

DISSERTATION

Mixed-Mode Device Simulation


ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik

von

KLAUS-TIBOR GRASSER

Matr. Nr. 8826125



Wien, im Mai 1999

.....

To Isabella and Caroline

Kurzfassung

Der rasante Fortschritt in der Entwicklung von Halbleiterbauelementen bringt immer komplexere Bauteilstrukturen hervor. Diese Entwicklung betrifft sowohl die Bauteilgröße, als auch die Verwendung von verschiedenen Materialkombinationen. Die starke Tendenz, die Bauteilgröße mehr und mehr zu reduzieren, erfordert immer komplexere Modelle, um brauchbare Simulationsergebnisse zu erzielen. In der traditionellen Bauteilsimulation wurden nur isolierte Bauteile unter künstlichen Randbedingungen untersucht. Da aber die Schaltung, in der diese Bauteile verwendet werden, einen wesentlichen Einfluß auf deren Verhalten haben kann, haben sich die Erkenntnisse, die durch verkoppelte Simulationen gewonnen werden können, als sehr wertvoll erwiesen. Die Lösung dieses Problems ist sehr komplex und die derzeit erhältlichen Programme können nur in beschränktem Maße dafür eingesetzt werden.

In dieser Arbeit werden verschiedene Kopplungsstrategien zwischen Bauteil- und Schaltungssimulatoren untersucht. Da ein kombinierter Schaltungs- und Bauteilsimulator am erfolgversprechendsten erschien, wurde der Bauteilsimulator MINIMOS-NT um die notwendigen Fähigkeiten zur Schaltungssimulation erweitert.

Das klassische Newton-Verfahren bietet quadratische Konvergenz für eine Anfangslösung, die nahe genug der Lösung ist. Um diesen Einzugsbereich zu vergrößern, ist es notwendig, die Lösung zu dämpfen. Viele verschiedene Dämpfungsstrategien werden derzeit für Schaltungs- und Bauteilsimulation verwendet. Die Nützlichkeit dieser Strategien wurde für das spezielle Problem der verkoppelten Bauteilsimulation untersucht. Eine Methode wird vorgestellt, die in der Lage ist, für mittelgroße Schaltungen eine Lösung innerhalb weniger Iterationen zu finden.

Wegen der fortschreitenden Miniaturisierung der Halbleiterbauelemente gewinnen nicht-lokale Effekte zunehmend an Bedeutung. Sie können in guter Genauigkeit mit einem hydrodynamischen Transportmodell berücksichtigt werden. Leider sind die Konvergenzeigenschaften des hydrodynamischen Transportmodells wesentlich schlechter als die des simpleren Drift-Diffusions-Transportmodells, welches aber wiederum keinerlei Information über nichtlokale Effekte bieten kann. Um den Einfluß dieser nichtlokalen Effekte auf die Simulationsgenauigkeit abschätzen zu können, bieten sich Vergleiche zwischen Drift-Diffusions- und hydrodynamischen Simulation an. Um aussagekräftige Ergebnisse zu erhalten, sind einige Rahmenbedingungen zu erfüllen, welche ausführlich behandelt werden.

Die neuen Fähigkeiten des Simulators wurden anhand typischer analoger und digitaler Schaltungen untersucht. Mit der neuen Einbettungsmethode konnte ein Arbeitspunkt in vielen Fällen ohne Zuhilfenahme einer guten Anfangslösung gefunden werden. Simulationsergebnisse mit Drift-Diffusions- und hydrodynamischen Transportmodellen wurden verglichen. Diese Ergebnisse unterstreichen die Notwendigkeit des hydrodynamischen Transportmodells für moderne Bauelemente.

Als abschließendes Beispiel wurde die thermische Rückkopplung eines kompletten Operationsverstärkers untersucht. Um die thermische Kopplung zwischen den einzelnen Transistoren zu modellieren, mußte die Wärme flu ß gleichung in Zusammenhang mit einer thermischen Ersatzschaltung gelöst werden. Diese thermische Ersatzschaltung dient der ungefähren Modellierung der Temperaturverteilung auf dem Chip. Die Komplexität dieser Simulation geht weit über die Fähigkeiten handelsüblicher Simulatoren hinaus.

Abstract

Recent advances in the development of semiconductor devices lead to more and more complex device structures. This concerns device geometry as well as the combination of different materials. Due to the rapid reduction of device geometries, the models describing the device physics increase in complexity. Traditional device simulation considered the behavior of isolated devices under artificial boundary conditions. To gain additional insight into the performance of devices under realistic dynamic boundary conditions imposed by a circuit, mixed-mode simulation has proven to be invaluable. However, the solution of this problem is very complex and only limited solutions have been available so far.

In this work different coupling strategies of device and circuit simulators are investigated. As the combined device-circuit simulator approach promised the best benefits the device simulator MINIMOS-NT has been extended with mixed-mode capabilities.

The classical Newton method provides quadratic convergence for an initial-guess sufficiently close to the final solution. This region of attraction can be enlarged by providing a damping algorithm for the solution variables. Completely different approaches are in use for circuit and device simulation. The usefulness of these algorithms is investigated for the problem of mixed-mode device simulation. A method is proposed which allows for solving medium sized circuits without a user-specified initial-guess with a small number of necessary iterations.

Due to the ongoing downscaling of semiconductor devices non-local effects become more and more pronounced. They can be modeled with a good accuracy using the hydrodynamic transport model. However, the convergence properties of the hydrodynamic transport model are inferior compared to the simpler drift-diffusion transport model which cannot cover non-local effects. To estimate the influence of these non-local effects comparisons between drift-diffusion and hydrodynamic simulations are necessary. Several conditions must hold for these comparisons to deliver meaningful results, an issue which is discussed in detail.

The new features of the simulator are tested with typical analog and digital circuits. The operating point could be found in many situations starting from the equilibrium without any initial-guess using the new embedding method of circuit elements. Drift-diffusion and hydrodynamic simulation results are compared pointing out the necessity of the hydrodynamic transport model for state-of-the-art devices.

As a final example thermal feedback of a complete operational amplifier is investigated. To model the thermal interaction between the transistors the lattice heat flow equation is solved in conjunction with a thermal network. This thermal network provides a connection of the input and output stage of the circuit thus approximating the temperature distribution along the chip. The complexity of this simulation can be considered well beyond the capabilities of commercially available simulators.

Acknowledgment

I would like to express my deep gratitude to a number of people who supported me during my work on this dissertation which would not have been possible - at least not on this level - without their help and challenge.

Prof. Siegfried Selberherr has not only provided me with one of the best working environments a student can ever dream of, he also knew to ensure the right combination of guidance and freedom. I am also indebted to Prof. Erich Gornik who agreed to take part in the examining committee at a very short notice.

I want to express my appreciation to Hiroo Masuda of the Hitachi Device Development Center in Tokyo, Japan, and to his group not only for the opportunity to learn a lot about semiconductors, but also for cultural engagement and for their friendship.

I also wish to thank my colleagues at the Institute for Microelectronics for their support and cooperation. First of all I am indebted to Dr. Claus Fischer and Dr. Thomas Simlinger for writing the core of MINIMOS-NT. It is due to their insight that my work was made possible. I wish to thank the other members of the MINIMOS-NT-crew Klaus Dragosits, Markus Gritsch, Robert Klima, Dr. Martin Knaipp, Vassil Palankovski, and Martin Rottinger for the excellent working climate and for keeping me busy with discussions and suggestions. Especially my longstanding friend Robert Klima provided the basis for my work while working on the input deck library. I am grateful to Rudolf Strasser, the creator of SIESTA, for revealing numerous bugs in MINIMOS-NT and for principal discussion on software-engineering. Together with Michael Stockinger he acted as imperturbable beta-tester which made MINIMOS-NT the reliable tool it is today. The flexibility of MINIMOS-NT was boosted by the model server which was implemented in an unequalled way by Robert Mlekus. He also supported me by spending endless hours on a project not directly related to his own.

I was lucky enough to work with Prof. Hans Kosina who was always willing to engage in one of our numerous discussions and letting me share his deep insight into semiconductor physics and modeling. The same is true for Dr. Gerhard Schrom who accompanied my 'career' starting from my masters thesis until the final proof read of this thesis. My gratitude is due to Christian Troger and Thomas Binder who solved all those nasty computer problems easily. In Peter Fleischmann I had a perfect roommate and friend who, sharing the same culinary interests, frequently joined me during lunchtime.

Last, but certainly not least I want to thank my family, without whom I would never have been able to achieve so much. I especially wish to express my love for my wife Isabella who not only endured my manifold activities but also provided inspiration and support for my inclination to perfectionism. Only she knows how much I am indebted to her.

Contents

1	Introduction	1
2	Circuit Equations	4
2.1	Nodal Approach	5
2.2	Modified Nodal Approach	8
2.3	Constitutive Relation Formulation	9
2.3.1	Thermal Simulation	10
2.3.2	Time Discretization	11
2.3.3	Devices	11
3	Device Equations	19
3.1	Sets of Partial Differential Equations	20
3.1.1	The Basic Semiconductor Equations	20
3.1.2	The Drift-Diffusion Transport Model	20
3.1.3	The Hydrodynamic Transport Model	20
3.1.4	The Lattice Heat Flow Equation	21
3.1.5	The Constant Quasi-Fermi Potential Approximation	21
3.2	Physical Parameters	22
3.2.1	Mobility Model	22
3.2.2	Energy Relaxation Time	24
3.2.3	Consistent Physical Parameters	24
3.2.4	Velocity Overshoot	31
3.2.5	Low-Field Mobility Reconsidered	31
3.3	Examples	32
3.3.1	Homogeneous Semiconductor	32
3.3.2	Gummel's Pentagon	33
3.3.3	NMOS Transistors	35
3.3.4	PMOS Transistor	36

4	Simulator Coupling	40
4.1	The Two-Level Newton Algorithm	40
4.1.1	Calculation of Conductances	41
4.1.2	Modified Two-Level Newton Algorithm	42
4.2	The Quasi Full-Newton Algorithm	43
4.3	The Full-Newton Algorithm	44
5	Contacts and Boundaries	45
5.1	Quantities, Segments, Boundaries, and Interfaces	45
5.2	Equation Assembly	46
5.3	Interfaces	47
5.3.1	Neumann Type (Explicit Flux)	47
5.3.2	Dirichlet Type (Implicit Flux)	48
5.4	Boundaries	48
5.4.1	Neumann Type	49
5.4.2	Dirichlet Type	49
5.5	Contact Model	49
5.6	Contact Voltage Variable	50
5.7	Example	50
5.8	Mixed-Mode	52
5.9	Thermal Simulation	54
6	Convergence and Damping	58
6.1	Device Equation Damping Schemes	59
6.2	Circuit Equation Damping Schemes	59
6.2.1	Source Stepping	59
6.2.2	Pseudo-Transient Method	59
6.2.3	Global Damping Strategy	60
6.2.4	Local Limiting	60
6.2.5	The New Method	61
6.3	Examples	66
6.3.1	Output Stage of an OpAmp	67
6.3.2	CML Inverter	69
6.3.3	CMOS Inverter	71
7	Simulation Results	74
7.1	Devices	74
7.1.1	MOS Transistors	74

7.1.2	Silicon-Germanium HBT	74
7.2	Five-Stage CMOS Ring Oscillator	82
7.3	Five-Stage CML Ring Oscillator	85
7.4	Thermal Analysis of an Emitter-Coupled Pair	89
7.5	Complete OpAmp μ A709	95
8	Outlook	104
A	Thermal Equivalents	105
B	Device Model Implementation	108
B.1	The Class MmxDevice	108
B.2	Real Device Classes	109
C	Overview of the Input Deck Programming Language	114
D	Iteration Schemes in MINIMOS-NT	117
D.1	Iteration Blocks	117
D.1.1	Equations and Quantities	119
D.1.2	Termination Criterion	125
D.1.3	Failure Criterion	126
D.1.4	Enter Criterion	127
D.1.5	Damping Scheme	127
D.1.6	Full versus Partial System Matrix	127
	Bibliography	128
	Own Publications	133

List of Symbols

Δt	...	a time step
Δx	...	a grid point difference
$\beta_n, \beta_p, \beta_\nu$...	electron, hole, and general exponents in mobility models
δ	...	damping control parameter
ε	...	dielectric constant
ε_r	...	relative dielectric constant
κ_L	...	lattice heat capacity
κ_n, κ_p	...	thermal conductivity of electron and hole gas
μ_ν	...	mobility of carrier type ν
μ_ν^L	...	mobility due to lattice scattering
μ_ν^{LI}	...	mobility due to lattice and impurity scattering
μ_ν^{LIS}	...	mobility due to lattice, impurity, and surface scattering
μ_ν^{LISF}	...	mobility including lattice, impurity, surface scattering, and high-field reduction
μ_ν^{LIST}	...	mobility including lattice, impurity, surface scattering, and high-temperature reduction
μ_n, μ_p	...	electron and hole mobilities
ρ_L	...	lattice mass density
$\tau_{\epsilon,n}, \tau_{\epsilon,p}, \tau_{\epsilon,\nu}$...	energy relaxation times for electrons, holes, and general carrier type
φ	...	node voltage for mixed-mode simulation
φ_C	...	node voltage for mixed-mode simulation applied to a contact
φ_n, φ_p	...	quasi-Fermi potentials for electrons and holes
ψ_C	...	contact potential for mixed-mode simulation
ψ	...	electrostatic potential
ψ_{bi}	...	Built-In potential
C	...	net doping concentration
\mathbf{E}	...	local electric field
E_b	...	local band edge energy for electrons or holes
E_C	...	local band edge energy for electrons
E_g	...	bandgap energy
E_i	...	intrinsic Fermi level
E_V	...	local band edge energy for holes
F_n, F_p	...	driving force for electrons and holes
$I_b^{i,j}$...	electron or hole current between the grid points i and j
$\mathbf{J}_n, \mathbf{J}_p$...	electron and hole current densities
N_C	...	effective density of states for electrons

$N_{C,0}$...	effective density of states for electrons evaluated at reference temperature T_0
N_i	...	concentration of the i-th active dopand
N_T	...	total impurity concentration
N_V	...	effective density of states for holes
$N_{V,0}$...	effective density of states for holes evaluated at reference temperature T_0
R	...	net recombination rate
R^{AU}	...	Auger recombination rate
R^{BB}	...	band to band tunneling recombination rate
R^{II}	...	impact ionization recombination rate
R_i	...	net recombination rate of the i-th box
R^{SRH}	...	SRH net recombination rate
R_T	...	thermal resistance
$\mathbf{S}_n, \mathbf{S}_p$...	electron and hole heat flux density
\mathbf{S}_L	...	lattice heat flux density
T_C	...	contact temperature
T_L	...	local lattice temperature
T_n, T_p	...	electron and hole temperatures
V_T	...	thermal voltage for a certain lattice temperature
V_i	...	the box volume of the i-th box
c_L	...	specific lattice heat capacity
d	...	damping parameter
\mathbf{f}_x	...	vector function of the equation system
h	...	Planck constant
\hbar	...	reduced Planck constant
k_B	...	Boltzmann constant
$m_{n,p}$...	absolute electron and hole mass
$m_{n,p}^*$...	effective electron and hole mass
n	...	electron concentration
p	...	hole concentration
q	...	elementary charge
\mathbf{u}_x	...	update vector of the equation system
$v_{n,300}, v_{p,300}$...	thermal electron and hole velocities at 300 K
\mathbf{x}	...	solution vector of the equation system
\mathbf{x}^*	...	new approach to the solution vector of the equation system
y^{ref}	...	surface reference distance

List of Figures

2.1	Definition of voltages and currents for two-terminal passive elements (a) and sources (b)	5
2.2	Simple circuit containing only conductances and current sources.	5
2.3	Currents and voltages for a device with N -terminals as used by MINIMOS-NT .	10
2.4	Interaction of the coupled electrical and thermal circuits.	10
2.5	Electro-thermal compound model for a heat dissipating resistor	12
2.6	Voltages and currents for controlled sources with the properties $I_1 = I_2$, $I_3 = I_4$, $V_{in} = \varphi_1 - \varphi_2$, and $V_{out} = \varphi_3 - \varphi_4$	16
2.7	Voltages and currents for a gyrator	17
2.8	Internal structure of the power monitor	18
3.1	Mobility vs. electric field in dependence of the basic parameters ξ and β_v . . .	26
3.2	Carrier temperature as a function of electric field for the approach of Hänsch as given by (3.44).	28
3.3	Carrier temperature as a function of electric field for the approach of Baccarani as given by (3.54).	28
3.4	Carrier mobility as a function of carrier temperature for the approach of Hänsch as given by (3.48)-(3.50).	29
3.5	Carrier diffusion coefficient as a function of carrier temperature for the approach of Hänsch.	29
3.6	Carrier energy relaxation time as a function of carrier temperature for the approach of Baccarani.	30
3.7	a) Carrier temperature response for a step in electric field when using the HD model. b) Carrier mobility and velocity response for a step in electric field when using the HD model.	32
3.8	a) Geometry of the homogeneous doped semiconductor. b) Geometry of Gummel's pentagon.	33
3.9	Influence of the boundary condition for the carrier temperatures on the distribution of the electric field inside the homogeneous p-resistor for various bias conditions. The horizontal lines belong to the special contact model (3.68). .	34

3.10	Influence of the boundary condition for the carrier temperatures on the distribution of the hole temperature inside the homogeneous p-resistor for various bias conditions. The horizontal lines belong to the special contact model (3.68).	34
3.11	I-V curves for the homogeneously n- and p-doped resistors for DD and HD simulations. In addition, for the p-doped resistor the current for $\beta_\nu = 2$ is shown.	35
3.12	Doping profiles of a) the long-channel ($L_G = 2.0 \mu\text{m}$) and b) the short-channel ($L_G = 0.2 \mu\text{m}$) NMOS transistor.	36
3.13	Electron concentration before the pinch-off point for the long-channel NMOS for both transport models ($x = 2.12 \mu\text{m}$).	37
3.14	Electron concentration in the pinch-off point for the long-channel NMOS for both transport models ($x = 2.12 \mu\text{m}$).	37
3.15	Comparison of the output characteristics of the long-channel NMOS for both transport models.	38
3.16	Comparison of the output characteristics of the short-channel NMOS for both transport models.	38
3.17	Comparison of the output characteristics of the long-channel PMOS for both transport models.	39
3.18	Comparison of the output characteristics of the short-channel PMOS for both transport models.	39
4.1	a) Original circuit and b) linearized companion model for the two-level Newton algorithm	40
4.2	Non-linear characteristic of the diode and the quantities for the linearized companion model	41
4.3	Comparison of the two different strategies: a) Device simulator as client. b) Device simulator as server	44
5.1	Splitting of interface points: Interface points as given in a) are split into two different points having the same geometrical coordinates b)	47
5.2	Effect of a separate potential variable on the initial-guess of the potential: a) with a separate potential variable the potential stays smooth inside the semiconductor region. b) directly applying the contact potential gives a large discontinuity of the potential.	51
5.3	Contact handling for mixed-mode	53
5.4	Lattice temperature distribution of a HBT with the isothermal contact model for different bias voltages.	54
5.5	Lattice temperature distribution of a HBT for different thermal contact conductances.	55
5.6	Heat generation distribution of a HBT for different thermal conductances.	56
5.7	Lattice temperature distribution of a HBT with the contact resistance model for different bias voltages.	57

6.1	Horizontal and vertical projection of the current solution for a diode.	60
6.2	Problematic constellation when using DBC.	63
6.3	Evolution of the node and contact voltages during iteration using DBC with $V_{DC} = 0$ V. Until convergence 10 iterations are needed.	63
6.4	Evolution of the node and contact voltages during iteration using DBC with $V_{DC} = 10$ V. Until convergence 35 iterations are needed.	64
6.5	Evolution of the node and contact voltages during iteration using RBC with $V_{DC} = 10$ V. As for $V_{DC} = 0$ V, 10 iterations are needed until convergence. . .	64
6.6	Effect of global and local damping on the solution variables: a) global damping b) local damping.	65
6.7	Placement of the iteration dependent conductance G_S^k for one terminal. . . .	66
6.8	Simplified output stage of the $\mu A709$ operational amplifier with $V_{CC} = 15$ V, $V_{EE} = -15$ V, $R_1 = 20$ k Ω , and $R_L = 500$ Ω	67
6.9	Evolution of the node voltages during DC operating point calculation for the OpAmp output stage with $\kappa = 4$	68
6.10	Comparison of the iteration counters for a DC transfer characteristic for the OpAmp output stage with $\kappa = 4$	68
6.11	Evolution of the node voltages during DC operating point calculation for the OpAmp output stage with constant $G_S^k = 1/G_{min}$	69
6.12	CML inverter with $V_{EE} = -5.2$ V. a) with simple current source b) and with current mirror.	70
6.13	Evolution of the node voltages during DC operating point calculation for the CML inverter with $\kappa = 3$	71
6.14	Simulated DC transfer characteristic for the CML inverter.	72
6.15	CMOS inverter with $V_{CC} = 1.2$ V.	72
6.16	Evolution of the node voltages during DC operating point calculation for the CMOS inverter with $\kappa = 1$	73
6.17	Simulated DC transfer characteristic for the CMOS inverter.	73
7.1	Geometry of the HBT.	75
7.2	Net doping profile of the HBT.	75
7.3	Electric field inside the HBT in dependence on V_{CE} for both DD and HD simulation.	77
7.4	Electron velocity inside the HBT in dependence on V_{CE}	78
7.5	Electron temperature inside the HBT in dependence on V_{CE}	78
7.6	Contact temperature of the HBT in dependence on V_{BE}	79
7.7	Collector and base current of the HBT in dependence on V_{BE} . For the self-heating model I_C is approximately independent of G_{th}	79
7.8	Comparison of the current gain of the HBT for self-heating with different G_{th} and $T_L = 300$ K.	80

7.9	Comparison of the current gain of the HBT for self-heating with different contact temperatures and $T_L = 300$ K.	80
7.10	Fictitious transient step response of the HBT when turning self-heating on ($V_{BE} = 1$ V).	81
7.11	Five-stage CMOS ring oscillator	82
7.12	Node voltages of the long-channel five-stage CMOS ring oscillator. DD and HD match perfectly.	84
7.13	Node voltages φ_1 and φ_2 of the short-channel five-stage CMOS ring oscillator for DD and HD.	84
7.14	Five-stage CML ring oscillator	85
7.15	Open-loop DC transfer characteristic for the CML ring oscillator.	87
7.16	Open-loop gain for the CML ring oscillator at the last three stages.	87
7.17	Oscillation of node voltage φ_1 of the five-stage CML ring oscillator. Large discrepancies between DD and HD are observed.	88
7.18	Oscillation of the collector current of T_1 of the five-stage CML ring oscillator. Current levels are approximately the same for both transport models.	88
7.19	Schematic of a differential pair	90
7.20	Thermal equivalent circuit for the differential pair	91
7.21	DC transfer characteristic of the emitter-coupled pair for different thermal models.	92
7.22	Contact temperatures of the transistors T_1 and T_2 against the input voltage.	93
7.23	Contact temperatures of the transistors T_3 and T_4 against the input voltage.	93
7.24	Transient response of the emitter-coupled pair for different thermal models.	94
7.25	a) Simple model illustrating thermal feedback and b) dissipated power P_d	95
7.26	a) Simple layout demonstrating mismatch due to improper placement of the input stage and b) improved placement of the input stage on thermal isolines.	96
7.27	Schematic of the μ A709 OpAmp.	99
7.28	Thermal equivalent circuit used to simulate thermal interaction for the μ A709 OpAmp.	100
7.29	DC transfer characteristic of the μ A709 for constant lattice temperature and considering thermal coupling of the input and output stage.	101
7.30	Open-loop gain of the μ A709 for constant lattice temperature and considering thermal coupling of the input and output stage.	101
7.31	Maximum and contact temperature of the output transistors T_9 and T_{15}	102
7.32	Temperature of the input transistors T_1 and T_2	102
7.33	Temperature difference of the input transistors T_1 and T_2 during the DC transfer characteristic.	103
7.34	Dissipated power P_d in the output stage during the DC transfer characteristic.	103
A.1	a) Geometry and b) temperature distribution of a material block.	105

A.2	a) Grid point i and 4 neighbors used for the discretization of the lattice heat flow equation. b) Electrical analog circuit for the lattice heat flow equation. .	106
A.3	Electrical and thermal modeling of several transistors.	107
B.1	Class hierarchy within the Algorithm Library and definition of the communication interface to the simulator.	109

Chapter 1

Introduction

Over the last decades numerous powerful circuit simulation programs have been developed. Amongst those are general purpose programs which have been designed to cope with all different kinds of circuits and special purpose programs which provide highly optimized algorithms for, e.g., filter design. General purpose programs can be divided into two categories. Programs belonging to the first category provide the user with a general purpose modeling language which can be used to define fairly arbitrary dependencies between the circuit elements. The most prominent member of this category is ASTAP [33] which was developed by IBM in the 70-ties. To provide the user with a maximum of flexibility, ASTAP generates FORTRAN source files which need to be compiled before execution. The other category consists of programs which only allow for a predefined set of circuit elements and dependencies. Although the flexibility is strongly diminished, this approach allows for a much faster execution and a compact, highly optimized simulator kernel. The most prominent member of this category is SPICE which was developed at the University of Berkeley [43].

Circuit simulation programs have in common that the electrical behavior of the devices is modeled by means of a compact model, that is, analytical expressions describing the device behavior. Once a suitable compact model is found, it can be evaluated in a very efficient way. However, this task is far from being trivial and many complicated models have been developed. Even if the behavior of the device under consideration can be mapped onto one of the existing compact models, the parameters of this compact model need to be extracted. In the case of the BSIM2 model [60] for short-channel MOS transistors 300 parameters are available for calibration purposes the identification of which is obviously a cumbersome task. If the device design is known and not modified, these parameters need to be extracted only once and can be used for circuit design if the accuracy of the models is sufficient. When there is need to optimize a device using modified geometries and doping profiles the compact model parameters have to be extracted for each different layout as many of these parameters are mere fit parameters without any physical meaning.

The electrical behavior of the devices can either be measured or simulated. When performing a device optimization, fabricating and measuring each optimization step would be very expensive. Hence, device simulators became more and more popular, e.g., DESSIS [34], MEDICI [69], MINIMOS [58], and PISCES [47]. These device simulators solve the transport equations for a device with given doping profiles and a given geometry. The transport equations form a highly nonlinear partial differential equation system which cannot be solved analytically. Numerical methods have to be used to calculate a solution by discretiz-

ing the equations on a suitable simulation grid. The data obtained from these simulations can be used to extract the parameters of the compact model.

Altogether, this subsequent use of different simulators and extraction tools is cumbersome and error-prone. To overcome these problems several solutions have been published where a device simulator was coupled to SPICE [12, 39, 49]. This is again problematic when considering the communication between two completely different simulators. On the other hand some solutions were presented where circuit simulation capabilities were added to a device simulator [40]. However, the restrictions imposed were so severe that many examples published in this thesis could not be properly dealt with. For example, the number of distributed devices was limited to either one or only a few.

To overcome these shortcomings, the device simulator MINIMOS-NT has been equipped with full circuit simulation capabilities with the only limitation being the amount of computer resources available. MINIMOS-NT is a general purpose device simulator developed as the successor of MINIMOS [55]. Simulations in MINIMOS are restricted to rectangular MOS structures, a too severe limitation for state-of-the-art devices. MINIMOS-NT can cope with arbitrary device structures and geometries. An important prerequisite for accurate simulation of non-planar structures are triangular grids which have been introduced into MINIMOS-NT.

The traditional drift-diffusion equations can be augmented by the lattice heat flow equation to account for self-heating effects. To account for non-local effects a hydrodynamic equation set is available. These equations can be solved for static and transient problems.

As heterostructures are becoming more and more important, proper handling of all kinds of heterostructures has been a major design goal for MINIMOS-NT [15]. This is achieved by splitting the device into logical units, so-called segments which can consist of different materials. Furthermore, these segments form the basic units of model selection. Due to the flexible equation assembly it is, for instance, possible to use a drift-diffusion model on one segment, a hydrodynamic model on another segment, and to account for self-heating on even another segment.

MINIMOS-NT employs a powerful input deck language, enabling the user to customize the simulation in many details. The basic idea is that the input deck is not evaluated once at the beginning of the simulation, but is stored as a database which can be accessed at runtime. Since each keyword in this input deck can be an arbitrary complex and time dependent expression, fine-tuning can be done without the need of any predefined heuristic algorithms, e.g., increasing the number of allowed fill-ins into the sparse system matrix only for one bias point or time steps with large curvature of the input signals.

Different materials are treated in an abstracted way since all material properties are handled via a database which controls the model server. The only information needed in MINIMOS-NT is the so-called material class, e.g., semiconductor or insulator. For each material class a distinct set of models is supplied via this database. The model server provides a C++ like interpreter language [41] allowing the user to add new models in a simple fashion. This implies that any material can be added to the simulator by either providing proper parameters for the existing models or by adding completely new models. These models can be organized in libraries to extend the simulator at runtime whenever needed.

So far many different devices have been simulated with MINIMOS-NT demonstrating its wide applicability. Besides traditional MOSFETS [15, 23, 24], Charge-Coupled-Devices (CCDs) [50], Poly-Emitter-Bipolar transistors [19], Silicon-On-Insulator (SOI) devices [36], High-Electron-Mobility-Transistors (HEMTs) [4, 15, 61, 62], Heterostructure-Bipolar-Transistors (HBTs) [21, 46], and Ultra-Low-Power technologies [52] were investigated.

With the new mixed-mode capabilities at hand devices can be characterized by their performance in a circuit as a function of transport models, doping profiles, mobility models, etc. This is of fundamental importance when investigating the behavior of modern submicron devices and non-mainstream devices like HBTs or HEMTs where compact models are not readily available. Furthermore, when the devices are scaled down, non-local effects become more and more pronounced which can alter the device behavior completely. This cannot be handled by scaling the parameters of compact models.

Chapter 2

Circuit Equations

A physical circuit consists of an interconnection of circuit elements. Two different aspects have to be considered when developing a mathematical model for a circuit.

- The circuit equations must satisfy Kirchhoff's topological laws which are
 - Kirchhoff's current law (KCL) which states that the algebraic sum of the currents leaving a circuit node must be zero at every instant of time and
 - Kirchhoff's voltage law (KVL) which states that the algebraic sum of voltages around any loop of the circuit must be zero at every instant of time.
- Each circuit element has to satisfy its branch relation which will be called a constitutive relation in the following. There are
 - current-defined branches where the branch current is defined in terms of circuit and device parameters and
 - voltage-defined branches where the branch voltage is defined in terms of circuit and device parameters.

Devices with N terminals can be described using $N \cdot (N - 1)/2$ branch relations.

It is not necessary to include all branch currents and voltages into the vector of unknowns \mathbf{x} . On the other hand it is possible to also include charges and fluxes into \mathbf{x} . The wide choice of possible unknown quantities leads to a wide variety of equation formulations that are available. Furthermore, depending on the choice of \mathbf{x} , different phenomena may be described and the complexity of the problem varies drastically. From the vast number of published methods, the following two are the most important:

- The nodal approach only allows for current-defined branches and only the node voltages φ are included into the solution vector \mathbf{x} . After obtaining the solution the currents can be calculated from these node voltages. Voltage-defined branches can be introduced without extending the formulation by the use of gyrators [66, 70]. To properly account for voltage-defined branches the modified nodal approach has been proposed which allows for introduction of arbitrary branch currents [31].
- Several variants of the tableau approach [6] have been published which can be distinguished by the choice of additional unknowns. This is the most general approach and

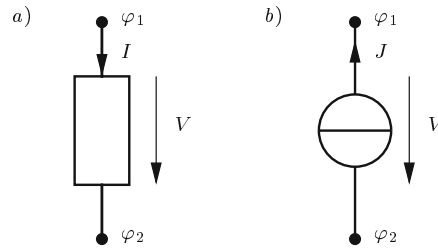


Figure 2.1: Definition of voltages and currents for two-terminal passive elements (a) and sources (b)

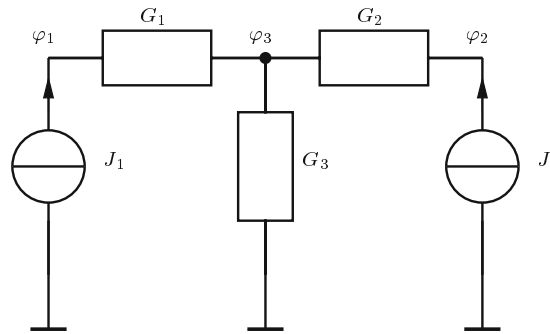


Figure 2.2: Simple circuit containing only conductances and current sources.

a wide variety of circuits can be simulated with it. Especially many idealized theoretical circuit elements which cannot be dealt with properly using nodal approaches can be handled using the tableau approach. However, there are several inherent disadvantages with these algorithms. First, they are more complicated to handle than algorithms belonging to the nodal approach family, and secondly they produce ill-conditioned equation systems which are more cumbersome to treat compared to the well-behaved equation systems resulting from the nodal approach.

As the main task of mixed-mode device simulation is to solve realistic devices under dynamic boundary conditions imposed by a circuit, the nodal approach perfectly suits the needs. Hence the following will concentrate on algorithms related to the nodal approach only.

2.1 Nodal Approach

For the derivation of the nodal approach (NA) a circuit containing linear conductances and independent current sources is assumed. The direction of the current flow is defined in such a way that for both passive elements and sources the current-voltage product is positive, that is $P = V \cdot I > 0$ and $P = V \cdot J > 0$ (see Fig. 2.1).

The independent variables of the NA are the node voltages φ of each circuit node to a ref-

erence node which can be chosen arbitrarily. This choice guarantees that KVL is fulfilled which requires that the sum of all voltages around any loop in the circuit are zero. Considering an arbitrary loop in an arbitrary circuit the sum of the voltages can be written as

$$\sum_j V_j = (\varphi_{j_1} - \varphi_{j_2}) + (\varphi_{j_2} - \varphi_{j_3}) + \dots + (\varphi_{j_{n-1}} - \varphi_{j_1}) = 0 \quad (2.1)$$

Kirchhoff's current law (KCL) is applied to each node other than the reference node in the circuit such that the summation of the currents leaving the node is equal to zero. This procedure is best illustrated on the example circuit shown in Fig. 2.2. Summing the currents at all nodes one gets

$$\text{node 0 :} \quad G_3 \cdot (\varphi_0 - \varphi_3) = -J_1 - J_2 \quad (2.2)$$

$$\text{node 1 :} \quad G_1 \cdot (\varphi_1 - \varphi_3) = J_1 \quad (2.3)$$

$$\text{node 2 :} \quad G_2 \cdot (\varphi_2 - \varphi_3) = J_2 \quad (2.4)$$

$$\text{node 3 :} \quad G_1 \cdot (\varphi_3 - \varphi_1) + G_2 \cdot (\varphi_3 - \varphi_2) + G_3 \cdot (\varphi_3 - \varphi_0) = 0 \quad (2.5)$$

or in matrix and vector notation

$$\underbrace{\begin{bmatrix} G_3 & & & -G_3 \\ & G_1 & & -G_1 \\ & & G_2 & -G_2 \\ -G_3 & -G_1 & -G_2 & G_1 + G_2 + G_3 \end{bmatrix}}_{\mathbb{Y}} \cdot \underbrace{\begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}}_{\boldsymbol{\varphi}} = \underbrace{\begin{bmatrix} -J_1 - J_2 \\ J_1 \\ J_2 \\ 0 \end{bmatrix}}_{\mathbf{J}} \quad (2.6)$$

The matrix \mathbb{Y} is the admittance matrix of the circuit. However, it is obvious that the equations (2.2) - (2.5) are not independent. In general it can be shown that for a circuit containing N nodes, exactly $N - 1$ independent KCL equations can be formulated [72]. As the reference voltage is fixed anyway, there are $N - 1$ equations for the remaining $N - 1$ unknowns. The absolute value of the reference voltage is of course arbitrary and normally assumed to be zero.

Another important issue is that \mathbb{Y} can be assembled by inspection on a per-element basis. The admittance matrices of the devices can simply be superpositioned to yield the circuit admittance matrix. Current sources contribute to the current source vector \mathbf{J} . Therefore these contributions are commonly referred to as *stamps* as they can be directly stamped into the system matrix element per element without the need to consider the rest of the circuit.

The admittance matrix for a conductance G reads

$$\mathbb{Y}_G = \begin{bmatrix} G & -G \\ -G & G \end{bmatrix} \quad (2.7)$$

whereas the contribution of an ideal current source J with the current J_0 to the current source vector is given as

$$\mathbf{J}_J = \begin{bmatrix} J_0 \\ -J_0 \end{bmatrix} \quad (2.8)$$

The stamps normally contain contributions to both \mathbb{Y} and \mathbf{J} , hence a stamp for a current source with shunt conductance reads

$y_{x,y}$	φ_1	φ_2	J
n_1	G	$-G$	J_0
n_2	$-G$	G	$-J_0$

and is simply a superposition of a conductor and a current source.

Non-linear branch relations are handled by introducing linearized companion models. A general non-linear device with branch relation

$$\mathbf{I} = \mathbf{I}(\varphi) \quad (2.9)$$

can be linearized using a Taylor expansion around the last iteration k which gives

$$\mathbf{I}(\varphi^{k+1}) = \underbrace{\frac{\partial \mathbf{I}}{\partial \varphi} \bigg|_{\varphi^k}}_{\mathbb{Y}^k} \cdot \varphi^{k+1} + \underbrace{\left(\mathbf{I}(\varphi^k) - \frac{\partial \mathbf{I}}{\partial \varphi} \bigg|_{\varphi^k} \cdot \varphi^k \right)}_{\mathbf{J}_{eq}^k} \quad (2.10)$$

$$\mathbf{J}_{eq}^k = \mathbf{I}(\varphi^k) - \mathbb{Y}^k \cdot \varphi^k \quad (2.11)$$

The partial derivative in the first term is the linearized admittance matrix \mathbb{Y}^k while the second term \mathbf{J}_{eq}^k is a constant current source which is independent of the new solution φ^{k+1} . Hence, at each iteration each non-linear device can be replaced by linear conductances and current sources resulting in a Newton-like method. Considering the branch relation of a diode

$$\mathbf{I}_D = \begin{bmatrix} I_D \\ -I_D \end{bmatrix} \quad (2.12)$$

with

$$I_D = I_S \cdot \left(\exp \left(\frac{V_D}{V_T} \right) - 1 \right) \quad (2.13)$$

one gets the following stamp

$y_{x,y}$	φ_1	φ_2	J
n_1	g_m^k	$-g_m^k$	J_{eq}^k
n_2	$-g_m^k$	g_m^k	$-J_{eq}^k$

with

$$g_m^k = \frac{I_D^k + I_S}{V_T} \quad (2.14)$$

$$J_{eq}^k = I_D^k - g_m^k \cdot V_D^k \quad (2.15)$$

For circuits containing only conductances and current sources, the condition of the resulting equation system is very good. In this case the NA produces diagonal-dominant matrices which are well suited for iterative solution procedures. Two additional devices can be modeled using the NA, namely a voltage controlled current source (cf. Section 2.3.3.8) and the gyrator (cf. Section 2.3.3.12). These devices destroy the diagonal dominance of the circuit admittance matrix.

One disadvantage of the NA is the inadequate treatment of voltage sources. Ideal voltage sources and current controlled elements cannot be modeled with this approach. However, a very large class of integrated circuits can be accommodated by adding a provision for grounded sources. Other workarounds for floating voltage sources have been proposed which can be quite awkward to handle [53, 65]. To model current dependencies small or negative resistances have been introduced which result in numerical instabilities [65].

2.2 Modified Nodal Approach

The modified nodal approach (MNA) [31] overcomes the above stated shortcomings by introducing branch currents as independent variables which are available to formulate the device constitutive relations. The MNA enjoyed large popularity due to its simplicity and ease of implementation and is employed in SPICE which can safely be labeled the standard simulation program. However, the numerically well behaved circuit matrix obtained by the NA is distorted by those additional equations, and some additional measures have to be taken to allow for the use of iterative solvers.

An ideal voltage source introduces a direct relationship between the node voltages it is connected to of the form

$$\varphi_1 - \varphi_2 = V_0 \quad . \quad (2.16)$$

The current flowing through this source is added as an additional unknown and is used to formulate KCL at the respective nodes. Hence, the following stamp is obtained

$y_{x,y}$	φ_1	φ_2	I	r
n_1			-1	
n_2			1	
I	-1	1		V_0

As pointed out above, these additional equations can produce a zero entry in the main-diagonal, hence an iterative solver cannot be used directly. Zero diagonals can be avoided by exchanging the rows of the admittance matrix \mathbb{Y} following a simple algorithm [43] which guarantees non-zero main-diagonal entries. Failure of this algorithm denotes voltage source loops which indicate an error in the circuit description.

In general, the MNA matrix can be expressed in the form

$$\underbrace{\begin{bmatrix} \mathbb{Y}_R & \mathbb{B} \\ \mathbb{C} & \mathbb{D} \end{bmatrix}}_{\mathbb{Y}} \cdot \underbrace{\begin{bmatrix} \varphi \\ \mathbf{I} \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{J} \\ \mathbf{F} \end{bmatrix}}_{\mathbf{r}} \quad (2.17)$$

where \mathbb{Y}_R is a reduced form of the nodal matrix excluding the contributions due to voltage sources, current controlling elements, etc., \mathbb{B} contains partial derivatives of the Kirchhoff current equations with respect to the additional current variables which are normally ± 1 . The additional branch constitutive relations, derived with respect to the unknown vector are represented by the matrices \mathbb{C} and \mathbb{D} . The vectors \mathbf{J} and \mathbf{F} are the excitation vectors. From an implementations point of view it is advantageous to consider the contributions of each circuit element to the MNA matrix separately.

2.3 Constitutive Relation Formulation

To be compatible with the constitutive relation formulation used for the transport equations ($\mathbf{f}(\mathbf{x}) = 0$) the constitutive relations for the device compact models are formulated in the same way rather than directly calculating linearized circuit elements. MINIMOS-NT solves for $\mathbf{f}(\mathbf{x}) = 0$ using a Newton algorithm:

$$-\mathbb{J}^k \cdot \mathbf{u} = \mathbf{f}(\mathbf{x}^k) \quad (2.18)$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{u} \quad (2.19)$$

$$\mathbb{J}^k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}^k} . \quad (2.20)$$

This formulation has the advantage that the error in the sum of branch currents is available as the residuum of the equations system at each iteration. Furthermore, the implementation of non-linear constitutive relations is simpler than deriving a linearized companion model. The original MNA formulation can be related to the above formulation by

$$\underbrace{-\mathbb{J}^k}_{\mathbb{Y}^k} \cdot \mathbf{x}^{k+1} = \underbrace{\mathbf{f}(\mathbf{x}^k) - \mathbb{J}^k \cdot \mathbf{x}^k}_{\mathbf{r}^k} . \quad (2.21)$$

Equation (2.21) is of fundamental importance as it allows for conversion between models implemented for SPICE into MINIMOS-NT models by general purpose wrapper functions. Only the right hand side needs to be modified to transform the values obtained from SPICE model call into the stamps needed by MINIMOS-NT.

In MINIMOS-NT the constitutive relations for the node voltages are defined to be the sum of all currents flowing *into* this node which must be zero. There is no separation between passive elements and source terms. It is to note that the opposite direction of current flow originates in (2.18) where the negative derivatives are used to assemble the system matrix. This definition is, of course, arbitrary, but defining the current direction this way positive main-diagonal elements are obtained as in the conventional formulation.

The constitutive relation for each node voltage is the sum of the device currents flowing into the node.

$$f_{\varphi_n} = \sum_d f_{\varphi_n}^d = \sum_d I_n^d = 0 \quad (2.22)$$

Branch currents are normally introduced for voltage sources and ammeters. The latter is a voltage source with zero branch voltage which introduces the branch current into the system matrix so that it can be used to control another element. The branch current constitutive relations state relationships between node voltages, e.g., $\varphi_A - \varphi_B = V_0$.

For an N -terminal device with B branch currents, one gets $N + B$ device relations.

$$\mathbf{f}^d = \begin{bmatrix} f_{\varphi_1}^d \\ \vdots \\ f_{\varphi_N}^d \\ f_{I_1}^d \\ \vdots \\ f_{I_B}^d \end{bmatrix} = \begin{bmatrix} I_1^d \\ \vdots \\ I_N^d \\ V_1^d \\ \vdots \\ V_B^d \end{bmatrix} \quad \text{with} \quad \sum_n I_n^d = 0 \quad (2.23)$$

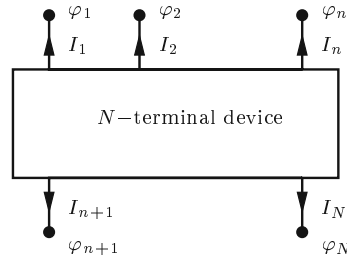


Figure 2.3: Currents and voltages for a device with N -terminals as used by MINIMOS-NT

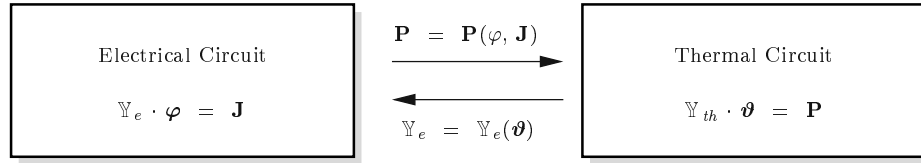


Figure 2.4: Interaction of the coupled electrical and thermal circuits.

The partial contribution of the device d to the Jacobian matrix is given by

$$\mathbb{J}^d = \frac{\partial \mathbf{f}^d}{\partial \mathbf{x}^d} \quad (2.24)$$

where \mathbf{x}^d contains all unknowns which control the device behavior.

2.3.1 Thermal Simulation

The standard way of treating temperature effects in semiconductor devices and circuits is based on the assumption of a constant device temperature which can be obtained by *a priori* assumptions on the dissipated power or by measurements. However, in general this *a priori* assumed dissipated power is not in accordance with the resulting dissipated power. Furthermore, devices may be thermally coupled resulting in completely different temperatures than would be expected from individual self-heating effects alone. This is of special importance as many circuit layouts rely on this effect, e.g., current mirrors and differential pairs [25]. Therefore, the temperature must not be considered a constant parameter, but must be introduced as an additional solution variable.

Thermal coupling can be modeled by a thermal circuit [25, 45]. The topological equations describing a thermal circuit are similar in form to Kirchhoff's equations and the branch relations map to familiar electrical branch relations. The electrical compact models have been extended to provide the device temperature as an external node. For distributed devices MINIMOS-NT solves the lattice heat flow equation to account for self-heating effects. This is of course far more accurate than assuming a spatially constant temperature in the device and estimating the dissipated power by Joule-heat terms alone as is done for the

compact models. To provide a connection to an external thermal circuit arbitrary thermal contacts can be defined.

The thermal heat flow P between two points with temperatures ϑ_1 and ϑ_2 is given by [51]

$$P = G_{th} \cdot (\vartheta_1 - \vartheta_2) \quad (2.25)$$

with G_{th} being the thermal conductance between these two points. In addition, the heat W stored in a device is proportional to the temperature difference

$$W = C_{th} \cdot (\vartheta_1 - \vartheta_2) \quad (2.26)$$

with C_{th} being the thermal capacitance. As the heat flow H is the derivative of W with respect to time one gets

$$H = \frac{\partial W}{\partial t} = C_{th} \cdot \frac{\partial(\vartheta_1 - \vartheta_2)}{\partial t} \quad (2.27)$$

Equation (2.25) is equivalent to Ohm's law with ϑ replacing the node voltage φ and H replacing the branch current I between these two nodes, and (2.27) is equivalent to an electrical capacitor. Hence the same methods as derived above can be used to model thermal interaction between circuit elements. The solution vector \mathbf{x} is augmented by the node temperature vector ϑ and the branch heat flow vector \mathbf{P} .

2.3.2 Time Discretization

Capacitances and inductances are considered in the time domain only and the derivatives are discretized using finite difference methods. The most common difference method is the backward Euler scheme which is an implicit first-order method. For example, for a linear capacitor the time derivative of the current $I = C \cdot dV/dt$ is represented in terms of the backward Euler scheme as

$$I = I(t) = C \cdot \frac{V - V_o}{\Delta t} \quad (2.28)$$

with V_o being the voltage of the last time step.

2.3.3 Devices

The stamps (partial Jacobian matrix entries) for the most important devices as implemented in MINIMOS-NT are summarized in the following.

2.3.3.1 Conductor

The constitutive relations for a linear, temperature dependent conductor are

$$I = V \cdot G(\vartheta) \quad (2.29)$$

$$G(\vartheta) = G_0 \cdot (1 + \alpha \cdot (\vartheta - \vartheta_{ref})) \quad (2.30)$$

$$P = V \cdot I \quad (2.31)$$

If $\alpha \neq 0$, the conductor will be modeled temperature dependently. For thermal simulation the dissipated power P will result in self-heating of the conductor and the problem becomes non-linear. Otherwise the temperature dependent entries are ignored. The stamp is given as

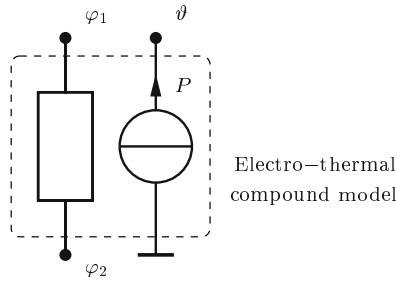


Figure 2.5: Electro-thermal compound model for a heat dissipating resistor

$y_{x,y}$	φ_1	φ_2	ϑ	f
n_1	G	$-G$	$-V \cdot \alpha \cdot G_0$	$-I$
n_2	$-G$	G	$V \cdot \alpha \cdot G_0$	I
ϑ	$-2 \cdot V \cdot G$	$2 \cdot V \cdot G$	$-V^2 \cdot \alpha \cdot G_0$	P

The conductor acts as a heat source connected to the thermal circuit node ϑ and to thermal ground (reference temperature).

2.3.3.2 Resistor

The constitutive relations for a linear temperature dependent resistor are

$$V = I \cdot R(\vartheta) \quad (2.32)$$

$$R(\vartheta) = R_0 \cdot (1 + \alpha \cdot (\vartheta - \vartheta_{ref})) \quad (2.33)$$

$$P = V \cdot I \quad (2.34)$$

The same considerations as for conductors apply for $\alpha \neq 0$. The stamp is given as

$y_{x,y}$	φ_1	φ_2	I	ϑ	f
n_1			1		$-I$
n_2			-1		I
I	-1	1	R	$R_0 \cdot \alpha \cdot I$	$V - I \cdot R$
ϑ	-I	I	-V		P

As the node voltages and the branch current are independent solution variables, it is not guaranteed that the expression $V - I \cdot R$ equals zero while iterating towards the final solution. In the above stamp, the independent solution variable for the current I can be eliminated making I a dependent variable. The resulting stamp reads

$y_{x,y}$	φ_1	φ_2	ϑ	f
n_1	G	$-G$	$V \cdot \alpha \cdot R_0 \cdot G^2$	$-I$
n_2	$-G$	G	$-V \cdot \alpha \cdot R_0 \cdot G^2$	I
ϑ	$-2 \cdot V \cdot G$	$2 \cdot V \cdot G$	$I^2 \cdot \alpha \cdot R_0$	P

with $G = 1/R$. Since I is a dependent variable, I and V may be used interchangeably, as long as $R \neq 0$. The above stamp is, of course, equal to the result obtained by directly considering the conductor $G = 1/R$ which, in this case, depends in a non-linear way on ϑ . However, the current I can only be eliminated if $R \neq 0$ so that the second stamp is somewhat more restrictive. Furthermore, this procedure shows how additional currents can be added or eliminated whenever needed as long as a unique inversion $V = g^{-1}(I)$ of the the branch relation $I = g(V)$ exists which is not the case for $R = 0$. Of course, I must not be used by other device models and hence be a local quantity of the device. In addition, V and I need not necessarily be defined for the same branch as is the case for current-controlled voltage sources.

Another interesting application can be found when adding the branch current for a conductor as an unknown. Neglecting temperature dependencies, the stamp reads

$y_{x,y}$	φ_1	φ_2	I	f
n_1			1	$-I$
n_2			-1	I
I	$-G$	G	1	$V \cdot G - I$

For an open circuit $G = 0$ while $R = 0$ results in a short circuit. By combining the above stamp with the stamp of the ideal resistor an ideal switch can be implemented whose stamp reads

$y_{x,y}$	φ_1	φ_2	I	f
n_1			1	$-I$
n_2			-1	I
I	$-S$	S	$1 - S$	$V \cdot S - I \cdot (1 - S)$

with S denoting the state of the switch. $S = 1$ gives a short circuit while $S = 0$ results in an open circuit.

2.3.3.3 Linear Capacitor

The constitutive relation for a linear capacitor is $I = C \cdot dV/dt$. For the backward Euler discretization scheme the equations are as follows

$$I = G \cdot (V - V_o) \quad (2.35)$$

$$G = \frac{C}{\Delta t} \quad (2.36)$$

The stamp is given as

$y_{x,y}$	φ_1	φ_2	f
n_1	G	$-G$	$-I$
n_2	$-G$	G	I

For thermal capacitors C is substituted by C_{th} and φ by ϑ , respectively.

2.3.3.4 Non-linear Capacitor

The constitutive relation for a non-linear capacitor is $I = dQ/dt$. Using the backward Euler discretization scheme to discretize dQ/dt one obtains

$$I = \frac{Q - Q_o}{\Delta t} \quad (2.37)$$

$$\approx G \cdot V - \frac{C_o \cdot V_o}{\Delta t} \quad (2.38)$$

$$G = \frac{C}{\Delta t} \quad (2.39)$$

Equation (2.38) is a commonly used approximation which, however, does not guarantee charge conservation [66, 67]. For constant C (2.38) of course simplifies to (2.35). The stamp is given as

$y_{x,y}$	φ_1	φ_2	f
n_1	G	$-G$	$-I$
n_2	$-G$	G	I

2.3.3.5 Linear Inductor

The constitutive relation for a linear inductor is $V = L \cdot dI/dt$. For the backward Euler discretization scheme the equations are as follows

$$f_I = V - R \cdot (I - I_o) = 0 \quad (2.40)$$

$$R = \frac{L}{\Delta t} \quad (2.41)$$

and the stamp is given as

$y_{x,y}$	φ_1	φ_2	I	f
n_1			1	$-I$
n_2			-1	I
I	-1	1	R	f_I

It is to note that the current I could be eliminated from the stamp given above as demonstrated for the resistor. However, the above formulation has the advantage that for operating point calculations the inductor can easily be replaced by an ideal voltage source with $R = 0$ and the voltage being the initial condition of the device without a topological change in the circuit.

2.3.3.6 Current Source

The constitutive relation for an ideal current source is given as $I = I_0(t)$. The current can be arbitrarily time-dependent and several common curve shapes have been implemented. However, no dependence on solution variables is allowed as this would result in a voltage or current controlled source (see Section 2.3.3.8 and Section 2.3.3.9). The stamp is given as

$y_{x,y}$	φ_1	φ_2	f
n_1			I
n_2			$-I$

The sign of the current is different as compared to the passive elements as it is defined to flow *out* of the source. Generalizing the branch relation to $I = I_0(t) - V \cdot G$, that is to a current source with shunt resistance, gives the following stamp

$y_{x,y}$	φ_1	φ_2	f
n_1	G	$-G$	I
n_2	$-G$	G	$-I$

which is of course the superposition of an ideal current source with an ideal conductor.

2.3.3.7 Voltage Source

The constitutive relation for an ideal voltage source is given as $V = V_0(t)$. The voltage can be arbitrarily time-dependent and several common curve shapes have been implemented. However, no dependence on solution variables is allowed as this would result in a voltage or current controlled source (see Section 2.3.3.10 and Section 2.3.3.11). The stamp is given as

$y_{x,y}$	φ_1	φ_2	I	f
n_1			-1	I
n_2			1	$-I$
I	1	-1		$V_0 - V$

Again, the sign of the current is different as compared to the passive elements as it is defined to flow *out* of the source. Generalizing the branch relation to $V = V_0(t) - I \cdot R$, that is to a voltage source with series resistance, gives the following stamp

$y_{x,y}$	φ_1	φ_2	I	f
n_1			-1	I
n_2			1	$-I$
I	1	-1	R	$V_0 - V$

Eliminating the current I results in the stamp for the current source with shunt resistance and corresponds to a Norton-Thevenin transformation of the source. For $V_0 = 0$ one gets the stamp of the linear resistor.

2.3.3.8 Voltage Controlled Current Source

The constitutive relation for a linear voltage controlled current source is given as $I_{out} = g \cdot V_{in}$ with g being the transconductance of the device. This is the simplest of all controlled sources and the only one which can be properly modeled using the traditional NA. The terminal voltages and branch currents are shown in Fig. 2.6 and the stamp is given as

$y_{x,y}$	φ_1	φ_2	φ_3	φ_4	f
n_1					
n_2					
n_3	g	$-g$			I_{out}
n_4	$-g$	g			$-I_{out}$

The voltage controlled current source obviously degrades to an ideal conductor if $n_1 = n_3$ and $n_2 = n_4$.

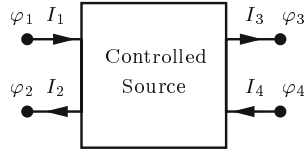


Figure 2.6: Voltages and currents for controlled sources with the properties $I_1 = I_2$, $I_3 = I_4$, $V_{in} = \varphi_1 - \varphi_2$, and $V_{out} = \varphi_3 - \varphi_4$.

2.3.3.9 Current Controlled Current Source

The constitutive relation for a linear current controlled current source is given as $I_{out} = k \cdot I_{in}$ with k being the current gain. To introduce I_{in} as an additional unknown a voltage source with $V_0 = 0$ V is used.

$y_{x,y}$	φ_1	φ_2	φ_3	φ_4	I_{in}	r
n_1					1	$-I_{in}$
n_2					-1	I_{in}
n_3					k	I_{out}
n_4					$-k$	$-I_{out}$
I_{in}	1	-1				$-V_{in}$

2.3.3.10 Voltage Controlled Voltage Source

The constitutive relation for a linear voltage controlled voltage source is given as $V_{out} = k \cdot V_{in}$ with k being the voltage gain.

$y_{x,y}$	φ_1	φ_2	φ_3	φ_4	I_{out}	f
n_1						
n_2						
n_3					-1	I_{out}
n_4					1	$-I_{out}$
I_{out}	k	$-k$	-1	1		$V_{out} - k \cdot V_{in}$

2.3.3.11 Current Controlled Voltage Source

The constitutive relation for a linear current controlled voltage source is given as $V_{out} = r \cdot I_{in}$ with r being the transresistance of the device. This is the most complicated controlled source as it requires two additional branch currents.

$y_{x,y}$	φ_1	φ_2	φ_3	φ_4	I_{in}	I_{out}	f
n_1					1		$-I_{in}$
n_2					-1		I_{in}
n_3						-1	I_{out}
n_4						1	$-I_{out}$
I_{in}	1	-1					$-V_{in}$
I_{out}			-1	1	r		$V_{out} - r \cdot I_{in}$

Again, I_{in} could be eliminated to reduce the rank of the stamp by one.

2.3.3.12 Gyrator

The constitutive relations for a linear gyrator are given as

$$I_{out} = g \cdot V_{in}, \quad (2.42)$$

$$I_{in} = g \cdot V_{out} \quad (2.43)$$

where g is the gyrator constant. The gyrator can be modeled with the classic NA as it is purely voltage controlled. The voltages and currents are shown in Fig. 2.7 and the stamp is given as

$y_{x,y}$	φ_1	φ_2	φ_3	φ_4	r
n_1			g	$-g$	$-I_{in}$
n_2			$-g$	g	I_{in}
n_3	$-g$	g			I_{out}
n_4	g	$-g$			$-I_{out}$

2.3.3.13 Power Monitor

To monitor the power dissipated inside a device, a power monitor has been proposed by Diaz *et al.* [7]. The power monitor implemented in MINIMOS-NT is a four-terminal device with three input nodes, but only one output node as shown in Fig. 2.8. To calculate the dissipated power both the current flowing through the device and the applied voltage are needed. It can be used to model self-heating effects for devices which have not been equipped with proper models as e.g., SPICE models. In this case no derivatives of the device current with respect to the temperature are available. Fortunately, the influence on the convergence properties was found to be negligible in most cases. The stamp looks as follows

$y_{x,y}$	φ_1	φ_2	φ_3	I	ϑ	f
n_2				-1		I
n_3				1		$-I$
I		1	-1			$-V$
ϑ	$-I$	I		$-V$		P

2.3.3.14 Distributed Device

For distributed devices the transport equations have to be solved. Of course it is possible to solve for Poisson's equation alone which is sufficient for the description of dielectric struc-

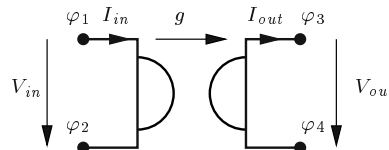


Figure 2.7: Voltages and currents for a gyrator

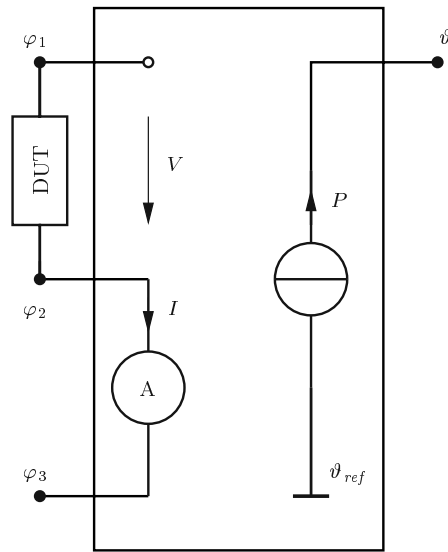


Figure 2.8: Internal structure of the power monitor

tures with complicated geometries. These devices are treated contact-wise in MINIMOS-NT. Each contact is connected to the rest of the circuit via a zero-valued voltage source. The contact current is part of the solution vector and the constitutive relation for the contact potential ψ_C reads

$$f_{\psi_C} = \psi_C - \varphi_C = 0 \quad (2.44)$$

with φ_C being the node voltage of the circuit node connected to the device. The stamp is given as

$y_{x,y}$	φ_C	ψ_C	I_C	f
φ_C			-1	I_C
ψ_C	1	-1		$\psi_C - \varphi_C$
I_C				

A detailed description of the calculation of the contact current and an example can be found in Chapter 5.

Chapter 3

Device Equations

In MINIMOS-NT carrier transport can be treated by the drift-diffusion (DD) and the hydrodynamic (HD) transport models. For either carrier type the transport model can be chosen independently. In addition, the lattice temperature can be treated either as a constant or as an unknown governed by the lattice heat flow equation. These equations will be reviewed in Section 3.1. It is worth mentioning that there is some confusion in the literature about the HD transport model. When deriving the HD model from Boltzmann's transport equation, the average carrier energies w_ν read

$$w_\nu = \frac{3}{2} \cdot k_B \cdot T_\nu + \frac{m^* \cdot v^2}{2} \quad (3.1)$$

with v being the average carrier velocity. For the derivation of (3.1) a momentum displaced Maxwellian has been assumed [3, 26]. When the second term in (3.1) and other related terms in the HD equations are neglected, the energy-transport model is obtained [26]. Depending on these simplifying assumptions, different energy-transport models have been used in device simulators. As it is common practice to refer to the simplified energy-transport model as hydrodynamic transport model, the same nomenclature is used in MINIMOS-NT and in this thesis.

The resulting equation system for the most general case is very complex and time consuming to solve. Simplifications should be made whenever possible, e.g., DD model instead of HD model or to completely neglect carrier transport by assuming a constant quasi-Fermi level for the respective carrier type. However, the validity of these simplifications must be carefully investigated. This is normally done by comparison of simulation results for different equation sets. Despite the obvious fact that depending on the equation set different principal physical effects are taken into account, e.g., self- and carrier-heating, the influence on the models for the physical parameters is more subtle. The main reason for this is that in the case of the HD model, information about the average carrier energy is available in form of the carrier temperature. Many physical parameters depend on this average carrier energy, e.g., the mobilities and the energy relaxation times. In the case of the DD model the carrier temperatures are assumed to be in equilibrium with the lattice temperature, that is $T_C = T_L$, hence, all energy dependent parameters have to be modeled in a different way. The carrier energies are estimated using the local energy balance equations which give expressions for the carrier temperatures as a function of the local electric field. These expressions, however, are only valid under homogeneous conditions. Models for the physical parameters for the DD and HD case will be called consistent when they deliver equivalent results under these homogeneous conditions as will be shown.

3.1 Sets of Partial Differential Equations

3.1.1 The Basic Semiconductor Equations

The basic equations are the Poisson equation and continuity equations for electrons and holes.

$$\operatorname{div}(\varepsilon \cdot \operatorname{grad} \psi) = q \cdot (n - p - C) \quad (3.2)$$

$$\operatorname{div} \mathbf{J}_n = q \cdot \left(R + \frac{\partial n}{\partial t} \right) \quad (3.3)$$

$$\operatorname{div} \mathbf{J}_p = -q \cdot \left(R + \frac{\partial p}{\partial t} \right) \quad (3.4)$$

The unknown quantities of this equation system are the electrostatic potential ψ and the electron and hole concentrations n and p , respectively. C denotes the net concentration of the ionized impurities, ε is the dielectric permittivity of the semiconductor, and R is the net recombination rate which has to be modeled properly.

3.1.2 The Drift-Diffusion Transport Model

The drift-diffusion current relations can, amongst others, be derived from the Boltzmann transport equation by the method of moments or from basic principles of irreversible thermodynamics under the assumption $T_\nu = T_L$. The electron and hole current densities are given by

$$\mathbf{J}_n = q \cdot \mu_n \cdot n \cdot \left(\operatorname{grad} \left(\frac{E_C}{q} - \psi \right) + \frac{k_B}{q} \cdot \frac{N_{C,0}}{n} \cdot \operatorname{grad} \left(\frac{n \cdot T_L}{N_{C,0}} \right) \right), \quad (3.5)$$

$$\mathbf{J}_p = q \cdot \mu_p \cdot p \cdot \left(\operatorname{grad} \left(\frac{E_V}{q} - \psi \right) - \frac{k_B}{q} \cdot \frac{N_{V,0}}{p} \cdot \operatorname{grad} \left(\frac{p \cdot T_L}{N_{V,0}} \right) \right). \quad (3.6)$$

Here, μ_n and μ_p denote the carrier mobilities, and T_L is the lattice temperature. These current relations account for position-dependent band edge energies, E_C and E_V , and position-dependent effective masses, which are included in the effective density of states, $N_{C,0}$ and $N_{V,0}$. The index 0 indicates that $N_{C,0}$ and $N_{V,0}$ are evaluated at some (arbitrary) reference temperature, T_0 , which is constant in real space regardless of what the local values of the lattice and carrier temperatures are.

3.1.3 The Hydrodynamic Transport Model

In the hydrodynamic transport model, carrier temperatures are assumed to be different from the lattice temperature. The basic equations (3.2) through (3.4) are augmented by energy balance equations which determine the carrier temperatures. The current relations take the form

$$\mathbf{J}_n = q \cdot \mu_n \cdot n \cdot \left(\operatorname{grad} \left(\frac{E_C}{q} - \psi \right) + \frac{k_B}{q} \cdot \frac{N_{C,0}}{n} \cdot \operatorname{grad} \left(\frac{n \cdot T_n}{N_{C,0}} \right) \right), \quad (3.7)$$

$$\mathbf{J}_p = q \cdot \mu_p \cdot p \cdot \left(\text{grad} \left(\frac{E_V}{q} - \psi \right) - \frac{k_B}{q} \cdot \frac{N_{V,0}}{p} \cdot \text{grad} \left(\frac{p \cdot T_p}{N_{V,0}} \right) \right). \quad (3.8)$$

The energy balance equations state conservation of the average carrier energies. In terms of the carrier temperatures, T_n and T_p , they can be written as

$$\text{div } \mathbf{S}_n = \text{grad} \left(\frac{E_C}{q} - \psi \right) \cdot \mathbf{J}_n - \frac{3 \cdot k_B}{2} \cdot \left(\frac{\partial (n \cdot T_n)}{\partial t} + R \cdot T_n + n \cdot \frac{T_n - T_L}{\tau_{\epsilon,n}} \right) \quad (3.9)$$

$$\text{div } \mathbf{S}_p = \text{grad} \left(\frac{E_V}{q} - \psi \right) \cdot \mathbf{J}_p - \frac{3 \cdot k_B}{2} \cdot \left(\frac{\partial (p \cdot T_p)}{\partial t} + R \cdot T_p + p \cdot \frac{T_p - T_L}{\tau_{\epsilon,p}} \right). \quad (3.10)$$

Here, $\tau_{\epsilon,n}$ and $\tau_{\epsilon,p}$ denote the energy relaxation times, while \mathbf{S}_n and \mathbf{S}_p are the energy fluxes.

$$\mathbf{S}_n = -\kappa_n \cdot \text{grad } T_n - \frac{5}{2} \cdot \frac{k_B \cdot T_n}{q} \cdot \mathbf{J}_n \quad (3.11)$$

$$\mathbf{S}_p = -\kappa_p \cdot \text{grad } T_p + \frac{5}{2} \cdot \frac{k_B \cdot T_p}{q} \cdot \mathbf{J}_p \quad (3.12)$$

The thermal conductivities, κ_n and κ_p , are assumed to obey a generalized Wiedemann-Franz law [54].

$$\kappa_n = \left(\frac{5}{2} + c_n \right) \cdot \frac{k_B^2}{q} \cdot T_n \cdot \mu_n \cdot n \quad (3.13)$$

$$\kappa_p = \left(\frac{5}{2} + c_p \right) \cdot \frac{k_B^2}{q} \cdot T_p \cdot \mu_p \cdot p \quad (3.14)$$

3.1.4 The Lattice Heat Flow Equation

To account for self-heating effects in semiconductor devices, the lattice heat flow equation has to be solved which reads

$$\text{div } \mathbf{S}_L = H - \rho_L \cdot c_L \cdot \frac{\partial T_L}{\partial t} \quad (3.15)$$

$$\mathbf{S}_L = -\kappa_L \cdot \text{grad } T_L \quad (3.16)$$

with \mathbf{S}_L being the lattice heat flow density. The coefficients of this equation are ρ_L , c_L , and κ_L , which denote the materials mass density, specific heat, and thermal conductivity, respectively. H is the generated local heat density and is modeled in dependence of the transport model. In the drift-diffusion case H equals the Joule heat,

$$H = \text{grad} \left(\frac{E_C}{q} - \psi \right) \cdot \mathbf{J}_n + \text{grad} \left(\frac{E_V}{q} - \psi \right) \cdot \mathbf{J}_p, \quad (3.17)$$

whereas in the hydrodynamic case the relaxation terms are used [36]

$$H = \frac{3 \cdot k_B}{2} \cdot \left(n \cdot \frac{T_n - T_L}{\tau_{\epsilon,n}} + p \cdot \frac{T_p - T_L}{\tau_{\epsilon,p}} \right). \quad (3.18)$$

3.1.5 The Constant Quasi-Fermi Potential Approximation

In unipolar devices like MOSFETs it is often possible to assume a constant quasi-Fermi potential for one carrier type. This implies that the current density for this carrier type vanishes.

The quasi-Fermi potential is used to calculate the local concentration of the considered carrier type. Each segment has a quasi-Fermi potential of its own which is determined by some contact. The advantage is that no continuity equation needs to be solved for this carrier system which reduces the simulation time. In case of a drift-diffusion equation set, the equation system size is reduced by about 1/3. In case of a hydrodynamic simulation the reduction of the equation system size is about 2/5 since the corresponding continuity equations and the energy flux equation are not solved, such that a five equation system reduces to a three equation system. This reference potential is then used to calculate the carrier concentrations in the respective segments. An example is an n-channel MOSFET, for which the holes in the substrate represent the minority carrier system. Since they contribute only marginally to the drain current, it is in many situations justified to ignore the hole current at all by assuming a constant quasi-Fermi level. The equations for the electron and hole concentration read

$$n = N_C \cdot \exp \left(-\frac{q \cdot (\varphi_n - \psi) + E_C}{k_B \cdot T_L} \right) \quad (3.19)$$

$$p = N_V \cdot \exp \left(\frac{q \cdot (\varphi_p - \psi) + E_V}{k_B \cdot T_L} \right) . \quad (3.20)$$

3.2 Physical Parameters

The basic semiconductor equations as given above determine the structure of the equation system. These equations need to be complemented by a set of physical parameters which depend on the materials of the device. The most important parameters are the generation/recombination rates, the carrier mobilities, and the energy relaxation times needed for HD simulations. Different mobility models are used for DD and HD simulations. These will be reviewed in Section 3.2.1 without consistency considerations.

However, especially when trying to compare DD to HD simulation results it is of utmost importance to use a consistent parameter set, in the sense that the HD model must deliver equivalent results to the DD model under homogeneous conditions. This issue will be subject to Section 3.2.3.

3.2.1 Mobility Model

The mobility model used has proven its usefulness during the last decades [57]. The temperature dependence of the lattice mobility in Si is modeled by a power law.

$$\mu_\nu^L = \mu_{\nu,300}^L \cdot \left(\frac{T_L}{300 \text{ K}} \right)^{\gamma_{0,\nu}}, \quad \nu = n, p \quad (3.21)$$

To account for mobility reduction due to ionized impurity scattering, the formula of Caughey and Thomas [5] is used in conjunction with temperature dependent coefficients. C_I denotes

the concentration of ionized impurities.

$$\mu_{\nu}^{\text{LI}} = \mu_{\nu}^{\text{min}} + \frac{\mu_{\nu}^{\text{L}} - \mu_{\nu}^{\text{min}}}{1 + \left(\frac{C_{\text{I}}}{C_{\nu}^{\text{ref}}}\right)^{\alpha_{\nu}}} \quad (3.22)$$

$$\mu_{\nu}^{\text{min}} = \begin{cases} \mu_{\nu,300}^{\text{min}} \cdot \left(\frac{T_L}{300 \text{ K}}\right)^{\gamma_{1,\nu}} & T \geq T_{\text{switch}} \\ \mu_{\nu,300}^{\text{min}} \cdot \left(\frac{2}{3}\right)^{\gamma_{1,\nu}} \cdot \left(\frac{T_L}{200 \text{ K}}\right)^{\gamma_{2,\nu}} & T < T_{\text{switch}} \end{cases} \quad (3.23)$$

$$C_{\nu}^{\text{ref}} = C_{\nu,300}^{\text{ref}} \cdot \left(\frac{T_L}{300 \text{ K}}\right)^{\gamma_{3,\nu}} \quad (3.24)$$

$$\alpha_{\nu} = \alpha_{\nu,300} \cdot \left(\frac{T_L}{300 \text{ K}}\right)^{\gamma_{4,\nu}} \quad (3.25)$$

Surface scattering is modeled by the following empirical expression [57]

$$\mu_{\nu}^{\text{LIS}} = \frac{\mu_{\nu}^{\text{ref}} + (\mu_{\nu}^{\text{LI}} - \mu_{\nu}^{\text{ref}}) \cdot (1 - F(y))}{1 + F(y) \cdot \left(\frac{S_{\nu}}{S_{\nu}^{\text{ref}}}\right)^{\gamma_{6,\nu}}} \quad (3.26)$$

$$\mu_{\nu}^{\text{ref}} = \mu_{\nu,300}^{\text{ref}} \cdot \left(\frac{T_L}{300 \text{ K}}\right)^{-\gamma_{5,\nu}} \quad (3.27)$$

$$F(y) = \frac{2 \cdot \exp\left(-\left(\frac{y}{y^{\text{ref}}}\right)^2\right)}{1 + \exp\left(-2 \cdot \left(\frac{y}{y^{\text{ref}}}\right)^2\right)} \quad (3.28)$$

The pressing forces S_n and S_p in (3.26) are equal to the magnitude of the normal field strength at the interface if the carriers are attracted by the interface, otherwise zero.

For the DD transport model mobility reduction due to a high-field is modeled by

$$\mu_{\nu}^{\text{LISF}} = \frac{2 \cdot \mu_{\nu}^{\text{LIS}}}{1 + \left(1 + \left(\frac{2 \cdot \mu_{\nu}^{\text{LIS}} \cdot F_{\nu}}{v_{\nu}^{\text{sat}}}\right)^{\beta_{\nu}}\right)^{1/\beta_{\nu}}} \quad (3.29)$$

$$F_{\nu} = \left| \text{grad } \psi + \frac{s_{\nu}}{\nu} \cdot \frac{k_B}{q} \cdot \text{grad } (T_L \cdot \nu) \right| \quad (3.30)$$

with F_{ν} representing the driving forces for carrier ν .

For the HD transport model deviation from the ohmic low-field mobility is modeled as a function of the carrier temperature, T_{ν} after Hänsch [30].

$$\mu_{\nu}^{\text{LIST}} = \frac{\mu_{\nu}^{\text{LIS}}}{1 + \alpha_{\nu} \cdot \Delta T_{\nu}} \quad (3.31)$$

$$\Delta T_{\nu} = T_{\nu} - T_L \quad (3.32)$$

$$\alpha_{\nu} = \frac{3 \cdot k_B \cdot \mu_{\nu}^{\text{LIS}}}{2 \cdot q \cdot \tau_{\epsilon,\nu} \cdot (v_{\nu}^{\text{sat}})^2} \quad (3.33)$$

where $\tau_{\epsilon,\nu}$ denotes the energy relaxation time.

3.2.2 Energy Relaxation Time

Two different models for the energy relaxation time $\tau_{\epsilon,\nu}$ are provided. The first model simply uses a constant, carrier temperature independent $\tau_{\epsilon,\nu}$. For electrons a second model is available which has been derived by curve-fitting to Monte-Carlo simulation results and reads

$$\tau_{\epsilon,n} = \tau_{\epsilon,0} + \tau_{\epsilon,1} \cdot \exp \left(C_1 \cdot \left(\frac{T_n}{300 \text{ K}} + C_0 \right)^2 + C_2 \cdot \left(\frac{T_n}{300 \text{ K}} + C_0 \right) + C_3 \cdot \left(\frac{T_L}{300 \text{ K}} \right) \right) \quad (3.34)$$

3.2.3 Consistent Physical Parameters

The high-field mobility formulas (3.29) and (3.31) need some further investigation to fully understand their impact on device modeling. In general the effective mobilities are defined by

$$\mu_\nu = q \frac{\tau_{m,\nu}}{m_\nu^*} \quad (3.35)$$

with $\tau_{m,\nu}$ being the momentum relaxation time and m_ν^* being the effective carrier mass of the respective carrier. By evaluating the moments of the distribution functions [29] the momentum relaxation times $\tau_{m,\nu}$ can be approximated as functions of the average carrier energies $w_\nu = 3k_B T_\nu / 2$ to give (3.31).

For DD $T_\nu = T_L$ is assumed which obviously cannot be used to calculate the carrier energy and its influence on the mobilities. However, the carrier temperatures are coupled to the electric field via (3.9) and (3.10). Assuming a homogeneously doped semiconductor, $\text{div } \mathbf{S}_\nu = 0$, and the energy balance equations (3.9) and (3.10) degenerate to the so-called local energy balance equations which read

$$E^2 \cdot \mu_\nu(T_\nu) = \frac{3 \cdot k_B \cdot \Delta T_\nu}{2 \cdot q \cdot \tau_{\epsilon,\nu}} \quad (3.36)$$

Under the assumption of field- and carrier temperature-independent $\tau_{\epsilon,\nu}$ equation (3.36) in combination with (3.31) gives a direct, local dependence of the carrier temperatures and the electric field

$$\Delta T_\nu(E) = \frac{1}{2 \cdot \alpha_\nu} \cdot \left(\sqrt{1 + \left(\frac{2 \cdot \mu_\nu^{\text{LIS}} \cdot E}{v_\nu^{\text{sat}}} \right)^2} - 1 \right) \quad (3.37)$$

which can be substituted into (3.31) to find

$$\mu_\nu^{\text{LISF}}(E) = \frac{2 \cdot \mu_\nu^{\text{LIS}}}{1 + \sqrt{1 + \left(\frac{2 \cdot \mu_\nu^{\text{LIS}} \cdot E}{v_\nu^{\text{sat}}} \right)^2}} \quad (3.38)$$

which is similar to (3.29) with $\beta_\nu = 2$. However, in (3.29) the driving force

$$F_\nu = \left| \text{grad } \psi + \frac{s_\nu}{\nu} \cdot \frac{k_B}{q} \cdot \text{grad } (T_L \cdot \nu) \right| \quad (3.39)$$

is used instead of the electrical field E . This is done for the following reasons. (3.38) has been derived under the assumption of a homogeneous semiconductor. In this case (3.39) reduces to $F_\nu = |\text{grad } \psi| = E$ which is consistent with the assumptions given above. On the other hand, (3.38) would give no velocity saturation for large carrier gradients and low electric fields. This is obviously unphysical as the carrier velocity for the resulting diffusion current is limited by the thermal velocity. Equation (3.38) is very similar to expressions found on a purely empirical basis. The most common ones are

$$\mu_\nu^{\text{LISF}}(F_\nu) = \frac{\mu_\nu^{\text{LIS}}}{\left(1 + \left(\frac{\mu_\nu^{\text{LIS}} \cdot F_\nu}{v_\nu^{\text{sat}}}\right)^{\beta_\nu}\right)^{1/\beta_\nu}} \quad (3.40)$$

as given by Caughey and Thomas [5] and

$$\mu_\nu^{\text{LISF}}(F_\nu) = \frac{2 \cdot \mu_\nu^{\text{LIS}}}{1 + \left(1 + \left(\frac{2 \cdot \mu_\nu^{\text{LIS}} \cdot F_\nu}{v_\nu^{\text{sat}}}\right)^{\beta_\nu}\right)^{1/\beta_\nu}} \quad (3.41)$$

as given by Reiser [48]. Both (3.40) and (3.41) are claimed to perfectly fit measured data with an appropriate β_ν , normally about $\beta_n = 2$ for electrons and $\beta_p = 1$ for holes. For $\beta_\nu = 1$ both models are equivalent.

One problem becomes obvious when comparing (3.40) and (3.41) with (3.31). Under homogeneous conditions different mobilities and hence currents will be obtained for DD and HD simulations except when using (3.41) with $\beta_\nu = 2$. This is of fundamental importance when comparing DD with HD simulations but unfortunately this problem is generally overlooked in available device simulators.

In general the problem can be stated as follows. In the homogeneous situation the electric field and the carrier temperatures are related by the local energy balance equation. The two parameters, μ and τ_ϵ , have to be modeled properly to guarantee consistency between the DD and the HD model. In principle one could start with either $\mu_\nu^{\text{LIST}}(T_\nu)$ or $\mu_\nu^{\text{LISF}}(E)$ to derive the other appropriate model. To derive $\mu_\nu^{\text{LIST}}(T_\nu)$ from the scattering term of the Boltzmann equation several assumptions on the distribution function are necessary to get a closed form solution like (3.31). From a practical point of view it might seem simpler to start with $\mu_\nu^{\text{LISF}}(E)$ as measured data are more easily available. Nevertheless, as long as closed form solutions exist, the order is of course irrelevant.

The following discussion will assume $\mu_\nu^{\text{LISF}}(E)$ of the form

$$\mu_\nu^{\text{LISF}}(E) = \frac{\mu_\nu^{\text{LIS}}}{\xi + \left(\zeta^{\beta_\nu} + \left(\frac{\mu_\nu^{\text{LIS}} \cdot E}{v_\nu^{\text{sat}}}\right)^{\beta_\nu}\right)^{1/\beta_\nu}} \quad (3.42)$$

With $\xi = 0$ and $\zeta = 1$ one obtains (3.40), and with $\xi = 1/2$ and $\zeta = 1/2$ (3.41). As these values give the most common expressions, only ξ will be used to distinguish between them and an explicit dependence

$$\zeta = 1 - \xi \quad (3.43)$$

will be assumed. It is to note that (3.43) is only used to simplify reference to the more familiar formulas (3.40) with $\xi = 0$ and (3.41) with $\xi = 1/2$.

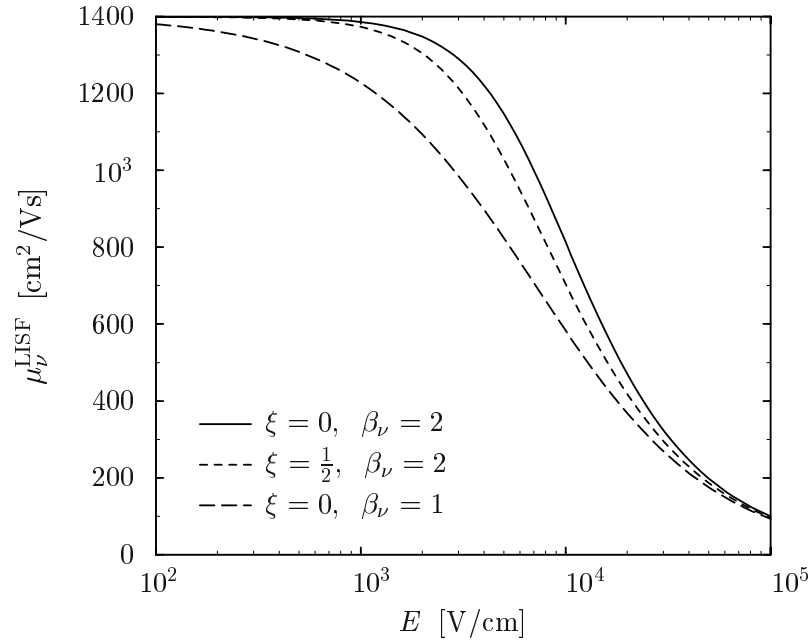


Figure 3.1: Mobility vs. electric field in dependence of the basic parameters ξ and β_ν .

For these values and $\beta_\nu = 1$ and $\beta_\nu = 2$ the mobility vs. field dependence is shown in Fig. 3.1. In the approach given above the only assumption made about $\tau_{\epsilon,\nu}$ was to be independent of the carrier energy. This approximation is valid for large carrier energies [16].

In the following the approach outlined above will be generalized starting from (3.42). Substituting (3.42) in (3.36) results in

$$\frac{E^2 \cdot \mu_\nu^{\text{LIS}}}{\xi + \left(\zeta^{\beta_\nu} + \left(\frac{\mu_\nu^{\text{LIS}} \cdot E}{v_\nu^{\text{sat}}} \right)^{\beta_\nu} \right)^{1/\beta_\nu}} = \frac{3 \cdot k_B \cdot \Delta T_\nu}{2 \cdot q \cdot \tau_{\epsilon,\nu}} \quad (3.44)$$

which must be solved for $E(T_\nu)$, which is then inserted into (3.42). The dependence of T_ν vs. E is shown in Fig. 3.2. Unfortunately (3.44) cannot be explicitly solved in general. However, it can be solved for the most important cases $\xi = 0$ with arbitrary β_ν and for $\xi = 1/2$ with $\beta_\nu = 1$ or $\beta_\nu = 2$. After some algebra one obtains

$$\xi = 0 : E(T_\nu) = \frac{v_\nu^{\text{sat}}}{2^{1/\beta_\nu} \cdot \mu_\nu^{\text{LIS}}} \cdot \left(a_\nu^{\beta_\nu} + \sqrt{a_\nu^{\beta_\nu} \cdot (4 + a_\nu^{\beta_\nu})} \right)^{1/\beta_\nu} \quad (3.45)$$

$$\xi = \frac{1}{2}, \beta_\nu = 1 : E(T_\nu) = \frac{v_\nu^{\text{sat}}}{2 \cdot \mu_\nu^{\text{LIS}}} \cdot \left(a_\nu + \sqrt{a_\nu \cdot (4 + a_\nu)} \right) \quad (3.46)$$

$$\xi = \frac{1}{2}, \beta_\nu = 2 : E(T_\nu) = \frac{v_\nu^{\text{sat}}}{\mu_\nu^{\text{LIS}}} \sqrt{a_\nu \cdot (1 + a_\nu)} \quad \text{with} \quad a_\nu = \alpha_\nu \cdot \Delta T_\nu \quad (3.47)$$

Of course (3.45) with $\beta_\nu = 1$ is identical to (3.46) as are the expressions resulting from

(3.42) with the same parameters. Inserting (3.45), (3.46), and (3.47) into (3.42) yields

$$\xi = 0 \quad : \quad \mu_{\nu}^{\text{LIST}}(T_{\nu}) = \frac{2^{1/\beta_{\nu}} \cdot \mu_{\nu}^{\text{LIS}}}{\left(2 + a_{\nu}^{\beta_{\nu}} + \sqrt{a_{\nu}^{\beta_{\nu}} \cdot (4 + a_{\nu}^{\beta_{\nu}})}\right)^{1/\beta_{\nu}}} \quad (3.48)$$

$$\xi = \frac{1}{2}, \beta_{\nu} = 1 \quad : \quad \mu_{\nu}^{\text{LIST}}(T_{\nu}) = \frac{2 \cdot \mu_{\nu}^{\text{LIS}}}{2 + a_{\nu} + \sqrt{a_{\nu} \cdot (4 + a_{\nu})}} \quad (3.49)$$

$$\xi = \frac{1}{2}, \beta_{\nu} = 2 \quad : \quad \mu_{\nu}^{\text{LIST}}(T_{\nu}) = \frac{\mu_{\nu}^{\text{LIS}}}{1 + a_{\nu}} \quad (3.50)$$

A comparison of these mobilities is given in Fig. 3.4. The diffusivity is defined by the generalized Einstein relation

$$D_{\nu}(T_{\nu}) = \frac{k_B \cdot T_{\nu}}{q} \cdot \mu_{\nu}^{\text{LIST}}(T_{\nu}) \quad (3.51)$$

and is shown in Fig. 3.5. In Fig. 3.5 $\alpha = 1/T_L$ has been assumed which gives a $1/T_{\nu}$ dependence of the mobility (3.50) and hence a constant diffusivity. This choice will be justified later.

A different approach has been proposed by Baccarani and Wordeman [2]. From the generalized Einstein relation it follows that the carrier temperature can be directly related to the applied field as

$$T_{\nu} = \frac{q}{k_B} \cdot \frac{D_{\nu}(E)}{\mu_{\nu}^{\text{LISF}}(E)} \quad (3.52)$$

They assumed a constant diffusivity

$$D_{\nu}(E) = D_{\nu 0} = \frac{k_B \cdot T_L}{q} \cdot \mu^{\text{LIS}} \quad (3.53)$$

which they justified with experimental data and MC simulation results. However, (3.53) overestimates $D_{\nu}(E)$ for large fields [2]. Using (3.52) and (3.53) the carrier temperature may be expressed as

$$T_{\nu} = T_L \cdot \frac{\mu^{\text{LIS}}}{\mu_{\nu}^{\text{LISF}}(E)} = T_L \cdot \frac{\mu^{\text{LIS}}}{\mu_{\nu}^{\text{LIST}}(T_{\nu})} \quad (3.54)$$

This dependence is shown in Fig. 3.3. With (3.42) and (3.54) $E(T_{\nu})$ reads

$$E(T_{\nu}) = \frac{v_{\nu}^{\text{sat}}}{\mu_{\nu}^{\text{LIS}}} \cdot \left(\left(\frac{T_{\nu}}{T_L} - \xi \right)^{\beta_{\nu}} - \zeta^{\beta_{\nu}} \right)^{1/\beta_{\nu}} \quad (3.55)$$

From the local energy balance equation the energy relaxation times $\tau_{\epsilon, \nu}$ may be expressed as

$$\tau_{\epsilon, \nu}(T_{\nu}) = \frac{3 \cdot k_B \cdot \Delta T_{\nu}}{2 \cdot q \cdot E^2 \cdot \mu_{\nu}^{\text{LIST}}(T_{\nu})} = \frac{3 \cdot k_B \cdot \mu^{\text{LIS}}}{2 \cdot q \cdot v_{\nu}^{\text{sat}^2}} \cdot \frac{T_{\nu} \cdot \Delta T_{\nu}}{T_L} \cdot \left(\left(\frac{T_{\nu}}{T_L} - \xi \right)^{\beta_{\nu}} - \zeta^{\beta_{\nu}} \right)^{-2/\beta_{\nu}} \quad (3.56)$$

An interesting special case is $\xi = 1/2$ and $\beta_{\nu} = 2$ in which (3.42) simplifies to (3.38). As (3.38) has been derived under the assumption of an energy independent $\tau_{\epsilon, \nu}$ one would

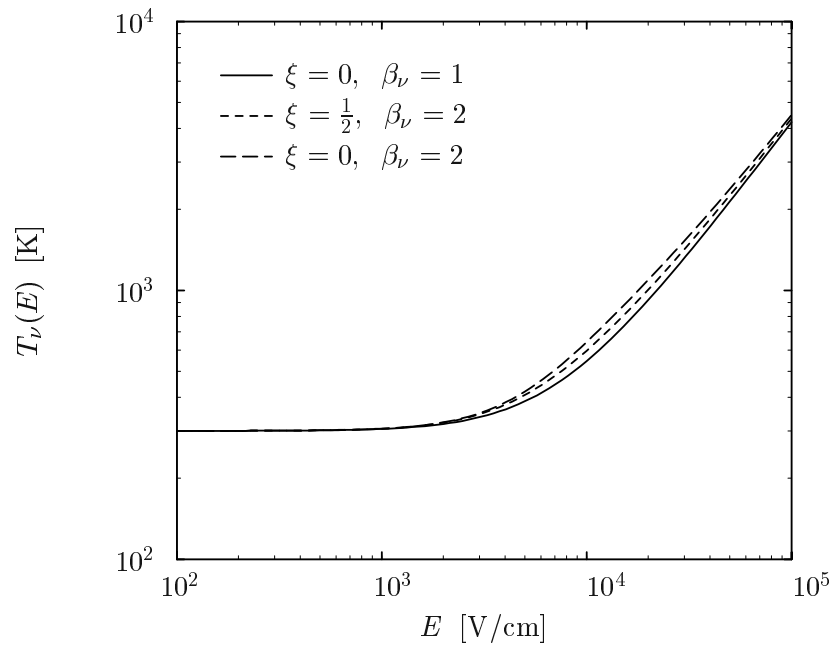


Figure 3.2: Carrier temperature as a function of electric field for the approach of Hänsch as given by (3.44).

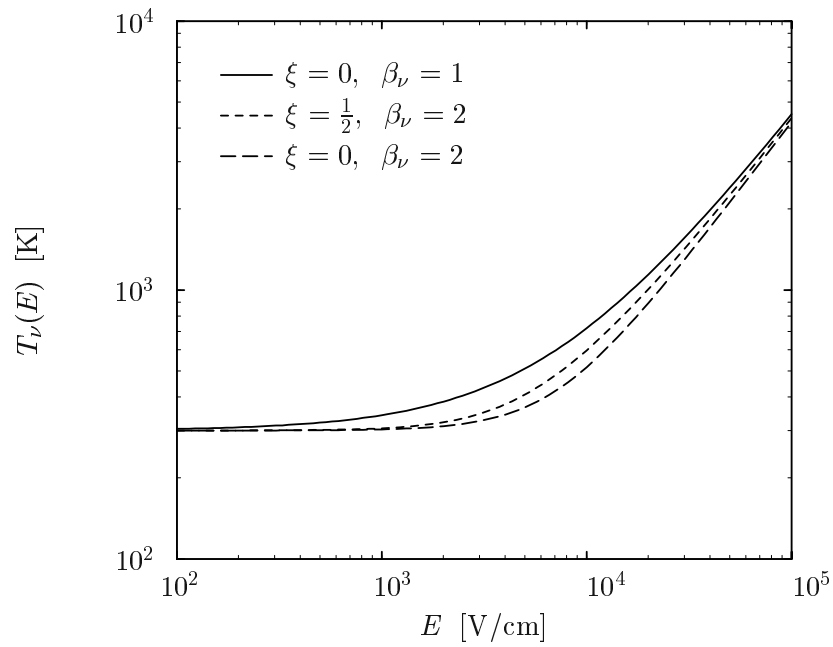


Figure 3.3: Carrier temperature as a function of electric field for the approach of Baccarani as given by (3.54).

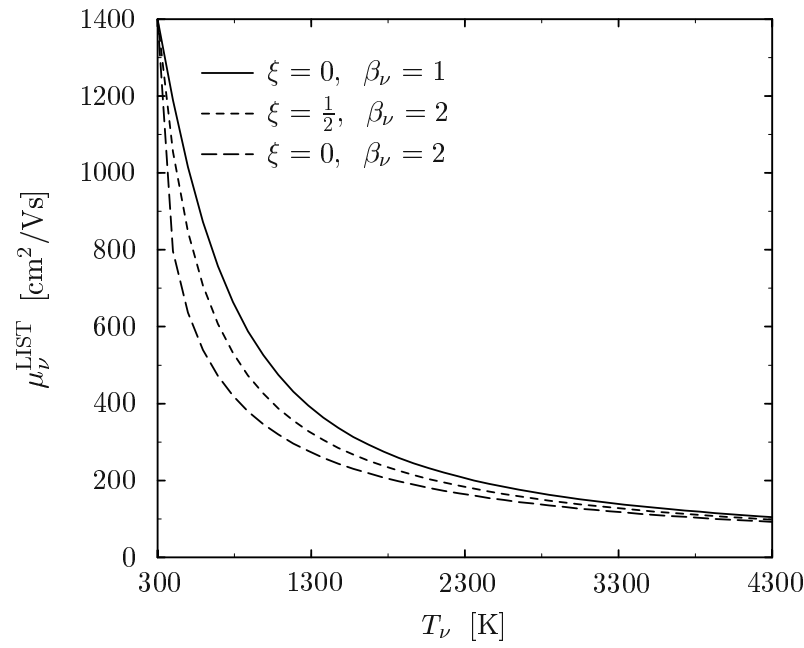


Figure 3.4: Carrier mobility as a function of carrier temperature for the approach of Hänsch as given by (3.48)-(3.50).

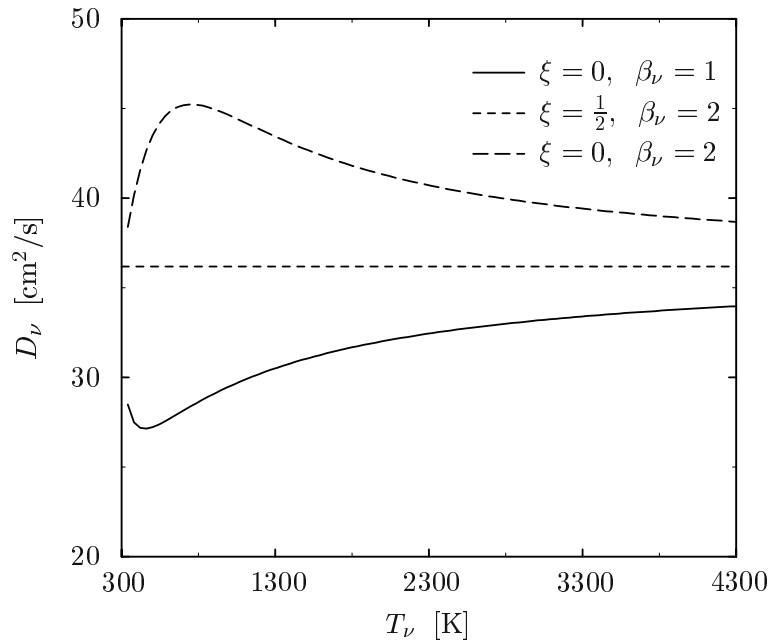


Figure 3.5: Carrier diffusion coefficient as a function of carrier temperature for the approach of Hänsch.

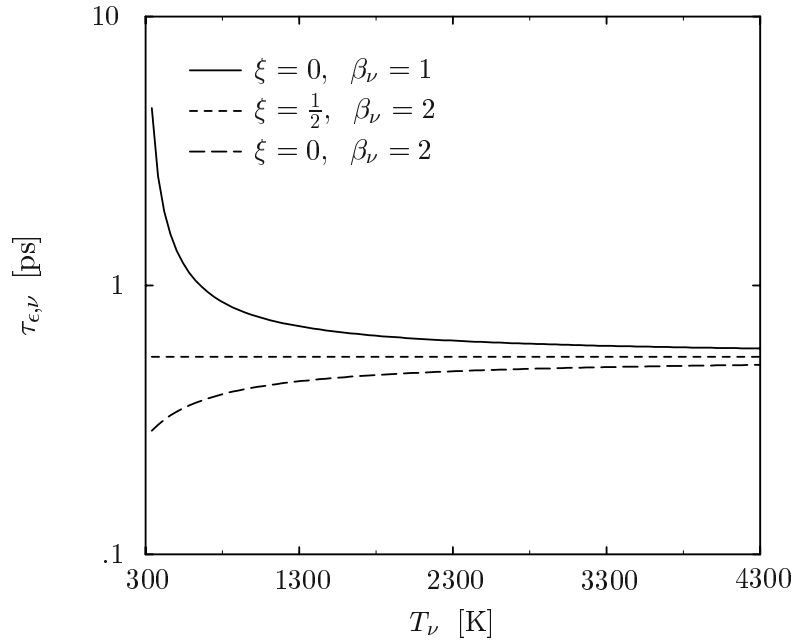


Figure 3.6: Carrier energy relaxation time as a function of carrier temperature for the approach of Baccarani.

also expect (3.56) to result in an energy independent $\tau_{\epsilon,\nu}$. This is indeed the case and one obtains

$$\tau_{\epsilon,\nu} = \frac{3 \cdot k_B \cdot \mu^{\text{LIS}} \cdot T_L}{2 \cdot q \cdot v_{\nu}^{\text{sat}^2}} = \tau_{\epsilon,\nu}^0 \quad . \quad (3.57)$$

Using (3.57) in (3.50) gives $\alpha_\nu = 1/T_L$ and thus

$$\mu_\nu^{\text{LIST}}(T_\nu) = \frac{\mu_\nu^{\text{LIS}}}{1 + \frac{\Delta T_\nu}{T_L}} = \mu_\nu^{\text{LIS}} \cdot \frac{T_L}{T_\nu} \quad (3.58)$$

which is in fact (3.54). It is to note that (3.57) gives a model for $\tau_{\epsilon,\nu}$ for which both assumptions produce the same results. However, (3.57) depends on the low-field mobility which can vary dramatically in the device. Most unfortunately no measured data could be found as yet in the literature to confirm this result.

Another interesting feature of (3.56) is

$$\lim_{T_\nu \rightarrow \infty} \tau_{\epsilon, \nu}(T_\nu) = \tau_{\epsilon, \nu}^0 \cdot \lim_{T_\nu \rightarrow \infty} \frac{T_\nu \cdot \Delta T_\nu}{T_L^2} \cdot \left(\left(\frac{T_\nu}{T_L} - \xi \right)^{\beta_\nu} - \zeta^{\beta_\nu} \right)^{-2/\beta_\nu} \quad (3.59)$$

$$= \tau_{\epsilon, \nu}^0 \cdot \lim_{T_\nu \rightarrow \infty} \frac{t^2 - t}{\left((t - \xi)^{\beta_\nu} - \zeta^{\beta_\nu} \right)^{2/\beta_\nu}} \quad \text{with} \quad t = \frac{T_\nu}{T_L} \quad (3.60)$$

$$= \tau_{\epsilon, \nu}^0 \cdot \lim_{T_\nu \rightarrow \infty} \frac{1 - \frac{1}{t}}{\left(\left(1 - \frac{\xi}{t} \right)^{\beta_\nu} - \frac{\zeta^{\beta_\nu}}{t^{\beta_\nu}} \right)^{2/\beta_\nu}} \quad (3.61)$$

$$= \tau_{\epsilon, \nu}^0 \quad (3.62)$$

independent of ξ and $\beta_\nu > 0$ as can also be seen in Fig. 3.6. With $\mu_\nu^{\text{LIS}} = 1400 \text{ cm}^2/\text{Vs}$ and $v_\nu^{\text{sat}} = 10^7 \text{ cm/s}$ equation (3.57) gives $\tau_{\epsilon, \nu}^0 = 0.54 \text{ ps}$. However, as it is not unrealistic for the low-field mobility to be reduced to 20% of its maximum value, $\tau_{\epsilon, \nu}^0$ is reduced in the very same way.

3.2.4 Velocity Overshoot

When the HD model is used velocity overshoot effects could become important when the carrier temperatures and the electric field are not in equilibrium. This is the case whenever the electric field changes rapidly, both in space or in time. An example is shown in Fig. 3.7a where a step like field profile is assumed. The carrier temperature will respond in the HD model with retardation. As the mobility is a function of the carrier temperature the same retardation is seen. As the carrier velocity is defined as $v_\nu = s_\nu \cdot \mu_\nu \cdot E$ there is a small transition region where the velocity exceeds its stationary value considerably as shown in Fig. 3.7b. Also shown is the velocity resulting from Monte-Carlo (MC) simulations. Due to backscattering the distribution function in front of the field discontinuity is altered, an effect which cannot be modeled using a HD model. Overshoot effects are important to model for small devices as they may alter the performance.

3.2.5 Low-Field Mobility Reconsidered

Another important issue when comparing DD and HD simulations is caused by the fact that in conventional mobility models the same low-field μ_ν^{LIS} mobility is used for both transport models. This is problematic for position-dependent local models as is the case with the MINIMOS mobility model. Comparing the diffusion component of the DD and HD current (assuming constant density of states)

$$\mathbf{J}_\nu^{\text{diff, DD}} = -s_\nu \cdot \mu_\nu \cdot k_B \cdot T_L \cdot \text{grad} \nu \quad (3.63)$$

$$\mathbf{J}_\nu^{\text{diff, HD}} = -s_\nu \cdot \mu_\nu \cdot k_B \cdot \text{grad} (\nu \cdot T_\nu) \quad (3.64)$$

$$= \frac{T_\nu}{T_L} \cdot \mathbf{J}_\nu^{\text{diff, DD}} - s_\nu \cdot \mu_\nu \cdot k_B \cdot \text{grad} T_\nu \quad (3.65)$$

it becomes obvious that the gradient of the carrier temperature causes another component of the diffusion current. Furthermore, the diffusion current due to the carrier gradient is enhanced by a factor T_ν/T_L . Both effects tend to broaden the carrier distributions in space.

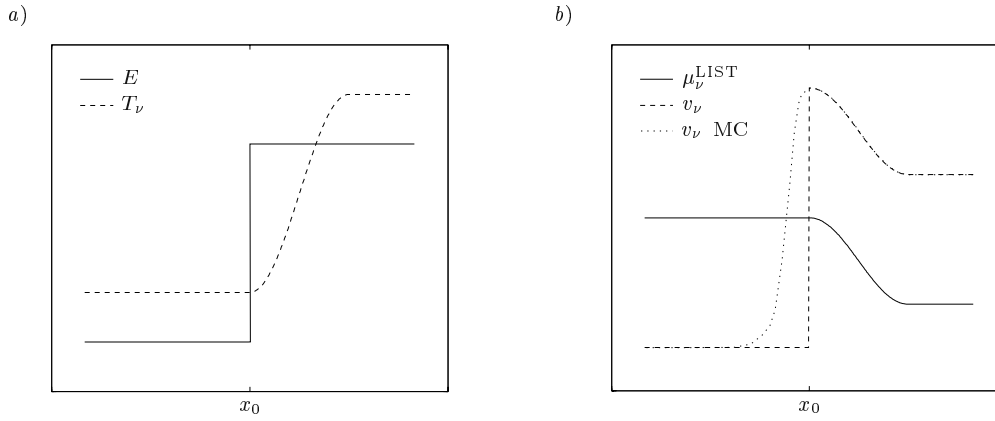


Figure 3.7: a) Carrier temperature response for a step in electric field when using the HD model. b) Carrier mobility and velocity response for a step in electric field when using the HD model.

This effect is best illustrated in the channel of an NMOS transistor. The carrier distributions before and at the pinch-off point are shown in Fig. 3.13 and Fig. 3.14, respectively.

Since there are less carriers at the surface, the surface mobility model has a different impact on the resulting current which will be larger in the HD case. To account for the different carrier distributions the reference distance y^{ref} in (3.28) is modified to

$$y^{\text{ref,HD}} = y^{\text{ref,DD}} \cdot \left(\frac{T_\nu}{T_L} \right)^\eta \quad (3.66)$$

These broadened carrier distributions are the reason why the DD model tends to overestimate the electric field as the carrier concentration increases the space charge density in the channel.

3.3 Examples

The above models were implemented in MINIMOS-NT and several simulations were carried out to confirm the theoretical results and the validity of the simplifying assumptions.

3.3.1 Homogeneous Semiconductor

As a first example homogeneously doped Si blocks with sidelength $a = 10 \mu\text{m}$ as shown in Fig. 3.8a were investigated. The doping levels were $N_D = 10^{17} \text{ cm}^{-3}$ and $N_A = 10^{17} \text{ cm}^{-3}$, respectively. Up to medium electric fields the DD and HD currents agree as one would expect. However, for extremely high-fields close to the breakdown voltage the difference was found to be 3%. This arises from the equilibrium contact condition for the carrier temperatures

$$T_\nu|_{\text{contact}} = T_L \quad (3.67)$$

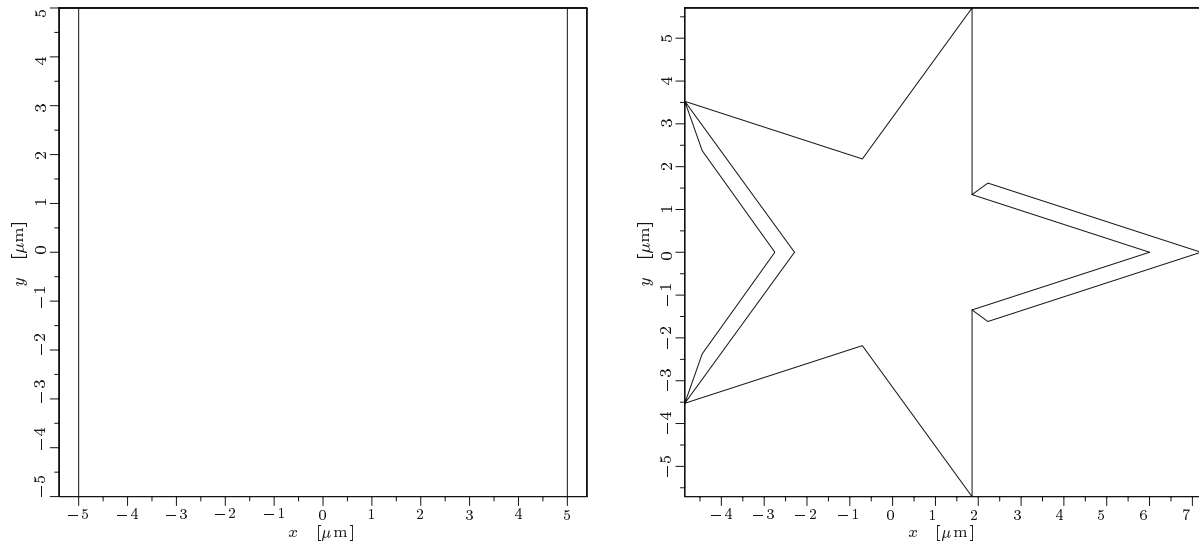


Figure 3.8: a) Geometry of the homogeneous doped semiconductor.
b) Geometry of Gummel's pentagon.

which dramatically violates the local energy balance equation by causing a strong gradient in the carrier temperature. As the electric field is constant and hence known at the contact one can derive a boundary condition from (3.36) which eliminates this discrepancy.

$$T_\nu|_{\text{contact}} = T_L + \frac{2 \cdot q \cdot \tau_{e,\nu} \cdot E^2 \cdot \mu_\nu^{\text{LIS}}(T_\nu)}{3 \cdot k_B} \quad (3.68)$$

The resulting electric fields and carrier temperatures for both contact models are shown in Fig. 3.9 and Fig. 3.10. Equation (3.68) results in a constant electric field and thus, via (3.36), in constant carrier temperatures. The conventional equilibrium contact model forces the hole temperature to be equal to the lattice temperature at the left contact. As the holes move from left to right, they pick up energy from the electric field and their temperature rises. As a consequence, the mobility reduces. To keep up the constant current density the electric field has to increase accordingly. However, for normal ohmic contacts the electric field at the contact is small, hence it is not necessary to use (3.68) except for this pathological situation.

The I-V curves for n-doped and p-doped semiconductors are shown in Fig. 3.11. In addition to the matching mobility models, the p-doped semiconductor was simulated using (3.41) with $\beta_p = 1$ in combination with (3.50) which is one of the most common errors. As can be seen, the error is intolerably large.

3.3.2 Gummel's Pentagon

The second example is a structure proposed by Gummel [28] as shown in Fig. 3.8b. It can be shown that the pentagon can be transformed into an equivalent quadratic resistor with side-length $2 \cdot a$. This is quite a severe test for both the grid generator and the implemented discretization of the semiconductor equations. The simulation results are identical to Fig. 3.11 and hence not shown.

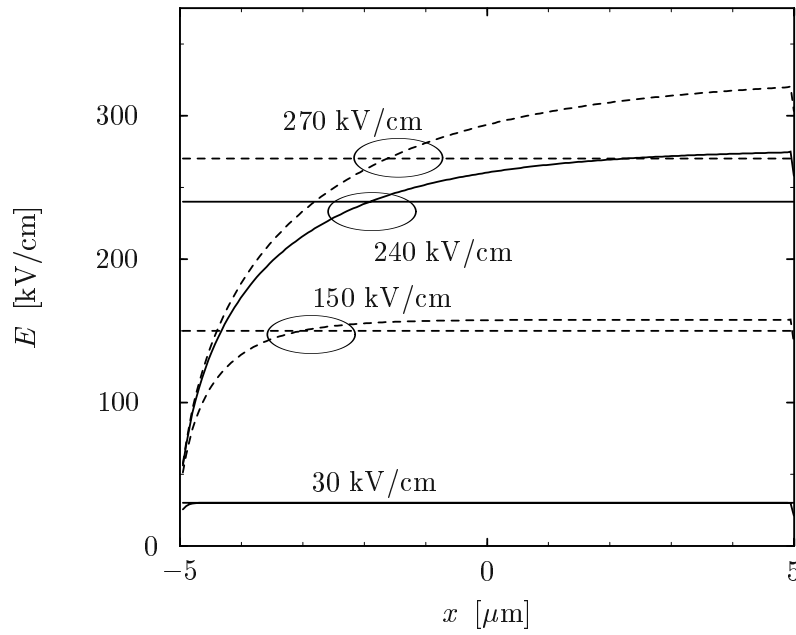


Figure 3.9: Influence of the boundary condition for the carrier temperatures on the distribution of the electric field inside the homogeneous p-resistor for various bias conditions. The horizontal lines belong to the special contact model (3.68).

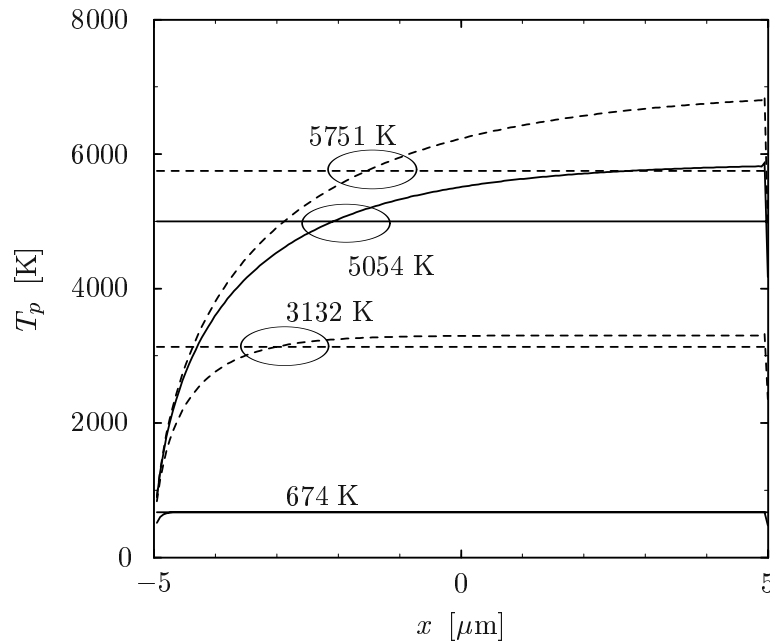


Figure 3.10: Influence of the boundary condition for the carrier temperatures on the distribution of the hole temperature inside the homogeneous p-resistor for various bias conditions. The horizontal lines belong to the special contact model (3.68).

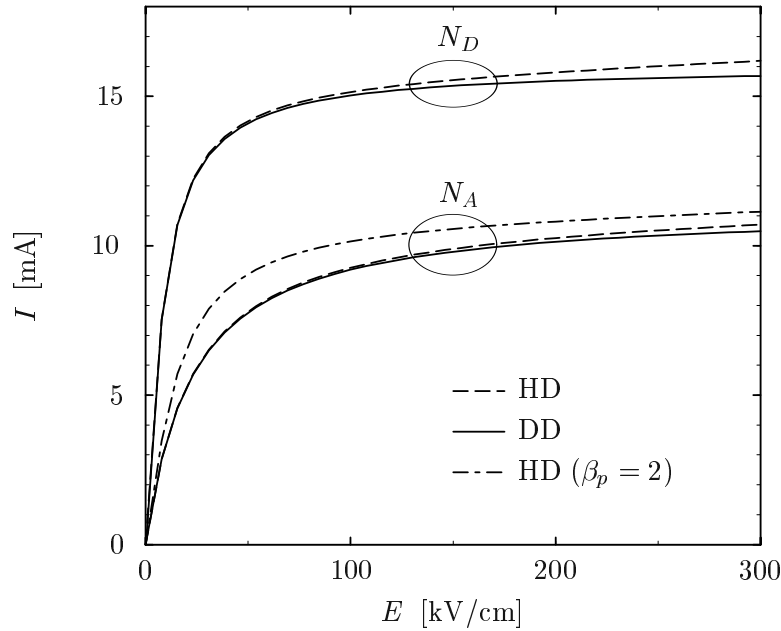


Figure 3.11: I-V curves for the homogeneously n- and p-doped resistors for DD and HD simulations. In addition, for the p-doped resistor the current for $\beta_p = 2$ is shown.

3.3.3 NMOS Transistors

A long-channel ($L_G = 2.0 \mu\text{m}$) and a short-channel ($L_G = 0.2 \mu\text{m}$) NMOS transistor were considered. The substrate doping level was $N_A = 10^{15} \text{ cm}^{-3}$ and the maximum of the drain and source Gaussian contact doping peaks was $N_D = 10^{20} \text{ cm}^{-3}$ for both transistors. The device thickness was $W = 1 \mu\text{m}$ and the oxide thickness $d_{ox} = 5 \text{ nm}$.

For the long-channel device non-local effects were expected to play a minor role. The doping profiles of both transistors are shown in Fig. 3.12a and Fig. 3.12b, respectively. Although these transistors are very simple compared to state of the art, they allow for studying the principal effects.

For long-channel devices, the drain current in the pinch-off region can be calculated from simple analytical models as [25]

$$I_D = \frac{\mu \cdot \varepsilon_{ox}}{2 \cdot d_{ox}} \cdot \frac{W}{L_G} \cdot (V_G - V_{th})^2 \quad (3.69)$$

with V_{th} being the threshold voltage. For short-channel devices (3.69) becomes invalid as velocity saturation occurs in the channel. However, the ratio W/L_G still determines the drain current. Thus, in the figures $I_D \cdot L_G/W$ is shown instead of I_D .

The numerical simulations were performed using the mobility models (3.41) for DD and (3.50) for HD. For the HD transport model, simulations with $\eta = 0$ (uncorrected surface distance model) and $\eta = 1$ (corrected surface distance model) were carried out. The broadening of the carrier distributions for the long-channel device is shown in Fig. 3.13 and Fig. 3.14 for immediately before the pinch-off point and inside the pinch-off point, respectively. A

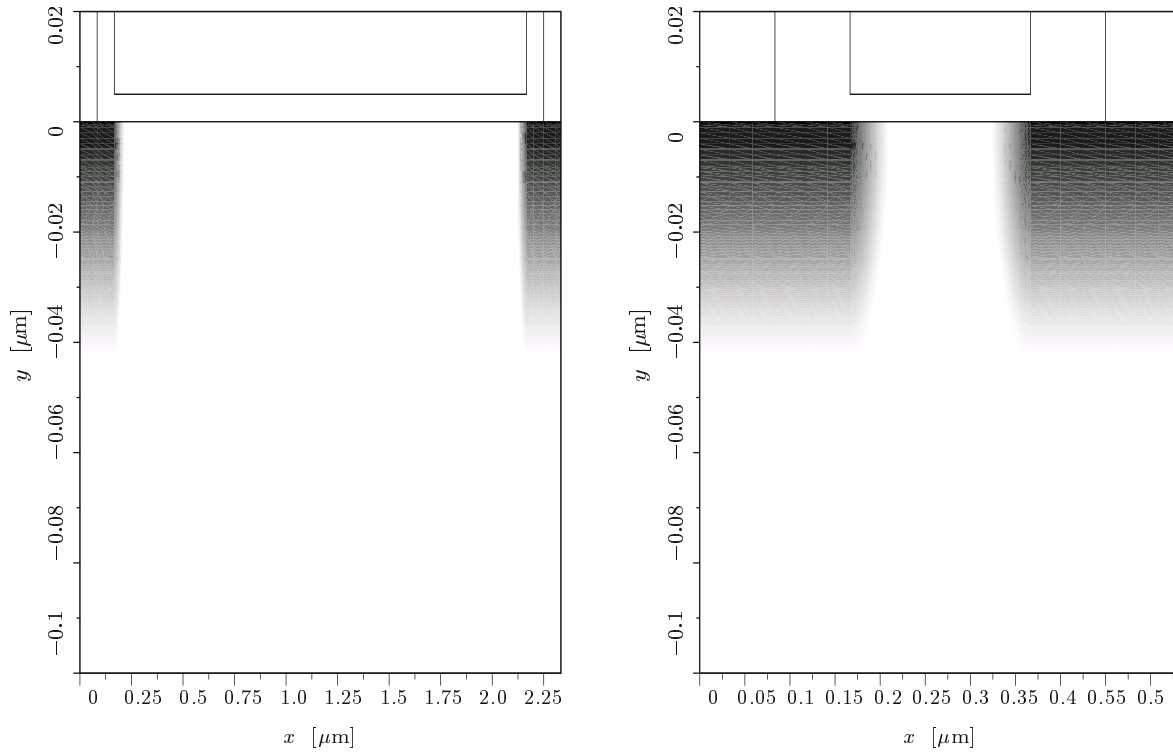


Figure 3.12: Doping profiles of a) the long-channel ($L_G = 2.0 \mu\text{m}$) and b) the short-channel ($L_G = 0.2 \mu\text{m}$) NMOS transistor.

comparison of the output characteristics for both transport models is shown in Fig. 3.15 and Fig. 3.16. As expected, the device with $L_G = 0.2 \mu\text{m}$ shows typical short-channel behavior and $I_D \cdot L_G/W$ is reduced by 50%. Due to velocity overshoot in the channel, the HD currents are considerably higher than for the DD transport model.

3.3.4 PMOS Transistor

The PMOS transistors were derived from the NMOS transistors by exchanging N_D with N_A . As the hole mobility is about 1/3 lower compared to the electron mobility the carrier temperatures do not rise to such high levels as compared to the NMOS. Hence, non-local effects do not play such an important role. This is confirmed by the simulated output characteristics which are shown in Fig. 3.17 and Fig. 3.18 for both devices. As for the homogeneously p-doped semiconductor both devices were simulated using (3.41) with $\beta_p = 1$ in combination with (3.50). As can be seen in Fig. 3.17 and Fig. 3.18, the error is again intolerably large.

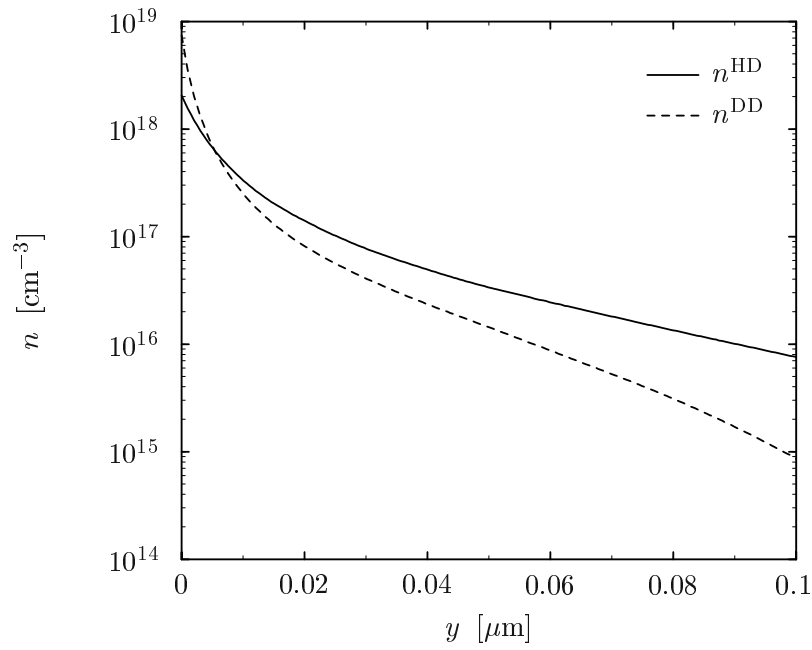


Figure 3.13: Electron concentration before the pinch-off point for the long-channel NMOS for both transport models ($x = 2.12 \mu\text{m}$).

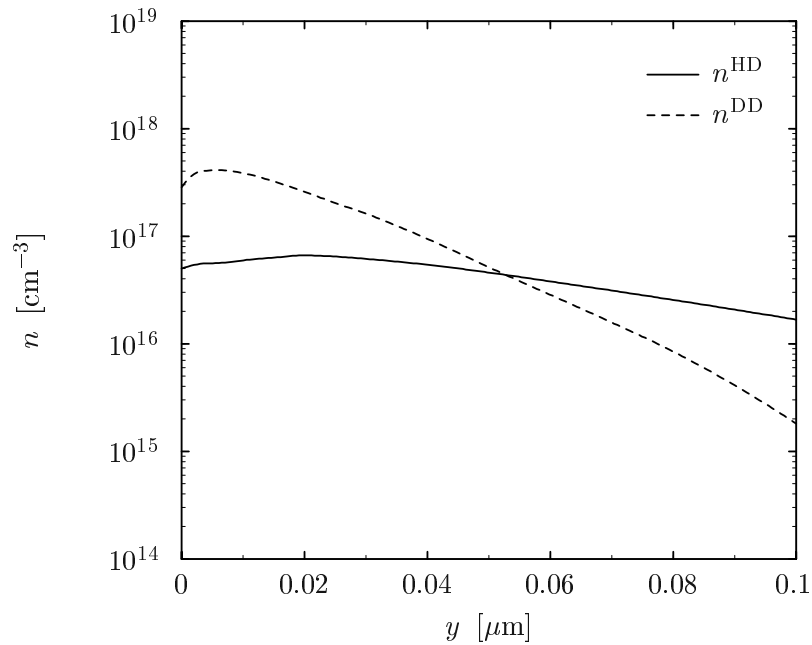


Figure 3.14: Electron concentration in the pinch-off point for the long-channel NMOS for both transport models ($x = 2.12 \mu\text{m}$).

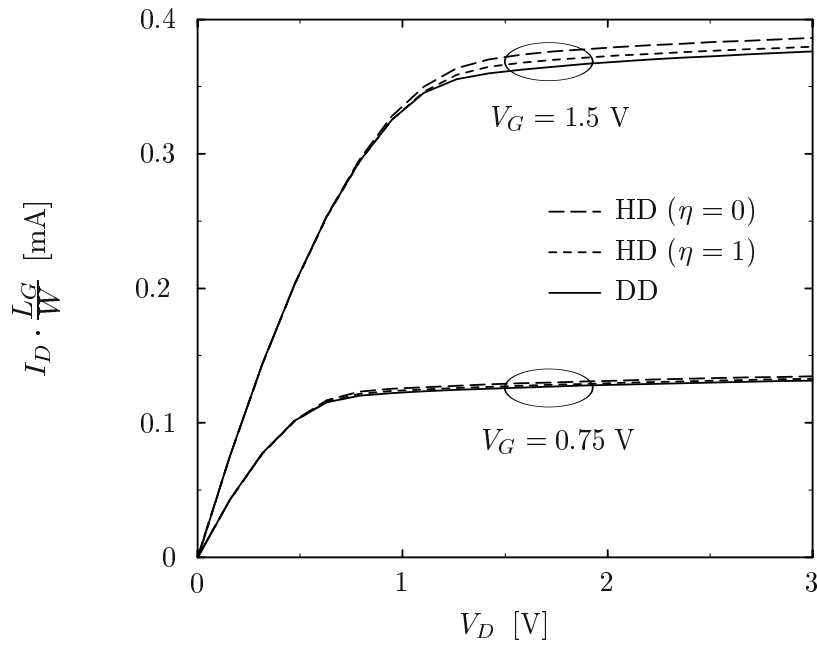


Figure 3.15: Comparison of the output characteristics of the long-channel NMOS for both transport models.

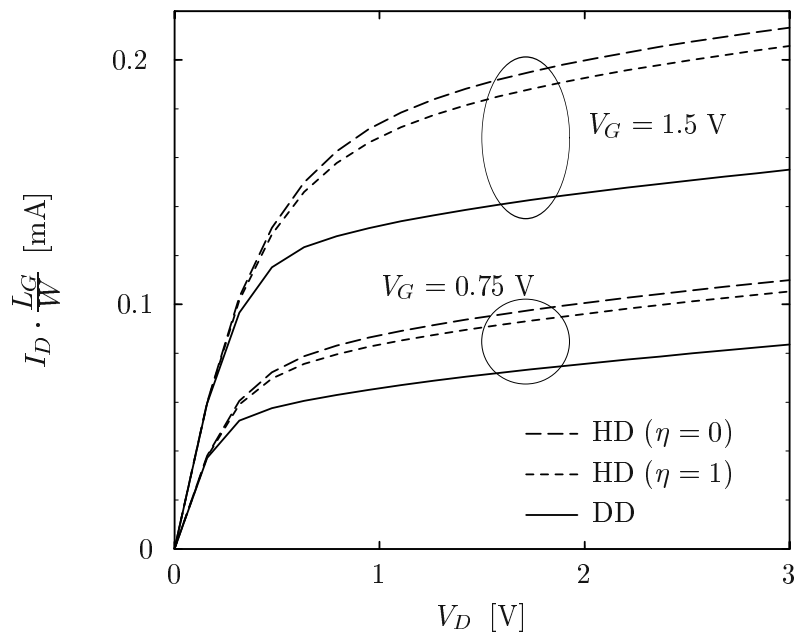


Figure 3.16: Comparison of the output characteristics of the short-channel NMOS for both transport models.

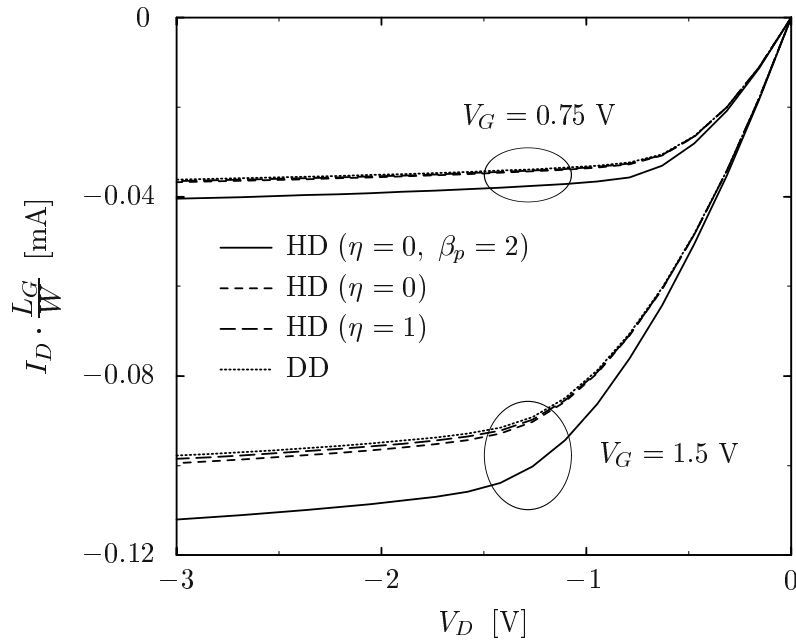


Figure 3.17: Comparison of the output characteristics of the long-channel PMOS for both transport models.

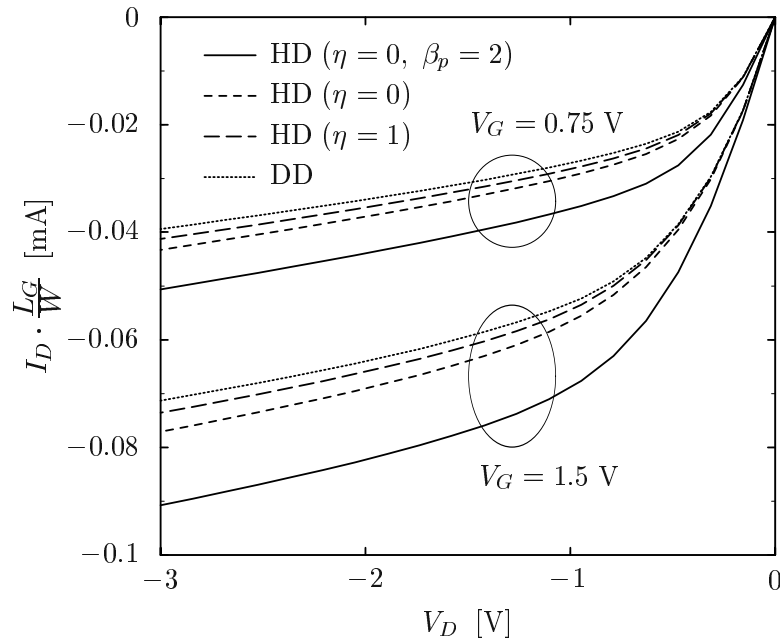


Figure 3.18: Comparison of the output characteristics of the short-channel PMOS for both transport models.

Chapter 4

Simulator Coupling

Several works dealing with circuit simulation using distributed devices have been published so far [12, 39, 40, 49]. Most publications deal with the coupling of device simulators to SPICE. This results in a two-level Newton algorithm since the device and circuit equations are handled subsequently. Each solution of the circuit equations gives a new operating point for the distributed devices. The device simulator is then invoked to calculate the resulting currents and the derivatives of these currents with respect to the contact voltages. The other approach is called full-Newton algorithm as it combines the device and circuit equations within one single equation system. This equation system is then solved applying Newton's algorithm. In contrast to the two-level Newton algorithm where the device and circuit unknowns are solved in a decoupled manner, here the complete set of unknowns is solved simultaneously.

4.1 The Two-Level Newton Algorithm

The problem of a two-level device and circuit simulation is demonstrated by an example circuit as shown in Fig. 4.1 where V_S is a constant voltage source and G is a linear conductance. The non-linear device is a diode whose characteristic is determined by the doping profile and the physical models used in the device simulation. The DC characteristic is

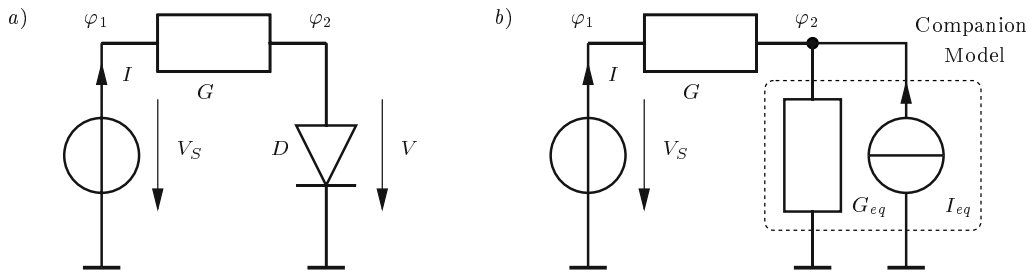


Figure 4.1: a) Original circuit and b) linearized companion model for the two-level Newton algorithm

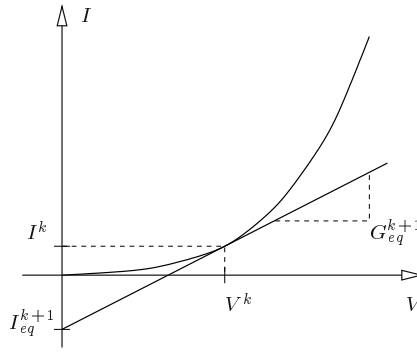


Figure 4.2: Non-linear characteristic of the diode and the quantities for the linearized companion model

shown in Fig. 4.2. In Fig. 4.1a the original circuit is shown. When using Newton's method to solve the non-linear circuit equations, a linear companion circuit [9] is solved at each iteration until some convergence criterion is met. The companion circuit for arbitrary non-linear two-terminal devices is shown in Fig. 4.1b. It consists of a linear conductance G_{eq} and a current source I_{eq} which depend on the operating point. They are derived from the non-linear characteristic as shown in Fig. 4.2 and read

$$G_{eq}^{k+1} = \left. \frac{dI}{dV} \right|_k \quad (4.1)$$

$$I_{eq}^{k+1} = I^k - G_{eq}^{k+1} \cdot V^k \quad (4.2)$$

with k being the iteration number of the Newton algorithm. For compact models, G_{eq} and I_{eq} are evaluated analytically. As this is not possible for distributed devices, the device simulator has to be called for each Newton iteration. The node voltages are initialized to values supplied by the user and default to zero. After this initialization, the device simulator has to be called to calculate the companion model.

For the calculation of the conductances an approach as proposed in [39] is well suited and will be outlined in the following section.

4.1.1 Calculation of Conductances

The discretized semiconductor equations are represented by the non-linear equation system $\mathbf{f}(\mathbf{x}) = 0$. The solution vector \mathbf{x} contains the carrier concentrations, the potential, the contact voltages, and some more quantities depending on the simulation mode. As the contact voltages are the quantities which determine the state of the device, the dependence of the solution vector on the contact voltages is explicitly written in the equation system. For Fig. 4.1 one gets

$$\mathbf{f}(\mathbf{x}(V), V) = 0 \quad (4.3)$$

By applying Newton's algorithm to (4.3) with $V = V_0$ one gets

$$-\mathbb{J}_{\mathbf{x}}^k \cdot \mathbf{u} = \mathbf{f}(\mathbf{x}^k, V_0) \quad \text{with} \quad \mathbb{J}_{\mathbf{x}}^k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}^k, V_0} \quad (4.4)$$

The iteration terminates after some convergence criterion is fulfilled. As a next step the contact current $I(\mathbf{x}, V_0)$ can be calculated. For the calculation of the linearized conductance G_{eq} the chain rule is applied to (4.1) which gives

$$G_{eq} = \frac{dI}{dV} = \frac{\partial I}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial V} + \frac{\partial I}{\partial V} \quad (4.5)$$

where $\partial I / \partial \mathbf{x}$ and $\partial I / \partial V$ are obtained by symbolic differentiation of the function $I(\mathbf{x}, V)$. The quantity $\partial \mathbf{x} / \partial V$ can be determined in two different ways. The first approach has been published in [76]. At first, (4.4) is solved until a user-defined convergence criterion is reached. Then a small perturbation ΔV is applied on V_0 . Using the Taylor expansion of (4.3) with respect to both \mathbf{x} and V around the DC solution, $\mathbf{f}(\mathbf{x}_0, V_0)$ results in

$$\mathbf{f}(\mathbf{x}_0 + \Delta \mathbf{x}, V_0 + \Delta V) \approx \mathbf{f}(\mathbf{x}_0, V_0) + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \big|_{\mathbf{x}_0, V_0} \cdot \Delta \mathbf{x}}_{\mathbb{J}_{\mathbf{x}}} + \underbrace{\frac{\partial \mathbf{f}}{\partial V} \big|_{\mathbf{x}_0, V_0} \cdot \Delta V}_{\mathbb{J}_V} + \dots = \mathbf{0} \quad (4.6)$$

As $\mathbf{f}(\mathbf{x}_0, V_0) = \mathbf{0}$, the above equation reduces to

$$\mathbb{J}_{\mathbf{x}} \cdot \Delta \mathbf{x} = -\mathbb{J}_V \cdot \Delta V \quad (4.7)$$

The Jacobian matrix $\mathbb{J}_{\mathbf{x}}$ is the same as for the last iteration of the DC solution and the vector $\mathbb{J}_V \cdot \Delta V$ on the right-hand side normally contains only a few non-zero elements as only the quantities at the boundaries directly depend on V . The solution of the linear system (4.7) $\Delta \mathbf{x}$ is proportional to ΔV , hence $\Delta \mathbf{x} / \Delta V$ is independent of ΔV and equal to the limiting case which is the derivative when ΔV becomes infinitesimally small.

$$\frac{\partial \mathbf{x}}{\partial V} = \frac{\Delta \mathbf{x}}{\Delta V} = -\mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbb{J}_V \quad (4.8)$$

Another equivalent possibility to calculate $\partial \mathbf{x} / \partial V$ is from the total derivative of (4.3) with respect to V which reads

$$\frac{d\mathbf{f}(\mathbf{x}(V), V)}{dV} = \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}}}_{\mathbb{J}_{\mathbf{x}}} \cdot \frac{\partial \mathbf{x}}{\partial V} + \underbrace{\frac{\partial \mathbf{f}}{\partial V}}_{\mathbb{J}_V} = \mathbf{0} \quad (4.9)$$

and also results in (4.8). To summarize, the equivalent conductance and current are calculated as follows

$$G_{eq}^{k+1} = \left(-\frac{\partial I}{\partial \mathbf{x}} \cdot \mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbb{J}_V + \frac{\partial I}{\partial V} \right)_k \quad (4.10)$$

$$I_{eq}^{k+1} = I^k - G_{eq}^{k+1} \cdot V^k \quad (4.11)$$

4.1.2 Modified Two-Level Newton Algorithm

A linear prediction step can be used to obtain an initial-guess from the previous solution. The two-level Newton scheme with a linear prediction step is referred to as the modified

two-level Newton algorithm. A first-order prediction is made by use of the forward-Euler scheme,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \left. \frac{\partial \mathbf{x}}{\partial V} \right|_k \cdot \Delta V \quad (4.12)$$

where ΔV is the change in voltage from circuit iteration k to $k + 1$, and $\partial \mathbf{x} / \partial V$ is calculated as in (4.8). In the investigations of [39] this algorithm had the most reliable convergence properties.

4.2 The Quasi Full-Newton Algorithm

In [39] a method was proposed which was termed full-Newton algorithm. However, this approach is very similar to the two-level method proposed in the same paper hence it is termed “quasi” full-Newton in this thesis. The basic idea will be demonstrated for the circuit shown in Fig. 4.1. The combined device and circuit equations read

$$\mathbf{f}(\mathbf{x}, V) = \mathbf{0} \quad (4.13)$$

$$I(\mathbf{x}, V) + G \cdot (V - V_S) = 0 \quad (4.14)$$

Applying Newton’s method to (4.13) and (4.14) one gets

$$\mathbb{J}_{\mathbf{x}} \cdot \Delta \mathbf{x} + \mathbb{J}_V \cdot \Delta V = -\mathbf{f}(\mathbf{x}, V) \quad (4.15)$$

$$\frac{\partial I}{\partial \mathbf{x}} \cdot \Delta \mathbf{x} + \frac{\partial I}{\partial V} \cdot \Delta V + G \cdot \Delta V = -I(\mathbf{x}, V) - G \cdot (V - V_S) \quad (4.16)$$

Rearranging (4.15) yields

$$\Delta \mathbf{x} = \mathbb{J}_{\mathbf{x}}^{-1} \cdot (-\mathbf{f}(\mathbf{x}, V) - \mathbb{J}_V \cdot \Delta V) \quad (4.17)$$

which can be rewritten as

$$\Delta \mathbf{x} = \Delta \hat{\mathbf{x}} - \mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbb{J}_V \cdot \Delta V \quad (4.18)$$

$$\Delta \hat{\mathbf{x}} = -\mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbf{f}(\mathbf{x}, V) \quad (4.19)$$

Substituting (4.18) in (4.16) yields

$$\left(-\frac{\partial I}{\partial \mathbf{x}} \cdot \mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbb{J}_V + \frac{\partial I}{\partial V} + G \right) \Delta V = -\frac{\partial I}{\partial \mathbf{x}} \cdot \Delta \hat{\mathbf{x}} - I(\mathbf{x}, V) - G \cdot (V - V_S) \quad (4.20)$$

This equation can be rewritten as

$$(G_{eq} + G) \cdot \Delta V = -I_T - G \cdot (V - V_S) \quad (4.21)$$

with

$$G_{eq} = -\frac{\partial I}{\partial \mathbf{x}} \cdot \mathbb{J}_{\mathbf{x}}^{-1} \cdot \mathbb{J}_V + \frac{\partial I}{\partial V} \quad (4.22)$$

$$I_T = \frac{\partial I}{\partial \mathbf{x}} \cdot \Delta \hat{\mathbf{x}} + I(\mathbf{x}, V) \quad (4.23)$$

Equation (4.21) is similar in form to that obtained by the two-level Newton algorithm. Hence, similar methods can be used to embed distributed devices into a circuit simulator and to provide a decoupling between both simulators even for this quasi full-Newton algorithm.

4.3 The Full-Newton Algorithm

The approaches discussed above combined a circuit simulator with a device simulator by adding a suitable interface to the circuit simulator. These approaches are easy to implement as only marginal changes in both simulators are required. The circuit simulator acts as server which controls the device simulator. At each Newton iteration of the circuit, an input deck for the device simulator has to be generated and the device simulator has to be called to calculate currents and conductances.

The approach taken in MINIMOS-NT is completely different. The capability to solve circuit equations was added to the simulator kernel. This allowed for assembling the circuit *and* the device equations into one system matrix which results in a real full-Newton method. There is no need to explicitly calculate the derivatives of the contact currents with respect to the contact voltages as the contact currents are solution variables which simply gives ± 1 as a derivative in the constitutive relations.

However, the benefits gained from using the numerous existing SPICE models must not be neglected. As SPICE has a well defined and documented interface, it is, in principle, straight-forward to implement a similar interface in MINIMOS-NT. SPICE was originally written in FORTRAN and has been ported to C (SPICE3). Extensive use had been made of C-preprocessor macros which caused some software-technical difficulties when designing a neat C++ interface as needed by MINIMOS-NT. Mapping the internal state of MINIMOS-NT for the SPICE models was also quite tedious. However, these problems have been nicely solved in [75] where a tool is presented which automatically generates C++ source code for MINIMOS-NT. These source files can be compiled and either directly linked to MINIMOS-NT or dynamically loaded at runtime. The second approach allows for a compact simulator which can be extended whenever needed.

A comparison of these different architectures is shown in Fig. 4.3. In Fig. 4.3a the device simulator acts as a client to the circuit simulator whereas in Fig. 4.3b the device simulator is extended with circuit simulator capabilities and can reuse circuit simulator models on demand.

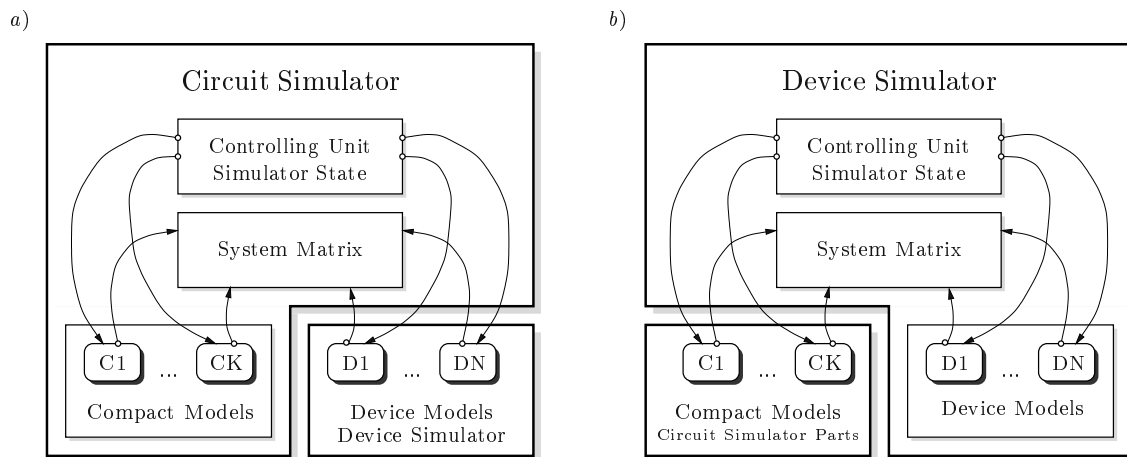


Figure 4.3: Comparison of the two different strategies: a) Device simulator as client. b) Device simulator as server

Chapter 5

Contacts and Boundaries

The calculation of fluxes over boundaries and interfaces is of fundamental importance to process and device simulation. These fluxes may be electrical currents, heat fluxes, or particle fluxes. The discussion in this thesis is restricted to device and circuit simulation but it should be mentioned that this approach is generally applicable. In case of mixed-mode device simulation the contact currents and their derivatives with respect to the contact voltages are of utmost importance. They are normally calculated in a post-processing step [39]. The problem can be simplified by using a separate variable for the contact current which was first reported in [14] and [40] in conjunction with a finite differences discretization scheme [56]. This concept has been extended by using separate variables for all fluxes at the contacts. It will be shown that with these fluxes the formulation of boundary conditions is extremely simplified, especially in the case of flux boundary conditions, e.g., current or charge controlled contacts or mixed-mode circuit simulation.

To simplify flux calculation on boundaries and interfaces, a special formulation has been developed and implemented in MINIMOS-NT [15]. The basic idea is to provide a line-transformation matrix which supports arbitrary linear combination of equations. Due to this approach the semiconductor and insulator segments can be modeled by general segment models *without* considering the boundaries. However, as the segment models only treat points of the same segment, flux conservation is violated at the boundary boxes. The boundary models have to either add the missing flux (Neumann boundary) or provide a substitute equation for the contact quantity (Dirichlet boundary). When providing a substitute equation, the incomplete equation can be reused to calculate the boundary flux. All this is accomplished by the line-transformation matrix which is also used to formulate interface conditions between adjacent segments in a very similar manner. As these substitute equations are not guaranteed to be diagonal-dominant, use of an iterative solver would be problematic. Hence, those equations can be marked for pre-elimination using a Gaussian elimination algorithm. The resulting equation system is solved using a BiCGStab iterative solver [10] and the pre-eliminated quantities are back-substituted.

5.1 Quantities, Segments, Boundaries, and Interfaces

Quantities are defined by the basic semiconductor equations and are normally ψ , n , p , T_n , T_p , and T_L . They are continuous functions in space. When discretizing these quantities on the simulation grid one obtains the *variables* on the grid points. Geometrical regions

within the device employing a distinct set of models are called *segments*. *Contact segments* are special segments which handle the contact models. Segments are connected to other segments by *interfaces* and to contact segments by *boundaries*.

5.2 Equation Assembly

MINIMOS-NT solves $\mathbf{f} = 0$ using a damped Newton algorithm:

$$-\mathbb{J}^k \cdot \mathbf{u} = \mathbf{f}(\mathbf{x}^k) \quad (5.1)$$

$$\mathbf{x}^* = \mathbf{x}^k + d \cdot \mathbf{u} \quad (5.2)$$

$$\mathbb{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \quad (5.3)$$

Applying the finite boxes discretization scheme to the basic semiconductor equations one gets for each quantity ν_i located at the grid point i constitutive relations $f_{\nu_i}^S$ of the form

$$f_{\nu_i}^S = \sum_j F_{\nu_i,j} + G_i = 0 \quad (5.4)$$

where j runs over all neighboring grid points in the same segment, $F_{\nu_i,j}$ is the flux between those grid points, and G_i is the source term at grid point i . (5.4) controls the value of ν . For grid points located at boundaries or interfaces, the situation is different. Despite of general boundary conditions, the two extreme cases Dirichlet (5.5) and Neumann (5.6) are of fundamental importance.

$$\nu = a \quad (5.5)$$

$$F_\nu = b \quad (5.6)$$

In case of a Dirichlet condition, (5.4) is not needed as the value of ν is determined by (5.5). In case of a Neumann condition (5.6) can be used to calculate (5.4) as fluxes over contacts and interfaces cannot be obtained by discretizing the semiconductor equations.

As is obvious from these considerations, boundary and interface points have to be handled in a different manner compared to points located inside the segments. In a generic device simulator like MINIMOS-NT which can handle an arbitrary number of contacts, segments, and interfaces, efficient handling of these situations is of utmost importance. This is achieved by treating the segments separately from the boundary conditions in such a way that all segment points are treated in the same manner. (5.4) is calculated for all segment points including boundary points. The derivatives of (5.4) for all grid points (\mathbf{f}_S) with respect to the solution vector \mathbf{x} yield the segment matrix \mathbb{J}_S .

The system matrix \mathbb{J} (Jacobian) will be assembled from two parts, namely the direct part \mathbb{J}_B (boundary models) and the transformed part \mathbb{J}_S (segment models). The latter will be multiplied by the so called row transformation matrix \mathbb{T}_B before contributing to the system matrix \mathbb{J} . The same treatment applies for the right hand side vector \mathbf{b} .

$$\mathbb{J} = \mathbb{J}_B + \mathbb{T}_B \cdot \mathbb{J}_S \quad (5.7)$$

$$\mathbf{b} = \mathbf{b}_B + \mathbb{T}_B \cdot \mathbf{b}_S \quad (5.8)$$

The direct part of the system matrix \mathbb{J}_B and the row transformation matrix \mathbb{T}_B are set up by the boundary models, while the transformed part \mathbb{J}_S will be handled by the segment

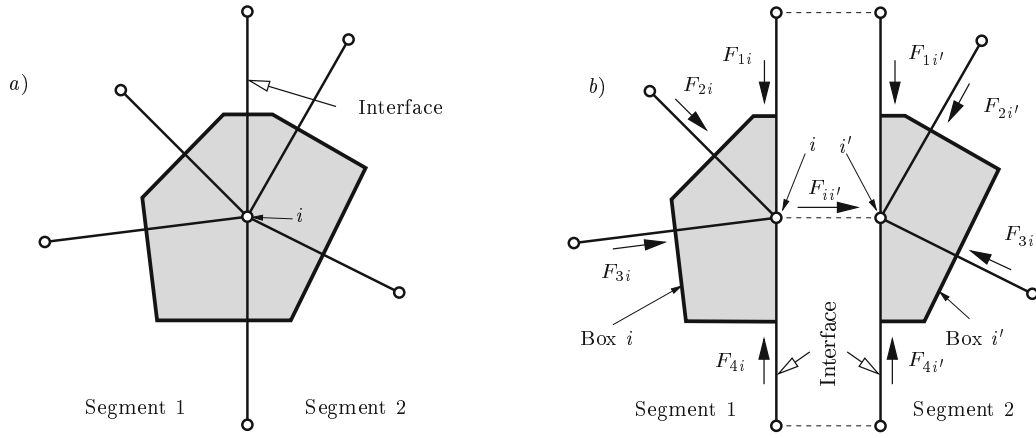


Figure 5.1: Splitting of interface points: Interface points as given in a) are split into two different points having the same geometrical coordinates b)

models. \mathbb{T}_B is initialized to be the unity matrix. In the actual implementation only entries different to the unity matrix will be stored. Direct matrix multiplication is avoided by assembling the transformation matrix in a pre-pass. The transformation matrix allows for arbitrary linear combination of the equations found in \mathbb{J}_S and is normally extremely sparse with most of the non-zero entries being ± 1 . For the special case with the main-diagonal entry $t_{i,i} = 0$, the constitutive relation for grid point i will not contribute to the final constitutive relation i . Another special case is $t_{i,j} = t_{j,j} = 1$ which simply combines rows i and j of \mathbb{T}_S and adds them to row j of \mathbb{T}_B .

5.3 Interfaces

To account for complex interface conditions, variables on grid points located at interfaces (see Fig. 5.1a) are allowed to have two values, one for each segment (see Fig. 5.1b) to account for abrupt changes at interfaces. In the following these interface points will be given as i and i' . The constitutive relations for both points are incomplete since in the segment models only the flux terms to points of the same segment are considered. Furthermore, an additional equation is required for this additional variable at the interface. Depending on the interface type two cases must be considered:

5.3.1 Neumann Type (Explicit Flux)

In case an expression for the flux $F_{\nu_{i,i'}}$ is available, the constitutive relations can easily be completed by

$$f_{\nu_i} = f_{\nu_i}^S + F_{\nu_{i,i'}} = 0 \quad (5.9)$$

$$f_{\nu_{i'}} = f_{\nu_{i'}}^S - F_{\nu_{i,i'}} = 0 \quad (5.10)$$

To accomplish this the respective part of the transformation matrix reads

$t_{x,y}$	ν_i	$\nu_{i'}$
ν_i	1	
$\nu_{i'}$		1

which is actually the default case and hence needs not to be explicitly specified.

The derivatives of the flux term with respect to the solution variables must be of the same magnitude as the other derivatives given by $f_{\nu_i}^S$ in order to guarantee good conditioning of the resulting equation system [15]. Since this cannot generally be guaranteed, an approach similar to Dirichlet interfaces is preferable [63]: Adding (5.9) to (5.10) one gets

$$f_{\nu_i} = f_{\nu_i}^S + F_{\nu_{i,i'}} = 0 \quad (5.11)$$

$$f_{\nu_{i'}} = f_{\nu_i}^S + f_{\nu_{i'}}^S = 0 \quad (5.12)$$

As for Dirichlet interfaces, (5.11) is pre-eliminated before the actual iterative solution process. The appropriate transformation matrix reads

$t_{x,y}$	ν_i	$\nu_{i'}$
ν_i	1	
$\nu_{i'}$	1	1

5.3.2 Dirichlet Type (Implicit Flux)

In case an expression $\nu_i = h(\nu_{i'})$ is available $F_{\nu_{i,i'}}$ can be eliminated by adding (5.9) to (5.10) and the resulting equations are of the following form:

$$f_{\nu_i} = \nu_i - h(\nu_{i'}) = 0 \quad (5.13)$$

$$f_{\nu_{i'}} = f_{\nu_i}^S + f_{\nu_{i'}}^S = 0 \quad (5.14)$$

As (5.13) is normally not diagonal-dominant it is eliminated in a pre-pass. It is important to note, that in this case the structure of the equation system changes. The constitutive relation for ν_i is now given by (5.13) whereas for $\nu_{i'}$ by (5.14), or vice-versa. This is accomplished by the following transformation matrix

$t_{x,y}$	ν_i	$\nu_{i'}$
ν_i		
$\nu_{i'}$	1	1

The fluxes contained in f_{ν_i} can also be used to calculate the total interface flux F_I by setting up the following equation:

$$f_{\nu_{F_I}} = \sum_i f_{\nu_i}^S + F_I = 0 \quad (5.15)$$

with i running over all interface points.

5.4 Boundaries

Boundaries are handled in a similar way to interfaces. However, in the boundary segment there is only one variable available for each solution quantity (ν_C). Furthermore the flux over the boundary is handled as another solution variable F_C (e.g., Q_C for Poisson's equation, I_{n_C} for the electron continuity equations, or H_C as the contact heat flow).

5.4.1 Neumann Type

For Neumann boundary conditions one gets

$$f_{\nu_i} = f_{\nu_i}^S + F_{\nu_i,C} = 0 \quad (5.16)$$

$$f_{F_C} = F_C + \sum_i f_{\nu_i}^S = 0 \quad (5.17)$$

with i running over all segment grid points. \mathbb{T}_B reads (for two example grid points i_1 and i_2)

$t_{x,y}$	ν_{i_1}	ν_{i_2}	F_C
ν_{i_1}	1		
ν_{i_2}		1	
F_C	1	1	

Since there is no segment model for F_C the respective column is arbitrary.

5.4.2 Dirichlet Type

For Dirichlet boundary conditions one gets

$$f_{\nu_i} = \nu_C - h(\nu_i) = 0 \quad (5.18)$$

$$f_{F_C} = F_C + \sum_i f_{\nu_i}^S = 0 \quad (5.19)$$

Here, ν_C is the boundary value of the quantity, which is a solution variable, whereas (5.19) is used as constitutive relation for the actual flow over the boundary F_C .

\mathbb{T}_B reads, again for two example grid points i_1 and i_2

$t_{x,y}$	ν_{i_1}	ν_{i_2}	F_C
ν_{i_1}			
ν_{i_2}			
F_C	1	1	

The rows for ν_{i_1} and ν_{i_2} are zero since substitute equations will directly be provided in the boundary matrix \mathbb{T}_B .

5.5 Contact Model

The generalized boundary condition is the constitutive relation for the contact potential ψ_C and reads

$$f_{\psi_C} = \alpha \cdot \psi_C + \beta \cdot I_C + \gamma \cdot Q_C - \delta = 0 \quad (5.20)$$

where Q_C is the contact charge and $I_C = I_{n_C} + I_{p_C} + \partial Q_C / \partial t$ the contact current. It should be noted that all these quantities are solution variables, which simplifies the formulation of the contact models.

For the special case of a traditional voltage controlled contact $\alpha = 1$, $\beta = \gamma = 0$, and $\delta = V_0$ and (5.20) degenerates to

$$f_{\psi_C} = \psi_C - V_0 = 0 \quad (5.21)$$

Modeling a series contact resistance using $\beta = R_C$ one gets

$$f_{\psi_C} = \psi_C + R_C \cdot I_C - V_0 = 0 \quad (5.22)$$

For a current controlled contact $\beta = 1$, $\alpha = \gamma = 0$, and $\delta = I_0$ and (5.20) degenerates to

$$f_{\psi_C} = I_C - I_0 = 0 \quad (5.23)$$

For a charge controlled contact $\alpha = \beta = 0$, $\gamma = 1$, and $\delta = Q_0$ and (5.20) degenerates to

$$f_{\psi_C} = Q_C - Q_0 = 0 \quad (5.24)$$

Using different units for the coefficients α , β , γ , and δ (5.20) can be interpreted in different ways. These include a parallel conductance and capacitance to ground for a current controlled contact, and a series resistance and parallel capacitance to ground for a voltage controlled contact.

As (5.20) is normally not diagonal-dominant it is pre-eliminated. In (5.23) the main-diagonal entry is zero, therefore f_{I_C} must be eliminated first in order to get the derivatives with respect to ψ_C .

5.6 Contact Voltage Variable

Having a separate solution variable for the contact voltage avoids numerical problems with large arguments of the Bernoulli function B . Using a Scharfetter-Gummel discretization scheme the expression for the current between two grid points i and j reads

$$I_{ij} = C_1 \cdot (B(\Delta) \cdot n_j - B(-\Delta) \cdot n_i) \quad (5.25)$$

$$\Delta = C_2 \cdot (\psi_j - \psi_i) + C_3 \quad (5.26)$$

with C_i being material parameters. Applying the contact voltage directly to the boundary grid point could cause large arguments of B and hence numerical problems. This is avoided by having a separate variable for the contact voltage. At the beginning of the iteration procedure the constitutive relation for ψ_C is violated and will only successively be adapted which guarantees numerical stability (see Fig. 5.2).

5.7 Example

As an example a one-dimensional semiconductor structure with a simple ohmic boundary condition is considered. The index of the quantities is the number of the grid point they belong to, with 0 being the left contact. For the electron concentration $n_0 = N_0$ with N_0 being a constant value depending only on the doping of the semiconductor. To further simplify the example only the static Poisson equation and the static continuity equation for electrons are considered. Furthermore, the separate variable for the electron contact current I_{C_n} is omitted. For a one-dimensional device the segment constitutive relations read

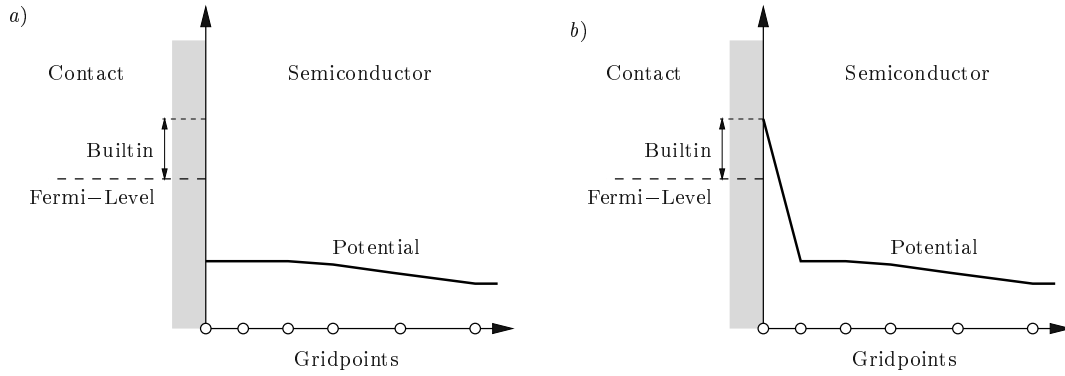


Figure 5.2: Effect of a separate potential variable on the initial-guess of the potential: a) with a separate potential variable the potential stays smooth inside the semiconductor region. b) directly applying the contact potential gives a large discontinuity of the potential.

$$\begin{aligned}
 f_{\psi_0}^S &= \Psi_{01} + q \cdot n_0 \cdot V_0 \neq 0 \\
 f_{\psi_1}^S &= \Psi_{10} + \Psi_{12} + q \cdot n_1 \cdot V_1 = 0 \\
 &\vdots \\
 f_{n_0}^S &= I_{01} \neq 0 \\
 f_{n_1}^S &= I_{10} + I_{12} = 0 \\
 &\vdots
 \end{aligned}$$

with

$$\begin{aligned}
 \Psi_{ij} &= \Psi(\psi_i, \psi_j) = -\Psi_{ji} \\
 I_{ij} &= I(n_i, n_j, \psi_i, \psi_j) = -I_{ji}
 \end{aligned}$$

At the boundary, the constitutive relations are

$$\begin{aligned}
 f_{\psi_0} &= \psi_0 - \psi_C = 0 \\
 f_{n_0} &= n_0 - N_0 = 0 \\
 f_{I_C} &= I_C + f_{n_0}^S = 0 \\
 f_{Q_C} &= Q_C + f_{\psi_0}^S = 0
 \end{aligned}$$

The boundary constitutive relations will be used to determine the quantity values at the boundary while the segment constitutive relations will be used to build up an expression for the boundary charge Q_C and for the boundary current I_C . This is achieved by the boundary models which set the appropriate entries in the transformation matrix \mathbb{T}_B which reads

$t_{x,y}$	ψ_0	ψ_1	n_0	n_1	I_C	Q_C
ψ_0						
ψ_1		1				
n_0						
n_1				1		
I_C			1			
Q_C	1					

The solution vector \mathbf{x} contains the following quantities

$$\mathbf{x} = (\psi_0, \psi_1, \dots, n_0, n_1, \dots, \psi_C, I_C, Q_C, \dots)^T$$

For voltage controlled contacts with V_0 applied to the contact one gets

$$f_{\psi_C} = \psi_C - V_0 = 0$$

When applying the current I_0 to the contact f_{ψ_C} changes to

$$f_{\psi_C} = I_C - I_0 = 0$$

The system matrix for iteration step k is

$j_{x,y}$	ψ_0	n_0	ψ_C	I_C	Q_C	\mathbf{r}
ψ_0	-1		1			$f_{\psi_0}^k$
n_0		-1				$f_{n_0}^k$
ψ_C				-1		$f_{\psi_C}^k$
I_C	$-\frac{\partial I_{01}}{\partial \psi_0}$	$-\frac{\partial I_{01}}{\partial n_0}$		-1		$f_{I_C}^k$
Q_C	$-\frac{\partial \Psi_{01}}{\partial \psi_0}$	$-q \cdot V_0$			-1	$f_{Q_C}^k$

As the constitutive relations for the quantities ψ_0 , n_0 , and I_C are eliminated first, one ends up with the following matrix

$j_{x,y}$	ψ_C	\mathbf{r}
ψ_C	$-\frac{\partial I_{01}}{\partial \psi_0}$	$f_{\psi_C}^k - f_{I_C}^k + f_{\psi_0}^k \cdot \frac{\partial I_{01}}{\partial \psi_0} + f_{n_0}^k \cdot \frac{\partial I_{01}}{\partial n_0}$

5.8 Mixed-Mode

In case of a mixed-mode simulation (5.20) is replaced by

$$f_{\psi_C} = \psi_C - \varphi_C = 0 \quad (5.27)$$

with φ_C being the node voltage of the circuit node connected to the device. This can be interpreted as a zero-valued voltage source connecting the circuit node to the device (see Fig. 5.3). The constitutive relation for φ_C follows from KCL and reads

$$f_{\varphi_C} = \sum_i I_{D_i} + I_C = 0 \quad (5.28)$$

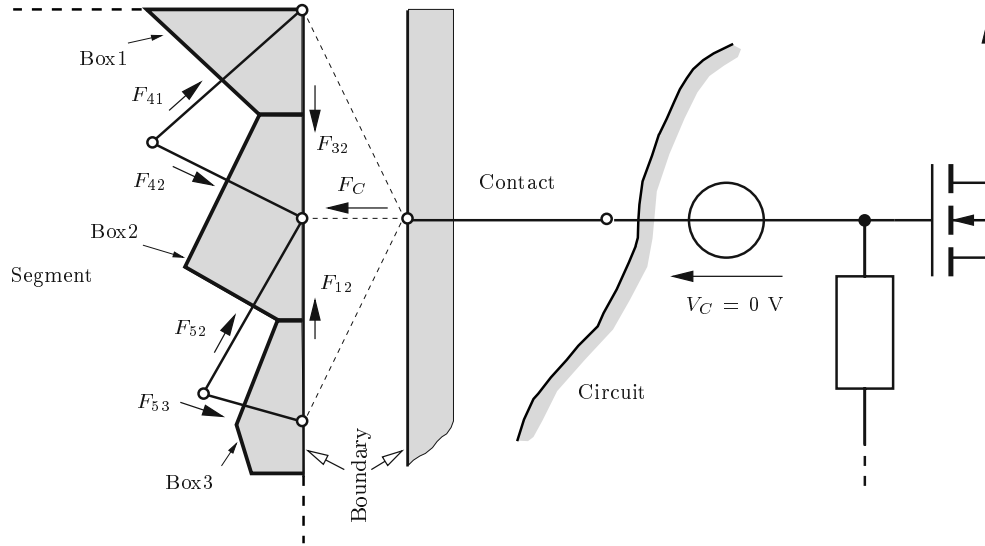


Figure 5.3: Contact handling for mixed-mode

with I_{D_i} being the currents of the compact models connected to the same node. In terms of the familiar MNA stamps

$j_{x,y}$	φ_C	ψ_C	I_C	\mathbf{r}
φ_C			-1	$f_{\varphi_C}^k$
ψ_C	1	-1		$f_{\psi_C}^k$
I_C			-1	$f_{I_C}^k$

For the same one-dimensional contact the relevant part of the system matrix reads

$j_{x,y}$	ψ_0	n_0	ψ_C	I_C	φ_C	\mathbf{r}
ψ_0	-1		1			$f_{\psi_0}^k$
n_0		-1				$f_{n_0}^k$
ψ_C			-1		1	$f_{\psi_C}^k$
I_C	$-\frac{\partial I_{01}}{\partial \psi_0}$	$-\frac{\partial I_{01}}{\partial n_0}$		-1		$f_{I_C}^k$
φ_C				-1		$f_{\varphi_C}^k$

Eliminating ψ_0 , n_0 , ψ_C , and I_C yields the desired result

$j_{x,y}$	φ_C	\mathbf{r}
φ_C	$-\frac{\partial I_{01}}{\partial \psi_0}$	$-f_{\psi_C}^k + f_{I_C}^k + f_{n_0}^k \cdot \frac{\partial I_{01}}{\partial n_0} - \frac{\partial I_{01}}{\partial \psi_0} \cdot (f_{\psi_0}^k + f_{\psi_C}^k)$

5.9 Thermal Simulation

As thermal circuit simulation is an equivalent problem to electrical circuit simulation MINIMOS-NT makes use of similar formulations. The thermal heat flow over the contact replaces the electrical current and the contact temperature the contact voltage, hence the contact condition reads

$$f_{T_C} = T_C - \vartheta_C = 0 \quad (5.29)$$

with T_C being the contact temperature and ϑ_C being the thermal equivalent to the node voltage, the node temperature.

In MINIMOS-NT two different thermal contact models are implemented. The first model implements an isothermal contact by simply setting the lattice temperature at the interface points equal to the contact temperature

$$T_L = T_C \quad . \quad (5.30)$$

The second model considers a thermal contact resistance and the thermal heat flow density S_L at the contact boundary and reads

$$S_L = n \cdot \frac{T_L - T_C}{\rho_{th}} \quad (5.31)$$

with ρ_{th} being the thermal contact resistivity. The contact conductance G_{th} is related to the thermal resistivity by $G_{th} = A/\rho_{th}$ with A as the contact area.

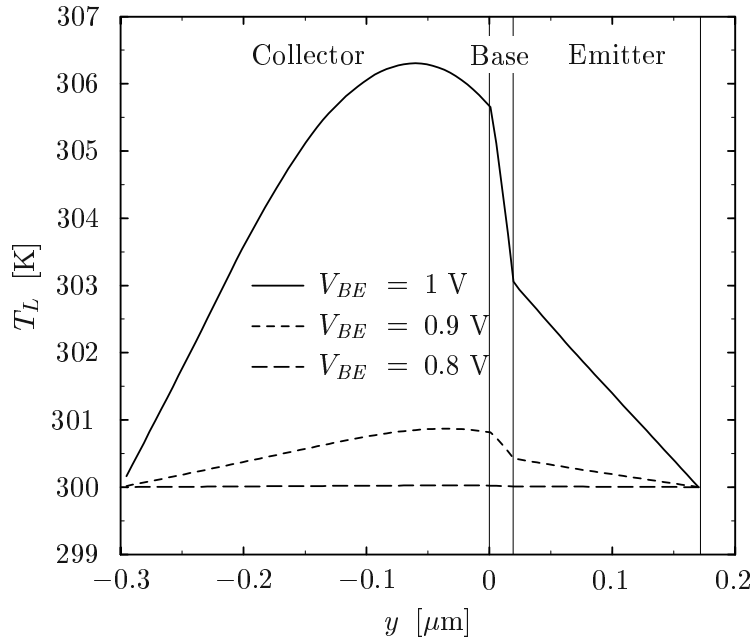


Figure 5.4: Lattice temperature distribution of a HBT with the isothermal contact model for different bias voltages.

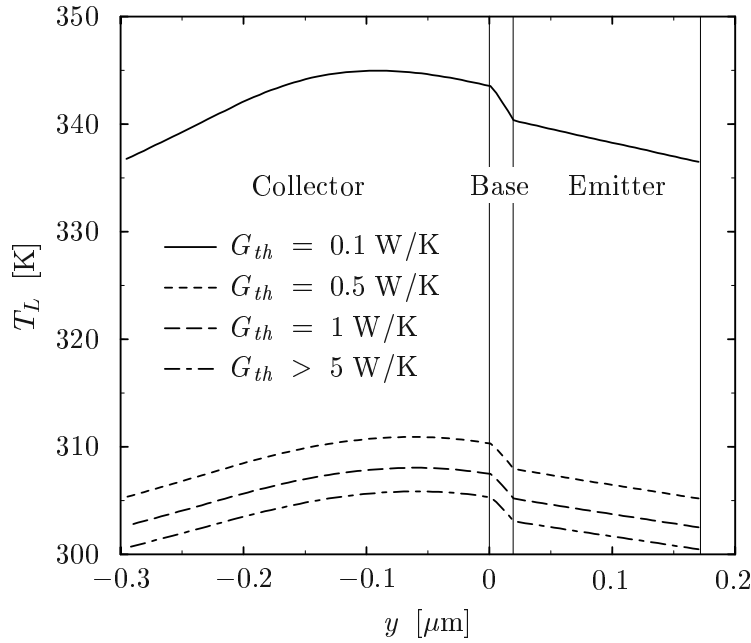


Figure 5.5: Lattice temperature distribution of a HBT for different thermal contact conductances.

The choice of the contact model has a fundamental influence on both the electrical and the thermal behavior of the device. Both models result in approximately the same temperature difference inside the device relative to the boundary values. However, the absolute temperature values can be completely different. Simulated temperature distributions inside the device for the HBT from Section 7 are shown in Fig. 5.4 and Fig. 5.5 for the isothermal and the resistance contact model, respectively, with $V_{CE} = 3.5$ V. Fig. 5.4 shows the temperature distribution for different base-emitter voltages V_{BE} whereas for Fig. 5.5 $V_{BE} = 1.0$ V was used and G_{th} was varied. For $V_{BE} = 1.0$ V both contact models generate the same relative temperature distribution but in the case of the resistance contact model the temperature is shifted by an offset which exponentially depends on G_{th} . For G_{th} as small as 10 mW/K no solution could be found at all as the lattice temperature would exceed 600 K which inhibits a successful simulation. Although $V_{BE} = 1.0$ V is quite high it must be pointed out that even for lower bias conditions the same situation occurs for improper choice of G_{th} .

These investigations show that the isothermal model can only be used when the exact contact temperatures are known. Simply assuming ambient temperature delivers completely wrong results as the simulated region of the device is normally reduced to the electrically active region which is only a small portion of the whole geometry. Hence, especially for mixed-mode simulations use of thermal contact resistances is mandatory.

In Fig. 5.6 the heat generation inside the device is shown for different values of G_{th} . As the current density remains approximately constant within this cross-section, the maximum of the heat generation is located at the base-collector space charge region where the electric field is maximal. As $V_{CE} = 3.5$ V was assumed which is quite moderate, even higher heat generation rates can be expected for the circuits simulated in Section 7. Although

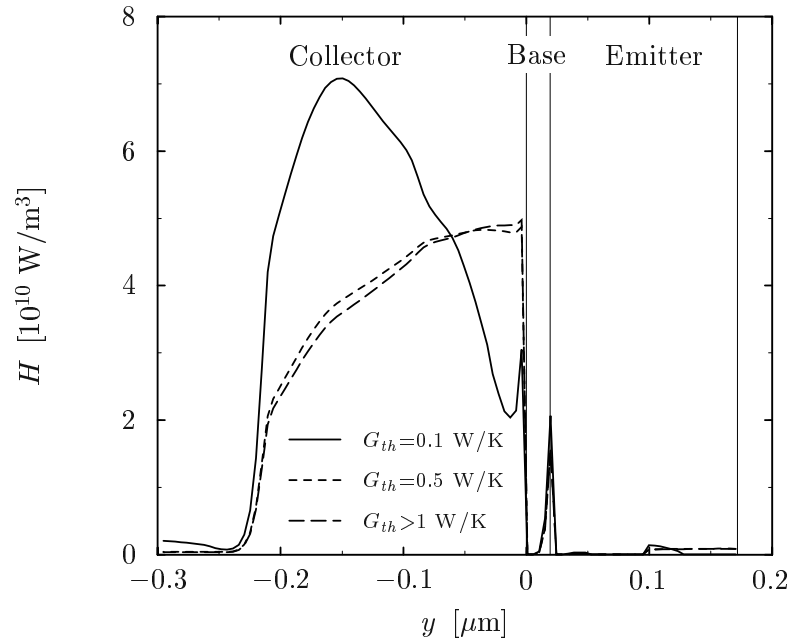


Figure 5.6: Heat generation distribution of a HBT for different thermal contact conductances.

the final values may give reasonable temperature distributions, during iteration the bias voltages of the device may vary considerably and can easily exceed $V_{BE} = 1.5$ V and $V_{CE} = 20$ V. This situation can occur during mixed-mode simulation of circuits with large supply voltages. Under these bias conditions, the thermal problem cannot be solved as the melting point of the device would be exceeded considerably. This causes excessive problems when simulating fully-coupled electro-thermal systems especially because measured values for G_{th} are in the range 1–10 mW/K.

In Fig. 5.7 the temperature distribution for different base-emitter voltages is shown generated with a quite large value of 100 mW/K for G_{th} . All these figures indicate, that the temperature difference *inside* the device is normally much smaller than the temperature difference induced by the contact model. For some simulations it might therefore be beneficial to substitute the heat flow equation by a spatially constant lattice temperature which is determined by the dissipated power of the device and thermal resistances at the contacts. For example, when the device is known to operate at 400 K, the local generated heat may be neglected.

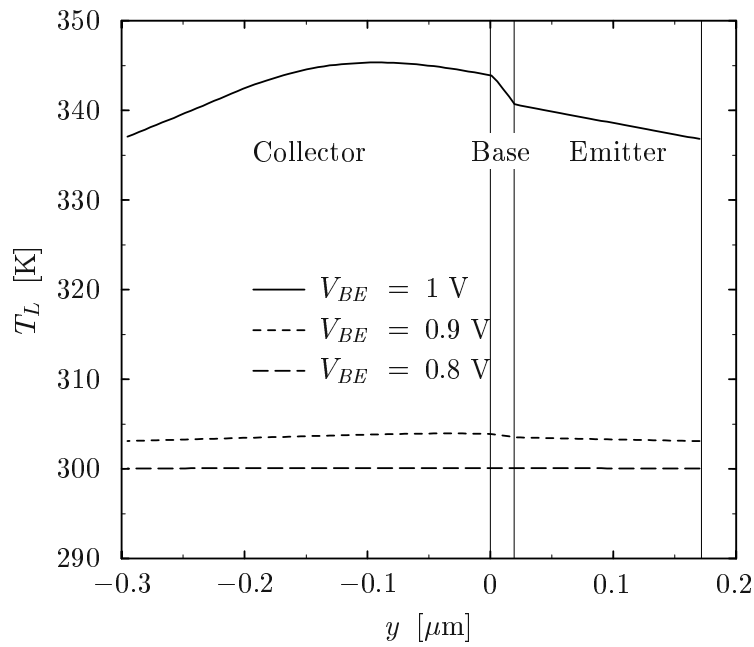


Figure 5.7: Lattice temperature distribution of a HBT with the contact resistance model for different bias voltages.

Chapter 6

Convergence and Damping

The system of equations which has to be solved for mixed-mode device simulation is non-linear and extremely sensitive to small changes in the solution variables. While the semiconductor equations are difficult to solve themselves the situation becomes even worse when using dynamic mixed-mode boundary conditions. To solve these equations the Newton method is used which is known to have quadratic convergence properties for an initial-guess sufficiently close to the final solution. However, such an initial-guess is hard to construct for both the distributed quantities inside the device and the circuit equations. Hence methods have to be found to enlarge the region of convergence to succeed even with a poor initial-guess. This is achieved by suitable damping schemes which will be discussed in the following sections. Damping schemes for the device equations will be discussed in Section 6.1 while Section 6.2 focuses on damping schemes for the circuit equations.

Especially important is a reliable method to obtain a DC operating point which is needed as a starting point for a transient analysis or a static transfer characteristic. Transient simulations are far better conditioned as the time derivatives provide main-diagonal entries and act as a natural damping. As the solution of the last timestep provides a good initial-guess it is normally possible to obtain convergence for a sufficiently small timestep. Although the conditioning of the equation system does not change for DC transfer analysis the last solution again provides a good initial-guess. In case the system fails to converge for a given step the step can normally be reduced in such a way to obtain convergence. Hence the following discussion will focus solely on DC operating point calculation.

To the best knowledge of the author no useful damping scheme for mixed-mode has been published so far. Only in [49] it was stated that the change of the node voltages was limited to a user-specified value which is in the range of $2 \cdot V_T$. This is, as pointed out in the same paper, far from being optimal as it guarantees a large number of iterations for larger supply voltages as is the case for some of the circuits simulated in this thesis. E.g., for the output stage of an OpAmp as shown in Fig. 6.8 the supply voltages are ± 15 V, hence it takes at least $15/0.05 = 300$ iterations to build up the supply voltages without even considering the effect of non-linearities. Furthermore it is stated in [49] that a solution can only be obtained for an initial-guess as close to the solution as ± 0.2 V for forward-biased junctions.

These restrictions of mixed-mode simulations seem to be generally accepted nowadays making it a challenging task to seek for alternatives. A new method is proposed in Section 6.2.5 which works admirably well for small circuits. Solutions could be found for several typical analog and digital circuits starting from the zero initial-guess for the node voltages and charge neutrality assumptions for the semiconductor devices within 20–50 iterations which is a comparable effort to SPICE which uses compact models.

6.1 Device Equation Damping Schemes

All damping schemes implemented in MINIMOS-NT have in common that they damp the solution vector by a damping factor d and hence obtain the following new solution

$$\mathbf{x}^* = \mathbf{x}^k + d \cdot (\mathbf{x}^{k+1} - \mathbf{x}^k) \quad (6.1)$$

to replace the undamped new solution \mathbf{x}^{k+1} . The computation of the damping factor d depends on the damping scheme selected. Investigations were made on several damping schemes and potential damping was found to deliver most reliable results [15].

$$d = \frac{1 + \delta \cdot \ln \frac{\|\mathbf{u}_\psi\|}{V_T}}{1 + \delta \cdot \left(\frac{\|\mathbf{u}_\psi\|}{V_T} - 1 \right)} \quad \text{with } 0 \leq \delta. \quad (6.2)$$

with δ being an adjustable parameter of the damping scheme, \mathbf{u}_ψ the update norm of the potential sub-vector, and V_T the thermal voltage. A larger δ results in more logarithm-like damping of the updates.

6.2 Circuit Equation Damping Schemes

The following sections give a short review of the damping schemes for the circuit equations which emerged during the last decades. They have been used in many different circuit simulators either stand-alone or in combination.

6.2.1 Source Stepping

In essence the method of source stepping is equivalent to determining the DC operating point with a DC transfer curve. All independent sources are not directly applied to the circuit but damped with a value α which is between zero and unity. After convergence for a distinct α is obtained, α is increased by a factor depending on the iteration count of the last solution until convergence is obtained. In case no convergence is obtained within a distinct number of iterations, α is decreased until convergence is obtained again. As pointed out in [43] this method needs an excessively large number of iterations for conventional circuit simulation. For good reasons, the same behavior can be expected for mixed-mode simulation. The method can be improved by making use of a predictor which calculates an initial-guess out of previous solutions [43].

6.2.2 Pseudo-Transient Method

The method applied by e.g., ASTAP [33, 73] is called the pseudo-transient method and can be considered a variant of the source stepping algorithm. Instead of the original circuit a transient simulation on a “pseudo” circuit is performed. A pseudo-inductor is inserted in series with each independent voltage source and each non-linear voltage-defined branch, and a pseudo-capacitor is inserted in parallel with each independent current source and each non-linear current-defined branch. The initial conditions for these pseudo-reactances are chosen such that the initial transient solution is zero.

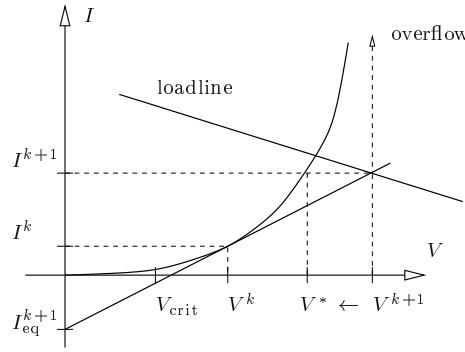


Figure 6.1: Horizontal and vertical projection of the current solution for a diode.

A transient analysis is performed upon this pseudo-circuit. As the transition from the zero solution to the final solution is of no interest in this analysis, the truncation error can be ignored as long as the solution converges to the correct equilibrium solution. For this reason, the timestep chosen is not determined by accuracy considerations; instead, the timestep is taken as large as possible, consistent only with the convergence of the Newton algorithm [43].

6.2.3 Global Damping Strategy

As for the device equations global damping strategies have been used for circuit equations. E.g., for JANAP [6] the damping scheme after Bank and Rose has been used. In numerical experiments it was tried to use global damping schemes for the fully coupled mixed-mode equation system. As the non-linearities are so strong, the damping for the node voltages was found to be so restrictive that they were only updated with a few milli-Volt or less.

6.2.4 Local Limiting

In conventional circuit simulation it is common practice to limit the contact voltages before evaluating the compact model. This is illustrated for a simple diode model represented by the equation

$$I = I_S \cdot \left(\exp \left(\frac{V}{V_T} \right) - 1 \right) \quad (6.3)$$

with I_S being the saturation current and $V_T = k_B \cdot T/q$ the thermal voltage. At the Newton iteration step k equation (6.3) is linearized around the current solution point V^k . The solution of the linearized system is found as V^{k+1} . However, directly using V^{k+1} (vertical projection) can easily lead to overflow of the exponential function in (6.3) as shown in Fig. 6.1. To overcome this problem, horizontal projection has been introduced. The appropriate expressions are easily derived using Fig. 6.1. The current change evaluates to

$$I^{k+1} - I^k = \left(V^{k+1} - V^k \right) \cdot \left. \frac{dI}{dV} \right|_{V=V^k} . \quad (6.4)$$

Inserting (6.3) gives

$$\exp \frac{V^* - V^k}{V_T} = 1 + \frac{\Delta V}{V_T} \quad (6.5)$$

$$\Delta V = V^{k+1} - V^k \quad (6.6)$$

and finally

$$V^* = V^k + V_T \cdot \ln \left(1 + \frac{\Delta V}{V_T} \right) . \quad (6.7)$$

However, horizontal projection is only useful for $V^{k+1} > V_{\text{crit}}$. Below V_{crit} vertical projection obviously delivers better results. Equation (6.7) can be generalized to arbitrary non-linearities [32]

$$V^* = V^k + \frac{\text{sign}(\Delta V)}{k} \cdot \ln(1 + k \cdot |\Delta V|) \quad (6.8)$$

with k being a new scaling factor which can be optimized for each single device in the course of iteration progress. A larger k gives stronger damping and hence increases reliability of the algorithm for the price of longer simulation times. It must be pointed out that this mixed method of updating is not amenable to an easy description in terms of matrices, Jacobians, etc., and hence convergence theorems for this method are impossible to formulate. However, actual experience with this procedure in handling all types of circuits during the last decades has proven its effectiveness. As the voltages are damped in an individual manner for each device this damping algorithm will be termed *local* in the following. When investigating (6.3) one notices that $\partial I / \partial V$ can become very small for $V \ll 0$ which could result in a singular matrix. To prevent this occurrence a small leakage conductance G_{\min} of typically 10^{-9} – 10^{-12} S is placed in shunt with each junction. Alternatively G_{\min} can be connected between each device node and ground which is similar from an electrical point of view but improves diagonal dominance compared to a shunt conductance. In addition, the exponential characteristic is replaced by a linear characteristic for bias voltages smaller than a few V_T . Although the above mentioned precautions are guided by numerical reasoning it should be noticed that they do not have any negative impact on the value of the solution as they might as well be justified on physical grounds.

6.2.5 The New Method

Damping of the contact voltages in general-purpose device simulation is different in two aspects. Firstly, arbitrary devices with arbitrary characteristics and an arbitrary number of nodes can be simulated. Secondly, for compact models only potential differences are used ($V = \varphi_A - \varphi_C$) whereas the contact models in device simulation normally use absolute potential values. This implies that a DC offset which does not change anything about the solution will waste computation time as it needs many iterations to build up the proper potential distribution inside the device. This is due to the fact, that the potential is initialized to the so-called built-in potential which evaluates to [15]

$$\begin{aligned} \psi_{\text{bi}} &= \frac{E_C}{q} + V_T \cdot \ln \left(\frac{1}{2 \cdot N_C} \cdot \left(N_T + \sqrt{N_T^2 + 4 \cdot n_i^2} \right) \right) & N_T > 0 \\ &= \frac{E_V}{q} - V_T \cdot \ln \left(\frac{1}{2 \cdot N_V} \cdot \left(-N_T + \sqrt{N_T^2 + 4 \cdot n_i^2} \right) \right) & N_T < 0 \end{aligned} \quad (6.9)$$

with N_T being the net dopant concentration. For $N_T > 0$ the first version and for $N_T < 0$ the second version of (6.9) should be used to avoid cancellation errors for large absolute values of N_T . (6.9) is an excellent guess for the potential in non-depletion regions when all contact voltages are zero. However, this initial-guess could be improved by adding the average of the contact voltages. Unfortunately this cannot be done for a mixed-mode simulation as the contact voltages evolve during iteration and hence are not known in advance.

To make use of the damping strategy (6.8) device nodes were grouped in pairs using available information about the device (diode, bipolar junction transistor or MOS transistor). Then the contact voltages were damped using (6.8). However, the solution of the semiconductor equations is damped using a global damping strategy

$$\mathbf{x}^* = \mathbf{x}^k + d \cdot \mathbf{u} \quad (6.10)$$

with a global damping factor d which applies to *all* solution variables in the same way. An important feature of (6.10) is that the direction of the update does not change which is not the case when applying (6.8). In experiments it was tried to limit the node voltages using (6.8) whereas for the rest of the solution vector (6.10) was applied. As the node voltages are directly coupled to the contact voltages by (5.27) and the contact voltages determine the potential inside the device this caused inconsistencies which lead to strong oscillations of the solution variables. Hence, further investigations of this mixed damping procedure were skipped.

A circuit revealing the problems caused by DC offsets is shown in Fig. 6.2. Here $V_1 = V_D + V_{DC}$ and $V_2 = V_{DC}$ with $V_D = 1$ V. First, the circuit is simulated using the *direct boundary condition* (DBC) given in (5.27). The evolution of the node voltages and the device contact voltages during iteration for $V_{DC} = 0$ V is shown in Fig. 6.3. Until convergence 10 iterations are needed. However, when setting $V_{DC} = 10$ V convergence properties deteriorate (35 iterations) as shown in Fig. 6.4 since it takes many iterations to build up the high potential inside the diode. As for the device operation only potential differences are relevant, a modified boundary condition can be formulated. Using one of the device terminal voltages as reference voltage the boundary condition (5.27) can be reformulated to yield

$$f_{\psi_C} = \psi_C - (\varphi_C - \varphi_{Ref}) = 0 \quad (6.11)$$

for a general node and

$$f_{\psi_{Ref}} = \psi_{Ref} = 0 \quad (6.12)$$

for the reference node. The MNA stamp for a general node reads

$j_{x,y}$	φ_C	φ_{Ref}	ψ_C	I_C	\mathbf{r}
φ_C				-1	$f_{\varphi_C}^k$
ψ_C	1	-1	-1		$f_{\psi_C}^k$
I_C				-1	$f_{I_C}^k$

whereas it simplifies to

$j_{x,y}$	ψ_C	I_C	\mathbf{r}
φ_C		-1	$f_{\varphi_C}^k$
ψ_C	1		
I_C		-1	$f_{I_C}^k$

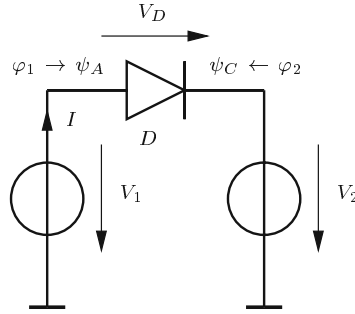


Figure 6.2: Problematic constellation when using DBC.

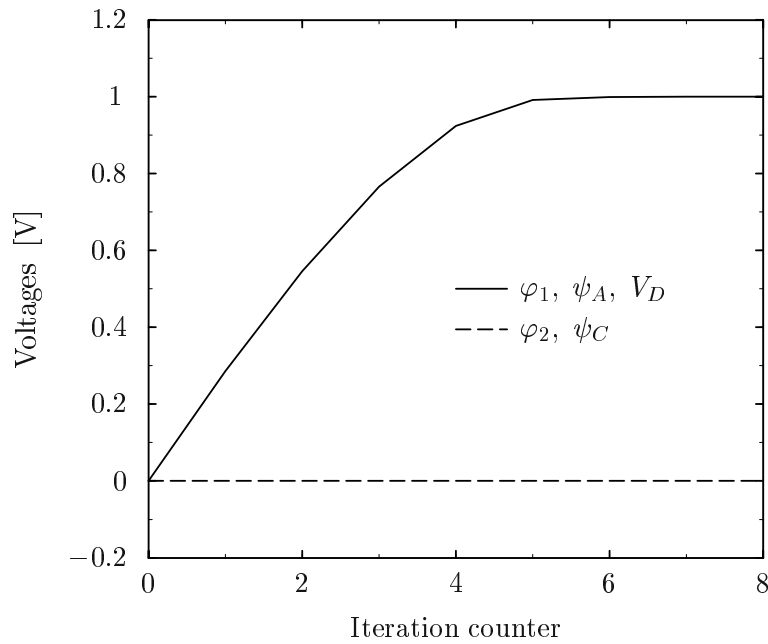


Figure 6.3: Evolution of the node and contact voltages during iteration using DBC with $V_{DC} = 0$ V. Until convergence 10 iterations are needed.

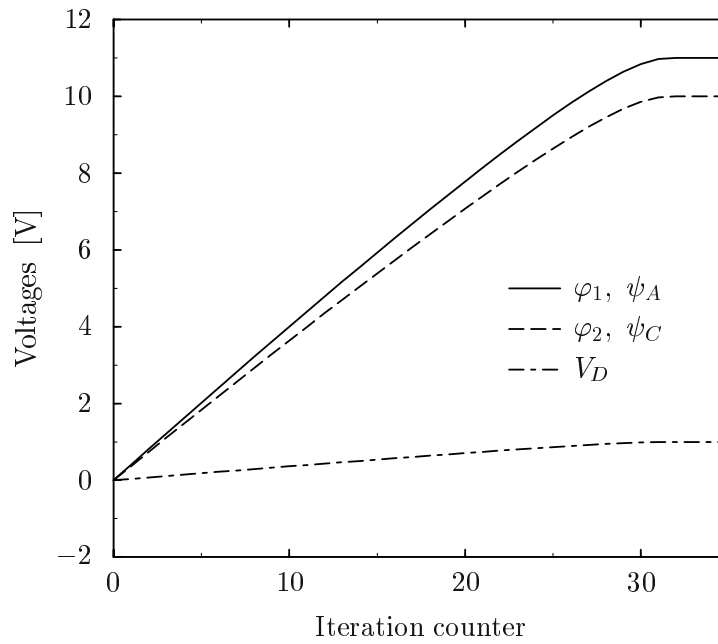


Figure 6.4: Evolution of the node and contact voltages during iteration using DBC with $V_{DC} = 10$ V. Until convergence 35 iterations are needed.

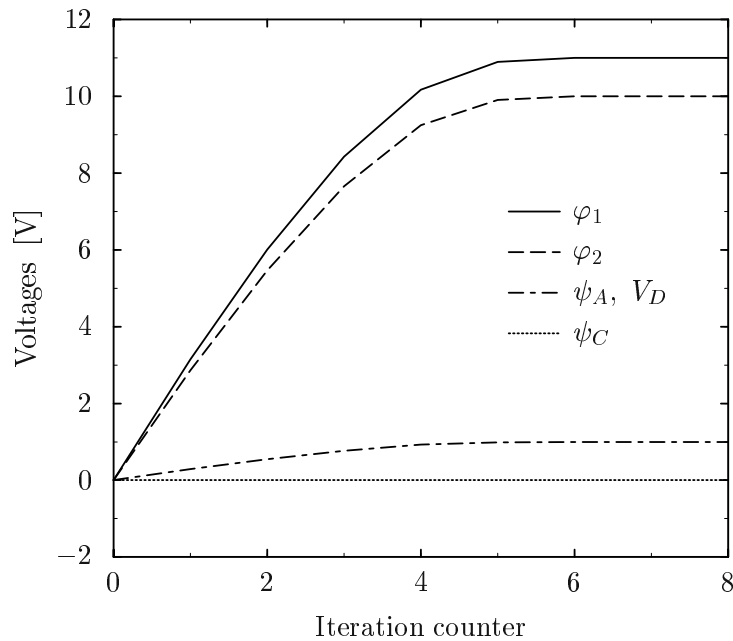


Figure 6.5: Evolution of the node and contact voltages during iteration using RBC with $V_{DC} = 10$ V. As for $V_{DC} = 0$ V, 10 iterations are needed until convergence.

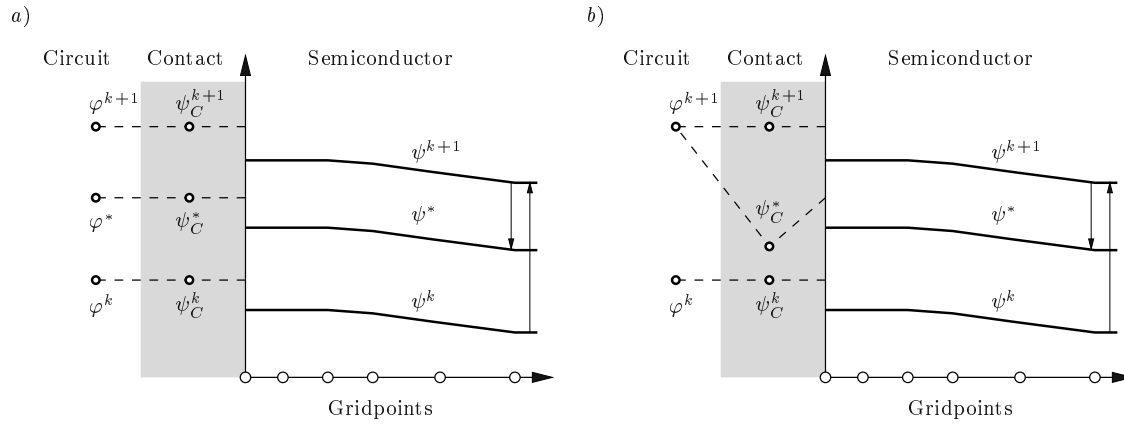


Figure 6.6: Effect of global and local damping on the solution variables: a) global damping b) local damping.

for the reference node. The simulation results using this *reference boundary condition* (RBC) are shown in Fig. 6.5. As for $V_{DC} = 0$ V, 10 iterations are needed until convergence. However, an imminent problem of this approach is that the boundary condition obtained for the reference node shows no dependence on the node voltage φ_C . This means that when f_{I_C} is pre-eliminated the main-diagonal will be zero for f_{φ_C} resulting in a singular equation system if the contact node is not connected to other devices providing main-diagonal entries. In addition, the choice of reference node is crucial and depends on the current operating condition of the device. It was found to be more useful to take the average of the node voltages

$$\bar{\varphi} = \frac{1}{n_C} \cdot \sum_C \varphi_C \quad (6.13)$$

as reference voltage with n_C being the number of contact nodes. This type of boundary condition will be referred to as *average boundary condition* (ABC). The MNA stamp for a general node reads

$j_{x,y}$	φ_C	$\bar{\varphi}$	ψ_C	I_C	r
φ_C				-1	$f_{\varphi_C}^k$
ψ_C	1	-1	-1		$f_{\psi_C}^k$
$\bar{\varphi}$	$-\frac{1}{n_C}$	$\frac{1}{n_C}$			$f_{\bar{\varphi}}^k$
I_C				-1	$f_{I_C}^k$

It is to note that the row $f_{\bar{\varphi}}$ is entered for each contact node, hence one obtains $\sum_C \frac{1}{n_C} = 1$. The convergence properties using ABC for the diode circuit Fig. 6.2 are similar to RBC as shown in Fig. 6.5. However, as $\bar{\varphi} = 0.5$ V the internal contact voltages are ± 0.5 V the built-in potential provides a better initial-guess and only 7 iterations are needed.

It has been observed that the full coupled system of device and circuit equations is extremely instable at the beginning of the iteration. Similar observations were made by Ho *et al.* [32] for FET circuits using compact models. They proposed to shunt a resistor of 3 k Ω

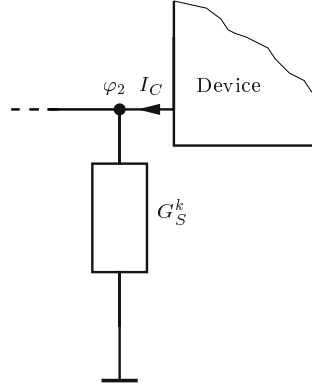


Figure 6.7: Placement of the iteration dependent conductance G_S^k for one terminal.

at the source and drain during the first three Newton iterations, to stabilize the coupled system and to slightly decouple the device from the circuit equations. This approach has been extended by introducing an iteration dependent conductance G_S^k between each device node and ground as shown in Fig. 6.7. The following purely empirical expression for G_S^k delivered very promising results

$$G_0 = 10^{-2} S \quad (6.14)$$

$$G_{min} = 10^{-12} S \quad (6.15)$$

$$G_S^k = \begin{cases} \max(G_{min}, G_0 \cdot 10^{-k/\kappa}) & \text{for } E_2(\mathbf{u}_\psi) > 0.1 \cdot V_T \\ G_{min} & \text{for } E_2(\mathbf{u}_\psi) \leq 0.1 \cdot V_T \end{cases} \quad (6.16)$$

$$\kappa = 1.0 \dots 4.0 \quad (6.17)$$

with k being the iteration counter. It is worthwhile to note that the algorithm worked equally well with $G_{min} = 0$ for the simulated circuits. However, this expression is purely empirical but unfortunately any attempt to use a more rigorous expression based on norms of the quantities did not work satisfactory.

6.3 Examples

The following small examples are aimed at proving the usefulness of one or the other approach. They have been selected for their simplicity and for their ability to represent typical or critical situations occurring in mixed-mode device simulations.

Interesting is a comparison of the convergence properties to the ideal case when the solution is used as an initial-guess for the node voltages and only the device equations need to be solved. With k_0 being the number of iterations in this ideal case one can define a measure for the convergence by

$$K = \frac{k_{MM}}{k_0} \quad (6.18)$$

with k_{MM} being the number of iterations for real mixed-mode simulation with vanishing initial-guess ($\mathbf{x} = 0$). K will be called the degradation factor in the following.

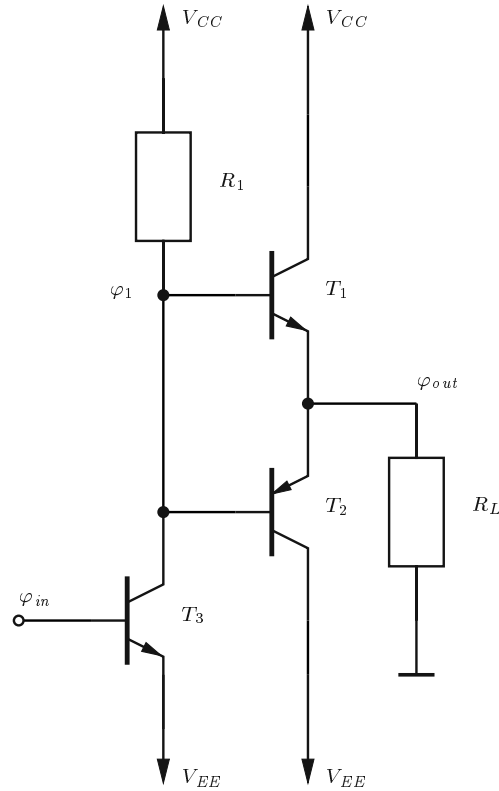


Figure 6.8: Simplified output stage of the μA709 operational amplifier with $V_{CC} = 15\text{ V}$, $V_{EE} = -15\text{ V}$, $R_1 = 20\text{ k}\Omega$, and $R_L = 500\text{ }\Omega$.

Mayaram and Pederson [39] stated that their simple circuits containing normally only one transistor converged within 8–9 iterations using the modified two-level Newton algorithm. As for each Newton step the device simulator has to be called which performs a Newton iteration itself, the total number of iterations is much larger. From a computational point of view the circuit iterations can be neglected. Assuming that the device simulator converges within 20 iterations for the first call and within 10 iterations for all consecutive calls gives a total of approximately 100 device simulator iterations. As this estimate is optimistic one can assume the modified full-Newton approach to be more efficient whenever convergence is obtained within less than 100 iterations since one iteration of the device simulator is approximately as expensive as one iteration of the complete equation system.

6.3.1 Output Stage of an OpAmp

A simplified schematic of the μA709 operational amplifier output stage is shown in Fig. 6.8. Transistor T_3 acts as a common-emitter driver stage for the complementary output devices T_1 and T_2 . Problems are caused by the large bias voltages of $\pm 15\text{ V}$ and by the sensitivity of the circuit to the state of transistor T_3 . The operating point is calculated for $\varphi_{in} = -14.5\text{ V}$ and from this point φ_{in} is stepped until -14.1 V . Within this interval the internal state of the circuit changes completely as φ_1 moves from $\approx 13.6\text{ V}$ down to $\approx -15\text{ V}$. With φ_1 the states of the two output transistors change.

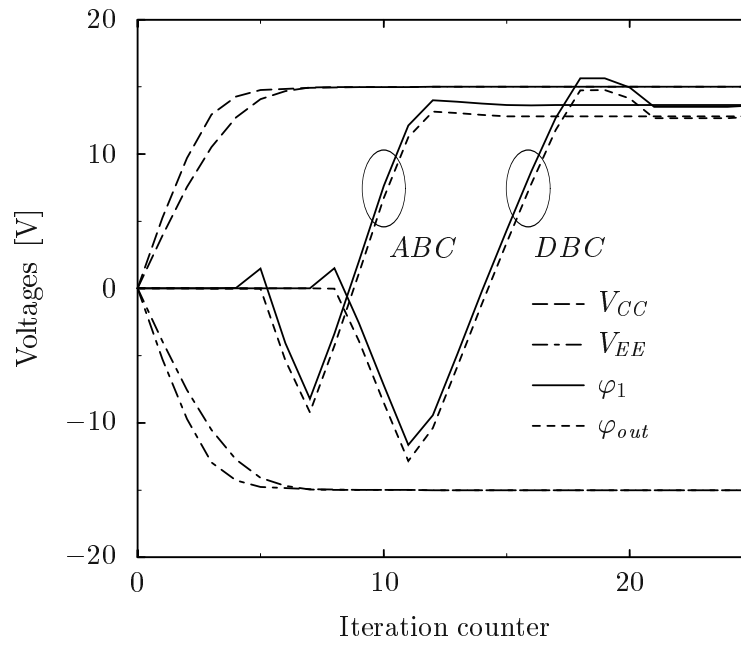


Figure 6.9: Evolution of the node voltages during DC operating point calculation for the OpAmp output stage with $\kappa = 4$.

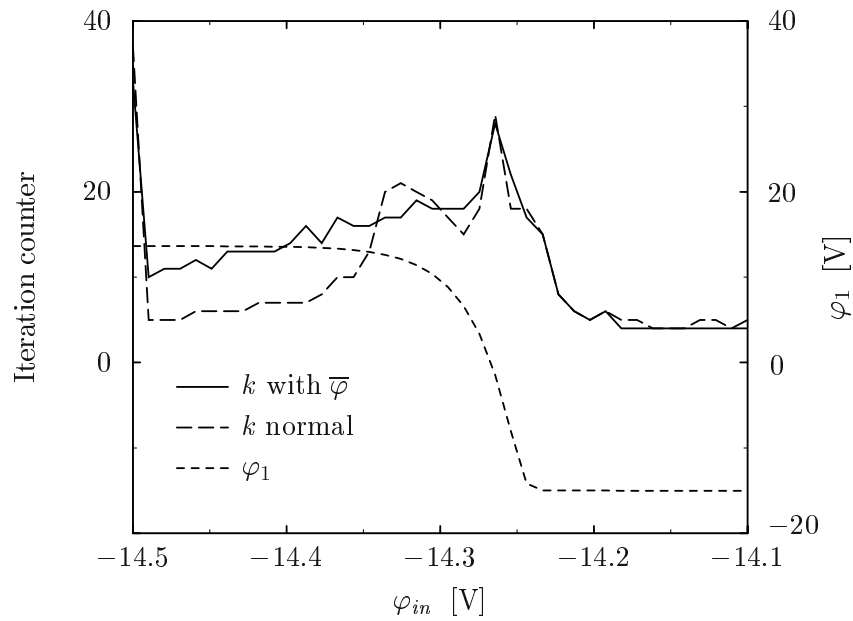


Figure 6.10: Comparison of the iteration counters for a DC transfer characteristic for the OpAmp output stage with $\kappa = 4$.

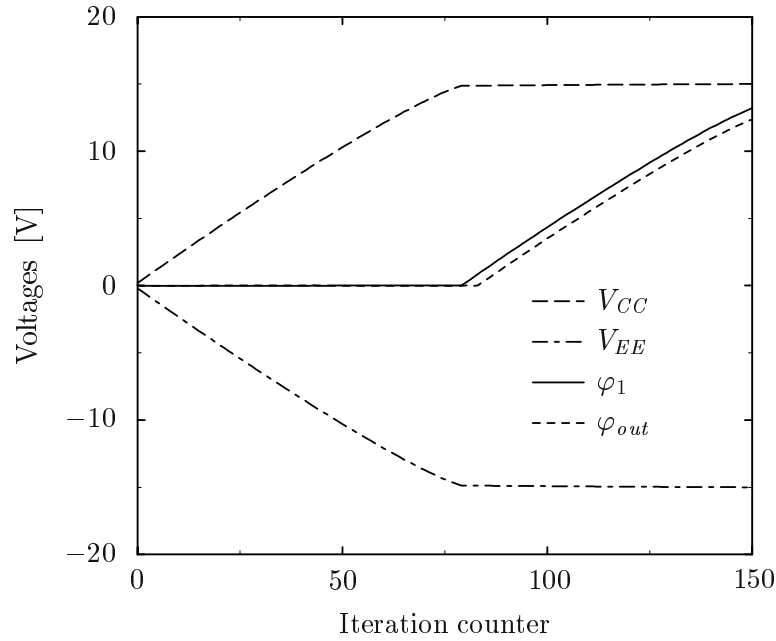


Figure 6.11: Evolution of the node voltages during DC operating point calculation for the OpAmp output stage with constant $G_S^k = 1/G_{min}$.

The evolution of the node voltages during DC operating point calculation is shown in Fig. 6.9. Best results were obtained with $\kappa = 4$. With ABC 34 iterations were needed and $K^A = 34/30 = 1.13$ whereas with DBC 38 iterations were needed and $K^D = 38/31 = 1.226$.

For the DC transfer characteristic the required number of iterations is shown in Fig. 6.10. In addition φ_1 is shown to represent the internal state of the circuit. In this case DBC is superior compared to ABC.

To demonstrate the importance of G_S^k the evolution of the node voltages for a constant $G_S^k = 1/G_{min}$ is shown in Fig. 6.11. To obtain convergence at all, the global damping parameter δ had to be increased by a factor of 100 and 170 iterations were necessary.

Although ABC required less iterations than DBC this type of boundary condition seems to have a negative impact on the condition of the system matrix as each contact voltage depends on the node voltages of all other contacts. Since an iterative solver is used which is very sensitive to the condition of the matrix the total simulation time is unfortunately larger than for DBC.

6.3.2 CML Inverter

The schematic of a current mode logic (CML) inverter is shown in Fig. 6.12a. The evolution of the node voltages during DC operating point calculation is shown in Fig. 6.13. Best results were obtained with $\kappa = 3$. With ABC 20 iterations were needed whereas 24 with DBC.

For the other points following in the DC transfer characteristic the required number of iter-

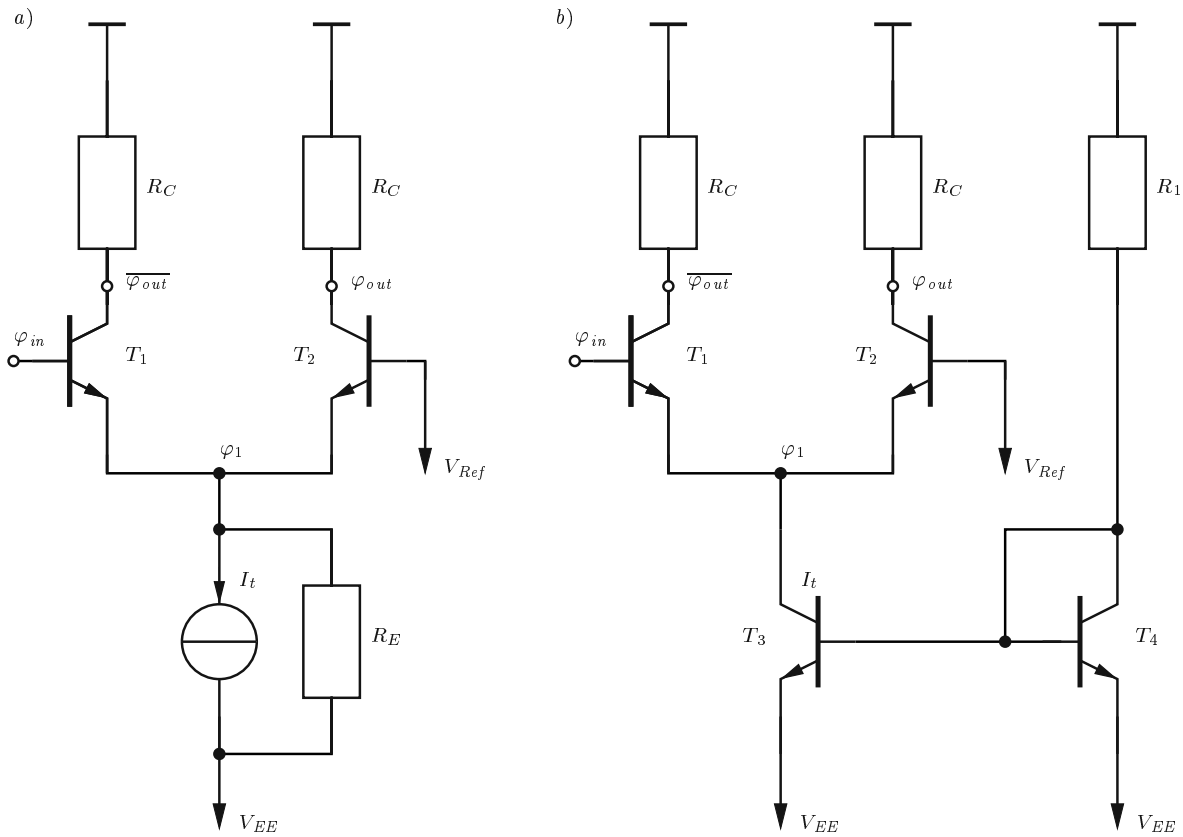


Figure 6.12: CML inverter with $V_{EE} = -5.2$ V. a) with simple current source b) and with current mirror.

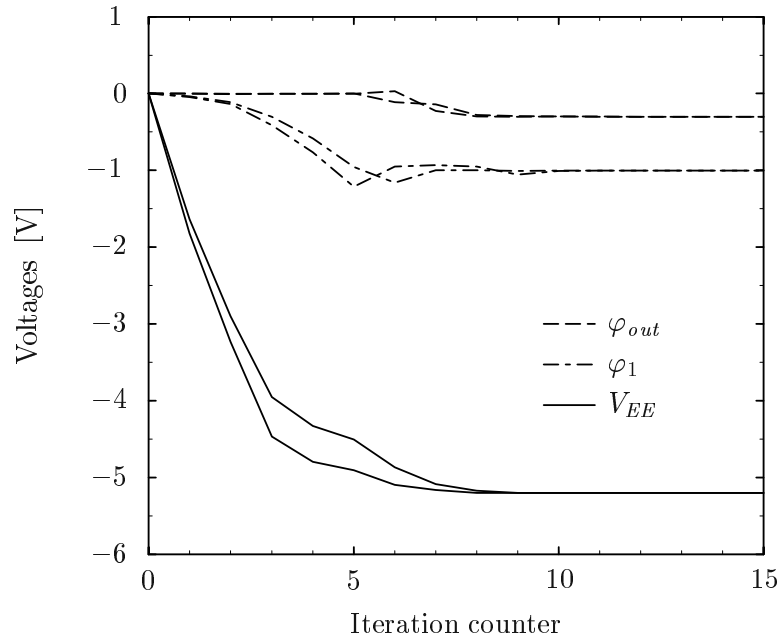


Figure 6.13: Evolution of the node voltages during DC operating point calculation for the CML inverter with $\kappa = 3$.

ations is exactly equal (12–18) and hence not shown. The simulated transfer characteristic is shown in Fig. 6.14.

For ABC one gets $K^A = 20/19 = 1.052$ and for DBC $K^D = 24/21 = 1.1428$ which both can be considered close to optimum.

In Fig. 6.12b the current source is realized as a current mirror. The number of unknowns increases to about 10000 and 48 iterations were needed using DBC and $\kappa = 3$. No convergence could be obtained for ABC.

6.3.3 CMOS Inverter

The schematic of a CMOS inverter is shown in Fig. 6.15. The evolution of the node voltages during DC operating point calculation is shown in Fig. 6.16. Best results were obtained with $\kappa = 1$. With ABC 21 iterations were needed whereas with DBC 25 iterations were needed.

For ABC one gets $K^A = 21/24 = 0.875$ and for DBC $K^D = 25/27 = 0.926$, both values being even better than the “ideal” simulation. Again, as for the CML inverter, each point of the DC transfer characteristic required the same number of iterations for both boundary conditions (7–17).

For this simple circuit convergence can be obtained without any G_S at a convergence rate similar to k_0 .

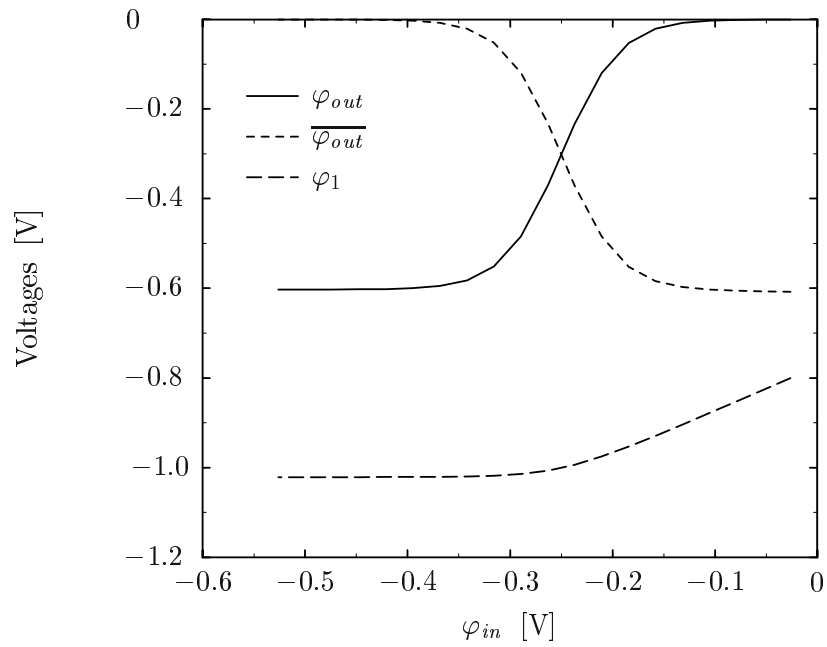


Figure 6.14: Simulated DC transfer characteristic for the CML inverter.

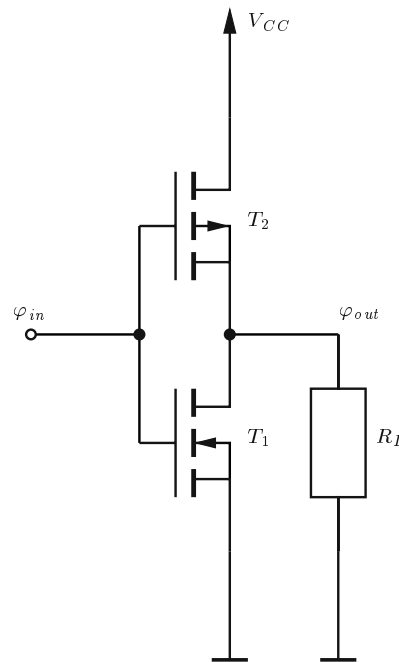


Figure 6.15: CMOS inverter with $V_{CC} = 1.2$ V.

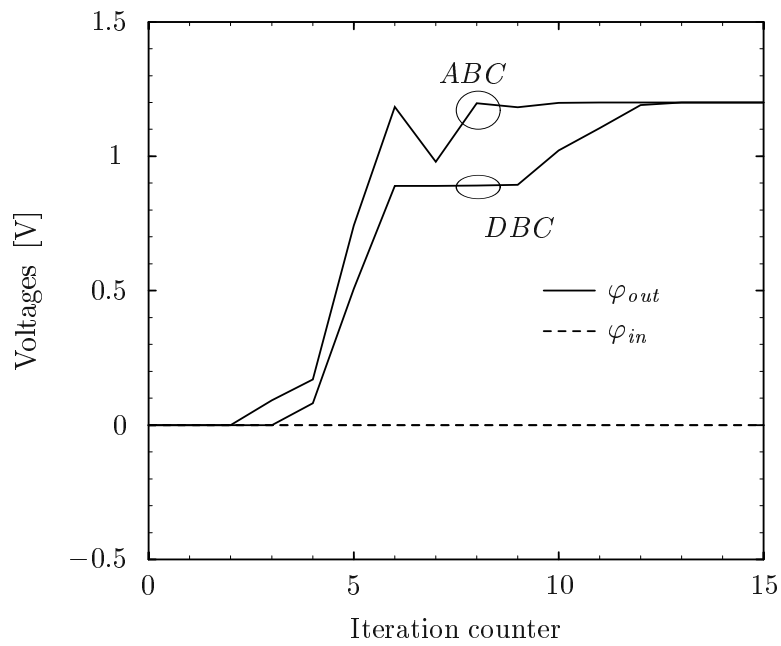


Figure 6.16: Evolution of the node voltages during DC operating point calculation for the CMOS inverter with $\kappa = 1$.

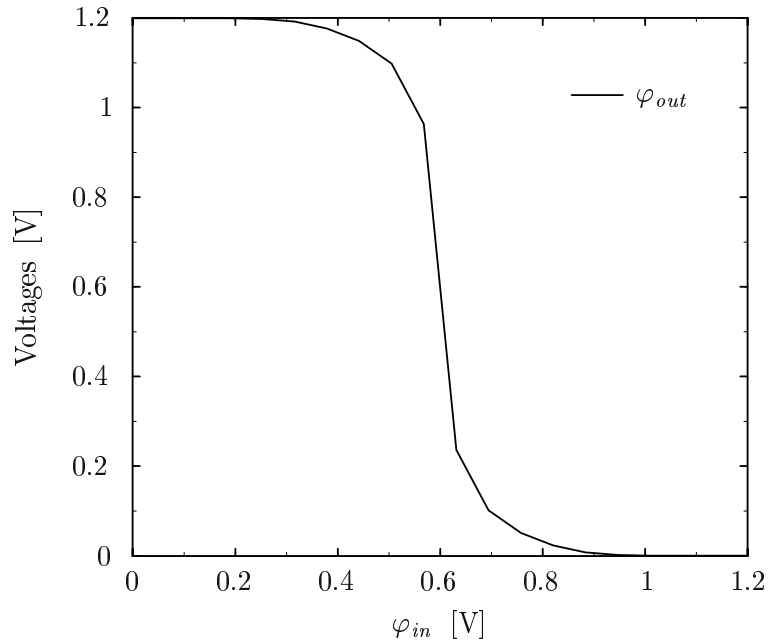


Figure 6.17: Simulated DC transfer characteristic for the CMOS inverter.

Chapter 7

Simulation Results

On the following pages simulation results for typical digital and analog circuits will be presented. These circuits have been selected to demonstrate typical problems occurring during mixed-mode device simulation. Most of these problems are related to high bias voltages and mutually coupled devices. For circuits containing only one distributed device, normally no convergence problems are encountered compared to simulations of the same device in single-mode. Except for the last circuit (μ A709 OpAmp), operating points could be easily found using the new shunt conductance technique as described in Section 6.2.5. However, to obtain a solution for HD and self-heating simulations iteration schemes had to be used to successively refine the solution starting from an initial DD solution.

7.1 Devices

7.1.1 MOS Transistors

The MOS transistors taken for the following simulation have been thoroughly discussed in Section 3.3.3 and Section 3.3.4. Both the long- and short-channel devices have been used to demonstrate the impact of non-local effects.

7.1.2 Silicon-Germanium HBT

As a bipolar device the SiGe heterojunction bipolar transistor (HBT) as shown in Fig. 7.1 was chosen which was published in [44]. It has a constant Ge doping in the base with a Ge mole fraction x of 0.2. The doping profile is shown in Fig. 7.2. The main advantage of this device design is the high emitter efficiency which can be obtained. This is due to energy gap difference between the emitter and base materials. In traditional homojunction transistors, the ratio of the doping concentrations in the emitter and base must be very high. This layout allows for a very high base doping to reduce the base resistance and a very thin base to reduce base transit time. Furthermore, the emitter may be lightly doped to reduce the base-emitter junction capacity without degrading performance. The current gain β can easily be adjusted by modifying the Ge mole fraction x . This has an important impact on the numerical behavior of the circuits as the current gain determines the coupling of the single stages of the circuits.

The mobility model used is very similar to the model discussed in Chapter 3. However,

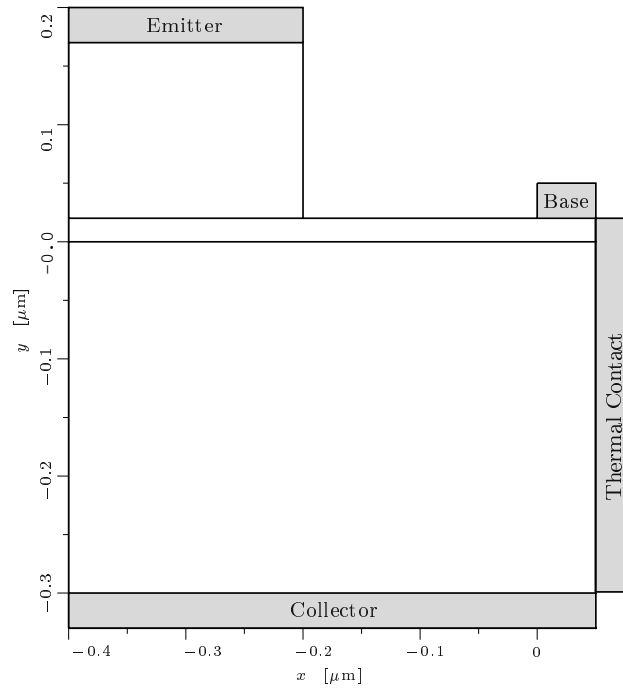


Figure 7.1: Geometry of the HBT.

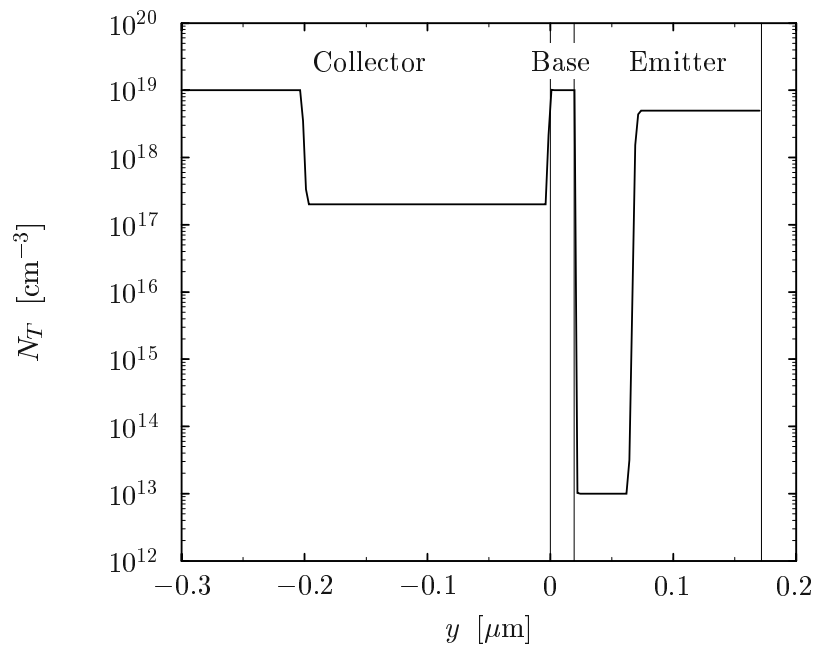


Figure 7.2: Net doping profile of the HBT.

to account for different Ge contents, it makes use of two different parameter sets, one for Si and one for Ge to calculate the two low-field mobilities μ^{Si} and μ^{Ge} . These low-field mobilities are combined by a harmonic mean

$$\frac{1}{\mu^{\text{SiGe}}} = \frac{1-x}{\mu^{\text{Si}}} + \frac{x}{\mu^{\text{Ge}}} + \frac{(1-x) \cdot x}{C} \quad (7.1)$$

with C being the bowing parameter.

For the high-field mobility model the saturation velocities of the basic materials are interpolated quadratically

$$v_{\nu}^{\text{SiGe}} = v_{\nu}^{\text{Si}} \cdot (1-x) + v_{\nu}^{\text{Ge}} \cdot x + C \cdot (1-x) \cdot x \quad (7.2)$$

The most important effect when comparing DD to HD simulations is the velocity overshoot occurring in the base-collector space charge region. This is due to a maximum of the electric field which occurs in the base-collector junction for normal forward operation of the transistor as shown in Fig. 7.3. No difference in the electric field was observed between both transport models. However, the velocity shown in Fig. 7.4 for different base-collector voltages V_{BC} and $V_{BE} = 0.75$ V shows a significant overshoot for the HD model. The velocities obtained with the DD model v_n are strictly limited by the saturation velocity $v^{\text{sat}} = 10^7$ cm/s. The overshoot occurs both in space and in time, resulting in an increased speed of the devices. A plot of the simulated electron temperature is shown in Fig. 7.5. The maximum of the temperature is shifted into the collector region as the carrier temperature cannot follow the high-field immediately. Velocity overshoot occurs in this region where the electric field is high and the temperature is only beginning to rise.

For holes similar velocity and temperature distributions were obtained. However, as the hole concentration in the collector is extremely low this had no impact on the solution. Hence, in all HD mixed-mode simulations the hole temperature T_p was set to T_L which is equivalent to performing a DD simulation without considering velocity saturation in the mobility model. This approach has two benefits. First, the matrix size is reduced by 1/5 and secondly because calculating T_p deteriorates the convergence rate of the whole system. Convergence could only be obtained by applying a three step iteration scheme: in the first block the DD equations were solved, followed by the HD equations without considering the hole temperature in the second block and only in the final block it was possible to add the hole temperature to the equation system. This final block took as many iterations as the first two blocks and only marginally changed the solution well below the accuracy of the linear solver.

The following simulations served to judge the influence of the thermal contact conductances when performing mixed-mode self-heating simulations. As a thermal contact an isothermal silicon block was added at the right side of the device. In Fig. 7.6 the temperature of this thermal contact is shown in dependence of the thermal contact conductance G_{th} . For a moderate $V_{CE} = 2$ V the contact temperature is very sensitive to the contact conductance while the collector and base current change only marginally as shown in Fig. 7.7. Hence, the current gain does not change noticeably for different values of G_{th} as shown in Fig. 7.7. This must not be confused with the strong contact temperature dependence of the current gain as shown in Fig. 7.9. This can occur when the contact temperature is determined by another device. This issue is of importance when simulating thermal coupling of devices.

The final simulation aimed to determine the self-heating time constant of the HBT. This was done by calculating the operating point without self-heating and turning self-heating

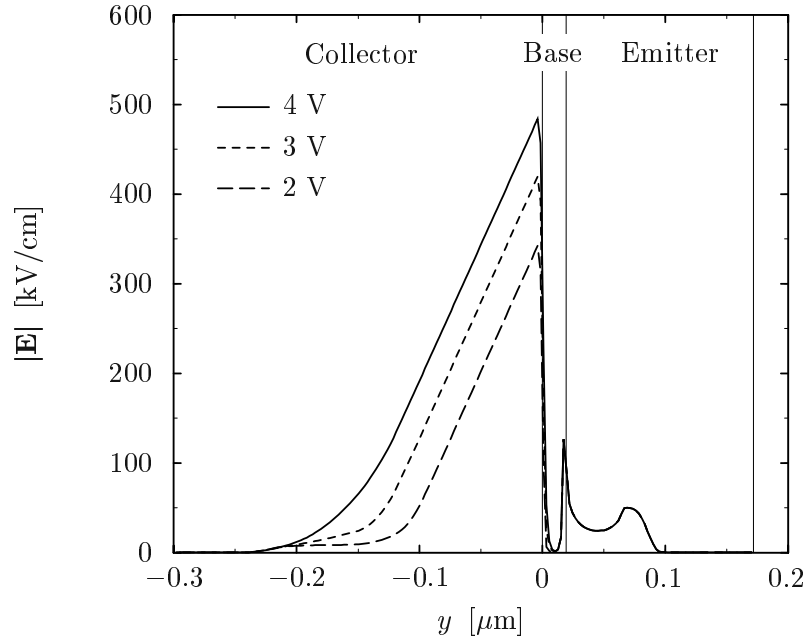


Figure 7.3: Electric field inside the HBT in dependence on V_{CE} for both DD and HD simulation.

on for the transient analysis which gave the fictitious step response of the lattice heat flow equation. The time constant was found to be approximately 1.8 ns. This time constant can be approximated using the thermal resistance and capacity of the material. With the expressions from Appendix A

$$R_{th} = \frac{W}{\kappa_L \cdot A} \quad (7.3)$$

$$C_{th} = A \cdot W \cdot \rho_L \cdot c_L \quad (7.4)$$

and the values from the MINIMOS-NT material database with $W = 0.5 \mu\text{m}$ and $A = 0.4 \mu\text{m}^2$ one obtains

$$\tau = R_{th} \cdot C_{th} = 4.3 \text{ ns} \quad (7.5)$$

which is of the same order as the simulated value.

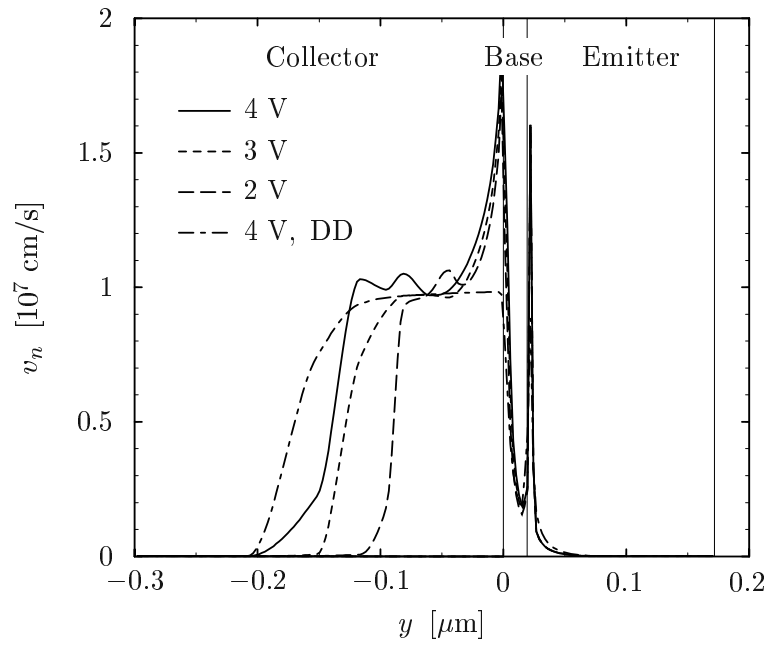


Figure 7.4: Electron velocity inside the HBT in dependence on V_{CE} .

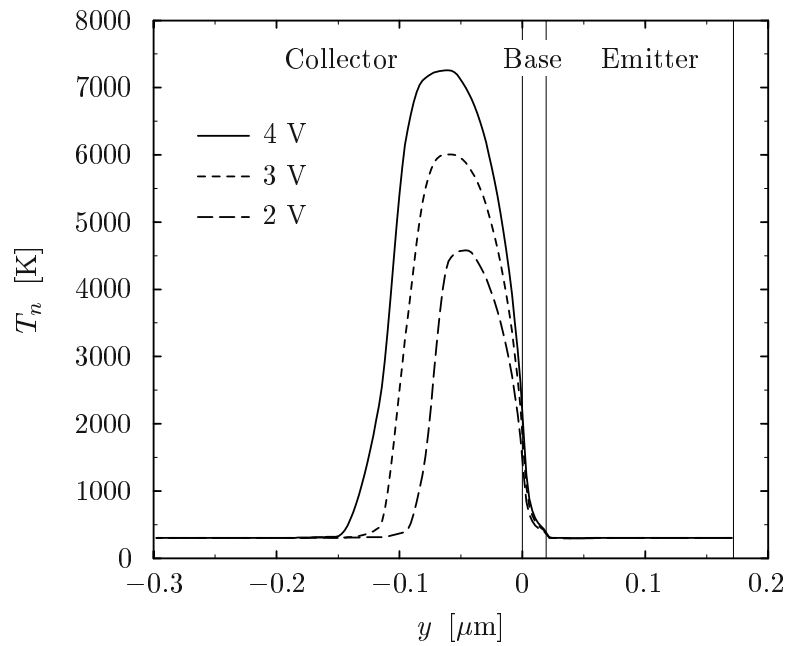


Figure 7.5: Electron temperature inside the HBT in dependence on V_{CE} .

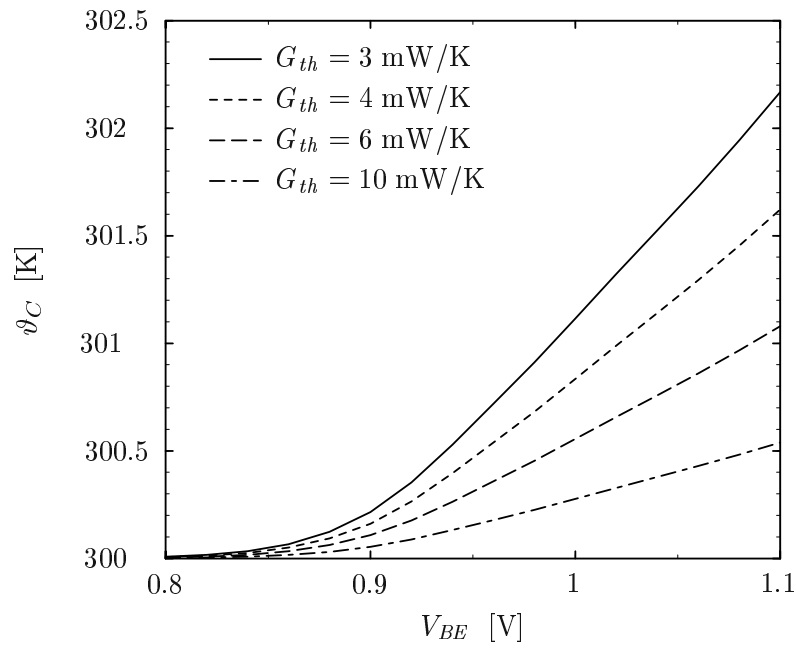


Figure 7.6: Contact temperature of the HBT in dependence on V_{BE} .

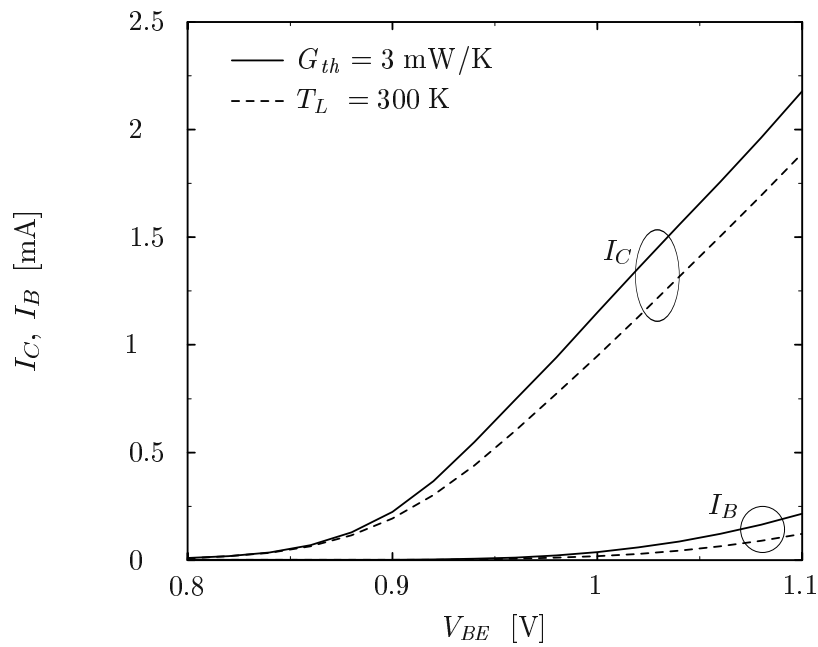


Figure 7.7: Collector and base current of the HBT in dependence on V_{BE} . For the self-heating model I_C is approximately independent of G_{th} .

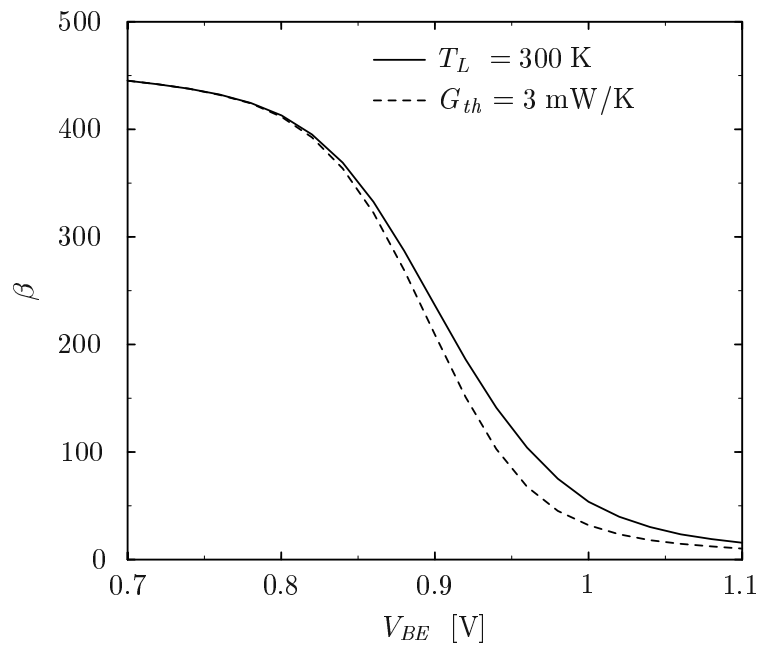


Figure 7.8: Comparison of the current gain of the HBT for self-heating with different G_{th} and $T_L = 300$ K.

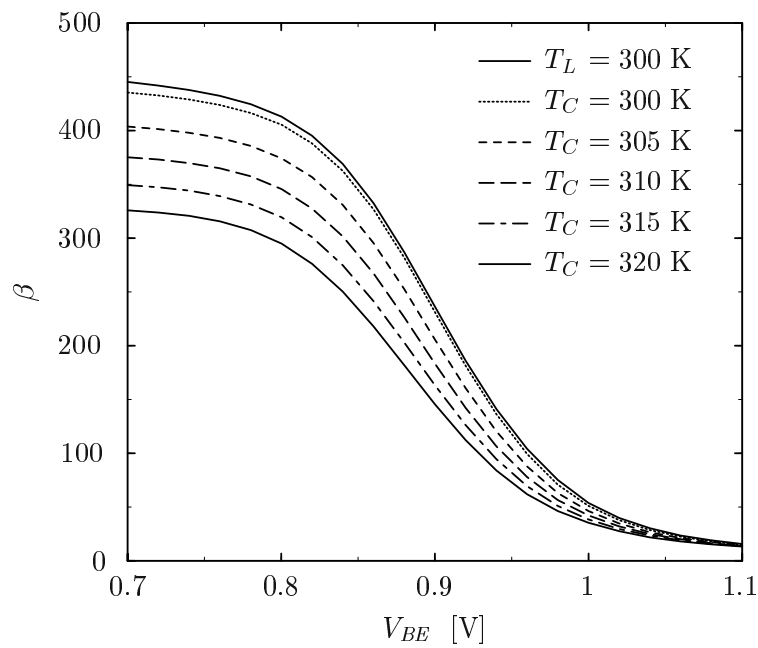


Figure 7.9: Comparison of the current gain of the HBT for self-heating with different contact temperatures and $T_L = 300$ K.

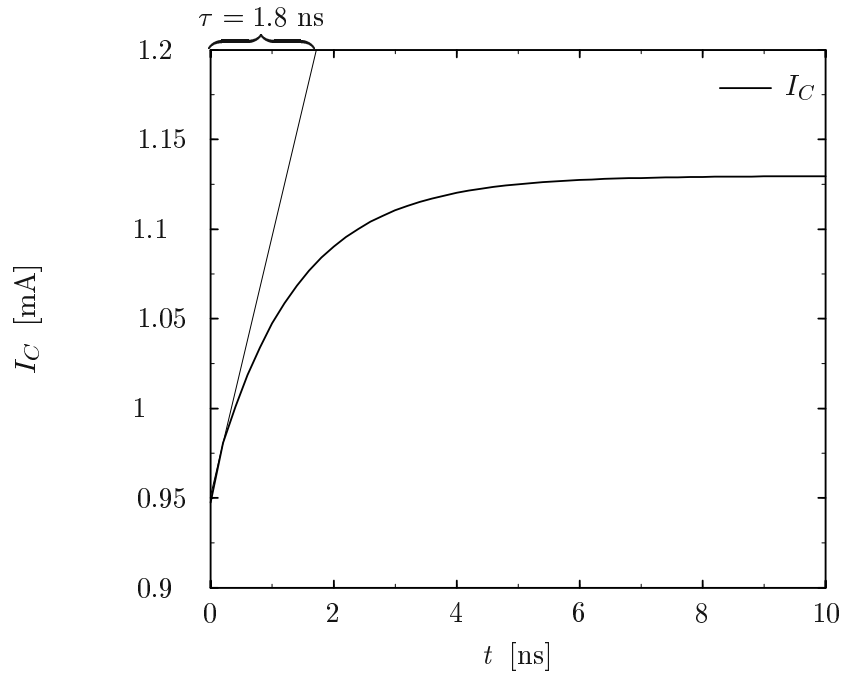


Figure 7.10: Fictitious transient step response of the HBT when turning self-heating on ($V_{BE} = 1$ V).

7.2 Five-Stage CMOS Ring Oscillator

Amongst the simplest digital circuits is the CMOS inverter as shown in Fig. 6.15. It consists of two complementary MOS transistors, an NMOS and a PMOS. During the two steady states only the leakage current flows through the devices and power dissipation is negligible. However, when switching from one state to the other both transistors conduct a much higher current which determines the power dissipation.

The measured delay time depends on the shape of the input curve and on the load at the output. Two conditions should hold: first, the inverter should only delay the signal without distorting it and secondly, the load at the output should be equal to the input impedance of the inverter. These conditions are provided by ring oscillators which are frequently used to measure the delay time of digital circuits [11]. A ring oscillator consists of $n = 2 \cdot k + 1$ inverters and the inverter delay time results to

$$\tau_d = \frac{1}{2 \cdot n \cdot f_{osc}} \quad (7.6)$$

with f_{osc} being the oscillation frequency.

A five-stage ring oscillator circuit is shown in Fig. 7.11. For both transistors a device width of $W = 1 \mu\text{m}$ was assumed. Normally, to achieve equal noise margins, a ratio of $W_p/W_n \approx 2.5$ is used to compensate for the poorer performance of the PMOS transistor [11]. To model the influence of the interconnect circuitry, an additional load capacity of 5 fF was used. The operating point calculation of this circuit provided no problems. However, to force the circuit into a predefined initial state, the input voltage φ_{in} of the first inverter was set to zero during operating point calculation.

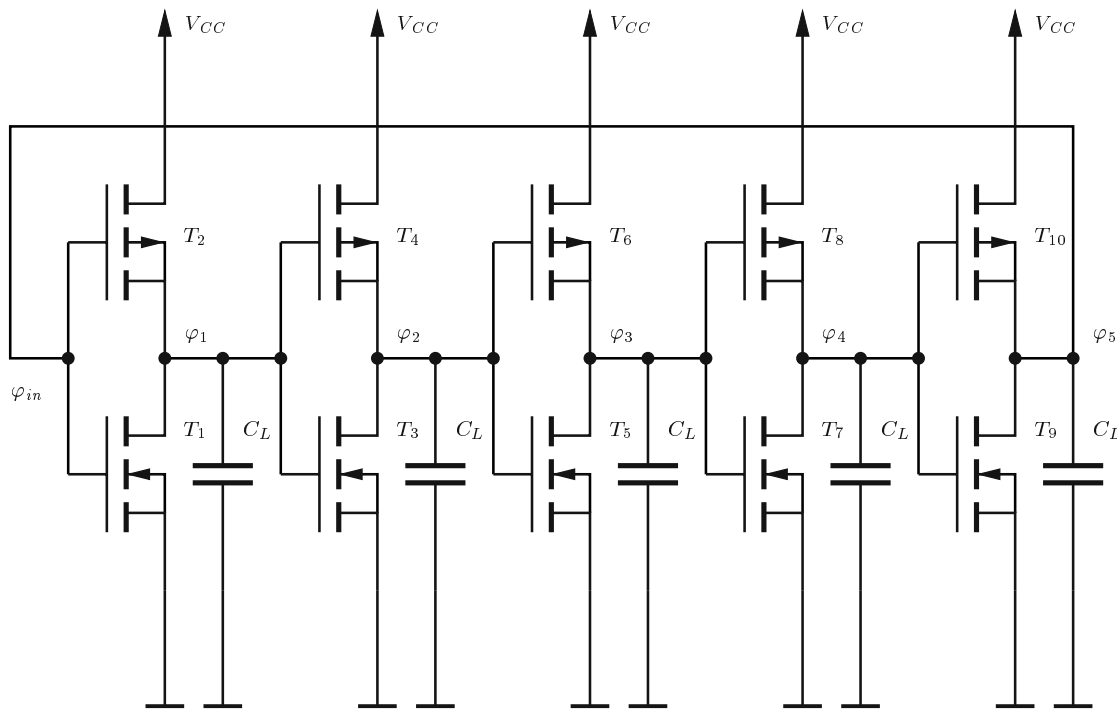


Figure 7.11: Five-stage CMOS ring oscillator

Two different ring oscillators have been simulated, one with the long-channel transistors, the other with the short-channel transistors. The respective simulation results are shown in Fig. 7.12 and Fig. 7.13. For the long-channel transistors, the simulation results obtained with the DD and HD transport models agree so closely, that in the graph no differences are visible. However, when using the short-channel devices, the differences are significant. This is due to the larger currents resulting from the HD transport model. The charging and discharging times of an inverter chain is given as [11]

$$\tau_{charge} \propto \frac{1}{I_D} \quad . \quad (7.7)$$

The simulated inverter delay times are $\tau_{DD} \approx 30$ ns and $\tau_{HD} \approx 26$ ns giving a difference of about 15 %. In Fig. 3.16 and Fig. 3.18 the HD currents are approximately 30 % and 5 % higher for the NMOS and the PMOS transistor, respectively. The average of these values (17.5 %) nicely corresponds to the simulated delay time difference of 15 %.

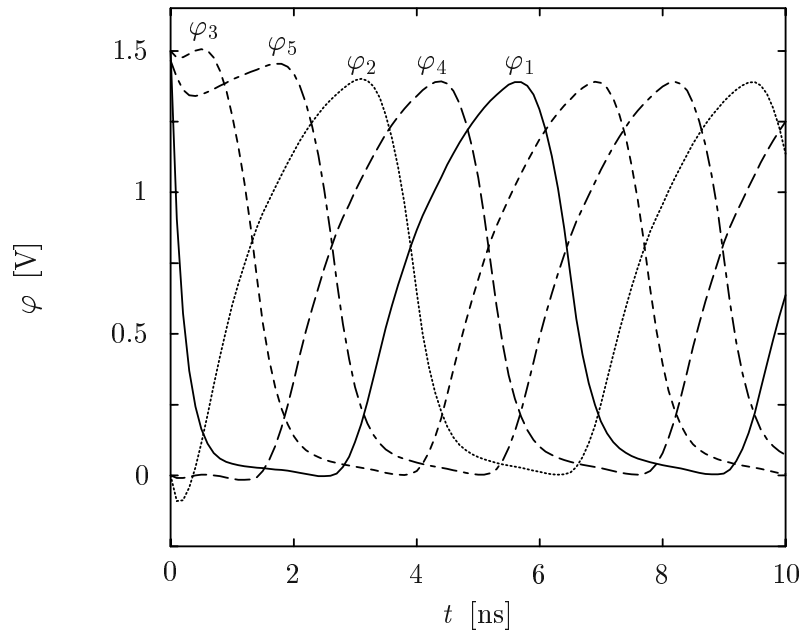


Figure 7.12: Node voltages of the long-channel five-stage CMOS ring oscillator. DD and HD match perfectly.

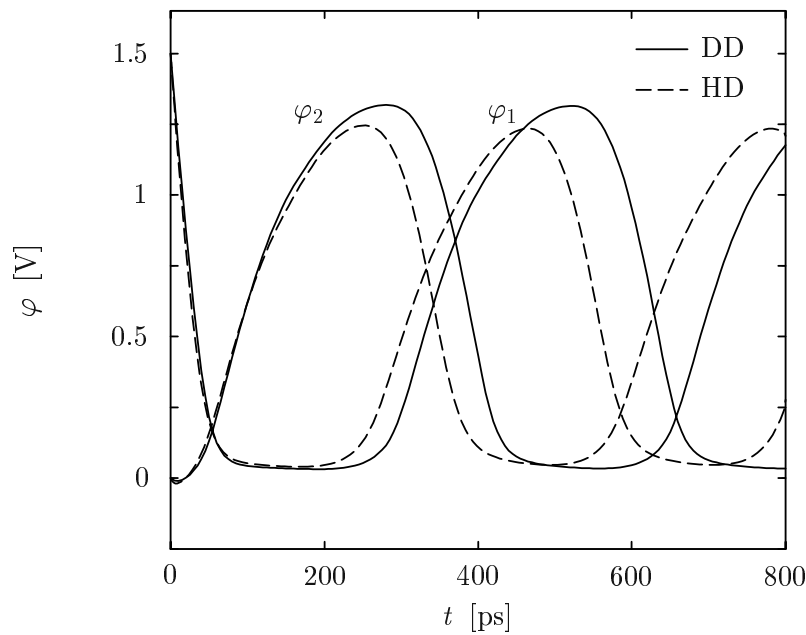


Figure 7.13: Node voltages φ_1 and φ_2 of the short-channel five-stage CMOS ring oscillator for DD and HD.

7.3 Five-Stage CML Ring Oscillator

Most of the bipolar logic families such as T^2L and I^2L are saturated-mode logic circuits. As the transistors are driven into saturation, circuit speed is diminished. Several techniques have been considered to prevent saturation such as Schottky clamping. To prevent the transistors from saturating, the current and voltage swings have to be limited. The operation of non-saturating logic is based on current switching and is hence known as current mode logic (CML). A CML gate is an emitter coupled logic (ECL) gate stripped of the emitter-follower [11] which provided the power gain for driving external circuits. Also the gate propagation delay is increased by the transit time of the transistor, the overall speed is more than compensated by the reduced RC time constant that the external load would have placed on the switching pair [71]. However, due to the improved fabrication technologies the situation is considerably different from that in which ECL was developed and the emitter-follower can be considered an unnecessary artifact of the past.

The DC transfer characteristic of a single stage without load from Fig. 6.12 can be approximated by assuming a simple Ebers-Moll model for the transistors [1, 71]. The input voltage

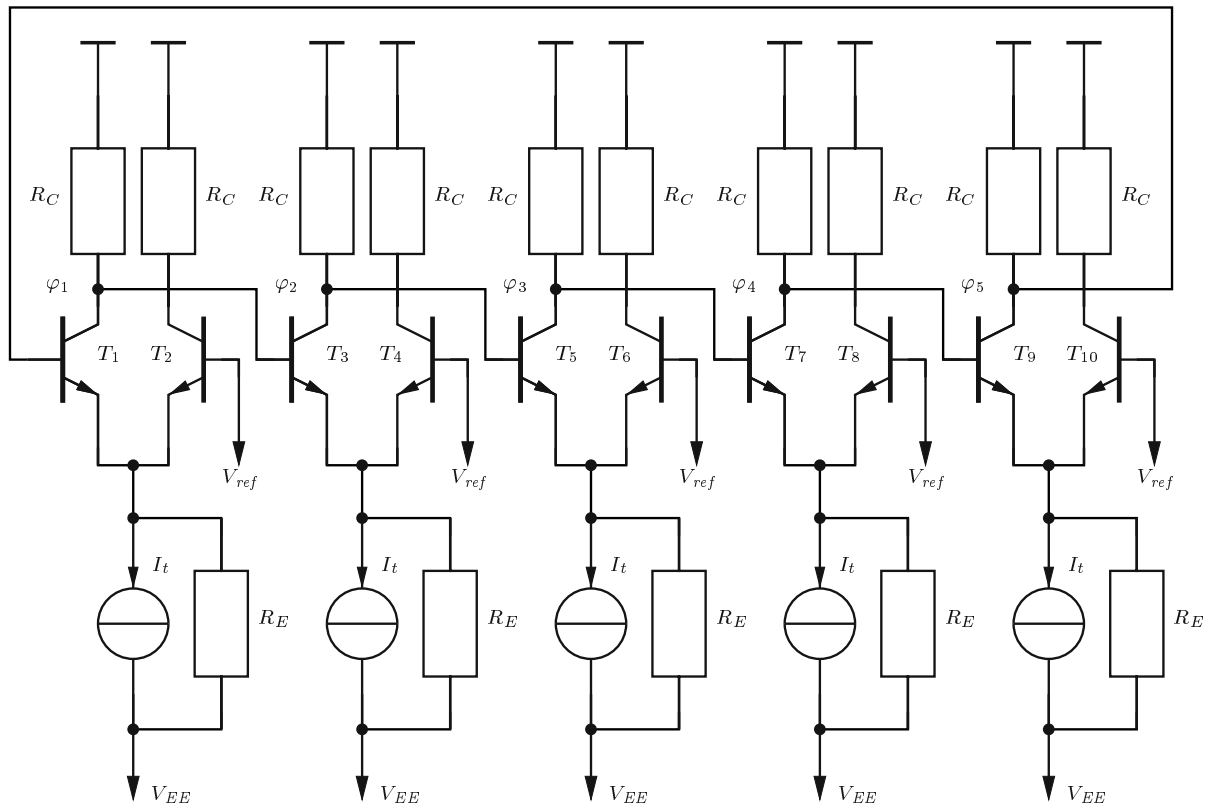


Figure 7.14: Five-stage CML ring oscillator

is compared to the reference voltage

$$V_{in} = \varphi_{in} - V_{ref} \quad (7.8)$$

$$= V_{BE1} - V_{BE2} \quad (7.9)$$

$$= V_T \cdot \left(\ln \frac{I_{C1}}{I_S} - \ln \frac{I_{C2}}{I_S} \right) = V_T \cdot \left(\ln \frac{I_{C1}}{I_{C2}} \right) \quad (7.10)$$

Neglecting the base currents and R_E , the sum of the collector currents must be equal to the current flowing through the current source, hence

$$I_{C1} + I_{C2} = I_t \quad (7.11)$$

With the voltage drops at R_C both $\overline{\varphi_{out}}$ and φ_{out} can be calculated as

$$\overline{\varphi_{out}} = -I_{C1} \cdot R_C = -\frac{V_S}{1 + \exp\left(-\frac{V_{in}}{V_T}\right)} \quad (7.12)$$

$$\varphi_{out} = -I_{C2} \cdot R_C = -\frac{V_S}{1 + \exp\left(\frac{V_{in}}{V_T}\right)} \quad (7.13)$$

with

$$V_S = I_t \cdot R_C \quad (7.14)$$

The voltage gain of the single inverter is

$$A_v = \frac{\partial \overline{\varphi_{out}}}{\partial V_{in}} = -\frac{V_S \cdot \exp\left(-\frac{V_{in}}{V_T}\right)}{V_T \cdot \left(1 + \exp\left(-\frac{V_{in}}{V_T}\right)\right)^2} \quad (7.15)$$

with a maximum at $V_{in} = 0$ V

$$A_v^{max} = -\frac{V_S}{4 \cdot V_T} \quad (7.16)$$

For A_v^{max} greater than unity V_S must be greater than $4 \cdot V_T$. The larger gain is used by the system to account for static and dynamic voltage drops that occur in practice. In practice, a minimum gain of 4 is needed to provide sufficient noise margins [52, 74].

For the simulation $V_S = 20 \cdot V_T$ has been assumed hence a gain of -5 could be expected. When considering an inverter chain consisting of 5 CML inverters as shown in Fig. 7.14 the total gain occurring at the last output node is $(-5)^5 = 3125$. With such a high gain, the circuit is too sensitive to the voltage changes occurring during iteration so no solution can be found without a proper initial-guess using conventional techniques. However, using the shunt conductance technique with $\kappa = 4$ a DC operating point can be easily obtained with only 34 iterations.

First, starting from the operating point obtained above, the DC open-loop transfer characteristic was determined which is shown in Fig. 7.15. The voltage gain of the circuit is shown in Fig. 7.16 which corresponds approximately to the simple results obtained above.

As there is no unique operating point for the closed-loop one of the node voltages had to be fixed to force the circuit into an initial state from which oscillations can start. Hence, the input voltage of the first inverter was fixed to 0 V during the operating point calculation.

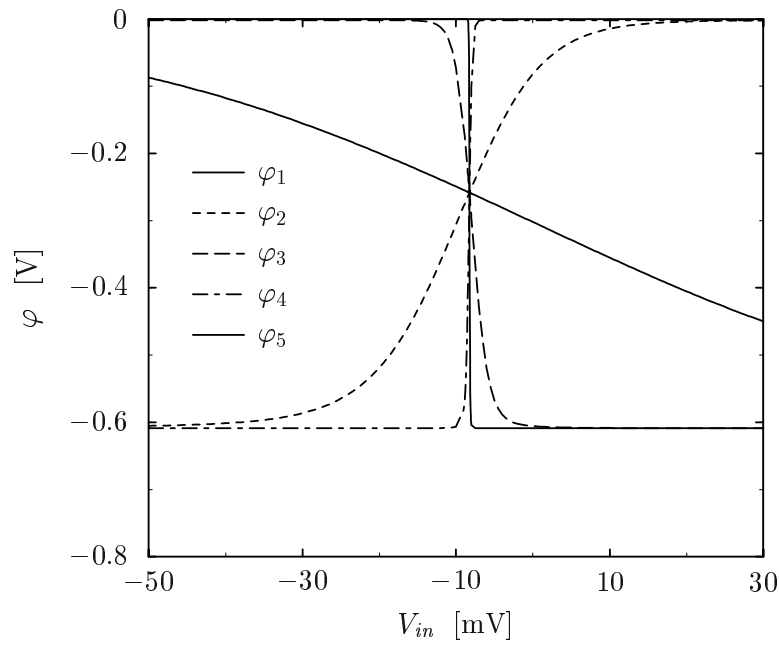


Figure 7.15: Open-loop DC transfer characteristic for the CML ring oscillator.

Oscillations start immediately with a frequency $f_{DD} = 6.8$ GHz for DD and $f_{HD} = 10.6$ GHz for HD which gives a relative difference of 36% for the DD model (Fig. 7.17). This is due to the velocity overshoot which occurs in the base-collector space charge region which cannot be modeled using a DD transport model. The current levels are approximately equal in both cases as shown in Fig. 7.18.

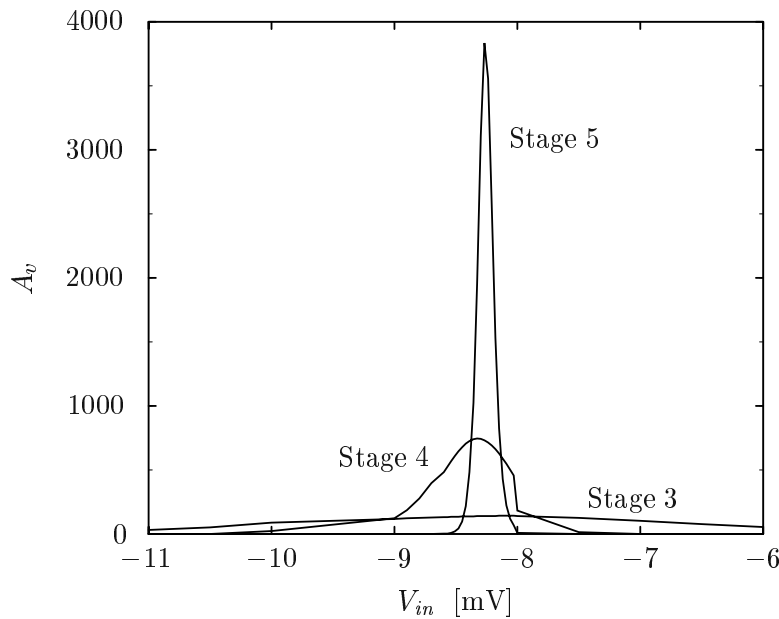


Figure 7.16: Open-loop gain for the CML ring oscillator at the last three stages.

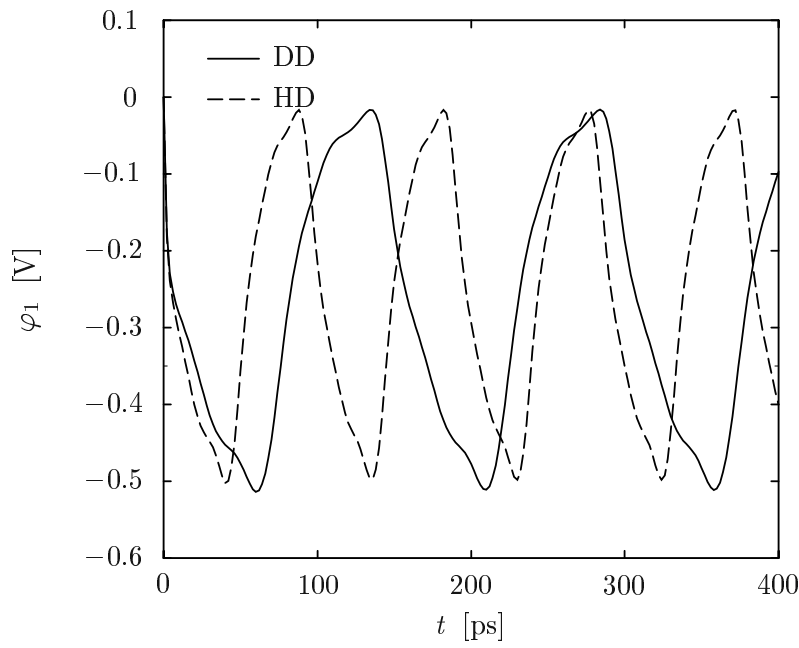


Figure 7.17: Oscillation of node voltage φ_1 of the five-stage CML ring oscillator. Large discrepancies between DD and HD are observed.

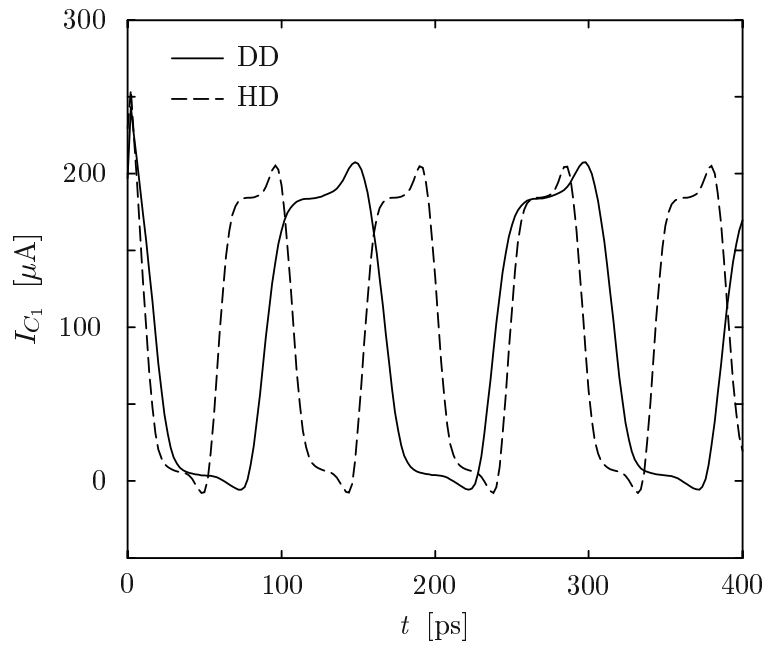


Figure 7.18: Oscillation of the collector current of T_1 of the five-stage CML ring oscillator. Current levels are approximately the same for both transport models.

7.4 Thermal Analysis of an Emitter-Coupled Pair

Matching considerations are extremely important for emitter-coupled pairs and current mirrors. A schematic of a typical circuit is shown in Fig. 7.19. Even though this circuit is topologically very similar to the CML inverter in Fig. 6.12 its numerical behavior is quite different due to the large dynamic range of the node voltages. Especially when considering thermal coupling between the transistors, the problem becomes extremely nasty as the transistors are very sensitive to the thermal boundary condition.

To aid convergence when simulating thermally fully coupled transistors a special iteration scheme was used. In the first block the thermal quantities were ignored until an electrical solution was found. In the second block, the lattice temperature was added to the solution vector without considering the coupling effects caused by the node temperatures. This was also found to be advantageous when stepping through the DC transfer curve hence this block was also used for the consecutive steps. However, as the condition of the transient problem is much better, this block is not used for transient simulation. Only after a proper temperature distribution inside the devices has been established for the new voltage boundary conditions, can the complete equation system be used. In terms of the IPL this iteration scheme reads

```
ThermalDD
{
  Electric : ~Iterate.Blocks.DDBlock
  {
    ignoreQuantities = "LatticeTemperature,NodeTemperature";
    enter            = ~Extern.firstStep;
  }
  ThermalPre : ~Iterate.Blocks.DDBlock
  {
    ignoreQuantities = "NodeTemperature";
    enter            = ~Extern.firstStep || !~Solve.transient;
  }
  Thermal : ~Iterate.Blocks.DDBlock;
}
```

When the transistors are not thermally coupled their characteristics will not match. This is due to the fact, that one of the transistors conducts more current than the other which results in different power consumption and hence different self-heating. To model the thermal coupling of the transistors the thermal circuit shown in Fig. 7.20 has been assumed. The power dissipated by the transistors T_i is given by P_i and a thermal case-ambient conductivity $G_i = 6 \text{ mW/K}$ has been assumed. In addition, T_1 is assumed to be thermally coupled with T_2 and T_3 with T_4 , respectively, using $G_{12} = G_{34} = 2 \text{ mW/K}$. Asymmetrical thermal coupling has been assumed with coupling conductivities $G_{14} = 1.8 \text{ mW/K}$ and $G_{32} = 0.2 \text{ mW/K}$ [42, 45].

The simulated DC transfer characteristic using three different thermal models is shown in Fig. 7.21. For the first model a constant lattice temperature $T_L = 300 \text{ K}$ was assumed. The second model considered self-heating of the transistors and case-ambient conductivity G_i but no thermal coupling between the transistors. Finally, the third model assumes thermal coupling between some of the transistors as stated above. The contact temperatures of the transistors are shown in Fig. 7.22 and Fig. 7.23, respectively. For self-heating without coupling the collector currents are overestimated and correspondingly the available output voltage underestimated due to the large voltage drops at R_C . When neglecting self-heating effects, the collector currents are too low.

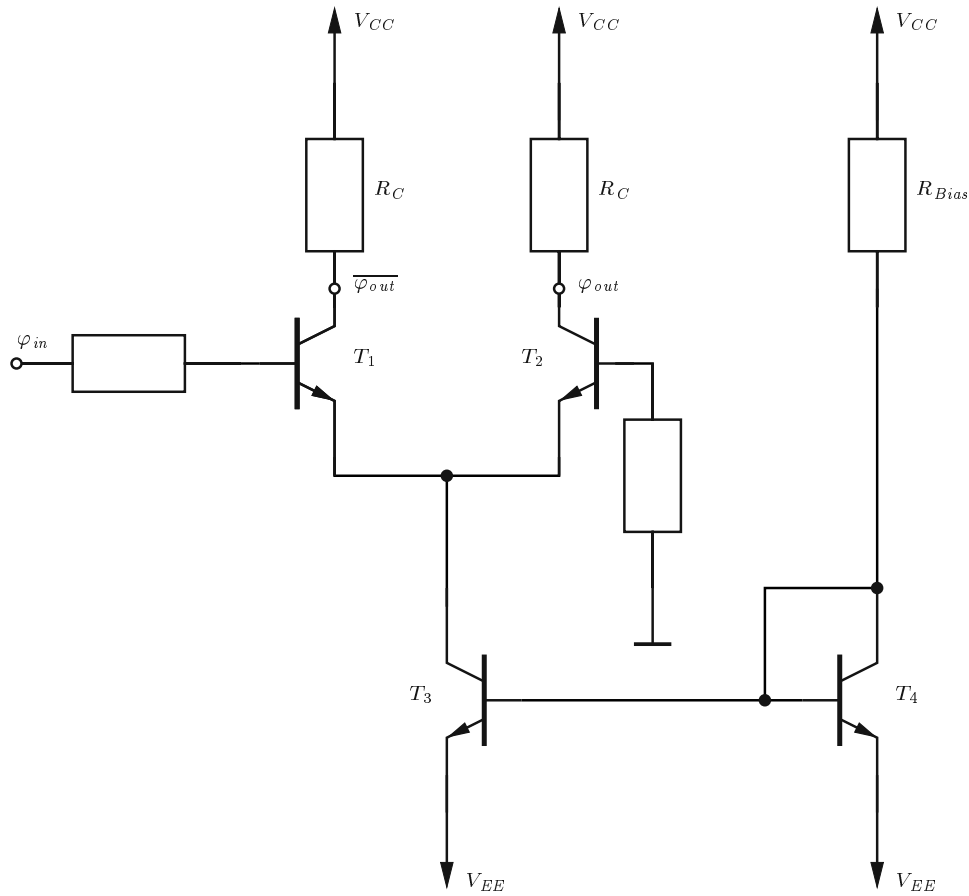


Figure 7.19: Schematic of a differential pair

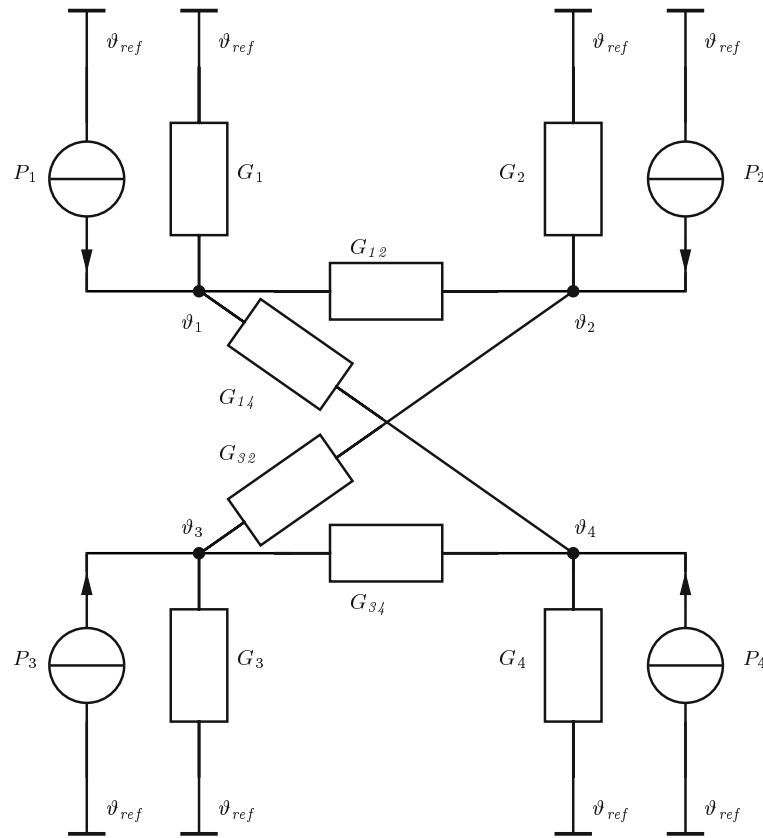


Figure 7.20: Thermal equivalent circuit for the differential pair

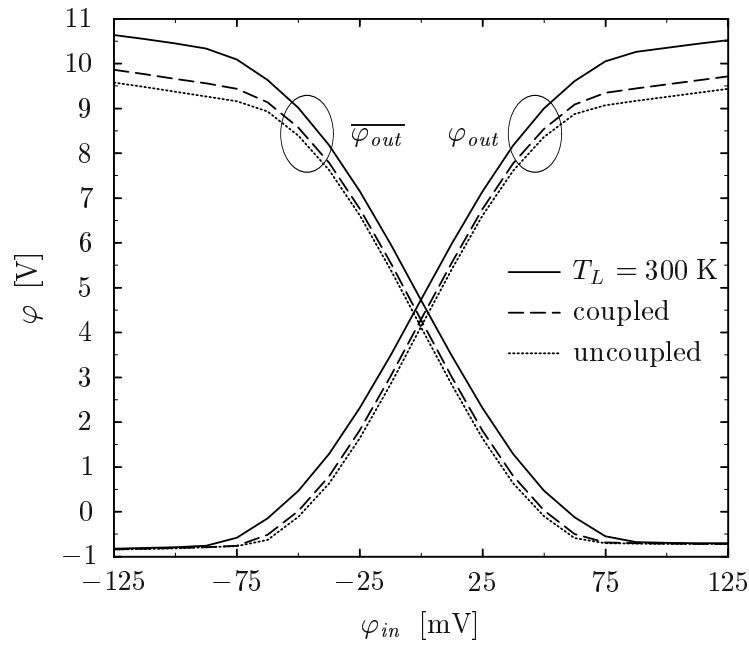


Figure 7.21: DC transfer characteristic of the emitter-coupled pair for different thermal models.

A similar behavior can be observed for transient simulation. The response of the circuit to a sinusoidal input signal with amplitude 0.1 V and $f = 5$ MHz is shown in Fig. 7.24. Although the thermal time constant is sufficiently small compared to the input signal frequency, the contact temperatures stay approximately constant over time because the circuit operates in small-signal mode.

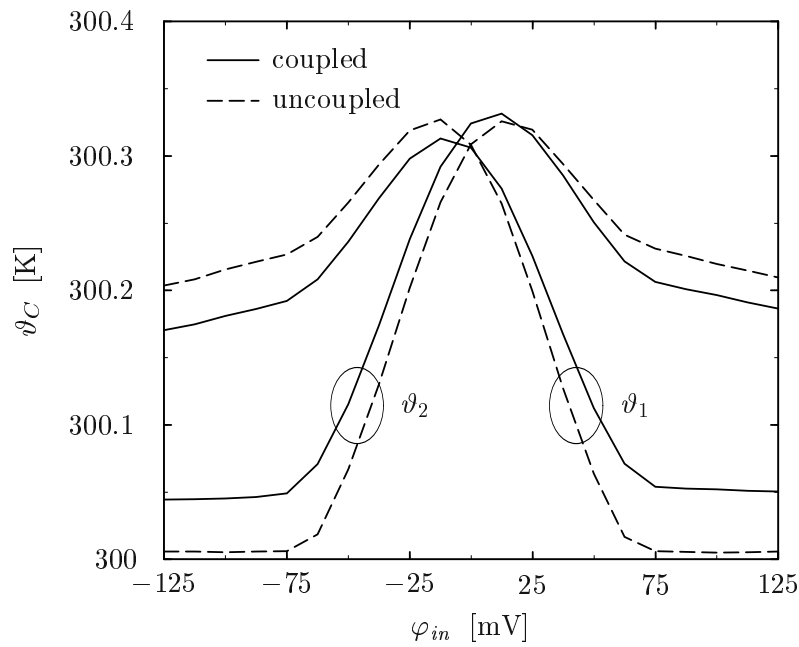


Figure 7.22: Contact temperatures of the transistors T_1 and T_2 against the input voltage.

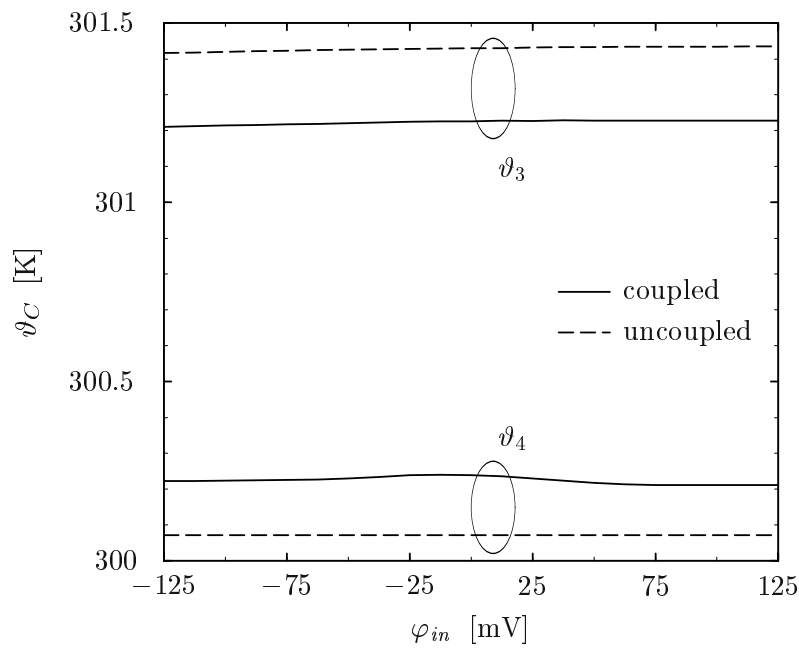


Figure 7.23: Contact temperatures of the transistors T_3 and T_4 against the input voltage.

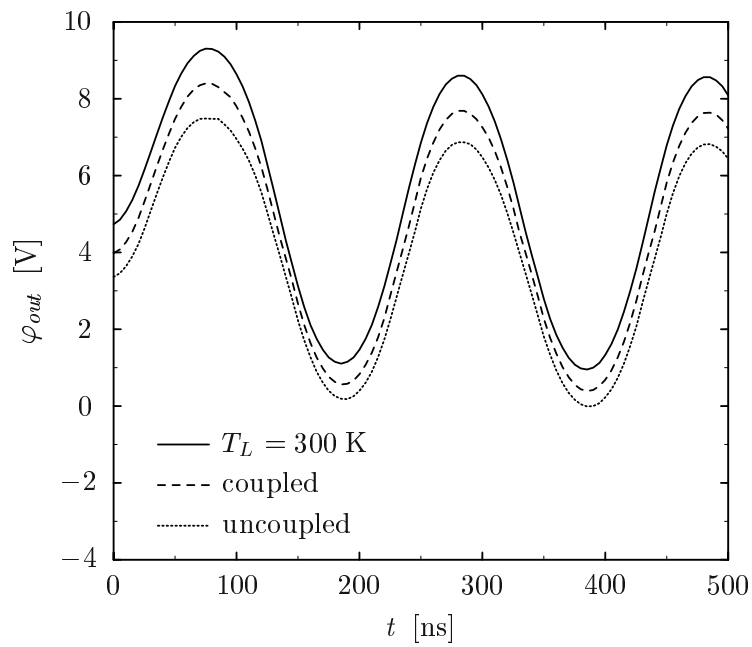


Figure 7.24: Transient response of the emitter-coupled pair for different thermal models.

7.5 Complete OpAmp μ A709

Thermal effects are of fundamental importance for the chip design of integrated circuits. Typical operational amplifiers (OpAmps) can deliver powers of 50–100 mW to a load, and as the output stage internally dissipates similar power levels the temperature of the chip rises in proportion to the dissipated output power [45, 64]. As the transistors are very densely packed, self-heating of the output stage will affect all other transistors. This is especially true as silicon is a good thermal conductor, so the whole chip tends to rise to the same temperature as the output stage. However, small temperature gradients develop across the chip with the output stage being the heat source. The temperature coefficient of the junction voltage for forward-biased pn-junctions is known to be approximately -2 mV/K, that is to obtain the same current a smaller junction voltage is needed. As shown in Fig. 7.25 these temperature gradients appear across the input components of the OpAmp and induce an additional input voltage difference which is proportional to the output dissipated power.

To give an approximate expression for this effect, it can be assumed, to a first order, that the temperature difference $T_2 - T_1$ across a pair of matched and closely spaced components is given simply by

$$T_2 - T_1 \approx \pm K_T \cdot P_d \quad (7.17)$$

with P_d being the power dissipated in the output stage, and K_T a constant with dimension K/W. As the direction of the thermal gradient is unknown the plus/minus sign is necessary in (7.17). In fact, the sign may reverse polarity during the output swing as the dominant source of heat shifts from one transistor to the other. For the simple circuit model shown in Fig. 7.25a the thermally induced input voltage V_{th}^{in} which adds to the input offset voltage

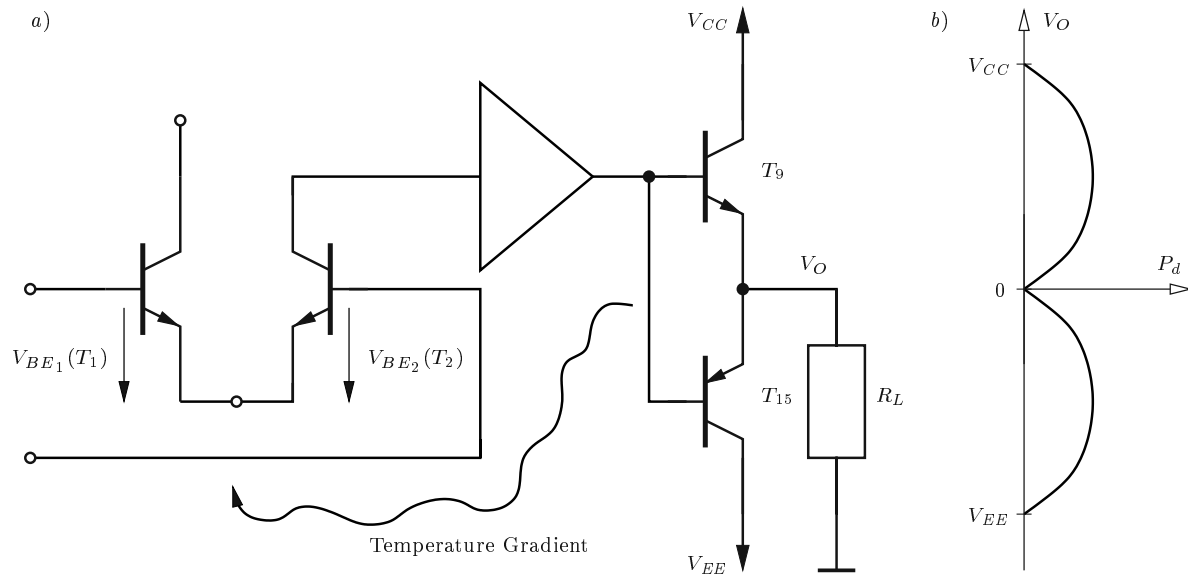


Figure 7.25: a) Simple model illustrating thermal feedback and
b) dissipated power P_d against the output voltage V_O .

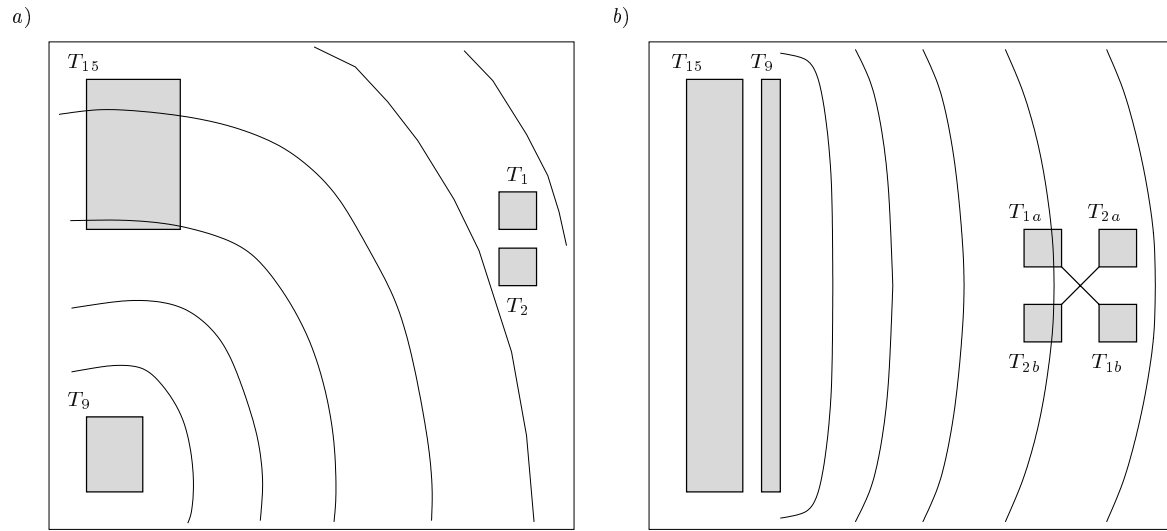


Figure 7.26: a) Simple layout demonstrating mismatch due to improper placement of the input stage and b) improved placement of the input stage on thermal isolines.

can be approximated as

$$V_{th}^{in} \approx \pm K_T \cdot P_d \cdot 2\text{mV/K} \quad (7.18)$$

$$\approx \pm \gamma_T \cdot P_d \quad (7.19)$$

For a thermally well designed OpAmp, typical values measured for K_T are ≈ 0.3 K/W and hence $\gamma_T \approx 0.6$ mV/W. A good thermal layout is obtained when all coupled pairs are placed on thermal isolines. This is somehow difficult to obtain as the output transistors are geometrically separated and hence the isolines will be different depending on the state of the circuit [18, 38]. A simple layout demonstrating this problem is shown in Fig. 7.26a whereas a configuration as shown in Fig. 7.26b can be used to minimize mismatch.

The dissipated power in the output stage of Fig. 7.25a can be written as

$$P_d = (V_S - V_O) \cdot I_L = \frac{V_S \cdot V_O - V_O^2}{R_L} \quad (7.20)$$

with

$$V_S = V_{CC} \quad \text{when } V_O > 0 \quad (7.21)$$

$$V_S = V_{EE} \quad \text{when } V_O < 0 \quad (7.22)$$

A graph of P_d is shown in Fig. 7.25b with zero dissipation occurring at $V_O = 0, V_{CC}, V_{EE}$ and maximum dissipation occurring at $V_O = V_{CC}/2$ and $V_{EE}/2$. It follows from (7.19) that V_{th}^{in} has the same double-humped shape since it is just equal to a constant times P_d under DC conditions. However, the sign of V_{th}^{in} may change during output swing in dependence of the layout. This is the case for the layout shown in Fig. 7.26a where the induced temperature difference changes sign when the output stage shifts operation from T_9 to T_{15} .

The complete μ A709 [25, 59] as shown in Fig. 7.27 has been simulated considering thermal interaction between the input and the output stage. This circuit is of special interest as it is

one of the SPICE benchmark circuits given in [43]. The DC transfer characteristic has been calculated with and without thermal interaction. Consideration of thermal interaction was done by solving the lattice heat flow equation for the transistors T_1, T_2, T_9 and T_{15} and by assuming a thermal network as shown in Fig. 7.28. The thermal conductances were assumed to be $G_1 = G_2 = 2 \text{ mW/K}$ and $G_9 = G_{15} = 10 \text{ mW/K}$ while the coupling mismatch was modeled by $G_{1,9} = G_{1,15} = G_k = 10 \text{ mW/K}$ and $G_{2,9} = G_{2,15} = G_k \cdot (1 - \Delta)$ with Δ being the mismatch parameter which is proportional to the temperature gradient across the input transistors [45].

The solution of the fully coupled equation system is possible with a proper iteration scheme. A small change in the output voltage during iteration causes a large change in the collector current of the conducting output transistor. The dissipated power changes and with it the temperature distribution inside the output transistor. This modified power alters the base-emitter voltages of the input transistors which produces a change in the base-emitter voltages of the output transistors. As all these coupling mechanisms are highly non-linear special precautions have to be taken. The iteration scheme given in Section 7.4 could be used. However, as the simulation failed very frequently for too large steps of the input voltage an additional failure criterion in the Thermal block was added. When the step of the input voltage was too large it caused oscillations in the solution which, due to the strong non-linearities, blew up the lattice temperatures. This took approximately 30 iterations which were very expensive in computational terms as each iteration took approximately 20–200 seconds depending on the condition of the system matrix. So this event had to be detected as soon as possible. It was found that an abnormal behavior of the potential update norm $E_\infty(\mathbf{u}_\psi)$ was a good indication of starting oscillations. Hence, whenever $E_\infty(\mathbf{u}_\psi)$ was larger than approximately $10^2 \cdot V_T$ after 10 iterations or whenever $E_\infty(\mathbf{u}_\psi)$ exceeded $10^5 \cdot V_T$ the iteration was canceled. Furthermore, the number of iterations was limited to 30.

An additional, even worse, phenomenon was observed when solving the fully coupled system in the Thermal block. Even for $E_\infty(\mathbf{u})$ already as small as 10^{-2} the iterative solver failed and also generated a failure event.

These failure events were handled by reducing the increment of the input voltage by a factor of 2. Whenever the system could again be solved within 15 iterations the step size was doubled until the original step size was obtained.

The iteration scheme was inherited from ThermalDD as given in Section 7.4 with an unmodified Electric block. The ThermalPre and the Thermal block were augmented by the failure criterion given above

```
ThermalDD_2 : ThermalDD
{
    aux f = !firstStep &&
        ((updateNormPot > 1e5) ||
         (iterationCount > 10 && updateNormPot > 1e3) ||
         (iterationCount > 30));

    ThermalPre
    {
        failure = ^f;
    }
    Thermal
    {
        failure = ^f;
    }
}
```

The DC transfer characteristic was calculated by stepping φ_{in} from -1 mV to 1 mV with

$\Delta\varphi_{in} = 20 \mu\text{V}$. From SPICE simulations the open-loop gain of the μ A709 was known to be approximately 35000 so for each step of $\Delta\varphi_{in}$ a step of 0.7 V could be expected for $\Delta\varphi_{out}$ which is quite large. However, no convergence problems occurred until φ_{out} approached 0 V. This was the most critical part of the simulation and several step reductions were necessary for both the pure electrical and the thermal simulation. The size of the system matrix was 37177 and 40449 for constant temperature and thermal simulation, respectively, and the simulation took 9 and 25 hours on a Linux Pentium II 350MHz workstation. For the thermal simulation the conditioning of the system matrix was found to be very poor and several step reductions were necessary.

The DC transfer characteristic is shown in Fig. 7.29 with the obvious humps resulting from thermal feedback effects. In Fig. 7.30 the open-loop voltage gain A_v is shown and the dramatic impact of thermal coupling. The thermal conductances assumed in this simulation were very optimistic and an even stronger impact of thermal coupling has been published in literature [18, 38]. For stronger coupling, even the sign of the open-loop voltage gain may change and cause the OpAmp to become unstable.

The maximum temperature and the contact temperature of the output stage are shown in Fig. 7.31. It is obvious that the self-heating inside the transistor plays only a minor role at these current levels. However, the power dissipated inside the device heats up the NPN transistor due to the resistive thermal boundary condition which obstruct the heat flow out of the transistor. This is in accordance to the commonly used assumption that the transistor can be modeled by a power source alone (cf. Appendix A). The PNP transistor has only a β of approximately 10 and comparable current levels have been obtained by increasing the emitter area of the transistor ($W_{PNP}/W_{NPN} = 5$). Hence the locally generated heat density H is even smaller than for the NPN transistor and the temperature drop inside the device is negligible.

A similar situation occurs for input transistors T_1 and T_2 . As they are biased with $I_C = 20 \mu\text{A}$ only self-heating is negligible and the contact temperature resembles the heat transferred from the output stage. As unsymmetric thermal conductivities have been assumed the temperature of T_1 is always slightly higher than the temperature of T_2 . The temperature difference $T_2 - T_1$ is shown in Fig. 7.33 with a maximum of only -22 mK . Even this small temperature difference has a strong impact on the output characteristic due to the high gain of the circuit.

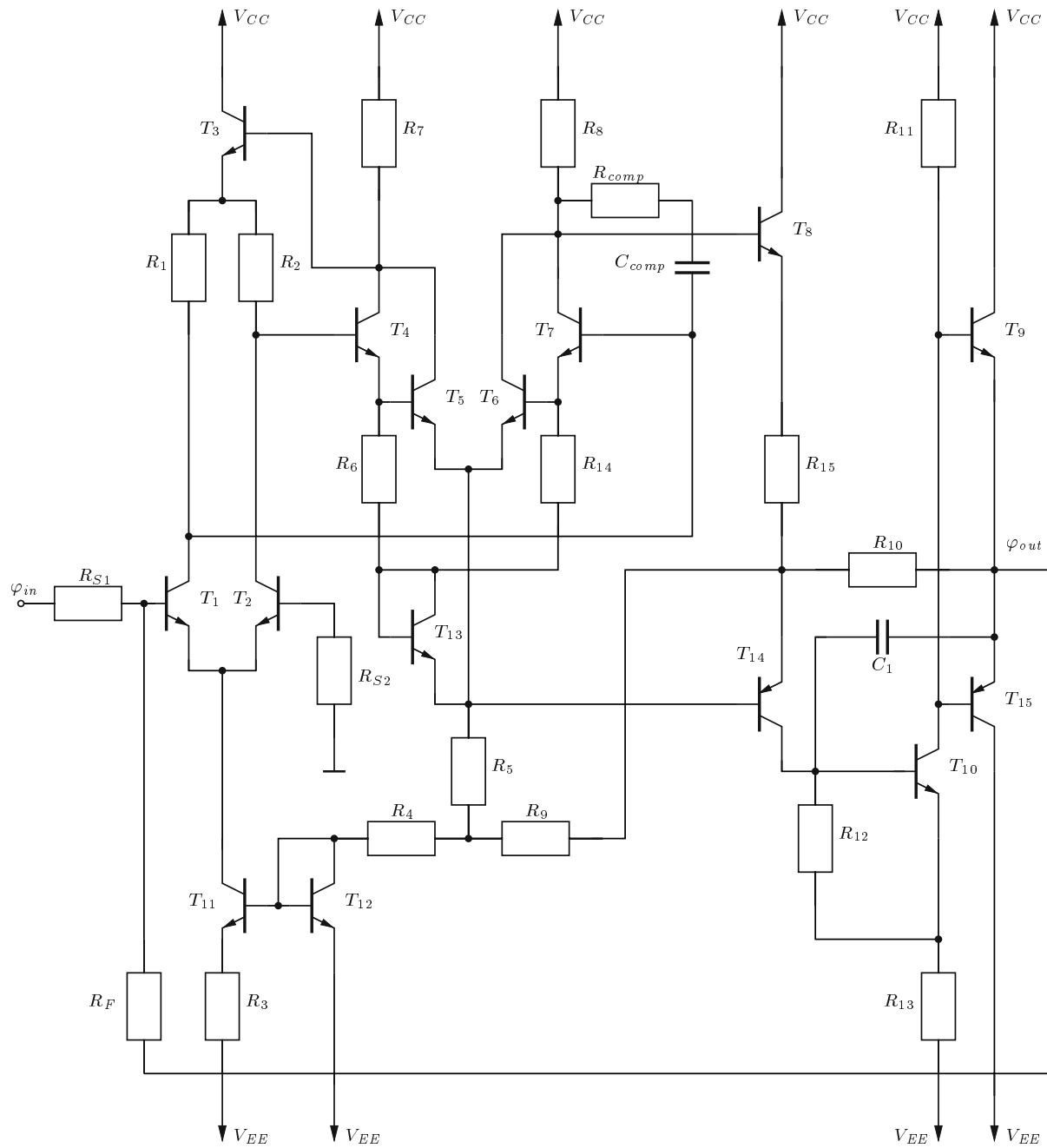


Figure 7.27: Schematic of the $\mu A709$ OpAmp.

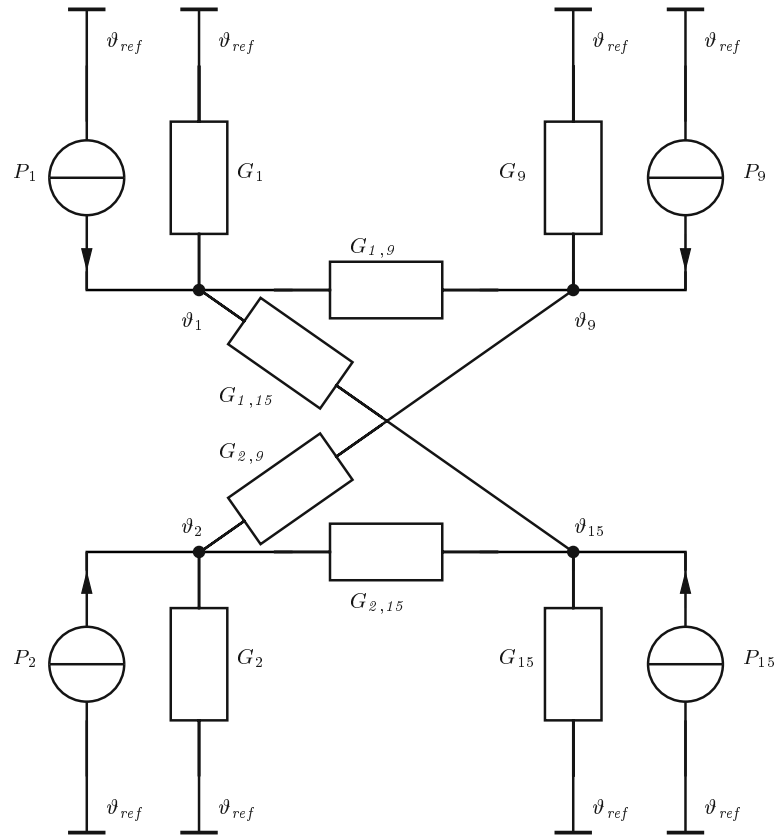


Figure 7.28: Thermal equivalent circuit used to simulate thermal interaction for the μA709 OpAmp.

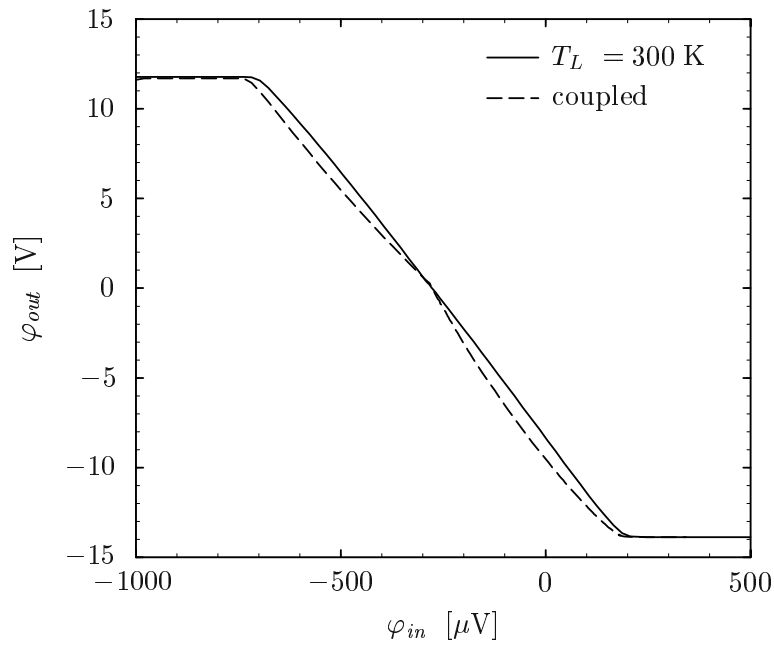


Figure 7.29: DC transfer characteristic of the $\mu A709$ for constant lattice temperature and considering thermal coupling of the input and output stage.

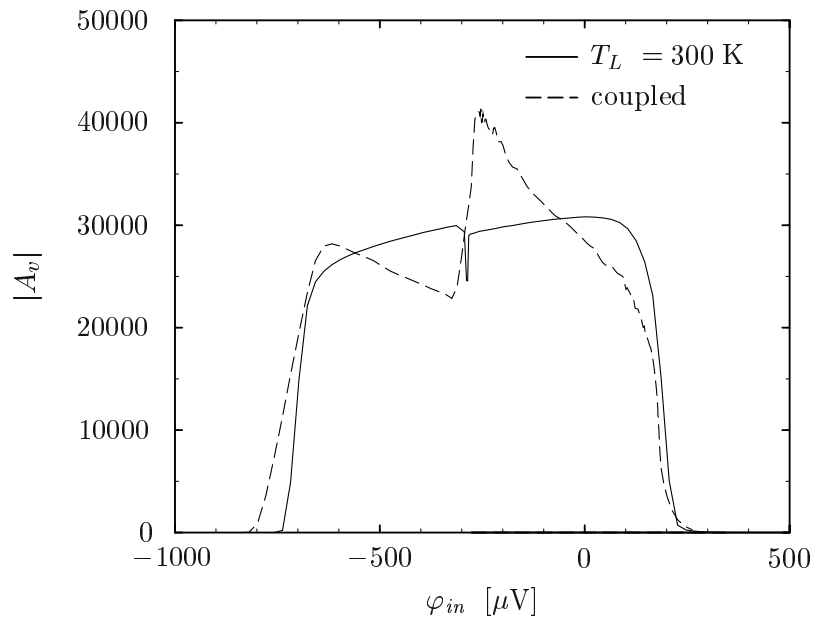


Figure 7.30: Open-loop gain of the $\mu A709$ for constant lattice temperature and considering thermal coupling of the input and output stage.

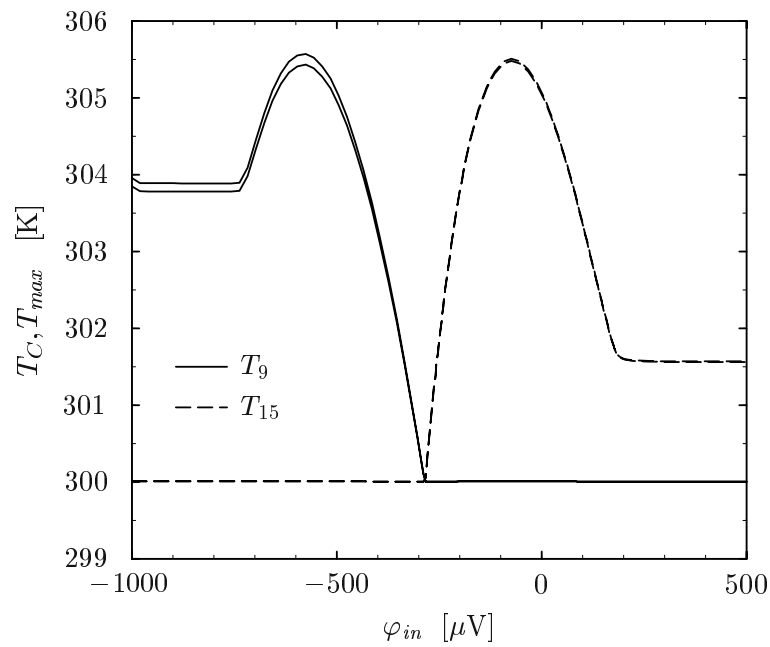


Figure 7.31: Maximum and contact temperature of the output transistors T_9 and T_{15} during the DC transfer characteristic.

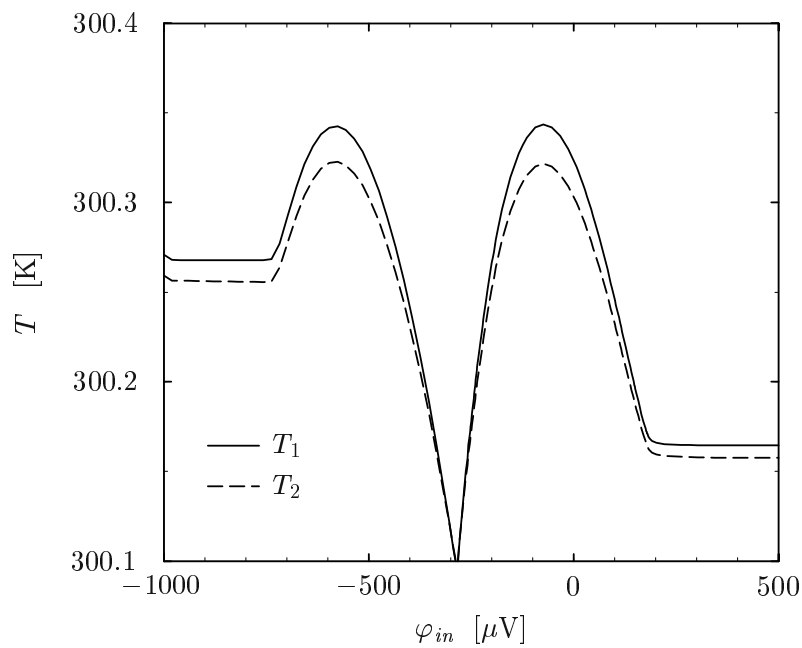


Figure 7.32: Temperature of the input transistors T_1 and T_2 during the DC transfer characteristic.

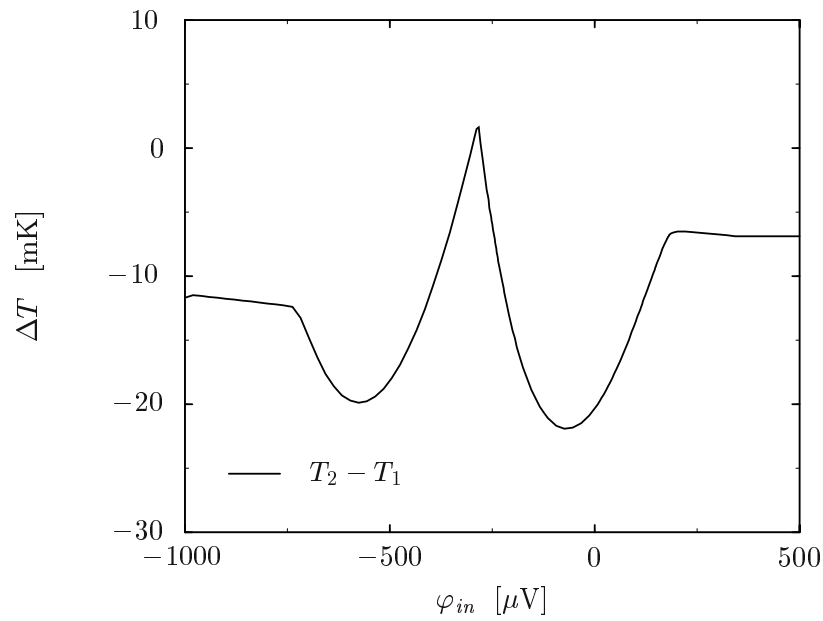


Figure 7.33: Temperature difference of the input transistors T_1 and T_2 during the DC transfer characteristic.

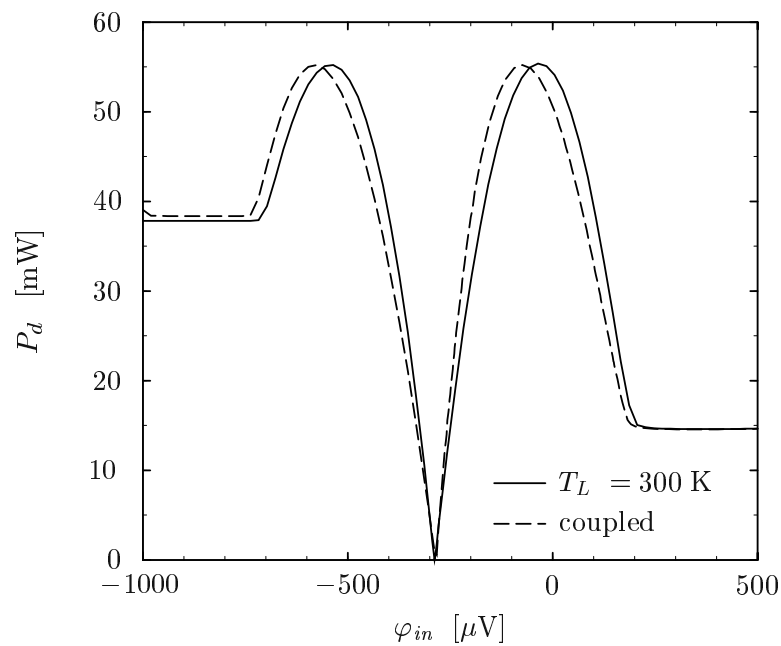


Figure 7.34: Dissipated power P_d in the output stage during the DC transfer characteristic.

Chapter 8

Outlook

The techniques presented in this thesis allow for the solution of circuits using distributed devices for which the semiconductor equations need to be solved. Although these techniques delivered very promising results there is still room for further improvements. Possible improvements are listed in the following.

- Although the shunt conductance technique proposed in this thesis delivered promising results for medium-sized circuits, operating point calculations of larger circuits like the μ A709 are still not possible without an initial-guess. Further improvements might still be possible considering the purely empirical expression for the shunt conductance.
- Considering the thermal equivalent circuit for the μ A709 the simplicity is striking. Only the principal effects can be covered with this extremely simple circuit. It might therefore be desirable to automatically extract a thermal equivalent circuit from available chip geometries to increase the accuracy.
- For a good characterization of devices and circuits a proper implementation of small-signal analysis is of utmost importance. In the current version transit frequencies and capacitances have to be extracted using a transient analysis followed by a Fast Fourier Transformation (FFT). This procedure is extremely time consuming and nearly prohibitive for practical circuits.

Appendix A

Thermal Equivalents

Solving the one-dimensional static lattice heat flow equation

$$\operatorname{div} \mathbf{S}_L = H - \rho_L \cdot c_L \cdot \frac{\partial T_L}{\partial t} \quad (\text{A.1})$$

$$\mathbf{S}_L = -\kappa_L \cdot \operatorname{grad} T_L \quad (\text{A.2})$$

one can derive a thermal equivalent model for a device with a geometry as shown in Fig. A.1a. With the boundary conditions $T_L(-a) = T_1$ and $T_L(a) = T_2$ and under the assumption of constant heat generation H the solution for the lattice temperature inside the device reads

$$T_L(x) = \frac{P_E}{8 \cdot G_{th}} \cdot \left(1 - \left(\frac{x}{a}\right)^2\right) - \frac{\Delta T}{2} \cdot \frac{x}{a} + \bar{T} \quad (\text{A.3})$$

with

$$\Delta T = T_1 - T_2, \quad \bar{T} = \frac{T_1 + T_2}{2} \quad (\text{A.4})$$

$$P_E = H \cdot A \cdot 2 \cdot a, \quad G_{th} = \frac{\kappa_L \cdot A}{2 \cdot a} \quad (\text{A.5})$$

It consists of a linear term arising from the boundary condition and a quadratic term arising from heat generation inside the device due to the dissipated electrical power P_E . The temperature distribution is shown in Fig. A.1b.

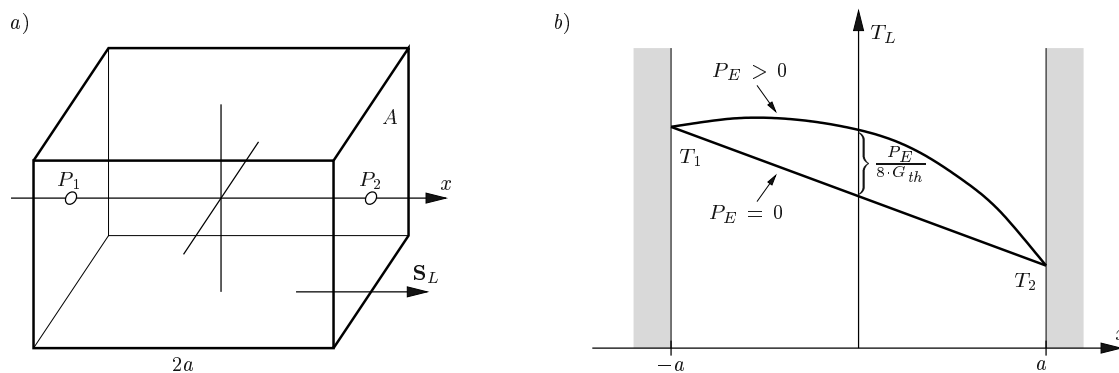
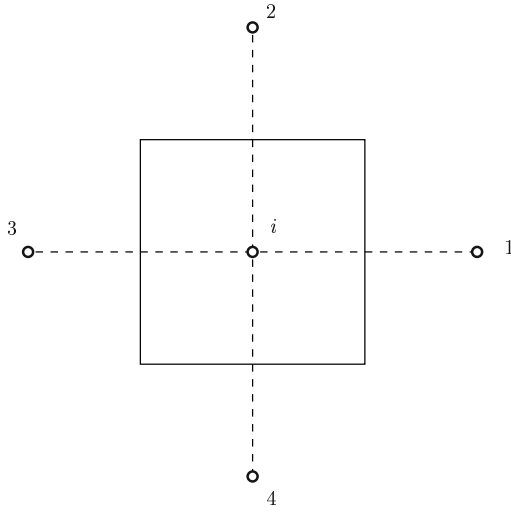


Figure A.1: a) Geometry and b) temperature distribution of a material block.

a)



b)

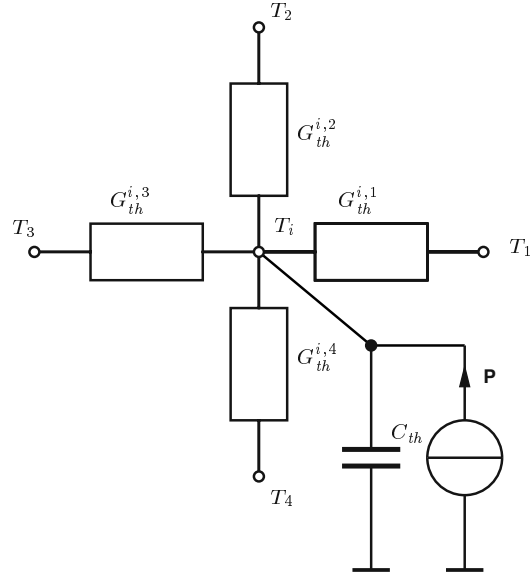


Figure A.2: a) Grid point i and 4 neighbors used for the discretization of the lattice heat flow equation. b) Electrical analog circuit for the lattice heat flow equation.

To derive a thermal equivalent circuit the discretized lattice heat flow equation as solved by MINIMOS-NT can be used. Assuming constant electrical power dissipation P_E and a constant thermal heat capacity c_L the expression for a grid point i reads [36]

$$\sum_j P_{i,j} = \sum_j \frac{P_E}{2} + C_{th} \cdot \frac{T_L - T_{L,o}}{\Delta t} \quad (\text{A.6})$$

$$P_{i,j} = G_{th} \cdot (T_i - T_j) \quad (\text{A.7})$$

$$C_{th} = V_i \cdot \rho_L \cdot c_L \quad (\text{A.8})$$

The sum in (A.6) considers the contribution of all neighbor points j with $P_{i,j}$ being the thermal heat flowing from point i to point j . Using heat flows instead of electrical currents and temperatures instead of voltages an electrical equivalent circuit can be used to model (A.6). This equivalent circuit is shown in Fig. A.2 for grid point i in a two-dimensional situation with four neighbor points. This equivalent circuit is approximately valid for the most general case. The geometry of a schematic layout is shown in Fig. A.3. To keep the problem tractable, device simulation is normally restricted to the electrical active regions which make up only a small portion of the chip. Heat generation is even further restricted to small areas of the device which normally are the space charge regions for bipolar devices and the channel regions for MOS devices. For other areas like buried collectors and substrates heat generation is normally negligible as either the electric field, the current density, or both are low. Therefore the heat generation term H can be neglected for these areas and the thermal equivalent circuit reduces to resistances and capacitances.

On the other hand, for the electrical active region the thermal resistances and capacitances are of minor importance due to their small size. This is the reason why power dissipation in electrical devices is normally modeled by a power source alone which can be considered

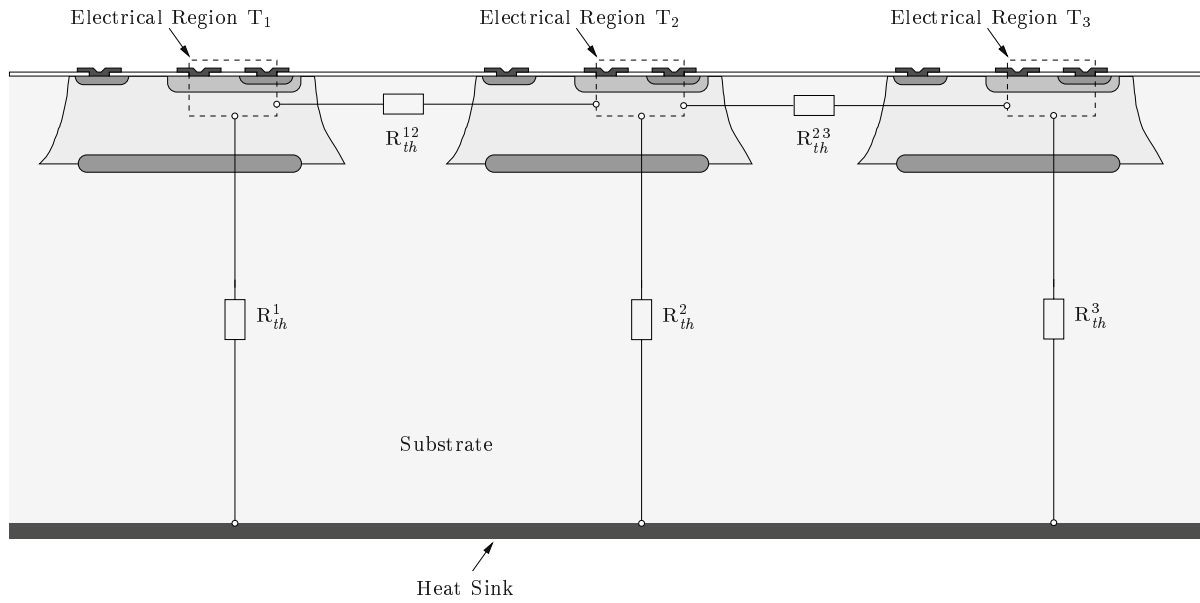


Figure A.3: Electrical and thermal modeling of several transistors.

as reducing the power source to a point source.

It is worthwhile to point out the simplifying assumptions made in the derivations above:

- A constant thermal conductivity κ_L has been assumed which is a very crude approximation. Using the exponential expression from [56]

$$\kappa_L = 1.5486 \frac{\text{W}}{\text{K cm}} \cdot \left(\frac{T}{300 \text{ K}} \right)^{-4/3} \quad (\text{A.9})$$

gives an error in the solution of the linear heat diffusion equation in the order of 30 % [8].

- Heat generation has been assumed constant inside the device. This gives a good approximation only when the temperature rise due to H is small compared to the temperature rise induced by the boundary condition.

Appendix B

Device Model Implementation

In MINIMOS-NT the Algorithm Library is used which was developed by Mlekus [41]. The library supports general C++ classes which simplify and standardize communication between models and the environment they are used in. Device models in MINIMOS-NT are represented by C++ classes [68] derived from a common base class `Model` which provides methods for the communication between the Algorithm Library, the actual `Model` instance and the simulator. `Model` classes encapsulate the algorithms, the private data values, an `Interface` containing the input and output `Parameters`, and the documentation of the `Model` within a single entity. The interface parameters provide the communication interface and are themselves classes. They are implemented as C++ templates `Param<type>` to allow for a maximum flexibility. The base class for all devices is `MmxDevice` which will be described in the following section. The Algorithm Library provides both C++ as models written in a special language called Model Definition Language (MDL). MDL was designed to be as much C++ compatible as possible. However, due to the non-standardized way C++ compilers internally treat variable and class names some restrictions are necessary. All devices are inherited from the class `MmxDevice` which defines the interface to the simulator. C++ does not support dynamic access to member functions and member variables by name (like e.g., `r = getValueOf("R0")`). This functionality is offered by the `Param` class in conjunction with the `Interface` class. The latter maintains a list of parameters which can be accessed by name via the `Interface` class. Hence, all member variables which must be accessible in both C++ and MDL models must be of class `Param`. Care had been taken not to deteriorate the performance but this issue is beyond the scope of this thesis.

B.1 The Class `MmxDevice`

The class `MmxDevice` is derived from `Model` and is a pure virtual class which serves only as a base class for the real device classes. It defines the basic member parameters and provides standardized communication means for the real devices. Among others, the following interface parameters are available: the partial Jacobian matrix (stamp) `Y` which must be calculated by the `evaluate` method of the real device in conjunction with the partial right-hand-side `rhs`. To do this, the `evaluate` method needs the partial solution vector `x`, the partial old solution vector `xold`, and information about which quantity is really solved for as found in `solv`. Furthermore, the current simulation time `t` and the size of the last time-step `dt` are available.

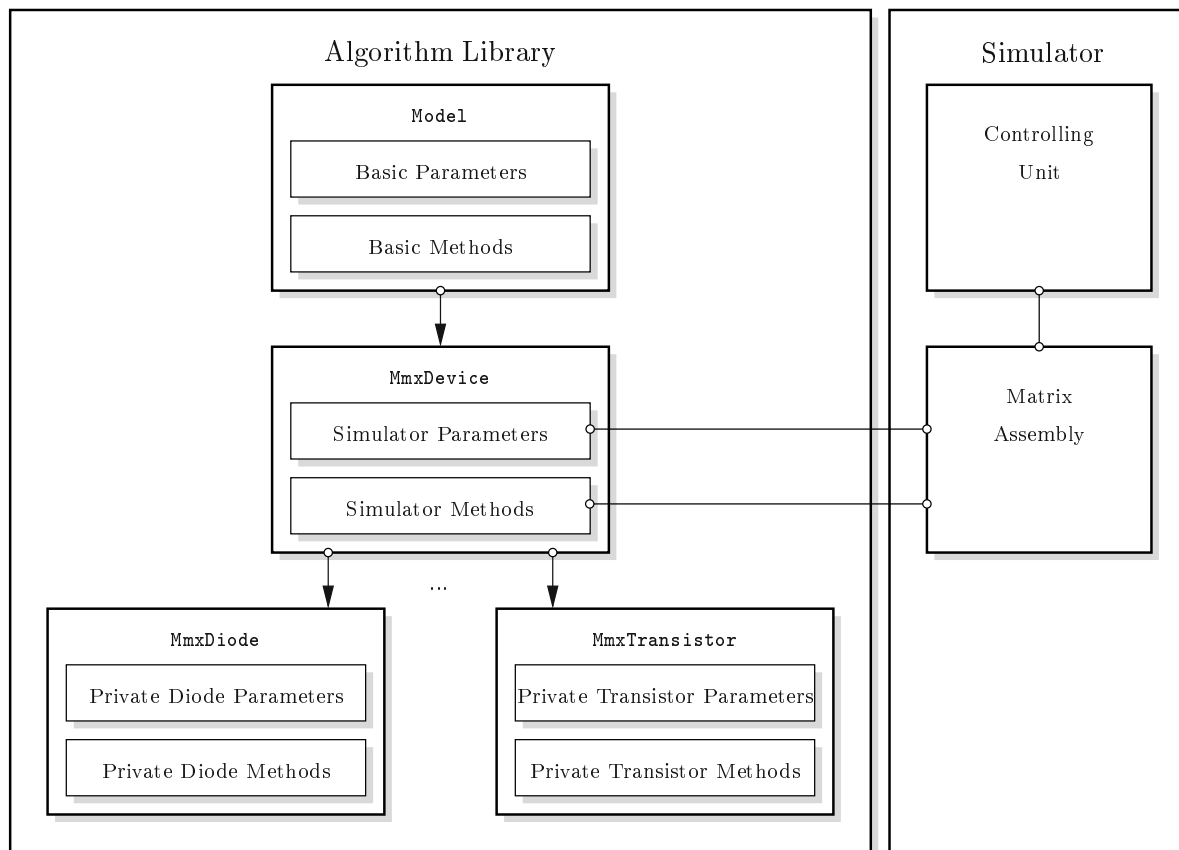


Figure B.1: Class hierarchy within the Algorithm Library and definition of the communication interface to the simulator.

Inside a circuit each device is defined by the number of terminals by which the device is connectable to the circuit. These terminals can be voltage or temperature controlled.

B.2 Real Device Classes

Class hierarchy is shown in Fig. B.1 Each instance is created completely dynamically. This is in contrast to e.g., SPICE where all necessary information is taken from statically allocated tables which is of course faster. However, the overhead is minimal and occurs only during initialization but the device can be optimally configured because only those parameters and keywords are added which are really needed. This is done in the `init` method where each device must declare its nodes and the keywords to configure the device model. Nodes correspond to solution variables and the number of nodes determines the size of the partial Jacobian matrix allocated for the device. A purely electric, two-terminal device would register two nodes and would in turn get a 2×2 partial Jacobian matrix Y assigned. For purely voltage controlled devices Y is of admittance form. Available node types are:

- Normal nodes which provide a connection to a circuit node voltage.
- Internal nodes which are unique and cannot be shared between different devices but

can be used to build internal subcircuits. A typical application is the addition of internal series resistances.

- Branch current nodes which introduce a new branch current into the equation system.
- Thermal nodes are the analog to normal nodes and provide a connection to a thermal circuit node.
- Internal thermal nodes can be used to build internal thermal circuits.
- Thermal branch currents introduce a new thermal branch current (heat flow) into the equation system.

The `init` method of a simple diode is shown in the following.

```
class MmxDiode : MmxDevice
{
    Param<double> Is, rs, V, I, T, gm;
    ...
}

vBoolean MmxDiode::init()
{
    addKeyword(Is, "Is", "Current");
    addKeyword(rs, "rs", "Resistance");

    readKeywords();

    addNode("A");          // Anode
    addNode("C");          // Cathode

    addThermalNode("T");

    if (rs > 0.0)
        addInternNode("intern");

    addOutput(V, "V", "Potential");
    addOutput(I, "I", "Current");
    addOutput(T, "T", "Temperature");
    addOutput(gm, "gm", "Conductance");

    return vTRUE;
}
```

The diode model implements the simple equations

$$I_d = I_s \cdot \left(\exp \left(\frac{q \cdot V_d}{k_B \cdot T} \right) - 1 \right) \quad (\text{B.1})$$

$$P_d = I_d \cdot V_d \quad (\text{B.2})$$

In addition, a series resistance r_s can be added. This is done via the keyword `rs` which is registered to be of quantity type "Resistance". Quantity types are available for many different physical quantities and provide access to SI-units, internal units, communication units and their respective scaling factors. Three different unit types are necessary because of the following reasons

- *SI-units* are used in all device models. All keywords, input and output parameters will be provided in SI-units.

- Properly scaled *internal units* are used in the simulator core of MINIMOS-NT to assure a stable equation system.
- *Communication units* are used for the communication with the user. They are the default units when reading values from the input deck if not overwritten there. When writing information to e.g., log-files quantities are written in communication units.

E.g., for concentrations the SI-units are m^{-3} , the internal units are $10^{19} cm^{-3}$ while the communication units are cm^{-3} . These quantity types guarantee consistent handling of the above units.

After registering the keywords with calls to the member function `addKeyword` the member function `readKeywords` is called which reads the values for this instance from the input deck. Then the two device nodes for the anode "A", the cathode "C" and the thermal node "T" are registered. Only when the user specified a series resistance an internal node is needed which is dynamically added. Finally, the instance output parameters are registered via calls to the member function `addOutput`. All parameters registered this way are written to the log-file as simulation result. In addition, they can be inquired within the input deck using the output function. The `init` method is called from the default constructor. For each iteration of the Newton method the `evaluate` method is called. Before each call `Y` and `rhs` are cleared and `x` initialized with proper values from the last solution. Furthermore, a special vector called `solv` is initialized. It states which quantities are really solved for to speed up calculation. The `evaluate` method is expected to fill in `Y`, `rhs` and the instance output parameters. In the following, a version stripped of all numerical subtleties is shown

```
vBoolean MmxDiode::evaluate()
{
    T = x[nT];
    V = x[n1] - x[n2];

    vDouble VTinv = Const.q / (Const.k * T);
    vDouble Ix = Is * Exp(V * VTinv);
    I = Ix - Is + V * gmin;
    gm = Ix * VTinv + gmin;

    Y[n1][n1] = gm;
    Y[n2][n2] = gm;
    Y[n1][n2] = -gm;
    Y[n2][n1] = -gm;

    rhs[n1] = -I;
    rhs[n2] = I;

    if (solv[nT])
    {
        vDouble dIdT = - V * gm / T;

        Y[n1][nT] = dIdT;
        Y[n2][nT] = - dIdT;
        rhs[nT] = I * V;
    }

    return vTRUE;
}
```

An additional feature of the Algorithm Library allows for interpreted models. From the users point of view these model are indistinguishable from built-in models except being slower by about a factor of 2. When they are compiled, no noticeable difference in performance

was found. This is because the `evaluate` method can be very easily mapped to C++ due to the similarity of MDL to C++. Providing the proper inheritance information and keywords is more complicated but fortunately only necessary once during initialization.

The following example shows the diode model as implemented in MDL syntax.

```
NewModel MdlDiode : Device
{
  Local
  {
    Parameter<double> VTinv, Ix, dIdT;
  }

  construct
  {
    key["Is"] = {{ Quantity Current }};
    key["rs"] = {{ Quantity Resistance }};

    call readKeywords;

    node["n1"] = {{ Voltage }};
    node["n2"] = {{ Voltage }};

    if (rs > 0)
      node["nT"] = {{ Thermal }};

    output["I"] = {{ Quantity Current }};
    output["V"] = {{ Quantity Voltage }};
    output["T"] = {{ Quantity Temperature }};
    output["gm"] = {{ Quantity Conductance }};

    call addNodesAndOutput;
  }

  evaluate
  {
    T = :x[nT];
    V = :x[n1] - x[n2];

    VTinv = Const.q / (T * Const.k);
    Ix = Is * exp(V * VTinv);
    I = Ix - Is + V * gmin;
    gm = Ix * VTinv + gmin;

    :Y[n1][n1] = gm;
    :Y[n2][n2] = gm;
    :Y[n1][n2] = -gm;
    :Y[n2][n1] = -gm;

    :rhs[n1] = -I;
    :rhs[n2] = I;

    if (solv[nT])
    {
      dIdT = - V * gm / T;

      :Y[n1][nT] = dIdT;
      :Y[n2][nT] = - dIdT;
      :rhs[nT] = I * V;
    }
  }
}
```

In the model constructor three lists are available to register keywords, nodes, and output parameters. For each argument of these lists, a local parameter is created which can be used in all other methods. The keyword list is the same as for the physical models. The node list registers the external and internal nodes of the device model. External nodes

are `Voltage` and `Temperature` for voltage and temperature nodes, respectively. Internal nodes of type `VoltageInternal` can be created which are not visible from outside. Branch currents and heat flows can be registered using `Current` and `HeatFlow`, respectively.

As interface parameters the solution of the previous iteration (`x`), the partial Jacobian matrix (`Y`) and the right hand side of the current iteration (`rhs`) are available. They are distinguished from the local parameters by the colon. The solution vector can be used to inquire the potentials and temperatures at the circuit nodes and the branch currents and heat flows of the device. The model must calculate the partial Jacobian matrix and the right hand side for the current operating point. This is done in the `evaluate` method.

Appendix C

Overview of the Input Deck Programming Language

MINIMOS-NT is equipped with a quite powerful control language called Input Deck Programming Language (IPL). During a simulation several decisions are imaginable, various parameters must be given. Using the IPL the user is able to customize his simulation by creating his own input deck files written in plain ASCII text. MINIMOS-NT provides default input deck files with standard settings. However, for standard simulations most of the IPL features are hidden. This section gives a rough overview of the most important features.

MINIMOS-NT uses IPL files as input files. First, the files are read and checked for correct syntax. All elements are stored in a kind of database and can be queried by the simulator afterwards.

Other input deck files can be inserted using the `#include` command, e.g.,

```
#include <defaults.ipd>
```

The basic unit is a *keyword*. MINIMOS-NT only reads those keywords, which are used for the current simulation type. Keywords may contain simple constants or complex expressions and can be of one of the following types:

Type	Example
Boolean	<code>a = true;</code>
Integer	<code>a = 3;</code>
Real	<code>a = 3.1415;</code>
Complex	<code>a = 4.3 + 3.1 j;</code>
Quantity	<code>a = 3.5 m;</code> <code>a = 7.38 "m/s";</code>
String	<code>a = "This is a string";</code>
Array	<code>a = [1, 2, 3];</code> <code>a = [1, "pi", 3 A];</code>

Table C.1: IPL types

Standard mathematical operations and functions are available which are defined for the types Integer, Real, and Complex in general. Additionally, several useful functions are defined. New functions can be simply defined by the user.

```

inc(x) = x + 1;
a = (17.2 + 22.9) * 2;           // 80.2
b = sin (a);                     // sin(80.2)
c = inc (b);                     // sin(80.2) + 1

```

Keywords can be grouped in arbitrarily nested sections enclosed in braces, e.g.,

```

X                                // section X
{
    a = yes;
    Y                            // subsection Y within section X
    {
        b = 3 V;
    }                            // end of subsection Y
    c = Y.b + 34 V;              // 37 V
}                                // end of section X

```

Keywords and sections have a so-called *full name* which also includes the location (directory path). In the example above the full name of b is ~X.Y.b. The tilde (~) denotes the root section hence this is called an absolute path. If the tilde is omitted, it is a *relative path* starting from the current position.

Sections may be inherited from base sections. Keywords and sections given in the base section are visible in the inherited section. They may be overwritten by a local value.

```

X
{
    a = yes;
    b = 3;
}

Y : X                            // inherit Y from X
{
    a = no;
}

```

Section Y contains a = no and b = 3.

Whenever a section is inherited from a base section, only variables given in the base section can be locally modified. New variables can be added by using the plus (+) sign. This is necessary in cases where keywords are not a priori known by the simulator (e.g., contacts).

```

X
{
    a = yes;
    b = 3;
}

Y : X                            // inherit Y from X
{
    +c = "Test";                 // add c to section Y
}

```

Section Y now contains a = yes, b = 3, and d = "Test".

Keywords can be classified by the access the simulator is allowed to perform: External keywords (defined by a leading ext which can be read and written, and 'normal' keywords (need not to be specified) which can be read only by the simulator, e.g.,

```

ext a1;                          // may be read and written by the simulator
a2 = 5;                          // may be read only by the simulator

```

In MINIMOS-NT most of the external keywords are located in the `Extern` section for convenience. It is important to note that all keywords are evaluated at run-time and may therefore vary with time in case they contain references to an external keyword.

```

Extern
{
    ext t = 0;                // the current simulation time
}

aux T = 300 ms;              // T is defined as a constant
a = sin(2*pi*Extern.t/T);    // depends on Extern.t which may vary

```

The qualifier `aux` denotes that the variable `T` cannot be inquired by the simulator. It is used in the default files to emphasize the fact that this is an auxiliary variable which will not be inquired and is just used to build more complex control expressions.

IPL supports comfortable unit handling. Units containing a division, multiplication, or power function must be double-quoted to avoid ambiguities.

```

s = 2 s;                      // s is the unit
m = 3 m/s;                    // error: (3 m)/s or 3 "m/s"
q = 3 "m/s";                  // OK
t = (s/1 s)*1 "m/s" + b;      // -> 4 "m/s"

```

Each quantity keyword has a default unit which is the SI-unit. Exceptions are given in Table C.2. In case of missing units these default units will be used.

Type	Default unit
length	"um"
volume	"um^3"
voltage	"V"
electric field	"V/cm"
velocity	"cm/s"
concentration	"cm^-3"
density of states	"cm^-3"
capture cross section	"cm^2"
thermal resistance	"K cm^2/W"
mobility	"cm^2/V s"
energy	"eV"

Table C.2: Default units

Appendix D

Iteration Schemes in MINIMOS-NT

The need for iteration schemes arises from the fact that, when solving very complex coupled equation systems, the solution can very often not be obtained from the available initial-guess as the region of attraction for the Newton scheme would be too small. Hence, the problem can be split into different levels of complexity with each of them using the previous level as an initial-guess to further refine the solution by applying more complicated models. This procedure will be called *iteration scheme* in the following. Typical iteration schemes are:

- For hydrodynamic simulations it is beneficial to use a drift-diffusion simulation as an initial-guess.
- For unipolar devices like MOSFETs it is sometimes useful to neglect the continuity equation for the minority carriers and to use a constant quasi-Fermi level approximation to calculate an initial-guess for the fully coupled system.
- Several relaxation schemes are known which solve subsets of the equation system alternately.

These iteration schemes are normally hard-coded in the simulator and can only be marginally adjusted by the developer. This makes it very cumbersome to implement new schemes or experiment with various different configurations. This is especially important when new or different equations are added since their influence on the coupled system can be more easily detected then. Hence it was decided to provide MINIMOS-NT with an interface so that iteration schemes can be arbitrarily programmed with several additional options making use of the features provided by the IPL.

An iteration scheme consists of arbitrarily nested iteration blocks. Each block can have subblocks which will be evaluated recursively. The following gives a short overview of the major features available for defining iteration blocks.

D.1 Iteration Blocks

Iteration blocks are implemented as IPL sections. They contain a predefined set of keywords to describe which equation system is to be used and how it is to be solved. All iteration blocks are inherited from a base iteration block which is defined as follows

```

GeneralBlock : ~IterateConfig, ~Extern
{
  models           = "";
  ignoreModels     = "";
  quantities       = "";
  ignoreQuantities = "";
  offDevices       = no;
  enter            = yes;
  failure          = iterationCount >= if (firstStep,
                                           iterationLimit,
                                           iterationLimitStep);
  while            = updateNorm > finalNorm && rhsNorm > rhsFinalNorm;
  ignoreDer        = "";
}

```

The base iteration block defines the default keywords for all iteration blocks. The iteration blocks are used to define the iteration sequence and are inherited from both the `IterateConfig` section and the `Extern` section to have simple access to the main external keywords. The `IterateConfig` section is defined as

```

IterateConfigDefaults
{
  maxDamp          = 1.0;
  minDamp          = 0.008;
  iterationLimit    = 500;
  iterationLimitStep = 200;
  finalNorm        = 1e-2;
  rhsFinalNorm     = 1e-10;
  transform        = yes;
  dampScheme       = "potential";
  ignoreDerNorm    = 1;
  delta            = 0.1;
}

IterateConfig : IterateConfigDefaults;

```

This somehow tricky inheritance allows to break up the predefined inheritance chain and to customize the default settings for personal needs. In IPL the ancestors of a section can be changed as long as no keywords have been changed. A private version of `IterateConfig` can be created to substitute the default inheritance.

```

MyIterateConfig : IterateConfigDefaults
{
  iterationLimit      = 100;
  iterationLimitStep  = 50;
  dampScheme         = "bank";
}

IterateConfig : MyIterateConfig;

```

The iteration blocks are also used to define the iteration sequence and are all in one or the other way inherited from `GeneralBlock`. The most important blocks are `Loop`, `DDBlock`, and `HDBlock` which in turn are used to build loops, DD, or HD iteration schemes.

```

Loop : GeneralBlock;

DDBlock : GeneralBlock
{
  models      = ~Eas.DD.TransOrDC;
  quantities  = ~Quantities.DD;
}

```

```

HDBlock : GeneralBlock
{
    models      = ~Eas.HD.TransOrDC;
    quantities  = ~Quantities.HD;
    ignoreDer   = if (updateNorm > ignoreDerNorm, "EleDEtm,HolDHtm", "");
}

```

Each iteration block can contain other arbitrarily nested sub-blocks. For each block the equations to use are defined which will be solved by the simulator using a Newton iteration scheme until a block-specific termination criterion is satisfied. The criteria can be arbitrary expressions containing norms, iteration counters, time step information, and much more. After each iteration the sub-blocks are entered recursively. Blocks can be empty to form a linear sequence of sub-blocks only. The following sections give more details on what can be defined for each iteration block.

D.1.1 Equations and Quantities

The user can select which equation set is to be solved. To allow for a more flexible equation assembly, the equations are split into their constituent terms which can be selected independently. This allows for implementing iteration schemes as reported in [13] where, e.g., the recombination terms in the continuity equations or the contribution of the carrier densities to the charge in Poisson's equation are neglected.

When used for the first time, for each equation the appropriate quantities are automatically created and initialized with proper initial values. These initial values are either taken from a previous simulation or are calculated to give reasonable defaults as follows.

- The potential is initialized with the built-in potential (cf. equation (6.9)).
- The initial carrier concentration is calculated using a charge neutrality assumption.
- The lattice temperature is set to the average of the contact temperatures.
- The carrier temperatures are set equal to the lattice temperature.

The quantities are added to the solution vector and remain there until the end of the simulation. Each quantity can be separately marked for solving in the next Newton iteration. Only the equations associated with the marked quantities will be used to build up the equation system and of course only the marked quantities will be updated after solving the equation system. However, all quantities already available in the solution vector can be used to calculate the system matrix. The available equations are summarized in the following. A more detailed description of the equations can be found in Chapter 3. The equations are given in differential form, whereas they are implemented in integral form using the box integration scheme [17].

$$\underbrace{\text{div}(\varepsilon \cdot \text{grad} \psi)}_{P_{\text{poiss}}} = \underbrace{q \cdot n}_{P_n} - \underbrace{q \cdot p}_{P_p} - \underbrace{q \cdot C}_{P_C} - \underbrace{q \cdot C_{\text{Trap}}}_{P_{C_{\text{Trap}}}} \quad (\text{D.1})$$

$$\underbrace{\text{div} \mathbf{J}_n}_{E_{\text{div}}} = \underbrace{q \cdot R}_{R} + \underbrace{q \cdot \frac{\partial n}{\partial t}}_{E_t} \quad (\text{D.2})$$

$$\underbrace{\text{div} \mathbf{J}_p}_{H_{\text{div}}} = - \underbrace{q \cdot R}_{R} - \underbrace{q \cdot \frac{\partial p}{\partial t}}_{H_t} \quad (\text{D.3})$$

$$\underbrace{q \cdot n}_{P_n^{QFL}} = q \cdot N_C \cdot \exp \left(- \frac{q \cdot (\varphi_n - \psi) + E_C}{k_B \cdot T_L} \right) \quad (\text{D.4})$$

$$\underbrace{q \cdot p}_{P_p^{QFL}} = q \cdot N_V \cdot \exp \left(\frac{q \cdot (\varphi_p - \psi) + E_V}{k_B \cdot T_L} \right) \quad (\text{D.5})$$

$$\underbrace{\text{div} \mathbf{S}_n}_{S_{n,\text{div}}} = \underbrace{\text{grad} \left(\frac{E_C}{q} - \psi \right) \cdot \mathbf{J}_n}_{S_{n,J}} - \frac{3 \cdot k_B}{2} \cdot \left(\underbrace{\frac{\partial (n \cdot T_n)}{\partial t}}_{S_{n,t}} + \underbrace{R \cdot T_n}_{S_{n,R}} + \underbrace{n \cdot \frac{T_n - T_L}{\tau_{\epsilon,n}}}_{S_{n,\tau}} \right) \quad (\text{D.6})$$

$$\underbrace{\text{div} \mathbf{S}_p}_{S_{p,\text{div}}} = \underbrace{\text{grad} \left(\frac{E_V}{q} - \psi \right) \cdot \mathbf{J}_p}_{S_{p,J}} - \frac{3 \cdot k_B}{2} \cdot \left(\underbrace{\frac{\partial (p \cdot T_p)}{\partial t}}_{S_{p,t}} + \underbrace{R \cdot T_p}_{S_{p,R}} + \underbrace{p \cdot \frac{T_p - T_L}{\tau_{\epsilon,p}}}_{S_{p,\tau}} \right) \quad (\text{D.7})$$

$$\underbrace{\text{div} \mathbf{S}_L}_{T_{\text{div}}} = \underbrace{H}_{T_H} - \underbrace{\rho_L \cdot c_L \cdot \frac{\partial T_L}{\partial t}}_{T_t} \quad (\text{D.8})$$

The unknown quantities of this equation system are the electrostatic potential ψ , the electron and hole concentrations n and p , the carrier temperatures T_n and T_p , and the lattice temperature T_L . C denotes the net concentration of the ionized dopants, C_{Trap} denotes the concentration of the occupied traps, ε is the dielectric permittivity of the semiconductor, and R is the net recombination rate. Since only a few combination of terms yield meaningful results, these terms are collected to so-called model *groups* to simplify the selection mechanism.

Currently two different versions of E_{div} exist depending on the current model used ((3.5) or (3.7)), namely E_{div}^{DD} and E_{div}^{HD} , vice-versa for holes. The groups resulting from the equation assembly models given above are listed in Table D.2.

Group name	Equation assembly models
electrons	E_{div}^{DD}, P_n
electronsTrans	E_t
electronsQFL	P_n^{QFL}
electronsHD	$E_{\text{div}}^{HD}, P_n, S_{n,\text{div}}, S_{n,J}, S_{n,\tau}$
electronsHDTrans	$E_t, S_{n,t}$

Group name	Equation assembly models
holes	H_{div}^{DD}, P_h
holesTrans	H_t
holesQFL	P_p^{QFL}
holesHD	$H_{div}^{HD}, P_p, S_{p,div}, S_{p,J}, S_{p,\tau}$
holesHDTrans	$H_t, S_{p,t}$
potential	P_{poiss}, P_C
<recombination>	$R,, P_{C_{Trap}}, S_{n,R}, S_{p,R}$
selfHeating	T_{div}, T_H
selfHeatingTrans	T_t

Table D.2: Example model groups

The important recombination groups are written as <recombination>, denoting that several different mechanisms exist which split the terms given above into terms handling only one single effect. Currently available are Shockley-Read-Hall recombination, Auger recombination, direct recombination, band to band tunneling, and impact ionization. In addition, interface groups are available which handle the interface and boundary conditions. These groups can be considered to be complete in the following sense: if it is desired to simulate the Poisson equation, e.g., inside an oxide, only the potential group is necessary. This is done by the following iteration block:

```
Iterate
{
  Scheme
  {
    models = "potential,potentialInt";
    ...
  }
}
```

If one is also interested in the electron concentration, e.g., in the channel of a HEMT, the electron group has to be added:

```
Iterate
{
  Scheme
  {
    models = "potential,potentialInt,electrons,electronsInt";
    ...
  }
}
```

If the derivative of the electron concentration with respect to time is relevant, e.g., for a transient simulation, the electronTrans group has to be added:

```
Iterate
{
  Scheme
  {
    models = "potential,potentialInt, "
            "electrons,electronsInt,electronsTrans";
    ...
  }
}
```

Group Name	Created Quantity Classes
electrons	Potential ElectronConcentration
electronsTrans	ElectronConcentration
electronsQFL	Potential ElectronConcentration
electronsHD	Potential ElectronConcentration ElectronTemperature
electronsHDTrans	ElectronConcentration
holes	Potential HoleConcentration
holesTrans	HoleConcentration
holesQFL	Potential HoleConcentration
holesHD	Potential HoleConcentration HoleTemperature
holesHDTrans	HoleConcentration
potential	Potential
<recombination>	Potential ElectronConcentration HoleConcentration
selfHeating	LatticeTemperature
selfHeatingTrans	LatticeTemperature
potentialInt	Potential on both segments
electronsInt	Potential on both segments ElectronConcentration on both segments
electronsHDInt	Potential on both segments ElectronConcentration on both segments ElectronTemperature on both segments
terminal	ContactVoltage ContactCurrent ContactCharge ContactTemperature BoundaryCurrent BoundaryElectronCurrent BoundaryHoleCurrent BoundaryConductionCurrent BoundaryDisplacementCurrent

Table D.3: Quantity classes and the groups which create them

Several tests are implemented to simplify model assembly. Among these test are the following:

- Each group has a list of material classes for which it can be used, e.g., the electrons group will not be used for materials of class Insulator.
- The carrier concentrations are controlled either by one of the transport equations or by the QFL approximation. So the groups electrons, electronsHD, and elec-

tronsQFL are implemented to be mutually exclusive. E.g., if one of the transport equations is available, the QFL model is automatically disabled. This is done without a warning to reduce the number of `if` statements in the input deck.

- From (D.2) and (D.3) it follows that recombination and generation modeling is only consistent with the rest of the equation system when *both* carrier types are properly modeled by transport equations. Hence all recombination models check for the existence of the two continuity equations. Again, in case this check fails, no warning is issued, the model is simply disabled.
- The interface and contact models automatically check for a proper material class combination in order to become active. E.g., the carrier contact model checks for an ideal-conductor semiconductor interface. Hence the interface groups can be used for all boundaries without worrying about further details.

As pointed out above, each group automatically creates the quantities which form the unknowns of the equation system. Each quantity belongs to a so-called quantity class, e.g., the potentials inside all insulator, conductor, and semiconductor segments belong to the quantity class `Potential`. Table D.3 lists the quantity classes and the model groups which create them. It is important to note that each quantity is created and initialized when it is first requested by one of the model groups. The mere creation of a quantity does not imply that there is a controlling equation available. In case the user registers only the `electrons` group on one segment, the quantities ψ and n are created. The potential ψ is initialized to the built-in potential whereas the electron concentration n is computed using a charge-neutrality assumption. As there has been no Poisson equation requested (group `Potential` has not been used), the electron continuity equation will be solved stand-alone.

The same is true for interface models. When using for instance the HD interface group, carrier temperature quantities on both sides of the interface are created and are initialized to the lattice temperature. However, by simply creating these quantities the interface model does not make any assumptions about the transport models used on either side. It is possible to use a DD model on one side and a HD model on the other side which implies an equilibrium boundary condition for the carrier temperatures as the carrier temperatures are set equal to the lattice temperature in this case.

The `terminal` group needs some further attention. It creates an abundance of quantities which might seem unnecessary at a first glance. The quantities whose names start with `Contact` are the terminal quantities which are used to formulate the boundary conditions. The quantities with names beginning with `Boundary` are supplementary quantities which represent the current components and are used to simplify calculation of the contact current.

For mixed-mode simulations the basic semiconductor equations are augmented by the circuit equations resulting from the modified nodal approach (MNA) and the MNA equations for the thermal equivalent circuit. It is worthwhile to note that for the KCL equation the node voltage vector is split into two quantity classes namely node voltages (`NodeVoltage`) and fixed node voltages (`FixNodeVoltage`), respectively. Since each quantity class can be marked separately for the solving process, this separation has the major advantage that the fixed node voltage class can be arbitrarily solved or not. One way to utilize this feature is to emulate the behavior of typical circuit simulators like SPICE. To aid convergence, SPICE allows the user to set some critical node voltages to a constant value, iterate until

convergence, release the critical node voltages, and continue iteration until final convergence. This has been implemented in the `MixedDD` and `MixedHD` iteration schemes.

```
Schemes
{
  MixedDD
  {
    DDPre : DDBlock
    {
      ignoreQuantities = "FixNodeVoltage";
    }
    DDPre : DDBlock;
  }
}

Iterate
{
  Scheme : ~Schemes.MixedDD;
}
```

However, due to the generality of this concept, it is possible to design own iteration schemes with a different behavior. One useful possibility might be to fix some node voltages only at distinct time points to their initial or last value (which is internally the same).

D.1.1.1 Segment and Device Specific Selection

The keywords `models`, `quantities`, and `ignoreQuantities` employ a distinct syntax to address a specific device or segment. Consider the following assignment to the `models` keyword:

```
models = "potential,electron[NMOS1(Channel)],hole[PMOS3(Semi)]";
```

In words: Use the `potential` group for devices on all segments (equivalent to `potential[* (*)]`), use the `electron` group for the device named `NMOS1` on segment `Channel` only, and use the `hole` group for the device named `PMOS3` on segment `Semi` only. The general syntax is:

```
keyword = "quantity[devices(segments)]";
```

Example:

```
keyword = "quantity[T*(Cap*,Channel),BJT*(Base,Coll*),NMOS]";
```

In words: Take quantity `quantity` for all devices with names beginning with the letter `T` on all segments beginning with `Cap` and the segment `Channel`, for all devices beginning with the letters `BJT` on all segments beginning with `Coll` and the segment `Base`, and for all segments of the device name `NMOS`.

The exclamation mark (!) can be used as negation for segment names and applies for all following segment names, e.g., use `quantity` on all segments *not* named `Channel` and `Spacer`:

```
keyword = "quantity[devices(!Channel,Spacer)]";
```

Only one exclamation mark is possible and must be given before any segment names. For most of the simulations this high flexibility might be an overkill. To simplify usage for standard cases auxiliary variables located in the `Phys` section of each device are available. These keywords are used to create the default model lists found in the `Eas` section.

Using the complete DD equation set for all segments the following model list is generated:

```
models = "terminal,potential,potentialInt, "
         "electrons,electronsInt,holes,holesInt,srh";
```

If only the majority carriers should be used for a CMOS inverter the following list is generated

```
models = "terminal,potential,potentialInt, "
         "electrons[NMOS],electronsInt[NMOS],holes[PMOS],holesInt[PMOS]";
```

For the μ A709 in Chapter 7 self-heating was only considered for the transistors T_1 , T_2 , T_9 , and T_{15} using the model list

```
models = "terminal,potential,potentialInt, "
         "electrons,electronsInt,holes,holesInt,selfHeating[T1,T2,T9,T15]";
```

D.1.2 Termination Criterion

The termination criterion defines under which conditions the block can be considered to be successfully solved. This is normally the case when the update and residuum norms fall under user-specified values.

```
Block
{
    while = updateNorm > finalNorm && rhsNorm > rhsFinalNorm;
}
```

Another possibility is to terminate a block when a maximum block iteration counter is exceeded as is the case, e.g., for the Gummel scheme [27] where each iteration block is solved only once.

```
Block
{
    while = iterationCount < 1;
}
```

The keywords `updateNorm` and `rhsNorm` are inherited from the `Extern` section, while `finalNorm` and `rhsFinalNorm` stem from `IterateConfig`. As `updateNorm` and `rhsNorm` contain *all* quantities of the solution vector their use might be problematic. Although the quantities are scaled to be approximately of the same order the range of possible values is still different. It might therefore be useful for special applications to refine the termination criterion by using only parts of the solution vector. These parts can be accessed through one of the following functions:

- `updateNorm2(quantityList)` gives the E_2 norm of the updates.
- `updateNormMax(quantityList)` gives the E_∞ norm of the updates.

- `rhsNorm2(quantityList)` gives the E_2 norm of the right-hand-side of the respective part of the equation system.
- `rhsNormMax(quantityList)` gives the E_∞ norm of the right-hand-side of the respective part of the equation system.

In all cases *quantityList* is a comma-separated list of quantity class names. Optional arguments of all functions are the name of the block and whether the value should be returned in internal or SI units. E.g., as the potential is scaled to the temperature voltage V_T , the value returned is normally a given as multiples of V_T while when SI units are requested the values are scaled back with V_T . An implementation of Gummel's scheme might look as follows.

```
DD_Gummel
{
  Ele : ~Iterate.Blocks.DDBlock
  {
    ignoreQuantities = "Potential,*Hole*";
    while             = iterationCount < 1;
  }
  Hol : Ele
  {
    ignoreQuantities = "Potential,*Electron*";
  }
  Pot : Ele
  {
    ignoreQuantities = "*Electron*,*Hole*";
  }

  while = updateNorm2("", block = "Ele") +
          updateNorm2("", block = "Hol") +
          updateNorm2("", block = "Pot") > finalNorm;
}
```

In the first block (Ele) only the electron continuity equation is solved. This is done by using all models of the DDBlock (keyword `models` not locally overwritten) but ignoring the potential and all quantity classes matching `*Hole*` which is true for `HoleConcentration` and `BoundaryHoleCurrent`. The latter must be ignored in order to get proper current components as the currents are zero when not solving the continuity equations.

The other blocks solve the hole continuity equation and Poissons equation and for the termination criterion a simple algebraic sum of the block-norms is used.

D.1.3 Failure Criterion

The failure criterion defines under which conditions a user-defined error handling or recovery should be invoked, e.g., when the limit for the iteration counter is exceeded. The simulation could then be restarted using, e.g., a smaller timestep.

```
Block
{
  failure = iterationCount >= if (firstStep,
                                iterationLimit,
                                iterationLimitStep);
}
```

D.1.4 Enter Criterion

The enter criterion can be used to enable or disable complete iteration blocks and their sub-blocks. An example might be a HD simulation which is preceded by a DD simulation to provide a good initial-guess. The iteration block which provides the drift-diffusion initial-guess is disabled for all subsequent simulations.

```
Schemes
{
  HD
  {
    DD : DDBlock
    {
      enter = firstStep;
    }
    HD : HDBlock;
  }
}

Iterate
{
  Scheme : ~Schemes.MixedDD;
}
```

D.1.5 Damping Scheme

For each iteration block a separate damping scheme can be selected which can vary depending on the iteration progress. All implemented damping schemes have in common that they damp the solution vector by a damping factor d . The computation of the damping factor d depends on the damping scheme selected (cf. Chapter 6).

D.1.6 Full versus Partial System Matrix

One problem with the Newton method arises from the fact, that the Jacobian matrix cannot be calculated exactly. Especially as the model parameters depend on many other quantities consideration of all derivatives would be very complicated. These missing derivatives can cause divergence of the Newton method at the beginning of the iteration. As reported in [36] it is advantageous to ignore some of the derivatives of the system matrix at the beginning of the iteration process. This reduces the coupling of the equations and is especially useful in case of HD simulations where the derivatives of the carrier concentrations with respect to the carrier temperatures deteriorate the convergence properties. As default we ignore $\partial n / \partial T_n$ and $\partial p / \partial T_p$ when $E_2(\mathbf{u}) > E_{th} = 1.0 \cdot V_T$. This is specified by

```
HDBlock
{
  ignoreDer = if (updateNorm > ignoreDerNorm, "EleDEtm,HolDHtm", "");
}
```


Bibliography

- [1] P. Antognetti and G. Massobrio. *Semiconductor Device Modeling with SPICE*. McGraw-Hill, 1988.
- [2] G. Baccarani and M.R. Wordeman. An Investigation of Steady-State Velocity Overshoot in Silicon. *Solid-State Electron.*, 28(4):407–416, 1985.
- [3] K. Blotekjaer. Transport Equations for Electrons in Two-Valley Semiconductors. *IEEE Trans. Electron Devices*, ED-17(1):38–47, 1970.
- [4] H. Brech. *Optimization of GaAs Based High Electron Mobility Transistors by Numerical Simulation*. Dissertation, Technische Universität Wien, 1998.
- [5] D.M. Caughey and R.E. Thomas. Carrier Mobilities in Silicon Empirically Related to Doping and Field. *Proc. IEEE*, 52:2192–2193, 1967.
- [6] J. Demel. *JANAP – Ein Programm zur Simulation von elektrischen Netzwerken*. Dissertation, Technische Universität Wien, 1989.
- [7] C.H. Diaz, S.M. Kang, and C. Duvvury. *Modeling of Electrical Overstress in Integrated Circuits*. Kluwer, 1995.
- [8] G. Digele, S. Lindenkreuz, and E. Kasper. Fully Coupled Dynamic Electro-Thermal Simulation. *IEEE Trans. VLSI Systems*, 5(3):250–257, 1997.
- [9] S.W. Director. *Circuit Theory: A Computational Approach*. Wiley, New-York, 1975.
- [10] M. Driessen and H.A. van der Vorst. Bi-CGSTAB in Semiconductor Modeling. In W. Fichtner and D. Aemmer, editors, *Simulation of Semiconductor Devices and Processes*, pages 45–54, Konstanz, 1991. Hartung-Gorre.
- [11] S.H.K. Embabi. *Digital BiCMOS Integrated Circuit Design*. Kluwer, 1993.
- [12] W. Engl, R. Laur, and H.K. Dirks. MEDUSA - A Simulator for Modular Circuits. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 1(2):85–93, 1982.
- [13] W.L. Engl and H. Dirks. Numerical Device Simulation Guided by Physical Approaches. In B.T. Browne and J.J. Miller, editors, *Numerical Analysis of Semiconductor Devices and Integrated Circuits*, volume I, pages 65–93, Dublin, 1979. Boole Press.
- [14] W.L. Engl and H. Dirks. Functional Device Simulation by Merging Numerical Building Blocks. In B.T. Browne and J.J. Miller, editors, *Numerical Analysis of Semiconductor Devices and Integrated Circuits*, volume II, pages 34–62, Dublin, 1981. Boole Press.

-
- [15] C. Fischer. *Bauelementsimulation in einer computergestützten Entwurfsumgebung*. Dissertation, Technische Universität Wien, 1994. <http://www.iue.tuwien.ac.at>.
 - [16] M.V. Fischetti. Monte Carlo Simulation of Transport in Technologically Significant Semiconductors of the Diamond and Zinc-Blende Structures—Part I: Homogeneous Transport. *IEEE Trans.Electron Devices*, 38(3):634–649, 1991.
 - [17] A.F. Franz, G.A. Franz, S. Selberherr, C. Ringhofer, and P. Markowich. Finite Boxes - A Generalization of the Finite Difference Method Suitable for Semiconductor Device Simulation. *IEEE Trans.Electron Devices*, ED-30(9):1070–1082, 1983.
 - [18] K. Fukahori and P.R. Gray. Computer Simulation of Integrated Circuits in the Presence of Electrothermal Interaction. *IEEE J.Solid-State Circuits*, SC-11(6):834–846, 1976.
 - [19] T. Grasser. Ein Kontaktmodell zur Simulation von Poly-Emitter-Bipolar-Transistoren. Diplomarbeit, Technische Universität Wien, 1995.
 - [20] T. Grasser, H. Kosina, and S. Selberherr. Consistent Comparison of Drift-Diffusion and Hydro-Dynamic Device Simulation. In *Simulation of Semiconductor Processes and Devices*, Kyoto, 1999. accepted for publication.
 - [21] T. Grasser, V. Palankovski, G. Schrom, and S. Selberherr. Hydrodynamic Mixed-Mode Simulation. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 247–250. Springer, Wien, New York, 1998.
 - [22] T. Grasser, R. Strasser, M. Knaipp, K. Tsuneno, H. Masuda, and S. Selberherr. Calibration of a Mobility Model for Quartermicron CMOS Devices. In R. Zobel and D. Moeller, editors, *12th European Simulation Multiconference – Simulation: Past, Present and Future*, pages 75–77, Manchester, UK, 1998. Society for Computer Simulation.
 - [23] T. Grasser, R. Strasser, M. Knaipp, K. Tsuneno, H. Masuda, and S. Selberherr. Device Simulator Calibration for Quartermicron CMOS Devices. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 93–96. Springer, Wien, New York, 1998.
 - [24] T. Grasser, K. Tsuneno, H. Masuda, and S. Selberherr. Mobility Parameter Tuning for Device Simulation. In A. Touboul, Y. Danto, J.-P. Klein, and H. Grünbacher, editors, *28th European Solid-State Device Research Conference*, pages 336–339, Bordeaux, France, 1998. Editions Frontieres.
 - [25] P.R. Gray and R.G. Meyer. *Analysis and Design of Analog Integrated Circuits*. Wiley, 1993.
 - [26] M. Gritsch. Implementation of a Non-Parabolic Energy-Transport Model. Diplomarbeit, Technische Universität Wien, 1999.
 - [27] H.K. Gummel. A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations. *IEEE Trans.Electron Devices*, ED-11:455–465, 1964.
 - [28] H.K. Gummel and S. Selberherr. Private Communication, 1987.
 - [29] W. Hänsch and M. Miura-Mattausch. The Hot-Electron Problem in Small Semiconductor Devices. *J.Appl.Phys.*, 60(2):650–656, 1986.
-

-
- [30] W. Hänsch, M. Orlowski, and W. Weber. The Hot-Electron Problem in Submicron MOSFET. In J.-P. Nougier and D. Gasquet, editors, *18th European Solid State Device Research Conference*, volume 49 of *Journal de Physique*, pages 597–606, Les Ulis Cedex, France, 1988. les éditions de physique.
 - [31] C.W. Ho, A.E. Ruehli, and P.A. Brennan. The Modified Nodal Approach to Network Analysis. *IEEE Trans.Circuits and Systems*, CAS-22(6):504–509, 1975.
 - [32] C.W. Ho, D.A. Zein, A.E. Ruehli, and P.A. Brennan. An Algorithm for DC Solutions in an Experimental General Purpose Interactive Circuit Design Program. *IEEE Trans.Circuits and Systems*, CAS-24(8):416–421, 1971.
 - [33] IBM. Advanced Statistical Analysis Program (ASTAP), Program Reference Manual. Technical Report SH20-1118-0, IBM, 1973.
 - [34] ISE Integrated Systems Engineering. *ISE TCAD Manuals vol. 4, release 4*, 1997.
 - [35] G. Kaiblinger-Grujin, T. Grasser, and S. Selberherr. A Physically-Based Electron Mobility Model for Silicon Device Simulation. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 312–215. Springer, Wien, New York, 1998.
 - [36] M. Knaipp. *Modellierung von Temperatureinflüssen in Halbleiterbauelementen*. Dissertation, Technische Universität Wien, 1998. <http://www.iue.tuwien.ac.at>.
 - [37] M. Knaipp, T. Grasser, and S. Selberherr. A Physically Based Substrate Current Simulation. In *27th European Solid-State Device Research Conference*, pages 196–199, Stuttgart, Germany, 1997. Editions Frontieres.
 - [38] S.S. Lee and D.J. Allstot. Electrothermal Simulation of Integrated Circuits. *IEEE J.Solid-State Circuits*, 28(12):1283–1292, 1993.
 - [39] K. Mayaram and D.O. Pederson. Coupling Algorithms for Mixed-Level Circuit and Device Simulation. *IEEE Trans.Computer-Aided Design*, 11(8):1003–1012, 1992.
 - [40] J.R.F. McMacken and S.G. Chamberlain. CHORD: A Modular Semiconductor Device Simulation Development Tool Incorporating External Network Models. *IEEE Trans.Computer-Aided Design*, 8(8):826–836, 1989.
 - [41] R. Mlekus. *Object-Oriented Algorithm and Model Management in TCAD Applications*. Dissertation, Technische Universität Wien, 1999. <http://www.iue.tuwien.ac.at>.
 - [42] P.C. Munro and F.Q. Ye. Simulating the Current Mirror with a Self-Heating BJT Model. *IEEE J.Solid-State Circuits*, 26(9):1321–1324, 1991.
 - [43] L.W. Nagel. SPICE2: A Computer Program to Simulate Semiconductor Circuits. Technical Report UCB/ERL M520, University of California, Berkeley, 1975.
 - [44] B. Neinhüs, P. Graf, S. Decker, and B. Meinerzhagen. Examination of Transient Drift-Diffusion and Hydrodynamic Modeling Accuracy for SiGe HBTs by 2D Monte-Carlo Device Simulation. In *27th European Solid-State Device Research Conference*, pages 188–191, Stuttgart, Germany, 1997. Editions Frontieres.
 - [45] K. Nemeth. On the Analysis of Nonlinear Resistive Networks Considering the Effect of Temperature. *IEEE J.Solid-State Circuits*, 1(1):550–552, 1976.
-

-
- [46] V. Palankovski, T. Grasser, and S. Selberherr. SiGe HBT in Mixed-Mode Device and Circuit Simulation. In *WOCSDICE*, pages 145–146, Berlin, 1998.
 - [47] M.R. Pinto. *PISCES IIB*. Stanford University, 1985.
 - [48] M. Reiser. A Two-Dimensional Numerical FET Model for DC, AC, and Large-Signal Analysis. *IEEE Trans. Electron Devices*, ED-20:35–44, 1973.
 - [49] J.G. Rollins and J. Choma. Mixed-Mode PISCES-SPICE Coupled Circuit and Device Solver. *IEEE Trans. Computer-Aided Design*, 7:862–867, 1988.
 - [50] M. Rottinger, T. Simlinger, and S. Selberherr. Two-dimensional transient simulation of charge-coupled devices using MINIMOS NT. In H. Ryssel and P. Pichler, editors, *Simulation of Semiconductor Devices and Processes*, volume 6, pages 440–443, Wien, 1995. Springer.
 - [51] H. Schaumburg. *Halbleiter*. Teubner Stuttgart, 1991.
 - [52] G. Schrom. *Ultra-Low-Power CMOS Technology*. Dissertation, Technische Universität Wien, 1998. <http://www.iue.tuwien.ac.at>.
 - [53] H.W. Schüßler. *Netzwerke, Signale und Systeme 1*. Springer, 1991.
 - [54] K. Seeger. *Semiconductor Physics*. Springer, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1989.
 - [55] S. Selberherr. *Zweidimensionale Modellierung von MOS-Transistoren*. Dissertation, Technische Universität Wien, 1981.
 - [56] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer, 1984.
 - [57] S. Selberherr, W. Hänsch, M. Seavey, and J. Slotboom. The Evolution of the MINIMOS Mobility Model. *Solid-State Electron.*, 33(11):1425–1436, 1990.
 - [58] S. Selberherr, A. Schütz, and H.W. Pötzl. MINIMOS—A Two-Dimensional MOS Transistor Analyzer. *IEEE Trans. Electron Devices*, ED-27(8):1540–1550, 1980.
 - [59] National Semiconductors. Product Folder. <http://www.nsc.com/pf/LM/LM709.html>, 1999.
 - [60] B.J. Sheu, D.L. Scharfetter, P.K. Ko, and M.C. Jeng. BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors. *IEEE J. Solid-State Circuits*, SC-22(4):558–566, 1987.
 - [61] T. Simlinger. *Simulation von Heterostruktur-Feldeffekttransistoren*. Dissertation, Technische Universität Wien, 1996. <http://www.iue.tuwien.ac.at>.
 - [62] T. Simlinger, R. Deutschmann, C. Fischer, H. Kosina, and S. Selberherr. Two-Dimensional Hydrodynamic Simulation of High Electron Mobility Transistors Using a Block Iterative Scheme in Combination with Full Newton Method. In G.L. Baldwin, Z. Li, C.C. Tsai, and J. Zhang, editors, *Fourth Int. Conf. on Solid-State and Integrated-Circuit Technology*, pages 589–591, Beijing, China, 1995.
 - [63] T. Simlinger, M. Rottinger, and S. Selberherr. A Method for Unified Treatment of Interface Conditions Suitable for Device Simulation. In *Simulation of Semiconductor Processes and Devices*, pages 173–176, Cambridge, Massachusetts, 1997.
-

- [64] J.E. Solomon. The Monolithic Op Amp: A Tutorial Study. *IEEE J.Solid-State Circuits*, SC-9(6):314–332, 1974.
- [65] H. Spiro. *Simulation integrierter Schaltungen*. Oldenburg, 1990.
- [66] A. Stach. Simulation von MOSFET-Schaltungen. Diplomarbeit, Technische Universität Wien, 1995.
- [67] K. Starnberger. Interpolationsbasierte Modellierung von MOS-Transistoren. Diplomarbeit, Technische Universität Wien, 1996.
- [68] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 1986.
- [69] Technology Modeling Associates, Inc., Palo Alto, CA. *TMA MEDICI, Two-Dimensional Device Simulation Program, Version 2.0*, September 1994.
- [70] U. Tietze and C. Schenk. *Halbleiter-Schaltungstechnik*. Springer, Berlin, 1971.
- [71] R.L. Treadway. DC Analysis of Current Mode Logic. *IEEE Circuits and Devices Magazine*, pages 21–35, March 1989.
- [72] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1983.
- [73] W.T. Weeks, A.J. Jimenez, G.W. Mahoney, and D. Mehta. Algorithms for ASTAP – A Network-Analysis Program. *IEEE Trans.Circuit Theory*, CT-20(4):628–634, 1973.
- [74] N.H.E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison Wesley, second edition, 1993.
- [75] A. Wolf. Automatic Binding of SPICE Models to MINIMOS-NT. Diplomarbeit, Technische Universität Wien, 1999.
- [76] Z. Yu, R.W. Dutton, and M. Vanzi. An Extension to Newton’s Method in Device Simulators – On an Efficient Algorithm to Evaluate Small-Signal Parameters and to Predict Initial Guess. *IEEE Trans.Computer-Aided Design*, CAD-6(1):41–45, 1987.

Own Publications

- [O1] T. Grasser, H. Kosina, and S. Selberherr. Consistent Comparison of Drift-Diffusion and Hydro-Dynamic Device Simulation. In *Simulation of Semiconductor Processes and Devices*, Kyoto, 1999. accepted for publication.
- [O2] T. Grasser, V. Palankovski, G. Schrom, and S. Selberherr. Hydrodynamic Mixed-Mode Simulation. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 247–250. Springer, Wien, New York, 1998.
- [O3] T. Grasser, R. Strasser, M. Knaipp, K. Tsuneno, H. Masuda, and S. Selberherr. Calibration of a Mobility Model for Quartermicron CMOS Devices. In R. Zobel and D. Moeller, editors, *12th European Simulation Multiconference – Simulation: Past, Present and Future*, pages 75–77, Manchester, UK, 1998. Society for Computer Simulation.
- [O4] T. Grasser, R. Strasser, M. Knaipp, K. Tsuneno, H. Masuda, and S. Selberherr. Device Simulator Calibration for Quartermicron CMOS Devices. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 93–96. Springer, Wien, New York, 1998.
- [O5] T. Grasser, K. Tsuneno, H. Masuda, and S. Selberherr. Mobility Parameter Tuning for Device Simulation. In A. Touboul, Y. Danto, J.-P. Klein, and H. Grünbacher, editors, *28th European Solid-State Device Research Conference*, pages 336–339, Bordeaux, France, 1998. Editions Frontieres.
- [O6] G. Kaiblinger-Grujin, T. Grasser, and S. Selberherr. A Physically-Based Electron Mobility Model for Silicon Device Simulation. In K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, pages 312–215. Springer, Wien, New York, 1998.
- [O7] M. Knaipp, T. Grasser, and S. Selberherr. A Physically Based Substrate Current Simulation. In *27th European Solid-State Device Research Conference*, pages 196–199, Stuttgart, Germany, 1997. Editions Frontieres.
- [O8] V. Palankovski, T. Grasser, and S. Selberherr. SiGe HBT in Mixed-Mode Device and Circuit Simulation. In *WOCSDICE*, pages 145–146, Berlin, 1998.

Curriculum Vitae

- April 28th, 1970 Born in Vienna, Austria
- June 1988 High School Graduation, BRG18 Wien
- October 1988 Enrolled in Electrical Engineering at the Technical University of Vienna
- November 1991 Passed 1. Diplomprüfung at the Technical University of Vienna (with honors)
- February 1994 Award “Leistungsstipendium” for Top Ranking at the Faculty of Electrical Engineers
- June 1995 Received degree of “Diplomingenieur” in Communications Engineering from the Technical University of Vienna (with honors)
- April 1996 Entered doctoral program at the Institute for Microelectronics, TU Vienna
- October 1997 - December 1997 Visiting Researcher at Device Development Center of Hitachi, Tokyo, Japan
- Since February 1998 Assistant Professor at the Institute for Microelectronics, TU Vienna