ionization effects are of critical concern in determining the high-voltage behavior of power devices. Although it is well-known that non-local effects are involved in carrier heating, to date there has been no completely reliable method to treat these effects - with the exception of directly solving for the carrier temperature. In the case of using the local "lucky electron" model, it is essential to base the computations on the term $J \cdot E$ where J is the local current density and E the electric field. Although our formulation of this term was correct¹⁴, the initial implementation was flawed. More recent PISCES versions have corrected that difficulty. As will be discussed in the future trends section, it becomes increasingly important to go beyond the limitations of the drift-diffusion and local model formulations in order to correctly predict the performance of power devices.

As a final theme in evaluating progress in 2D modeling for power devices, let us consider some of the process dependencies. The above examples have illustrated the innovative opportunities that come from using advanced processing to improve power device performance. In particular, trench etching has become a key part of both isolation and intrinsic device operation. In contrast to the limitations of present double-diffused and high-concentration diffusion models, advances in trench etching and nonplanar oxidation of complex geometries have generally kept pace with the development of new power device technologies. Figure 9 shows the simulation of simultaneous etching and deposition processes. A two step etching process is considered - the first being isotropic and the second anisotropic - thus creating the "champagne glass structure". In order to correctly simulate the anisotropic effects, the role of polymer deposition is critical. The SPEEDIE simulator accounts for not only such precursor effects but a variety of angle-dependent ion bombardment effects that influence the anisotropic behavior¹⁵.

Figure 9 shows both the simulated results and experimental SEM profiles. Note that, at the lower edge of the horizontal section, a "bump" is created, owing to the buildup of materials due to adsorption/re-emission mechanisms.

Turning to the oxidation of trench and other nonplanar etched surfaces, there are a number of factors that influence the growth of layers over such nonplanar surfaces. In particular, the role of stress on the growth of oxides at both convex and concave surfaces has been the subject of intense interest and investigation¹⁶. Based on both experimental and theoretical efforts, a more robust numerical implementation has been developed for handling the stress dependence^{17,18}. SUPREM IV is now capable of reliable and accurate simulations of complex oxidation conditions used in advanced IC technologies - suitable not only for power device structures but also advanced Dynamic Random Access Memory (DRAM) technologies as well.

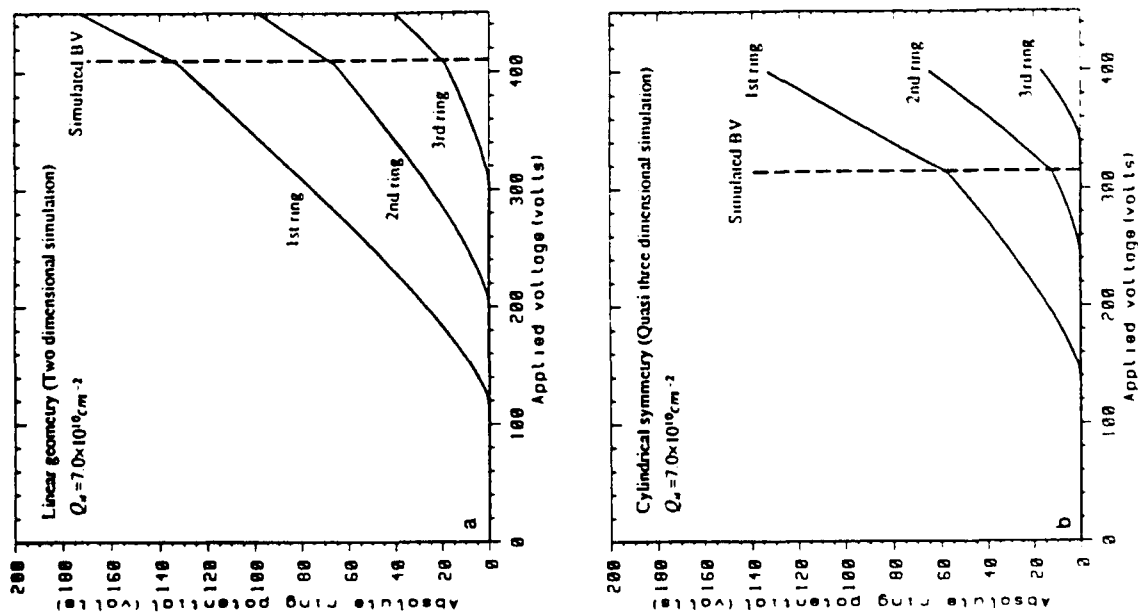


Fig. 8. Absolute ring potential versus applied voltage of the floating-field limiting ring structure with linear geometry with interface charge of $7 \times 10^{10} \text{ cm}^{-2}$ obtained using a) two dimensional simulation, b) Quasi three dimensional simulation¹⁴.

the difference of potential between any of the adjacent rings. Other details of the differences are discussed in ¹⁴.

Impact ionization effects are of critical concern in determining the high-voltage behavior of power devices. Although it is well-known that non-local effects are involved in carrier heating, to date there has been no completely reliable method to treat these effects - with the exception of directly solving for the carrier temperature. In the case of using the local "lucky electron" model, it is essential to base the computations on the term $J \cdot E$ where J is the local current density and E the electric field. Although our formulation of this term was correct¹⁴, the initial implementation was flawed. More recent PISCES versions have corrected that difficulty. As will be discussed in the future trends section, it becomes increasingly important to go beyond the limitations of the drift-diffusion and local model formulations in order to correctly predict the performance of power devices.

As a final theme in evaluating progress in 2D modeling for power devices, let us consider some of the process dependencies. The above examples have illustrated the innovative opportunities that come from using advanced processing to improve power device performance. In particular, trench etching has become a key part of both isolation and intrinsic device operation. In contrast to the limitations of present double-diffused and high-concentration diffusion models, advances in trench etching and nonplanar oxidation of complex geometries have generally kept pace with the development of new power device technologies. Figure 9 shows the simulation of simultaneous etching and deposition processes. A two step etching process is considered - the first being isotropic and the second anisotropic - thus creating the "champagne glass structure". In order to correctly simulate the anisotropic effects, the role of polymer deposition is critical. The SPEEDIE simulator accounts for not only such precursor effects but a variety of angle-dependent ion bombardment effects that influence the anisotropic behavior¹⁵.

Figure 9 shows both the simulated results and experimental SEM profiles. Note that, at the lower edge of the horizontal section, a "bump" is created, owing to the buildup of materials due to adsorption/re-emission mechanisms.

Turning to the oxidation of trench and other nonplanar etched surfaces, there are a number of factors that influence the growth of layers over such nonplanar surfaces. In particular, the role of stress on the growth of oxides at both convex and concave surfaces has been the subject of intense interest and investigation¹⁶. Based on both experimental and theoretical efforts, a more robust numerical implementation has been developed for handling the stress dependencies^{17,18}. SUPREM IV is now capable of reliable and accurate simulations of complex oxidation conditions used in advanced IC technologies - suitable not only for power device structures but also advanced Dynamic Random Access Memory (DRAM) technologies as well.

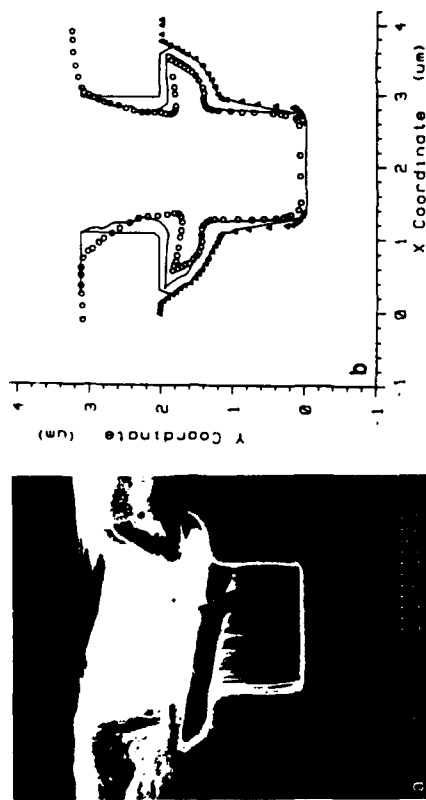


Fig. 9. Simulation and experimental results of simultaneous etching and deposition in a "Champagne Glass" test structure¹⁵; a) experimental results, b) SPEEDIE simulations

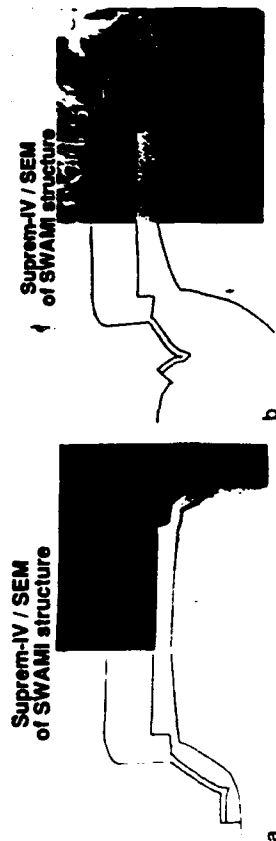


Fig. 10. Comparison of SUPREM-IV simulations to Transmission Electron Microscope (TEM) photographs of a SWAMI isolation process. The initial structure (a) and final structure after oxidation (b) are shown¹⁹.

Based on characterization of properties on the nitride/oxide layers, the improved model coefficients now make it possible to achieve good accuracy on complex structures¹⁹. For example, Fig. 10 shows the results of SUPREM simulations for an advanced "LOCally Oxidized Silicon" (LOCOS)-process (i.e. the so-called "SWAMI" type). The results show excellent agreement between measurements and simulation, both qualitatively and quantitatively. In addition to these fully-recessed LOCOS structures, SUPREM-IV can handle trenched devices and poly-buffered LOCOS as well.

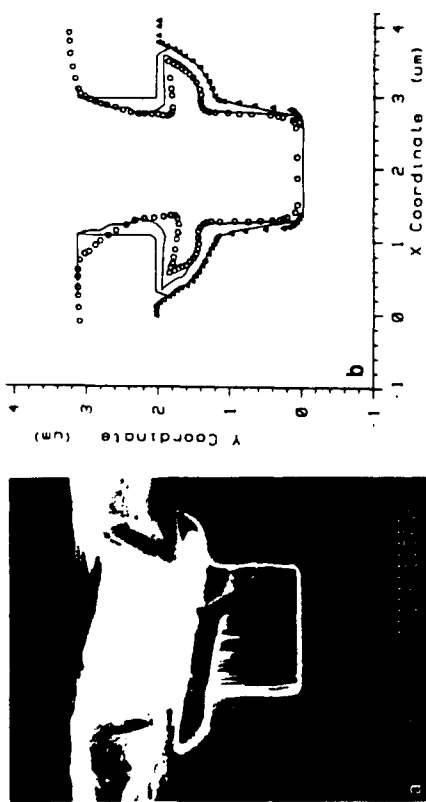


Fig. 9. Simulation and experimental results of simultaneous etching and deposition in a "Champagne Glass" test structure: a) experimental results, b) SPEEDIE simulations

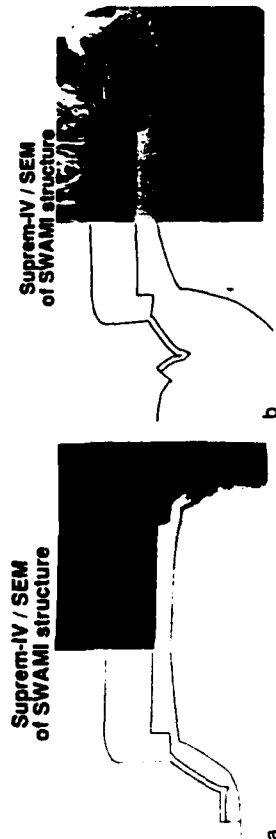


Fig. 10. Comparison of SUPREM-IV simulations to Transmission Electron Microscope (TEM) photographs of a SWAMI isolation process. The initial structure (a) and final structure after oxidation (b) are shown¹⁹

Based on characterization of properties on the nitride/oxide layers, the improved model coefficients now make it possible to achieve good accuracy on complex structures¹⁹. For example, Fig. 10 shows the results of SUPREM simulations for an advanced "LOCally Oxidized Silicon" (LOCOS)-process (i.e. the so-called "SWAMI" type). The results show excellent agreement between measurements and simulation, both qualitatively and quantitatively. In addition to these fully-recessed LOCOS structures, SUPREM-IV can handle trench devices and poly-buffered LOCOS as well.

4. MIXED-MODE SIMULATION FOR POWER IC'S

In addition to the unique features of the device physics and process technology used for power ICs, there is a diversity of circuit constraints that apply to the design of power ICs. At the most basic level, the normal SPICE models used for low power devices quickly fail as the operating conditions go into the high-level injection regime. Moreover, the operating temperature of power devices is higher than that of the surrounding control devices and these effects must be considered. Although there have been various attempts to customize SPICE and its models to account for temperature gradients across the chip (i.e. TSPICE), it is clear that more robust techniques are needed for future power IC designs.

Mixed-mode simulation which couples circuit and device analysis is one attractive answer to the problems stated above. Early work by the Aachen group resulted in the MEDUSA mixed-mode simulator²⁰. The first results demonstrated with this approach considered output power devices in an operational amplifier. These devices directly influence transient behaviour of the circuit and by using only conventional device models, the results show major qualitative and quantitative errors in the waveforms²⁰. Over the past few years, other approaches and refinements to mixed-mode simulation have emerged. The University of California (UC) Berkeley group has created the CODECS simulator based on SPICE 3 for the circuit analysis portion and the matrix routines are interfaced to a custom device solver²¹. The group at University of South California (USC) has modified PISCES directly and included the circuit matrix terms as part of the device solver²². In addition to the device-circuit level of mixed-mode for power devices, other tools are emerging that handle multiple levels of abstraction, specifically for power systems. The SABER simulator provides capabilities to model a variety of nonlinearities besides just the IC's²³. Although it is beyond the scope of this paper to consider the full spectrum of these mixed-mode simulators in detail, this paper will demonstrate representative examples of results for several interesting cases. The next section turns to the final issue of tool integration which encompasses not only the process and device levels of TCAD but also mixed-mode and other higher level tools that include the electronic circuit and system aspects.

5. INTEGRATING TCAD INTO EDA FRAMEWORKS

The interest and activity in development of CAD frameworks has grown steadily over the last decade. Stimulated in part by the concepts of "structured" IC design, first layout interchange formats were developed (i.e. Caltech Intermediate Form, CIF) followed by formalization of a much broader standard through the Electronic Design Interchange

Format (EDIF) Committee²⁴. Over the last few years an international CAD Framework Initiative (CFI) has emerged²⁵ and initial efforts have focused on the needs related to Electronic Design Automation (EDA). Recently a TCAD working group was formed with the objective to leverage from the broad objectives of CFI and at the same time to develop the essential and unique framework tools needed for TCAD.

To date the interface between TCAD and EDA has traditionally been thought of in terms of lumped values such as circuit model parameters and design rules. EDA framework-based design allows tools to access data from many levels of abstraction (including layout, schematic, netlist, etc.) through well defined interfaces - giving each tool freedom to determine the necessary inputs from all aspects of the design process. Modular interfaces allow framework compliant tools to be easily swapped in and out of the framework, providing rapid transfer of enhanced modeling ability to the user. Extending EDA frameworks to include TCAD is essential for integrating TCAD tools into the design process and to support the mixed-mode simulation tools discussed in the last section.

TCAD frameworks must address both deployment (integrating existing tools) and development (creating new tools). The use of common representations address both areas by minimizing the number of translators needed. It also encourages the creation of shared libraries of functions and services (e.g. I/O routines). By using these library routines within a framework context, developers can focus on the development of the physically-based models that provide the intrinsic benefit of TCAD to EDA. This is especially important in the area of power devices where resources for unique tool development are scarce and efforts must be focused. In the following discussion we review several key aspects of recent progress in developing the TCAD framework - especially the semiconductor wafer representation (SWR) is discussed which is the structural "back-bone" of TCAD.

Semiconductor Wafer Representation²⁶ - Wafer state describes the structures resulting from the fabrication and simulation of integrated circuits. Examples of wafer state information include topography, dopant profiles, and current density. This wafer state is created and modified via processing steps (e.g. deposition and implantation), and analyzed by device/circuit simulation.

The following describes the representation that the CFI TCAD Framework Group is developing^{26,27}. The Semiconductor Wafer Representation (SWR) is based on an object oriented approach - tools query and modify the representation using a functional interface. The representation can be partitioned into two major components - Geometry and Fields. Geometry provides structural and topological information about the wafer while Fields provide information about properties that vary within geometric regions, such as dopant concentration.

The basic unit of Geometry is the cell, a collection of zero-, one-, two-, and three-dimensional point sets. Operations on cells include create, destroy, query, and modification. Cells can be created by combining primitive cell - unions of rectangular or triangular cell - or by combining lower-dimensional objects (for example, constructing a face from a set of pair-wise adjacent edges). Because adjacency information must be readily available in TCAD simulators, sets of nonoverlapping cells can be grouped into special sets called "cell complexes". Within complexes, most queries about cell relationships (e.g. adjacency) can be answered quickly and accurately. Queries include classification (e.g. is this point inside this cell complex) and sectioning (e.g. return a 1D slice out of a 2D cell). Modification functions include "inset" (for deposition), and "subtraction" (for etching).

Fields represent mappings from a domain, usually a cell associated with the geometry, to a range. A specialization of field is "field on mesh". Numerical methods for solving equations that do not possess closed form solutions typically use meshed regions. Facilities are provided for automatically creating and refining several types of commonly used meshes (triangular elements, tensor product). Meshes can also be constructed from lower-dimensional elements in a manner similar to the creation of geometry cells. An important operation is evaluation of a particular field at a point in space (e.g. find the boron concentration at this location). Modifications of fields usually result from application-specific numerical operations, for example, solving the diffusion equations.

Inconsistency between fields and geometry is allowed to enable client applications to define their own set of consistency functions. For example, a string-based etching simulator can modify Geometry but has little knowledge of which algorithm should be used to force the existing "fields on mesh" to follow the new boundary. A subsequent diffusion client, however, could make the old mesh consistent with the new boundary in a way that minimizes discretization error for the diffusion equation.

The following program code shows how a tool simulating the photolithography process would manipulate the wafer representation through the functional interface. Fig. 11 shows snapshots of the wafer between function calls.

```
/* retrieve the existing wafer (Fig. 11a). */
cell_complex1 = getCellComplex(wafer);
/* deposit_layer uses the geometry component to create a cell based
on the photoresist deposition process. A reference to the new cell is
returned (Fig. 11b). */
dcell1 = deposit_layer(cell_complex1);
/* incorporate the cell into the existing wafer (Fig. 11c). */
wafer.insetCell(dcell1);
/* set the material of the cell to photoresist. */
dcell1.setMaterial(Resist);
```

The tool developer needs only to provide code for "deposit_layer", since the additional functions necessary to access the wafer boundary and create the new geometry are provided through the programming interface "get" and "insert" methods. Specialized functions such as "insetCell" are extremely complex and should be written by computational geometry experts.

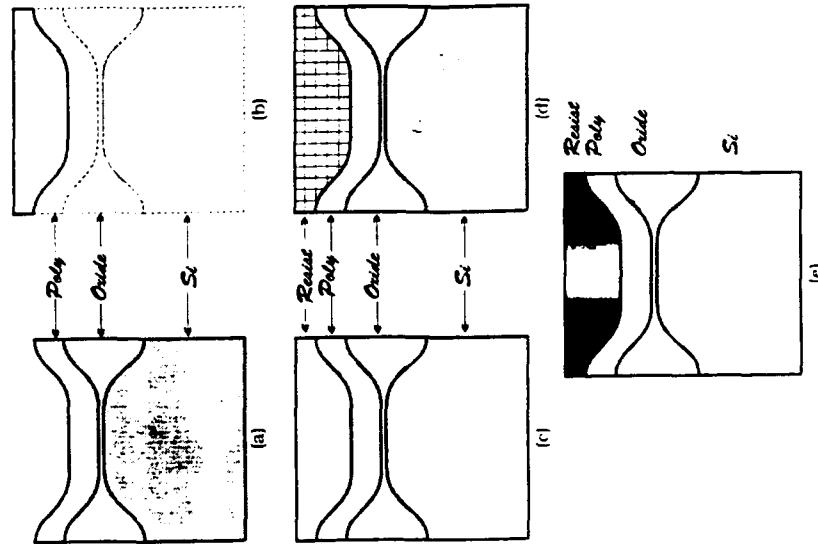


Fig. 11. "Snapshots" during SWR example execution: Before (a) and after (b) calling "deposit_layer", after calling "setMaterial" (c), after calling "createMeshedField" (d), and after calling "perform_lithography", showing the field values for illumination per square micron (e).

The next code sequence continues processing by having another tool perform photolithography on the deposited photoresist.

```
/* createMeshedField creates a new "field on mesh" associated with a
particular cell. The cell is automatically meshed if no tensor mesh is
present (Fig. 11d). "Hints" describes a data object containing in-
formation for the meshing algorithm, such as desired mesh density.
A reference to the field is returned. */
exposure = dcell.createMeshedField(TENSOR, hints);
/* the photolithography client, for example SAMPLE, takes in-
formation provided by the cell complex and fields, determines ex-
posure intensity as a function of position, and puts values in the field
(Fig. 11e) */
perform_lithography(exposure, wafer);
```

Again, the tool developer is only required to provide code for the "perform_lithography" routine. Mesh generation and management of field information are provided through the programming interface. "CreateMeshedField" replaces a long sequence of steps that developers traditionally used to update wafer state. One of the more important aspects of the "createMeshedField" routine is the ability to automatically mesh a cell, an operation that requires knowledge of both computational geometry and numerical analysis. Meshers are difficult to write and should be provided to the typical developer.

As can be seen, the SWR interface makes the manipulation of wafer state transparent to the tool developer. In the past, each model developer typically had his own implementation of the needed operations, instead of reusing modular pieces of code. Because many programs can use the same implementation of the SWR, its providers can concentrate on improved programs for these core services, in turn freeing the users to concentrate on the physics being modeled.

The above discussion of the SWR and the example given in Fig. 11 illustrate some of the underlying issues of the TCAD framework. At the higher levels, there are other system integration challenges, for example control of tool execution and the substantial data base problems that come with applications such as statistical design. Moreover, as suggested by the previous discussion of mixed-mode simulation, there are a growing set of heterogeneous tools with interfaces that will require new methods and paradigms for design using such tools. In the domains of power systems, the range of devices to be controlled is indeed diverse and the complex interactions of not only electrical and thermal effects but also mechanical and even noise sources pose unique challenges for system modeling and tool integration. In the final section of this paper we turn to a more limited but important set of future challenges, consid- eration of 3D modeling and electron temperature effects.

6. FUTURE TRENDS

In the discussion of current trends in 2D TCAD modeling, we briefly explored two points that are now discussed further - 3D analysis and non-thermal equilibrium carrier transport effects. Here we broaden the discussion of these topics in light of recent trends both in the emerging TCAD Framework and High Performance Computing (HPC).

In the area of device analysis, there has been steady progress in development of new 3D tools. However, two limitations have impeded the broad engineering application of such 3D tools - specifically, computation time and user interface related issues. Limitations of computer resources, both in terms of CPU requirements and memory bandwidth, is one major problem. Even with increased CPU performance into the tens-of-MFlop range, the memory size and speed limitations of most Electronic Work Station (EWS) systems support only modest scale 3D computations. Over the past several years the Stanford group has actively pursued 3D device analysis, using a Multiple-Instruction Multiple-Data (MIMD) model for computation-sometimes called "coarse-grain" parallelism. Using a third-generation Intel iPSC/860 we have now demonstrated very impressive results using a 32 processor configuration²⁸.

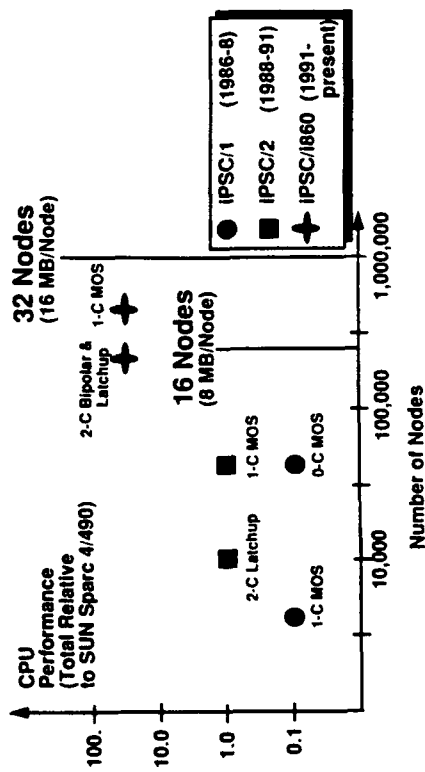


Fig. 12. Log CPU Speedup (units relative to SUN 4/490) versus number of nodes for large 3D problems on three generations of Intel iPSC computers.

Figure 12 shows a plot of Central Processing Unit (CPU) performance compared to a single processor SUN 4/490 versus number of analysts nodes for a variety of 3D MOS, bipolar, and CMOS latchup examples. Also shown on the plot are the memory limitations for the three generations of systems. Because of the substantial improvements in

speed of the iPSC/860 system, users have been more willing to input larger problems and the computation time is still reasonable. For example, a 50,000 bipolar analysis takes approximately 2 minutes per bias point²⁹. This performance is orders of magnitude improved over earlier generations of hypercubes and is comparable to state-of-the-art super computers. We are now in the process of formulating and testing even larger problems using a prototype Intel system with more than 500 computational nodes. Here, we hope to surpass the performance of present commercially available systems.

This brings us to the second limitation in 3D analysis, the inherent difficulties in formulating the problems and interpreting the results. In contrast to the 2D device technologies cross-sections used extensively as examples in the previous sections, the 3D domain requires a new look at the required data from an IC mask layout perspective. Our recent experience, using the Berkeley SIMPL-IPX³⁰ tool as a key part of the user interface has been productive and will be discussed. By means of supplemental user panels, interfaces that support not only simplified definition of geometry but also virtual measurement tools for parameter extraction have been demonstrated. Fig. 13 shows a sample of how "virtual" measurement panels are used to both control the simulator execution and also display the results for parametric extraction shown in Fig. 13b is biased models. The generalized bipolar structure shown in Fig. 13b is biased according to an interactive electrode selection. An HP 4145 "virtual measurement" panel is used to apply bias conditions, exactly as they will occur using the real equipment (see Fig. 13a). For the bipolar transistor shown, a macro-measurement routine can be used to create a procedure to extract early voltage as shown in Fig. 13c. These "macro routines" both drive PISCES and can also be used to down-load into the actual measurement equipment. The resulting capabilities allow the user to go smoothly and efficiently from layout-based device specification to parameter extraction that supports circuit modeling. Such an automated approach is of growing importance to assist non-experts in the use of TCAD.

The final topic to be discussed in this section is thermal effects in power devices. As stated in the consideration of impact ionization, non-thermal equilibrium effects are now essential in not only power devices but short channel MOS devices for digital applications too. There continues to be a vigorous discussion how to correctly formulate the problem not only for the carrier temperature but also for the lattice and ambient conditions³¹. In the context of the emerging TCAD framework, it is essential to support this type of formulation since it clearly seeks to merge the disparate areas of electrons, phonons and the systems as a whole. In this section of the paper, I will show some recent efforts in improved carrier temperature models.

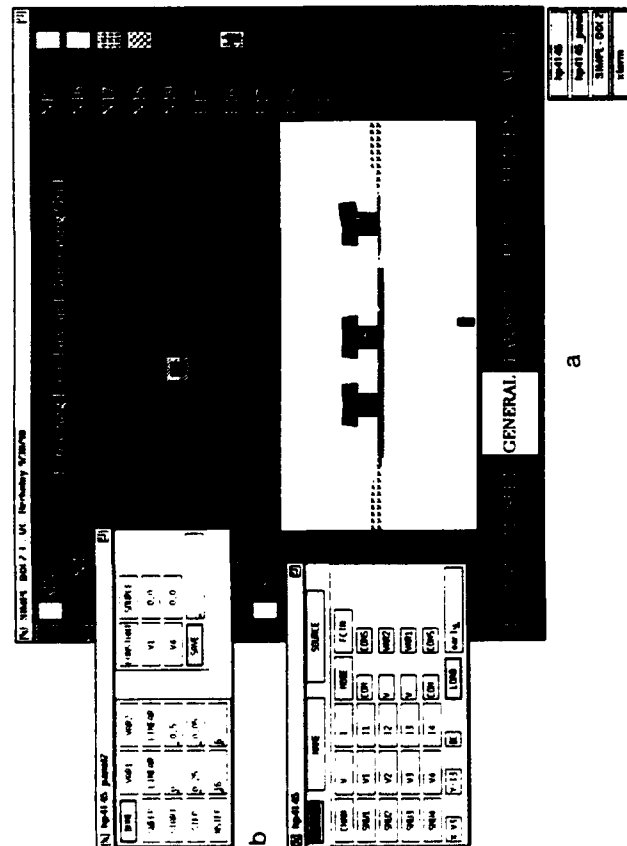


Fig. 13. Layout and virtual instrument-driven interfaces for TCAD: (a) HP4145 virtual panel interface, (b) bipolar device cross-section and electrode assignment, (c) I-V data and algorithm used for extraction of Early voltage.

Recent work, jointly with University of Illinois, has yielded an improved Energy Transport (ET) formulation³². To date the well known problem with Hydrodynamic (HD) type of formulations is the unphysical velocity peaking in transition regions between low and high doping. In the new ET model, the Boltzmann transport equation is re-examined and subtle differences in position of terms and order of integrations have revealed the cause of the peaking anomalies. Moreover, a physical explanation is offered³³

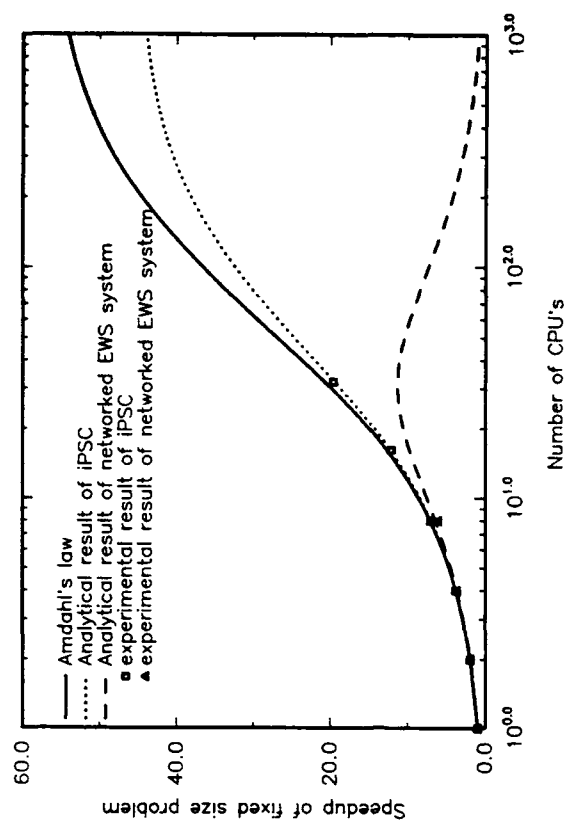


Fig. 14. Plot of CPU speed up factor (referenced to a single CE) versus number of CE's. Data and analytical curves compare performance for the Intel iPSC/860 and a network of EWS³⁴.

It is generally accepted that, in order to calibrate HD and ET formulations, the use of Monte Carlo (MC) simulators is a major benefit. Yet MC simulation is costly in terms of simulation time. The use of parallel computation offers a major benefit in this area. The use of both a network of EWS and the Intel iPSC/860 have been compared for MC computations in MOS devices³⁴. Fig. 14 shows a comparison of speed-up factor versus number of Computational Elements (CEs) for the two approaches. Also shown are the analytical projections, based on equations derived from the experimental data for the two cases. Experiments were performed with up to 32 computational elements (CEs). Clearly, the communications for the hypercube (Intel) system is superior and is sufficient to sustain high performance even with more than 100 CEs. On the other hand, for the network of EWS, the combination of network limitations

and software protocols makes it counterproductive to use more than a few dozen CEs. It is expected that improved communications networks will substantially enhance both approaches to MC calculations. Moreover, the opportunities to utilize "windowed" approaches³⁵ for ET, HD, and MC analysis can provide something like a mixed-mode benefit in understanding the physics with reasonable CPU expenses.

7. CONCLUSIONS

This paper started from an historic view of evolution from 1D TCAD models for power devices, looking carefully at the crucial role of diffusion modeling and parameter extraction based on SRP data. Recent progress in 2D modeling of power devices such as the CDMOS and FATO was reviewed, based on PISCES simulations. The importance of nonplanar technology for power devices was discussed and advanced process models in this area were illustrated using the SPEEDIE and SUPREM-IV simulators. Next, several of the issues involving mixed-mode simulation were reviewed in light of recent development of both university and commercial tools. The motivation to create a TCAD framework and recent progress in developing a prototype implementation has been discussed. Issues of importance to the power device community focus on both the heterogeneous tool requirements and the ability to effectively leverage from a broader community of IC ECAD/TCAD development effort already in progress. In the area of future trends, the issues of High Performance Computing (HPC) for 3D and requirements for modeling non thermal equilibrium effects were discussed. Results based on use of the Intel iPSC/860 computer with 32 nodes were demonstrated both for 3D modeling and MC analysis.

8. ACKNOWLEDGMENT

The authors acknowledge a number of government and industrial sponsors of this research. We gratefully acknowledge support from DARPA and ARO in the areas of advanced device modeling and HPC. Broad industry support of our TCAD modeling - both the process and device aspects - comes through the SRC. There have been a number of industrial contracts and visitors specifically in the area of power devices. Dr. Christiaan Abbas, from ABB was one of our first industrial visitors and his work led to many key changes in PISCES that leveraged subsequent work. Mr. Akira Yabuta of MEW was another industrial contributor who continued shaping the development of PISCES for quasi-3D power device applications. Other corporate sponsors of this work include: General Electric, General Motors, Matsushita (MEW, MEI, MEC), National Semiconductor, Phillips, Kodak, and Samsung. We have been continually stimulated from this diverse group of international contacts and we ap-

preciate ongoing support to continue our TCAD efforts in this exciting area.

REFERENCES

- 1 H. G. Sigg, et al "D-MOS Transistor for Microwave Applications", *IEEE Trans. Elect. Dev.*, Vol. ED-16, pp. 45-53, Jan. 1972.
- 2 J. D. Plummer, et al. "An Ultrasonic Imaging System for Real Time Cardiac Imaging," *ISSCC Tech. Digest*, pp. 162-163, Feb. 1974.
- 3 M. D. Pocha, et al. "Threshold Voltage Controllability in Double-Diffused MOS Transistors," *IEEE Trans. ED*, vol. ED-21, pp. 778-784, Dec. 1974.
- 4 R. W. Dutton, et al. "Correlation of Fabrication Process and Electrical Device Parameter Variations," *IEDM Tech. Digest*, pp. 609-613, Dec. 1976.
- 5 D. C. D'Avanzo, et al. "Effects of the Diffused Impurity Profile on the DC Characteristics of VMOS and DMOS Devices," *IEEE J. Solid State Circuits*, Vol. SC-12, pp. 356-362, Aug. 1977.
- 6 M. Orlowski, "Impurity and Point Defect Redistribution in the Presence of Crystal Defects", *IEDM Technical Digest*, pp. 729-732, Dec. 1990.
- 7 D. C. D'Avanzo, "Modeling and Characterization of Si-Port-Channel Double Diffused MOS Transistors," Stanford University PhD thesis, March 1980.
- 8 C. H. Price, "Two-Dimensional Numerical Simulation of Semiconductor Devices," Stanford University PhD thesis, May 1982.
- 9 C. J. Petti, "Physics and Technology of the Field-Assisted Turn-off Thyristor: A Regenerative Device with Voltage-Controlled Turn-off", Stanford University PhD thesis, Dec. 1989.
- 10 D. K-Y Liu, "Physics and Technology of Conductivity Modulated Power MOSFETs", Stanford University PhD thesis, September 1989.
- 11 D. M. Boisvert, "Physics and Technology of High-Speed, Low On-Resistance Power Devices," Stanford University PhD thesis, April 1990.

- 12 J.M. Dorkel and P. Leturcq, "Carrier Mobilities in Silicon Semi-Emiprically Related to Temperature, Doping and Injection Level", *Solid State Electron.*, vol 24, pp. 821-825, 1981.
- 13 G. Anderson, et al. "Metamorphosis of PISCES-Application-Oriented Transformation of 2D Device Simulation" *TECHCON 1990*, San Jose, CA, October 15-18, 1990.
- 14 A. Yabuta et al. "Numerical Analysis of Breakdown Voltage Using Quasi Three Dimensional Device Simulation," *IEEE Trans. Elect. Dev.*, Vol. 37, No. 4, April 1990.
- 15 J.P. McVittie, J.C. Rey, A.J. Bariya, M.M. Islamraja, S. Ravi and K.C. Saraswat, "SPEEDIE: Simulation of Profile Evolution During Etching and Deposition", *Proceedings SPIE Symp. Adv. Techniques for Integrated Circuits Processing*, Vol. 1392, pp. 126-138 (1990).
- 16 D. B. Kao, "Two-Dimensional Oxidation Effects in Silicon Experiments and Theory," Stanford University PhD Thesis, 1986.
- 17 C.S. Rafferty, "Stress Effects in Silicon Oxidation-Simulation and Experiments," Stanford University PhD Thesis, 1989.
- 18 Y. Oda, et al. "Numerical Techniques on Enhancing Robustness for Stress Dependent Oxidation Simulation Using Finite Element Method in SUPREM-IV", *IEICE Trans. Elect.*, vol. E75-C, no. 2, pp. 150-155, Feb. 1992.
- 19 P. B. Griffin, C. S. Rafferty, "A Viscous Nitride Model for Nitride/Oxide Isolation Structures," *International Electron Devices Meeting, Technical Digest*, pp. 741-743, Dec. 1990.
- 20 W.L. Engl et al. "MEDUSA-A simulator for Modular Circuits" *IEEE Trans. on CAD*, vol. CAD-1, No. 2, pp. 85-93, April 1982.
- 21 K. Mayaram and D.O. Pederson, "CODECS: A Mixed-level Device and Circuit Simulator", *ICCAD Technical Digest*, pp. 112-115, November 1988.
- 22 J.G. Rollins and J. Choina Jr., "Mixed Mode PISCES-SPICE Coupled Circuit and Device Solver", *IEEE Trans. on CAD*, Vol. CAD-7, No. 8, pp. 862-67, Aug. 1988.
- 23 M. Vlach, "Modeling and Simulation with Saber", *Proceedings of The Third Annual IEEE ASIC Seminar and Exhibit*, Sept. 17-21, 1990.

- 24 EDIF Steering Committee, "EDIF - Electronic design Interchange format version 2.0.0", Electronic Industries Association, Washington, D. C., 1987.
- 25 CFI Technical Coordinating Committee, "CFI Candidate Proposal, CAD Framework Users, Goals and Objectives - Version 0.91," CAD Framework Initiative, Inc., Austin, TX, 1991.
- 26 CFI TCAD Framework Group, Semiconductor Wafer Representation Working Group, "Semiconductor wafer representation architecture-version 0.1," CAD Framework Initiative, Inc., Austin, TX, 1991.
- 27 CFI TCAD Framework Group, Semiconductor Wafer Representation Working Group, "Semiconductor wafer representation procedural interface - version 0.09," CAD Framework Initiative, Inc., Austin, TX, 1991.
- 28 K. C. Wu, et al. "New Approaches in a 3-D One-Carrier Device Solver," *IEEE Trans. CAD/ICAS*, Vol. 8, No. 5, May 1989.
- 29 K. C. Wu, et al. "A STRIDE Towards Practical 3D Device Simulation-Numerical and Visualization Considerations," To be published *IEEE Trans. CAD*, Sept. 1991.
- 30 E. W. Scheckler, et al. "A Utility-Based Integrated Process Simulation System", *1990 Symposium on VLSI Technology-Digest of Technical Papers*, pp. 97-98, 1990.
- 31 G. K. Wachutka, "Rigorous Thermodynamic Treatment of Heat Generation and Conduction in Semiconductor Device Modeling," *IEEE Trans. CAD*, Vol. 9, No. 11, Nov. 1990.
- 32 D. Chen, et al. "A self-consistent Discretization Scheme for Current and Energy Transport Equations," *SISDEP '91*, Zurich, Sept. 1991.
- 33 D. Chen, et al. "Analysis of Spurious Velocity Overshoot in Hydrodynamic Simulations of Ballistic Diodes," *NUPAD IV (Numerical Process and Device Modeling Workshop) Digest*, pp. 109-114, Seattle, May 31 - June 1, 1992.
- 34 S. Sugino, et al. "Parallelization of Monte Carlo Analysis on Hypercube Multiprocessors and on a Networked EWS System," *SISDEP '91*, Zurich, Sept. 1991.
- 35 D. Y. Cheng, et al. "PISCES-MC, A Multi-window, Multi-Method 2D Device Simulator," *IEEE Trans. CAD/ICAS*, Vol. CAD-7, No. 9, Sept. 1988.

Robust and Efficient *ac* Analysis of High-speed Devices ^[8]

Ke-Chih Wu Zhiping Yu Lydia So Robert W. Dutton Junko Sato-Iwanaga*
Center for Integrated Systems, Stanford University, CA 94305

Abstract

A robust iterative algorithm for high frequency small-signal *ac* analysis is presented which combines a novel matrix transformation scheme and a memory-efficient iterative method - TFQMR (Transpose-Free Quasi-Minimal Residual). Excellent convergence behavior is demonstrated for frequencies up to 10^{15} Hz without requiring increase in array size used in *dc* analysis. Device simulator (PISCES) using this new algorithm extends the frequency range of *ac* analysis far beyond present measurement capabilities, enabling us to investigate the device behavior at extremely high frequencies with reasonable computation time and memory requirement. The results shown here are for GaAs MESFETs but the techniques are applied to bipolar and other high-speed devices as well.

Introduction

Small-signal *ac* analysis is an important aspect of device simulation, not only for the determination of the frequency response of devices but also for the extraction of device parameters such as the cutoff frequency, f_T , and S -parameters. However, the conventional SOR (Successive Over-Relaxation) algorithm used in many device simulators fails to converge when the frequency approaches f_T [1], and algorithms which eventually use the direct solution method at high frequencies [3] require substantial memory space that must be allocated even for *dc* or low-frequency *ac* analysis. In this paper, we demonstrate a new algorithm which combines a novel matrix transformation scheme and a memory-efficient iterative method - TFQMR (Transpose-Free Quasi-Minimal Residual) [5] - which shows excellent convergence behavior (up to 10^{15} Hz in our test examples) without requiring increases in the array size used in *dc* analysis. Device simulator (PISCES) using this new algorithm extends the frequency range of *ac* analysis far beyond present measurement capabilities, enabling us to investigate the device behavior at extremely high frequencies with reasonable computation time and memory requirement. The results shown here are for GaAs MESFETs but the techniques can be applied to bipolar and other high-speed devices as well.

*Semiconductor Research Center, Matsushita Electric Industrial Co., Moriguchi-shi, Osaka 570, Japan

The New Iterative *ac* Analysis Algorithm

The small-signal *ac* simulation involves the solution of the following linear system for $\tilde{x} = x_R + jx_I$ where $j = \sqrt{-1}$:

$$A_o x = \begin{pmatrix} J & -\omega D \\ \omega D & J \end{pmatrix} \begin{pmatrix} x_R \\ x_I \end{pmatrix} = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad (1)$$

where J is the Jacobian matrix from the *dc* solution, ω is the angular frequency under analysis, D is a diagonal matrix with zero entries for the Poisson's equation and B is the stimulus in real number which has non-zero entries only for the Poisson's equation. Thus $DB = 0$, the property of which is to be used in Eq. (3). When ω increases, the off-diagonal blocks ωD becomes increasingly important, eventually leading to the divergence in the SOR method.

However, considering A_o as consisting of 2 by 2 matrix blocks, the presence of off-diagonal blocks with opposite signs actually enhances the determinant of the whole matrix. This observation suggests that the difficulties in obtaining convergent solutions may be overcome by proper matrix transformation [2]. Indeed, if a transformation matrix as indicated below

$$T = \begin{pmatrix} I & \omega DD_J^{-1} \\ -\omega DD_J^{-1} & I \end{pmatrix} \quad (2)$$

is introduced, where I is the identity matrix and D_J is the diagonal of J , Eq. (1) becomes: $A_T x =$

$$\begin{pmatrix} J + \omega^2 DD_J^{-1} D & \omega DD_J^{-1} J_0 \\ -\omega DD_J^{-1} J_0 & J + \omega^2 DD_J^{-1} D \end{pmatrix} \begin{pmatrix} x_R \\ x_I \end{pmatrix} = \begin{pmatrix} B \\ 0 \end{pmatrix} \quad (3)$$

where $J_0 = J - D_J$. It is immediately clear that although the magnitude of off-diagonal blocks still increases proportionally with ω , that of the diagonal entries are also enhanced by terms proportional to ω^2 , avoiding the problem of the diagonal blocks being overwhelmed by the off-diagonals. There are three distinctive ranges of frequencies where the transformed matrix shows different characteristics. At low frequencies, which can be quantified by all the non-zero entries in ωDD_J^{-1} being much less than unity, off-diagonal blocks can be considered as small perturbations with respect to the diagonal blocks and iterative algorithms should do well with or without the transformation. At very high frequencies, where all the non-zero entries of ωDD_J^{-1} are much greater than unity, the

enhancement to the diagonal blocks make the entire matrix become increasingly diagonally dominant which implies improving performance for the iterative algorithms. At the intermediate frequencies between these extremes, however, the situation is not as clearly cut since the off-diagonal blocks can no longer be considered as small perturbations while the growth in the enhancements to the diagonal blocks has not caught up with the growth in the off-diagonals. The practical implication of these characteristics will be examined in more detail with simulation results. The above discussion suggests that in iterative algorithms employing preconditioner, M , use of

$$M_T = \begin{pmatrix} J + \omega^2 DD_j^{-1} D & 0 \\ 0 & J + \omega^2 DD_j^{-1} D \end{pmatrix} \quad (4)$$

at the intermediate and extremely high frequencies gives better convergence properties than does the use of

$$M_o = \begin{pmatrix} J & 0 \\ 0 & J \end{pmatrix} \quad (5)$$

As an iterative algorithm, the SOR method has the advantage of simplicity and minimum requirement for data storage. In practice, however, we have found that its convergence depends critically on the condition that the frequency is kept below a sharply defined threshold above which the SOR iteration would simply stop converging and would diverge even for a small increase in the frequency. Additional difficulties arise in SOR from its need to supply a proper relaxation coefficient, which is difficult to obtain. In contrast, there are several attractive advantages with the so-called QMR (Quasi-Minimal-Residual) iterative algorithm [4]. Until quite recently, all existing iterative algorithms either use non-minimization procedure in generating the update vectors, which results in a very volatile residual reduction process, or use a full minimization procedure which requires increasing data storage as the iteration proceeds. By applying a quasi-minimal residual approach, QMR is able to retain the advantage of requiring only fixed amount of data storage. At the same time, the residual reduction process becomes very smooth with only occasional increases of insignificance. Furthermore, like all other Krylov subspace method based on Lanczos vectors, no external parameters are needed [4]. The particular QMR algorithm we have implemented is the TFQMR (Transpose-Free Quasi-Minimal-Residual) [5] with the left preconditioner $M = M_T$ for $A = A_T$ and $M = M_o$ for $A = A_o$. The procedure of TFQMR is summarized in the following:

(1) Initialization:

- (a) $b = M^{-1}B$;
- (b) $y_1 = u_0 = r_0 = \tilde{r} = b$, $x_0 = d_0 = 0$;

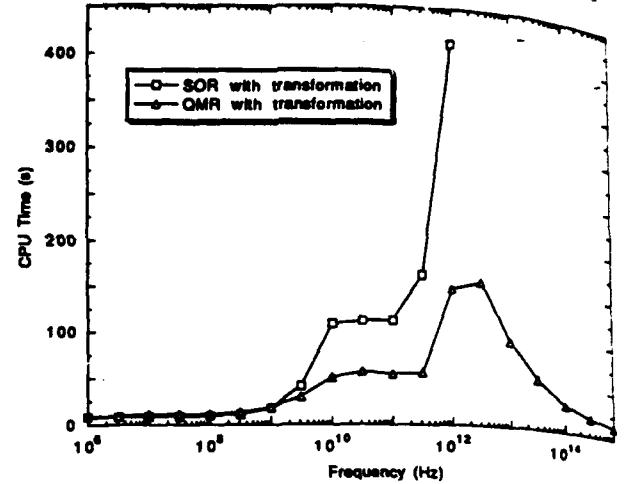


Figure 1: Comparison of SOR and QMR with matrix transformation.

$$(c) \rho_0 = \tau_0 = \|\tau_0\|, \theta_0 = \eta_0 = 0$$

(2) For $n = 1, 2, \dots$ do

- (a) $v_{n-1} = M^{-1}Ay_{2n-1}$;
- (b) $\sigma_{n-1} = \tilde{r} \cdot v_{n-1}$, $\alpha_{n-1} = \rho_{n-1}/\sigma_{n-1}$;
- (c) $y_{2n} = u_{n-1} - \alpha_{n-1}v_{n-1}$;
- (d) $r_n = r_{n-1} - \alpha_{n-1}M^{-1}A(u_{n-1} + y_{2n-1})$;
- (e) $\omega_{2n+1} = \|r_n\|$, $\omega_{2n} = \sqrt{\|r_n\| \cdot \|r_{n-1}\|}$;
- (f) For $m = 2n - 1, 2n$ do
 - $d_m = y_m + (\theta_{m-1}^2 \eta_{m-1} / \alpha_{n-1}) d_{m-1}$;
 - $\theta_m = \omega_{m+1} / \tau_{m-1}$, $c_m = 1 / \sqrt{1 + \theta_m^2}$;
 - $\tau_m = \omega_{m+1} c_m$, $\eta_m = c_m^2 \alpha_{n-1}$;
 - $x_m = x_{m-1} + \eta_m d_m$;
 - If x has converged: stop
- (g) $\rho_n = \tilde{r} \cdot r_n$, $\beta_n = \rho_n / \rho_{n-1}$;
- (h) $u_n = r_n + \beta_n y_{2n}$;
- (i) $y_{2n+1} = u_n + \beta_n (y_{2n} + \beta_n y_{2n-1})$

In [5], $\sqrt{1 + m\tau_m}$ is given as an upper limit for the residual norm. In our implementation, the residual norm is initially assumed to be equal to τ_m (a coefficient of unity). When τ_m drops below the tolerance, the true residual norm is evaluated which also gives out the real proportional coefficient between the residual norm and τ_m . If the residual norm is not below the tolerance, the iteration continues with the new coefficient. The comparisons of the numerical performance of QMR and SOR with and without the matrix transformation are shown in Figures 1 and 2. As Figure 1 indicates, the difficulty in selecting a proper relaxation coefficient causes SOR method to under-perform QMR above 1GHz. Also, the QMR curve

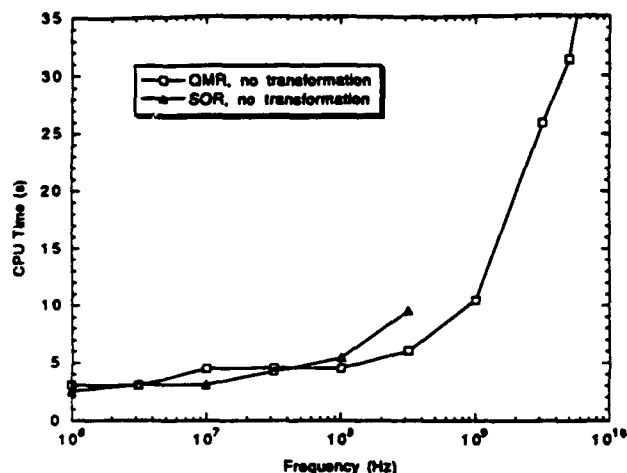


Figure 2: Comparison of SOR and QMR without matrix transformation.

extends up to 10^{15} Hz to verify that its convergence indeed improves at extremely high frequencies. The maximum of iteration number in the QMR curve seems to be closely related to the ratio of maximum and minimum values of non-zero entries in DD_j^{-1} . The ratio for this example is about 130. At low frequencies, a major overhead for using the matrix transformation is the need to carry out a Jacobi matrix decomposition at every frequency which dominates the CPU time. Since the matrix transformation achieves little at these frequencies, it is desirable to use it only when needed. In this regard, the magnitude of ωDD_j^{-1} seems to be a good measure since it determines how far the transformation matrix deviates from the unity matrix. In practice, the maximum value of the non-zero entries of ωDD_j^{-1} is used as the figure of merit. As can be seen from Figure 2, without matrix transformation QMR is quite competitive with SOR when both methods converge. However, SOR fails to converge well below 1 GHz while QMR continues to converge at much higher frequencies. In fact, until about 5 GHz, QMR without transformation outperforms QMR with transformation. At still higher frequencies, QMR without transformation never stops to converge but the rate of convergence falls off very quickly. From these results, a demarcation value for the figure of merit of about 0.3 has been selected above which the matrix transformation is carried out.

Application to GaAs MESFET Analysis

GaAs MESFETs with gate length of $1\mu\text{m}$ and width of 2 mm have been fabricated and measured for h - and S -parameters for various frequencies at different bias. Figure 3 shows the current gain comparison at $V_{DS} = 5\text{V}$ and $V_{GS} = -0.5\text{V}$. For all the data available, simulation results closely track those of the measurement, and

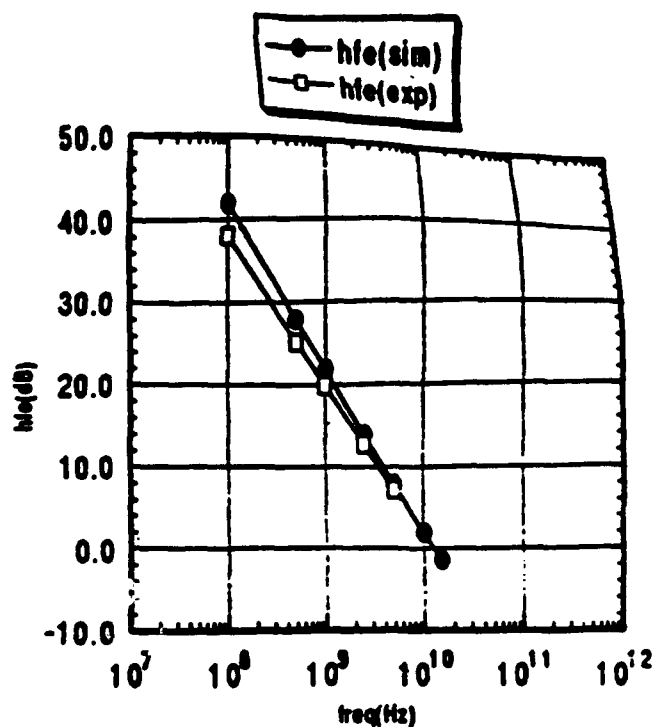


Figure 3: Simulated and measured current gain h_{fe} at $V_{DS} = 5\text{V}$ and $V_{GS} = -0.5\text{V}$.

beyond the measurable frequency range simulation predicts the device characteristics well in line with the expected (extrapolated) behavior providing an estimated f_T of about 14 GHz. This suggests that the key device behavior such as h_{fe} can be simulated quite well with high frequency simulations using PISCES. Figure 4 shows the Smith chart of S -parameters at $V_{DS} = 2\text{V}$ and $V_{GS} = 0\text{V}$. In this case, however, there are some discrepancies between the measurement and simulation results. The fact that S_{11} goes into the upper half circle at high frequencies is a clear indication of inductive behavior which can only be attributed to the wire inductance and is not included in the PISCES simulation. To verify this point, MESFET is modelled with additional parasitics around the intrinsic device and the complete circuit model is shown in Figure 5. By using the y -parameters from PISCES simulation for the intrinsic device, inductance of 0.075 nH have been added to the gate and drain electrodes (L_g and L_d in Figure 5). The simulation results thus obtained are shown in Figure 4 with squares. Indeed a much better agreement is achieved.

Conclusions

A robust iterative algorithm for high frequency, small-signal ac analysis is presented, which combines a novel

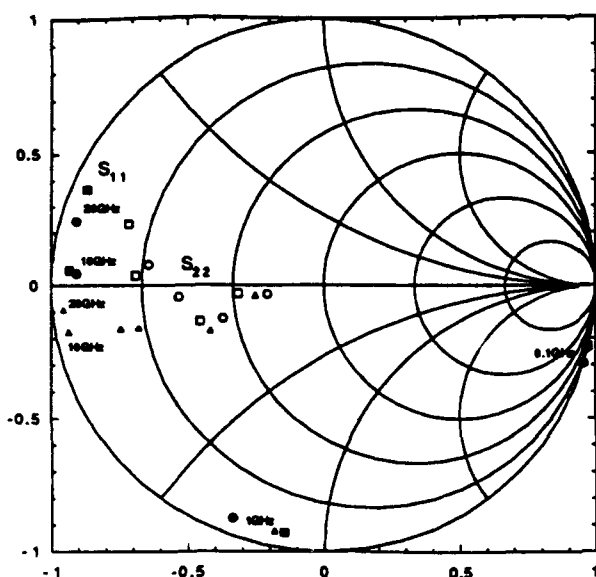


Figure 4: Comparison of measured and simulated S_{11} and S_{22} at $V_{DS} = 2V$ and $V_{GS} = 0V$. The dark symbols are for S_{11} and open ones for S_{22} . Circles are those from measurement, triangles from PISCES simulation for the intrinsic device, and squares from the mixed device/circuit simulation including lead parasitics.

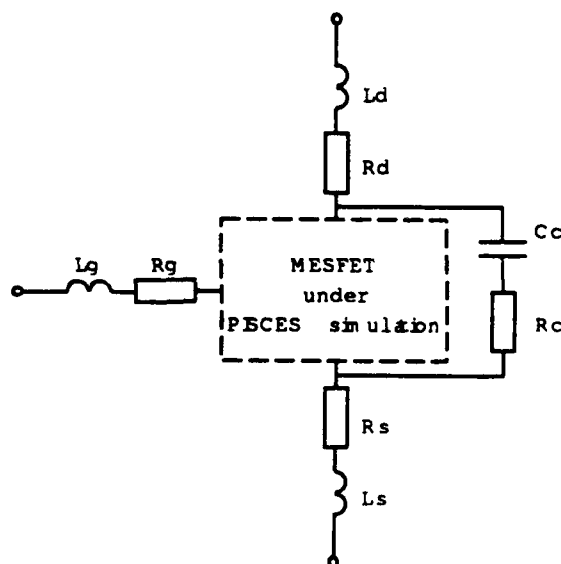


Figure 5: MESFET Model with parasitics.

matrix transformation scheme and a memory-efficient iterative method - TFQMR (Transpose-Free Quasi-Minimal Residual). Excellent convergence behavior is demonstrated for frequencies up to 10^{15} HZ without requiring increase in array size used in dc analysis. High frequency analysis of high-speed GaAs MESFETs has been performed using the new algorithm and the essential behavior of the device is well emulated. By adding parasitic elements to the intrinsic device, better agreement is obtained in the S -parameters of the device.

Acknowledgment

The authors would like to thank Mr. M. Maeda, Mr. Y. Oda and Mr. O. Ishkawa of Semiconductor Research Center, Matsushita Electric Industrial Co. for valuable discussions and MESFET measurements, Mr. Xiaowei Zhan of Stanford University for insightful discussions on QMR algorithm. The first four authors also acknowledge the support of U.S. Army Research Office under Contract DAAL03-91-G-0152 and U.S. Defense Advanced Research Project Agency under Contract DAAL-01-91-K-0145.

References

- [1] S. E. Laux, *IEEE Trans. Electron Devices*, vol. ED-32, No. 10, pp. 2028-2037, 1985.
- [2] Z.-Y. Wang, K.-C. Wu and R. W. Dutton *IEEE Trans. CAD*, vol. 11, No. 10, 1992.
- [3] D. R. Apte and M. E. Law, *IEEE Trans. CAD*, vol. 11, No. 5, pp. 671-673, 1992.
- [4] R. W. Freund and N. M. Nachtigal, RIACS Tech. Rep. 90.51, NASA Ames Res. Ctr., Dec. 1990.
- [5] R. W. Freund, RIACS Tech. Rep. 91.18, NASA Ames Res. Ctr., Sept. 1991.

Integrated TCAD for OEIC Applications ^[10]

R. W. Dutton, F. Rotella, Z. Sahul, L. So, and Z. Yu

Center for Integrated Systems
Stanford University, Stanford, CA 94305

ABSTRACT

Progress in technology CAD (TCAD) tools that span the range of IC circuit design, semiconductor device modeling, and material processing simulation is discussed. Special emphasis is given to the application in the opto-electronic devices and systems. Promising techniques for the mixed device/circuit simulation are presented using UC Berkeley's SPICE and Stanford's PISCES programs. Results show the capability to optimize the circuit timing and impedances to match the internal physical effects and the resulting optical properties in AlGaAs based LEDs. 3D solid modeling, a new approach for interactive device design based on the mask layout information, is discussed. Automatic gridding to support a unified process/device modeling capability for GaAs structures using SUPREM-IV.GS is also introduced. Taken as a whole, the unification of these TCAD tools provides a powerful design approach for the world of the opto-electronic ICs (OEIC).

1 Introduction

Application of semiconductor lasers and LEDs in all aspects of electronic systems is growing rapidly. Especially in the context of data communications, the need for coupling electronic circuit and optical device design is critical. Based on the experience of modeling silicon IC processes and devices over the past two decade, a new TCAD tool kit targeted at the opto-electronic applications is being developed at Stanford in close collaboration with industrial partners such as HP. In this paper, we will address the major issues related to the modeling of opto-electronic devices and the simulation of interaction between electronic circuits and opto-electronic devices. These topics are: a unified model for carrier transport in heterostructure with varying lattice/carrier temperature, implementation of 2D and quasi-3D device simulation for OE-ICs, framework issues related to gridding and solid modeling, and mixed-mode device/circuit simulation. Application examples will be provided.

2 DUET Model for Carrier Transport in Heterostructures Including Temperature Effects

This section considers the electrical modeling of the opto-electronic devices, a key factor in overall efficiency of OEICs. To describe the carrier transport in a heterostructure the additional driving forces caused by

changes of band parameters such as the effective mass, electron affinity, and bandgap, have to be taken into consideration. Moreover, the effects of carrier degeneracy (i.e., Fermi-Dirac statistics as opposed to the Boltzmann or Maxwellian distribution) and the lattice/carrier temperature play an even greater role in compound semiconductors than for the elemental semiconductors such as silicon because of the poor thermal conductivity and smaller effective density of states. Following Stratton's original approach [1], we have developed a general transport model based on the moment approach to solving Boltzmann transport equation (BTE) to describe the carrier transport in heterostructures with temperature effects included. Because this transport model covers both the carrier and lattice temperatures, we name the model as DUET for dual energy transport.

For complete modeling, six variables are needed to determine uniquely the device state and they are: electrostatic potential (ψ), electron and hole concentrations (n and p), electron and hole carrier temperatures (T_n and T_p), and the lattice temperature (T_L). Carrier temperature is defined through the temperature parameter in the statistics for carriers at quasi-thermal equilibrium, which is the Fermi-Dirac distribution in our model, and the lattice temperature is a measure of the kinetic energy of phonons. The distribution function of carriers (f), in the real (r) and momentum (p) space at any given non-equilibrium state is constructed using the quasi-thermal equilibrium distribution function (f_0) as follows.

$$f(r, p) = f_0(r, p) - \tau(r, p) \left(\frac{p}{m^*} \nabla f_0 - \frac{q}{m^*} F \cdot p \frac{\partial f_0}{\partial E'} \right) \quad (1)$$

where a constant effective mass (m^*) is assumed, and E' is the kinetic energy and for electrons

$$E'(r) = \frac{p^2}{2m^*} = E - E_C(r) \quad (2)$$

with E the total energy and E_C the energy at the conduction band edge. f_0 is taken from the Fermi-Dirac statistics as

$$f_0 = \frac{2}{1 + e^{[E - \zeta(x)]/k_B T_n(x)}} \quad (3)$$

where both the quasi-Fermi energy (ζ) and electron temperature are position-dependent, and k_B is the Boltzmann constant. F is the field acting on the charged carrier and for electrons

$$F = \frac{1}{q} \nabla E_C(r) \quad (4)$$

The formulation for holes are completely analogous. τ in Eq. (1) is the microscopic relaxation time which depends in principle on r and p , but is usually assumed to only depend on the carrier energy, E , and some other position-dependent scattering mechanisms such as impurity scattering.

Based on the knowledge of the distribution function, physical quantities such as the carrier concentration (n), the kinetic energy density (w_n), the electric current density (j_n), and the kinetic energy flux density (s_n) can all be found from their definitions. Thus,

$$n = \int f d^3p = \int f_0 d^3p = N_C F_{1/2}(\eta) \quad (5)$$

$$w_n = \int f \frac{p^2}{2m^*} dp^3 = \int f_0 \frac{p^2}{2m^*} dp^3 = \frac{3}{2} n k_B T_n \quad (6)$$

$$j_n = -\frac{q}{m^*} \int f p d^3p$$

$$= qD\nabla n - \frac{3}{2}qnD\nabla \ln m^* + n\mu\nabla E_C + qnD^T\nabla T_n \quad (7)$$

$$= n\mu\nabla \zeta + qn\mu Q\nabla T_n \quad (8)$$

$$\mathbf{s}_n = \frac{1}{m^*} \int f \mathbf{p} E' dp^3 = -P_n T_n \mathbf{j}_n - \kappa_n \nabla T_n \quad (9)$$

where

$$\eta = [\zeta(x) - E_C(x)]/k_B T_n \quad (10)$$

F_ν is the Fermi integral of order ν , and all other symbols and coefficients have conventional meaning used in solid state physics and can be derived knowing the relaxation time and distribution function.

The equations to be solved are the Shockley semiconductor equations, which include the Poisson's equation and carrier continuity equations, and the energy balance equations based on the Fick's second law for the kinetic energy of carriers and the lattice. The complete set of equations are listed below:

$$\nabla \cdot (-\epsilon \nabla \psi) = q(p - n + N_D^+ - N_A^-) \quad (11)$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{j}_n - u \quad (12)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{j}_p - u \quad (13)$$

$$\begin{aligned} \frac{\partial w_n}{\partial t} = & -\nabla \cdot \mathbf{s}_n + \mathbf{j}_n \cdot \mathbf{F}_n - (u_{SRH} + u_{rad})\gamma(\eta_n)\frac{3}{2}k_B T_n + (u_{n,Auger} - g_{n,imp}) \\ & \left[E_g(T_L) + \gamma(\eta_p)\frac{3}{2}k_B T_p \right] + g_{p,imp}\gamma(\eta_n)\frac{3}{2}k_B T_n - \frac{w_n - w_{n0}(T_L)}{\tau_{wn}} \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial w_p}{\partial t} = & -\nabla \cdot \mathbf{s}_p + \mathbf{j}_p \cdot \mathbf{F}_p - (u_{SRH} + u_{rad})\gamma(\eta_p)\frac{3}{2}k_B T_p + (u_{p,Auger} - g_{p,imp}) \\ & \left[E_g(T_L) + \gamma(\eta_n)\frac{3}{2}k_B T_n \right] + g_{n,imp}\gamma(\eta_p)\frac{3}{2}k_B T_p - \frac{w_p - w_{p0}(T_L)}{\tau_{wp}} \end{aligned} \quad (15)$$

$$\begin{aligned} c_L \frac{\partial T_L}{\partial t} = & \nabla \cdot (\kappa_L \nabla T_L) + u_{SRH} \left[\gamma(\eta_n)\frac{3}{2}k_B T_n + E_g(T_L) + \gamma(\eta_p)\frac{3}{2}k_B T_p \right] \\ & + \frac{w_n - w_{n0}(T_L)}{\tau_{wn}} + \frac{w_p - w_{p0}(T_L)}{\tau_{wp}} \end{aligned} \quad (16)$$

where T_L is the lattice temperature, and τ_{wn} and τ_{wp} are energy relaxation times for electrons and holes, respectively, and they characterize the carrier kinetic energy loss due to the phonon scattering.

The direct energy exchange between the two carrier subsystems is through Auger recombination and impact ionization, while the exchange among carrier subsystems with lattice is through Shockley-Read-Hall (SRH) recombination and phonon scattering. The carriers also release energy to emit photons through radiative recombination. The generation of electron-hole pairs due to the absorption of photons is not taken into account in this formulation. For the simulation of laser diodes, often the carrier temperature is not a major concern but the lattice temperature is. One can then set $T_n = T_p = T_L$ and Eqs. (14)–(16) collapse into the following form:

$$\left[c_L + \frac{3}{2}(n + p)k_B \right] T = \nabla \cdot (\kappa_L \nabla T - \mathbf{s}_n - \mathbf{s}_p) + \mathbf{j}_n \cdot \mathbf{F}_n + \mathbf{j}_p \cdot \mathbf{F}_p + u_{SRH} E_g(T) \quad (17)$$

where $j_n \cdot F_n + j_p \cdot F_p$ can be identified as the Joule heat which represents the conversion of the electrostatic potential energy to the carrier kinetic energy, and the term $u_{SRH} E_g(T)$ represents the conversion of the carrier potential energy to the phonon (kinetic) energy through SRH recombination.

The above equations and expressions are good for any dimensionality, either 1D, 2D, or 3D. In order to solve the system of partial differential equations as described in this section, proper boundary conditions are critical – especially for the thermal simulation. We will not detail the discussion here, but the interested readers are referred elsewhere [2].

3 Model Implementation in 2D and Quasi-3D (Cylindrical Symmetry)

The DUET model has been implemented in Stanford's PISCES-2ET. The results for the electrical simulation of heterostructure devices such as HBTs, LEDs, and SELs (surface emitting lasers) have been compared to the measurement data. Currently four material systems are available: ternary compounds $Al_xGa_{1-x}As$ and $Al_xIn_{1-x}As$, quaternary compound $Ga_xIn_{1-x}As_yP_{1-y}$, as well as binary compound system for Ge_xSi_{1-x} . The base materials corresponding to these compounds are GaAs, InAs, InP, and Si. The material parameters have been calibrated either through values available in the literature or by the measured data obtained from HP. More than one material system can be incorporated in a single device region. Non-stationary transport mechanisms, i.e., those where the concept of Fermi level doesn't apply, such as the thermal emission and tunneling through an abrupt barrier have not been implemented currently.

Although PISCES-2ET is a 2D device simulator in general, in the special case of cylindrical symmetry, 3D simulation can also be performed. This applies to the simulation of light emitting diodes (LEDs) which have cylindrically symmetrical structure. The algorithm for the cylindrical coordinate simulation is to sweep the element in the 2D cross section around the central axis and because the flux calculated in 2D plane is always perpendicular to the surface of the 3D elementary volume, the assembly of the control volume equation is essentially the same as in the 2D case. Since there is no approximation involved, it is a rigorous 3D simulation in this case.

In the following we show the simulation of a LED with cylindrical symmetry. The half structure is shown in Figure 1, and the center of the diode is on the right side. The anode contact is on the bottom and the cathode contact is on the top of the structure with a ring shape to allow light to be emitted from the central opening of the diode. The unique feature and the challenge to the numerical simulation is the existence of a floating n^+ layer in the p -type region. The function of this floating layer is to confine the current flow in the central region so that the emitting efficiency will be enhanced, and the determination of its shape, location, and doping level is one of the key issues in LED design. From the numerical point of view, the heavily doped floating layer essentially blocks the flow of the majority carriers through it and hence can cause severe numerical instability. There are several ways to get around of this problem; some involve elaborate algorithms. In our current approach, we simply attach an external contact to the layer and connect a current source to the ground node. If the current source is set to zero value, it is essentially an open circuit condition. This scheme works very well when the diode current has a substantial value, i.e. for the "on" state. When the diode is turned off, the convergence of numerical iterations becomes more difficult but the exact results have less concern to the designer. The simulated $I - V$ characteristics are shown in Figure 2. In the later section, we will show how this LED is used in a transmitter circuit for data communication and the mixed-mode device/circuit simulator will be invoked to find the frequency

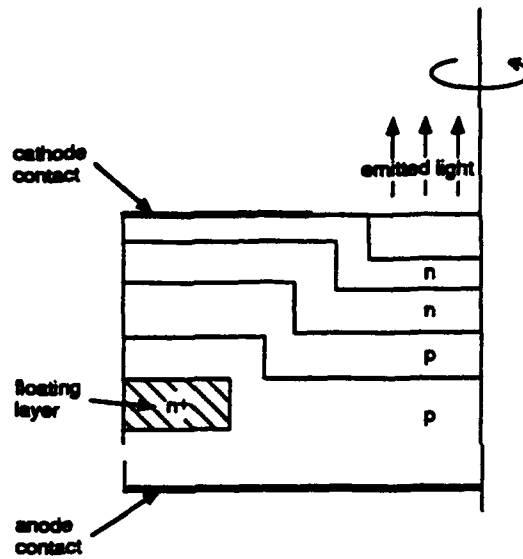


Figure 1: Device structure of cylindrically symmetric LED. The symmetric axis is the right edge of the simulation region.

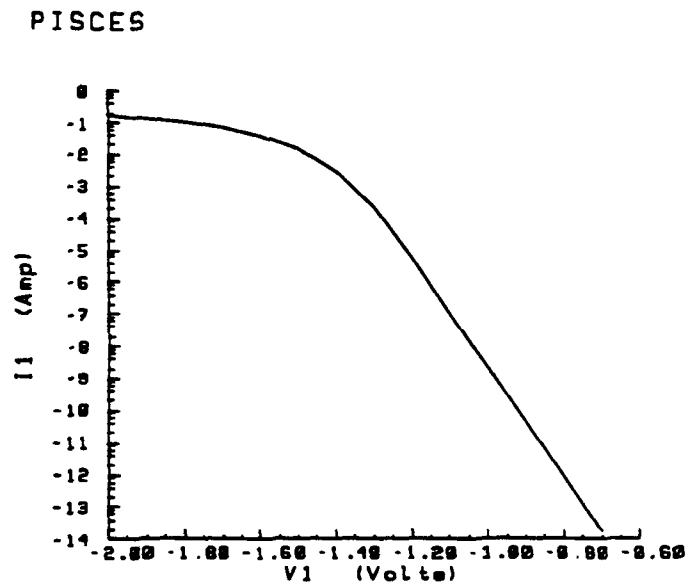


Figure 2: Simulated $I-V$ characteristics of LED. Notice that the bias is applied to the top of the structure (n -layer), so it is negative.

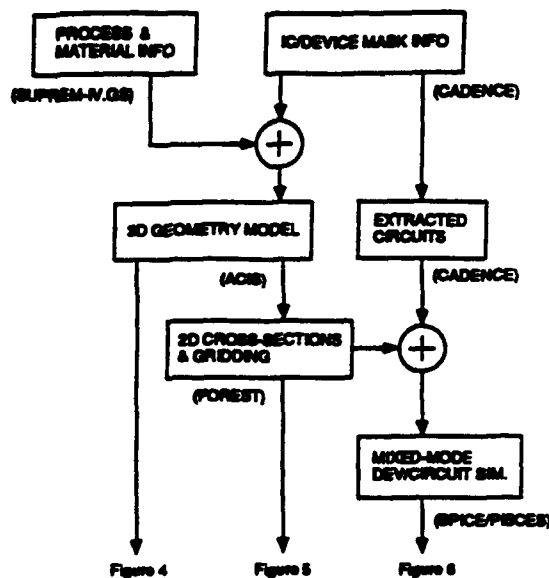


Figure 3: The framework of tools used in IC/OEIC device/circuit design.

response.

4 OEIC Design Framework – 3D Solid Modeling and Gridding

The design of OEICs requires not only the consideration of devices with complex material structures as shown in Figure 1 but also the integration of circuit functionality that meets the electrical constraints needed for proper system operation. Moreover, there is growing emphasis on tightly integrated systems and even monolithic solutions. For these purposes the use of device/IC mask layout information as well as 3D solid geometry models are of great interest as utilities for both visualizing and manipulating the designs. Figure 3 shows key portions of a design framework for OEIC applications that utilizes both commercially available tools and custom utilities for supporting functional analysis of an OEIC subsystem. The layout, solid modeling, and circuit extraction information come from commercial tools (i.e., Cadence and ACIS). Discussion of process modeling is beyond the scope of this paper and relevant materials can be found in [3]. The following discussion focuses on the issue of solid geometry modeling and gridding. The next section considers the issues of mixed-mode simulation.

As indicated in Figure 3, 3D structures created by solid modeling can be of great value in designing OEICs. The concept of the solid modeling includes the visualization of the actual structure with constituent layers in a 3D perspective, meshing on the surface, dissection of the structure from different view angles, and the simulation of the structure either in a 2D cross section or with a truly 3D simulation. Solid modeling has been very useful in diagnosis of high density structure such as memory cell design and in the analysis of parasitic components that are frequently encountered in high speed microwave circuits.

Figure 4 shows a 3D rendering of a six transistor (6T) SRAM cell where the interconnection layers are easily recognized. This figure clearly indicates the complexity of the structure that can easily be created and analyzed using the utilities and information models presented in Figure 3.

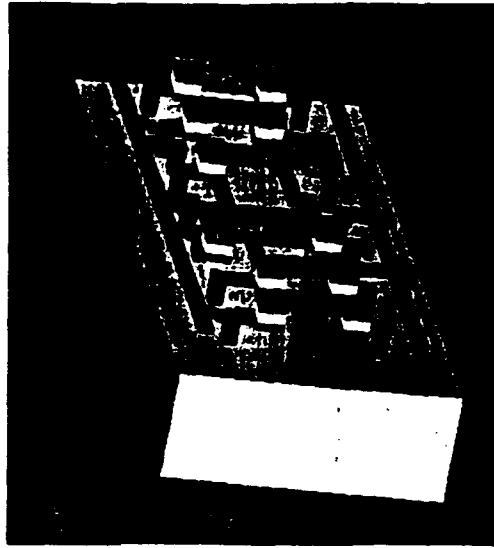


Figure 4: 3D rendering of six-transistor SRAM cell using solid modeling. The cut plane is for the 2D cross section later used for device simulation.

One of the major obstacles towards the 3D device/process simulation is the discretization of the object in space, i.e. gridding or meshing. Especially for the opto-electronic devices, there are multi-layers in the structure with different material properties and doping levels. The interface between different layers often changes abruptly. We have developed various automatic mesh generators to facilitate the simulation of IC devices and processes. The most recent development is a program called FOREST [4], which is targeted at the auto-meshing of complex multi-layer device and process simulation. In the following we describe briefly the algorithm and feature of the program. An illustrative example will be given to show the steps in meshing process and the capability of the program.

FOREST is based on the quadtree decimation algorithm, which guarantees high quality of the leaf elements – triangles in the 2D simulation. FOREST considers only 2D structures, and all triangles generated from it are either acute or right with aspect ratio as close to one as possible. Fineness of the space partition is controlled and can be dynamically adapted/refined according to a number of criteria such as the doping profile and solution error tolerance. Besides its application in device simulation, for which the boundary of the structure is usually fixed, the program is also suitable for the multi-layer process simulation where the topography undergoes rapid changes (e.g. SUPREM-IV.GS). A graphics user interface is also provided with the program.

Figure 5 shows how FOREST grids in a step-by-step manner a complex structure with non-planar surfaces and even interior (or back-side) regions. This particular example comes from silicon process simulation (i.e., SUPREM-IV.GS) and gives emphasis to the need for not only boundary definitions but also wide dynamic range in feature sizes. The issue of gridding backside or interior regions (i.e. air bridges) can be especially important for multi-chip and other packaging concerns such as OEIC connectors.

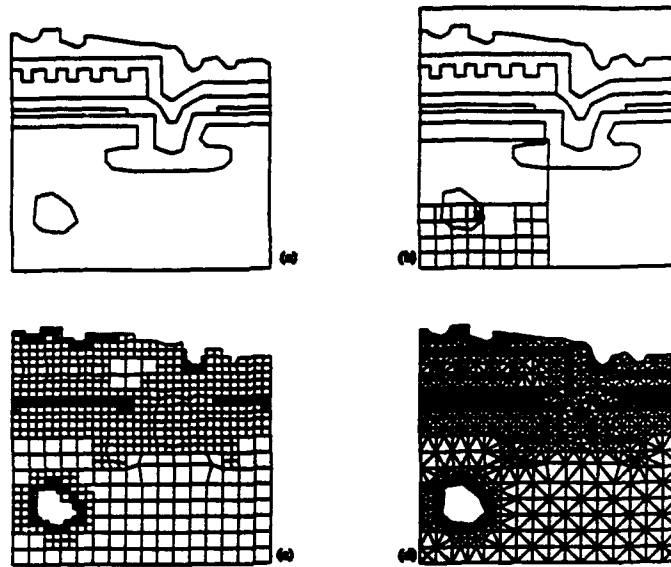


Figure 5: Process of FOREST mesh generation for a complex device structure with hole in it.

5 Mixed Device/Circuit Simulation of Optical Data Communication

The performance of an electronic/opto-electronic system depends not only on that of the individual devices, but also on the optimized design of peripheral driving circuitry and minimization of effects from parasitics. Thus to evaluate a system design circuit level simulation is required. However, the circuit simulation uses analytical (compact) models for all non-linear devices, which are not always available when a new device is emerging. Especially for opto-electronic devices where the optical performance is based on the electrical response, and compact models are often either not yet developed or not easily characterized. We have developed a modularized mixed device/circuit simulator where the device simulator provides a means to model complex device structures numerically in concert with the circuit simulation [5]. The shell of the mixed simulator is SPICE 3 from UC Berkeley, and device simulator is "plugged in" as a loosely coupled module - currently PISCES-2ET is used in this work, but other device simulators can also be used as long as they provide not only the DC $I - V$ characteristics but also the slope (more broadly, y -parameters) information.

Although, several examples have been demonstrated for mixed device/circuit simulation from various sources, either academically or industrially, they all focus on electronic components. We present here a simulation of drive circuitry in an optical transmitter using an LED to link to optical fiber. The key requirement of this configuration is the fast optical response of the LED, which can be characterized by the conductive (diffusive) current component in the ac/transient response. The detailed circuit is shown in Figure 6 and the LED has the same structure as the one used in the device simulation. The exact structure is used in the mixed simulation. The LED is driven by the ECL output either as sinusoidal or square wave. The waveform for the current through the LED from the time transient analysis is shown in Figure 7 and comparison is made to the analytical diode model. In this case the compact model has been experimentally optimized so the fit with the numerical model is quite good. For new generations of technology where experimental data is not available, the use of this mixed-mode analysis is more critical.

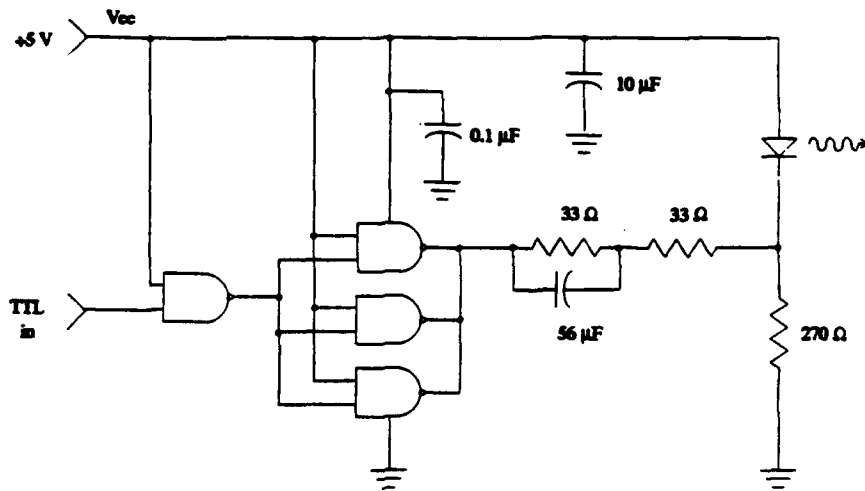


Figure 6: Driving circuit for LED linked to optical fiber for high-speed data communication (courtesy of HP).

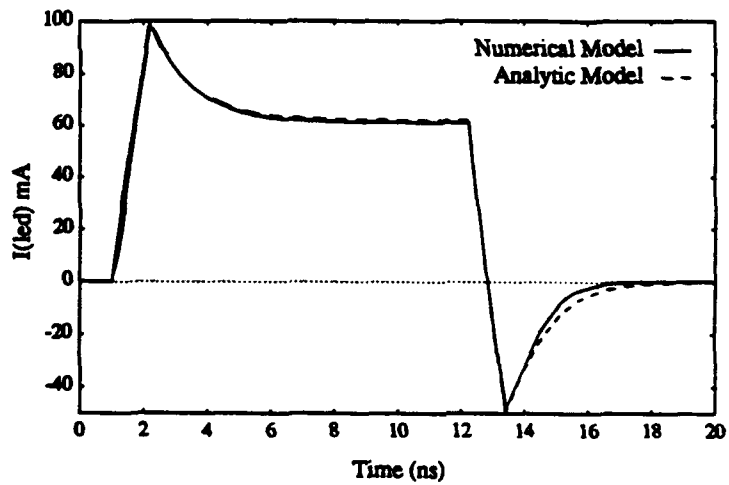


Figure 7: Comparison of simulated waveforms from analytical and numerical models in circuit simulation.

6 Conclusions

This paper has considered a range of issues involved in the simulation of OEIC devices and circuits. Starting from the issues of electrical behavior of opto-electronic devices, the details of a new dual energy transport (DUET) model have been presented. Key features of this new capability include the ability to couple lattice temperature with the carrier temperature system of equations. Especially for compound material systems with poor thermal conductivity, this coupled analysis is of growing importance. Another unique analysis capability that is demonstrated in this paper is that of mixed-mode simulation where optoelectronic effects can be simulated at the device level in concert with conventional circuit simulation of the purely electrical effects. In this way the details of the optical and electronic couplings can be analyzed and correlated directly with the materials and geometry effects rather than by means of compact equivalent circuit models that are difficult to characterize. Finally, some of the issues related to OEIC design frameworks have been presented. Automated gridding of devices and 3D geometry modeling that facilitates visualization and extraction of parasitic interactions have been presented. The information models for such applications and the availability of commercial tools (ie Cadence, ACIS, AVS etc.) provide a powerful environment from which to develop a more specific set of applications and other utilities to support the development of OEICs. This paper and the associated examples provide motivation to go further in the development of OEIC design system based on the growing power of such information models and tools.

7 Acknowledgement

This work has been supported through the Army Research Office (ARO contract DAAL03-91-G-0152) and ARPA/MTO (contract DAAL01-91-K-0145). Much of the framework and gridding efforts has been supported through SRC and ARPA contracts in Manufacturing Science.

Authors would like to thank our industrial collaborators from HP, and they are Drs. K. L. Chen, Jim Prettyleaf, and Bob Weissman of Optoelectronics Component Division, San Jose, and Drs. Mike Tan and W. Ishak of HP Labs, Palo Alto.

References

- [1] R. Stratton, "Semiconductor current-flow equations (diffusion and degeneracy)," *IEEE Trans. Electron Devices*, Vol. ED-19, pp. 1288-1292, Dec. 1972.
- [2] R. W. Dutton and Z. Yu, *Technology CAD - Computer simulation of IC Processes and Devices*, Appendix A, Kluwer Academic Pub., Boston, 1993.
- [3] *SUPREM-IV.GS - Two Dimensional Process Simulation for Silicon and Gallium Arsenide*, Ed. S. E. Hansen and M. D. Deal, IC Lab, Stanford University, Stanford, CA 94305.
- [4] Z. Sahul, E. McKenna, and R. Dutton, "Grid techniques for multi-layer process and device simulation," *TECHCON '93*, Atlanta, GA, Sept. 1993.
- [5] Z. Yu, R. W. Dutton, and H. Wang, "A modularized, mixed IC device/circuit simulation system," *Proceedings SASIMI '92*, pp.444-448, Kobe, Japan, April 1992.

Algorithms and TCAD Software using Parallel Computation [11]

Robert W. Dutton
Stanford University
Stanford, California 94305-4055

Abstract

Progress in algorithms and simulators that exploit parallel computers are reported. Results using two generations of the Intel iPSC architecture for device analysis of 2D and 3D bipolar problems are used to illustrate substantial progress being made in parallelization. New approaches in the areas of hydrodynamic and Monte Carlo analysis are also discussed.

Introduction

Over the past decade the computational demands of TCAD have grown as have the power and availability of new parallel computers. Specifically, requirements for large 3D process and device simulations with grid numbers reaching several million have become essential---especially to analyze parasitic effects where multiple devices are involved. On the hardware side, tightly coupled shared memory machines (Cray and Alliant for example) have given way to distributed memory machines (Intel and Thinking Machines) and now experimental architectures that attempt to provide distributed and shared memory---for example the DASH Project at Stanford are being pursued. The focus of this talk is primarily aimed at summarizing experiments at Stanford in 2D and 3D device simulation made over the past several years using parallel computers. This involves a variety of algorithmic approaches including: drift-diffusion, Monte Carlo and most recently hydrodynamic formulations. The primary class of computers used in this study was the Intel iPSC series (including the commercial i860 and the experimental "Delta" machine at Caltech). Results show very promising performance and potential for achieving TeraFLOP performance using 1000s of processors within a few years.

Physical Models and Algorithms

There is a range of physical models that can be applied to semiconductor device modeling problem. Basically there are three major classes of ways to solve the Boltzmann Transport Equations using:

1. The first moment (drift-diffusion or D/D)
2. Higher order moments (hydrodynamic(HD) , energy transport (ET)...))
3. Monte Carlo (MC) as a means to directly build the statistics.

In this talk we will give examples of how each of these approaches can exploit parallel computation.

Drift-diffusion (D/D)---In the case of the classical D/D formulation, we have previously reported results on parallelization of the 3D STRIDE code. Here we report parallelization of the well-known 2D PISCES code. Figure 1a illustrates all steps needed to parallelize the PISCES code and the associated percentages of time required in the simulation. Clearly, the matrix solution part is a major but not dominant fraction. Also, one can note that parallelization of assembly of matrix elements is every bit as important in the overall effort. Figure 1b shows a profile of actual run times (wall clock time) versus the number of processors on the Intel iPSC/i860 for two problems. For comparison the times on a single CPU SUN 4/670 are also listed. Note that for the 27,600 equation problem, the speed-up factors are substantial. Also note that beyond 16 nodes for this particular problem the speed-up factor degrades due to communications limitations. Further details will be discussed in the presentation.

In previous papers we have discussed progress in 3D D/D modeling using the prototype STRIDE code [1][2]. Over the past year we have scaled the problem size and number of processors used in the analysis to much larger numbers than previously reported. Figure 2 shows the wall-clock time and grid density (two y-axes) versus the number of processors used on the Delta machine at Caltech. This is an experimental version of the Intel iPSC architecture with a backplane interconnect designed at Caltech. As can be observed from the data, we see excellent performance and efficiency even up to more than 500 processors and grid density approaching 5 Million for a bipolar example. The sustained performance was 1.7 GFlops. In this talk we will discuss some of the algorithms used with regards to preconditioning that are essential in achieving such favorable results.

Hydrodynamic (HD) and beyond---The limitations of the D/D formulation are well-known and more advanced models such as the HD or ET formulations are being used extensively. From a physical modeling point of view we continue to extend the ET approach [3]. From a numerical point of view we are investigating the HD formulation in the context of major advances in the CFD domain [4]. Specifically, we are using the time-space GLS formulation in concert with the HD carrier transport equations and developing a parallelized version of the ENSA solver code (Euler Navier-Stokes Analyzer). Since a paper on this subject has been submitted to this conference, we will only briefly highlight the results.

Monte Carlo (MC)--- is a still more complete physical approach to solving the BTE. While there are many limitations in applying MC analysis due to boundary conditions, especially across heterojunction and dielectric interfaces, there are many aspects of hot carrier phenomena that can be effectively solved using MC analysis. At SISDEP we have presented results of efforts in parallelization of MC using the University of Bologna's BEBOP code [5]. Further efforts in this direction are being reported by the Matsushita group. The basic factors affecting parallelization of MC analysis are straightforward from an algorithmic point of view. On the other hand, there are still many innovations that are emerging to use either harmonic expansions or alternate formulations for the variables, including symmetry relationships. The discussion of MC analysis will center on promising new approaches.

Discussion

The above examples illustrate the progress that is being made at three levels in developing algorithms that exploit new parallel computers. The efforts in parallelization of the classical D/D formulation show that substantial speed-ups and robust convergence on very tough bipolar problems can be achieved well into the multi-million grid domain. As for higher moments of the BTE, we see two important trends. First, the development of new algorithms such as the GLS formulation used in CFD and its parallelization should yield major benefits. Second, the opportunities to advance Monte Carlo methods are far from "out of gas" in terms of innovation and computational enhancements. A final area of discussion in this paper is the growing importance of overall support in the development and parallelization of TCAD codes. The most obvious of these needs is in the area of gridding and the domain decomposition. Other areas include: user interfaces, control of both nonlinear and linear solvers and alternative formulations. These issues are related to frameworks, standards and code development strategies.

Conclusions

Progress in developing TCAD algorithms and particularly device analysis codes using parallel computation is reported. Excellent efficiency and convergence behavior on 2D and 3D bipolar problems is demonstrated using production and experimental versions of the Intel iPSC architecture. Progress in solving higher moments of the BTE are reported and new algorithmic advances are discussed. Finally, challenges of overall software engineering are considered.

Acknowledgments

This work has been supported by not only government agencies (ARO and DARPA) but also through an industrial consortium (CompTech under the Department of Commerce, State of California). We especially appreciate the encouragement and ongoing support from the CSTO office of DARPA and Dr. Gil Weigand. The specific results quoted on STRIDE came from Dr. K.C. Wu, those on PISCES-MP from Mr. Bruce Herndon and the Aladdin-CAD project contributions came from Dr. Ronald Goossens.

References

- [1] K. C. Wu, G. R. Chin, R. W. Dutton, "A STRIDE Towards Practical 3D Device Simulation--Numerical and Visualization Considerations," *IEEE Trans. CAD*, Vol. 10, No. 9, pp. 1132-1140, Sept. 1991.
- [2] Z. Y. Wang, K. C. Wu, R. W. Dutton, "An Approach to Construct Pre-Conditioning Matrices for Block Iteration of Linear Equations," *IEEE Trans. on CAD*, Vol. 11, No. 11, pp. 1334-1343, Nov. 1992.
- [3] D. Chen, E. C. Kan, U. Ravaioli, C. Shu and R. W. Dutton, "An Improved Energy Transport Model for Hot-electron Device Simulation," *IEEE Elec. Dev. Lett.*, vol. 13, no. 1, pp. 26-28, Jan. 1992.
- [4] N. R. Aluru, A. Raefsky, P. M. Pinsky, K. H. Law, R. J. G. Goossens, and R. W. Dutton, "A Finite-Element Formulation for the Hydrodynamic Device Equations," Accepted for publication in *Computer Methods in Applied Mechanics and Engineering*.
- [5] S. Sugino C.S. Yao and R.W. Dutton, "Parallelization of Monte Carlo Analysis on Hypercube Multiprocessors and on a Networked EWS System," *SISDEP '91 Digest*, Vol. 4, pp. 275-284, Zurich, Sept. 1991.

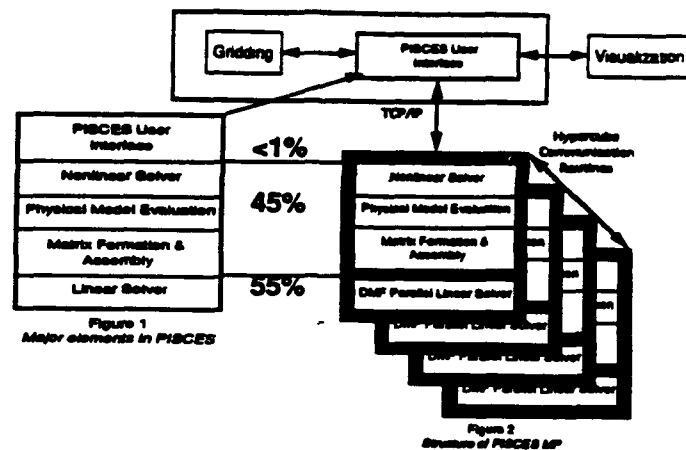


Figure 1(a): Structure of PISCES-MP showing how different portions of the code affect execution and resulting parallelization requirements

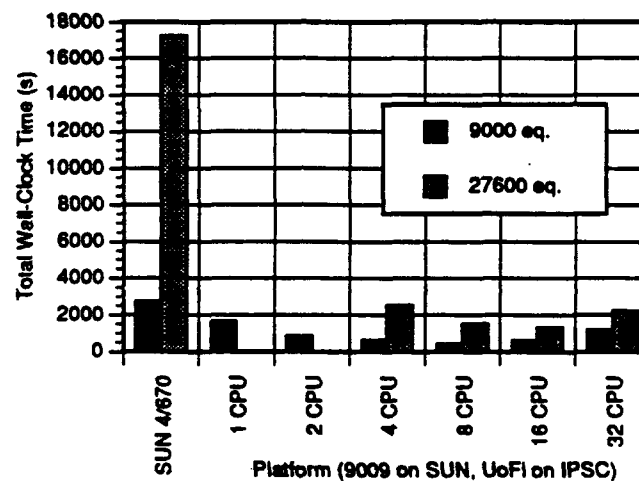


Figure 1(b): Performance of PISCES-MP showing wall-clock time versus number of CPUs on the Intel iPSC/1860 for two bipolar problems (9000 and 27,600 equations). Benchmarks for same problems on a SUN 4/670 are also shown.

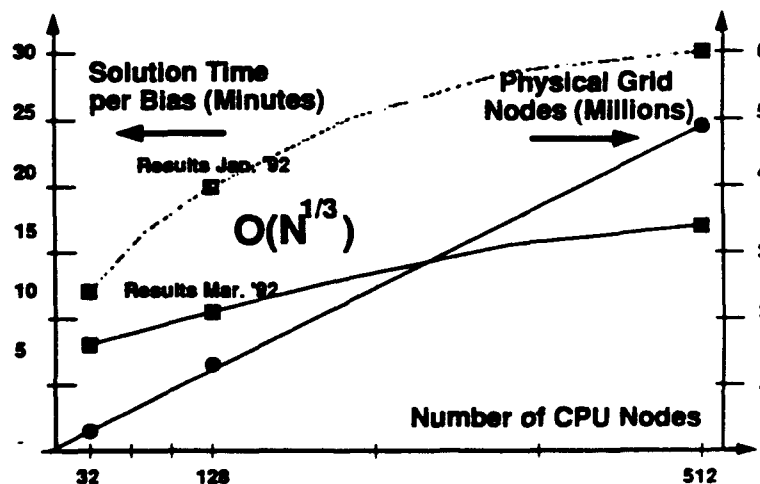


Figure 2: Solution time and number of grid in millions as a function of number of CPU nodes on the Intel Delta machine at Caltech. The problem is a 3D bipolar simulation and the code used in the experimental STRIDE program [1].

A STRIDE Towards Practical 3-D Device Simulation—Numerical and Visualization Considerations^[13]

Ke-Chih Wu, *Member, IEEE*, Goodwin R. Chin, *Student Member, IEEE*, and Robert W. Dutton, *Fellow, IEEE*

Abstract—A 3-D device solver (STRIDE), capable of solving grids up to 250 000 nodes, has been developed on a message passing multiprocessor. By the use of iterative matrix solvers and Gummel style nonlinear iteration schemes, user memory per node is reduced over use of direct solvers and Newton schemes. By using an independent-edge-grouping scheme to increase the vector length to the order of the number of variables, the vector processing efficiency is significantly increased without additional floating point operations. We extend the modified-singular-perturbation (MSP) scheme to two-carrier simulations. This significantly speeds up the convergence rate of Gummel style nonlinear iterations. Physical insight gained from the MSP schemes also leads to an automatic switching scheme between various nonlinear schemes based on the monitoring of certain matrix parameters. This allows the incorporation of a previously proposed Newton-1C scheme which offers the best CPU performance for normal bipolar simulations. When combined with current convergence criterion, a set of MSP inspired convergence criterion are better able to recognize a practically converged solution. A novel global convergence scheme is also developed based on insight from MSP principles. Interactive user interface and links to graphics tools are provided to support the tool integration efforts. Application of STRIDE is demonstrated by an analysis of latchup trigger current dependence on layout arrangement.

I. INTRODUCTION

WITH THE continuing miniaturization of integrated circuits, 3-D effects significantly impact device characteristics. A robust and efficient 3-D device solver will give device engineers significant leverage in pursuing state-of-the-art IC technologies. Various 3-D simulators (e.g., [1]–[4]) have appeared to address these needs. However, one of the major hurdles which has prevented widespread use of 3-D device simulation is the vast amount of computational resources required for such an endeavor, as the number of variables can easily run into hundreds of thousands, or even millions. Multiprocessors, which connect together a large number of inexpensive processors, provide a cost-effective platform for CPU-intensive 3-D simulations. To explore the potential

for a cost effective 3-D device simulator, we have developed STRIDE (Stanford ThRee dIimensional DEvice simulator) on a message passing multiprocessor (Intel iPSC2). This paper describes the progress that has been made since the previous report [5], and mainly centers around the various computational aspects with special emphasis on bipolar simulations. Our experience in 3-D visualization is also discussed.

Section II gives an overview of the device solver. Section III discusses schemes which increase the vector length to the order of number of variables for the sparse matrices encountered in 3-D simulation. In Section IV, various modified singular perturbation (MSP) schemes are introduced for two-carrier simulations which significantly improve the convergence of Gummel style nonlinear iterations. The results of a previously proposed Newton-1C scheme will be presented which offers the best CPU performance with less memory than the full-Newton scheme. A MSP inspired matrix parameter will be introduced which allows a switching scheme that automatically chooses the best nonlinear scheme for the simulation. In Section V, other applications of MSP principles will be discussed which include a new set of convergence criterion capable of determining practically converged solutions and a novel global convergence scheme. Section VI discusses our approaches in developing better user interfaces and on tool integration aspects. Section VII, presents an application example of STRIDE in the analysis of the latchup trigger current's dependence on electrode arrangement. Finally, conclusions are drawn in Section VIII.

II. OVERVIEW OF STRIDE

In STRIDE, up to two current continuity equations are solved together with Poisson's equation. In normalized form, these equations are given by

$$\nabla \cdot (\epsilon \nabla \psi) = n - p + N_A - N_D \quad (1)$$

$$\nabla \cdot J_n - U = 0 \quad (2)$$

$$\nabla \cdot J_p + U = 0 \quad (3)$$

where $n = n_{ie} \exp(\psi - \phi_n)$, $p = n_{ie} \exp(\phi_p - \psi)$, $J_n = -q\mu_n n \nabla \phi_n$, and $J_p = q\mu_p p \nabla \phi_p$. The normalization constants used to obtain (1)–(3) are: thermal voltage (kT/q)

Manuscript received July 3, 1990. This work was supported by the Intel Corporation, Texas Instruments, and the U.S. Army Research Office under Contract DAAL03-87-K-0077. This paper was recommended by Guest Editor M. Law.

The authors are with the Integrated Circuits Laboratory, AEL 231 F, Stanford University, Stanford, CA 94305.

IEEE Log Number 9101058.

for electrostatic potential ψ and quasi-Fermi levels ϕ_n and ϕ_p , intrinsic carrier concentration n_i for carrier and impurity concentrations, and the intrinsic Debye length $\sqrt{\epsilon kT/q^2 n_i}$. Effective intrinsic carrier concentration n_{ie} is obtained using the Slotboom bandgap narrowing model [6]. Boltzmann statistics are assumed as can be seen in the formula for n and p . Tabulated doping-dependent mobility values are used. Tangential field dependent mobility is implemented with the Caughey-Thomas model [7].

The main development vehicle for STRIDE has been a message passing hypercube—the Intel iPSC2. The advantages of the hypercube architecture are that it scales to massively parallel systems and that the diameter of the system (average communication delay between the processors) grows only logarithmically with the number of processors. An important feature of the Intel hypercube is the amount of the memory per processor. The system used in this work has 16 processors, each with 8 Mbytes of memory. The sustainable performance for the system is about 1.5 MFlops [8] in which each processor constitutes an Intel 386 paired with a 387 math coprocessor. The development has recently been shifted onto an iPSC/860 system which has 32 processors each with 16 Mbytes of memory. Preliminary results have shown a system performance of approaching 100 MFlops. STRIDE also runs on Convex C1 and Cray YMP.

The simulation domain is currently approximated by a 3-D rectangular grid with provisions for nonplanar structure [5]. Work is going on to develop parallel algorithms for dealing with general grids generated by grid generators such as OMEGA [9]. Equations (1)–(3) are discretized using the finite difference method. In discretizing the continuity equations, Scharfetter-Gummel current formulation [10] is used.

The discretization of (1)–(3) yields a nonlinear system of algebraic equations which are solved by one of several nonlinear iteration schemes implemented in STRIDE. For each nonlinear iteration in Gummel's scheme, the discretized Poisson's equations ($F_\psi(\psi, \Phi_n, \Phi_p) = 0$) are solved for the update vector $\delta\psi$ holding Φ_n and Φ_p constant.¹ This is achieved by repeatedly solving

$$A_{\psi\psi}\delta\psi = -F_\psi \quad (4)$$

given the current estimate of ψ , Φ_n and Φ_p . In (4), $A_{\psi\psi} \equiv (\partial F_\psi(\psi, \Phi_n, \Phi_p)/\partial\psi)$ is called the main matrix of Poisson's equation. Other matrices are similarly defined. The discretized current continuity equations ($F_{\Phi_n}(\psi, \Phi_n, \Phi_p) = 0$ and $F_{\Phi_p}(\psi, \Phi_n, \Phi_p) = 0$) are then solved. Since $A_{\Phi_n\psi}$ and $A_{\Phi_p\psi}$ are linear, one matrix solution will suffice. This process repeats until convergence is achieved. For other nonlinear schemes, two of the equations are solved together while the other is solved separately. More details of these schemes are discussed in Section IV. The convergence criteria are the maximum magnitude of ψ updates, terminal current conservation, and relative change in the

magnitude of terminal currents and of terminal charges. Further discussions are deferred to Section V.

The matrix solutions are the most CPU time intensive steps in STRIDE. The incomplete Cholesky conjugate gradient (ICCG) algorithm [11] is used to solve the symmetric matrices, while asymmetric matrices are solved using the incomplete LU decomposition conjugate gradient squared (ILUCGS) algorithm [12]. The parallel implementation of these algorithms, which are based on domain decomposition, are described in [13] and [14]. The parallel efficiency achieved by these algorithms, while running on 16 processors, is more than 80%² when the problem size exceeds 50 000 nodes.

The maximum number of grid points that can be handled by STRIDE on the 16-node iPSC/2 system is over 100 000,³ which translates to a cubic grid of 47 points in each dimension. This is the direct result of not using the full-Newton scheme which would nearly double the memory per node. CPU time per bias point is about 1.5 h for a 70K node bipolar example. This is averaged from a I_c versus V_{be} curve with $V_{ce} = 5$ V. In this curve, V_{be} increases from 0.4 to 1 V in 0.1-V steps.

III. VECTORIZATION SCHEMES

Vectorization is an important aspect of reducing the execution time of the program. Since a majority of CPU time is spent solving matrices, our efforts have concentrated on vectorizing the iterative matrix solvers.

The principle behind the vectorization is to group together long chains of repetitive operations which are mutually independent. This independence is essential so that vector processing will not produce different results from the scalar operations. Thus the key to vectorization is to identify such groups of operations. For most iterative matrix solution algorithms, most of the operations involve vector-vector or matrix-vector products. Although the former is trivially vectorized, the latter takes some effort when the matrices are sparse. A matrix-vector product can be considered to be the sum of many vector-vector products which can be easily vectorized. This works well for dense matrices which have long rows. However, when the matrix is sparse, the length of these vectors becomes very short (typically, three to six) which seriously impedes vector processing performance.

One approach to increase the vector length is to split the matrices into many small dense matrices obtained from the elements of the simulation domain, such as triangle elements in 2-D simulation [15]. When two elements contain no common node, their matrices are independent and can be grouped together. This grouping can be called independent element grouping.

Building upon this idea, we implemented an independent edge grouping scheme. In terms of group theory, a

²Previously, we have reported a parallel efficiency of about 60% when the concurrent ICCG algorithm ran on iPSC. The improvement in efficiency is a result of the ten-fold improvement in the data latency for iPSC/2 than iPSC.

³The maximum grid count is increased to more than 250 000 on the new 32-node iPSC/860 system with 16 Mbytes per node

¹ Φ_n and Φ_p indicates the use of Slotboom variable in the continuity equation.

matrix element (A_{ij}) can be considered as an edge between the row node (i) and the column node (j). When two edges contain no common node, the matrix-vector operations they represent are independent and can be grouped together. Due to the above restriction, there can be only one edge that contains a particular node in each group. Thus the minimum number of groups is the maximum number of edges a node has. For a seven-point stencil, this number is six (the diagonal elements are already grouped together in the sparse matrix data structure). The grouping is achieved by a greedy algorithm searching through the edges represented in the sparse matrix pointers. So far, optimal grouping, in the sense that six groups are sufficient to cover all the edges, has always been achieved with our present ordering schemes of the nodes. The average size of the groups, which equals the average vector length, is about half the number of nodes.

When compared with the element grouping, the edge grouping has the advantage of not requiring extra double precision storage and extra floating point operations. Furthermore, it is compatible with the parallel implementation of the matrix solver on the hypercube [14]. The disadvantage is that more indirect addressing is needed which slows down the vector operations. This disadvantage is partially alleviated by re-ordering the matrix elements so that the indirect addressing is not needed to access the matrix. This change has resulted in a 25% increase in CPU performance for Cray YMP, which is also the average improvement seen on Convex C1.

With the matrix-vector product vectorization issue resolved by the above mentioned schemes, all the operations in the conjugate gradient algorithm are now well vectorized. Extra efforts are needed to vectorize the operations involving the IC or ILU preconditioner which account for more than one third of the total operation counts. A well-known scheme is to color order the nodes. Coloring divides the nodes into several groups such that a node will not be in the same group with any nodes that share an edge with it. For the seven-point stencil finite-difference scheme currently used in STRIDE, only two colors are necessary and the ordering scheme is called red-black ordering. The price for the red-black ordering is an increase in iteration number. From our experience, the increase in the iteration numbers is about 30% for symmetric matrices derived from uniform grids and about double for asymmetric matrices. Still the advantages stemming from the ability to fully vectorize the entire matrix solver operation outweighs its penalties. The performance of the algorithms were measured in terms of CPU time per linear iteration. When measured in terms of CPU time per linear iteration per variable, the raw speed advantage of vector over scalar for ICCG and ILUCGS type of iterative matrix solvers is about 4.5 using the Convex C1 and 9.5 using the Cray YMP.⁴ Therefore, even with

⁴The vectorization on iPSC/2 was not pursued because of the design flaw in vector processing unit (VPU). As it stands, a VPU can only access about one eighth of the total memory and a complete vectorization of the iterative solvers would entail constant data swapping. The newly available i860-based systems does not have such a problem.

the worst-case situation, red-black ordering reduces the computation time of ICCG and ILUCGS operations by more than 50% on Convex C1 and more than 80% on Cray YMP.

When implemented on Convex C1 and Cray YMP, the iterative matrix solvers in STRIDE are able to run at 2 MFlops on the C1 and 100 MFlops on the YMP.

IV. ACCELERATION OF TWO-CARRIER GUMMEL STYLE ITERATIONS

Having achieved dramatic improvement in the convergence performance of Gummel style nonlinear scheme at high level injection using a MSP scheme [5], our attention turned to the application of MSP and its extensions to Gummel style iterations in two-carrier simulation. We will call the MSP scheme proposed in [5] MSP-1C, with 1C added for one-carrier.

For completeness, the key formula for MSP-1C is shown in the following:

$$D_{\psi\psi}\delta\bar{\psi} + D_{\psi\phi_n}\delta\Phi_n = -F_{\psi}. \quad (5)$$

The key point from the discussion of MSP-1C [5] is that in the n -type region where the charge neutrality prevails, (5) is quite accurate and its substitution into the linearized continuity equation will retain much of the coupling between Poisson and continuity equations, thereby improving the convergence performance of the Gummel style nonlinear iteration scheme.

Two simple extensions of the MSP-1C scheme, which retain the advantage of low computational cost per iteration, can be used in two-carrier simulations. One is to apply MSP-1C to the "main" carrier equation, such as the electron continuity equation in n-p-n transistor simulations. The other is to use MSP-1C separately on each continuity equation. The advantage of these extensions is low computational cost per iteration. For both cases, the presence of the other carrier is ignored as far as MSP-1C is concerned. Therefore, dramatic improvement in convergence performance is not to be expected. Nevertheless, significant improvement has been observed over the traditional Gummel's scheme with the asymptotic convergence rate for these schemes ranging from four to six of that for the Gummel iteration in the high-level injection regime. However, these increasing convergence rates are still very low with the error typically halving every six to seven iterations.

These unsatisfactory results prompted us to explore new schemes. Our first approach was to use a "true" extension of MSP-1C, the MSP-2C scheme. The key formula for this MSP-2C scheme is shown as follows:

$$D_{\psi\psi}\delta\bar{\psi} + D_{\psi\phi_n}\delta\Phi_n + D_{\psi\phi_p}\delta\Phi_p = -F_{\psi}. \quad (6)$$

Comparing (5) with (6), the terms associated with changes in both carrier variables are included, thereby the name MSP-2C. When (6) is substituted into the linearized continuity equations of both carriers, we obtain a matrix with a dimension of $2N$ by $2N$. This matrix can be expressed in terms of the original matrices as follows:

$$A_{\text{MSP-2C}} = \begin{bmatrix} -A_{\psi_n\psi} D_{\psi\psi}^{-1} & I & 0 \\ -A_{\psi_p\psi} D_{\psi\psi}^{-1} & 0 & I \end{bmatrix} \begin{bmatrix} D_{\psi_n\psi} & D_{\psi\psi} \\ A_{\psi_n\psi} & D_{\psi_n\psi_p} \\ D_{\psi_p\psi} & A_{\psi_p\psi_p} \end{bmatrix}. \quad (7)$$

As the size of $A_{\text{MSP-2C}}$ indicates, the two continuity equations are solved together with the MSP-2C while Poisson's equation is still solved separately. Solution of larger matrices results in an increase in both the data storage and CPU time per iteration. This is the major factor that reduces the maximum simulatable node count from that previously reported to 130 K to about 100 K. CPU time per iteration has been observed to roughly double.

Somewhat to our surprise, the incorporation of MSP-2C does little to improve the convergence of normal transistor simulations beyond what has been achieved by the MSP-1C scheme, although in the latch-up analysis, the convergence behavior of the latched device improves dramatically with the error at least halving for every iteration.

In order to stay within the memory requirement of MSP-2C scheme instead of going full Newton and a double of memory requirement, we next turned an algorithm which solves Poisson equation with the "main" carrier equation such as the electron equation in a n-p-n transistor, the Newton-1C scheme.⁵ An additional motivation for using the Newton-1C scheme was the observation that throughout the simulation of normal bipolar transistor operation, the coupling between Poisson and the "minor" carrier equation remained very weak⁶ even though the device itself had gone into the strong high-level injection regime.

Fig. 1 shows the convergence results for Gummel, MSP-1C and Newton-1C schemes for simulations done on a bipolar transistor. V_{CE} is fixed at 5 V. The number of nodes is about 13 700 and the simulations are executed on eight processors with an estimated parallel efficiency of 72%. As shown in Fig. 1, at the highest injection level, MSP-1C is about three times faster than Gummel, while Newton-1C is still three times faster than MSP-1C, despite the doubling in CPU time per iteration. Although the full Newton scheme is not yet available from STRIDE, Newton-1C is expected to be faster than the full Newton scheme since CPU time per iteration for the full Newton is expected to be twice of that for Newton-1C. For instance, for the test example in the next section, a total of eighteen iterations are needed for convergence, which is CPU equivalent to less than eight full Newton iterations. Given the severity of the test example, it is very unlikely that the full Newton scheme can converge in less than eight iterations.

It should be noted⁷ that the kind of matrix solvers used in a device solver affects the results obtained for using the

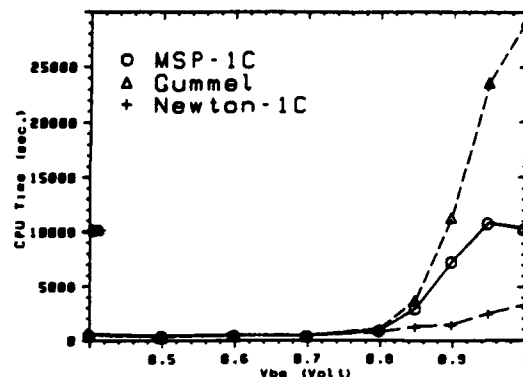


Fig. 1. Two-carrier convergence results of various schemes.

MSP and Newton-1C schemes. When a device solver uses iterative device solvers, forward elimination and backsubstitution as well as matrix-vector multiplication are the most CPU intensive operations. The cost of these operations grows as the *square* of the number of variables per node. In a device solver using a direct matrix solver, however, the cost of matrix factorization usually dominates CPU time and it grows as the *cube* of the number of variables per node. Therefore, the proper use of the MSP and Newton-1C schemes in such a device solver are expected to yield an even more favorable result in comparison with the use of the full Newton scheme.

Given the results of these schemes, one might ponder the reason behind the relative success of MSP in one-carrier simulation and its relative ineffectiveness in two-carrier simulations even though the minor carrier equation is only weakly coupled to the Poisson equation. The key formula of the MSP scheme, (5) and (6), relates the relevant carrier concentration with the value of ψ at the high carrier concentration nodes. Therefore, MSP performs best when the high-level injection only causes the local coupling between Poisson and continuity equation(s), that is, when the value of ψ at a node is the dominant factor in determining the carrier concentration at that node. This is the situation in the inversion layer of MOSFET's where the conduction charge is induced electrostatically. The situation for the two-carrier simulation is very different. High-level injection of a bipolar transistor almost always accompanies the Kirk effect [16], i.e., the base push-out effect. When the Kirk effect occurs, the carrier concentration in the lightly doped collector region is determined not by the local values of ψ , but rather the amount of the collector current that needs to be sustained. This is in turn determined by the injection level at the base-emitter junction. On the other hand, the amount of carrier concentration also significantly impacts the ψ distribution in the lightly doped collector region. Therefore, a nonlocal coupling exists between the Poisson and continuity equations. Since MSP schemes are only capable to take into account the local coupling between Poisson and continuity equation(s), they are relatively ineffective in improving the convergence of Gummel style iterations in normal bipolar simulations. On the other hand, since it takes into account the complete coupling between Poisson and the

⁵The Newton-1C scheme was used in some early works on device simulation, such as an early version MINIMOS and Dr. J. W. Slotboom's initial work on 2-D simulation some 15 a ago.

⁶This can be ascertained by noticing that the error of the other continuity equation is several orders below that of the main continuity equation.

main carrier equation by solving them together, the Newton-1C scheme is able to achieve full-Newton competitive convergence performances, given that the minor carrier equation is virtually independent of the Poisson equation.

The fact that Newton-1C is more expensive than MSP-1C in CPU time per iteration presents an issue of when to switch from Gummel or MSP-1C schemes to Newton-1C in order to optimize the overall solution time. During the implementation of MSP-1C scheme, we found an interesting parameter which has important bearing on this issue. Excluding the recombination terms, the discrete current continuity equation at node i consists of the sum of all the current components flowing into the node. For an electron current component which flows into node i along the edge connecting the nodes i and j ($I_{n,ij}$), there are contributions to diagonal $A_{\Phi_n\Phi_n}(ii)$ ($\partial I_{n,ij}/\partial \Phi_n(i)$) and off-diagonal $A_{\Phi_n\Phi_n}(ij)$ ($\partial I_{n,ij}/\partial \Phi_n(j)$). From the point of view of matrix formulation, the application of MSP-1C scheme means a modification to the original matrix of the current equation (i.e., $A_{\Phi_n\Phi_n}$ is modified by $-A_{\Phi_n\psi} \cdot DD_{\psi\psi}^{-1} D_{\psi\Phi_n}$ for electron equation [5]). For $I_{n,ij}$, the modification to $A_{\Phi_n\Phi_n}(ii)$ is $-(\partial I_{n,ij}/\partial \psi(i)) D_{\psi\psi}^{-1}(i) D_{\psi\Phi_n}(i)$. A detailed analysis reveals that, if $\Phi_n(i) < \Phi_n(j)$, then the modification strengthens $A_{\Phi_n\Phi_n}(ii)$, i.e., it has the same sign with $A_{\Phi_n\Phi_n}(ii)$; otherwise, $A_{\Phi_n\Phi_n}(ii)$ is weakened by the modification. This diagonal weakening merits special attention, since, according to our experience, all the diagonals in a matrix must have the same sign for the iterative matrix solvers to converge. It turns out that the ratio ($\beta_{n,ij}$) between the modification $(-(\partial I_{n,ij}/\partial \psi(i)) D_{\psi\psi}^{-1}(i) D_{\psi\Phi_n}(i))$ and the original term $(\partial I_{n,ij}/\partial \Phi_n(i))$ is never less than minus one. This ensures the diagonals of $A_{\Phi_n\Phi_n}$ will not become negative as the result of MSP modifications. Furthermore, the most negative of all the $\beta_{n,ij}$ can be used as a parameter, indicating the degree to which the original $A_{\Phi_n\Phi_n}$ has been modified.⁷ Our experience indicates that when the magnitude of this parameter is small, the Gummel iteration is just as effective as any other nonlinear scheme. When the magnitude approaches to one, however, the Gummel iteration becomes very slow and a more elaborate scheme has to be employed to accelerate the convergence. Therefore, this parameter is also a very good indication of the strength of the coupling between Poisson's equation and the continuity equation. A scheme is thereby implemented in STRIDE to use Newton-1C, if the user desires to do so, when the magnitude of this matrix parameter is large enough. Currently, the threshold value is 0.25. The low bias results of Newton-1C curve in Fig. 1 were actually obtained with Gummel or MSP-1C schemes. By the same token, this switch scheme can also be extended to use the full Newton scheme when it is truly necessary. In short, the introduction of this matrix parameter makes it feasible for the program to automatically choose the best nonlinear scheme at its disposal. Although originated in the context

of the finite difference approach, this parameter can also be calculated in device solvers using the finite element approach with only minimum overhead.

V. CONVERGENCE CRITERION AND DAMPING SCHEMES

Traditionally, the convergence criterion for the carrier variables have been that the maximum relative change of all the variables is below a certain value.⁸ In contrast to this relative criterion, there is also the absolute criterion which measures the convergence of residual of the difference equations. Although the absolute criterion is useful in a global damping scheme [17], its application as a convergence criterion is less useful. For example, the residual of the current continuity equation which can be tolerated depends heavily on the amount of current flowing through the simulating device, which can differ by many orders of magnitude. It is, therefore, not always possible to determine *a priori* what is the appropriate tolerance level for the residual. Another important criteria that has not been widely used is the convergence of the terminal currents, which also entails the conservation of all the terminal currents. When we monitor the convergence of both the currents and the variables, we found that in the low current regime, the variables may converge before the currents do. Instances were observed in which the currents do not converge until the maximum relative change in carrier variable fell below 10^{-9} or even lower. By combining these two convergence criteria, we are able to reduce the lower limit above which the currents can be calculated self-consistently. On the other hand, at high current levels, variables may lag far behind the currents in convergence. In this regard, we observed instances in which the convergence of current was achieved before the maximum relative change in the carrier variable reached below 10^{-2} . We feel that it is of not much use to calculate the variables to five or six digits of accuracy when the currents have converged. Based on the previous observations, we have chosen a combination of a relatively loose variable convergence criterion about 10^{-2} with a current convergence criterion of about 10^{-4} which has worked quite well for us so far regardless of current levels.

A more serious issue is that for a device simulator using iterative matrix solvers, the absolute convergence of carrier variables becomes much more difficult since matrices are usually solved to a less accurate extent than direct solvers are. In fact, we found that when the traditional Slotboom variables are used, the convergence of these variables become almost impossible in typical two-carrier simulations. When the scaled Slotboom variables are used, however, more often than not, this apparent non-convergence of the concentration variables does not impede the convergence of current which is also accompanied by the convergence in the errors in the continuity equations. Therefore, we have encountered situations in

⁷ Although an electron equation is used as the example, a similar parameter can also be calculated for the hole equation.

⁸ The convergence criteria for ψ usually measures its maximum absolute change since it is well scaled ($\delta\psi \ll 1$ means the solution is very close.)

which the solutions have converged for all practical purposes, but were not recognized by the traditional convergence criterion for the concentration variables. Therefore, the question is whether we can find a better criterion which can tell when a nonconvergence of carrier variables is for real. The clue again lies in (5) and (6). In the first-order approximation, the terms like $-D_{\psi\psi}^{-1} D_{\psi n}$ and $-D_{\psi\psi}^{-1} \cdot D_{\psi p}$ represent the changes in ψ necessary to restore the charge balance required by Poisson's equation. If these changes are very small after the solution of carrier variables, then there are virtually no changes necessary in ψ and we have a converged solution. A detailed look at these terms reveals that they are the relative changes in carrier variables multiplied by a weighting vector whose values are no larger than unity. Therefore, the convergence criterion for the carrier variables can be these weighted relative changes. The weighting coefficients are very small for the minority carrier, and approach one for the majority carrier in the charge neutral region. In essence, this criterion discounts the changes in minority carrier concentration as long as these changes do not significantly perturb Poisson's equation nor impede the convergence of terminal currents. There is a very good correlation between the new measures and the maximum error in the continuity equations and we are able to distinguish the practically converged solutions from the apparently non-converging solutions according to the old measures. The data storage and computational cost of the new measures are minimum since only vector (not matrix) operations are involved.

One of the challenging issues of the nonlinear iteration schemes is how to choose a robust damping scheme to ensure global convergence. The schemes used in STRIDE for Gummel and MSP schemes have been discussed in [5]. For the MSP-2C scheme, both $D_{\psi\psi}^{-1} D_{\psi n}$ and $D_{\psi\psi}^{-1} D_{\psi p}$ are considered and different limiting values are used. The problem for the Newton-1C scheme is more difficult since we have two variables with vastly different ranges and a change in ψ impacts the carrier concentration exponentially. Our first attempt was to try various residual limiting schemes such as suggested in [17]. Applications based on their norm reduction principles have proven to be quite successful in nonlinear iteration of Poisson's equation [5]. The results have not been consistent, however. The key difficulty here is how to weigh the residuals from the Poisson and continuity equations.

Our recent attempts are based on a different principle. Since ψ impacts carrier concentration exponentially, the matrix will change dramatically for a large change in ψ . Therefore, the changes in ψ should be restricted so as not to cause large changes in carrier concentration which would greatly upset the charge balance in the previous solution. Similarly, the changes in the carrier variable should be restricted to require only a modest change in ψ to restore the charge balance. In essence, our scheme is a trusted region approach, widely used in the nonlinear optimization community, with the trusted region determined based on the specific knowledge of semiconductor equa-

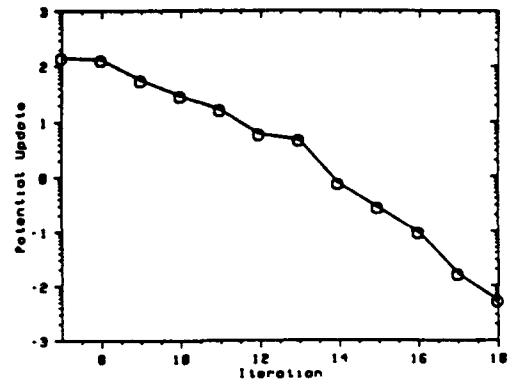


Fig. 2. Global damping scheme: evolution of potential update.

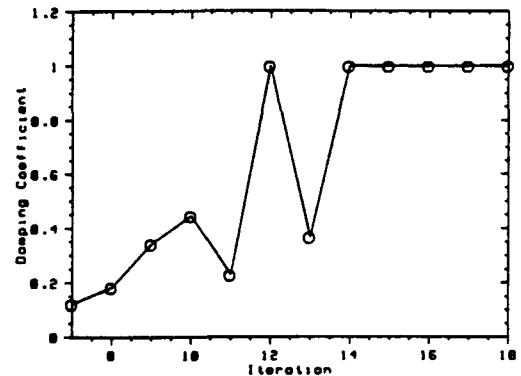


Fig. 3. Global damping scheme: evolution of alpha.

tions. In the actual implementation, a damping coefficient α is first calculated based upon the above principles, and the errors in Poisson and continuity equation are re-evaluated after the damped variable update to monitor the change in these errors. The use of a damped update on the first try prevents the problem of machine overflow which may occur with a full update when $\delta\psi$ become too large. It is rare that a such calculated α still produces an update which causes a dramatic increase in these errors. When this occurs, α is further reduced to brighten down these increases. Our experience shows that it is essential not to insist on the reduction of residuals as long as their increase is moderate. Otherwise, excessive damping may occur which slows the convergence process to a crawl. Fig. 2 shows the evolution of the magnitude of ψ update generated by the Newton-1C scheme during a n-p-n bipolar transistor simulation for an initial bias of $V_{CE} = 5$ V and $V_{BE} = 1$ V. This is a very severe test example for the global damping scheme in that a bipolar transistor is biased into heavy high-level injection regime in one step. Fig. 3 shows the values of the damping coefficient α used, while Fig. 4 shows the errors in the Poisson and continuity equations. Both errors are normalized by their respective starting errors at iteration 7. The number of iteration starts at 7 since Newton-1C was first engaged at this iteration after solution was settled down somewhat. Although the initial (normalized) ψ update is more than 100, which translated to about 3 V, it goes down to less than one (about 26 mV) after just seven iterations. After-

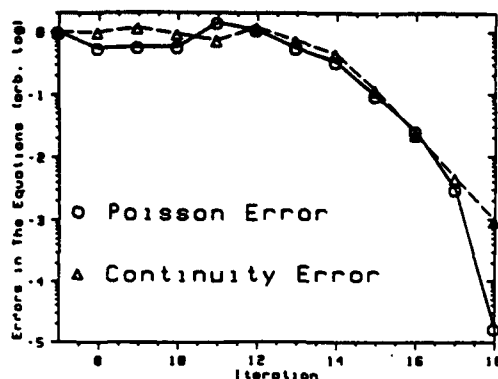


Fig. 4. Global damping scheme: evolution of the residuals.

wards, the solution converges superlinearly as can be seen from the evolution of Poisson errors.

VI. USER INTERFACES AND TOOL INTEGRATION

To provide easier use of STRIDE, we have created a utility program that encapsulates the details of the program and its algorithms from the user. The program, based on the PISCES-IIIB utility, BIPMESH, provides automated input deck generation, which includes mesh, model, and bias information. The program asks the user questions about structure geometry, doping profiles, and bias conditions. From this input, mesh is automatically generated using heuristics based on our experience with PISCES-IIIB simulation. Doping profiles may be either analytic functions, SUPREM-IV profiles, or SUPREM-III profiles. The extension in the other dimension(s) for SUPREM-IV(III) profiles is(are) based on an error function approximations in the other dimension(s).

The use of automated input deck generation facilitates integration of STRIDE into an integrated simulation system for TCAD. STRIDE will be included as part of a suite of device simulation tools in an integrated simulation environment based upon SIMPL-IPX [18].

To help the user interpret the output of STRIDE, a link to Visualization tools such as NCSA XImage [19] and Spyglass Dicer are provided. By mapping solution values to color, one is able to see spatial variations of the solution variable much more rapidly. In addition one can animate sequences of frames, useful for simulating transient effects and for looking at dc sweeps. The translators from STRIDE data format to the format of these Visualization tools operate as stand-alone utilities.

VII. 3-D LATCH-UP ANALYSIS WITH STRIDE

To fully appreciate the benefits of 3-D device simulation, the relation of layout to latch-up conditions were investigated. Many studies in 2 D, such as [20] and [21], have shown that latch-up trigger current is lower than the expected value based on circuit models [22] due to debiasing under the P+ contact as shown in Fig. 5. The current flow in the n-well under the P+ induces a voltage drop. This voltage drop in turn forward biases the P+/n-well junction, resulting in injected current into the

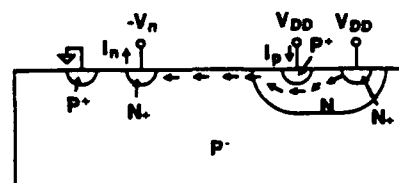


Fig. 5. Structure for 2-D latch-up analysis.

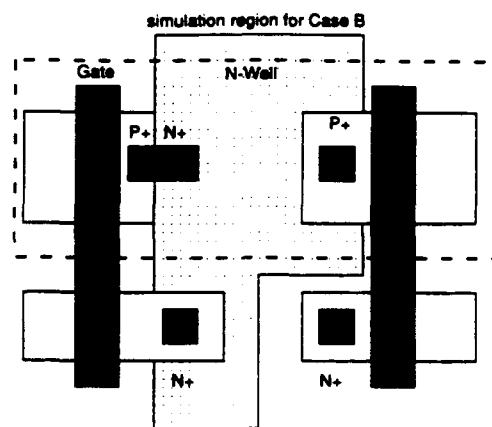


Fig. 6. CMOS layout.

substrate. The injected current into the substrate debiases the injector contact, resulting in sustained latch-up. In terms of the circuit model, lateral injection causes the vertical p-n-p to conduct, which in turn increases the biases on the lateral n-p-n.

The trigger current calculated from the 2-D structure can be judged as very conservative when compared to the case seen in an actual layout of an inverter as shown in Fig. 6, where all the injected current does not flow under the P+. Furthermore, the N+ S/D can face either a N-well contact (case B) as shown or a P+ S/D (Case A). For most standard cell layout, case B is more typical.

The simulation structure for 3 D contains about 10 000 nodes with dimensions of 10 μm in the x-direction, 10 μm in the y-direction, and 7 μm in the z-direction. A buried layer is used on the bottom of the structure. Four contacts are used—three on the surface, with two different injector contact positions as shown in Figs. 7 and 8, and one on the bottom acting as a substrate contact. Doping profiles are analytic functions for simplicity. The n-well and P+ contact inside the n-well are held at 2-V potential, while the substrate contact is held at 0 V. The voltage at the N+ injector is negatively biased until the current at the node dramatically increases and the carriers flood the device—indicating a latch-up condition. Using MSP-2C, the average time to simulate the trigger current is about 6 h on the 16 node hypercube, corresponding to the use of four bias points. The output data from STRIDE are post-processed into a format which is readable by the visualization tool Spyglass Dicer.

The positioning of the injector contact in the substrate is very important in determining the value of trigger current of this structure. Based on the debiasing mechanism

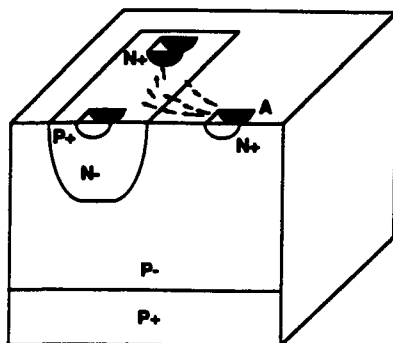


Fig. 7. 3-D latchup analysis structure: case A.

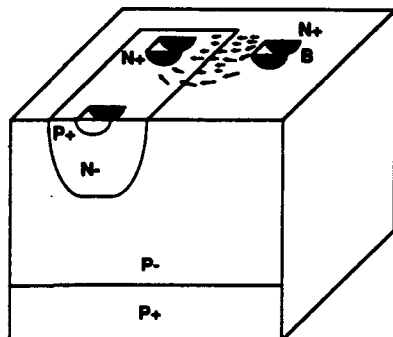


Fig. 8. 3-D latchup analysis structure: case B.

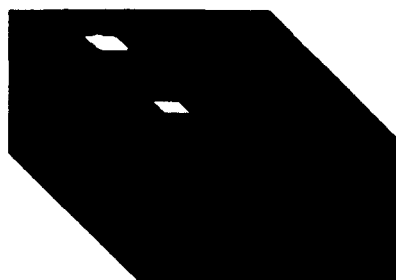


Fig. 9. 3-D plot of voltage drop at latchup onset: case A.

for the 2 D discussed above, it would seem plausible that the trigger current from case A would be lower than that from case B since more of the injected current would tend to flow under the P+ contact for case A. STRIDE simulations do indeed confirm this, as seen from Figs. 9 and 10. These 3-D pictures, taken from Spyglass Dicer, show a voltage drop (ψ minus its equilibrium value) as a function of the position prior to latch up (injected currents (Table I) are different). Differences in shading corresponds to gradients in voltage drop.

The main difference between the gray scale figures can be seen at the N+ well contact (top left corner of each of the images). It is not possible to examine the voltage drops at the P+ contact in the N-well since at the onset of latch-up the amount of debiasing is similar due to the exponential voltage dependence of current. The point to remember is that the device will latch up at a particular current flow in the well. For case A, the region around the

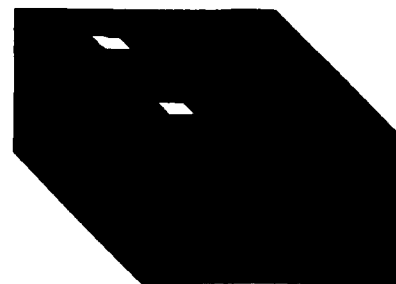


Fig. 10. 3-D plot of voltage drop at latchup onset: case B.

TABLE I
LATCH-UP TRIGGER CURRENT

Trigger Current (mA)	
Case A	Case B
0.244	0.482

N+ contact in the direction away from the injector is dark, corresponding to an asymmetric current flow into the contact. This current is mostly traveling through the well in the direction of the P+ contact. In contrast, case B shows uniform shading around the N+ contact, showing that the injected current is being effectively collected by the both sides of the N+ contact. Thus case B should be collecting more total injected current at the onset of latchup. This finding is confirmed in Table I.

VIII. CONCLUSIONS

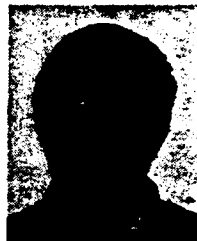
In this paper, we have reported progress in 3-D device simulation, focusing on computational aspects. By exclusively using iterative matrix solvers and insisting on low memory usage nonlinear iteration schemes, approximately 100 000 nodes can be solved on a user memory of about 100 M bytes. ICCG and ILUCGS types of iterative solvers are efficiently vectorized by using both the independent edge grouping scheme and the red-black ordering. Various MSP schemes are explored to improve the convergence of two carrier simulation using Gummel style nonlinear schemes. They not only offer significant improvement by themselves, but also provide the insight leading to the automatic selection of nonlinear schemes which offers the best CPU performance. Issues of the convergence criterion and global convergence scheme have also been successfully addressed with the insight provided by MSP schemes. A better user interface has been developed to facilitate program usage and tool integration. Aided with graphics capabilities made available through various graphics tool links, layout dependence of latch-up trigger current has been successfully analyzed.

ACKNOWLEDGMENT

The authors wish to thank Dr. Arthur Raefsky for discussions on vectorization schemes and visualization issues.

REFERENCES

- [1] T. Toyable, H. Masuda, Y. Aoki, H. Shukuri, and T. Hagiwara, "Three-dimensional device simulator CADDETH with highly convergent matrix solution algorithms," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 2038-2043, Oct. 1985.
- [2] M. Thurner and S. Selberherr, "The extension of MINIMOS to a 3D simulation program," in *Proc. NASCODE V*, Dublin, 1987, pp. 327-332.
- [3] J. Burgler, P. Conti, G. Heiser, S. Paschedag, and W. Fichtner, "Three-dimensional simulation of complex semiconductor structures," in *Proc. Int. Symp. on VLSI Processes, Device and Systems*, Taiwan, 1989.
- [4] J.-H. Chern, J. T. Maeda, L. A. Arledge, Jr., and P. Yang, "SIERRA: A 3-D device simulator for reliability modeling," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 516-527, May 1989.
- [5] K.-C. Wu, R. F. Lucas, Z.-Y. Wang, and R. W. Dutton, "New approaches in a 3-D one-carrier device solver," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 528-537, May 1989.
- [6] J. W. Slotboom, "The p-n product in silicon," *Solid-State Electron.*, vol. 20, pp. 279-283, 1977.
- [7] D. M. Caughey and R. E. Thomas, "Carrier mobilities in silicon empirically related to doping and field," *Proc. IEEE*, vol. 55, pp. 2192-2193, 1967.
- [8] R. F. Lucas, private communication, Stanford Univ., Stanford, CA, 1988.
- [9] P. Conti, N. Hitchfeld, and W. Fichtner, "OMEGA—An octree-based mixed element grid allocator for adaptive 3D device simulation," in *Tech. Dig.—Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits: NUPAD III*, Honolulu, Hawaii, June 1990.
- [10] D. Scharfetter and H. K. Gummel, "Large-signal analysis of a silicon Read diode oscillator," *IEEE Trans. Electron Devices*, vol. ED-16, pp. 64-77, 1969.
- [11] J. A. Meijerink and H. A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix," *Math. Computation*, vol. 31, no. 137, pp. 148-162, Jan. 1977.
- [12] C. den Heijer, "Preconditioned iterative methods for nonsymmetric linear systems," in *Proc. Int. Conf. on Simulation of Semiconductor Devices and Processes*, June 1984.
- [13] R. F. Lucas, K. Wu, and R. W. Dutton, "A parallel 3-D Poisson solver on a hypercube multiprocessor," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 1987, pp. 442-445.
- [14] R. F. Lucas, "Solving planar systems of equations on distributed-memory multiprocessors," Ph.D. dissertation, Stanford Univ., Dec. 1987.
- [15] F. Shakib, "Finite element analysis of the compressible Euler and Navier-Stokes equations," Ph.D. dissertation, Stanford Univ., Nov. 1988.
- [16] C. T. Kirk, Jr., "A theory of transistor cutoff frequency falloff at high current densities," *IRE Trans. Electron Devices*, pp. 164-174, Mar. 1962.
- [17] R. E. Bank and D. J. Rose, "Global approximate Newton methods," *Numer. Math.*, vol. 37, pp. 279-295, 1981.
- [18] W. Scheckler, A. S. Wong, R. H. Wang, G. Chin, J. R. Camagna, K. H. Toh, K. H. Tadros, R. A. Ferguson, A. R. Neureuther, and R. W. Dutton, "A utility-based integrated process simulation system," in *1990 Symp. on VLSI Technology—Dig. Tech. Papers*, June 1990, pp. 97-98.
- [19] "NCSA XImage for the X window system, version 1.0," National Center for Supercomputing Applications, Nov. 1989.
- [20] R. Menozzi, L. Selmi, E. Sangiorgi, G. Crisenza, T. Canioni, and B. Ricco, "Layout dependence of CMOS latchup," *IEEE Trans. Electron Devices*, vol. 35, pp. 1892-1901, Nov. 1988.
- [21] M. R. Pinto and R. W. Dutton, "Accurate trigger condition analysis for CMOS latchup," *IEEE Trans. Electron Device Letters*, vol. EDL-6, pp. 100-102, Feb. 1985.
- [22] D. B. Estreich, "The physics and modeling of latch-up and CMOS integrated circuits," Tech. Rep., G201-9, Stanford Electron. Labs, Stanford, CA, Nov. 1980.



Ke-Chih Wu (M'87) graduated from Fudan University, Shanghai, China in 1977, and received the M.S. and Ph.D. degrees in electrical engineering from Stanford University, CA, in 1981 and 1987, respectively.

From 1977 to 1978, he worked as a research assistant in the area of CCD technology with Technology and Physics Research Institute in Shanghai, China. Currently, he is a Research Associate in the Department of Electrical Engineering, Stanford University. His main research inter-

ests are in the area of multidimensional numerical analysis of semiconductor devices on parallel computers.



Goodwin R. Chin (S'84-M'85-S'86) received the B.S. degree in electrical engineering and computer science and materials science and engineering from the University of California at Berkeley in 1983, and the M.S. degree in materials science from Stanford University in 1984. He is currently working towards the Ph.D. degree in electrical engineering at Stanford University.

From 1984 to 1985 he worked at General Electric Corporate Research and Development, Schenectady, NY, in the areas of circuit design and advanced technology development. From 1985 to 1987 he worked at GE Intersil in the areas of analog and digital circuit design and advanced process development. From 1988 to 1989 he worked as a private consultant in the area of circuit design.

Mr. Chin is a member of Eta Kappa Nu and Tau Beta Pi.



Robert W. Dutton (S'67-M'70-SM'80-F'84) received the B.S., M.S. and Ph.D. degrees from the University of California, Berkeley, in 1966, 1967, and 1970, respectively.

He has held summer staff positions at Fairchild, Bell Telephone Laboratories, Hewlett-Packard and IBM Research in 1967, 1973, 1975, and 1977, respectively. He is currently a Professor of Electrical Engineering and Director of Research in the Center for Integrated Systems, Stanford University. He has published more than 200 articles in technical journals. His research interests include integrated circuit process, device, and circuit technologies, especially the use of computer-aided design and parallel computational methods.

Dr. Dutton received the 1987 IEEE J. J. Ebers Award and the 1988 Geiggenheim Fellowship. From 1984 to 1986, he was the Editor of IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN.

A Methodology for Parallelizing PDE Solvers: Application to PISCES^[15]

Bruce P. Herndon

Integrated Circuits Laboratory, Stanford University
231A Applied Electronics Lab
Stanford University, Stanford, CA 94305-4055
Tel: (415) 723-1482, Fax: (415) 725-7298
e-mail: herndon@gloworm.stanford.edu

Arthur Raefsky

Centric Engineering Systems
3801 E. Bayshore Rd.
Palo Alto, CA 94303
Tel: (415) 960-3600, Fax: (415) 940-1252
e-mail: raefsky@centric.com

Ronald J.G. Goossens

National Semiconductor Corporation
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-2420, Fax: (408) 721-7266
ronald@ammon.nsc.com

Robert W. Dutton

Integrated Circuits Laboratory, Stanford University
203 Applied Electronics Lab
Stanford University, Stanford, CA 94305-4055
Tel: (415) 723-4138, Fax: (415) 725-7298
e-mail: dutton@gloworm.stanford.edu

Abstract

This paper presents a methodology for adapting PDE solvers for parallel execution based upon the single-program-multiple-data programming paradigm. Our approach minimizes changes to existing code and data structures, thereby preserving the value captured within "dusty-deck" programs. The resulting parallel application is easily portable to most message-passing distributed-memory architectures. To demonstrate the viability of our methodology, the commercially-available, semiconductor modelling program, PISCES, has been adapted for parallel execution. Our experience shows that all non-local references can be resolved through careful choreography without extensive modifications to the original code. The parallel simulator currently runs on the Intel iPSC/860 and the Thinking Machines CM-5. Simulating realistic complex device structures, we have achieved remarkable performance gains over high-performance serial workstations. We also demonstrate the ability, due to the scalability of the parallel simulator, to simulate structures too large for our existing serial computers. This simulation capability could provide immeasurable benefits in the competitive semiconductor industry.

1. Introduction

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.

In recent years, distributed-memory parallel computers built with high-performance microprocessors and supporting the message passing paradigm have become the standard commercial parallel architecture. Most commercial vendors (e.g., IBM, Cray, Intel, and Thinking Machines) produce machines of this type. Likewise, the single-program-multiple-data (SPMD) parallel programming model [1] has become a popular paradigm for developing parallel applications for this class of machines. Emergence of a single hardware abstraction and its requisite programming model should portend the migration of parallel machines from research laboratories to industrial production environments. Unfortunately, one key piece of the puzzle is missing: *industrial applications*. Although many researchers have developed codes for parallel architectures, commercial software development has been slow because the programming model required to take advantage of these architectures is a radical departure from traditional paradigms. Additionally, most commercial users are unwilling to discard the knowledge and expertise captured by their existing "dusty-deck" programs in exchange for a faster yet unproved and unfamiliar parallel code. One way to bootstrap parallel application development is to port existing, well accepted industrial applications to parallel machines while retaining the original code and data structures without major modifications. A method of parallelization that preserves knowledge inherent in "dusty-decks" pleases users by maintaining a familiar environment while providing an improved solution capability. This approach also aids parallel software and hardware designers by rapidly exposing issues involved with running full-scale industrial applications on parallel machines. In this paper we present a methodology for creating parallel grid-based partial differential equation (PDE) solvers from serial ones with minimal changes to the existing code and data structures. Our

method is based upon the SPMD programming model and is easily portable to most message-passing distributed-memory machines. To verify our method and to illustrate the potential for providing vastly improved "dusty-deck" performance, we have adapted a commercially-supported and industry-standard device simulator, PISCES [2,3], for parallel execution. The code currently runs on two distributed-memory architectures: the Intel iPSC/860 and the Thinking Machines CM-5.

An outline of this paper follows: After a brief description of the serial semiconductor device simulator, we present in Section 3 our methodology for adapting PDE solvers for parallel execution accompanied by a discussion of our experiences adapting PISCES for parallel execution. In Section 4, simulation results from a series of increasingly complex grids defining a large, multi-device structure illustrate the utility of the parallel application. Finally, conclusions are drawn in Section 5.

2. Overview of PISCES

PISCES is a large and complex application which solves problems strategic to the semiconductor industry. The code encompasses many areas of research in both physics and computational mathematics. It is a two-dimensional, two-carrier semiconductor device modeling program developed primarily at Stanford University over the past fifteen years and is well known throughout the semiconductor industry. The program is available from Stanford or from one of several Technology CAD vendors who support the code commercially. To date, there are more than one thousand industrial and academic users. PISCES solves Poisson's equation and the semiconductor continuity equations below:

$$\nabla (\epsilon \nabla \Psi) = -q(p - n + N_D^+ - N_A^-) - \rho_F \quad \text{Poisson's equation}$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot J_n - U_n \quad \text{Electron Continuity}$$

$$\frac{\partial p}{\partial t} = \frac{1}{q} \nabla \cdot J_p - U_p \quad \text{Hole Continuity}$$

Ψ is the electric potential, and n and p are the electron and hole concentrations. N_D^+ and N_A^- are the ionized impurity concentrations and ρ_F is the fixed-charge density. J_n and J_p are the electron and hole current densities and U_n and U_p are the electron and hole recombination rates.

These coupled nonlinear PDEs are discretized using a finite volume formulation on a non-

uniform triangular grid. The resulting algebraic equations are solved using a nonlinear iteration method. The coarse-grain structure of the simulator is shown in Figure 1. This program structure is typical of many grid-based nonlinear PDE solvers. The user interface (UI) parses input files, performs file I/O functions, and displays simulation results via plotting utilities. The nonlinear solver supports a wide range of solution techniques and contains both a fully-coupled Newton [4] solver and staggered Gummel [5] scheme. PISCES offers a large and diverse set of physical models. Improvements and additions to the physical models are the most active areas in the continuing development of the simulator. During the nonlinear solution process, a sparse linear system is assembled in a triangle-by-triangle fashion using a subset of the available physical models. It has been shown in [6] that the matrices arising in this type of device simulation are extremely ill-conditioned and not easily solved using iterative techniques. Therefore, the sparse system of linear equations is solved using an optimized sparse direct solver as described in [7].

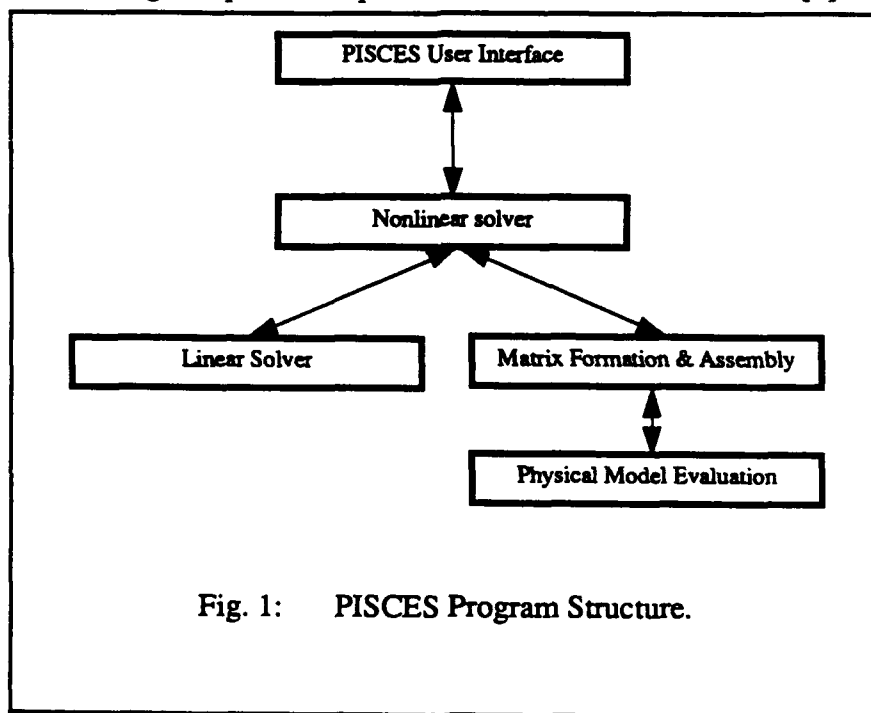


Fig. 1: PISCES Program Structure.

An example PISCES simulation grid is shown in Figure 2. The structure represents the 2D cross-section of a CMOS inverter. This coarse grid contains roughly 1,600 nodes and 3,000 triangles. The structure was created using standard one micron layout rules and was derived from the model of an SRAM cell produced by our lab in conjunction with an industrial partner. We ran a very simple simulation to compute the DC terminal characteristics using a finer mesh than shown (roughly 8,000 nodes and 15,000 triangles) for improved accuracy. After reading the grid and set-

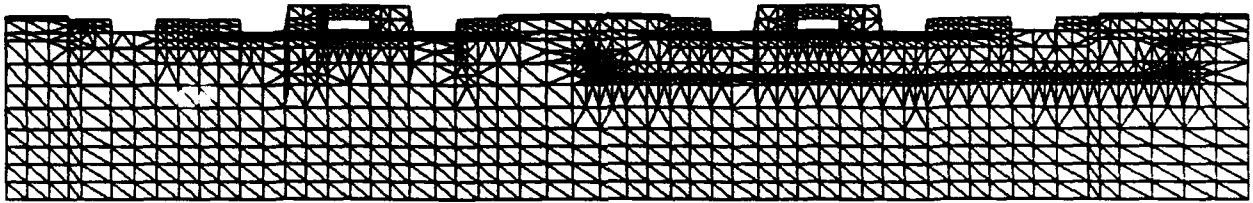
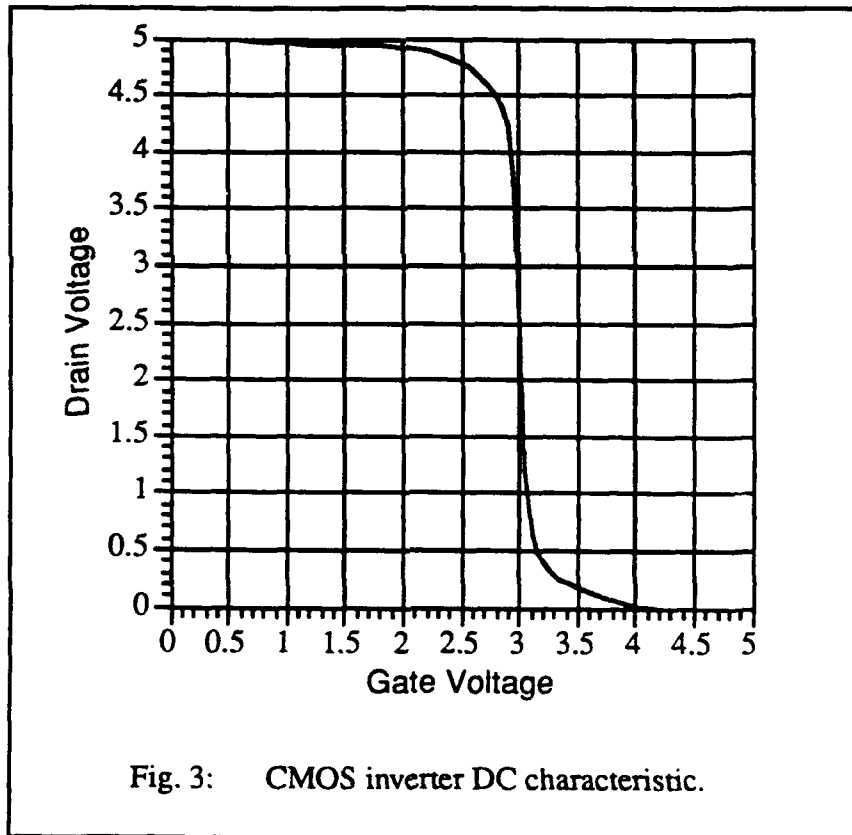


Fig. 2: Grid structure of a CMOS inverter.

ting the proper parameters, the simulation brings the power supply to 5V and then ramps the input gates from 0V to 5V. The output is shown in Figure 3. As expected in an inverter circuit, the output experiences a rapid transition from 5V to 0V as the input voltage increases.



This simple simulation required more than two hours on an IBM RS/6000 Model 560F, the fastest serial computer available in our lab. Using a slightly finer mesh containing nearly 11,000 nodes and 21,000 triangles, the simulation required more than four hours. Simulations to perform more detailed analyses of this structure can easily take several days to complete. If we

were to perform multiple simulations as part of a manufacturability study, several weeks would be needed. A significant boost in the performance of PISCES will be needed if it is to keep pace with current and future demands.

3. A Recipe for Parallelization of PDE Solvers

Before blindly initiating the development of any parallel application, one should estimate the expected benefits and determine if the expected effort will be rewarded. We know that parallel grid-based PDE solvers have had measurable success [8-11]. However, the current version of PISCES consists of approximately 40,000 lines of FORTRAN-77. During its fifteen-year life-span several generations of researchers have modified the code. Due to this continual development, the code has become unwieldy.

Despite its inelegant program structure, great care has been taken to validate the code as well as to improve and calibrate the physical models. In fact, industrial users have invested considerable effort calibrating PISCES with experimental data to further improve accuracy. Moreover, its long lifetime has allowed a large, experienced, and satisfied user base to be established. These users understand the subtleties of the simulator and have developed sophisticated solution strategies unique to PISCES. Finally, both the long lifetime and large user base mean that many bugs and inconsistencies in the code have been exposed and either corrected or worked around. Developing a new application would force users and developers to repeat an enormous amount of non-trivial and time-consuming work beyond initial code development. These facts make strong arguments for parallelizing the current code provided that the effort is not too great and the parallel code maintains consistency with the original serial code. This requires that changes to the existing code be minimized and localized. We can not fundamentally alter either the data structures or the algorithms if the value of the code is to be preserved.

To insure that the parallelization can be done quickly, smoothly, and without major changes to the program, it is crucial to select an amenable parallel programming model. The two dominant parallel programming paradigms are data-parallel [12] and SPMD. Data-parallel compilers require the parallelism to be made explicit in the data structures. Modifying the data structures in PISCES to be compliant with this model would require significant recoding, precluding a data-parallel implementation if our original development goals are to be met. On the other hand, the SPMD model appears to provide a flexible and intuitive framework for parallelization of grid-based PDE solvers without forcing significant changes to either the data structures or the code. Using well-known grid decomposition techniques [13-17], the simulation grid can be broken into

disjoint pieces. Each processor of the parallel machine can run a copy of the device simulator to solve one section of the partitioned grid. Data dependencies will exist along the shared boundaries between partitions. If these data dependencies can be satisfied through inter-processor communication, a global solution identical to the serial solution can be computed. This approach allows the bulk of the code and data structures to remain unchanged. Using this technique, we have parallelized PISCES for distributed-memory computers. The Intel iPSC/860 hypercube in our lab was used for the initial development. Once the parallel application was completed, we moved the code to the Thinking Machines CM-5 to verify portability.

All of the work described below required roughly eight months to complete. We feel that is a reasonable amount of time to preserve a code with a fifteen-year history while giving it a vastly improved solution capability. A step-by-step review of our scheme for porting grid-based PDE codes using the SPMD model and our experiences adapting PISCES using the model is given in the following sections.

3.1 Deciding what to parallelize.

It was observed by Lucas[18] that for small simulation grids (less than 2000 grid points) PISCES normally spends more than 95% of its runtime solving the coupled nonlinear device equations. The remaining fraction of time is spent in the UI. Within the nonlinear solver roughly half the time is spent computing linear solutions to obtain updates to the nonlinear system. The other half of the nonlinear solution time is spent creating initial guesses, forming and assembling the Jacobian matrices, evaluating physical models, and applying nonlinear updates. More recent analysis shows nonlinear solution times grow to be more than 99% of the total runtime for larger grid sizes. As grid sizes increase, the linear solver scales more poorly than other parts of the code; nevertheless, for structures containing tens of thousands of grid points, 15-20% of the runtime remains outside of the linear solver. Amdahl's Law [19] tells us that merely adding a parallel solver to PISCES would yield at best a 5x speedup for many realistic grid sizes.

Restructuring the nonlinear solver and all of its requisite routines to run in parallel is vital to extending the utility of the simulator. However, the UI routines take negligible time and are inherently serial. In order to accommodate this dichotomy, we propose splitting the application into a front-end program to handle the serial UI tasks and a separate parallel program containing the actual device solver.

3.2 Separating the serial code from the parallel code.

Before embarking upon the parallelization of the device solver, we propose to bifurcate PISCES into two serial programs: the serial UI program and a serial solver application (SA). This initial program decomposition resolves the data dependencies between the UI and serial SA, sets up a communication mechanism between the two, and creates a coarse-grain choreographer to coordinate their interaction. The resulting global structure of the simulator is shown in Figure 4. Since managing the hypercube already taxes the control processor of the iPSC/860, we opted to run the UI program on a remote SUN workstation. The serial SA was put on one node of the hypercube.

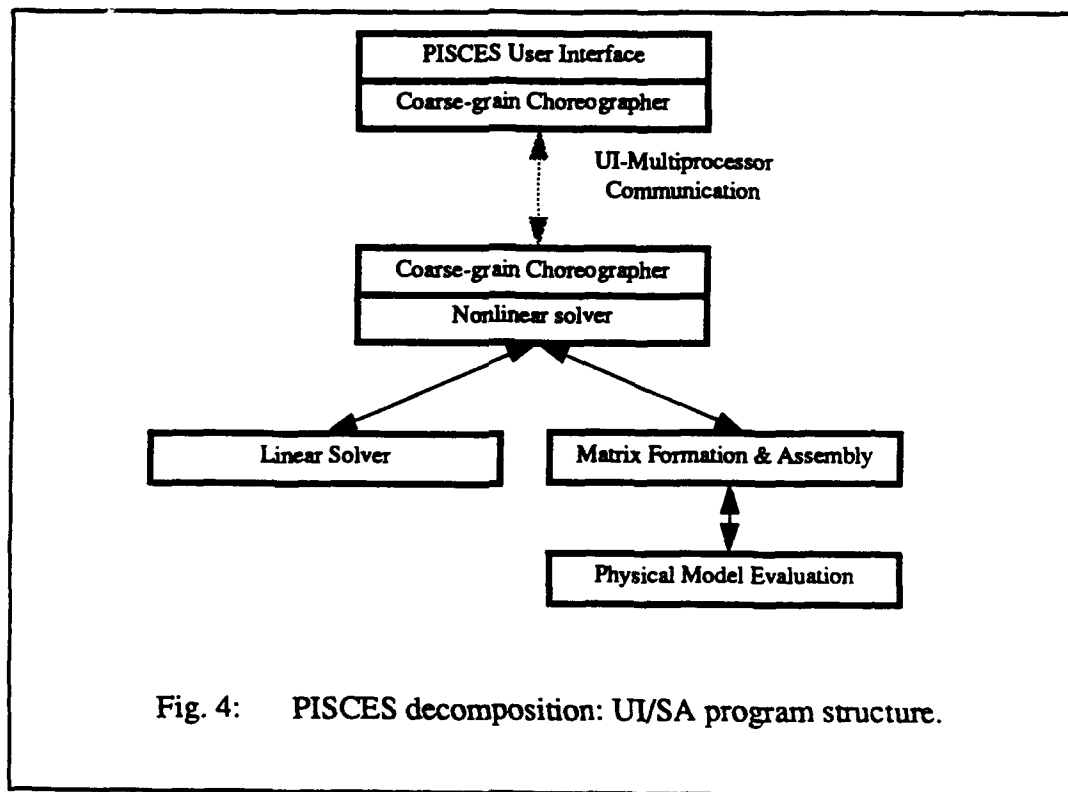


Fig. 4: PISCES decomposition: UI/SA program structure.

After splitting the code between the two programs, we identified all of the data structures required by the serial SA. Like most large FORTRAN-77 programs, PISCES data structures are defined and passed via COMMON blocks. It was a relatively simple task to identify all of the COMMON blocks needed by the serial SA. Preceding computation, the UI performs a wholesale transfer of necessary COMMON blocks to the serial SA using the communication routines described below.

To facilitate the data transfer, we created a communicational package using TCP/IP [20] to

connect the two programs. TCP/IP provides a very slow connection. (The data transfer rate is roughly 65kB/s in our network environment.) However, the communication package can easily be replaced when faster communication resources are available. The main complication involved accommodating the differing machine-word byte orderings of the Intel hypercube and the SUN workstation. In our communication routines, the receiver is responsible for byte swapping data upon receipt. To simplify this procedure, we separated the data into the five types used by PISCES: integer, real, double precision, character, and logical. Similarly typed data are grouped into large blocks for transmission between the UI and SA. This makes the swapping operation simpler and more efficient. Once properly received and reordered, data values are loaded into their corresponding COMMON blocks. In this fashion the data transfer is isolated from and transparent to the original code.

A coarse-grain choreographer to coordinate the transfer of data and solution commands was also necessary. The coarse-grain commands consist of a series of bias conditions (each bias condition requires a separate nonlinear solution) to be solved by the SA. The UI choreographer interfaces with the main dispatch loop of PISCES (the routine which parses user's commands and calls the appropriate subroutines) and intercepts commands that require remote processing. The UI choreographer then transfers the data necessary for the computation requested followed by the actual command. It then awaits notification from the SA choreographer that the command has been executed and receives the results. At the other end, the SA choreographer waits to receive messages, initiates the requested action, and returns the results. The changes to the original code were minimal since the choreographer is entirely external to both the UI and the SA. Only the dispatch loops were modified to interface with the choreographer.

3.3 Adding Domain Decomposition

We expose the parallelism in the problem through grid decomposition. Naturally, the first task for parallelization was to add a domain decomposition module (DDM) to the UI. At runtime, the user's grid structure will be given to the DDM which will then spread the grid among the processors. An example of grid-based decomposition is shown in Figure 5. The CMOS structure described earlier has been divided into sixteen partitions. The partitioning was created using the recursive spectral bisection (RSB) algorithm of Pothen *et. al.* [14]. The goal of the DDM is to divide the work equally among the processors while minimizing the communication by keeping boundaries small. We chose to use the RSB algorithm as the heart of our DDM. The DDM consists of the RSB code, a pre-processor to convert a PISCES grid description into an RSB grid

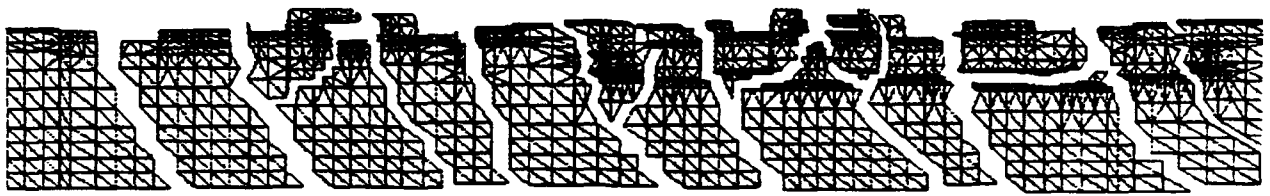


Fig. 5: 16-processor decomposition using Recursive Spectral Bisection.

description, and a post-processor to correct any inconsistencies in the partitioning. The changes to the UI code were minor: a call to the DDM after the grid structure is loaded produces the partitioning.

RSB was designed to handle homogenous grids (i.e., all nodes in the grid have equal characteristics). The addition of boundary conditions in the form of external electrodes (Figure 6) can

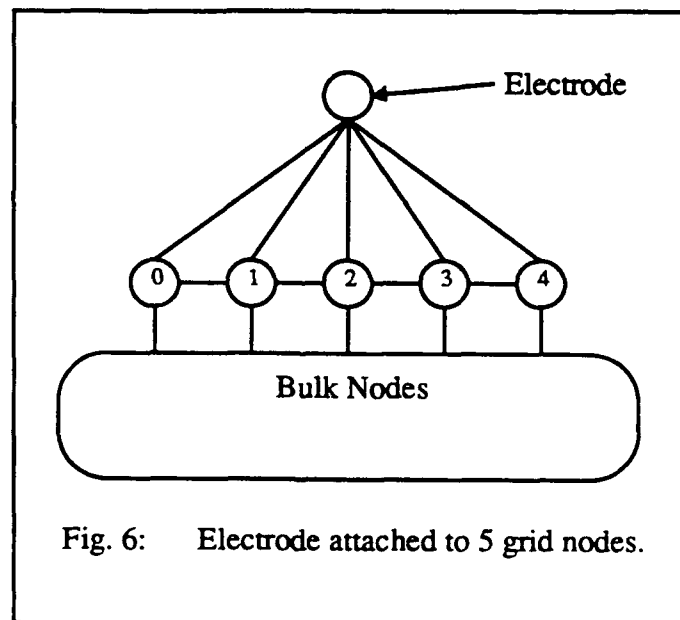
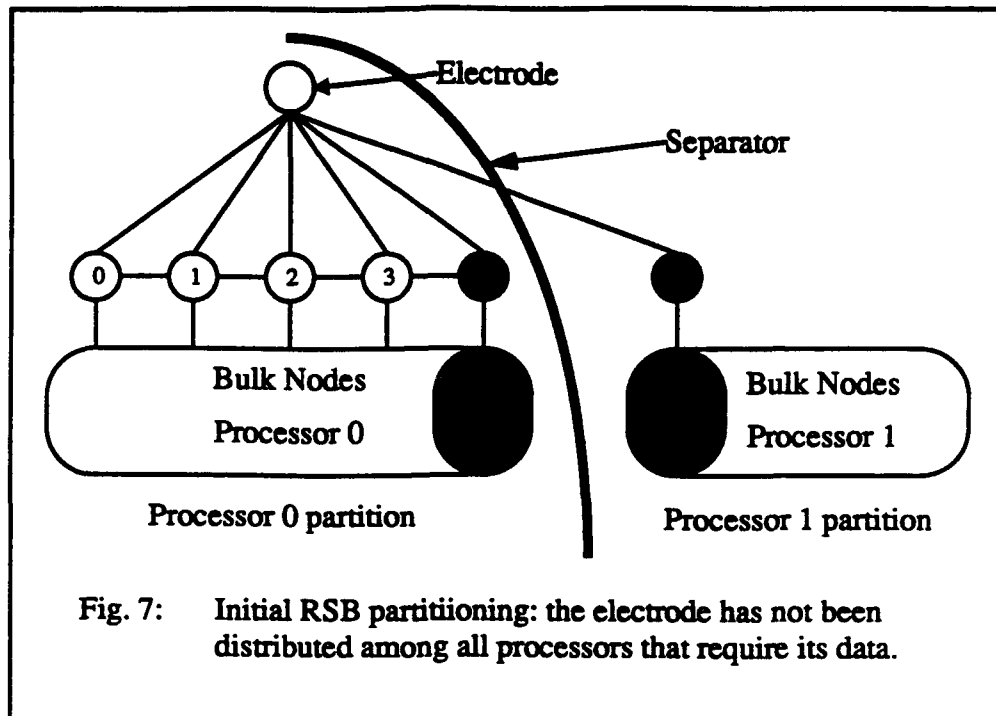
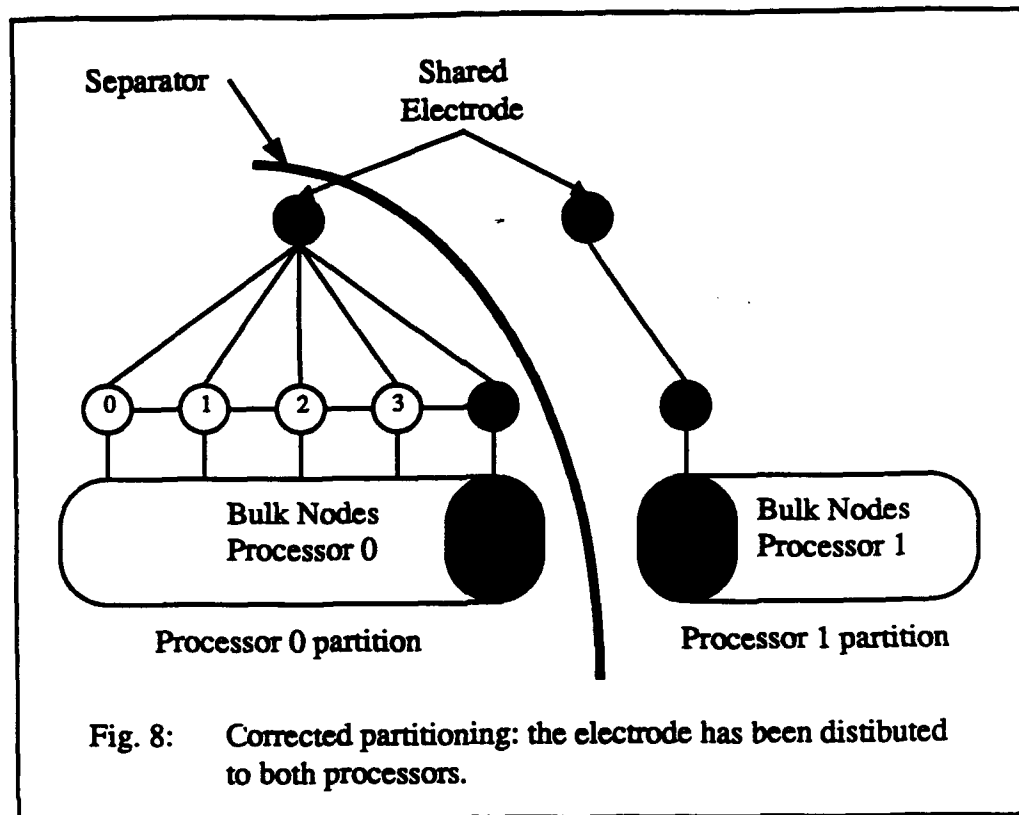


Fig. 6: Electrode attached to 5 grid nodes.

create inconsistencies which must be corrected by post-processing the RSB partitions. The electrodes are included in the grid description passed to the DDM; however, they are different from other nodes in the grid since they affect the assembly of all nodes connected to them. If the set of nodes attached to an electrode is spread across multiple processors by the domain decomposer (Figure 7), irregularities will occur if the electrode is not designated as shared among the processors. This can easily occur since the RSB cannot guarantee the an electrode will be divided among all partitions containing attached nodes. The post-processor examines the partitioning and insures



that all processors involved with nodes attached to an electrode are also involved with that electrode. The result of applying this operation causes electrodes to be shared across all involved processors. (Figure 8) and the boundary information to be distributed properly.

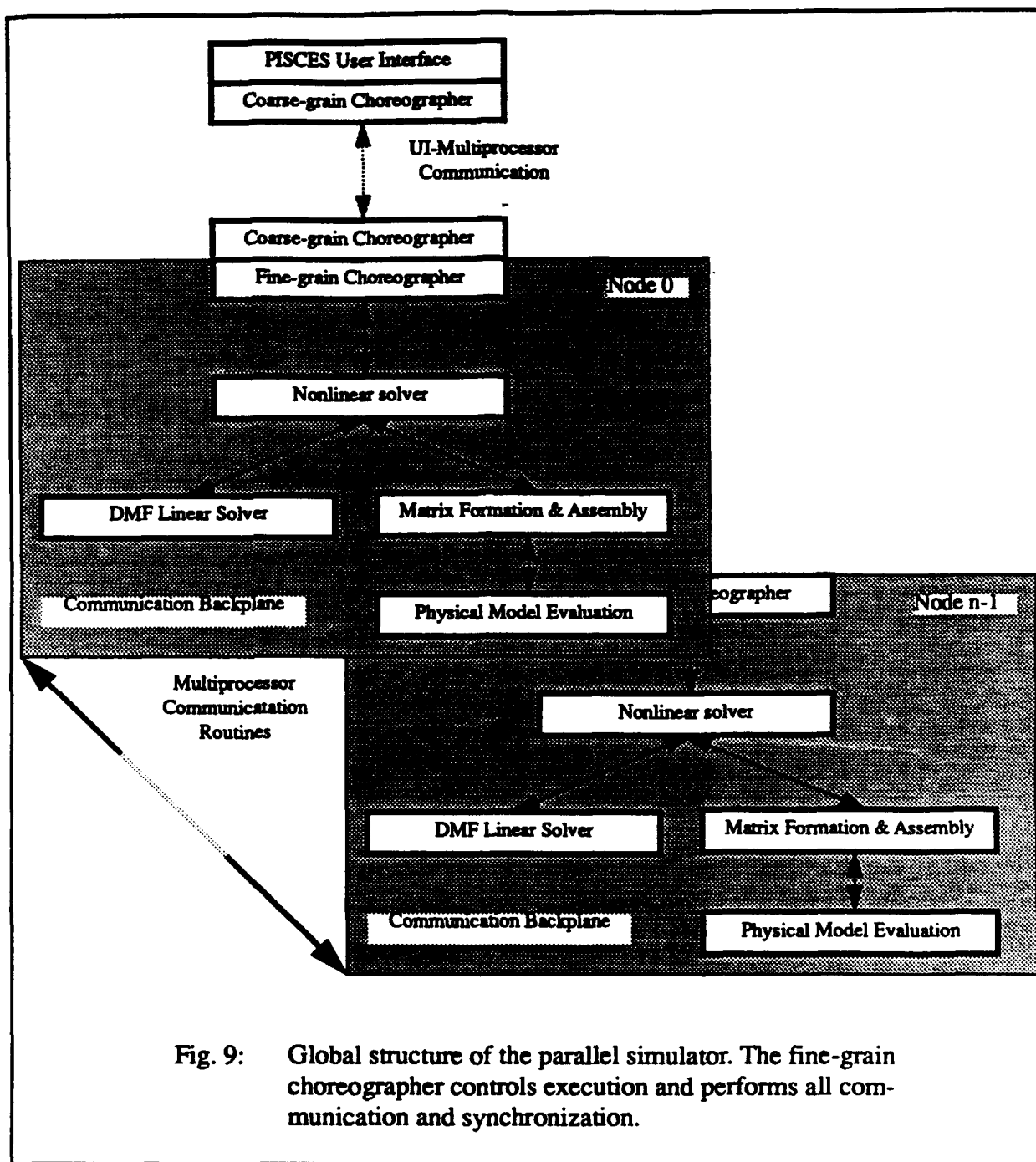


3.4 Parallel Data Transfer and Control

Once the UI has the ability to partition grids for parallel solution, it must be modified to send data to all processors in the parallel partition. Since it would be difficult to manage a separate TCP/IP connection to each processor (some machines have several hundred or more), the UI continues to connect to only one processor, *node_0*. Data to other nodes is sent through *node_0* and forwarded to the destination via the parallel message passing library. The Node Executive (NX), Intel's message passing library[21], is much faster than the TCP/IP connection making the extra hop through *node_0* relatively inexpensive.

Once the data is distributed, we can begin to parallelize the computations. We intend to have each processor run a copy of the SA on a subgrid, communicating with other processors to resolve data dependencies. To manage this communication and synchronization we designed a fine-grain choreographer to coordinate the processors much in the same fashion as our coarse-grain choreographer coordinates actions between the UI and SA. *Node_0* oversees the parallel action by sending cues the other nodes to initiate communication, synchronization, and computa-

tion. The structure of the parallel simulator is shown in Figure 9.



3.5 Nonlinear Solution

The nonlinear solver creates an initial guess at the solution and then repeatedly asks for a new Jacobian matrix and residual vector to be formed, requests the linear solution of this system, and applies the resulting update until the convergence criteria are met. In the nonlinear solver

module itself, only the initial guess routines require parallel communication. The matrix formation routines and the linear solver reside in separate modules and will be discussed later.

The initial guess routines need non-local information when creating a new guess at the solution based upon bias conditions. Once again, the boundary conditions cause problems. For example, to compute the extrapolation factor for a projection properly, all bias conditions must be examined. The projection is scaled by the differences between the new bias conditions and the previous values. To obtain this information, it is necessary to have a global picture of the boundary conditions. We evaluated two possible solutions:

- Add a communication phase to the algorithm to make all processors agree upon the correct extrapolation factors. This approach has the disadvantages of adding communication overhead and requiring a moderate amount of code modification.
- Add extra data structures, duplicated on each processor, which contain global boundary condition information. This has the disadvantage of requiring additional data structures to store the duplicated global information. It has the advantages of requiring no communication and minimizing the code changes. (We simply modify the call to the projection algorithm so that the global, rather than the local, boundary conditions are passed.)

We chose the latter solution for three reasons: existing data structures were unaffected, the code modifications were minimal, and the amount of extra data was small and did not impact memory requirements measurably.

3.6 Matrix Form and Assemble

The assembly process proceeds in a finite element fashion, evaluating on a triangle-by-triangle basis. Each triangle forms its local contribution which is then added to the global system. If all information needed during assembly is local to a triangle, assembly may take place independently on all processors. This is usually the case; however, some of the physical models do require non-local data. Unlike in the initial guess routines, there is no simple way to avoid communicating to resolve non-local references as the data needed may be constantly updated by its owner.

Semiconductor devices often contain different materials within a single structure. One example is a silicon-oxide interface (Figure 10) common in MOS structures [9]. The mobility of the carriers along these interfaces differs from the mobility in bulk silicon. If the user desires to model this mobility variation, each silicon triangle along the interface must have knowledge of the material type and present electrical parameters of its neighbors in order to properly compute mobility at the interface. We were forced to modify this model so that interface triangles could

obtain the material parameters of neighboring triangles on remote processors. Before evaluating this mobility model, the fine-grain choreographer initiates a communication phase to exchange the data required for model evaluation. The model itself was modified to access a temporary buffer containing the non-local information. By performing all communication beforehand, code changes were again minimized. Fortunately, models of this type are rare in the current release version of PISCES.

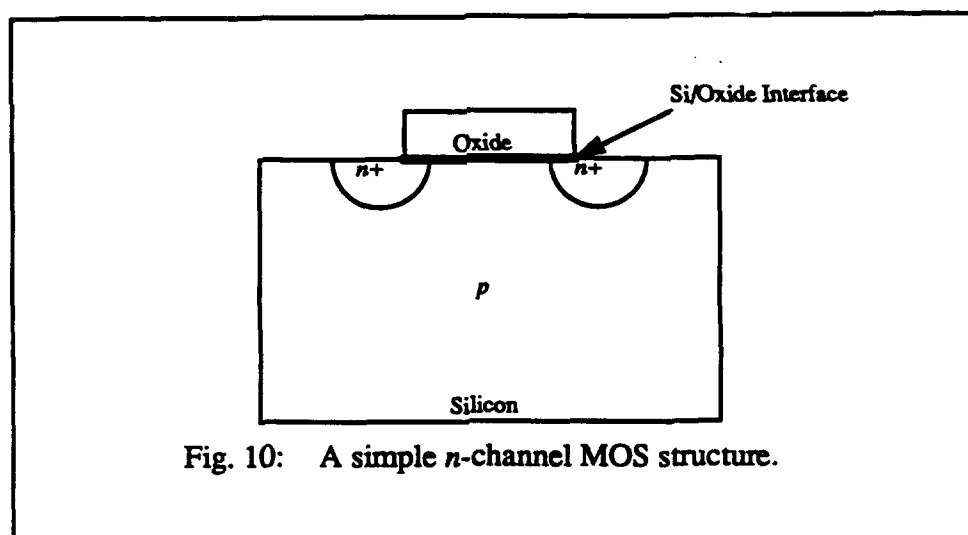


Fig. 10: A simple n -channel MOS structure.

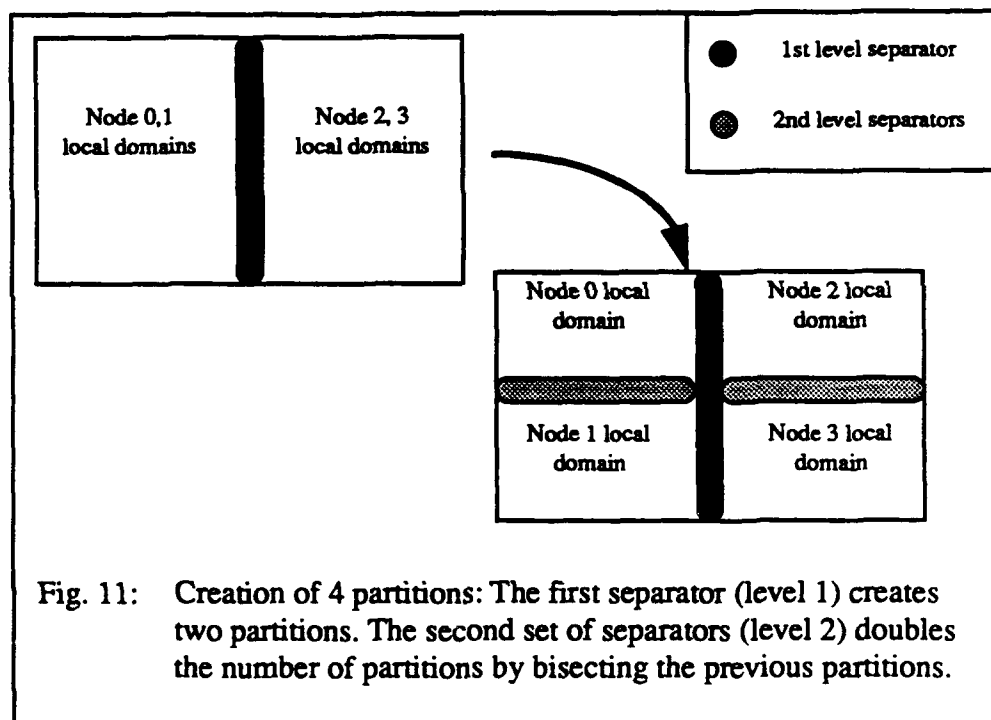
3.7 Linear Solution

The linear solution code is distinct from the rest of the simulator and has a well-defined interface. The linear solver accepts a matrix and a right-hand-side vector from the nonlinear solver and returns a solution vector. As long as the solution is correct and computed efficiently, the linear solver can be treated as a “black box” by the rest of the application. The serial version of PISCES uses a public domain sparse direct solver[7]. Although written more than fifteen years ago, it continues to be quite efficient on most serial architectures. However, adapting this code for parallel execution is problematic as the algorithms and data structures do not map well to modern distributed-memory architectures. Fortunately, the segmentation between the linear solver and the remainder of the code allows us to swap one “black box” for another one with identical behavior. The replacement of the linear solver had no effect on the remainder of the code.

We replaced the existing solver with a distributed multi-frontal (DMF) solver [23, 24] developed in our lab and modelled after the work of [18]. Sparse direct methods generally contain two phases: a symbolic factorization followed by a series of numeric factorizations. The symbolic phase first reorders the equations to minimize non-zero fill-in during elimination. This also mini-

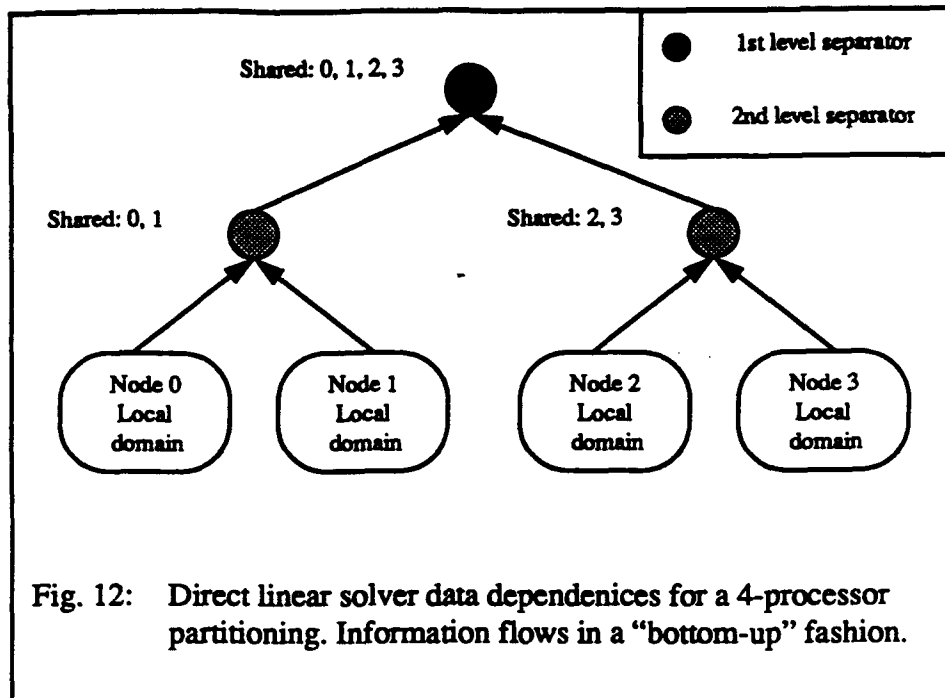
mizes the floating point operations to compute the factor and the memory required to store the factor. Once a reordering has been computed, the matrix is symbolically factored to determine the exact non-zero structure of the factor. Using this information, the requisite data structures are created. The numeric factorization performs the actual floating point work using the elimination ordering and storage prepared by the symbolic phase.

In parallel, the symbolic and numeric factorizations become more complicated due to both the constraints on elimination order posed by the separators and the communication required to factor them. RSB creates partitions in a top-down fashion (Figure 11). We exploit this behavior when computing the order of elimination by factoring the local domains independently followed by the sets of RSB separators. The sets of separators are factored in the inverse order in which they were created by RSB (Figure 12).



To accommodate the constrained order of elimination, a two-phase reordering is used. First, the multiple minimum degree (MMD) algorithm [25] is used to reorder each processor's local domain. Afterward, equations within the separators are added to the elimination ordering. The interiors are factored independently; but, the processors must communicate to factor the separators.

The non-zero fill-in during a sparse direct factorization adds coupling between equations residing on different processors. To insure a correct linear solution, processors must have knowl-



edge of some equations that are resident on other machines. The UI adds the extra grid points that generate these equations to the processors' data. These grid points are used solely by the linear solver and have no effect upon the remainder of PISCES.

3.8 Retrospective: What additional permanent data structures were necessary

In the previous sections we have described the procedures we followed for implementing our parallel PDE solver. To provide a complete picture of the parallelization process, it is useful to describe the major data structures added during development. All of these data structures are used by the choreographers. They are external to PISCES and have no effect upon the existing data structures. They are summarized below:

- *local_to_global_map*[1..#local grid points]: Each processor keeps an array which maps the local node numbering of its subgrid to the global node numbering. All communication among the processors uses the global numbering system.
- *owner*[1..#local grid points]: The array used to determine the processor in charge of accumulating updates to each shared grid point. The owner is responsible for communicating a consistent set of values for the grid point to other processors sharing the node.
- *share_count*[1..#local grid points]: This array, in conjunction with *share_list*, provides a complete picture of the interactions with other processors for each the shared grid point. The data is stored using a quotient list structure [24]. For each grid point, *share_count*

points to the starting location in *share_list* of that node's list of involved processors (Figure 13). The number of processors sharing node *i* is given by *share_count*[*i*+1] - *share_count*[*i*].

- *share_list*[1..*share_count*[#points+1]]: This array (Figure 13) contains lists, one for each grid point, of the processors sharing a grid point. It is indexed by *share_count*.

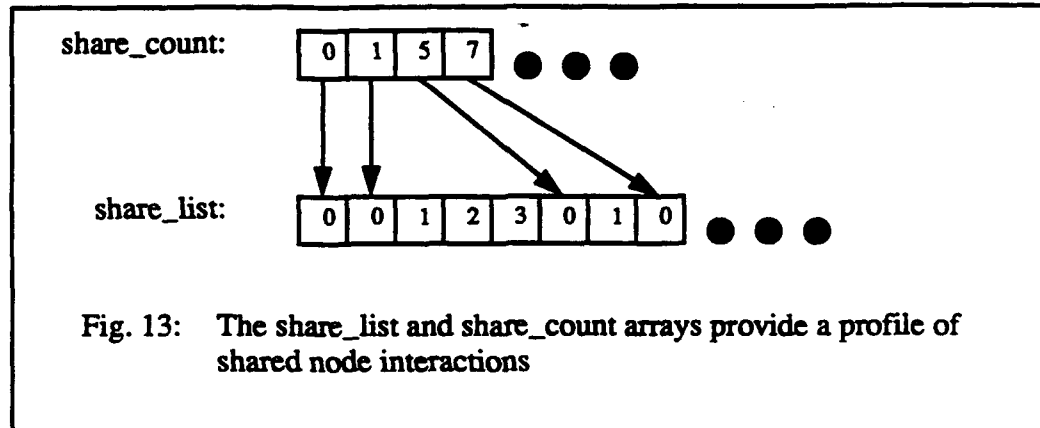


Fig. 13: The *share_list* and *share_count* arrays provide a profile of shared node interactions

- *separator_level*[1..*#local* grid points]: Tells which level separator a shared grid point belongs in. It is used exclusively by the linear solver as described earlier.
- *Global Electrode Data*: This is a complete set of boundary condition data for the global system to insure consistency across the processors as discussed above.
- *elem_owner*[1..*#global* triangles]: This array is used solely by the UI program to determine the proper destination for each triangle. It is not sent to the SA.

3.8 Porting to the CM-5

To prove that our methodology creates applications that can be easily ported to other parallel computers supporting the SPMD model, we moved the code to the CM-5. Although the CM-5 was originally designed to support the dataparallel programming model[26], it also supports the SPMD programming model through its CMMD message passing library[27]. Each processing node consists of a SPARC processor controlling four floating-point vector units. Unfortunately, the vector units can only be utilized by programs written using a dataparallel language[28]. As declared earlier, we were unwilling to make wholesale changes to the code. This meant that we were confined to the SPARC processor and unable to access the high-performance vector units from our application. However, the machine still offers reasonable performance as demonstrated in our results section.

The porting process was surprisingly straightforward. Most of the work involved changing

the UI/SA communication layer. Since the CM-5 has one (or more) SUN workstation equivalents as control processors, we chose to run the UI on a control processor rather than a remote workstation. The control processors provided acceptable performance and provided a 20Mb/s connection connected to the attached parallel machine, a considerable improvement over TCP/IP communication. Also, the host can communicate directly with all nodes, making the intermediate hop through *node_0* unnecessary. The host uses the same byte ordering as the parallel processors, making byte reordering unnecessary.

Otherwise, we translated NX calls into corresponding CMMD calls. In most cases, a simple wrapper was sufficient and the existing code was left untouched. In rare cases, such as the hypercube specific *cubedim()* function, we substituted functions more appropriate to the CM-5. The porting work required less than two weeks and minimal code changes were necessary.

4. Results

Using the new parallel device simulator, we can scale the computational resources to match problem requirements. This should provide not only the ability to simulate existing grids in less time but also the ability to simulate large, complex grids which are computationally infeasible on serial architectures. In order to demonstrate the utility of our new parallel application, we ran the simulation of the CMOS inverter structure described in Section 2 (a total of 29 bias steps) on several different machines. We simulated four varying grid sizes as described in Table 1. They

Table 1: Simulation grid statistics

	2K Grid	8K Grid	11K Grid	18K grid
Number of grid points	1,607	7,844	10,728	18,503
Number of triangles	2,963	15,158	20,873	36,400
Number of equations	4,821	23,532	32,184	55,509

comprise a realistic set of grids to study various characteristics of this complex device. The lower resolutions are suitable to model the macroscopic behavior. The finer resolutions are required to capture small-scale effects within the device.

Only the CM-5 had sufficient memory to simulate the 18k grid. The iPSC/860 contains enough total memory to run the 18K grid on 16 or 32 processors; however, a poor load balance caused a small subset of the processors to run out of memory. In the following tables, an "N.A."

entry indicates that insufficient memory existed to run the simulation.

4.1 Serial Timings

To provide a set of serial benchmarks for comparison, we ran the simulation on the three fastest serial computers available in our lab. The simulation times are shown in Table 2. Not sur-

Table 2: Serial Solution Times For CMOS Inverter

	2K Grid	8K Grid	11K Grid
SUN 4/670	1,214s	40,032s	N.A.
IBM RS/6000 Model 530H	417s	11,263s	23,822s
IBM RS/6000 Model 560F	279s	7,239s	15,687s

prisingly, the IBM Model 560F, the fastest floating point processor, completed the simulations in the shortest time. All of the comparisons between the serial simulator and the parallel simulator are based upon the corresponding time on the Model 560F. The other solution times are included for reference.

4.2 Intel iPSC Timings

We ran the parallel code using the SUN 4/670 as the front end for our 32-processor Intel iPSC/860. For the small number of bias points in this simulation, the setup time using TCP/IP contributes greatly to the runtime. For example, the data transmission to set up a 32-node partition takes roughly fifteen minutes. Had we run a more thorough simulation, the setup time would have been better amortized. The timings shown in Table 3 include all setup time. To provide a better understanding of the asymptotic behavior of the parallel code, Table 4 contains timings with the data transmission times subtracted. In Figure 14, we have plotted the best serial solution time for each grid size as well as the best iPSC solution times both including and discounting the setup communication overhead.

Table 3: iPSC Solution Times for CMOS Inverter

	2K Grid	8K Grid	11K Grid
iPSC/860 1PE	1,515s	N.A.	N.A.
iPSC/860 2PE	829s	N.A.	N.A.

Table 3: iPSC Solution Times for CMOS Inverter

	2K Grid	8K Grid	11K Grid
iPSC/860 4PE	635s	N.A.	N.A.
iPSC/860 8PE	597s	2,687s	3,611s
iPSC/860 16PE	799s	1,900s	3,231s
iPSC/860 32PE	1279s	2,298s	3,107s

Table 4: iPSC Solution Times for CMOS Inverter without TCP/IP

	2K Grid	8K Grid	11K Grid
iPSC/860 1PE	1,480s	N.A.	N.A.
iPSC/860 2PE	778s	N.A.	N.A.
iPSC/860 4PE	522s	N.A.	N.A.
iPSC/860 8PE	372s	2,462s	3,386s
iPSC/860 16PE	349s	1,450s	2,781s
iPSC/860 32PE	375s	1,394s	2,203s

For the 2K grid, the iPSC cannot outperform the IBM 560F. This first observation begs the question: why did we develop a parallel simulator? The answer is: we need the ability to solve existing large problems quickly and the ability to solve larger problems which are computationally infeasible on current serial machines. For small problems, the overhead of setup, communication, and synchronization may make some simulations run slower in parallel than serially. The 2K grid is such a problem. However, the 8K grid is more encouraging. We achieve 4x speedup over the IBM including overhead and greater than 5x speedup with overhead discounted. The 11K grid is more encouraging still. We obtain a 5x speedup with overhead and a 7x speedup without it.

Figure 14 shows that as grid sizes increase, the solution times of the parallel code increase much more slowly than the serial solution times. The ability to scale more efficiently with grid

size will prove significant as the sizes and runtimes of simulations grow.

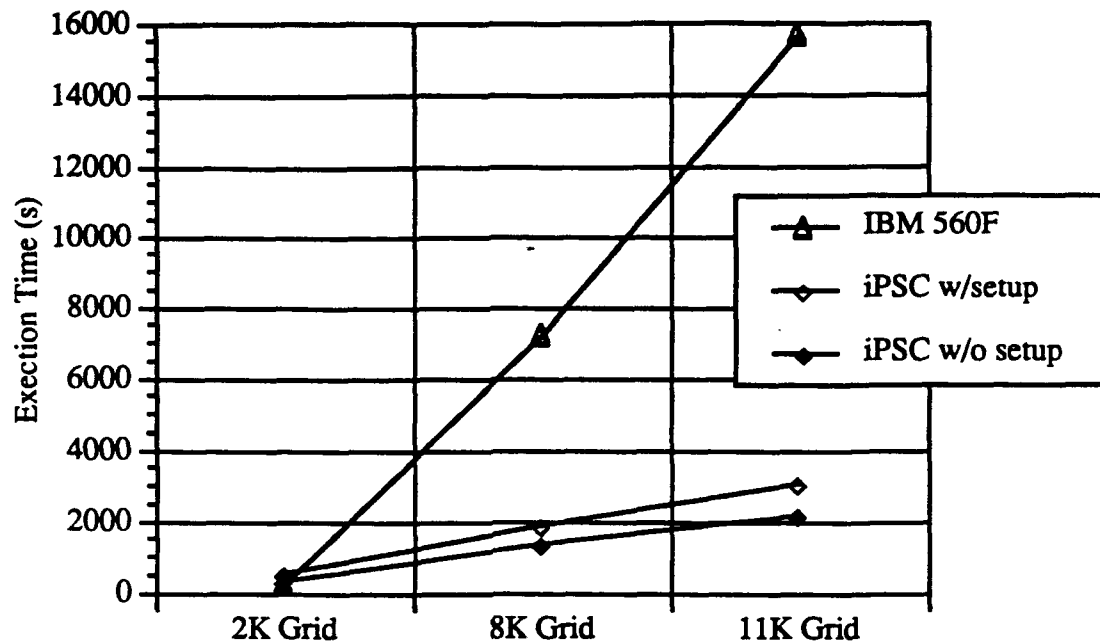


Fig. 14: Graph of the best solution times for the IBM 560F and the iPSC/860.

4.3 CM-5 Timings

We ran the simulations on two CM-5e's. (The CM-5e uses the newer SUN Viking processor in place of the original SPARC processors.) The CM-5 manages to barely outrace the IBM

Table 5: CM-5 Solution Times for CMOS Inverter

	2K Grid	8K Grid	11K Grid	18K Grid
CM-5e 32PE	262s	1,150s	1,792s	3,054s
CM-5e 64PE	270s	832s	1,232s	2,094s

560F on the 2K grid. For the 8K grid and the 11K grid, we observe speedups over the IBM of nearly 9x and 13x, respectively. Finally, with the 18K grid, we demonstrate the ability to simulate problems computationally infeasible on our serial machines. The ability to scale computational resources makes this simulation possible. Plotting the data (Figure 15) reinforces our assertion

that the parallel simulator offers superior scalability as problem sizes increase.

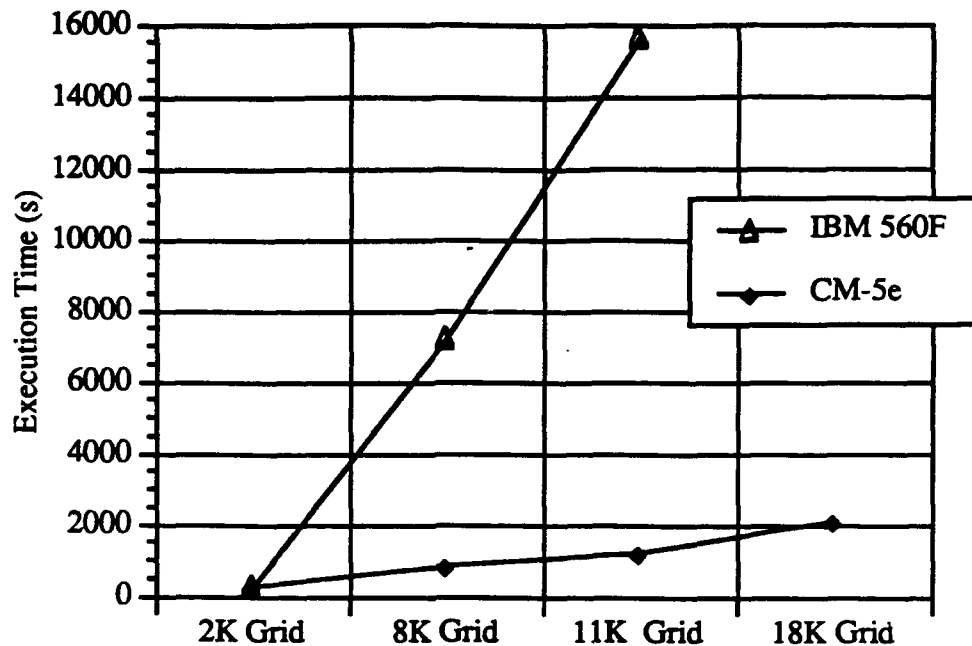


Fig. 15: Graph of the best solution times for the IBM 560F and the CM-5e.

5. Conclusions

In this paper, we have demonstrated that PISCES, a commercially-supported “dusty-deck” PDE solver, can be adapted for parallel execution with minimal changes to the existing serial code. In this fashion, we have retained the value captured in the long-term development of the program. The proven portability of our methodology allows the code to easily migrate to other distributed-memory architectures. We believe the methodology described in this paper should work well for any PDE solver with a similar program structure.

Maintaining the original code and data structures does not prevent the parallel implementation from providing striking reductions in simulation times for realistic large grids. We have already demonstrated order of magnitude (or more) improvements in simulation times for current problems as well as the ability to simulate grids too large for our serial computers. In fact, the behavior of the parallel code as grid sizes increase suggests a vast potential for simulating large and ultra-large structures. In the semiconductor industry, the competitive advantage gained could have immeasurable benefits.

Acknowledgments

We would like to thank Horst Simon for providing the spectral partitioning code. We also like to thank Lennart Johnsson and Thinking Machines Corporation for providing access to their computing resources. This research was sponsored by the State of California's Department of Commerce under grant number C90-072 and by DARPA.

References

- [1] M.T. Heath, "The hypercube: a tutorial overview," In *Hypercube Multiprocessors*, pp.7-10, SIAM, Philadelphia, PA, 1986.
- [2] M.R. Pinto, C.S Rafferty, and R.W. Dutton, "PISCES-II: Poisson and continuity equation solver," Stanford Electronics Lab, Stanford Univ., Stanford, CA, 1985.
- [3] M.R. Pinto, C.S Rafferty, H.R. Yeager, and R.W. Dutton, "PISCES-IIB Supplementary Report," Stanford Electronics Lab, Stanford Univ., Stanford, CA, 1985.
- [4] B.V. Ghokale, "Numerical solutions for one-dimensional silicon n-p-n transistor," *IEEE Trans. Electron Devices*, Vol. ED-17 (1970), pp. 594-602.
- [5] H.K. Gummel, "A self-consistent iterative scheme for one-dimensional steady state transistor calculations," *IEEE Trans. Electron Devices*, Vol. ED-11 (1964), pp. 455-465.
- [6] C.S. Rafferty, M.R. Pinto, and R.W. Dutton, "Iterative methods in semiconductor device simulation," *IEEE Trans. Electron Devices*, Vol. ED-32 (1985), pp. 2018-2027.
- [7] D.A. Calahan and P.G. Buning, "Vectorized General Sparsity Algorithms with Backing Store," SEL Report 96, Systems Engineering Laboratory, University of Michigan, Ann Arbor, MI, 1977.
- [8] Z. Johan, "Data parallel finite element techniques for large-scale computational fluid dynamics," Ph.D. Thesis, Stanford University, 1992.
- [9] B. Nour-Omid, A. Raefsky, and G. Lyzenga, "Solving finite element equations on concurrent computers," *Procedures of the Symposium on Parallel Computations and their Impact on Mechanics*, Boston, 1987, ASME, 1988.
- [10] K. Wu, R.F. Lucas, Z. Wang, and R.W. Dutton, "New approaches in a 3-D one-carrier device solver," *IEEE Trans. CAD*, Vol. 8, No. 5 (1989), pp. 528-537.
- [11] V. Venkatakrishnan, H.D. Simon and T.J. Barth, "A MIMD implementation of a parallel Euler solver for unstructured grids," *The Journal of Supercomputing*, 6 (1992) 117-127.
- [12] P.J Hatcher and M.J. Quinn, *Data-parallel Programming on MIMD Computers*, MIT Press, Cambridge, MA, 1991.
- [13] H.D Simon, "Partitioning of unstructured problems for parallel processing," *Computing Systems in Engineering*, 2 (1991) 135-148.
- [14] A. Pothen, H.D. Simon, and K.P. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs," *SIAM J. Mat. Anal. Appl.*, 11 (1990), pp. 430-452
- [15] J.G. Malone, "Automated mesh decomposition and concurrent finite element analysis for hypercube computers", *Comp. Meth. Appl. Mech. Eng.*, Vol. 70, No., 1, (1988), pp. 27-58.
- [16] C. Farhat and M. Lesoinne, "Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics," *Internat. J. Numer. Meth. Eng.*, Vol. 36, No. 5, (1993), pp. 745-764.
- [17] C. Farhat, "A simple and efficient automatic FEM domain decomposer," *Computers and*

- Structures, Vol. 28, No. 5 (1988), pp. 579-602.
- [18] R.F. Lucas, "Solving Planar Systems of Equations on Distributed-Memory Multiprocessors," Ph.D. Dissertation, Stanford Univ., 1987.
 - [19] J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, San Mateo, CA, 1990.
 - [20] iPSC CIO Ethernet Reference Manual, Intel Scientific Computers, Beaverton, OR, 1990.
 - [21] iPSC Concurrent Programming Reference Manual, Intel Scientific Computers, Beaverton, OR, 1990.
 - [22] R.S. Muller and T.I. Kamins, Device Electronics for Integrated Circuits, John Wiley and Sons, New York, 1977.
 - [23] I.S. Duff, "Parallel implementation of multifrontal schemes," Parallel Computing, Vol. 3, pp. 193-204.
 - [24] I.S. Duff, A.M. Erisman, and J.K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, London, 1986.
 - [25] A. George and J.W.H. Lui, "The evolution of the minimum degree algorithm," SIAM Review, Vol. 31, pp. 1-19, 1989.
 - [26] The Connection Machine CM-5 Technical Summary, Thinking Machines Corporation, Cambridge, MA, 1992.
 - [27] CMMD Reference Manual, Version 3.0, Thinking Machines Corporation, Cambridge, MA, 1993.
 - [28] CM Fortran Reference Manual, Version 2.0 Beta, Thinking Machines Corporation, Cambridge, MA, 1992.

An Approach to Construct Pre-Conditioning Matrices for Block Iteration of Linear Equations^[17]

Ze-Yi Wang, Ke-Chih Wu, and Robert W. Dutton, *Fellow, IEEE*

Abstract—The approximate block elimination method (ABE) for constructing pre-conditioning matrices in the block iteration of systems of linear equations is presented. Four families of pre-conditioning matrices based on this approach are identified for solving the 2×2 block equations with the Jacobian resulted from the 3-D one-carrier semiconductor device analysis problem. The fully coupled Newton's method is used as a test case on a parallel computer, the Intel iPSC/2 hypercube. There are 10 different pre-conditioning transformations including the alternate block factorization (ABF) and the modified singular perturbation (MSP) schemes. The results from computational experiments indicate that eight of these pre-conditioning matrices significantly extend the convergence region of the block Gauss-Seidel iteration process of Newton steps and five of them speed up the convergence with a factor of up to more than an order of magnitude.

I. INTRODUCTION

IN semiconductor device simulation a set of elliptic partial differential equations with appropriate boundary conditions should be solved approximately. After its discretization by using the finite difference method (FDM) or finite element method (FEM) a set of nonlinear algebraic equations is obtained. Gummel's and Newton's methods are frequently used to solve this system of nonlinear algebraic equations [1]. Gummel's method forms smaller and more tractable systems of equations; for very low bias voltages it shows good performance and is even better than Newton's method in terms of the total CPU expense [14]. When using quasi-Fermi variables it can converge well in a large region far from the solution [3], [4]. Newton's method is, however, often required to overcome slow convergence of Gummel's method in the strongly coupled situation such as high biases.

In Newton's method, a system of linear equations with the Jacobian should be solved. Generally, the size of this system is too big to use a direct method in the 3-D sim-

ulation. Even when it is based on some recent results using parallel computational methods, the direct solution suffers from both performance and CPU degradation [2]. So, it should be solved by an inner iteration. Block Gauss-Seidel (G-S) iteration is an obvious approach. Whole solution procedure of the nonlinear equation set is referred to as Newton-Gauss-Seidel method [8], Newton step for linearization of the nonlinear equation set, and G-S block iteration for the linear block equations. Its primary advantage comes from easy implementation and reduced storage. But its poor efficiency has been reported [7], [8]. The results in [7] show that the block G-S works well only at very low biases. It suffers from a difficulty similar to Gummel's method owing to the strong coupling between the subsystems which occurs when higher bias is applied. This means that Newton's method using the block G-S iteration without matrix transformation cannot obtain any advantage under conditions when Gummel's method fails. The reason for this is that the off-diagonal blocks are significant in comparison with the diagonal blocks of the Jacobian [7].

This suggests that the Newton-Gauss-Seidel method should use an efficient pre-conditioning transformation to reduce the off-diagonal entries as much as possible for improving the convergence of G-S block iteration. Two pre-conditioning strategies, the modified singular perturbation (MSP) method [6] and the alternate block factorization (ABF) technique [8], were proposed. Algebraically, the MSP is equivalent to a left pre-conditioning of the original system, and the ABF, a right pre-conditioning (postconditioning). The ABF approach used is a more general pre-conditioning technique which can accommodate $m \times m$ block equations while the MSP is only suitable for the 2×2 case. The success of the MSP encouraged further development and algebraical study of both the MSP and ABF techniques.

In this paper, a more general approach to construct block pre-conditioning matrices has been developed—we will refer to it as the approximate block elimination method (ABE). Both ABF and MSP methods can be derived from the ABE method. Section II discusses how to construct the pre-conditioning matrices by using the ABE method. The justification for this approach is discussed in Section III. Section IV presents experimental results for comparison. Some conclusions are given in Section V.

Manuscript received December 19, 1990; revised January 18, 1992. This work was supported in part by Intel Corporation, in part by Texas Instruments Inc., and in part by the U.S. Army Research Office under Contract DAAL03-87-K-0077. This paper was recommended by Associate Editor D. Rose.

Z.-Y. Wang was with the Integrated Circuits Laboratory, Stanford University, Stanford, CA. He is now with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 China.

K.-C. Wu and R. W. Dutton are with the Integrated Circuits Laboratory, Stanford University, Stanford, CA 94305.

IEEE Log Number 9200904.

II. CONSTRUCTION OF THE PRE-CONDITIONING MATRICES

We temporarily leave the semiconductor simulation in Sections II and III for more generality and convenience. In this section, the idea of the ABE is presented and then several examples with some trail of our background problem will be given to illustrate our idea. The solution of a general system with $m \times m$ block equations is considered. Suppose

$$Bx = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = b \quad (2.1)$$

where $x_1 \in R_1^n, x_2 \in R_2^n, \dots, x_m \in R_m^n$.

As the first step in the ABE, we construct an approximate matrix K to the original matrix B . One of the requirements for this construction is that the resulting approximate equation:

$$Kx = b \quad (2.2)$$

is easy to solve. The second requirement is that the approximate matrix K is as close to B as possible. We denote the approximation of matrix element B_{11} by K_{11} , and similarly for the other terms. The approximate matrix K can then be written as follows:

$$K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1m} \\ K_{21} & K_{22} & \cdots & K_{2m} \\ \vdots & \vdots & & \vdots \\ K_{m1} & K_{m2} & \cdots & K_{mm} \end{bmatrix} \quad (2.3)$$

The second step in ABE is to determine a block elimination matrix C which changes K into one of the three forms ($\bar{K}$'s) listed below:

$$\bar{K}_a = \begin{bmatrix} \bar{K}_{11} & 0 & \cdots & 0 \\ 0 & \bar{K}_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \bar{K}_{mm} \end{bmatrix} \quad (2.4)$$

or

$$\bar{K}_b = \begin{bmatrix} \bar{K}_{11} & 0 & \cdots & 0 \\ \bar{K}_{21} & \bar{K}_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \bar{K}_{m1} & \bar{K}_{m2} & \cdots & \bar{K}_{mm} \end{bmatrix} \quad (2.5)$$

or

$$\bar{K}_c = \begin{bmatrix} \bar{K}_{11} & \bar{K}_{12} & \cdots & \bar{K}_{1m} \\ 0 & \bar{K}_{22} & \cdots & \bar{K}_{2m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \bar{K}_{mm} \end{bmatrix} \quad (2.6)$$

To get any one of three eliminated forms of $\bar{K}$ from K we can use two different matrix operations. The first matrix operation, i.e., the left-block elimination, means

$$C_L K = \bar{K} \quad (2.7)$$

where C_L presents the block elimination matrix from the left of K . The other operation, i.e., the right-block elimination, means

$$K C_R = \bar{K} \quad (2.8)$$

where C_R presents the block elimination matrix from the right of K . C_L and C_R are dependent on selection of $\bar{K}$'s, i.e., $\bar{K}_a$, $\bar{K}_b$, and $\bar{K}_c$.

For example, by using a left-block elimination matrix C_L the approximate equation (2.2) is changed into the following form:

$$\bar{K}x = C_L b \quad (2.9)$$

where $\bar{K}$ is one of three block forms given in (2.4)–(2.6). That is to say that C_L can eliminate the off-diagonal blocks of the coefficient matrix K in (2.2) completely or partly and let matrix K become one form of the block diagonal $\bar{K}_a$, the block lower triangular $\bar{K}_b$ and the block upper triangular $\bar{K}_c$. It means that C_L accelerates the block G-S iteration of the approximate equation (2.2) extremely. Hence it can be expected that C_L , as a pre-conditioning matrix, is able to partly recouple the original equation (2.1), i.e., reduce the off-diagonal blocks of matrix B to a certain extent. We call C_L the left pre-conditioning matrix. Similarly, C_R is called the right pre-conditioning matrix here or the postconditioning matrix in [8].

Determination of the block elimination matrices C_L or C_R is an important and difficult problem. The ABF proposed by Bank *et al.* [8] gave an approach for finding an alternate blocking by using a permutation matrix. It can handle the case $K_{ij} = \text{diag}(B_{ij})$, ($i, j = 1, \dots, m$). Unfortunately, it is hard to say presently whether it can be suitable for general cases. However, it should be pointed out that the difficulty for determining C_L or C_R has been reduced because $\bar{K}$ has a more general form, i.e., it contains not only $\bar{K}_a$ but also $\bar{K}_b$ and $\bar{K}_c$.

In the following we will illustrate the procedure through several examples and point out some important considerations along the way.

Pre-Conditioning From the Left

First, consider the construction of the left pre-conditioning matrix C for the 2×2 block equation. With an approximate matrix K of the form (2.3), denote that the

block elimination matrix C is of the form:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

We now multiply both sides of the approximate equation (2.2) from the left by matrix C :

$$CKx = Cb \quad (2.10)$$

which gives

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Assume $C_{11} = C_{22} = I$, and the existence of K_{11}^{-1} . If we choose to change K into the form given by $\bar{K}_c$ and let $C_{12} = 0$, then, we have

$$C = \begin{bmatrix} I & 0 \\ -K_{21}K_{11}^{-1} & I \end{bmatrix} \quad (2.11)$$

and (2.10) becomes

$$\begin{bmatrix} K_{11} & K_{12} \\ 0 & K_{22} - K_{21}K_{11}^{-1}K_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - K_{21}K_{11}^{-1}b_1 \end{bmatrix}. \quad (2.12)$$

Since K_{21} in the approximate equation (2.2) has vanished by using matrix C , as seen in (2.12), the block G-S iteration needs to solve two block diagonal equation sets of (2.12) sequentially only one time, i.e., an iteration method becomes a direct method. In other words, matrix C gives the block G-S iteration maximum acceleration for solution of the approximate equation (2.2). Using C as a left pre-conditioning matrix, (2.1) becomes

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21} - K_{21}K_{11}^{-1}B_{11} & B_{22} - K_{21}K_{11}^{-1}B_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - K_{21}K_{11}^{-1}b_1 \end{bmatrix}. \quad (2.13)$$

It is easy to see that $(B_{21} - K_{21}K_{11}^{-1}B_{11})$ of (2.13) is equal to 0 if $K_{11} = B_{11}$ and $K_{21} = B_{21}$. But under this condition it is likely that the solution of K_{11}^{-1} is difficult and the nonzero structure of $(B_{22} - K_{21}K_{11}^{-1}B_{12})$ given in (2.13) will be changed when compared with B_{22} , owing to the fact that the nonzero structure of a product matrix is usually more complex than that of the factor matrices. Therefore, although ideally the approximate matrix K should be the same as the original matrix B , a practical constraint is that it should be easy to invert the sub-blocks K_{11} and K_{22} .

To be specific, we choose the approximate matrix K_1 of B to be as follows:

$$K_1 = \begin{bmatrix} D_{11} & D_{12} \\ B_{21} & D_{22} \end{bmatrix} \quad (2.14)$$

where D_{11} is the diagonal part of B_{11} , and similarly for the other terms. We should then choose a pattern of the

block elimination form $\bar{K}_a$, $\bar{K}_b$, and $\bar{K}_c$. If we select $\bar{K}_c$ so as to recouple the off-diagonal block $K_{21} = B_{21}$, then the block elimination matrix C_1 can be obtained easily from (2.11):

$$C_1 = \begin{bmatrix} I & 0 \\ -B_{21}D_{11}^{-1} & I \end{bmatrix}. \quad (2.15)$$

The pre-conditioned block equation is the following:

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21}(I - D_{11}^{-1}B_{11}) & B_{22} - B_{21}D_{11}^{-1}B_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - B_{21}D_{11}^{-1}b_1 \end{bmatrix}. \quad (2.16)$$

The block G-S iteration can be then employed to solve this transformed block set (2.16), which is the MSP that was presented in NUPAD II and proved to be a rather good acceleration of the block G-S iteration in 3-D one-carrier device simulation [6].

Choosing a block elimination pattern of $\bar{K}_a$, we can obtain a different block elimination matrix C_2 :

$$C_2 = \begin{bmatrix} I & -D_{12}D_{22}^{-1} \\ -B_{21}D_{11}^{-1} & I \end{bmatrix}. \quad (2.17)$$

The transformed approximate equation becomes

$$\begin{bmatrix} D_{11} - D_{12}D_{22}^{-1}B_{21} & 0 \\ 0 & D_{22} - B_{21}D_{11}^{-1}D_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 - D_{12}D_{22}^{-1}b_2 \\ b_2 - B_{21}D_{11}^{-1}b_1 \end{bmatrix}$$

after using C_2 as a block elimination matrix. The effects of the transformation matrices C_1 and C_2 are almost the same, as demonstrated later by several device simulation examples in Table III. The reason for this will be discussed in Sections III and IV.

Other left pre-conditioning matrices, as listed in the Appendix, can be similarly constructed using different K and/or $\bar{K}$. We will now turn to discuss how to construct the right pre-conditioning matrices.

Pre-Conditioning From the Right

As an alternative, the right pre-conditioning matrices can also be constructed. For example, again choose K_1 in (2.14) as the approximate matrix to B and select a block elimination pattern of $\bar{K}_a$, the corresponding block elimination matrix C_3 is

$$C_3 = \begin{bmatrix} I & -D_{11}^{-1}D_{12} \\ -D_{22}^{-1}B_{21} & I \end{bmatrix}. \quad (2.18)$$

Suppose that C_3^{-1} exists, we have

$$K_1 C_3 C_3^{-1} x = b \quad (2.19)$$

or

$$(\bar{K}_1)_a y = b \quad (2.20)$$

where

$$(\bar{K}_1)_a = K_1 C_3$$

$$= \begin{bmatrix} D_{11} - D_{12} D_{22}^{-1} B_{21} & 0 \\ 0 & D_{22} - B_{21} D_{11}^{-1} D_{12} \end{bmatrix} \quad (2.21)$$

and

$$y = C_3^{-1} x. \quad (2.22)$$

C_3 can be used as a pre-conditioning matrix of (2.1) and the coefficient matrix of the pre-conditioned equation is

$$R = BC_3$$

$$= \begin{bmatrix} B_{11} - B_{12} D_{22}^{-1} B_{21} & B_{12} - B_{11} D_{11}^{-1} D_{12} \\ B_{21} - B_{22} D_{22}^{-1} B_{21} & B_{22} - B_{21} D_{11}^{-1} D_{12} \end{bmatrix}. \quad (2.23)$$

So we must solve

$$Ry = b \quad (2.24)$$

for the variable y first then get x from

$$x = C_3 y. \quad (2.25)$$

We can choose K_2 as another approximate form of the matrix B

$$K_2 = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \quad (2.26)$$

and construct several pre-conditioning matrices from the left and right by the idea similar to that given above. Here, we just mention one of them as the ABF named by Bank *et al.* [8] to illustrate how to construct the ABF by our idea which extends the method of Bank *et al.* Suppose the existence of K_2^{-1} and select $\bar{K}$ to be the identity matrix I , a special form of $\bar{K}_a$. Hence take $C_4 = K_2^{-1}$ as a recoupling matrix of the approximate equation

$$K_2 x = b \quad (2.27)$$

$$R = CB = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} - D_{21} D_{11}^{-1} B_{11} & B_{22} - D_{21} D_{11}^{-1} B_{12} & B_{23} - D_{21} D_{11}^{-1} B_{13} \\ B_{31} + C_{31} B_{11} + C_{32} B_{21} & B_{32} + C_{31} B_{12} + C_{32} B_{22} & B_{33} + C_{31} B_{13} + C_{32} B_{23} \end{bmatrix}.$$

from the right for changing the variables. We have

$$C_4 = K_2^{-1} = \begin{bmatrix} D_{22} Q & -D_{12} Q \\ -D_{21} Q & D_{11} Q \end{bmatrix} \quad (2.28)$$

where $Q = (D_{11} D_{22} - D_{12} D_{21})^{-1}$. The right pre-conditioned matrix of (2.1) is thus

$$R = BC_4$$

$$= \begin{bmatrix} (B_{11} D_{22} - B_{12} D_{21}) Q & (B_{12} D_{11} - B_{11} D_{12}) Q \\ (B_{21} D_{22} - B_{22} D_{21}) Q & (B_{22} D_{11} - B_{21} D_{12}) Q \end{bmatrix}. \quad (2.29)$$

Similarly, other pre-conditioning transformations for the 2×2 case can be constructed when other suitable

approximate matrices are formed and block elimination patterns are selected.

We now derive a pre-conditioning matrix as an example for the 3×3 block equation after detailing for the 2×2 case above. Suppose the block equation

$$Bx = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = b \quad (2.30)$$

where $x_1 \in R_1^n$, $x_2 \in R_2^n$, and $x_3 \in R_3^n$, and an approximate matrix of B in (2.30):

$$K_3 = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ B_{31} & D_{32} & D_{33} \end{bmatrix}. \quad (2.31)$$

We multiply both sides of the corresponding block equation with coefficient matrix K_3 from the left by using matrix C to let K_3 become a block upper triangular denoted by $\bar{K}_c$ in (2.6). That means

$$CK_3 x = Cb$$

and then

$$(\bar{K}_3)_c x = Cb.$$

Suppose $C_{11} = C_{22} = C_{33} = I$, $C_{12} = C_{13} = C_{23} = 0$ and the existence of D_{11}^{-1} and $(D_{22} - D_{21} D_{11}^{-1} D_{12})^{-1}$. It is easy to determine the rest of the elements in C and they are

$$C_{21} = -D_{21} D_{11}^{-1}$$

$$C_{31} = -((B_{31} D_{11}^{-1} D_{12} - D_{32})(D_{22} - D_{21} D_{11}^{-1} D_{12})^{-1} D_{21} + B_{31}) D_{11}^{-1}$$

$$C_{32} = (B_{31} D_{11}^{-1} D_{12} - D_{32})(D_{22} - D_{21} D_{11}^{-1} D_{12})^{-1}.$$

The matrix R pre-conditioned by C from the left is

If we take the approximate matrix of B as

$$K_4 = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \quad (2.32)$$

and choose a block elimination pattern of I . Then if K_4^{-1} exists, K_4^{-1} can become the pre-conditioning matrix named the ABF method [8]. It is easy to find that our method includes the ABF and greatly extends it. Thereby the ABE provides more flexibility for better using the features of the original coefficient matrix B in the construction of the pre-conditioning transformations. This means that we

have more choices in constructing the pre-conditioning matrices in two-carrier device simulation.

Similarly, we can construct other pre-conditioning matrices by using the idea above if the number of the blocks in the block equation is more than 3.

III. ANALYSIS OF THE IDEA FOR 2×2 BLOCK

We will show that the approach of constructing the pre-conditioning transformation matrices mentioned in Section II is reasonable under some requirements of matrix B in (2.1). The discussion is just limited in the situation of 2×2 , i.e., $m = 2$ in (2.1).

Suppose that $B_{11} \in R_1^n \times R_1^n$, $B_{22} \in R_2^n \times R_2^n$, and the strict diagonal dominance of both B_{11} and B_{22} . If B_{11} is written as

$$B_{11} = D_{11} + \epsilon_{11} \quad (3.1)$$

where $\epsilon_{11} \in R_1^n \times R_1^n$, the strict diagonal dominance of B_{11} means

$$\sum_j |(\epsilon_{11})_{ij}| < |(D_{11})_i|, \quad i = 1, \dots, n. \quad (3.2)$$

The strict diagonal dominance of B_{22} means an inequality similar to (3.2) holds

$$\sum_j |(\epsilon_{22})_{ij}| < |(D_{22})_i|, \quad i = 1, \dots, n \quad (3.3)$$

where $\epsilon_{22} = B_{22} - D_{22} \in R_2^n \times R_2^n$.

If using C_2 of (2.17) as a left pre-conditioning matrix of (2.1), we have

$$\begin{aligned} & \begin{bmatrix} I & -D_{12}D_{22}^{-1} \\ -B_{21}D_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} I & -D_{12}D_{22}^{-1} \\ -B_{21}D_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \end{aligned}$$

and can then rewrite the above equation as

$$\begin{aligned} Rx &= \begin{bmatrix} B_{11} - D_{12}D_{22}^{-1}B_{21} & \tilde{B}_{12} - D_{12}D_{22}^{-1}\epsilon_{22} \\ -B_{21}D_{11}^{-1}\epsilon_{11} & B_{22} - B_{21}D_{11}^{-1}B_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} b_1 - D_{12}D_{22}^{-1}b_2 \\ b_2 - B_{21}D_{11}^{-1}b_1 \end{bmatrix} \end{aligned} \quad (3.4)$$

where $\tilde{B}_{12} = B_{12} - D_{12}$. A significant thing arises in the matrix R which is transformed from B . Let's look into R_{21} by using the strict diagonal dominance, as shown in (3.2), of B_{11} :

$$\begin{aligned} \|R_{21}\|_{\infty} &= \|B_{21}D_{11}^{-1}\epsilon_{11}\|_{\infty} \\ &\leq \|B_{21}\|_{\infty} \|D_{11}^{-1}\epsilon_{11}\|_{\infty} < \|B_{21}\|_{\infty}. \end{aligned} \quad (3.5)$$

This means that the transformed block equation (3.4) has been recoupled by using the left pre-conditioning matrix C_2 to some extent. The smaller $\|D_{11}^{-1}\epsilon_{11}\|_{\infty}$ is, i.e., the stronger the strict diagonal dominance of B_{11} is, the better the effect of the left pre-conditioning transformation will be. The block G-S iteration for solving (3.4) needs to be

done only one time, in the extreme, if $\|D_{11}^{-1}\epsilon_{11}\|_{\infty} = 0$, i.e., B_{11} becomes a diagonal matrix. From (3.5) we see that the strict diagonal dominance of B_{11} can accelerate convergence of the block G-S iteration by using some suitable pre-conditioning transformations of the ABE method.

Besides, if B_{12} is a diagonal, then $B_{12} = D_{12}$ or $\tilde{B}_{12} = 0$. Under these conditions we have

$$R_{12} = -D_{12}D_{22}^{-1}\epsilon_{22} \quad (3.6)$$

$$R_{22} = B_{22} - B_{21}D_{11}^{-1}D_{12} \quad (3.7)$$

and a relationship $\|R_{12}\|_{\infty} < \|B_{12}\|_{\infty}$, similar to (3.5), holds because of the strict diagonal dominance of B_{22} . It means that the coupling between both variable sets x_1 and x_2 is further reduced and in turn the further acceleration of the block G-S iteration can be expected.

In 3-D one-carrier device simulation, the block matrices B_{11} , B_{22} , and B_{21} have the identical seven-diagonal structure resulted from the finite difference scheme for discretizing the equations. It can be seen that the nonzero structure of R_{22} is the same as B_{22} , and it is beneficial to solve the block equation with the coefficient matrix R_{22} resulting in a saving of both storage and CPU time.

The effects of the pre-conditioning may not always be beneficial when only one of the matrices B_{11} and B_{22} is strictly diagonal dominant or $\tilde{B}_{12} \neq 0$. For example, suppose that only B_{11} is strictly diagonal dominant and $B_{12} = D_{12}$, a left pre-conditioning matrix is

$$C_5 = \begin{bmatrix} I & -D_{12}D_{22}^{-1} \\ 0 & I \end{bmatrix}. \quad (3.8)$$

Then we have $R_{21} = B_{21}$ and $R_{12} = -D_{12}D_{22}^{-1}\epsilon_{22}$. It can be seen that B_{21} remains unchanged in the pre-conditioned matrix R ; and if B_{22} is a row zero-sum matrix, then R_{12} is merely D_{12} redistributed by $-D_{22}^{-1}\epsilon_{22}$. This means that the original block equation (2.1) does not benefit from the pre-conditioning matrix C_5 . The results for the device simulation case (Tables III and V) confirm the analysis here since we have the relationship:

$$C_5 = C_{1b}^1 = C_{1b}^2. \quad (3.9)$$

Although we have only presented the heuristic analysis on the pre-conditioning matrices of C_2 and C_5 , other transformations can be analyzed in a similar manner. Most conclusions still hold whether the left or right pre-conditioning matrices are formed from an approximate matrix of K_1 or K_2 . The discussion for device simulation will be given in detail in Section IV after the experimental results are presented.

The above analysis indicates that it is very important to improve the performance in iteratively solving block equation (2.1) by exploiting the characteristics of the original coefficient matrix B . On the other hand, it appears that more characteristics in the original Jacobian can be exploited in addition to the strict diagonal dominance, for example, the ABF with variables n and p has demon-

strated promising results [8] without the strict dominance of the diagonal blocks. This indicates the need to extend the above basic analysis of the ABF method.

IV. NUMERICAL EXAMPLES IN DEVICE SIMULATION

Having developed the underlying matrix techniques presented in Sections II and III, let us now return to the semiconductor problem. For the sake of simplicity, it is assumed that the electron current continuity equation is to be solved together with Poisson's equation. In normalized form, the equations are given by

$$\begin{aligned}\nabla \cdot (\epsilon \nabla \psi) &= n - p + N_A - N_D \\ \nabla \cdot J_n &= 0\end{aligned}\quad (4.1)$$

in Ω , where $n = \exp(\psi - \phi_n)$, $p = \exp(\phi_p - \psi)$ and $J_n = -q\mu_n n \nabla \phi_n$. We assume that Ω is a simply connected cube region. As can be easily verified, the normalization constants are: thermal voltage (kT/q) for electrostatic potential ψ and quasi-Fermi levels ϕ_n and ϕ_p , intrinsic carrier concentration n_i for carrier and impurity concentrations, and the intrinsic Debye length $\sqrt{\epsilon kT/(q^2 n_i)}$ for the spatial dimension. Boltzmann statistics are assumed as can be seen in the formulas for n and p . Tabulated doping dependent mobility values are used. At present, bandgap narrowing effects at high doping concentrations and field dependent mobility are neglected.

The simulation domain Ω is approximated by a 3-D rectangular grid. Equations (4.1) and (4.2) are discretized by the finite difference method. The numerical experiments were implemented in a hypercube multiprocessor, the Intel iPSC2™. Our recent experimental problem has been presented in detail in [6].

We employ Slotboom variables $\Phi_n = \exp(-\phi_n)$ and $\Phi_p = \exp(\phi_p)$ as the independent variables because of the known advantages [11], [12], [7]. To increase the bias range two scaling schemes which preserve the symmetry of main diagonal blocks in the Jacobian are used (see [6] for detail). The block equation with the Jacobian in each Newton step can be written as follows:

$$Bx = \begin{bmatrix} A_{\psi\psi} & D_{\psi\phi_n} \\ A_{\phi_n\psi} & A_{\phi_n\phi_n} \end{bmatrix} \begin{bmatrix} \Delta\psi \\ \Delta\Phi_n \end{bmatrix} = \begin{bmatrix} -F_\psi \\ -F_{\phi_n} \end{bmatrix} = b. \quad (4.3)$$

One primary advantage is that the matrix $A_{\psi\psi}$ in block equation (4.3) is strictly diagonal-dominant, symmetric, and positive definite, and $A_{\phi_n\phi_n}$ is symmetric. The structure of the three matrices $A_{\psi\psi}$, $A_{\phi_n\phi_n}$ and $A_{\phi_n\psi}$ in the Jacobian of equation (4.3) consist of seven diagonals with the exception of $D_{\psi\phi_n}$ being a diagonal. Two approximations to the Jacobian B

$$K_1 = \begin{bmatrix} D_{\psi\psi} & D_{\psi\phi_n} \\ A_{\phi_n\psi} & D_{\phi_n\phi_n} \end{bmatrix} \quad (4.4)$$

and

$$K_2 = \begin{bmatrix} D_{\psi\psi} & D_{\psi\phi_n} \\ D_{\phi_n\psi} & D_{\phi_n\phi_n} \end{bmatrix} \quad (4.5)$$

are used to construct the pre-conditioning matrices. We denote C_{La}^1 as a left pre-conditioning matrix which transforms K_1 into the form of $(\bar{K}_1)_a$ in (2.4). For example, with

$$C_{Rc}^1 = \begin{bmatrix} I & 0 \\ -D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & I \end{bmatrix} \quad (4.6)$$

we have

$$K_1 C_{Rc}^1 = \begin{bmatrix} D_{\psi\psi} - D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & D_{\psi\phi_n} \\ 0 & D_{\phi_n\phi_n} \end{bmatrix} \quad (4.7)$$

and

$$B C_{Rc}^1 = \begin{bmatrix} A_{\psi\psi} - D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & D_{\psi\phi_n} \\ A_{\phi_n\psi} - A_{\phi_n\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & A_{\phi_n\phi_n} \end{bmatrix}. \quad (4.8)$$

So we can obtain 6 different transformation matrices such as C_{La}^1 , C_{Lb}^1 , C_{Lc}^1 , C_{Ra}^1 , C_{Rb}^1 and C_{Rc}^1 from the approximate matrix K_1 , and similarly another six from the approximate matrix K_2 . It should be noted that $C_{Lb}^1 = C_{Lc}^1$ and $C_{Rb}^1 = C_{Rc}^1$. Therefore, there are total of ten different pre-conditioning matrices and the corresponding pre-conditioned matrices (see the Appendix).

We have tested the block G-S iteration convergence of these pre-conditioned matrices evaluated at the first non-linear Newton iteration of the device simulation. These tests were made at several combinations of bias voltages for V_{gs} and V_{ds} . In the block G-S iteration, we employed the ICCG algorithm [9] for the solution of the symmetric matrices and, the ILU-CGS [10], [6] and the ILU-GMRES [11], [6] for the asymmetric matrices. We summarize the numerical results using the following four comparisons.

1) *The Convergence Comparison of ABF, MSP, and Plain G-S:* The values in Table I indicate iteration number k when both following convergence criteria are satisfied:

$$\left| \frac{\Delta\psi^k - \Delta\psi^*}{\Delta\psi^*} \right| < 10^{-3} \quad (4.9)$$

$$\left| \frac{\Delta\Phi_n^k - \Delta\Phi_n^*}{\Delta\Phi_n^*} \right| < 10^{-3} \quad (4.10)$$

where $\Delta\psi^*$ and $\Delta\Phi_n^*$ are the solutions exact enough, i.e., those with error less than 10^{-6} in the convergent block iteration.

Table I shows the convergence of three block G-S iterations. For simplicity of the table, the ABF method [8] is expressed as A , the MSP method [6] as M and, the plain G-S, which does not accept any pre-conditioning, as G . Row voltages are V_{ds} , stepsize is 0.5 V, and column voltages are V_{gs} . Obviously, the convergent region of the ABF and the MSP is much larger than that of plain G-S. As shown in the Table I, all iteration numbers of the ABF and MSP are always limited within ten, while those of the plain G-S are over tens even more one hundred except $V_{gs} = 0.5$ V. So, the convergence rate improvement of the

TABLE I
THE CONVERGENCE COMPARISON OF ABF, MSP AND PLAIN G-S AT THE FIRST NONLINEAR NEWTON ITERATION (A: ABF, M: MSP AND G: PLAIN G-S)

V_{gs}	V_{ds}	0.5	1.0	1.5	2.0	2.5	Method
0.5	2	2	2	2	2	2	A
	2	2	2	2	2	2	M
	3	3	2	3	3	3	G
1.0	5	5	5	6	6	6	A
	5	5	5	6	6	6	M
	22	142	x	x	x	x	G
1.5	5	7	6	6	6	6	A
	4	5	5	6	6	6	M
	41	77	>150	x	x	x	G
2.0	5	6	6	6	6	6	A
	5	5	6	6	6	6	M
	53	>130	>150	x	x	x	G
2.5	4	5	6	7	7	7	A
	4	6	6	6	6	6	M
	63	97	117	>150	>150	>150	G

ABF and MSP can be as large as more than an order of magnitude in comparison with the plain G-S. Another important fact is that the iteration numbers of the ABF and MSP increase slowly when bias voltages grow.

Table II summarizes part of the results from Table I to give a deeper impression about the convergence of the plain G-S iteration. We use three simple symbols $\checkmark$, $*$, and $\times$ to illustrate the different convergence of it in images. The symbol $\checkmark$ indicates that the iteration number is competitive with the ABF and MSP, $*$ represents the iteration whose number exceeds 20, and $\times$ divergent.

Table II shows that plain G-S iteration is very poor in the Newton's steps used for device simulation. It can converge rapidly only for a very small V_{gs} region, and its convergence performance deteriorates quickly as V_{gs} grows. In contrast, the results in Tables I and II show that the ABF and MSP can work well whereas the plain G-S method shows its poor convergence.

2) *The Comparisons of Four Pre-Conditioning Families:* We give the convergence comparisons of four pre-conditioning families under the same conditions stated at (4.9) and (4.10).

It is easy to see from Tables III-VI that all of the pre-conditioning transformations except C_{La}^2 and C_{Lc}^2 improve the convergence of block G-S iteration more or less. Especially, the transformations C_{La}^1 , C_{Lc}^1 (MSP), C_{Ra}^1 , C_{Rb}^1 ($= C_{Rb}^2$) and C_{Ra}^2 (ABF) demonstrate very high convergence speed. If noting that $\tilde{A}_{\psi\psi} = \epsilon_{11}$ of (3.1) and so on, it is clear that the relationships $\|D_{\psi\psi}^{-1}\tilde{A}_{\psi\psi}\|_{\infty} < 1$ and $\|\tilde{A}_{\psi\psi}D_{\psi\psi}^{-1}\|_{\infty} < 1$ are held because of the strict diagonal dominance of $A_{\psi\psi}$. According to (3.5), we observe that one of the two off-diagonal blocks in those five corresponding pre-conditioned matrices (see the Appendix) was reduced by $D_{\psi\psi}^{-1}\tilde{A}_{\psi\psi}$ or $\tilde{A}_{\psi\psi}D_{\psi\psi}^{-1}$ and it improves convergence of the original matrix B .

The effect of C_{Lb}^1 ($= C_{Lb}^2$), C_{Rc}^1 , and C_{Rc}^2 is not so good as the five kinds mentioned earlier. The reason for this is

TABLE II
THE CONVERGENCE OF PLAIN G-S ITERATION ($\checkmark$: COMPETITIVE WITH ABF AND MSP, $*$: ITERATION NO. OVER 20, $\times$: DIVERGENT)

V_{gs}	V_{ds}	0.5	1.0	1.5	2.0	2.5
0.5		$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
1.0		*	*	$\times$	$\times$	$\times$
1.5		*	*	*	$\times$	$\times$
2.0		*	*	*	$\times$	$\times$
2.5		*	*	*	*	*

TABLE III
THE CONVERGENCE COMPARISONS OF PRE-CONDITIONING MATRICES C_{La}^1 , C_{Lb}^1 , AND C_{Lc}^1 . THIS LEFT PRE-CONDITIONING FAMILY IS FORMED BY K_1

	0.5	1.5	2.5	V_{ds}	V_{gs}
C_{La}^1	1	2	2	0.5	
	4	5	7	1.5	
	4	5	6	2.5	
C_{Lb}^1	2	2	2	0.5	
	20	31	54	1.5	
	29	51	55	2.5	
C_{Lc}^1	2	2	2	0.5	
	4	5	6	1.5	
	4	5	6	2.5	

TABLE IV
THE CONVERGENCE COMPARISONS OF PRE-CONDITIONING MATRICES C_{Ra}^1 , C_{Rb}^1 , AND C_{Rc}^1 . THIS RIGHT PRE-CONDITIONING FAMILY IS FORMED BY K_1

	0.5	1.5	2.5	V_{ds}	V_{gs}
C_{Ra}^1	2	2	2	0.5	
	5	6	7	1.5	
	4	6	6	2.5	
C_{Rb}^1	2	2	2	0.5	
	5	6	6	1.5	
	4	6	7	2.5	
C_{Rc}^1	2	2	2	0.5	
	24	48	58	1.5	
	29	64	97	2.5	

TABLE V
THE CONVERGENCE COMPARISONS OF PRE-CONDITIONING MATRICES C_{La}^2 , C_{Lb}^2 , AND C_{Lc}^2 . THIS LEFT PRE-CONDITIONING FAMILY IS FORMED BY K_2

	0.5	1.5	2.5	V_{ds}	V_{gs}
C_{La}^2	1	2	2	0.5	
	>150	>150	$\times$	1.5	
	$\times$	$\times$	$\times$	2.5	
C_{Lb}^2	2	2	2	0.5	
	20	31	54	1.5	
	29	51	55	2.5	
C_{Lc}^2	2	2	2	0.5	
	$\times$	$\times$	$\times$	1.5	
	$\times$	$\times$	$\times$	2.5	

that neither of off-diagonal blocks in corresponding pre-conditioned matrices was reduced significantly because $A_{\phi\phi}$ is just a row zero-sum matrix and not a strictly diagonal dominant one. The limitation of C_{Lb}^1 ($= C_{Lb}^2$), i.e.,

C_5 in (3.8), was discussed in Section III. From the Appendix it is to be found that the situation for C_{Rc}^1 and C_{Rc}^2 is almost the same as $C_{Lb}^1 (= C_{Lb}^2)$.

We observe that C_{La}^2 and C_{Lc}^2 show worse convergence than the plain G-S iteration does even. It should seem that the off-diagonal block $(\tilde{A}_{\Phi n\psi} - D_{\Phi n\psi}^{-1} \tilde{A}_{\psi\psi})$ in the matrices R_{La}^2 and R_{Lc}^2 (shown in the Appendix) pre-conditioned by C_{La}^2 and C_{Lc}^2 plays a bad role in the block iteration. Although the diagonal elements of $(\tilde{A}_{\Phi n\psi} - D_{\Phi n\psi}^{-1} \tilde{A}_{\psi\psi})$ were canceled maybe some off-diagonal entries of it were enlarged greatly.

It should be pointed out that the results for C_{Ra}^2 shown in Table VI came from using the pre-conditioning transformation C_{Ra}^2 in the Appendix exactly, not C_4 in (2.28). We can say that the ABF [8], whose pre-conditioning matrix is given by C_4 , proposed by Bank *et al.* is normal because the diagonal elements of the pre-conditioned coefficient matrix become 1. The computational results for several block equations with Jacobian indicate that the difference between the normal ABF and the transformation C_{Ra}^2 can be ignored within the error. But the total expense of the normal ABF in per block iteration is much more than the C_{Ra}^2 because there are two asymmetrical main diagonal blocks and not one in the pre-conditioned matrix of the normal ABF. The CPU time for solution of an asymmetrical equation often is four times more than the symmetrical one.

3) *The Direction Convergence Comparison:* The damping technique in Newton's steps is used to guide the Newton's iteration to convergence. This means that in solving the linear equations (4.3) occurred in a damping Newton step, it is more important to get the accurate direction of the update vector than its accurate magnitude. We show the direction convergence for comparing the ABF with the MSP in two examples. The direction convergence means

$$(DRT)_{\psi}^k = \cos(\Delta\psi^k, \Delta\psi^*) \quad (4.11)$$

$$(DRT)_{\Phi n}^k = \cos(\Delta\Phi_n^k, \Delta\Phi_n^*) \quad (4.12)$$

where $(\Delta\psi^k, \Delta\psi^*)$ is the angle between $\Delta\psi^k$ and $\Delta\psi^*$, which has the same meaning as (4.9), and similarly for (4.12).

We can see that $(DRT)_{\psi}^k$ of the MSP often is larger than $(DRT)_{\psi}^k$ of the ABF from Tables VII and VIII. In the fact, the feature above came true at most combinations of $V_{\Phi s}$ and $V_{\Phi s}$. We found that the MSP often took less CPU time than the ABF at damping Newton method because of the special importance of $\Delta\psi$ at each Newton's step.

4) *The Effects of K_1 and K_2 :* Tables III-VII indicate that the closer to the original coefficient matrix B the approximate matrix K is the better the pre-conditioning effect is. As the final comparison, we summarize the results of Tables III-VI by using three symbols $\checkmark$, $*$ and $\times$ to illustrate the different effects of both approximate matrices K_1 and K_2 in Table IX. Symbol $\checkmark$ represents the particular pre-conditioning matrix C which greatly improves the convergence of the block G-S iterations, while

TABLE VI
CONVERGENCE COMPARISONS OF PRE-CONDITIONING MATRICES C_{La}^2 , C_{Lb}^2 ,
AND C_{Rc}^2 . THIS RIGHT PRE-CONDITIONING FAMILY IS FORMED BY K_2

	0.5	1.5	2.5	$V_{\Phi s}$ $V_{\Phi s}$
C_{La}^2	2	2	2	0.5
	5	6	6	1.5
	4	6	7	2.5
C_{Lb}^2	2	2	2	0.5
	5	6	6	1.5
	4	6	7	2.5
C_{Rc}^2	3	2	2	0.5
	42	78	115	1.5
	59	72	77	2.5

TABLE VII
DIRECTION CONVERGENCE COMPARISON OF ABF AND MSP WHEN
 $V_{\Phi s} = 1.5$ V, $V_{\Phi s} = 1.5$ V

k	$(DRT)_{\psi}^k$		$(DRT)_{\Phi n}^k$	
1	0.887740	0.990813	0.999710	0.999009
2	0.997389	0.999398	0.999989	0.999994
3	0.999806	0.999973	0.999999	0.999998
4	0.999994	0.999998	1.000000	1.000000
5	1.000000	1.000000	1.000000	1.000000
Method	ABF	MSP	ABF	MSP

TABLE VIII
DIRECTION CONVERGENCE COMPARISON OF ABF AND MSP WHEN
 $V_{\Phi s} = 2.5$ V, $V_{\Phi s} = 2.5$ V

k	$(DRT)_{\psi}^k$		$(DRT)_{\Phi n}^k$	
1	0.854619	0.989042	0.999390	0.999444
2	0.995715	0.998595	0.999996	0.999907
3	0.999661	0.999826	0.999999	0.999997
4	0.999975	0.999986	1.000000	1.000000
5	0.999998	0.999999	1.000000	1.000000
6	1.000000	1.000000	1.000000	1.000000
Method	ABF	MSP	ABF	MSP

TABLE IX
COMPARISON OF THE EFFECTS OF THE APPROXIMATE MATRICES K_1 AND K_2

K_1		K_2	
C_{La}^1	$\checkmark$	C_{La}^2	$\times$
C_{Lb}^1	$*$	C_{Lb}^2	$*$
C_{Lc}^1	$\checkmark$	C_{Lc}^2	$\times$
C_{Ra}^1	$\checkmark$	C_{Ra}^2	$\checkmark$
C_{Rb}^1	$\checkmark$	C_{Rb}^2	$\checkmark$
C_{Rc}^1	$*$	C_{Rc}^2	$*$

* represents that pre-conditioning matrix C which improves the convergence somewhat. The $\times$ indicates that C has actually degraded the convergence.

We can see from Table IX that the overall effect of using K_1 is better than that of using K_2 . K_1 contributes four excellent acceleration transformations, while K_2 gives only two. K_1 is always able to improve the block G-S iteration, but K_2 not. Since K_1 is closer to B than K_2 , it is no surprise that the above fact is observed.

V. CONCLUSIONS

We have presented a new approach, ABE, to construct the pre-conditioning transformations for block equations and shown its numerical results for device simulation. Our primary analysis in Section III can coincide with the computational results. It is known that the effect of the pre-

conditionings is good if the features such as strict diagonal dominance of the main diagonal blocks and simplicity of the off-diagonal blocks can be exploited. Therefore, we should choose one or two of the best transformations based on analysis or calculating. Further work involving the construction of the needed matrices and a complete theoretical analysis is still under development.

APPENDIX

$$\begin{aligned}
 C_{La}^1 &= \begin{bmatrix} I & -D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} \\ -A_{\phi_n\psi} D_{\psi\psi}^{-1} & I \end{bmatrix} \\
 R_{La}^1 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & -D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} \bar{A}_{\phi_n\phi_n} \\ -A_{\phi_n\psi} D_{\psi\psi}^{-1} \bar{A}_{\psi\psi} & A_{\phi_n\phi_n} - A_{\phi_n\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \end{bmatrix} \\
 C_{Lb}^1 &= \begin{bmatrix} I & -D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} \\ 0 & I \end{bmatrix} \\
 R_{Lb}^1 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & -D_{\psi\psi_n} D_{\phi_n\phi_n}^{-1} \bar{A}_{\phi_n\phi_n} \\ A_{\phi_n\psi} & A_{\phi_n\phi_n} \end{bmatrix} \\
 C_{Lc}^1 &= \begin{bmatrix} I & 0 \\ -A_{\phi_n\psi} D_{\psi\psi}^{-1} & I \end{bmatrix} \\
 R_{Lc}^1 &= \begin{bmatrix} A_{\psi\psi} & D_{\psi\phi_n} \\ -A_{\phi_n\psi} D_{\psi\psi}^{-1} \bar{A}_{\psi\psi} & A_{\phi_n\phi_n} - A_{\phi_n\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \end{bmatrix} \\
 C_{Ra}^1 &= \begin{bmatrix} I & -D_{\psi\psi}^{-1} D_{\psi\phi_n} \\ -D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & I \end{bmatrix} \\
 R_{Ra}^1 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & -\bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \\ -\bar{A}_{\phi_n\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & A_{\phi_n\phi_n} - A_{\phi_n\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \end{bmatrix} \\
 C_{Rb}^1 &= \begin{bmatrix} I & -D_{\psi\psi}^{-1} D_{\psi\phi_n} \\ 0 & I \end{bmatrix} \\
 R_{Rb}^1 &= \begin{bmatrix} A_{\psi\psi} & -\bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \\ A_{\phi_n\psi} & A_{\phi_n\phi_n} - A_{\phi_n\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \end{bmatrix} \\
 C_{Rc}^1 &= \begin{bmatrix} I & 0 \\ -D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & I \end{bmatrix} \\
 R_{Rc}^1 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & D_{\psi\phi_n} \\ -\bar{A}_{\phi_n\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & A_{\phi_n\phi_n} \end{bmatrix} \\
 C_{La}^2 &= \begin{bmatrix} I & -D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} \\ -D_{\phi_n\psi} D_{\psi\psi}^{-1} & I \end{bmatrix} \\
 R_{La}^2 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} A_{\phi_n\psi} & -D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} \bar{A}_{\phi_n\phi_n} \\ \bar{A}_{\phi_n\psi} - D_{\phi_n\psi} D_{\psi\psi}^{-1} \bar{A}_{\psi\psi} & A_{\phi_n\phi_n} - D_{\phi_n\psi} D_{\psi\psi}^{-1} D_{\psi\phi_n} \end{bmatrix} \\
 C_{Lb}^2 &= \begin{bmatrix} I & -D_{\psi\phi_n} D_{\phi_n\phi_n}^{-1} \\ 0 & I \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
R_{Lb}^2 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\psi} D_{\psi\psi}^{-1} A_{\psi\psi} & -D_{\psi\psi} D_{\psi\psi}^{-1} \bar{A}_{\psi\psi} \\ A_{\psi\psi} & A_{\psi\psi} \end{bmatrix} \\
C_{Lc}^2 &= \begin{bmatrix} I & 0 \\ -D_{\psi\psi} D_{\psi\psi}^{-1} & I \end{bmatrix} \\
R_{Lc}^2 &= \begin{bmatrix} A_{\psi\psi} & D_{\psi\psi} \\ \bar{A}_{\psi\psi} D_{\psi\psi} D_{\psi\psi}^{-1} \bar{A}_{\psi\psi} & A_{\psi\psi} - D_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} \end{bmatrix} \\
C_{Ra}^2 &= \begin{bmatrix} I & -D_{\psi\psi}^{-1} D_{\psi\psi} \\ -D_{\psi\psi}^{-1} D_{\psi\psi} & I \end{bmatrix} \\
R_{Ra}^2 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} & -\bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} \\ \bar{A}_{\psi\psi} - \bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} & A_{\psi\psi} - A_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} \end{bmatrix} \\
C_{Rb}^2 &= \begin{bmatrix} I & -D_{\psi\psi}^{-1} D_{\psi\psi} \\ 0 & I \end{bmatrix} \\
R_{Rb}^2 &= \begin{bmatrix} A_{\psi\psi} & -\bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} \\ \bar{A}_{\psi\psi} & A_{\psi\psi} - A_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} \end{bmatrix} \\
C_{Rc}^2 &= \begin{bmatrix} I & 0 \\ -D_{\psi\psi}^{-1} D_{\psi\psi} & I \end{bmatrix} \\
R_{Rc}^2 &= \begin{bmatrix} A_{\psi\psi} - D_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} & D_{\psi\psi} \\ \bar{A}_{\psi\psi} - \bar{A}_{\psi\psi} D_{\psi\psi}^{-1} D_{\psi\psi} & A_{\psi\psi} \end{bmatrix}
\end{aligned}$$

where $A_{\psi\psi} \equiv A_{\psi\psi} - D_{\psi\psi} \bar{A}_{\psi\psi} \equiv A_{\psi\psi} - D_{\psi\psi}$, matrices C_{La}^1 and others are defined in Section IV, $R_{La}^1 \equiv C_{La}^1 B$, and similarly for the others.

REFERENCES

- [1] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. Vienna, Austria: Springer-Verlag, 1984.
- [2] R. F. Lucas, K. Wu, and R. W. Dutton, "A parallel 3-D Poisson solver on a hypercube multiprocessor," *IEEE Int. Conf. on Computer-Aided Design*, pp. 442-445, 1987.
- [3] K. Kerkhoven, "On the effectiveness of Gummel's method," *SIAM J. Sci. Stat. Comp.*, vol. 9, no. 1, pp. 48-60, 1988.
- [4] —, "A spectral analysis of the decoupling algorithm for semiconductor simulation," *SIAM J. Numer. Anal.*, vol. 25, no. 6, pp. 1299-1312, 1988.
- [5] C. Ringhofer, "A decoupled iteration method for the basic semiconductor device equations," *NUPAD I*, 1986.
- [6] K. Wu, R. F. Lucas, Z. Wang, and R. W. Dutton, "New approaches in a 3-D one-carrier device solver," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 528-537, 1989.
- [7] C. S. Refferty, M. R. Pinto, and R. W. Dutton, "Iterative methods in semiconductor device simulation," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 2018-2027, 1985.
- [8] R. E. Bank, T. F. Chan, W. M. Coughran, and R. K. Smith, "The alternate-block-factorization procedure for system of partial differential equations," *B.I.T.*, vol. 29, no. 4, pp. 938-954, 1989.
- [9] J. A. Meijerink and H. A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix," *Math. Comput.*, vol. 31, no. 137, pp. 148-162, Jan. 1977.
- [10] R. E. Bank, D. J. Rose, and W. Fichtner, "Numerical methods for semiconductor device simulation," *IEEE Trans. Electron Devices*, vol. ED-30, pp. 1031-1041, Sept. 1983.
- [11] T. Toyabe, H. Masuda, Y. Aoki, H. Shukuri, and T. Hagiwara, "Three-dimensional device simulator CADDETH with highly convergent matrix solution algorithms," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 2038-2043, Oct. 1985.
- [12] C. den Heijer, "Preconditioning iterative methods for nonsymmetric linear systems," in *Proc. Int. Conf. on Simulation of Semiconductor Device and Processes*, Swansea, U.K., June 1984.
- [13] Y. Saad and M. H. Shultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comp.*, vol. 7, pp. 856-869, 1986.
- [14] E. M. Buturla and P. E. Coltrell, "Simulation of semiconductor transport using coupled and uncoupled solution techniques," *Solid-State Electron.*, vol. 23, no. 4, pp. 331-334, 1980.

Ze-Yi Wang received a degree majoring in computational mathematics from Xian-Jiaotong University, Xian, China, in 1965.

Since 1965, he has been with Tsinghua University, Beijing, China, where he is now an Associate Professor in Department of Computer Science and Technology. From 1987 to 1988, he was a visiting scholar at Stanford University working on 3-D device simulation program on a parallel computer. His main research interests are the application and research of numerical methods, including the parallel computations in the areas of VLSI CAD such as circuit analysis, device simulation and circuit extraction.



Ke-Chih Wu (M'87) graduated from Fudan University, Shanghai, China in 1977. He received the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1981 and 1987, respectively.

From 1977 to 1978, he worked as a Research Assistant in the area of CCD technology with the Technology & Physics Research Institute in Shanghai, China. Currently, he is a Research Associate in the Department of Electrical Engineering, Stanford University. His main research interest is in the area of multidimensional numerical analysis of semiconductor devices on parallel computers.

Robert W. Dutton (S'67-M'70-SM'80-F'84), for photograph and biography please see page 920 of the July 1992 issue of this TRANSACTIONS.