Modeling of Semiconductor Devices

Tibor Grasser

360.033

SS 2010

Ghostwritten by:
Karl Rupp

Based on input from:
Andreas Gehring
Wolfgang Gös
Tibor Grasser
Philipp Hehenberger
Franz Schanovsky
Paul-Jürgen Wagner

Institute for Microelectronics

TU Vienna

Contents

1 Semiconductor Equations **1**

1.1 The (Electrostatic) Field Side of the Problem . . . 1
1.2 Continuity Equations . . . 2
1.3 Charge Transport – The Drift-Diffusion-Model . . . 3
1.4 Summary – The Basic Semiconductor Equations . . . 4
1.5 Outlook – Lattice Heat-Flow . . . 5

2 Carrier Transport **7**

2.1 The Distribution Function . . . 7
2.2 The Equilibrium Case . . . 10
2.3 Boundary Conditions . . . 13
2.3.1 Quasi-Fermi Levels . . . 13
2.3.2 Contact Potentials . . . 17
2.3.3 The Built-in Potential . . . 20
2.4 Ohmic Contacts . . . 21

3 Basics of Numerical Analysis **25**

3.1 Introduction to Finite Differences . . . 25
3.2 Numerical Solution of Differential Equations . . . 28
3.3 The Second Order Derivative . . . 30
3.4 Accuracy of Different Discretization Schemes . . . 32
3.5 Solution of Linear Systems of Equations . . . 34
3.6 Solution of Nonlinear Systems of Equations . . . 35

4 Two-Dimensional Simulation and Grids **41**

4.1 Two-Dimensional Laplace Equation . . . 41
4.2 Box Integration Method . . . 44
4.2.1 Example: The Poisson Equation . . . 45
4.2.2 Example: Extraction of Capacitances . . . 46

5 Tessellation of Unstructured Meshes **51**
5.1 Voronoi Tessellation . . . 52
5.2 Triangular Delaunay Meshes . . . 52
5.3 Skeleton Mesh . . . 54
5.4 Auxiliary Functions . . . 57
5.5 Skeleton Mesh – Boundaries . . . 59
5.6 Skeleton Mesh – Interfaces . . . 62
5.7 Mesh Refinement . . . 64

6 Transport Phenomena and their Numerical Analysis **69**
6.1 Discretization in the Time Domain . . . 70
6.2 Stability of Discretization Schemes . . . 73
6.3 Diffusive Problems . . . 75
6.4 Convective Problems . . . 80
6.5 Diffusive and Convective Problems . . . 83
6.5.1 Scharfetter-Gummel Discretization . . . 87

7 Parameter Modeling **91**
7.1 Carrier Mobilities . . . 91
7.1.1 Lattice Scattering . . . 92
7.1.2 Ionized Impurity Scattering . . . 93
7.1.3 Surface/Interface Scattering . . . 93
7.1.4 Carrier Heating . . . 95
7.2 Carrier Generation and Recombination . . . 96
7.2.1 Phonon Assisted Recombination and Generation . . . 97
7.2.2 Photon Transition . . . 100
7.2.3 Auger Generation-Recombination . . . 101
7.2.4 Impact Ionization . . . 102

8 Devices in Detail **105**
8.1 Analytical Diode Model . . . 105
8.2 Small Signal Analysis of a Diode . . . 110
8.3 Analytical Bipolar Junction Transistor Model . . . 112
8.4 The Metal-Oxide-Semiconductor Capacitor . . . 116
8.5 The Metal-Oxide-Semiconductor Field-Effect-Transistor (MOSFET) . . . 122
8.6 CMOS Design Issues . . . 125

A Partial Differential Equations **131**

A.1 Boundary and Initial Conditions 131
A.2 Classification 133

B Vector Analysis and Its Implementation in SGFramework **135**
B.1 Divergence 135
B.2 Curl (Rotation) 137
B.3 Gradient 138
B.4 Nabla 139
B.5 Manipulating Expressions 140
B.6 Identities 141
B.7 Integral Theorems of Stokes and Gauss 141

C Basics of Electromagnetism – Maxwell's Equations **143**
C.1 Interface Conditions 144
C.2 Continuity Equation 145

D Vocabulary **147**

Bibliography **150**

Index **150**

Preface[1]

Attempts to model semiconductor devices are nearly as old as the devices themselves. Any good model should be based on a **sound**[2] physical understanding of the particular device and the use of **dumb**[3] empirical fits should be avoided. The theoretical basis we require in this lecture has been developed in the various lectures dealing with the fundamentals of semiconductor physics. The response of carriers to external stimulus and internal forces and fields essentially **determines**[4] the device behavior and can be expressed by a set of differential equations. Various degrees of **sophistication**[5] exist but we will limit ourselves to the most **rudimentary**[6] model, the drift-diffusion model.

Using the drift-diffusion model, the behavior of most (conventional) semiconductor devices can be understood and modeled. Unfortunately, no closed form analytic solutions of the drift-diffusion model exists and one has to **resort**[7] to alternative techniques. Roughly speaking, modeling attempts fall into the two categories compact modeling and numerical modeling. By considering the **peculiarities**[8] of each device **subjected**[9] to particular bias conditions, the art of compact modeling tries to simplify the underlying equations in such a way that an approximate analytic expression is obtained. Conventionally, a number of assumptions enters the derivation and sometimes these assumptions are difficult to justify. Also the analytical solutions are often used without detailed knowledge of these simplifying assumptions, which can dramatically limit the validity of compact models. Thus, it is often of fundamental importance to solve the semiconductor device equations with as few approximations as possible using numerical techniques. Unfortunately, a number of difficulties have to be **surmounted**[10] in order to achieve this goal. As such, numerical device modeling is a highly interdisciplinary subject, requiring a good knowledge of semiconductor device physics, electrical engineering, numerical mathematics, and computer science. In this lecture we try to provide you with the necessary basics, the **big picture**[11], so to speak. It has to be understood, however, that it is impossible to cover the rich spectrum of knowledge available in all these fields and the interested student is **encouraged**[12] to consult appropriate scientific literature. Access to numerous scientific journals is provided for free to our students by the Technology Library (`www.ub.tuwien.ac.at/eng/index.html`). A good starting point would be search engines such as Scopus (`www.scopus.com`) and IEEE Explore (`ieeexplore.ieee.org`).

Although a good theoretical knowledge is mandatory, the real understanding of a topic is best gained in a hands-on approach. As such, a strong focus is put on the practical (home-work) part of this lecture where (simple) realistic device structures have to be numerically modeled.

---

[1] **preface** [ˈprefɪs]**, NOT** [ˈprɪːfeɪs]: Vorwort [2] **sound** [saʊnd]: auch: vernünftig, sinnvoll [3] **dumb** [dʌm]: einfältig, primitiv [4] **to determine** [dɪˈtɛː.mɪn]**, NOT** [determaɪn]: bestimmen, festlegen [5] **sophistication** [səˌfɪs.tɪˈkeɪ.ʃᵊn]: Raffinesse [6] **rudimentary** [ruːdɪˈmen.tᵊr.i]: rudimentär, elementar [7] **to resort to sth.** [rɪˈzɔːrt]: auf etw. zurückgreifen [8] **peculiarity** [pɪˌkjuː.liˈer.ə.t̬i]: Eigenheit, Ausprägung [9] **to be subjected to sth.** [sʌb.dʒekt]: etwas ausgesetzt werden [10] **to surmount** [səˈmaʊnt]: bewältigen, überwinden [11] **big picture** [bɪg pɪk.tʃɚ]: Das große Ganze, ein erster Einstieg [12] **to encourage** [ɪnˈkʌr.ɪdʒ]: animieren, ermuntern

In order to avoid the typical **pitfalls**[1] associated with C++ compilers, operating systems, file formats, etc., we provide you with a web interface to the simulation engine SGFRAMEWORK (`meb.iue.tuwien.ac.at`).

Being the dominant language in technical sciences, these lecture notes are written in English, to allow you to develop the necessary vocabulary required in this field. We have taken the liberty to remind you of the **pronunciation**[2] and translation of frequently used words and terms using footnotes. In particular, we have tried to highlight words which are often subjected to creative pronunciation solutions by non-native speakers, such as **determine**[3]. Pronunciation notes are taken from the Cambridge online dictionaries (`dictionary.cambridge.org`) and follow the conventions of American English.

These lecture notes are based on my slides compiled for this lecture and additional input contributed by various people. I am particularly **indebted**[4] to Karl Rupp for **painstakingly**[5] collecting, unifying, extending, and in very large parts completely rewriting the existing material to form this hopefully consistent first version of our lecture notes.

Tibor Grasser
Wien, March 2009

[1] **pitfall** [pɪt.faːl]: Fallgrube, Fallstrick, Fehler [2] **pronunciation** [prəˌnʌntsiˈeɪʃᵊn]: Aussprache [3] **to determine** [dɪˈtɛː.mɪn], **NOT** [determaɪn]: bestimmen, festlegen [4] **indebted** [ɪnˈdet.ɪd]: verpflichted, verschuldet [5] **painstakingly** [peɪnzˌteɪ.kiŋ.li]: sorgfältig [Man beachte den Wortstamm *pain* ;-), Anm. K.R.]

Symbols

Mathematical Symbols

Ω	bounded domain
$\partial\Omega$	boundary of a domain Ω
$\mathbb{R}$	real numbers
$\boldsymbol{A} = \left(a_{ij}\right)_{i,j=1}^{n}$	$n \times n$ matrix
$\boldsymbol{b} = (b_i)_{i=1}^{n}, \boldsymbol{x} = (x_i)_{i=1}^{n}$	vectors with n entries
$\frac{\mathrm{d}}{\mathrm{d}x}, \frac{\mathrm{d}\cdot}{\mathrm{d}t}, \dots$	derivative with respect to x, t, ...
$\frac{\mathrm{d}^2}{\mathrm{d}x^2}, \frac{\mathrm{d}^2\cdot}{\mathrm{d}x^2}, \dots$	second order derivative with respect to x, t, ...
$\frac{\partial}{\partial x}, \frac{\partial\cdot}{\partial t}, \dots$	partial derivative with respect to x, t, ...
$\frac{\partial^2}{\partial x^2}, \frac{\partial^2\cdot}{\partial t^2}, \dots$	second order partial derivative with respect to x, t, ...
$\exp(x), \mathrm{e}^x$	exponential function
∇	Nabla operator
$\nabla\cdot$	divergence
$\nabla\times$	rotation
$\Delta = \nabla^2$	Laplace operator

Physical Quantities and Constants

$\boldsymbol{E}$	electric field
$\boldsymbol{D}$	electric flux density
$\boldsymbol{B}$	magnetic field
$\boldsymbol{H}$	magnetizing field
ψ	(electrostatic) potential
$\hat{\varepsilon}$ or ε	permittivity (first case: a tensor)
ρ	charge density
n	electron concentration
p	hole concentration
$N_\mathrm{A}, N_\mathrm{D}$	acceptor and donor concentration
n_i^2	intrinsic carrier concentration
q	elementary charge ($1.602176487(40) \times 10^{-19}$ C)
$\boldsymbol{J}_n, \boldsymbol{J}_p$	electron/hole current density
R	recombination rate

μ_n, μ_p	electron/hole mobility
$\boldsymbol{v}_n$, $\boldsymbol{v}_p$	electron/hole velocity
σ_n, σ_p	electron/hole conductivity
D_n, D_p	electron/hole diffusion coefficient
$\mathrm{k_B}$	Boltzmann constant ($1.3806503 \times 10^{-23}\,\mathrm{m^2\,kg\,s^{-2}\,K^{-1}}$)
T_L	lattice temperature
V_T	thermal voltage
E_g	band gap energy
E_F	Fermi energy
V_T	thermal voltage

Chapter-specific Symbols and Quantities

Chapter 1

κ	thermal conductivity
ϱ	mass density
c	specific heat
H	heat source term

Chapter 2

$\boldsymbol{q}$	momentum
$\boldsymbol{F}$	force field
$\boldsymbol{R}$	random force field (scattering)
$\boldsymbol{u}$	(particle) velocity
$f(\boldsymbol{p}, \boldsymbol{r}, t)$	(carrier) distribution function
$Q(f)$	scattering operator
$S(\cdot,\cdot)$	scattering rate
$\mathcal{E}_\mathrm{pot}$	potential energy
$\mathcal{E}_\mathrm{kin}$	kinetic energy
E_tot	total energy
E_c	conduction band edge energy
$E_\mathrm{c,0}$	conduction band edge energy without external bias
m^*	effective mass
v_th	thermal velocity
N_c, N_v	effective densities of states (aka. band weights) of conduction and valence band
$E_{\mathrm{F}n}$, $E_{\mathrm{F}p}$	quasi Fermi levels for electrons and holes
$\mathcal{E}_\nu$, $\nu \in \{\mathrm{c}, \mathrm{v}, \mathrm{i}\}$	Energy of conduction band, valence band and intrinsic energy level.
V_{12}	contact potential between material 1 and material 2
$\mathrm{q}\Phi$	work function
E_vac	vacuum energy
V_D	diffusion voltage (aka. built-in potential)

Chapter 3

h_i, Δx_i	grid size, distance between adjacent grid points
x_i	i-th grid point (1 dimensional grid)
u_i	value of a quantity u evaluated at the i-th grid point x_i
$\boldsymbol{J_F}(x)$	Jacobian matrix of a vector-valued function $\boldsymbol{F}$ evaluated at x

Chapter 4

$\mathcal{V}$	a small box
$\partial\mathcal{V}$	surface of a box $\mathcal{V}$

Chapter 5

Ω_i	i-th box of a tessellation of Ω
V_i	volume of the box Ω_i
$\bigcup$	union
$\mathcal{N}_i$	set of all nodes that are neighbors of the i-th node
Δx, Δy	grid size in x and y direction, distance between adjacent grid points
$x_{i,j}$	grid point
$u_{i,j}$	value of a quantity u evaluated at $x_{i,j}$
$d_{i,j}$	length of the edge that connects i and its neighbor-point j
$A_{i,j}$	area of the surface element of Ω_i that interfaces the box Ω_j
$D_{i,j}$	electric flux density in the outward direction at $A_{i,j}$
$M_{i,j}$	refinement measure

Chapter 6

$\frac{\partial}{\partial \boldsymbol{n}}$	normal derivative
$\mathscr{F}$	Fourier transformation
$\eta = \frac{\Delta t}{(\Delta x)^2}$	discretization parameter
$\lambda = \frac{\Delta t}{\Delta x}$	discretization parameter
$\mathcal{B}(x) = x/(\mathrm{e}^x - 1)$	Bernoulli function

Chapter 7

μ^{L}	mobility considering scattering at lattice atoms or defects
μ^{I}	mobility considering scattering at charges or neutral impurities
μ^{S}	mobility considering surface roughness scattering
μ^{F}	mobility considering increased scattering due to heating
μ^{LISF}	effective mobility
τ_{n}, τ_{p}	carrier lifetimes for electrons and holes respectively
$f_{\mathrm{t},0}$	Fermi-Dirac distribution function
e^-	electron
h^+	hole

R^{RSH}	Read-Shockley-Hall recombination rate
R^{opt}	recombination due to optical generation and radiative recombination
R^{AU}	Auger recombination rate
G^{II}	impact ionization generation rate

Chapter 8

l_{p}	width of the space-charge region in p-Si
l_{n}	width of the space-charge region in n-Si
V_{c}	contact voltage
n_{n0}, p_{p0}	equilibrium majority carrier concentration in n– and p–region respectively
p_{n0}, n_{p0}	equilibrium minority carrier concentration in n– and p–region respectively
C'_{J}	junction capacitance (or depletion capacitance) per unit area
C'_{D}	diffusion capacitance (or storage capacitance) per unit area
$\Delta n := n - n_{\mathrm{p0}}$	deviation from equilibrium minority carrier concentration in the npn-base
$L_{\mathrm{B}} = \sqrt{D_{\mathrm{n}}\tau_{\mathrm{n}}}$	diffusion length
W	base width
V_{FB}	flat band voltage
χ_{S}	electron affinity
ϕ_{S}	surface potential
V_{t}	threshold voltage
I_{D}	drain current
γ	body factor
λ	channel length parameter
S	subthreshold slope

Pronunciation

Since the **pronunciation**[1] of English words if often a source of confusion among non-native speakers, a phonetic transcription of critial words is given. This phonetic transcription uses the standard International Phonetic Alphabet (IPA) and is based on the Cambridge Dictionary of American English.

Symbol	Description	Example
ɑː		calm [kɑːm], heart [hɑːrt], far [fɑːr]
æ		act [ækt]
aɪ		dive [daɪv], cry [kraɪ]
e		met [met], lend [lend], pen [pen]
eɪ		say [seɪ], weight [weɪt]
ɚ	r-colored schwa	mother [mʌðɚ]
ɪ		fit [fɪt], win [wɪn]
iː		feed [fiːd], me [miː]
oʊ		note [noʊt], coat [koʊt]
ɔː		more [mɔːr], cord [kɔːrd]
ɔɪ		boy [bɔɪ], joint [dʒɔɪnt]
ʊ		could [kʊd], stood [stʊd]
uː		you [juː], use [juːz]
ɛː		turn [tɛːn], third [θɛːrd]
ʌ		fund [fʌnd], must [mʌst]
ə	schwa	about [əˈbaʊt]
ᵊ	optional schwa	label [ˈleɪbəl̩] or [ˈleɪbl̩]
i		very [ˈveri]
l̩		handle [ˈhændl̩]
ŋ	velar nasal n (no g!)	bring [brɪŋ]
ʃ		ship [ʃɪp]
ʒ		measure [meʒɚ]
t̬	american flapped t	butter [ˈbʌt̬ɚ]
θ	voiceless dental fricative	thing [θɪŋ]
ð	voiced dental fricative	then [ðen]
dʒ		joy [dʒɔɪ]

[1] **pronunciation** [prəˌnʌntsiˈeɪʃon]: Aussprache

Chapter 1

Semiconductor Equations

In this section we will derive the set of equations needed to describe microelectronic devices. We reduce Maxwell's equations to the absolute minimum necessary and add equations that describe the behavior of the semiconductor material. Later in the lecture, this equation system will be solved numerically and approximated analytically.

1.1 The (Electrostatic) Field Side of the Problem

As a first step we show that the use of quasi-electrostatic approximations is **justified**[1]. Our condition for that is that the characteristic length of our system is considerably (by a factor of, say 10) smaller than the shortest electromagnetic wavelength present in the device. Given an upper limit of 100 GHz for the frequency of our electromagnetic field results in a wavelength of $\lambda = c/f = 877\mu$m; the characteristic device dimension in microelectronics is about 1 µm, confirming our **assumption**[2] of a quasi-static situation.

But what exactly does 'quasi-electrostatic approximation' mean? First of all, both the displacement current $\partial \boldsymbol{D}/\partial t$ and the induction $\partial \boldsymbol{B}/\partial t$ are **neglected**[3]. This simplifies Maxwell's equations to a much higher extent than one may assume at a first sight: Not only do two terms **vanish**[4], but the former coupled system of differential equations **decouples**[5], i.e. the direct coupling between the electric and the magnetic part of the field and their corresponding equations disappears. The only remaining interaction between the two field components is through the relation between the electric field $\boldsymbol{E}$ and the electric current density $\boldsymbol{J}$, which causes a magnetic field $\boldsymbol{H}$. To simplify things further, even this magnetic field is usually neglected, making the two Maxwell equations for the magnetic field completely **obsolete**[6] in our applications. The fact that the right hand side of $\nabla \times \boldsymbol{E} = -\partial \boldsymbol{B}/\partial t$ is zero permits the introduction of a *scalar potential* ψ, with the electric field being $\boldsymbol{E} = -\nabla\psi$ (the minus sign is here for historical reasons; it is in no way physically justified, nor mandatory). Assuming linear, but possibly anisotropic and inhomogeneous material, the electric elasticity equation $\boldsymbol{D} = \hat{\varepsilon} \cdot \boldsymbol{E}$ holds, where the permittivity tensor $\hat{\varepsilon}$ in general depends on the spatial coordinates. Together with Gauss' law we get

$$\nabla \cdot (\hat{\varepsilon} \cdot \nabla\psi) = -\rho \quad . \tag{1.1}$$

[1] **to justify** [dʒʌs.tɪ.faɪ]: rechtfertigen [2] **assumption** [əˈsʌmp.ʃən]: Annahme [3] **to neglect** [nɪˈglekt]: vernachlässigen [4] **to vanish** [væn.ɪʃ]: verschwinden [5] **to decouple** [dɪkʌp.l]: entkoppeln [6] **obsolete** [ɔb.səlˈiːt]: hinfällig

The charge density ρ is composed of three components. Electrons possessing enough thermal energy to detach from the dopants and therefore obtain the ability to move around constitute the electron density n. Clearly, they leave behind a positively charged atom unable to move. But the electron's place can be taken by another electron, which itself leaves an empty electron position somewhere else. As more electrons move 'downstream' to fill the electron **vacancies**[1], the vacancy itself moves 'upstream'. This constitutes a **fictitious**[2], positively charged carrier type called a *hole*. The hole density, which is actually the density of 'missing' electrons, is denoted by p. Electrons and holes are present even in a perfectly pure semiconductor. In real semiconductors, impurities are always present. Moreover, they are often introduced **deliberately**[3] to control the electric conductivity of the material, in which case they are called *dopants*. The concentration of ionized **impurities**[4] and dopants is summed up in the concentration C. Putting all together **yields**[5] the space charge density

$$\rho = \mathrm{q}(p - n + C) \ , \tag{1.2}$$

where q is the elementary charge. Finally, assuming the permittivity to be scalar and spatially independent (so it can be moved out of the nabla operator), we arrive at the equation

$$\nabla^2 \psi = \mathrm{q}(n - p - C)/\varepsilon \ , \tag{1.3}$$

which is known as *Poisson's equation*. In case $\rho \equiv 0$, one obtains *Laplace's equation*.

1.2 Continuity Equations

In pretty much the same way as the charge density consists of various contributions, the current density is **decomposed**[6] into an electron current density $\boldsymbol{J}_n$ and a hole current density $\boldsymbol{J}_p$—the impurities and dopants are fixed in the crystal and therefore do not contribute to the current density. Hence, the charge continuity equation reads

$$\nabla \cdot (\boldsymbol{J}_n + \boldsymbol{J}_p) + \mathrm{q}\frac{\partial}{\partial t}(p - n) = 0 \ . \tag{1.4}$$

(Note that since $\partial C/\partial t \equiv 0$, the respective term in ρ was dropped in the equation above.) This equation can formally be split into two equations; at the right hand side, a new term R is introduced:

$$\nabla \cdot \boldsymbol{J}_n - \mathrm{q}\frac{\partial n}{\partial t} = \mathrm{q}R \ , \tag{1.5}$$

$$\nabla \cdot \boldsymbol{J}_p + \mathrm{q}\frac{\partial p}{\partial t} = -\mathrm{q}R \ . \tag{1.6}$$

The interpretation of these two equations is as follows: Since charge particles actually can not be 'generated' or 'destroyed' (the right hand side of the continuity equation is zero), every additional electron that shows up in a left-alone semiconductor leaves an additional hole. Since these two have opposite charges, they appear in their respective continuity equation with opposite signs. The quantity R is the rate at which electron-hole-pairs are generated minus the rate at which they *recombinate*. In equilibrium, the *net* recombination rate is zero (detailed balance), thus $R \equiv 0$; but also out of equilibrium R is often neglected, because it considerably simplifies the problem. Depending on the type of the device, the inclusion of carrier generation and recombination models is mandatory for a realistic description.

[1] **vacancy** [veɪ.kənt.si]: freie Stelle, insbes. Gitterfreistelle [2] **fictitious** [fɪk'tɪʃ.əs]: fiktiv [3] **deliberately** [dɪˈlɪb.ər.ət.li]: absichtlich [4] **impurity** [ɪmˈpjʊəriti]: Störstelle, Störatom [5] **to yield sth.** [jiːld]: etwas ergeben, etwas hervorbringen [6] **to decompose** [diː.kəmˈpəʊz]: aufteilen, spalten

1.3 Charge Transport – The Drift-Diffusion-Model

The *structure* of our field problem is described by Poisson's equation and the two continuity equations. Neglecting R, five quantities are involved (ψ, n, p, $\boldsymbol{J}_n$, and $\boldsymbol{J}_p$), so we are short of two equations. They are provided by the microscopic model of the material considered, which describes how the field (mainly governed by Poisson's equation) acts on the charge particles (mainly governed by the continuity equations). The simplest model available is the so-called *drift-diffusion model*, which considers two distinct charge carrier transport mechanisms: Charge carrier *drift* due to the presence of an electric field, which is the customary transport mechanism in ordinary conductors, e.g. metals; and charge carrier *diffusion*. Diffusion is a fundamental process, which tries to establish a thermodynamic equilibrium in an initially imbalanced physical system. In our case, the physical system is the spatially non-constant **distribution**[1] of charge particles (n and p, where $\nabla n \neq 0$ and $\nabla p \neq 0$ in general). The thermodynamic equilibrium would be a situation in which $\nabla n = \nabla p = 0$, i.e. the carriers are **evenly**[2] distributed in the crystal. This situation is established through carrier migration from areas with high concentration to areas where the concentration of particles is lower. Mathematically speaking, the particles move in the opposite direction of the concentration gradient.

The drift component is expressed using the concept of *carrier mobility*, which is the proportionality factor between field strength and (average) carrier velocity. Denoting the individual mobilities for electrons and holes by μ_n and μ_p, we have

$$\boldsymbol{v}_n = -\mu_n \boldsymbol{E} \quad \text{and} \quad \boldsymbol{v}_p = \mu_p \boldsymbol{E} \ . \tag{1.7}$$

(Note the minus sign, because electrons with their negative charge travel – for historical reasons – *against* the field direction!). Charge carriers moving with some average velocity $\boldsymbol{v}$ constitute an electric current density $\boldsymbol{J}$, whose magnitude is proportional not only to the velocity itself, but also to the *absolute number* of charge carriers per unit area that are on their way. But this number of carriers per unit area is directly related to the carrier density, which is n for electrons, each carrying a charge $-\mathrm{q}$ (mind the sign!) and p for holes (with charge q each). All in all we have

$$\boldsymbol{J}_n^{\text{Drift}} = -\mathrm{q}n\boldsymbol{v}_n = \mathrm{q}n\mu_n\boldsymbol{E} \quad \text{and} \quad \boldsymbol{J}_p^{\text{Drift}} = \mathrm{q}p\boldsymbol{v}_p = \mathrm{q}p\mu_p\boldsymbol{E} \ ; \tag{1.8}$$

again note the signs: Electrons move against the field direction, because of their negative charge. But for the same reason, the current they constitute points in the opposite direction of their velocity vector. In the end, both minus signs cancel. By introducing the conductivities $\sigma_n = \mathrm{q}n\mu_n$ and $\sigma_p = \mathrm{q}p\mu_p$ the relations above take the form of Ohm's law,

$$\boldsymbol{J}_n^{\text{Drift}} = \sigma_n\boldsymbol{E} \quad \text{and} \quad \boldsymbol{J}_p^{\text{Drift}} = \sigma_p\boldsymbol{E} \ . \tag{1.9}$$

The diffusion component is described using the notion of a *particle flux density* $\boldsymbol{F}$, which is proportional to the negative gradient of the particle density. The proportionality factor is called the *diffusion coefficient*—since electrons and holes diffuse separately, two distinct diffusion coefficients D_n and D_p are involved:

$$\boldsymbol{F}_n = -D_n\nabla n \ , \qquad \boldsymbol{F}_p = -D_p\nabla p \ . \tag{1.10}$$

The current densities are simply the flux densities multiplied with the individual charge of the carriers,

$$\boldsymbol{J}_n^{\text{Diffusion}} = -\mathrm{q}\boldsymbol{F}_n = \mathrm{q}D_n\nabla n \ , \qquad \boldsymbol{J}_p^{\text{Diffusion}} = \mathrm{q}\boldsymbol{F}_p = -\mathrm{q}D_p\nabla p \ . \tag{1.11}$$

[1] **distribution** [dɪˈstrɪb.juː.ʃᵊn]: Verteilung [2] **evenly** [iː.vᵊn.li]: gleichmäßig

Close to equilibrium the carrier mobility and the diffusion coefficient are linked by the *Einstein relation*:

$$D_{n,p} = \frac{k_B T}{q} \mu_{n,p} = V_T \mu_{n,p} \; ; \tag{1.12}$$

the quantity V_T is referred to as the *thermal voltage* which is around 26mV at room temperature. The Einstein relation is only approximately valid for the non-equilibrium case we are interested in. The temperature T in (1.12) is the temperature of the electrons[1]. Although in many situations the **lattice**[2] temperature matches the electron temperature, there are cases (like in a MOSFET-channel) where electrons have a much higher "temperature" (i.e.[3] energy) than the lattice (*hot electrons*).

Note that carrier diffusion is not part of classical electrodynamics, where $\boldsymbol{J}$ and $\boldsymbol{E}$ are linked by Ohm's law. The reason for this is simple: In classical electrodynamics, matter is either a conductor or an insulator. In the latter case, there's no carrier transport at all, while in the former case the number of free charge carriers is assumed to be very high. This is the case for metals, where each atom at least contributes one electron to the electron gas, and for **ionic conductors**[4], where each atom is a free charge carrier itself. In a system with that many free charged particles, a local disturbance in the carrier density is equilibrated almost instantaneously by the surrounding charges, causing the carrier densities to be virtually constant **throughout**[5] the whole body. The fact that in a semiconductor there is only a single electron for, say, every 100.000th atom makes the diffusion of carriers visible to a macroscopic observer.

1.4 Summary – The Basic Semiconductor Equations

Finally, we are able to put together a set of equations of first and second order that describes the basic behavior of a semiconductor:

$$\nabla^2 \psi = q(n - p - C)/\varepsilon \; , \tag{1.13}$$

$$\nabla \cdot \boldsymbol{J}_n - q\frac{\partial n}{\partial t} = qR \; , \tag{1.14}$$

$$\nabla \cdot \boldsymbol{J}_p + q\frac{\partial p}{\partial t} = -qR \; , \tag{1.15}$$

$$\boldsymbol{J}_n = qn\mu_n \boldsymbol{E} + qD_n \nabla n \; , \tag{1.16}$$

$$\boldsymbol{J}_p = qp\mu_p \boldsymbol{E} - qD_p \nabla p \; . \tag{1.17}$$

Substituting for the electric field $\boldsymbol{E} = -\nabla\psi$ and using the Einstein relation,

$$\nabla^2 \psi = q(n - p - C)/\varepsilon \; , \tag{1.18}$$

$$\nabla \cdot \boldsymbol{J}_n - q\frac{\partial n}{\partial t} = qR \; , \tag{1.19}$$

$$\nabla \cdot \boldsymbol{J}_p + q\frac{\partial p}{\partial t} = -qR \; , \tag{1.20}$$

$$\boldsymbol{J}_n = -q\mu_n (n\nabla\psi - V_T \nabla n) \; , \tag{1.21}$$

$$\boldsymbol{J}_p = -q\mu_p (p\nabla\psi + V_T \nabla p) \; . \tag{1.22}$$

[1] The temperature of electrons will be discussed in Chapter 2 [2] **lattice** [læt.ɪs]: Kristallgitter [3] i.e. (lat. id est): "that is" [4] **ionic conductor** [aɪˈɔn.ɪk kənˈdʌk.tɚ]: Ionenleiter [5] **throughout** [θruːˈaʊt]: durchweg, hindurch

The current relations can be inserted into the continuity equations, yielding the system of second order partial differential equations

$$\nabla^2\psi = \mathrm{q}(n-p-C)/\varepsilon \ , \tag{1.23}$$

$$\nabla\cdot(\mu_n n\nabla\psi - \mu_n V_\mathrm{T}\nabla n) + \frac{\partial n}{\partial t} = -R \ , \tag{1.24}$$

$$\nabla\cdot(\mu_p p\nabla\psi + \mu_p V_\mathrm{T}\nabla p) - \frac{\partial p}{\partial t} = R \ . \tag{1.25}$$

These equations constitute the *drift-diffusion-model (DD-model)*, which, despite its limitations, is still the most widely used semiconductor model today. It was first derived by Van Roosbroeck back in 1950. [**?**]

1.5 Outlook – Lattice Heat-Flow

Basically all microphysic phenomena in solids are temperature dependent. Temperature enters the semiconductor equations directly via the thermal voltage V_T and indirectly via the recombination rate R and the temperature dependence of the mobilities μ_n and μ_p. Since the local temperature is an additional quantity in our semiconductor model, an additional equation must be provided. Heat enters the balance equations in the same way as charge does; the equation for the temperature distribution therefore pretty much looks like the charge continuity equation:

$$\nabla\cdot(\kappa\nabla T_\mathrm{L}) - \varrho c\frac{\partial T_\mathrm{L}}{\partial t} = -H \tag{1.26}$$

and is of a structure similar as (1.25). Heat is redistributed via phonons, but there is no drift component since atoms are fixed within the lattice. T_L is the temperature, more precisely, the temperature of the lattice—in some cases, e.g. in the channels of field effect transistors where the electrons reach saturation velocity, they are attributed a higher temperature than that of the lattice *(hot electrons)*, requiring a **distinction**[1] between lattice temperature and carrier temperature. The stationary temperature distribution is governed by the thermal conductivity κ, while the initial transient response to a change in the heat sources H is **determined**[2] by the mass density ϱ ($2328\,\mathrm{VAs^3m^{-5}}$ in silicon) and the specific heat c ($703\,\mathrm{m^2s^{-2}K^{-1}}$ in silicon). Lastly, the heat generation term H provides the 'back end' of the coupling between the heat-flow and the drift-diffusion equations, since heat in the semiconductor is either generated by a first order Joule-term $\boldsymbol{E}\cdot\boldsymbol{J}$ or by carrier recombination: Every recombination process sets free an energy amount at least equal to the semiconductor's band gap energy E_g; every generation of an electron-hole-pair withdraws E_g from the crystal. Therefore, we have

$$H \approx \boldsymbol{E}\cdot\boldsymbol{J} + RE_\mathrm{g} \ . \tag{1.27}$$

Interestingly, a temperature gradient inside the crystal also **provokes**[3] a carrier diffusion process, yielding additional terms in the drift-diffusion current relations:

$$\boldsymbol{J}_{n,\mathrm{th}} = \mathrm{q}D_{n,\mathrm{th}}\nabla T_\mathrm{L} \ , \qquad \boldsymbol{J}_{p,\mathrm{th}} = -\mathrm{q}D_{p,\mathrm{th}}\nabla T_\mathrm{L} \ ; \qquad D_{n,p,\mathrm{th}} = D_{n,p}/(2T_\mathrm{L}) \ . \tag{1.28}$$

[1] **distinction** [dɪˈstɪŋk.ʃᵊn]: Unterscheidung [2] **to determine** [dɪˈtɛː.mɪn]**, NOT** [determaɪn]: bestimmen, festlegen
[3] **to provoke** [prəˈvoʊk]: auslösen, bewirken

These current contributions are required for the modeling of the *thermoelectric effect*, **aka**[1] *Seebeck effect*. Also the reverse process can be observed, the transfer of heat by an electric current is known under the name *Peltier effect*.

[1] **aka, also known as** [ɔːlsou noʊŋ æz]: auch bekannt unter, so genannt

Chapter 2

Carrier Transport

In this chapter we will look at carrier transport in greater detail, that is, how electrons and holes in a semiconductor respond to local and external forces. In the first chapter we have briefly introduced the drift-diffusion model, where we solve for the electron and hole concentrations. This means that electrons and holes in this model are **solely**[1] characterized by their locations in space, just as impurities have fixed positions in the lattice. As we know from semiconductor physics, an energy (or momentum) can be assigned to each electron, and the allowed energy levels are obtained from the band structure of a material. Instead of looking at the spatial distribution of electrons only, we will also briefly discuss electron energies, i.e. the population density of the energy bands. As we will see, the additional effort of considering the electron momentum pays off in the form of increased insight in the semiconductor device internals.

2.1 The Distribution Function

When no external electric field is applied and all transient processes have relaxed, we say that the semiconductor is in *thermal equilibrium*. In thermal equilibrium the electron gas is in equilibrium with the lattice. This does not mean, however, that the electrons rest in their equilibrium position: They rather fly around in a random fashion with their average energy equal to the thermal energy. Their average velocity, however, is zero because there is no preferred direction of movement and thus on average no electrical current flows. When an electric field is applied, the electrons are accelerated in **accordance**[2] with Newton's law of motion

$$\frac{\partial \boldsymbol{p}}{\partial t} = -\mathrm{q}\boldsymbol{E} \, . \tag{2.1}$$

Assuming that the electric field is kept constant and never turned off, it follows from (2.1) that the electrons reach an infinite velocity, or, taking relativistic effects into account, the speed of light. In a real semiconductor, however, an important process sets a limit to the maximum velocity: **scattering**[3]. In the models we will consider here, scattering processes are traced back to changes of the local potential seen by electrons. These changes can be caused (besides many other effects) by lattice vibrations which change the local band edge energy or by the electric field emerging from ionized impurities.

At any temperature above zero Kelvin, lattice atoms vibrate, **disturbing**[4] the perfect periodicity of the otherwise stationary lattice. This breaking of the periodicity leads to an energy exchange

[1] **solely** [soʊl.li]: lediglich [2] **accordance** [əˈkɔːr.dᵊnts]: Übereinstimmung [3] **to scatter** [skæt̬.ɚ]: streuen, zerstreuen [4] **to disturb** [dɪˈstɜːb]: stören, durcheinanderbringen

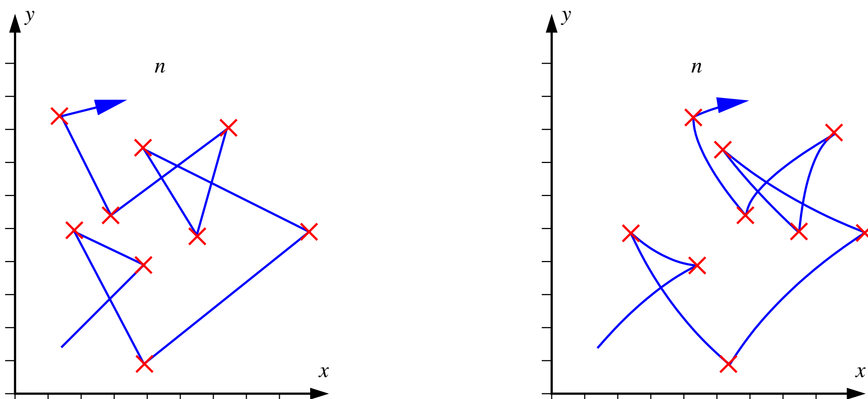


Figure 2.1: Thermal motion of a carrier without an externally applied field (left) and under the influence of a very strong electric field (right).

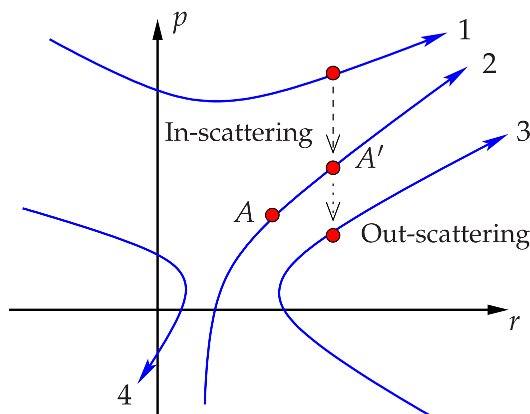


Figure 2.2: Illustration of trajectories in position-momentum space. Carriers move along a trajectory according to Newton's Laws. Occasionally they scatter to another trajectory. Scatting instantly changes the carrier's momentum, but does not affect its position.

of the electrons with the lattice. Lattice vibrations are characterized by *phonons*, that is why the resulting scattering process is termed *phonon scattering*. Where the electron movement is impeded by phonon scattering, the mobility decreases with increasing temperature. Consequently, a reduction of phonon scattering can be achieved by lowering the temperature, but since the ambient temperature is around 300 Kelvin in almost all cases, we have to live with this scattering mechanism.

The second possibility for a **deviation**[1] from perfect periodicity of the lattice potential is due to impurities: When there is a significant number of impurity atoms, the local lattice potential is distorted and scatters passing electrons. This scattering mechanism is called *impurity scattering* and can be controlled — unlike phonon scattering — rather easily: It can be eliminated by removing impurities from the material. However, a removal of impurities has other side-effects like increased resistance, which is often not desired. Since impurities are the essential building blocks of semiconductor devices and are used to form *pn* junctions, the scattering associated with impurities cannot be completely avoided. Nonetheless, under certain circumstances it is

[1] **deviation** [diː.viˈeɪ.ʃᵊn]: Abweichung

indeed required to find a suitable tradeoff between shorter space charge regions resulting due to higher doping and increased mobility resulting from lower doping concentrations.

Thus, an application-dependent tradeoff between increased mobility and reduced resistance is necessary. Most importantly, impurities are the fundamental basis for the functionality of all semiconductor devices.

Under the influence of an electric field the electrons accelerate and change their momentum according to (2.1). It is important to realize, however, that this change in momentum is normally small compared to the thermal energy. Therefore the electrons still move in a chaotic thermal way and only slightly change their momentum. Assuming that these particles behave like classical particles we can specify both their position and momentum at the same time. It is important to understand that in a more rigorous approach, where these particles are treated as quantum-mechanical wave packets, this is not possible because it violates Heisenberg's uncertainty principle

$$\Delta \boldsymbol{p} \Delta \boldsymbol{r} \geq \frac{\hbar}{2} \,.$$

In theory, we could solve Newton's equations of motion

$$\frac{\mathrm{d}\boldsymbol{p}_i}{\mathrm{d}t} = \boldsymbol{F}(\boldsymbol{p},\boldsymbol{r},t) + \boldsymbol{R}(\boldsymbol{p},\boldsymbol{r},t) \tag{2.2}$$

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = \boldsymbol{u}_i(t) \qquad\qquad i = 1,\ldots,N \tag{2.3}$$

for each of the N carriers. The position of the carrier i is given as $\boldsymbol{r}_i(t)$ and its momentum as $\boldsymbol{p}_i(t)$. $\boldsymbol{R}(\boldsymbol{p},\boldsymbol{r},t)$ is a random force which introduces the effect of lattice vibrations and impurities into the model, $\boldsymbol{F}(\boldsymbol{p},\boldsymbol{r},t)$ is the externally applied force, which is given for electrons as $\boldsymbol{F}(\boldsymbol{p},\boldsymbol{r},t) = -\mathrm{q}\boldsymbol{E}(\boldsymbol{r},t)$ (for negligible magnetic fields).

The current flowing out of a device can be obtained by counting the carriers moving through the contact area. In a realistic device the number of free carriers N can be very large ($N \gg 10^{20}$) and therefore only a representative sample of carriers can be considered in a practical implementation. However, even such a representative sample requires a large number of particles ($N > 10^5$) and a direct solution of (2.2) and (2.3) is thus very time consuming. Such an approach is, despite its pitfalls, used in so-called *Monte Carlo simulations* to provide very accurate solutions of the problem.

Instead of considering a large number of carriers represented by their momentum and position $(\boldsymbol{p}_i(t), \boldsymbol{r}_i(t))$ we just consider their statistical properties. We do this by defining a distribution function $f(\boldsymbol{p},\boldsymbol{r},t)$ in such a way that $f(\boldsymbol{p},\boldsymbol{r},t)\mathrm{d}\boldsymbol{p}\mathrm{d}\boldsymbol{r}$ gives the probability of finding a carrier with a momentum in the range $[\boldsymbol{p}, \boldsymbol{p}+\mathrm{d}\boldsymbol{p}]$ and a position inside the volume $[\boldsymbol{r}, \boldsymbol{r}+\mathrm{d}\boldsymbol{r}]$. The distribution function considered here is therefore a *classical* concept as it defines both momentum and position of the particles and thus is in contradiction to Heisenbergs uncertainty principle.

The distribution function $f(\boldsymbol{p},\boldsymbol{r},t)$ satisfies the *Boltzmann Transport Equation*

$$\frac{\partial f}{\partial t} + \boldsymbol{u} \cdot \nabla_{\boldsymbol{r}} f + \boldsymbol{F} \cdot \nabla_{\boldsymbol{p}} f = Q(f) \,, \tag{2.4}$$

where the *scattering operator* Q is given as

$$Q(f) = \sum_{\boldsymbol{p}'} f(\boldsymbol{p}')[1 - f(\boldsymbol{p})]S(\boldsymbol{p}',\boldsymbol{p}) - \sum_{\boldsymbol{p}'} f(\boldsymbol{p})[1 - f(\boldsymbol{p}')]S(\boldsymbol{p},\boldsymbol{p}') \,. \tag{2.5}$$

The equation describes the kinetic behavior of gases, in our case the electron gas.

Scattering is modeled as a number of (elastic or inelastic) collisions between particles and the lattice. The function $S(\boldsymbol{p}', \boldsymbol{p})$ gives the transition rates for particles from the state denoted by the momentum $\boldsymbol{p}'$ to the state of momentum $\boldsymbol{p}$ after the scattering event and vice versa for $S(\boldsymbol{p}, \boldsymbol{p}')$. It has to be **emphasized**[1] that the scattering operator $Q(f)$ in (2.5) takes the *Pauli Principle* into account: The terms $f(\boldsymbol{p}')$ and $f(\boldsymbol{p})$ require the present states $\boldsymbol{p}'$ and $\boldsymbol{p}$ to be occupied and the new states $\boldsymbol{p}$ and $\boldsymbol{p}'$ to be empty, hence the terms $[1 - f(\boldsymbol{p})]$ and $[1 - f(\boldsymbol{p}')]$. Note that the Pauli principle makes the Boltzmann transport equation *nonlinear* and thus much more complicated to solve. For non-degenerate semiconductors, where the carrier concentration is relatively small ($f \ll 1$), the Pauli principle can be neglected and we obtain with

$$Q(f) \approx \sum_{\boldsymbol{p}'} f(\boldsymbol{p}')S(\boldsymbol{p}', \boldsymbol{p}) - \sum_{\boldsymbol{p}'} f(\boldsymbol{p})S(\boldsymbol{p}, \boldsymbol{p}') \tag{2.6}$$

a linear equation.

The solution of Boltzmann's transport equation provides excellent results, but it is very difficult to solve due to its higher dimensionality compared to the drift-diffusion model. For a full three-dimensional simulation, we have three spatial coordinates, three momentum coordinates and one time coordinate, thus a seven-dimensional simulation would be necessary. This means that using a **discretization**[2] with 100 unknowns in each coordinate direction, we would have to deal with a total of 10^{14} points! If we assume 7×8 bytes (i.e. eight byte for each coordinate) of memory for storing the location of each point, 5.600 Terabytes of memory would be needed for storing the locations of the points in the seven-dimensional simulation space only!

2.2 The Equilibrium Case

In the equilibrium case, the time **derivative**[3] in the Boltzmann transport equation (2.4) vanishes, hence $f(\boldsymbol{p}, \boldsymbol{r}, t) = f(\boldsymbol{p}, \boldsymbol{r})$. Furthermore, in equilibrium the scattering operator Q vanishes, because transitions from momentum $\boldsymbol{p}'$ to momentum $\boldsymbol{p}$ have the same probability as transitions from momentum $\boldsymbol{p}$ to momentum $\boldsymbol{p}'$ (principle of detailed balance). It can be shown that the equilibrium solution is the *Fermi-Dirac distribution*

$$f(\boldsymbol{p}, \boldsymbol{r}) = \frac{1}{1 + \exp\left(\frac{E_{\mathrm{tot}}(\boldsymbol{p}, \boldsymbol{r}) - E_{\mathrm{F}}}{k_{\mathrm{B}} T_{\mathrm{L}}}\right)} , \tag{2.7}$$

where E_{F} is the Fermi level and $E_{\mathrm{tot}}(\boldsymbol{p})$ the total carrier energy given as the sum of potential and kinetic energy of the carriers:

$$E_{\mathrm{tot}}(\boldsymbol{p}) = E_{\mathrm{pot}}(\boldsymbol{r}) + \mathcal{E}_{\mathrm{kin}}(\boldsymbol{p}, \boldsymbol{r}) . \tag{2.8}$$

The potential energy of an electron is given by the band edge of the conduction band subject to an externally applied potential ψ

$$E_{\mathrm{pot}}(\boldsymbol{p}) = E_{\mathrm{c}} = E_{\mathrm{c},0}(\boldsymbol{r}) - \mathrm{q}\psi(\boldsymbol{r}) \tag{2.9}$$

for electrons, so that we find

$$E_{\mathrm{tot}}(\boldsymbol{p}) = E_{\mathrm{c}}(\boldsymbol{r}) + \mathcal{E}(\boldsymbol{p}, \boldsymbol{r}) . \tag{2.10}$$

[1] **to emphasize** [emp.fə.saɪz]: betonen, hervorheben [2] **discretization** [dɪˈskriːtɪˈseɪ.ʃᵊn]: Diskretisierung, vgl. Kapitel 3 [3] **derivative** [dɪˈrɪv.ə.t̬ɪv]: Ableitung

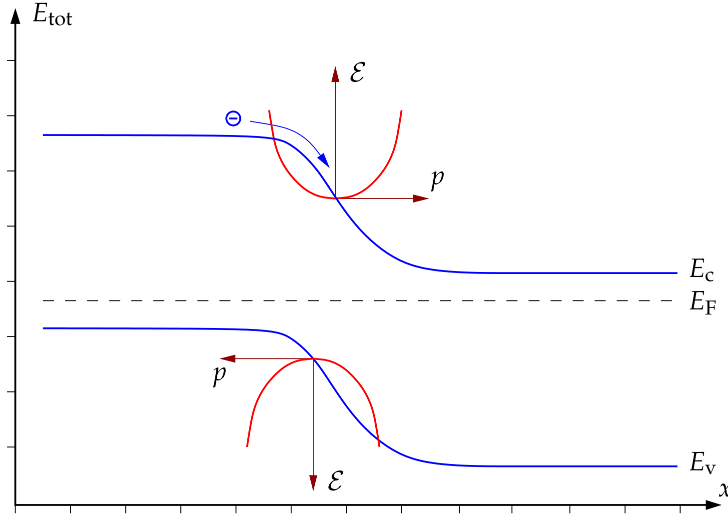


Figure 2.3: Close to the band edges, which determine the potential energy (blue), the dispersion relation can be approximated by parabolas. The effective masses of electrons and holes give the curvature (which may also depend on the location in space) and determine the kinetic energy (red).

$\mathcal{E}(\boldsymbol{p}, \boldsymbol{r})$ is the kinetic energy given by the band structure. In many cases, this *dispersion relation* (or $E - k$-relation) is assumed to be spheric[1] (isotropic) parabolic [2] near the band edge, hence

$$\mathcal{E}(\boldsymbol{p}, \boldsymbol{r}) = \frac{|\boldsymbol{p}|^2}{2m^*(\boldsymbol{r})} = \frac{m^*(\boldsymbol{r})|\boldsymbol{u}|^2}{2}, \tag{2.11}$$

where m^* is the *effective mass*. Please keep in mind the analogy with the classical formula $E_{\text{kin}} = p^2/2m = mv^2/2$: The effective mass can thus be interpreted as a (fictitious) mass of the electron such that the classical energy formula holds. In more mathematical terms, m^* arises from the Taylor expansion of any arbitrarily complicated band structure and is defined in such a way that the parabolic relation for $\mathcal{E}(\boldsymbol{p}, \boldsymbol{r})$ has the "correct" curvature near the band edge with respect to $\boldsymbol{p}$.

As already mentioned above, the Fermi-Dirac distribution honors the *Pauli Principle*: Each state can only be occupied by two electrons. Thus, by means of the Fermi-Dirac distribution it is possible to simulate degenerate semiconductors[3].

For non-degenerate semiconductors there holds $\mathcal{E}_c - E_F \gg k_B T_L$, so (2.7) simplifies to the *Maxwell-Boltzmann distribution*

$$f(\boldsymbol{p}, \boldsymbol{r}) = \exp\left(\frac{E_F - E_{\text{tot}}(\boldsymbol{p})}{k_B T_L}\right) = \exp\left(\frac{E_F - E_c}{k_B T_L}\right) \exp\left(-\frac{\mathcal{E}(\boldsymbol{p})}{k_B T_L}\right) = A \exp\left(-\frac{\mathcal{E}(\boldsymbol{p})}{k_B T_L}\right). \tag{2.12}$$

Unlike the Fermi-Dirac distribution, the Maxwell-Boltzmann distribution neglects the Pauli-Principle, thus it is not suitable for the simulation and modeling of degenerate semiconductors. From a formal point of view, the Maxwell-Boltzmann distribution is the solution of the Boltzmann equation if the scattering operator $Q(f)$ in (2.6) is used instead of (2.5).

[1] i.e. independent of the angle, a function of $|\boldsymbol{p}|$ only [2] i.e. a term of power two [3] A semiconductor is called *degenerate*, if its doping is so high that the individual dopant states in the band diagram merge to a so-called impurity band (located in the band gap of the intrinsic material). The material does not show the typical characteristics of a semiconductor anymore.

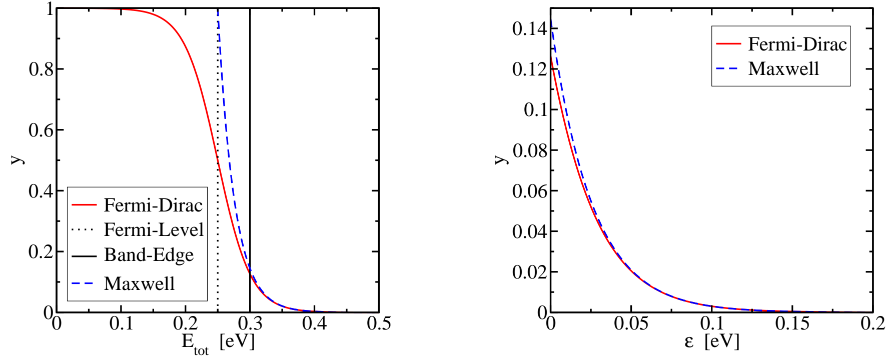


Figure 2.4: A comparison of the Fermi-Dirac and the Maxwell-Boltzmann distributions. It can be seen that the approximation in the conduction band (right) is quite good, but as soon as the Fermi level E_F approaches E_c (or E_v), the Fermi-Dirac distribution has to be used.

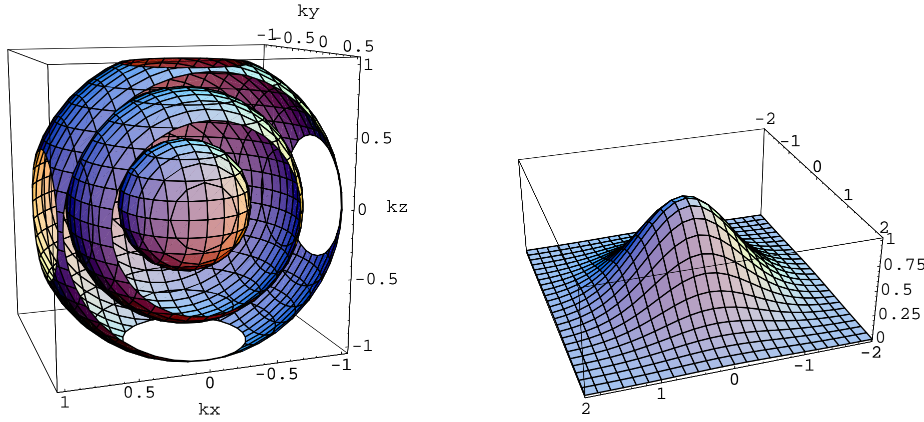


Figure 2.5: Maxwell-Boltzmann distribution for parabolic bands (2.12). Isosurfaces are plotted in three dimensions (left). The two-dimensional plot better illustrates the underlying Gaussian normal distribution (right).

For parabolic bands with $\mathcal{E}(\boldsymbol{p}) = |\boldsymbol{p}|^2/(2m^*)$ we obtain:

$$f(\boldsymbol{p},\boldsymbol{r}) = A\exp\left(-\frac{|\boldsymbol{p}|^2}{2m^*\mathrm{k_B}T_\mathrm{L}}\right) , \tag{2.13}$$

which corresponds to a normal distribution with mean zero and variance $\sigma^2 = m^*\mathrm{k_B}T_\mathrm{L}$. A comparison of the Fermi-Dirac and the Maxwell-Boltzmann distributions is given in Fig. 2.4, while from Fig. 2.5 we see that the average momentum is zero. However, a significant number of electrons occupy higher momentum states.

From the distribution of momentum we are now able to find the associated electron velocities: Let us consider the momentum $\boldsymbol{p}_\mathrm{th}$ such that $f(\boldsymbol{p}_\mathrm{th}) = A\exp(-1)$, such that 85 percent (which corresponds to a deviation of $\pm\sigma\sqrt{2}$ from the mean value of the normal distribution) of the particles are covered. From (2.12) we find the *thermal velocity* v_th as

$$|\boldsymbol{p}_\mathrm{th}| = \sqrt{2m^*\mathrm{k_B}T_\mathrm{L}} \quad \Rightarrow \quad v_\mathrm{th} = \frac{|\boldsymbol{p}_\mathrm{th}|}{m^*} = \sqrt{\frac{2\mathrm{k_B}T_\mathrm{L}}{m^*}} \approx 10^7\mathrm{cm/s} . \tag{2.14}$$

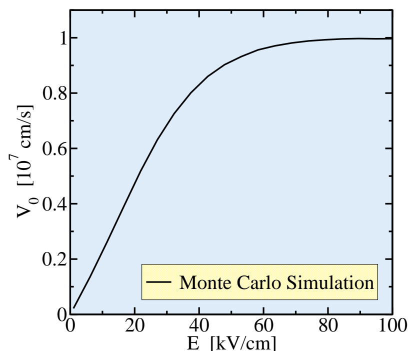


Figure 2.6: Results obtained by a Monte Carlo simulation show that the velocity of electrons due to an externally applied field saturates.

Let us compare the thermal velocity with typical velocities due to externally applied fields: With an electron mobility of $\mu_\mathrm{n} = 200\ \mathrm{cm}^2/(\mathrm{Vs})$ and an electric field of $E = 50\ \mathrm{kV/cm}$, the relation $v = \mu_\mathrm{n} E$ yields $v \approx 10^7 \mathrm{cm/s}$. We will see in Chapter 7 that higher electric fields do not lead to higher velocities — the velocity saturates. This so-called *saturation velocity* is approximately equal to the thermal velocity in silicon, but in other semiconductors these two velocities differ. Results obtained from Monte Carlo simulations are shown in Fig. 2.6.

2.3 Boundary Conditions

The basic semiconductor equations are posed in a bounded domain $\Omega \in \mathbb{R}^n$ $(n = 1, 2, 3)$. At the boundary $\partial\Omega$ of Ω appropriate boundary conditions need to be specified for the variables ψ, n, and p. Two simple cases will be discussed in the following: Ideal Ohmic contacts and artificial boundary conditions. Models for another important case, Schottky contacts, can be found for instance in [**?**, **?**]. Before we can discuss these boundary conditions, the issue of contact potentials, Fermi- and quasi-Fermi levels is briefly reviewed.

2.3.1 Quasi-Fermi Levels

Let us consider the following pnp-diode consisting of three consecutive regions: The width of each region is $L = 1\mu\mathrm{m}$ and the dopings are $N_\mathrm{A} = N_\mathrm{D} = 10^{16}\mathrm{cm}^{-3}$. In the stationary case all time derivatives vanish, hence we have to solve the following equations:

$$\nabla^2\psi = \frac{\mathrm{q}}{\varepsilon}\left(n - p + N_\mathrm{A} - N_\mathrm{D}\right) \tag{2.15}$$

$$\nabla \cdot \left(+\mu_\mathrm{n} \mathrm{k_B} T_\mathrm{L} \nabla n - \mathrm{q}\mu_\mathrm{n} n \nabla\psi\right) = +\mathrm{q}R \tag{2.16}$$

$$\nabla \cdot \left(-\mu_\mathrm{p} \mathrm{k_B} T_\mathrm{L} \nabla p - \mathrm{q}\mu_\mathrm{p} p \nabla\psi\right) = -\mathrm{q}R \tag{2.17}$$

The first equation is the Poisson equation for the potential, while the second and third equations refer to electron and hole conservation and are obtained from (1.23)-(1.25) by setting the time derivative to zero. We have to supply boundary conditions for each second order partial differential equation[1]. Intuitively, we may use as boundary conditions for Poisson's equations

[1] In the present one-dimensional stationary setting, each unknown quantity depends on one variable only, thus we are dealing with a system of ordinary differential equations.

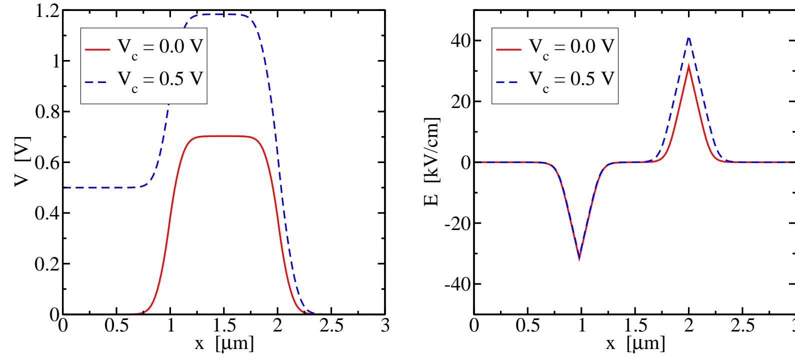


Figure 2.7: Simulation results for a pnp-diode for two different bias conditions. For opposite polarity of V_c, the results are reversed due to symmetry considerations.

the potentials applied to the contacts on either side of the device:

$$\psi(x=0) = V_c, \quad \psi(x=3L) = 0\,.$$

As boundary conditions for the continuity equations it appears natural to use the thermal equilibrium concentrations at the contacts (Ohmic contacts, that will be explained later):

$$p(x=0) = N_A\,, \qquad p(x=3L) = N_A\,, \tag{2.18}$$

$$n(x=0) = n_i^2/N_A\,, \qquad n(x=3L) = n_i^2/N_A\,. \tag{2.19}$$

Simulation results for this setting are given in Fig. 2.7. It is interesting to see that there is a **overshoot**[1] of the electrostatic potential in the center (n-region). This contradicts at first sight the maximum principle for elliptic PDEs, but on closer inspection we see that the maximum principle cannot be applied for the stationary semiconductor device equation. We will come back to this later.

In thermal equilibrium, the number of electrons and holes is given by Boltzmann statistics:

$$n_0 = N_c \exp\left(\frac{E_F - E_c}{k_B T_L}\right) = n_i \exp\left(\frac{E_F - E_i}{k_B T_L}\right)\,, \tag{2.20}$$

$$p_0 = N_v \exp\left(\frac{E_v - E_F}{k_B T_L}\right) = p_i \exp\left(\frac{E_i - E_F}{k_B T_L}\right)\,. \tag{2.21}$$

Here, N_c and N_v are the *effective density of states* or *band weights* of the conduction and valence band respectively. Multiplying these two equations with each other we obtain $n_0 p_0 = n_i p_i = n_i^2$. One has to keep in mind that the thermal equilibrium does *not* require the potential to be position-independent: Typically, we have

$$E_c = E_{c,0}(\boldsymbol{r}) - q\psi(\boldsymbol{r})\,, \tag{2.22}$$

$$E_v = E_{v,0}(\boldsymbol{r}) - q\psi(\boldsymbol{r})\,, \tag{2.23}$$

$$E_i = E_{i,0}(\boldsymbol{r}) - q\psi(\boldsymbol{r})\,. \tag{2.24}$$

[1] **overshoot** [əʊ.və'ʃu:t]: die Überhöhung

Away from the thermal equilibrium, the situation is more complicated. Let us rewrite the current relation (1.21) in such a way that the current is the gradient of a some quantity:

$$\begin{aligned} \boldsymbol{J}_n &= q\mu_n V_T \nabla n - q\mu_n n \nabla \psi \\ &= q\mu_n n \Big(V_T \frac{1}{n} \nabla n - \nabla \psi \Big) \\ &= q\mu_n n \Big(V_T \frac{n_i}{n} \nabla \frac{n}{n_i} - \nabla \psi \Big) \\ &= q\mu_n n \Big(V_T \nabla \ln\Big(\frac{n}{n_i}\Big) - \nabla \psi \Big) \\ &= q\mu_n n \nabla \underbrace{\Big(V_T \ln\Big(\frac{n}{n_i}\Big) - \psi \Big)}_{=: -\phi_n} . \end{aligned}$$

Writing the electron concentration n as a function of ϕ_n, we obtain

$$n = n_i \exp\Big(-\frac{\phi_n}{V_T}\Big) \exp\Big(\frac{\psi}{V_T}\Big) . \tag{2.25}$$

Comparing with (2.20), the introduction of *quasi-Fermi levels* is apparent. They are also called *imref*s, which stands for *im*aginary *ref*erence and quite conveniently corresponds to the word Fermi spelled backwards. In contrast to the Fermi level, which is a unique global energy level, quasi-Fermi levels refer to a subsystem such as electrons and holes only, may depend on the location, and are introduced via the relationships

$$n = n_i \exp\left(\frac{E_{Fn} - E_i}{k_B T_L}\right) = n_i \exp\left(\frac{E_{Fn} - E_{i,0}}{k_B T_L}\right) \exp\left(\frac{\psi}{V_T}\right) , \tag{2.26}$$

$$p = p_i \exp\left(\frac{E_i - E_{Fp}}{k_B T_L}\right) = p_i \exp\left(\frac{E_{i,0} - E_{Fp}}{k_B T_L}\right) \exp\left(-\frac{\psi}{V_T}\right) . \tag{2.27}$$

While in thermal equilibrium $n_0 p_0 = n_i^2$ holds, we find after multiplication of (2.26) with (2.27) that

$$pn = n_i^2 \exp\left(\frac{E_{Fn} - E_{Fp}}{kT}\right) \tag{2.28}$$

and we see that in equilibrium indeed $E_{Fn} = E_{Fp} = E_F$ holds, so that the definition of quasi Fermi levels is consistent. The auxiliary quantity ϕ_n relates to the quasi-Fermi levels as

$$-q\phi_n = E_{Fn} - E_{i,0} . \tag{2.29}$$

Since the intrinsic energy $E_{i,0}$ is a globally constant level, we therefore arrive at

$$\boldsymbol{J}_n = n\mu_n \nabla E_{Fn} , \tag{2.30}$$

while a similar **calculation**[1] for holes gives

$$\boldsymbol{J}_p = p\mu_p \nabla E_{Fp} . \tag{2.31}$$

Thus, in the most general case, *the current depends on the gradient of the quasi-Fermi-levels, not on the gradient of the potential*!

[1] **calculation** [kæl.kjʊˈleɪ.ʃᵊn]: Berechnung

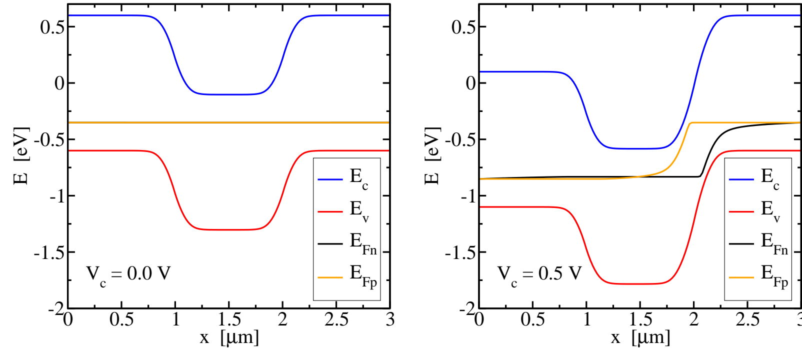
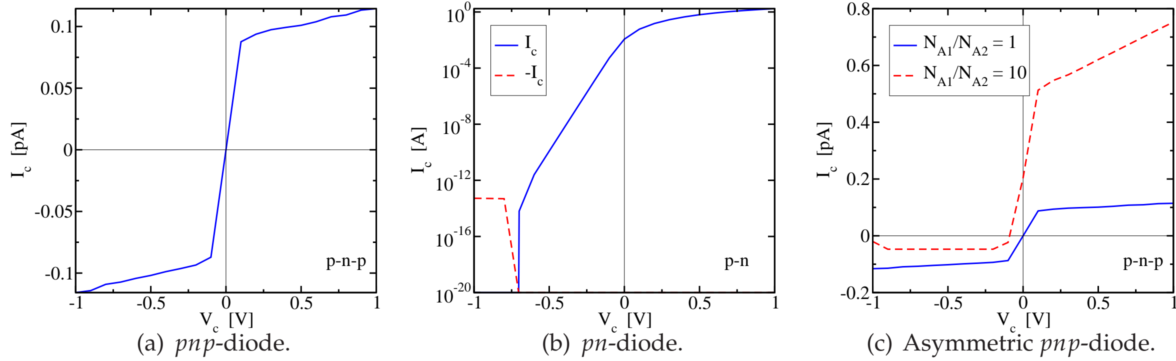


Figure 2.8: Quasi-Fermi levels of the *pnp*-diode for the equilibrium and non-equilibrium case.



(a) *pnp*-diode. (b) *pn*-diode. (c) Asymmetric *pnp*-diode.

Figure 2.9: Current-voltage relations for simple devices.

Let us come back to the *pnp*-diode. If we assume that a solution of (2.15) - (2.17) is available, we can compute the quasi-Fermi levels in a post-processing step by rearranging (2.26) and (2.27):

$$E_{Fn} = E_{i,0} - q\psi + qV_T \ln\left(\frac{n}{n_i}\right) = E_{i,0} - q\psi + k_B T_L \ln\left(\frac{n}{n_i}\right) , \tag{2.32}$$

$$E_{Fp} = E_{i,0} - q\psi - qV_T \ln\left(\frac{p}{n_i}\right) = E_{i,0} - q\psi - k_B T_L \ln\left(\frac{p}{n_i}\right) . \tag{2.33}$$

Since we are free to select a reference energy level, we select $E_{i,0} = 0$ and find with $E_g \approx 1.2\text{eV}$, $E_{c,0} \approx E_{i,0} + E_g/2$ and $E_{v,0} \approx E_{i,0} - E_g/2$ the quasi-Fermi levels shown in Fig. 2.8, where one should note the voltage drop on the second *np*-junction because of the reverse-bias there.

Since we can see the *pnp*-diode as a sequence of two *pn*-diodes with opposite conducting direction, we expect a current blockade for both forward- and reverse-bias. Looking at the current-voltage relation of the *pnp*-diode (Fig. 2.9) for applied voltages between -1V and $+1\text{V}$ shows — despite the rather limited range of applicability of such a device — indeed good agreement with the expected behavior. However, if we remove the second *p*-region, the obtained *pn*-diode behaves like a battery (i.e. it sources current without any external voltage applied). The situation is similar if we choose different doping levels for the two *p*-regions in our *pnp*-diode (Fig. 2.9).

Such passive devices cannot behave like batteries, therefore a quite natural idea is to "shift" the current-voltage relation such that it crosses the origin. In the next section we will investigate the reasons for the observed "offset" and find that such a "shift" of the current-voltage relation

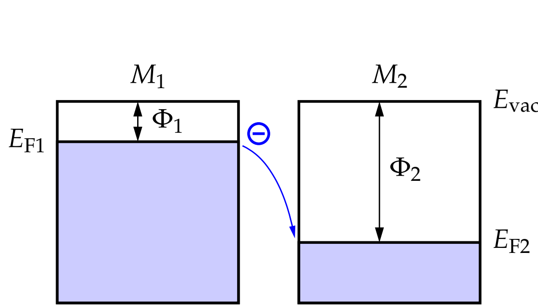


Figure 2.10: Materials M_1 and M_2 before joining, where each material has its own Fermi-level.

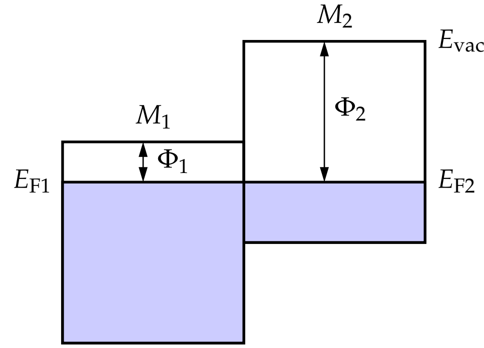


Figure 2.11: After joining materials M_1 and M_2, band energies are shifted so that a single global Fermi-level is obtained.

is not deliberate but in fact required and has a sound physical background.

2.3.2 Contact Potentials

The Fermi level plays an important role in formulating equilibrium conditions when two materials are brought into contact. The combined system will be in thermal equilibrium only when E_F is the same in both parts, because all quantum levels at a given energy must have equal occupation probability at thermal equilibrium. If E_F in each material – relative to a common reference – is not initially equal before contact, then on contact there will be a flow of electrons from the material with the higher initial E_F to the material with the lower initial E_F. This electron flow will continue until equality of the Fermi energies of the two systems is achieved [**?**].

No electric field exists in the initially neutral materials. However, as each **electron**[1] crosses the **junction**[2], it leaves behind a net charge of opposite polarity, and an electric field is thus established in the **vicinity**[3] of the junction, which tends to inhibit the movement of the electrons. This movement results in space-charge regions of different sign on either side of the junction, where a **dipole**[4] layer of finite thickness is formed which we are going to model with an idealized abrupt profile. An electrostatic potential change is then encountered when going from one material, through the junction, to the other material. In thermal equilibrium, the total potential drop in going from M_1 to M_2 is called the *contact potential* ψ_{12} of material M_1 to M_2 [**?**].

$$q\psi_{12} = E_{F1} - E_{F2} \tag{2.34}$$

Consider the case depicted in Fig. 2.10, where $E_{F1} > E_{F2}$. Material M_1 will lose electrons and will thus become positively charged, while M_2 will receive electrons and will thus become negatively charged. Therefore, the potential difference between M_1 and M_2 will be positive, as stated by (2.34). From this model it appears that there is a discontinuity of the potential at the metal-metal interface. This is, however, not the case: The potential changes continuously in a very thin (dipole) layer located at the interface.

Frequently one uses the *work-functions* $q\Phi_1$ and $q\Phi_2$, which are positive quantities and give the distance between the Fermi energies and the *vacuum energy* E_{vac}, i.e. they correspond to the

[1] **electron** [ɪˈlektrɑːn], **NOT** [ˈəlektrɑːn]: Elektron [2] **junction** [ˈdʒʌŋkʃᵊn]: die Sperrschicht [3] **vicinity** [vəˈsɪnəṭi]: Umgebung [4] **dipole** [ˈdaɪpoʊl]: Dipol

work needed to remove an electron from the lattice,

$$q\Phi_1 = E_{\mathrm{vac}} - E_{\mathrm{F1}} ,$$
$$q\Phi_2 = E_{\mathrm{vac}} - E_{\mathrm{F2}} .$$

With this definition the contact potential ψ_{12} can be expressed in terms of the work functions of the materials as (note the change in sign!)

$$\psi_{12} = \Phi_2 - \Phi_1 . \tag{2.35}$$

If N materials are considered in series, the potential ψ_{1N} between the first and the last material is expressed in terms of the work functions in the loop. Using (2.35) yields

$$\psi_{1N} = (\Phi_2 - \Phi_1) + (\Phi_3 - \Phi_2) + \ldots + (\Phi_N - \Phi_{N-1}) .$$

It is clear that, with the exception of Φ_1 and Φ_N, each contact potentials appears twice in the sum: once with a plus sign and once with a minus sign. Therefore

$$\psi_{1N} = \Phi_N - \Phi_1 . \tag{2.36}$$

Thus **no matter**[1] how many materials are in the chain, the electrostatic potential difference between its two ends depends *only* on the *first* and the *last* material (cf. electrochemical series).

When neutral n-type and p-type semiconductors are brought together to form a junction, at the interface a space-charge region forms, and a contact potential ψ_{D} appears. ψ_{D} is also referred to as *diffusion voltage* or *built-in potential* of the pn–junction, and is given by (2.34) as

$$q\psi_{\mathrm{D}} = E_{\mathrm{F}n} - E_{\mathrm{F}p} .$$

Note that the built-in potential is defined as the contact potential between the n– and the p–layer, which is thus a positive quantity, but one commonly speaks of pn–junctions rather than of np–junctions.

According to (2.32) and (2.33) the Fermi energies of neutral semiconductors are

$$E_{\mathrm{F}n} = E_{\mathrm{i}} + \mathrm{k_B} T_{\mathrm{L}} \ln\left(\frac{n}{n_{\mathrm{i}}}\right) ,$$
$$E_{\mathrm{F}p} = E_{\mathrm{i}} - \mathrm{k_B} T_{\mathrm{L}} \ln\left(\frac{p}{n_{\mathrm{i}}}\right) .$$

Setting $n = N_{\mathrm{D}}$ and $p = N_{\mathrm{A}}$ the contact potential of the pn-junction is obtained as

$$\psi_{\mathrm{D}} = \frac{\mathrm{k_B} T_{\mathrm{L}}}{q} \ln\left(\frac{np}{n_{\mathrm{i}}^2}\right) = V_{\mathrm{T}} \ln\left(\frac{N_{\mathrm{A}} N_{\mathrm{D}}}{n_{\mathrm{i}}^2}\right) . \tag{2.37}$$

As a next step we consider such a pn-diode sandwiched between two metals M_1 and M_2 as shown in Fig. 2.12 and Fig. 2.13. The contact potential to the metal contacts is idealized as a discontinuity in the potential, even though there cannot be discontinuities of the potential. We can, however, think of a very thin layer at the interface in which the potential changes continuously, so that the assumption of a discontinuous potential can be justified from a modeling point of view. The contact potentials between the individual material layers are then obtained as

$$q\psi_{\mathrm{M1}p} = E_{\mathrm{FM1}} - E_{\mathrm{F}p} ,$$
$$q\psi_{pn} = E_{\mathrm{F}p} - E_{\mathrm{F}n} = -q\psi_{\mathrm{D}} ,$$
$$q\psi_{n\mathrm{M2}} = E_{\mathrm{F}n} - E_{\mathrm{FM2}} .$$

[1] **no matter** [noʊ ˈmæt̬.ɚ]: ganz egal

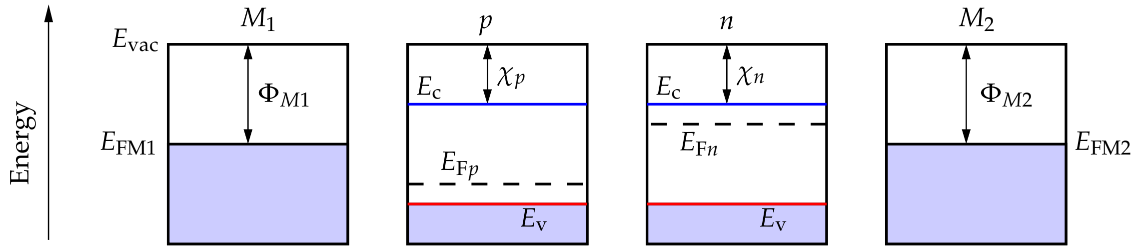


Figure 2.12: A pn–junction sandwiched between two metals M_1 and M_2 before contact.

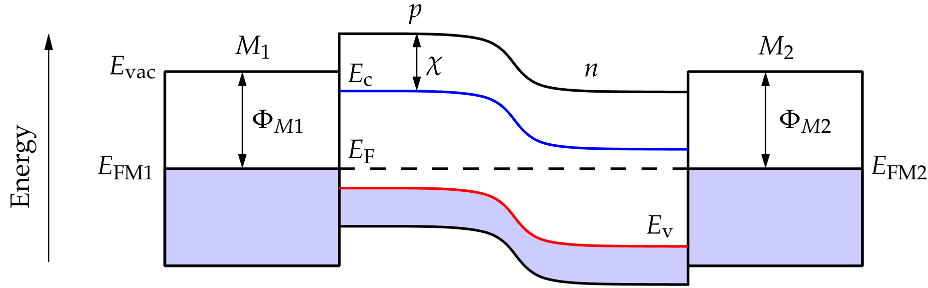


Figure 2.13: A pn–junction sandwiched between two metals M_1 and M_2 after contact. Shown are the band edges inside the semiconductor. The contact potential to the metal contacts is idealized as a discontinuity in the potential and models an infinitesimal thin dipole layer.

By recalling that the difference between the two metal Fermi levels equals the contact potential

$$E_{\mathrm{FM2}} - E_{\mathrm{FM1}} = \mathrm{q}\psi_{12}\,,$$

we rewrite the contact potential $\mathrm{q}\psi_{n\mathrm{M2}}$ as

$$\mathrm{q}\psi_{n\mathrm{M2}} = E_{\mathrm{F}n} - E_{\mathrm{FM2}} = E_{\mathrm{F}n} - E_{\mathrm{FM2}} + E_{\mathrm{FM1}} - E_{\mathrm{FM1}} = E_{\mathrm{F}n} - \mathrm{q}\psi_{12} - E_{\mathrm{M1}}\,.$$

As the reference energy is **arbitrary**[1] we choose $E_{\mathrm{M1}} = 0$ and obtain

$$\psi_{\mathrm{M1}p} = -E_{\mathrm{F}p}/\mathrm{q}\,,$$
$$\psi_{n\mathrm{M2}} = +E_{\mathrm{F}n}/\mathrm{q} - \psi_{12}\,.$$

Since the metals used to contact a semiconductor are normally the same, we continue with the special case where $M_1 = M_2 = M$ and we obtain

$$\psi_{\mathrm{M1p}} = -E_{\mathrm{F}p}/\mathrm{q}\,,$$
$$\psi_{\mathrm{nM2}} = +E_{\mathrm{F}n}/\mathrm{q}\,.$$

The potential distribution inside a pn–junction is shown in Fig. 2.14. The potential ψ_{L} on the left side of the semiconductor is given by

$$\psi_{\mathrm{M1}} - \psi_{\mathrm{L}} = \psi_{\mathrm{M1p}}\,,$$

[1] **arbitrary** ['ɑːrbətreri]: willkürlich, beliebig

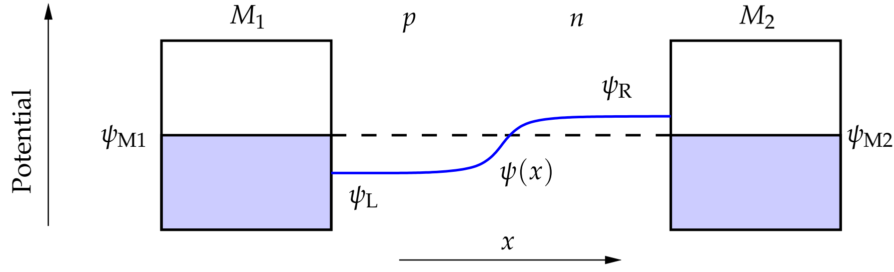


Figure 2.14: The potential distribution of a pn–junction.

and therefore we obtain

$$\psi_{\mathrm{L}} = \psi_{\mathrm{M1}} - \psi_{\mathrm{M1p}} = \psi_{\mathrm{M1}} + E_{\mathrm{F}p}/\mathrm{q}\,.$$

Similarly we obtain for ψ_{R}

$$\psi_{\mathrm{R}} - \psi_{\mathrm{M2}} = \psi_{\mathrm{nM2}}\,,$$

and thus

$$\psi_{\mathrm{R}} = \psi_{\mathrm{M2}} + \psi_{\mathrm{nM2}} = \psi_{\mathrm{M2}} + E_{\mathrm{F}n}/\mathrm{q}\,.$$

Note that $E_{\mathrm{F}p} < 0$ and $E_{\mathrm{F}n} > 0$.

In summary, the potentials at the left and right sides of the semiconductor are given by

$$\begin{aligned} \psi_{\mathrm{L}} &= \psi_{\mathrm{M1}} + E_{\mathrm{F}p}/\mathrm{q} \\ \psi_{\mathrm{R}} &= \psi_{\mathrm{M2}} + E_{\mathrm{F}n}/\mathrm{q} \end{aligned}$$

2.3.3 The Built-in Potential

Assume we have given an inhomogeneous electron concentration $n(\boldsymbol{r})$ inside a semiconductor, which is in thermal equilibrium. We now ask for the potential distribution $\psi(\boldsymbol{r})$ which is necessary to keep the electron distribution $n(\boldsymbol{r})$ in this steady state.

In thermal equilibrium the current densities of either carrier type vanish,

$$\boldsymbol{J}_n = \boldsymbol{J}_p = \boldsymbol{0}\,,$$

which means that an existing diffusion component has to be compensated exactly by a drift component. To guarantee $\boldsymbol{J}_n = \boldsymbol{0}$, it follows from (2.30), that the quasi-Fermi potential has to be constant across the semiconductor and we obtain

$$\psi_n = \psi - V_{\mathrm{T}} \ln\left(\frac{n}{n_{\mathrm{i}}}\right) = \mathrm{const}\,.$$

In this equation one has again the freedom to choose a reference point. It is sound, although arbitrary, to define the quasi-Fermi potential to be zero if the distribution function represents thermal equilibrium, which is the case for a structure to which no external forces are applied.

With this convention, the potential required to retain a given equilibrium carrier concentration $n(\boldsymbol{r})$ is found to be

$$\psi_{\mathrm{bi}}(\boldsymbol{r}) = V_{\mathrm{T}} \ln\left(\frac{n(\boldsymbol{r})}{n_{\mathrm{i}}}\right) . \tag{2.38}$$

ψ_{bi} is termed *built-in potential*. Note that ψ_{bi} depends on the position $\boldsymbol{r}$. Equivalently, one obtains a built-in potential of the form

$$\psi_{\mathrm{bi}}(\boldsymbol{r}) = -V_{\mathrm{T}} \ln\left(\frac{p(\boldsymbol{r})}{n_{\mathrm{i}}}\right) \tag{2.39}$$

by setting $\psi_p = \text{const}$. This equation can also be obtained from (2.38) by setting $n(\boldsymbol{r}) = n_{\mathrm{i}}^2 / p(\boldsymbol{r})$. Although it is already implicitly included in the calculation of this subsection, we have to emphasize again that the built-in potential is the solution for the potential (up to a constant shift) in the drift-diffusion model for the case that the current through the device is zero (i.e. $\boldsymbol{J} = 0$).

The exact built-in potential is a-priori unknown, but for a first guess one can derive an approximation by assuming charge neutrality:

$$p - n + N_{\mathrm{D}}^{+} - N_{\mathrm{A}}^{-} = 0 . \tag{2.40}$$

From (2.38) and (2.39), both n and p can be expressed as functions of the built-in potential so that (2.40) becomes

$$n_{\mathrm{i}} \exp\left(-\frac{\psi}{V_{\mathrm{T}}}\right) - n_{\mathrm{i}} \exp\left(\frac{\psi}{V_{\mathrm{T}}}\right) + N_{\mathrm{D}}^{+} - N_{\mathrm{A}}^{-} = 0 .$$

The potential ψ occuring in the above equation is the built-in potential we have been looking for. This equation can easily be solved and gives

$$\psi_{\mathrm{bi}} = V_{\mathrm{T}} \operatorname{arsinh}\left(\frac{N_{\mathrm{D}}^{+} - N_{\mathrm{A}}^{-}}{2n_{\mathrm{i}}}\right) . \tag{2.41}$$

One should keep in mind that this derivation is strictly valid only for homogeneously doped semiconductors when $\rho = 0$. For non-uniform doping the potential will be a function of position and hence, an electric field will exist. This is possible only if there is a space-charge density, a fact which is contrary to (2.40). Nevertheless, the built-in potential for the inhomogeneous case is frequently approximated by (2.41) in a first step. The obtained solution is then used as an initial guess for the solution of the full non-linear drift-diffusion model by means of Newton's method.

2.4 Ohmic Contacts

There are two possible scenarios of a metal and a semiconductor being brought into contact: The first case is obtained when the Fermi level of the metal is lower than the Fermi level of the semiconductor. One then obtains the so-called *Schottky contact* with a non-linear, rectifying transfer characteristic (Schottky diode). On the other hand, contacting a metal with a semiconductor such that the Fermi level of the metal is higher than that of the semiconductor, a non-rectifying *Ohmic contact* is found in general[1]. The latter is shown in Fig. 2.15.

[1] It is possible to get an Ohmic contact even if the Fermi-level of the metal is lower than that of the semiconductor: With high doping concentrations, the resulting potential barrier is so thin that electrons tunnel through. This is the case for doping concentrations larger than $10^{17}\mathrm{cm}^{-3}$ (concentrations around $10^{20}\mathrm{cm}^{-3}$ are normally used).

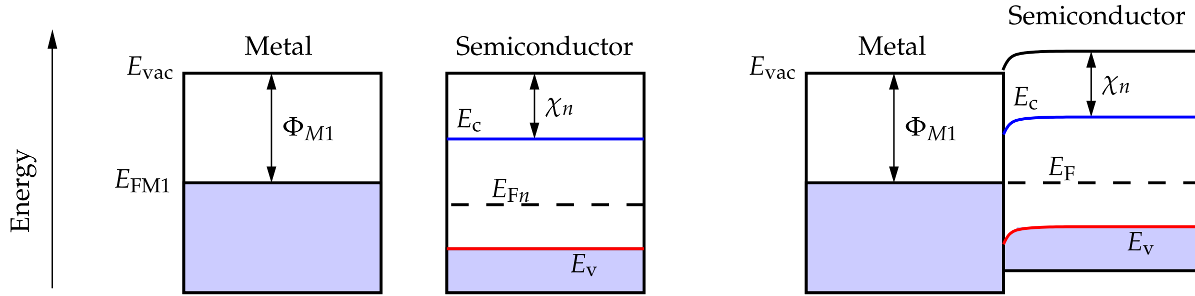


Figure 2.15: Band diagram before (left) and after (right) contacting a metal and a semiconductor in the case of an Ohmic contact.

Ideal Ohmic contacts are the most commonly used boundary conditions for the simulation of semiconductor devices. The primary reason is that they are simple to implement in the form of Dirichlet boundary conditions for the potential and carrier concentrations, leading to very stable computations. From a physical point of view ohmic contacts are rather **crude**[1] and unlikely to be correct. Nevertheless, the hope is that there is minor impact of the contact region on the device behavior, so that ohmic contacts can be justified.

In the contact region any voltage drops are neglected in the simple model for an Ohmic contact. In the band diagram the Fermi level of the metal is continuously connected to the quasi-Fermi level of the semiconductor. Near the contact thermal equilibrium and vanishing space charge are assumed, so that we have (with $C = N_D^+ - N_A^-$)

$$np = n_i^2 \qquad\qquad p - n + C = 0\,.$$

From the resulting quadratic equation we get

$$n = \frac{1}{2}\left(\sqrt{C^2 + 4n_i^2} + C\right),$$
$$p = \frac{1}{2}\left(\sqrt{C^2 + 4n_i^2} - C\right).$$

The important limiting cases for high and low doping are

$$C \gg n_i\ : \qquad n = C\,, \qquad p = n_i^2/C\,,$$
$$C \ll n_i\ : \qquad n = n_i\,, \qquad p = n_i$$

and vice versa for holes. The case $C \gg n_i$, which is usually fulfilled, leads to the boundary conditions we have used so far, thus we can keep them in most cases.

When we apply a bias V_c, we have for the boundary conditions of the potential at an ideal Ohmic contact

$$\psi = V_c - \psi_{MS}\,,$$

[1] **crude** [kruːd]: grob, ungehobelt

where ψ_{MS} is the contact potential of the metal-semiconductor contact obtained from

$$\mathrm{q}\psi_{\mathrm{MS}} = E_{\mathrm{FM}} - E_{\mathrm{FS}} \,.$$

The Fermi level E_{FS} in the semiconductor can be found from the charge neutrality condition $\rho = 0$ ($np = n_{\mathrm{i}}^2$ and thus $E_{\mathrm{F}n} = E_{\mathrm{F}p}$) at the contact:

$$\begin{aligned} E_{\mathrm{FS}} = E_{\mathrm{F}n} = E_{\mathrm{F}p} &= E_{\mathrm{i}} + \mathrm{k_B}\, T_{\mathrm{L}}\, \ln\Big(\frac{n}{n_{\mathrm{i}}}\Big) \\ &= E_{\mathrm{i}} + \mathrm{k_B}\, T_{\mathrm{L}}\, \ln\Big(\frac{1}{2n_{\mathrm{i}}}\Big(\sqrt{C^2 + 4n_{\mathrm{i}}^2} + C\Big)\Big) \\ &= E_{\mathrm{i}} + \mathrm{k_B}\, T_{\mathrm{L}}\, \mathrm{arcsinh}\Big(\frac{C}{2n_{\mathrm{i}}}\Big) \\ &= E_{\mathrm{i}} + \mathrm{q}\psi_{\mathrm{bi}} \,. \end{aligned} \tag{2.42}$$

This finally gives the doping concentration dependent contact potential

$$\psi_{\mathrm{MS}} = \psi_{\mathrm{MS}}(C) = \Phi'_{\mathrm{MS}} - V_{\mathrm{T}}\, \mathrm{arcsinh}\Big(\frac{C}{2n_{\mathrm{i}}}\Big) \,, \tag{2.43}$$

with the zero-doping work function difference $\Phi'_{\mathrm{MS}} = (E_{\mathrm{FM}} - E_{\mathrm{i}})/\mathrm{q}$.

Let us consider the two special cases of highly doped regions as will be used in all simulation examples in this lecture:

- **Highly doped *n*-region**: For $N_{\mathrm{D}}/n_{\mathrm{i}} > 20$, and with $\sqrt{C^2 + 4n_{\mathrm{i}}^2} \approx \sqrt{{N_{\mathrm{D}}}^2 + 4n_{\mathrm{i}}^2} \approx N_{\mathrm{D}}$ we obtain

$$\boxed{\begin{aligned} & n \approx N_{\mathrm{D}} \,, \\ & p \approx n_{\mathrm{i}}^2/n \,, \\ & \psi = V_{\mathrm{c}} - \Phi'_{\mathrm{MS}} + V_{\mathrm{T}}\, \ln(N_{\mathrm{D}}/n_{\mathrm{i}}) \,. \end{aligned}} \tag{2.44}$$

- **Highly doped *p*-region**: For $N_{\mathrm{A}}/n_{\mathrm{i}} > 20$ and with $\sqrt{C^2 + 4n_{\mathrm{i}}^2} \approx \sqrt{{N_{\mathrm{A}}}^2 + 4n_{\mathrm{i}}^2} \approx N_{\mathrm{A}}$ we obtain

$$\boxed{\begin{aligned} & p \approx N_{\mathrm{A}} \,, \\ & n \approx n_{\mathrm{i}}^2/p \,, \\ & \psi = V_{\mathrm{c}} - \Phi'_{\mathrm{MS}} - V_{\mathrm{T}}\, \ln(N_{\mathrm{A}}/n_{\mathrm{i}}) \,. \end{aligned}} \tag{2.45}$$

As closing example we come back to the *pn*-diode where we have found shifted current-voltage characteristics in Sec. 2.3.1. Our aim is to find the device characteristics for an externally applied bias voltage V_{c}. We already know that applying V_{c} at the *p*-region and grounding the *n*-region does not lead to the expected results, because the built-in potential is not taken into account. So, let us denote with V_{L} the voltage applied to the *p*-region (on the left) and with V_{R} the voltage applied to the *n*-region, taking built-in potentials into account. From (2.44) and (2.45) we deduce

$$\begin{aligned} V_{\mathrm{L}} &= V_{\mathrm{c}} - \Phi'_{\mathrm{MS}} - V_{\mathrm{T}}\, \ln(N_{\mathrm{A}}/n_{\mathrm{i}}) \,, \\ V_{\mathrm{R}} &= 0 - \Phi'_{\mathrm{MS}} + V_{\mathrm{T}}\, \ln(N_{\mathrm{D}}/n_{\mathrm{i}}) \,. \end{aligned}$$

This way, we have to modify the voltages applied to both electrodes. However, since the potential reference can be set arbitrarily, we may add $\Phi'_{MS} + V_T \ln(N_A/n_i)$ to both V_L and V_R and get

$$V_L = V_c \,,$$
$$V_R = V_T \ln(N_D/n_i) + V_T \ln(N_A/n_i) = V_T \ln(N_A N_D/n_i^2) = V_D \,.$$

Thus, the simulation results in Fig. 2.9 for the *pn*-diode are indeed shifted by ψ_D because contact potentials have not been taken into account.

Chapter 3

Basics of Numerical Analysis

So far we have considered the continuous mathematical description of physical quantities in a semiconductor. However, for numerical simulation these continuous quantities have to be discretized. This chapter is **devoted**[1] to the basic numerical principles such as the discretization of derivatives and the solution of linear and non-linear systems of equations. We will **readdress**[2] these topics many times throughout this course. A direct translation of vector operators (grad $= \nabla$, div $= \nabla\cdot$ and curl $= \nabla\times$) from their continuous to their numerical representations can be found in Appendix B. In the following chapters we are going to extend the methods shown here with more involved techniques, until we can finally handle the complete set of the semiconductor equations on complex geometries.

3.1 Introduction to Finite Differences

The goal of *finite differences methods* is to approximate differential equations by a system of algebraic equations. This process, known as discretization, involves replacing the derivatives of quantities with differences of the same quantities, **evaluated**[3] at discrete locations.

To introduce numerical differentiation we will make use of variables that are set up as *arrays*. A continuous function is sampled at discrete points such as in Fig. 3.1. Recalling the Nyquist-Shannon sampling theorem in signal processing, the distance between two **adjacent**[4] discrete points is an important parameter for the quality of the numerical approximation.

For the hands-on part we will use SGFRAMEWORK, a **handy**[5] simulation environment developed for the simulation of semiconductors devices. The underlying equations are specified directly as code and a built-in solver computes a solution of the specified system of equations. The equations have to be supplied in a discretized form, which means that we have to apply a discretization method to the continuous formulation. One of the features of SGFRAMEWORK that makes it particularly useful is that it can solve simultaneous equations when the quantity that is being calculated is represented by an array. For example, if we want to find the potential ψ inside a rectangular simulation domain, we would have to settle for finding it at a set of points in the interior. The values of ψ at these points may be represented by an array. The points will form a grid in the simulation domain.

If the independent coordinate **employed**[6] is x and each grid point is at $x_i = i\Delta x$, where i is

[1] **to devote** [dɪˈvoʊ.t̬ɪd]: widmen [2] **to readdress** [ˌriːəˈdres]: sich nocheinmal zuwenden [3] **to evaluate sth.** [ɪˈvæl.ju.eɪt]: etwas auswerten [4] **adjacent** [əˈdʒeɪ.sənt]: benachbart, angrenzend [5] **handy** [hæn.di]: praktisch, geschickt [6] **to employ** [ɪmˈplɔɪ]: einführen, einsetzen

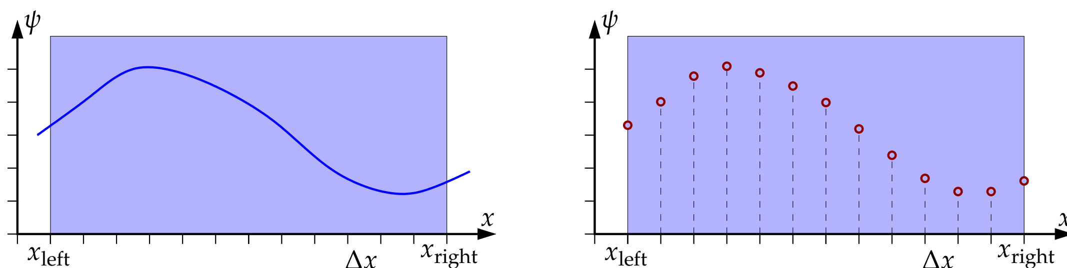


Figure 3.1: A continuous function (left) is represented by a discretized version (right).

an integer and Δx is the distance between two adjacent grid points, we can *label (address)* each x-value by its i-value (Fig. 3.1). If the problem is one-dimensional, we only use one coordinate $x = i\Delta x$, and the array for the quantity is simply $\psi[i]$. From the definition of the derivative

$$\frac{\mathrm{d}\psi(x)}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{\Delta \psi}{\Delta x} = \lim_{\Delta x \to 0} \frac{\psi(x + \Delta x) - \psi(x)}{\Delta x} \tag{3.1}$$

we can find an approximate expression for $\mathrm{d}\psi/\mathrm{d}x$ at the point $x_i = i\Delta x$. This leads in a natural way to the one-sided differences

$$\left.\frac{\mathrm{d}\psi(x)}{\mathrm{d}x}\right|_{x=i\Delta x} \approx \frac{\psi[i+1] - \psi[i]}{\Delta x} \qquad \text{(right-sided difference)} \tag{3.2}$$

and

$$\left.\frac{\mathrm{d}\psi(x)}{\mathrm{d}x}\right|_{x=i\Delta x} \approx \frac{\psi[i] - \psi[i-1]}{\Delta x} \qquad \text{(left-sided difference)} \ . \tag{3.3}$$

Taking the mean value of these two schemes (see Fig. 3.2),

$$\left.\frac{\mathrm{d}\psi(x)}{\mathrm{d}x}\right|_{x=i\Delta x} \approx \frac{\psi[i+1] - \psi[i-1]}{2\Delta x} \qquad \text{(central difference)} \tag{3.4}$$

is obtained. Its name is derived from the fact that it uses values of ψ at points on either side of the point i.

One-sided approximations are usually less **accurate**[1] than the central difference scheme, since they give weight to one side of the point i and ignore the other side, but sometimes they are useful, especially at the boundary of a domain. The replacement of the derivative by the difference's quotient leads to *approximation errors*, which will become larger the further apart the

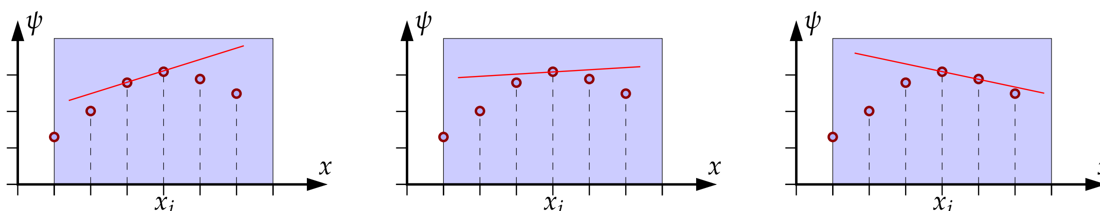


Figure 3.2: Comparison of various numerical difference schemes for $\mathrm{d}\psi/\mathrm{d}x$ at the point x_i: left-sided differences (left), central differences (middle), right-sided differences (right).

[1] **accurate** [æk.jʊ.rət]: genau

discrete points are. On the other hand, Δx cannot be made too small due to the finite representation of floating point numbers in a computer.

As an example let us consider the calculation of the electric field from a given potential. In the one-dimensional case the electric field is defined as

$$E(x) = -\nabla\psi(x) = -\frac{d\psi(x)}{dx} \ . \tag{3.5}$$

Discretizing (3.5) using right-sided differences, we obtain

$$\begin{aligned}
E[0] &= -(\psi[1] - \psi[0])/\Delta x \\
E[1] &= -(\psi[2] - \psi[1])/\Delta x \\
&\vdots \\
E[N-1] &= -(\psi[N] - \psi[N-1])/\Delta x \ .
\end{aligned}$$

This can be written in matrix form

$$\begin{pmatrix} E[0] \\ E[1] \\ \vdots \\ E[N-1] \end{pmatrix} = \frac{1}{\Delta x} \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} \psi[0] \\ \psi[1] \\ \vdots \\ \psi[N-1] \\ \psi[N] \end{pmatrix} \ . \tag{3.6}$$

or in a more compact form as $\boldsymbol{E} = \boldsymbol{A}\psi$. Note that the dimension of the matrix $\boldsymbol{A}$ is $N \times (N+1)$ and that we cannot evaluate $E[N]$ with right-sided differences because there is no right neighbor at $N+1$. We could evaluate $E[N]$ by using left-sided differences as well. This is done in the next example, where left- and right-sided differences are used at the boundary points $i = 0$ and $i = N$ and central differences elsewhere. We obtain

$$\begin{aligned}
E[0] &= -(\psi[1] - \psi[0])/\Delta x \\
E[1] &= -(\psi[2] - \psi[0])/(2\Delta x) \\
&\vdots \\
E[N] &= -(\psi[N] - \psi[N-1])/\Delta x
\end{aligned}$$

or, in compact form

$$\begin{pmatrix} E[0] \\ E[1] \\ \vdots \\ E[N-1] \\ E[N] \end{pmatrix} = \frac{1}{2\Delta x} \begin{pmatrix} 2 & -2 & 0 & 0 & 0 & \cdots \\ 1 & 0 & -1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 & 0 & -1 \\ 0 & \cdots & 0 & 0 & 2 & -2 \end{pmatrix} \begin{pmatrix} \psi[0] \\ \psi[1] \\ \vdots \\ \psi[N-1] \\ \psi[N] \end{pmatrix} \ . \tag{3.7}$$

In matrix form we again have $\boldsymbol{E} = \boldsymbol{A}\psi$, but this time $\boldsymbol{E}$ is of dimension $N+1$ as it should be and $\boldsymbol{A}$ is a square $(N+1) \times (N+1)$ matrix.

File `basics1.sg` shows an SGFRAMEWORK program that allows us to differentiate a one-dimensional function. To do this we use two arrays. We start with $\psi[i]$, the electrostatic potential. The electric field is given by (3.5), so if we know the values in the array $\psi[i]$ we can find the values of the electric field array $E[i]$ from it.

```
const PI  = 3.14159265358979323846;    // pi
const DIM = 101;                       // number of mesh points
const DX  = 2.0*PI / (DIM-1);          // mesh spacing (cm) into (DIM-1) parts

var x[DIM], psi[DIM], E[DIM];

// these equations could be assignment statements
equ E[i=0]        -> E[i] = -(psi[i+1] - psi[i] )  / (1.0*DX);
equ E[i=1..DIM-2] -> E[i] = -(psi[i+1] - psi[i-1]) / (2.0*DX);
equ E[i=DIM-1]    -> E[i] = -(psi[i]   - psi[i-1]) / (1.0*DX);

begin main
  assign x  [i=0..DIM-1] = i*DX;
  assign psi[i=0..DIM-1] = cos(x[i]);
  solve;
  write;
end
```

source_code/basics1.sg

SGFRAMEWORK can solve equation systems of the form $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y}$. In our special case $\boldsymbol{y}$ is given and $\boldsymbol{x}$ can be obtained via a simple matrix multiplication. In real applications, however, $\boldsymbol{x}$ is given and the solution $\boldsymbol{y}$ has to be obtained by formally inverting $\boldsymbol{A}$, that is, solving the equation system $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y}$.

The `assign` command was used to initialize $\psi[i]$. An `assign` statement could have been used as well to find $E[i]$ from $\psi[i]$ since in this case all the **quantities**[1] on the right hand side of the equation are known, but for demonstration purposes this was not done here.

The `equ` statement implies that those values of E indicated are to be found from that equation. The `solve` statement tells the software to solve the equations appearing in the `equ` statement(s). The `equ` statement is a more powerful way of writing an equation than the `assign` statement. In an `assign` statement the right hand side of the equation must be an explicit expression in terms of known quantities, which will be evaluated to give the value of the left-hand side. The expression to the right of the arrow in an `equ` statement is evaluated to give the value of the quantity on the left of the arrow. The quantity to the left of the arrow must appear in the expression to the right. However, in an `equ` statement the expression to the right does not need to be an explicit expression for the quantity on the left. It could (for instance) be a non-linear equation for the quantity on the left which cannot be solved analytically. The left and right hand sides of the expression could both contain the quantity being solved for, or just one of them may contain it. That is, the expression to the right of the arrow can read $f(x) = g(x)$ where x is the unknown. Finally, the expression may contain other unknown quantities which will only become known when a matrix is inverted, as is the case in the next example we will be looking at.

If E and ψ are plotted (Figs. 3.3, 3.4), it should be clear that E does look like $-\mathrm{d}\psi/\mathrm{d}x$ with some small errors due to the way ψ and $\mathrm{d}\psi/\mathrm{d}x$ are discretized using arrays at points $x = i\Delta x$.

3.2 Numerical Solution of Differential Equations

We can immediately modify the equations we just solved and use the same expressions for the derivatives in a different way. Up to now we have used the equations to find the derivative,

[1] **quantities** [ˈkwɑːn.t̬ə.t̬i]: Grøße, hier speziell: Matrixelemente

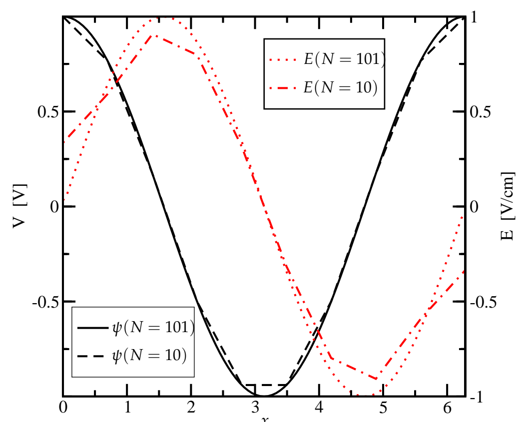


Figure 3.3: Influence of the number of grid points on the final result (10 versus 101 grid points).

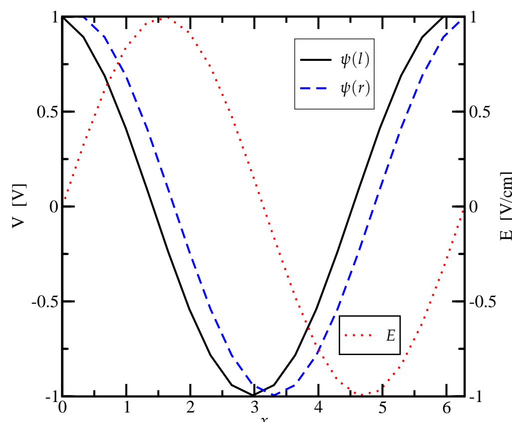


Figure 3.4: Influence of the discretization scheme on the final result (left-sided versus right-sided).

that is, we computed $d\psi/dx$ from ψ. Instead, we can easily use the same approach to find the quantity being differentiated, for instance, we can compute the solution ψ of a differential equation from $d\psi/dx$. In the program `basics2.sg`, we solve a differential equation to illustrate the procedure. (In many of the numerical simulations given here, even small changes in the input file can cause a failure of convergence of the solution process. This in turn can lead to overflows or to domain errors. These will be reported as such, in most cases; but sometimes they will cause the simulation to crash.)

```
const PI  = 3.14159265358979323846;    // pi
const DIM = 101;                       // number of mesh points
const DX  = 2.0 * PI / (DIM-1);        // mesh spacing (cm)

var x[DIM], psi[DIM], E[DIM];

equ psi[i=0]         -> psi[i] = 1.0;  // boundary condition at the left side
equ psi[i=1..DIM-1] ->    E[i] = -(psi[i]-psi[i-1]) / DX;

begin main
  assign x[i=0..DIM-1] = i * DX;
  assign E[i=0..DIM-1] = sin(x[i]);
  solve;
  write;
end
```

source_code/basics2.sg

In this file, instead of initially knowing ψ and calculating E from it, suppose we know E and we **seek**[1] ψ. As a result we have to solve a differential equation of ψ. There is only one vector representing an 'independent variable' in this problem, namely x.

One thing to notice in this example is the boundary condition in this simulation, namely $\psi[0] = 1$. If the statement were **omitted**[2], $\psi[0]$ would be automatically set to zero, since all variables and array elements are initialized to zero until we explicitly change them. Then all the other $\psi[i]$ can be computed from this. The equation in this input file finds $\psi[i]$ from $E[i]$

[1] **to seek sth.** [siːk]: etwas suchen [2] **to omit** [oʊˈmɪt]: auslassen

and from $\psi[i-1]$, and it starts at $i = 1$, so in case the system is solved in this way, $\psi[0]$ has to be specified. Internally, SGFRAMEWORKautomatically handles boundary conditions and then solves the equations appropriately.

The other thing **worth noticing**[1] here is that one of the less accurate one-sided differences was used to represent $\mathrm{d}\psi/\mathrm{d}x$. This type of difference can be easier to handle than the central difference used in the previous example since it allows us to start calculating values at one side of the range of x and work across the domain towards the other side. A central difference scheme would have complicated this process, since $\psi[i+1]$ would appear in the equation for $\psi[i]$, and $\psi[i+1]$ is not found until after $\psi[i]$ is found, if we are working from one side to the other. The one-sided difference allows an explicit expression for $\psi[i]$ to be written in terms of known quantities.

Since we use the `equ` statement to find $\psi[i]$ the use of the one-sided difference was not actually necessary; the `equ` statement calls matrix routines which could have handled the central difference. On the other hand, a little algebra would have been required before we could use an assignment statement appropriate for central differences. The algebra would have allowed us to rewrite our equation so that the right hand side was explicitly known, with $\psi[i]$ being on the left hand side.

3.3 The Second Order Derivative

The equation solved in the previous example was a first order ordinary differential equation (ODE), because the 'highest' derivative that appeared was a first order derivative. The electrostatic potential can also be determined from a given charge density ρ instead of an electric field $\boldsymbol{E}$, using Poisson's equation. In one dimension Poisson's equation is

$$\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2} = -\frac{\rho}{\varepsilon} \quad . \tag{3.8}$$

This is a second order elliptic ODE. To solve it we need to rewrite $\mathrm{d}^2\psi/\mathrm{d}x^2$ using the differences between adjacent values of ψ the same way as we have done for $\mathrm{d}\psi/\mathrm{d}x$.

If we use the same approach as before, the derivative evaluated at the point $x = i\Delta x$ is

$$\left(\frac{\mathrm{d}\psi}{\mathrm{d}x}\right)_{x=i\Delta x} \approx \frac{\psi[i+1]-\psi[i-1]}{2\Delta x} \quad .$$

But if we use the same **reasoning**[2] to find the derivative of $\mathrm{d}\psi/\mathrm{d}x$ then

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}\psi}{\mathrm{d}x}\right)_{x=i\Delta x} \approx \left(\left(\frac{\mathrm{d}\psi}{\mathrm{d}x}\right)_{i+1} - \left(\frac{\mathrm{d}\psi}{\mathrm{d}x}\right)_{i-1}\right)\frac{1}{2\Delta x} \quad .$$

To find the derivative at $x = (i+1)\Delta x$ we take the expression for the derivative at $x = i\Delta x$ and add one to every occurrence of i. Similarly, at $x = (i-1)\Delta x$ we subtract one from i at each occurrence in the approximated derivative. Inserting these differences in the expression above gives

$$\begin{aligned}\left(\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2}\right)_{x=i\Delta x} &\approx \left(\frac{\psi[i+2]-\psi[i]}{2\Delta x} - \frac{\psi[i]-\psi[i-2]}{2\Delta x}\right)\frac{1}{2\Delta x} \\ &= \frac{\psi[i+2]-2\psi[i]+\psi[i-2]}{(2\Delta x)^2} \quad .\end{aligned}$$

[1] **worth noticing** [wɜːθ noʊ.t̬ɪsɪŋ]: erwähnenswert [2] **to reason** [riː.zᵊn]: begründen, überlegen

This second order derivative is evaluated using points $2\Delta x$ away from the central point $x = i\Delta x$ and therefore its **denominator**[1] is $(2\Delta x)^2$. We can think of using an imaginary grid with half the spacing of the real grid. If we had evaluated $\mathrm{d}^2\psi/\mathrm{d}x^2$ on the finer grid, we would have had taken points located $\pm\frac{1}{2}\Delta x$ away from the center point $x = i\Delta x$ and would have had divided by Δx^2 instead of $(2\Delta x)^2$. This means that we can also use

$$\left(\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2}\right)_{x=i\Delta x} \approx \frac{\psi[i+1] - 2\psi[i] + \psi[i-1]}{(\Delta x)^2} \quad . \tag{3.9}$$

An alternative way to obtain the second derivative involves finding the first central derivative at each of two fictional points, $i+1/2$ and $i-1/2$. These are found from the differences $\psi[i+1] - \psi[i]$ and $\psi[i] - \psi[i-1]$ respectively, each divided by Δx. To find the second derivative one then takes the difference between these first derivatives, and divides again by Δx, which again results in the above equation.

The only thing we still have to care about before being able to numerically solve a differential equation are the necessary boundary conditions. A **straightforward**[2] way to see this is to look at the finite difference equation. Whichever value of ψ we are solving for, $\psi[i-1]$, $\psi[i]$ or $\psi[i+1]$, we need to know the other two to find the value we want. For general i we for example find $\psi[i+1]$ from $\psi[i]$ and $\psi[i-1]$. To be able to start this iterative process, we need to know any two $\psi[k]$ and $\psi[j]$ for $k \neq j$. Clearly, the scheme can be applied directly if $k = j \pm 1$ (allows the computation of $\psi[k-1]$ and $\psi[k+2]$, which allows the computation of $\psi[k-2]$ and $\psi[k+3]$ and so on), but even the general case allows the computation of all $\psi[i]$ by solving a system of equations. Our difference equation of the order two therefore requires indeed two boundary conditions[3] . Note that, in contrast to analytical solutions of linear differential equations or difference equations where a *general solution* can be formulated as a linear combination of elements from a solution basis with unknown coefficients, numerical solution by nature can only yield a single solution of a particular problem, and therefore *always requires boundary conditions*.

```
1  const DIM = 100;                    // number of mesh points
2  const W   = 1.0;                    // width of mesh (cm)
3  const DX  = W/(DIM-1);              // mesh spacing (cm)
4  const EPS = 8.854e-14;              // permittivity of free space (F/cm)
5
6  var x[DIM], psi[DIM], rho[DIM];
7
8  // this equation cannot be an assignment statement
9  // the boundary conditions are implicit (psi[0] = 0.0, psi[DIM-1] = 0.0)
10 equ psi[i=1..DIM-2] -> (psi[i+1]-2.0*psi[i]+psi[i-1])/sq(DX) = -rho[i]/EPS;
11
12 begin main
13   assign x[i=all] = i*DX;
14   assign rho[i=1*DIM/4..3*DIM/4] = 1.0e-10*sign(W/2.0-x[i]);
15   solve;
16   write;
17 end
```

source_code/basics3.sg

In program `basics3.sg` the Poisson equation is discretized using the scheme (3.9). In this case, opposed to the examples involving first order differences above, one cannot write an explicit expression for $\psi[i]$ in terms of known quantities. Hence an `equ` statement must be used. SGFRAMEWORK solves the system of equations using matrix techniques when instructed

[1] **denominator** [dɪˈnɑː.mə.neɪ.t̬ɚ]: Nenner [2] **straightforward** [streɪtˈfɔː.wəd]: unkompliziert [3] to be more precise, on each characteristic line, two boundary or initial conditions must be given

to do so by the `solve` statement. The boundary conditions are formulated implicitly: Instead of explicitly specifying values for ψ, the initial setting of $\psi[i] = 0$ provided by SGFRAMEWORK is used. Since the second order finite difference scheme for point i uses the neighboring points $i-1$ and $i+1$, no equations must be given for points $i = 0$ and $i = DIM$.

The 'boxed' shape of ρ in the example leads to a parabolic shape of the potential where $\rho \neq 0$ (**cf**[1]. Fig. 3.5. Remember: A constant charge density gives a linearly increasing magnitude of the electric field by integrating once, and a quadratic dependence of the potential after the second integration). In the regions where $\rho = 0$, the potential increases linearly to satisfy the boundary conditions $\psi[0] = \psi[DIM-1] = 0$.

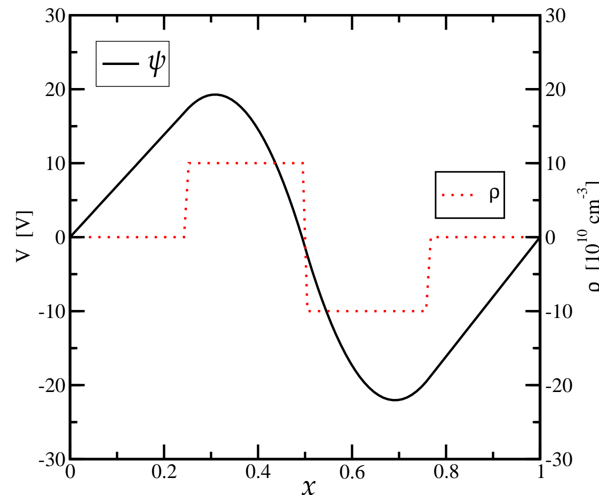


Figure 3.5: The solution of Poisson's equation for a given density of charges.

3.4 Accuracy of Different Discretization Schemes

In numerical analysis a continuous function is replaced by a sampled *(discretized)* one. This discretization happens on a *grid*. The quality of approximation of the continuous function by its sampled counterpart clearly depends on how fine the underlying grid is. To quantify how fine a grid is, the increment (*grid size*) $h_i = x_{i+1} - x_i$ can be defined. Using the increment, grids can be categorized as follows:

- Uniform grid: All h_i are equal, so that the grid consists of equidistant points.
- Quasi-uniform grid: Consecutive increments are of almost the same magnitude. More precisely: $h_{i+1} = h_i(1 + O(h_i))$, where $O(h_i)$ has the meaning of a small local perturbation—the grid distance h_i only slightly differs from its neighbors.
- Non-uniform grid: The h_i are arbitrary.

Even though the discretized version of a function contains less information than the continuous one, it is still possible to find numerical approximations of its derivatives. For a twice differentiable function u, and using the **abbreviations**[2] $u_i = u(x_i)$ (and hence $u_{i+1} = u(x_{i+1}) =$

[1] **cf, to confer** [kən'fɜːʳ]: vergleichen, konsultieren [2] **abbreviation** [ə.briː.vi'eɪ.ʃᵊn]: Abkürzung, Kurzwort

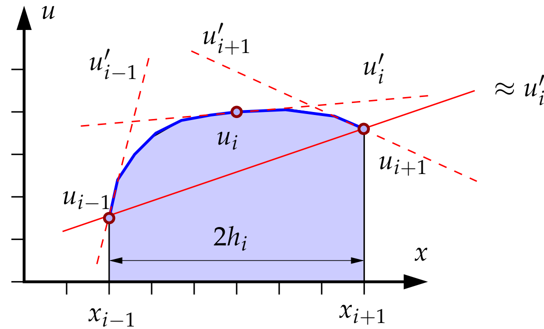


Figure 3.6: Derivative of u at different locations, and the approximation by finite differences.

$u(x_i + h_i))$ for the sake of **brevity**[1], a Taylor **expansion**[2] of u at u_i gives

$$u_{i+1} = u_i + u'_i h_i + O(h_i^2) \quad \Rightarrow \quad u'_i = \frac{u_{i+1} - u_i}{h_i} + O(h_i) \ .$$

The first term on the right hand side is nothing more than the right-sided difference encountered in the previous section. The second term tells us that the *error* introduced by discretization **diminishes**[3] at least linearly with the 'fineness' of the underlying grid, i.e. doubling the number of grid points reduces the discretization error bound by a factor of two. We call a discretization scheme with such a behavior as *first order accurate*. Using Taylor expansions of u_{i+1} and u_{i-1} and assuming a uniform grid with grid size h,

$$\left.\begin{aligned} u_{i+1} &= u_i + u'_i h + O(h^2) \\ u_{i-1} &= u_i - u'_i h + O(h^2) \end{aligned}\right\} \quad \Rightarrow \quad u'_i = \frac{u_{i+1} - u_{i-1}}{2h} + O(h) \ ,$$

it can easily be seen that the discretization error of a central difference scheme also depends linearly on the grid increment. Fig. 3.6 shows that the secant representing the central finite difference is an approximation to the 'real' derivative u'_i. The approximation error clearly depends on the grid size h_i and on the *curvature*, i.e. on the higher order derivatives, of u.

The general error introduced by the finite difference approximation can be analyzed by expanding u into a Taylor series at x_i

$$\begin{aligned} u_{i+1} &= u_i + u'_i h_i + \frac{1}{2!} u''_i h_i^2 + \frac{1}{3!} u_i^{(3)} h_i^3 + O(h_i^4) \ , \\ u_{i-1} &= u_i - u'_i h_{i-1} + \frac{1}{2!} u''_i h_{i-1}^2 - \frac{1}{3!} u_i^{(3)} h_{i-1}^3 + O(h_{i-1}^4) \ . \end{aligned}$$

Subtracting both equations gives for a central difference approximation

$$u'_i = \frac{u_{i+1} - u_{i-1}}{h_i + h_{i-1}} + \frac{h_i - h_{i-1}}{2} \, u''_i + O\left(\frac{h_i^3 + h_{i-1}^3}{h_i + h_{i-1}}\right) \ .$$

The first order term in h_i disappears for $h_i = h_{i-1}$ (uniform grid), so that the local **truncation error**[4] T is bounded by $O(h^2)|u''(x)|$, i.e. quadratically for a uniform grid with grid size h.

[1] **brevity** [brev.ɪ.ti]: Kürze [2] **expansion** [ikˈspæn.tʃᵊn]: hier: Entwicklung [3] **to diminish** [dɪˈmɪn.iʃ]: abnehmen, abklingen [4] **truncation error** [trʌŋˈkeɪ.ʃᵊn ˈer.ɚ]: Abschneidefehler

Thus, if the grid size is reduced by a factor of two, T is usually reduced by a factor of four, which we will refer to as *second order accurate*. For quasi-uniform grids, the truncation error is found to be bounded by $O(h^2)|u''| + O(h^2)|u^{(3)}|$, while for non-uniform grids only a linear dependence ($T < O(h)|u''(x)|$) between truncation error and grid size is obtained. A similar analysis can be done for the second order derivative. Starting from a Taylor series expansion at x_i and eliminating u'_i, the second order derivative is found as

$$u''_i = \frac{\left(\frac{u_{i+1} - u_i}{h_i}\right) - \left(\frac{u_i - u_{i-1}}{h_{i-1}}\right)}{\left(\frac{h_i + h_{i-1}}{2}\right)} + \left(\frac{h_i - h_{i-1}}{3}\right) u_i^{(3)} + O(h^2)\,. \tag{3.10}$$

Consequently, for the truncation error T we find similar results: Quadratic dependence of the truncation error with the grid size for uniform and quasi-uniform grids and linear dependence for non-uniform grids.

3.5 Solution of Linear Systems of Equations

Although a call to `solve` in `SGFramework` is **sufficient**[1] to solve a system of equations, some deeper knowledge about the solution process is of advantage. Especially three-dimensional device simulations lead to very high numbers of unknowns, which may slow down the solution process considerably if an improper solution algorithm is chosen.

For linear systems, a vector of unknowns $\boldsymbol{x} = (x_i)_{i=1}^n$ is sought such that

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \boldsymbol{b},$$

with system matrix $\boldsymbol{A} = \left(a_{ij}\right)_{i,j=1}^n$ and right hand side vector $\boldsymbol{b} = (b_i)_{i=1}^n$. Formally, the solution is given as $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$, but computing the matrix inverse explicitly is computationally too expensive and usually requires a lot of computer memory.

We will start with *direct solvers*, where the solution is found by a rather well known number of manipulations:

- *Gauss' method*. This method is one of the oldest methods. Matrix entries a_{ij} with $i > j$ in the lower left triangle of $\boldsymbol{A}$ are eliminated by means of linear combinations with rows $i-1, i-2, \dots, 1$. Finally, the solution vector is found using bottom-up insertion. The computational effort is $O(2n^3/3)$.

- *LU decomposition*. An invertible matrix $\boldsymbol{A}$ whose **principal minors**[2] are all different from zero can be decomposed into

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix} = \boldsymbol{LU},$$

[1] **sufficient** [səf.ɪʃ.ᵊnt]: ausreichend [2] **principal minors** [prɪnt.sɪ.pᵊl maɪ.nəʳ]: Hauptminoren

where $\boldsymbol{L}$ is a lower triangular matrix with ones in the main diagonal and $\boldsymbol{U}$ is a upper right triangle matrix. The solution $\boldsymbol{x}$ is found in two steps: First, the (triangular) system $\boldsymbol{Ly} = \boldsymbol{b}$ is solved for $\boldsymbol{y}$, then the triangular system $\boldsymbol{Ux} = \boldsymbol{y}$ is solved for $\boldsymbol{x}$.

The overall computational effort is again $O(2n^3/3)$ and is due to the decomposition process. Since the solution of $\boldsymbol{Ly} = \boldsymbol{b}$ and $\boldsymbol{Ux} = \boldsymbol{y}$ requires only $O(n^2)$ operations (triangular matrices!), the LU-decomposition is attractive in case several systems $\boldsymbol{Ax}_1 = \boldsymbol{b}_1$, $\boldsymbol{Ax}_2 = \boldsymbol{b}_2, \ldots, \boldsymbol{Ax}_k = \boldsymbol{b}_k$ with the same system matrix $\boldsymbol{A}$ need to be solved one after another, but cannot be solved simultaneously.

- *Cholesky decomposition*. A symmetric, positive definite matrix $\boldsymbol{A}$ can be decomposed into $\boldsymbol{A} = \boldsymbol{LL}^T$, where $\boldsymbol{L}$ is a lower left triangular matrix. The solution vector $\boldsymbol{x}$ is then found via a two-step process just as for the LU-decomposition.

  The computational effort is only $O(n^3/3)$, so it is asymptotically twice as fast as the Gauss method.

The above methods work well with dense matrices, but show poor performance for *sparse* matrices. In a sparse matrix, most entries are equal to zero and thus need not be stored, reducing memory consumption tremendously. We already encountered sparse matrices in (3.6) and (3.7) with entries unequal to zero around the main diagonal only. Even though direct methods for such matrices exist, *iterative solvers* are preferred when some properties like diagonal dominance or positive definiteness of $\boldsymbol{A}$ can be assured. These methods include Jacobi iteration, Gauss-Seidel iteration, Steepest Descent Method and the Conjugate Gradient algorithm.

We will not go into further detail here, however, the reader should be aware of the existence of iterative solvers, whose asymptotic run-time behavior is usually superior to that of direct solvers.

3.6 Solution of Nonlinear Systems of Equations

For the solution of non-linear equations the most widespread method is *Newton's method*. It is best explained in one dimension (Fig. 3.8): For a point x_k where a function f is evaluated to $f(x_k)$, a reasonable guess x_{k+1} for the location of a **root**[1] y (such that $f(y) = 0$) is the intersection of the tangent with the x-axis. The tangent can be obtained from a first order Taylor expansion of f in x_k:

$$f(x_{k+1}) \approx f(x_k) + f'(x_k)(x_{k+1} - x_k) .$$

Since x_{k+1} should be a root of f, we set $f(x_{k+1}) = 0$, so that

$$0 \overset{!}{=} f(x_k) + f'(x_k)(x_{k+1} - x_k) .$$

remains. After a rearrangement we find

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} .$$

Newton's method in algorithmic form is thus:

1. Start with an initial guess x_0 and $k = 0$.

[1] **root** [ru:t]: Wurzel, Ursprung, hier: Nullstelle

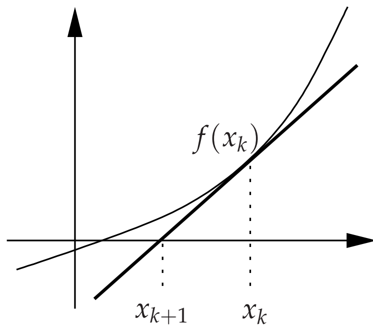


Figure 3.7: One-dimensional motivation for Newton's method.

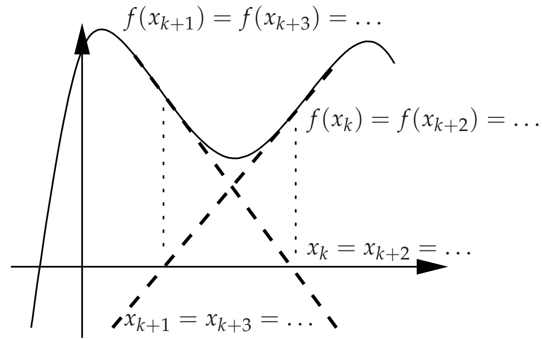


Figure 3.8: Newton's method may fail if the initial guess is too far away from a root.

k	x_k	$f(x_k)$	$\Delta x = -f(x_k)/f'(x_k)$
0	1.65	0.17186502845391	0.2967688297482
1	1.94676882974821	-0.04323343404123	-0.0498553585067
2	1.89691347124147	-0.00116331450812	-0.0014180411387
3	1.89549543010275	-9.52580161861505E-07	-1.1630679886549E-06
4	1.89549426703476	-6.41042774418565E-13	-7.8269248466524E-13
5	1.89549426703398	0.0	0.0

Table 3.1: Finding the root of $f(x) = \sin(x) - 0.5x$ using Newton's method.

2. Evaluate $f(x_k)$ and stop if $|f(x_k)| < \varepsilon$.

3. Evaluate $f'(x_k)$ to set up the tangent at $f(x_k)$ and find the intersection x_{k+1} with the abscissa. It is given as $x_{k+1} = x_k - f(x_k)/f'(x_k)$.

4. Increase k (i.e. take the new approximation as the new guess) and go to 2.

Instead of checking the absolute value of $f(x_k)$ in step 2 only, one can also check the magnitude of $\Delta x_k = -f(x_k)/f'(x_k)$ compared to x_k.

Unfortunately, Newton's method fails if $f'(x_k) = 0$, which means that the tangent is parallel to the x-axis and no intersection can be found. If the function $f(x)$ is continuously differentiable, its derivative does not vanish at α and it has a second derivative at α then the convergence is quadratic or faster, provided that the initial guess is in a neighborhood of α. This means that the number of valid digits doubles with each step, as long as we start with a good initial guess. What a "good" initial guess actually is, strongly depends on the given problem.

Although Newton's method appears to be a root-finding algorithm only, it can be used to solve arbitrary non-linear equations: In order to solve $f(x) = g(x)$ for x, rewrite the equation in implicit form as $f(x) - g(x) = 0$, set $\tilde{f}(x) := f(x) - g(x)$ and apply Newton's method to $\tilde{f}(x)$.

For example, consider the equation $\sin(x) = x/2$ and solve for x. With $f(x) = \sin(x) - 0.5x$ we get $f'(x) = \cos(x) - 0.5$. The numbers finally resulting from Newton's method with initial guess $x_0 = 1.65$ can be found in Tab. 3.1.

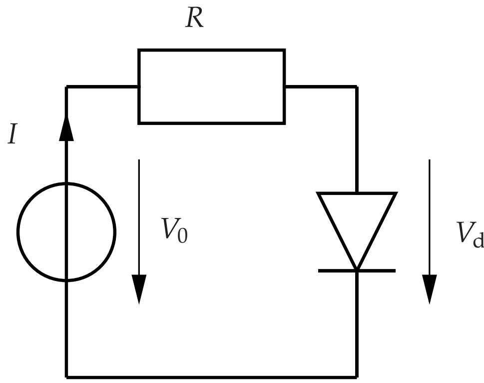


Figure 3.9: Example circuit for demonstration of Newton's method.

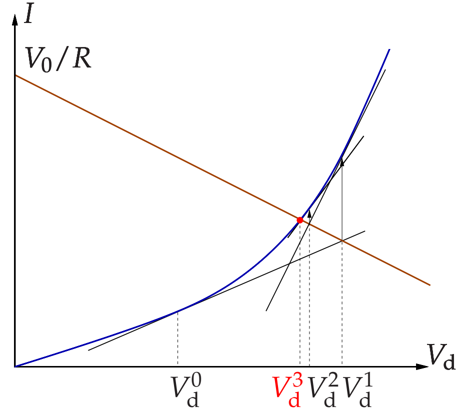


Figure 3.10: Newton's method for the solution of (3.11).

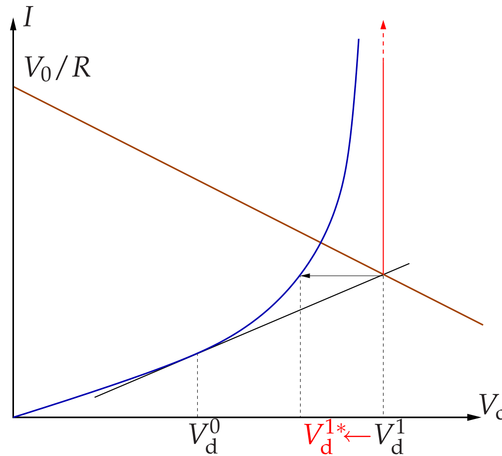


Figure 3.11: Newton's method may fail if the starting point is not chosen properly. The use of damping improves stability, but a good initial guess is still needed.

A more practical example is the determination of a current through a diode. Consider the circuit given in Fig. 3.9. The governing equation is

$$\frac{V_0 - V_\mathrm{d}}{R} = I_\mathrm{d}(V_\mathrm{d}) = I_\mathrm{s}\big(\exp(V/V_\mathrm{T}) - 1\big), \tag{3.11}$$

which is non-linear in V_d with known I_s, V_0 and V_T because of the exponential current characteristic of the diode. In Fig. 3.10 the solution using Newton's method is demonstrated: Starting with the initial guess V_d^0, the tangent to the diode characteristic is drawn. Then, the **intersection**[1] with the load line (i.e. the right hand side of (3.11)) is determined, resulting in a new approximation V_d^1. This procedure is repeated until one is sufficiently close to the solution.

Nevertheless, the above example can easily be modified so that Newton's method fails: Assume that the diode characteristic has a pole instead of exponential behavior. Then, a starting point that is too far away from the true solution does not converge, as is illustrated in Fig. 3.11. The numerical stability of Newton's method can be improved by *damping*: With a damping factor $d \in [0, 1]$ and notation as in the previous example, the next approximation V_d^{i+1} is now

[1] **intersection** [ɪn.təˈsek.ʃᵊn]: Schnittmenge, Schnittpunkt

computed as

$$V_{\mathrm{d}}^{i+1} = V_{\mathrm{d}}^{i} - d\frac{f(V_{\mathrm{d}}^{i})}{f'(V_{\mathrm{d}}^{i})} \,. \tag{3.12}$$

If d is chosen too small, the number of iterations increases unnecessarily, while d chosen too large may cause the solution process to fail. A clever trade-off is thus necessary in practice.

The issue of initial solutions that are sufficiently close to the true solution is especially a considerable problem for complicated systems. As for semiconductor device simulation, an initial guess uses built-in potentials (Chapter 2) and assumes charge neutrality (like for Ohmic contacts, Chapter 2).

A generalization to systems of n non-linear equations is rather straightforward, but mathematical analysis becomes much harder. In order to solve the coupled equation system

$$\begin{aligned} F_1(\boldsymbol{x}) &= 0 \\ F_2(\boldsymbol{x}) &= 0 \\ &\vdots \\ F_n(\boldsymbol{x}) &= 0 \end{aligned}$$

for a solution vector $\boldsymbol{x}$ of dimension n, introduce the vector-valued function

$$\boldsymbol{F}\colon \boldsymbol{x} \mapsto \begin{pmatrix} F_1(\boldsymbol{x}) \\ F_2(\boldsymbol{x}) \\ \vdots \\ F_n(\boldsymbol{x}) \end{pmatrix} .$$

In the algorithm given above, one then has to multiply with the inverse of the *Jacobian matrix* $\boldsymbol{J_F}(\boldsymbol{x}_k)$ of dimension $n \times n$ instead of dividing by $f'(x_k)$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{J}_{\boldsymbol{F}}^{-1}(\boldsymbol{x}_k)\boldsymbol{F}(\boldsymbol{x}_k) \,.$$

However, the inverse of the Jacobian is not computed explicitly, but the system

$$\boldsymbol{J_F}(\boldsymbol{x}_k)\boldsymbol{y} = \begin{pmatrix} \frac{\partial F_1}{\partial x_1}(\boldsymbol{x}_k) & \frac{\partial F_1}{\partial x_2}(\boldsymbol{x}_k) & \dots & \frac{\partial F_1}{\partial x_n}(\boldsymbol{x}_k) \\ \frac{\partial F_2}{\partial x_1}(\boldsymbol{x}_k) & \frac{\partial F_2}{\partial x_2}(\boldsymbol{x}_k) & \dots & \frac{\partial F_2}{\partial x_n}(\boldsymbol{x}_k) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(\boldsymbol{x}_k) & \frac{\partial F_n}{\partial x_2}(\boldsymbol{x}_k) & \dots & \frac{\partial F_n}{\partial x_n}(\boldsymbol{x}_k) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} F_1(\boldsymbol{x}_k) \\ F_2(\boldsymbol{x}_k) \\ \vdots \\ F_n(\boldsymbol{x}_k) \end{pmatrix} = \boldsymbol{F}(\boldsymbol{x}_k)$$

is solved using a linear equation solver. To sum up, the vector-valued Newton's method reads as follows:

1. Start with an initial guess $\boldsymbol{x}_0$ and $k = 0$.
2. Evaluate $\boldsymbol{F}(x_k)$ and stop if $\|\boldsymbol{F}(x_k)\| < \varepsilon$ (i.e. a sufficiently good approximation to the true solution is found).
3. Solve $\boldsymbol{J_F}(\boldsymbol{x}_k)\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x}_k)$ for $\boldsymbol{y}$.
4. Set $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \boldsymbol{y}$, increase k and go to 2.

To improve the convergence behavior, a damping factor $0 < d < 1$ is commonly used. The update is then $\boldsymbol{x}_k := \boldsymbol{x}_k - d\boldsymbol{y}$.

As a final example, consider the system

$$f_1(x,y) = 3x + 4y = 4\,,$$
$$f_2(x,y) = 7x + 3y = 9\,.$$

Define the implicit form

$$F_1(x,y) = 3x + 4y - 4 = 0\,,$$
$$F_2(x,y) = 7x + 3y - 9 = 0\,,$$

whose Jacobian is

$$\boldsymbol{J_F} = \begin{pmatrix} 3 & 4 \\ 7 & 3 \end{pmatrix}\,.$$

Starting with $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ results in

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = -\boldsymbol{J_F}^{-1} \begin{pmatrix} F_1(0,0) \\ F_2(0,0) \end{pmatrix} = -\begin{pmatrix} 3 & 4 \\ 7 & 3 \end{pmatrix}^{-1} \begin{pmatrix} -4 \\ -9 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 7 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 4 \\ 9 \end{pmatrix}.$$

For this system of linear equations the Jacobian is nothing but the initial system in matrix-vector form! Please note that an additional damping factor requires several solutions for the same system matrix, although the solution could be obtained in one step only.

Chapter 4

Two-Dimensional Simulation and Grids

In the previous chapter the numerical approximation of derivatives in one dimension was shown. This chapter goes one step further and applies these approximation techniques in two dimensions. First, the discretization and solution of the two-dimensional Laplace equation using finite differences will be discussed, after that, the flux-conserving box-integration method will be presented. The latter is also applicable to unstructured meshes, which will be the topic of Chapter 5.

4.1 Two-Dimensional Laplace Equation

The Laplace operator $\Delta = \nabla^2$ is a second order differential operator in the n-dimensional Euclidean space. It is defined as the divergence of the gradient ∇. Laplace's equation is the homogeneous form of the Poisson equation, i.e. without charge density ρ on the right hand side. In two dimensions it reads

$$\nabla^2 \psi(x,y) = \frac{\partial^2 \psi(x,y)}{\partial x^2} + \frac{\partial^2 \psi(x,y)}{\partial y^2} = 0 \tag{4.1}$$

in **Cartesian**[1] coordinates. The solutions of the Laplace equation are called *harmonic functions*. To determine the solution of (4.1) completely it is necessary to specify appropriate boundary conditions.

We begin our discussion of differencing schemes for elliptic equations by looking at the Poisson equation (4.1) in the unit square with equidistant grid spacing in the x- and y- directions. The standard central difference approximations for the second order derivatives with

$$\left(\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2}\right)_{x=i\Delta x} \approx \frac{\psi[i+1] - 2\psi[i] + \psi[i-1]}{(\Delta x)^2} \,,$$

are now applied in two dimensions thus giving

$$\nabla^2 \psi(x,y) \approx \frac{\psi[i+1,j] - 2\psi[i,j] + \psi[i-1,j]}{(\Delta x)^2} + \frac{\psi[i,j+1] - 2\psi[i,j] + \psi[i,j-1]}{(\Delta y)^2} \,. \tag{4.2}$$

[1] **Cartesian** [kɑːˈti.zi.ən]: kartesisch

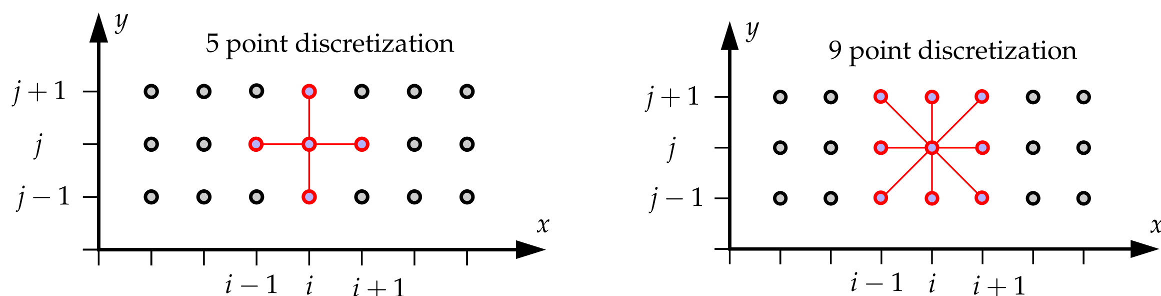


Figure 4.1: Five-point Laplacian (left) and nine-point Laplacian (right).

On an equidistant square grid ($\Delta y = \Delta x$) this becomes

$$\nabla^2\psi(x,y) \approx \frac{\psi[i+1,j] + \psi[i,j+1] - 4\psi[i,j] + \psi[i-1,j] + \psi[i,j-1]}{(\Delta x)^2}\ .$$

The difference operator on the right hand side is called *five-point* (discrete) **Laplacian**[1] (aka *five-point* **stencil**[2], because the point operation pattern is applied to all points of the grid) with second order accuracy. Another possible approximation of the Laplace equation is the fourth order accurate *nine-point* Laplacian. For further information the reader is referred to the book of Strikwerda [**?**].

```
const NX = 21,      NY = 21;      // number of mesh points
const LX = NX - 1, LY = NY - 1; // index of last mesh point
const DX = 1.0;                   // mesh spacing
var    psi[NX,NY];

// Laplace's equation (only inner points)
equ psi[i=1..LX-1,j=1..LY-1]  ->
              {psi[i-1,j] + psi[i,j-1] - 4*psi[i,j] +
               psi[i+1,j] + psi[i,j+1]} / sq(DX) = 0.0;
begin main
  assign psi[i=all,j=all] =  0.0; // Dirichlet boundary condition
  assign psi[i=0,  j=all] = 10.0; // Dirichlet boundary condition
  solve;
  write;
end
```

source_code/laplace.sg

In `laplace.sg` the potential of `psi[i=0, j=all]` is set equal to `10.0`, which constitutes a Dirichlet boundary condition and overwrites the previous assignment `psi[i=all, j=all] = 0.0`. The command `solve` lets SGFRAMEWORK solve the system of equations given by the `equ` command. The resulting potential can be seen in Fig. 4.2. Due to the discontinuous boundary conditions given, singularities of the electrical field (which is the negative gradient of the potential) occur. Such singularities cannot simply be swept under the carpet: They may influence the global quality of approximation through a reduction of the convergence rate.

Motivated from the singularities that show up with Dirichlet boundary conditions, one may think that large gradients are in general a consequence of the source/sink term g and of discontinuities in the boundary conditions only. However, this is not true. Even if the source/sink

[1] **Laplacian** [laplasıæn]: Laplace-Operator [2] **stencil** [stent.sᵊl]: Vervielfältigungsmatrix

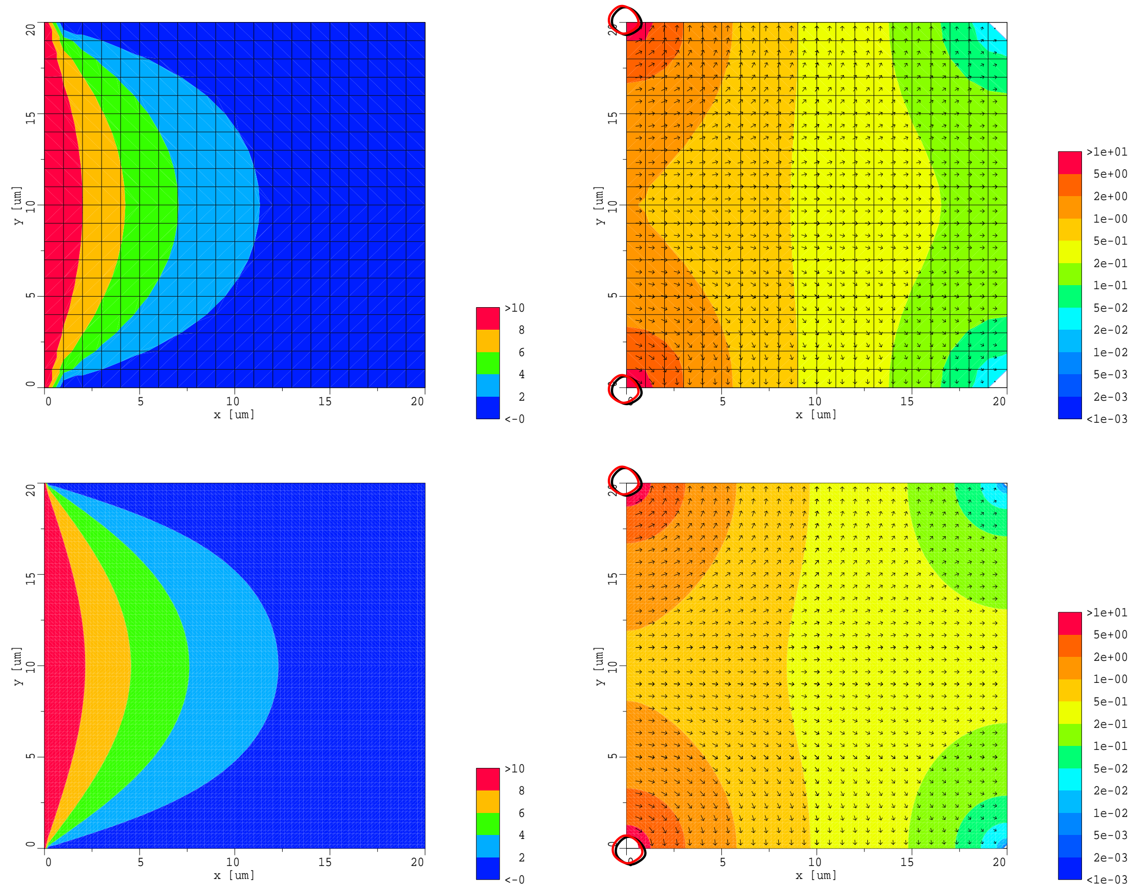


Figure 4.2: Potential (left) and electric field in log-scale (right) with 21 × 21 points (top) and 101 × 101 points (bottom) with Dirichlet boundary conditions. The singularities occurring at the edges are marked with circles.

term g were perfectly smooth and no discontinuities of the boundary conditions occured, the domain Ω may consist of **reentrant corners**[1], that will always provoke singularities in the solution.

To come back to our simple rectangular domain, replacing the Dirichlet boundary condition with a homogeneous Neumann boundary condition (i.e. zero flux) at the boundary section `j=LY` results in $\psi[i, LY] = \psi[i, LY-1]$ for all i. As a consequence, one row in the array is wasted, since two rows carry the same information.

```
    equ psi[i=1..LX-1,j=LY] -> {psi[i,j] - psi[i,j-1]} / sq(DX) = 0.0;
```

source_code/neumann_boundary1.sg

One can get rid of this waste of memory by using a little trick: We image the existence of another row at `j=LY+1`, so that the boundary condition at `j=LY` is $(\psi[i, LY+1] - \psi[i, LY])/DX = 0.0$;, which simply states that $\psi[i, LY+1] = \psi[i, LY]$. Now we consider the five-point stencil

```
    equ psi[i=1..LX-1,j=1..LY-1]  -> {psi[i-1,j] + psi[i,j-1] - 4*psi[i,j] +
                                      psi[i+1,j] + psi[i,j+1]} / sq(DX) = 0.0;
```

source_code/neumann_boundary2.sg

[1] **reentrant corners** [riː.en.trənt]: einspringende Ecken

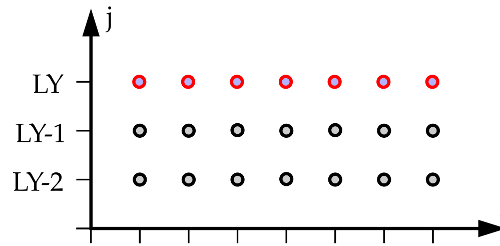


Figure 4.3: One row seems to be wasted when using Neumann boundary condition.

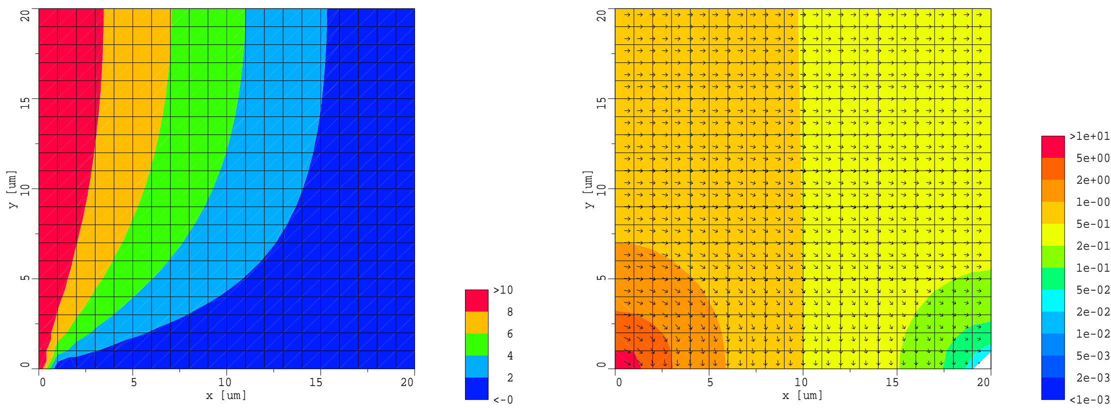


Figure 4.4: Potential (left) and electric field in log-scale (right) with 21 $\times$ 21 points including Dirichlet and Neumann boundary conditions.

at $j = LY$ and replace all occurrences of $\psi[i, LY + 1]$ with $\psi[i, LY]$, as required by the Neumann boundary condition. After collection of terms, the new equation for the last row is

```
1    equ psi[i=1..LX-1,j=LY] -> {psi[i-1,j] - 3*psi[i,j] + psi[i+1,j] + psi[i,j-1]} /
        sq(DX) = 0.0;
```

source_code/neumann_boundary3.sg

and has to be implemented into the `laplace.sg` SGFRAMEWORK file.

The resulting potential can be seen in Fig. 4.4. At the Neumann boundary, the solution is very smooth, the only discontinuity that shows up is due to the discontinuous Dirichlet boundary conditions. Note that the field in Fig. 4.4 resembles the field in the lower half of the problem space of Fig. 4.2. This is not a coincidence, it is a general property of homegeneous Neumann boundary conditions, which is why they are also called mirror boundary conditions (cf. method of images in electrostatics).

4.2 Box Integration Method

The **box integration method** (aka. *finite volume method*, suitable also for unstructured meshes) is a subset of finite difference methods, and is very important in 2D semiconductor device simulations. Box integration provides a convenient method of discretizing a large class of PDEs on both regular and irregular meshes by constructing a box around each node in such a way that

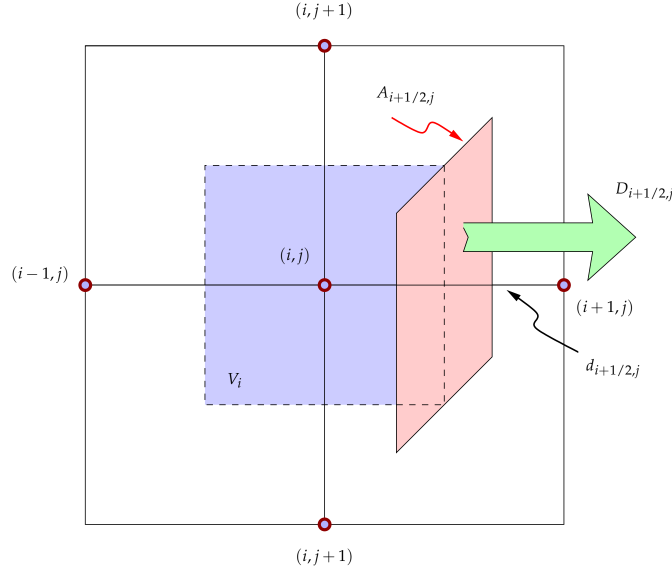


Figure 4.5: Approximated closed-loop path integral is made up of its four 'constant D' edge contributions.

the boxes cover the whole simulation domain. The differential equations are then individually integrated over each of the subdomains.

4.2.1 Example: The Poisson Equation

In order to demonstrate the box method, we integrate the Poisson equation $\nabla \cdot \boldsymbol{D} = \rho$ over a small region (box) $\mathcal{V}$ with volume V_i and obtain

$$\int_{\mathcal{V}} \nabla \cdot \boldsymbol{D} \, \mathrm{d}x = \int_{\mathcal{V}} \rho \, \mathrm{d}x \;\approx\; \rho_i \, V_i \, ,$$

where ρ_i is the charge density evaluated at some representative point x_i inside $\mathcal{V}$. We will discretize the left-hand side using *Gauss' integral theorem*, which states that

$$\int_{\mathcal{V}} \nabla \cdot \boldsymbol{D} \, \mathrm{d}x = \int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \, ,$$

where $\boldsymbol{n}$ is the outward pointing local normal vector of the enclosing surface. In two dimensions, $\partial\mathcal{V}$ is a closed-loop path.

When discretizing a closed-loop path integral, the contributions along the path are approximated and accumulated. For the example in Fig. 4.5 this means that for instance $D_{i+1/2,j}$ is assumed constant along the edge $(i+1/2, j-1/2)$ and $(i+1/2, j+1/2)$. The other three sections of the path are approximated in a similar fashion. As we make the box volume smaller, the approximation error becomes smaller. Hence, the approximation of the closed-loop path integral gives

$$\int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \approx D_{i+1/2,j} A_{i+1/2,j} + D_{i-1/2,j} A_{i-1/2,j} + D_{i,j+1/2} A_{i,j+1/2} + D_{i,j-1/2} A_{i,j-1/2}$$

with the approximation of the fluxes

$$\begin{aligned}
D_{(i,j)\to(l,m)} &= \boldsymbol{D}\Big|_{x=(i+l)/2, y=(j+m)/2} \cdot \boldsymbol{e}_{(i,j)\to(l,m)} \\
&= (\varepsilon \boldsymbol{E})\Big|_{x=(i+l)/2, y=(j+m)/2} \cdot \boldsymbol{e}_{(i,j)\to(l,m)} \\
&= (-\varepsilon \nabla \psi)\Big|_{x=(i+l)/2, y=(j+m)/2} \cdot \boldsymbol{e}_{(i,j)\to(l,m)} \\
&\approx -\frac{\varepsilon_{i,j} + \varepsilon_{l,m}}{2} \frac{\psi_{l,m} - \psi_{i,j}}{d_{(i,j)\to(l,m)}} .
\end{aligned}$$

Assuming an equidistant mesh ($\Delta y = \text{const}$, $\Delta x = \text{const}$),

$$A_{i+1/2,j} = A_{i-1/2,k} = \Delta y w, A_{i,j+1/2} = A_{i,j-1/2} = \Delta x w, V_i = \Delta x \Delta y w ,$$

with w as depth of the simulation domain and constant permittivity we get

$$\begin{aligned}
\int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \approx &- \varepsilon \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x} w \Delta y - \varepsilon \frac{\psi_{i-1,j} - \psi_{i,j}}{\Delta x} w \Delta y \\
&- \varepsilon \frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta y} w \Delta x - \varepsilon \frac{\psi_{i,j-1} - \psi_{i,j}}{\Delta y} w \Delta x \\
= &- \varepsilon \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x} w \Delta y \\
&- \varepsilon \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta y} w \Delta x \\
= &\, \rho_i \, \Delta x \, \Delta y \, w .
\end{aligned}$$

Dividing by V_i we finally obtain

$$-\varepsilon \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{(\Delta x)^2} - \varepsilon \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{(\Delta y)^2} = \rho_i$$

which is exactly the same result as obtained by finite differences (4.2). Assuming for example an inhomogeneous permittivity $\varepsilon = \varepsilon(y)$ one has to modify the previous equations to

$$\begin{aligned}
\int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \approx &- \varepsilon_j \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x} w \Delta y \\
&- \frac{\varepsilon_j + \varepsilon_{j+1}}{2} \frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta y} w \Delta x \\
&- \frac{\varepsilon_j + \varepsilon_{j-1}}{2} \frac{\psi_{i,j-1} - \psi_{i,j}}{\Delta y} w \Delta x .
\end{aligned}$$

4.2.2 Example: Extraction of Capacitances

When working with Dirichlet boundary conditions for the box integration method, two implementation models are appropriate. Either a substitute equation, like $\psi_{LY} = V_c$ is implemented or the Dirichlet boundary conditions are explicitly inserted into the `.sg` file in place of ψ_{LY}. While the first possibility wastes variables, the latter requires more careful index-checking.

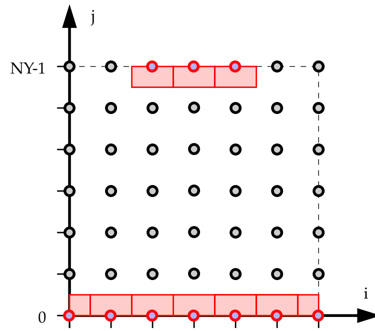


Figure 4.6: Dirichlet Boundary Conditions

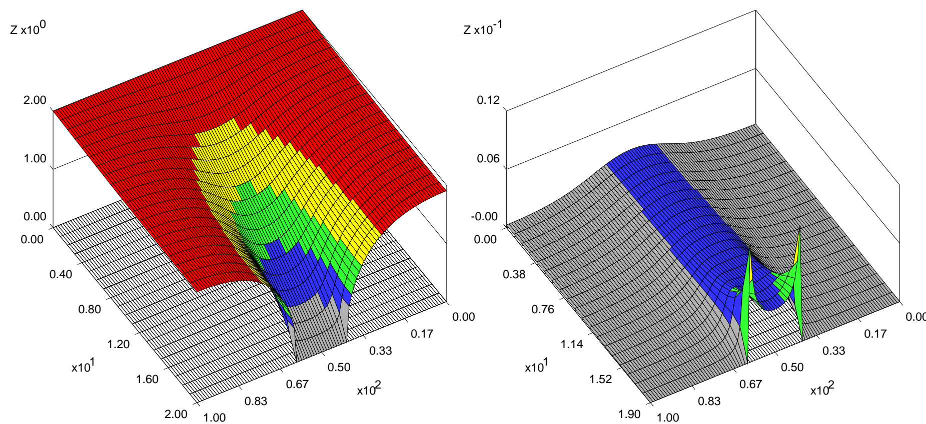


Figure 4.7: Potential ψ (left) and displacement D_y (right)

Neumann boundary conditions for zero out-flux are dealt with very simply by omitting corresponding contributions within the surface integral. For a box centered at point (r, s) with a homogeneous Neumann boundary condition along the edge at $(r, s+1/2)$, we have

$$\int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \approx D_{r+1/2,s} A_{r+1/2,s} + D_{r-1/2,s} A_{r-1/2,s} + \underbrace{D_{r,s+1/2} A_{r,s+1/2}}_{=0} + D_{r,s-1/2} A_{r,s-1/2} \,.$$

In contrast to that, Neumann boundary conditions with a non-zero boundary flux D_c are added explicitly

$$\int_{\partial\mathcal{V}} \boldsymbol{D} \cdot \boldsymbol{n} \, \mathrm{d}x \approx D_{r+1/2,s} A_{r+1/2,s} + D_{r-1/2,s} A_{r-1/2,s} + D_\mathrm{c} A_{r,s+1/2} + D_{r,s-1/2} A_{r,s-1/2} \,.$$

`extraction_capacitances.sg` shows a program to simulate a capacitor with the results shown in Fig. 4.7 and Fig. 4.8.

```
1  const NX = 101, NY = 21, DX = 1.0e-9, DY = DX;
2  const Vc = 2,     A = DX;
3  const X1 = 40,   X2 = 60;
4
5  const RHO   = 1e-14;
6  const EPSr  = 3.9;         // relative permittivity region 1
7  const EPSo  = 8.854e-12;   // permittivity of vacuum (approximately)
```

```
8
9  var psi[NX,NY], Ex[NX-1,NY], Ey[NX,NY-1], Dx[NX-1,NY], Dy[NX,NY-1];
10 var E[NX-1,NY-1], D[NX-1,NY-1], C[2], Qs[NX,NY];
11
12 unknown psi[0..NX-1,0..NY-1];
13 known   psi[X1..X2,NY-1];
14 known   psi[all,0];
15
16 // Poisson Equation for the inner grid points
17 equ psi[i=1..NX-2,j=1..NY-2] ->
18    {                 psi[i,j-1] +
19     psi[i-1,j] - 4.0*psi[i, j ] + psi[i+1,j]
20               +      psi[i,j+1]              } / sq(DX) = RHO / ( EPSo * EPSr);
21
22 // Neumann boundary condition
23 equ psi[i=0,     j=0..NY-2]     -> psi[i,j] - psi[i+1,j] = 0;
24 equ psi[i=NX-1,  j=0..NY-2]     -> psi[i,j] - psi[i-1,j] = 0;
25 equ psi[i=0..X1-1,     j=NY-1] -> psi[i,j] - psi[i,j-1] = 0;
26 equ psi[i=X2+1..NX-1,  j=NY-1] -> psi[i,j] - psi[i,j-1] = 0;
27
28 begin main
29
30   // Contact voltages, Dirichlet boundary condition
31   assign psi[i=X1..X2, j=NY-1] = 0;
32   assign psi[i=all,    j=0]    = Vc;
33
34   solve;
35
36   // Find E and D from the potential psi using right sided differences
37   assign Ex[i=all,j=all] = - (psi[i+1,j] - psi[i,j])/DX ;  // Ex = - dpsi/DX
38   assign Ey[i=all,j=all] = - (psi[i,j+1] - psi[i,j])/DX ;  // Ey = - dpsi/dy
39   assign E [i=all,j=all] = sqrt(sq(Ex[i,j]) + sq(Ey[i,j]));
40
41   assign Dx[i=all,j=all] = EPSr * EPSo * Ex[i,j];
42   assign Dy[i=all,j=all] = EPSr * EPSo * Ey[i,j];
43   assign D [i=all,j=all] = sqrt(sq(Dx[i,j]) + sq(Dy[i,j]));
44
45   // sum up charges at Vc-electrode
46   assign Qs[i=0,       j=0]       =                                A/2 * Dy[i,j];
47   assign Qs[i=1..NX-2,j=0]        = Qs[i-1,j]                    + A   * Dy[i,j];
48   assign Qs[i=NX-1,    j=0]       = Qs[i-1,j]                    + A/2 * Dy[i,j];
49
50   // sum up charges at ground electrode (do not forget Dx-components!)
51   assign Qs[i=X1,         j=NY-1] =             A/2 * Dx[i-1,j] + A   * Dy[i,j-1];
52   assign Qs[i=X1+1..X2-1,j=NY-1] = Qs[i-1,j]                    + A   * Dy[i,j-1];
53   assign Qs[i=X2,         j=NY-1] = Qs[i-1,j] + A/2 * Dx[i,  j] + A   * Dy[i,j-1];
54
55   // compute capacitances from the auxiliary charge
56   assign C[i=0] = Qs[NX-1,0]   / Vc;
57   assign C[i=1] = Qs[X2,NY-1] / (- Vc);
58
59   write;
60
61 end
```

source_code/extraction_capacitances.sg

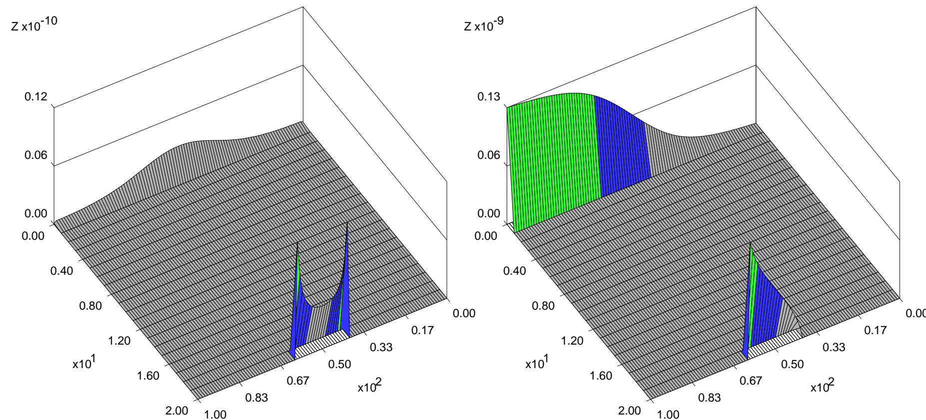


Figure 4.8: Real charge per node (left) and auxiliary charge Q_s (right)

The capacitance $C = Q/V$ is calculated via the charge on the contact surface, which is

$$Q = \int_{\mathcal{V}_{\text{contact}}} \rho \,\mathrm{d}V = \int_{\partial\mathcal{V}_{\text{contact}}} \boldsymbol{D} \cdot \boldsymbol{n} \,\mathrm{d}x = \sum_{\text{all contact points}} D_y[i]A[i] \;.$$

Because there is no direct support for integration of a function in SGFRAMEWORK, a workaround in the form of a sum over all contact points has to be assigned. Starting at `i=0,j=0` the first element is calculated (be aware that for the first and the last cell only a half of each edge belong to the contact). The charge is then added up from the first to the last cell. The implementation trick in SGFRAMEWORK can be understood when looking at Fig. 4.6 and is requires some extra attention when dealing with the last row `j=NY-1`. The code in `extraction_capacitances2.sg` displays the trick and Fig. 4.8 shows the result on the right hand side. Starting at $x = 0$, the (auxiliary-) charge is growing along the positive x-axis until its peak value at the end of the contact where it represents the approximated value of the integral. Again, the accuracy only depends on the grid spacing.

```
1  assign Qs[i=0,        j=0]      =                A/2 * Dy[i,j];
2  assign Qs[i=1..NX-2,j=0]        = Qs[i-1,j] + A    * Dy[i,j];
3  assign Qs[i=NX-1,     j=0]      = Qs[i-1,j] + A/2 * Dy[i,j];
4  assign C[i=0]                   = Qs[NX-1,0] / Vc;
```

source_code/extraction_capacitances2.sg

Chapter 5

Tessellation of Unstructured Meshes

If we need a high accuracy of the quantity of interest in a certain region of the device only, structured grids like Manhattan geometries[1]) or ortho-product grids require a refinement outside the simulation domain as well. For example, for a two-dimensional simulation of a MOSFET with 100 grid points in both spatial directions, we have to deal with 10 000 unknowns. Typically, we require a high resolution of the channel, which is only a very small part of the simulation domain. As can be seen in Fig. 5.1, a refinement of the channel (critical region) would also result in a refinement along strips parallel to the x- and y-axis. This way, many additional unknowns are introduced outside the channel (critical region) which are not needed.

By means of the box integration method, unstructured meshes can be used for solving partial differential equations. This allows meshes with very fine grid spacing in the area of interest (like the aforementioned channel of a MOSFET), while maintaining a larger grid spacing in regions where the quantities are known in advance to have small variations (like in the deep bulk region of a MOSFET).

This chapter deals with the generation and handling of such unstructured meshes and describes both the resulting benefits and the **subtleties**[2].

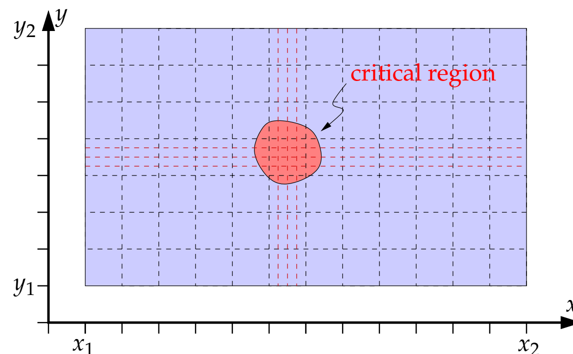


Figure 5.1: Although only nine additional points are required in the critical region, the total number of grid points N_{Points} increases from 104 to 176.

[1] The upright projection of a transistor may look like the street map of Manhattan

[2] **subtlety** [sʌt.l.ti]: Schwierigkeit, Raffinesse

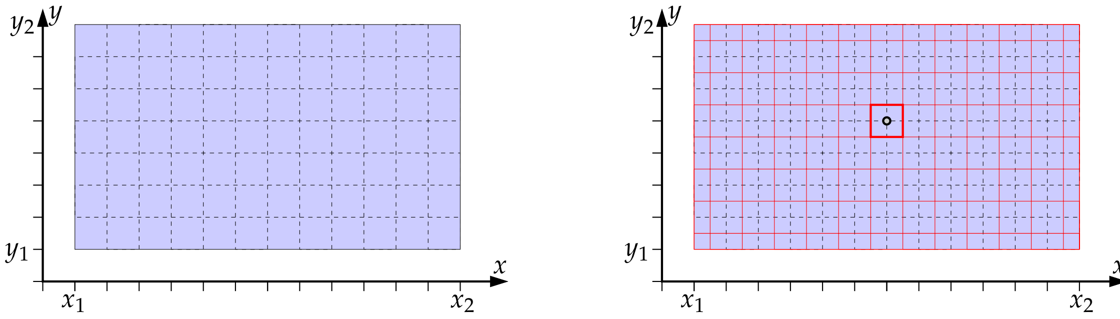


Figure 5.2: Ortho-product grid (left) and its corresponding Voronoi tessellation (right).

5.1 Voronoi Tessellation

When the points are located arbitrarily in space rather than on a regular grid, we need a proper generalization of the ortho-product grid, called Voronoi **tessellation**[1]: Consider a finite set of points $\mathcal{D} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots \boldsymbol{r}_n\}$ in a subdomain Ω of $\mathbb{R}^n$. A Voronoi region Ω_i is the set of all points of Ω that are closer to $\boldsymbol{r}_i$ than to any other point of $\mathcal{D}$:

$$\Omega_i = \{\boldsymbol{r} \in \Omega \,|\, \|\boldsymbol{r} - \boldsymbol{r}_i\| < \|\boldsymbol{r} - \boldsymbol{r}_j\| \; \forall j \neq i\} \tag{5.1}$$

The resulting Voronoi regions Ω_i form a tessellation of $\Omega = \bigcup_i \Omega_i$, given in Fig. 5.2.

The construction of a Voronoi tessellation works as follows: The edges of the boxes are obtained by drawing the **perpendicular bisecting**[2] segments of edges that connect the mesh nodes, as shown in Fig. 5.3. The bisectors are only allowed to extend until they cut another bisector.

5.2 Triangular Delaunay Meshes

It can be shown that every Voronoi tessellation has a dual triangular mesh called **Delaunay**[3] mesh (can be seen in the lower right of Fig. 5.3). Since it is usually easier to construct a triangular mesh than to construct the Voronoi tessellation directly, we will use this duality to construct a Voronoi tessellation from a triangulation. Unfortunately, not every triangulation is the dual of a Voronoi tessellation, but it is possible to modify triangulations (meshes) such that they become duals of Voronoi tessellations, i.e. Delaunay meshes.

Let us have a look at the construction of a Delaunay mesh out of a point cloud. In order to obtain a Delaunay mesh, we need a criterion that allows us to determine whether a triangulation is a Delaunay mesh (Fig. 5.4):

- *Empty Sphere Criterion*: The open discs (balls) circumscribing the triangles (or tetrahedra) must not contain any other mesh point (Fig. 5.5).

It turns out that the local property of fulfilling the Empty Sphere Criterion directly extends to a global property:

---

[1] **tessellation** [tes.ᵊlˈeɪ.ʃᵊn]: Mosaik [2] **to perpendicular bisect** [pɛː.pᵊnˈdɪk.jʊ.lər baɪˈsekt]: in zwei gleich große Teile teilen [3] **Delaunay** [deloʊneɪ]: Delaunay

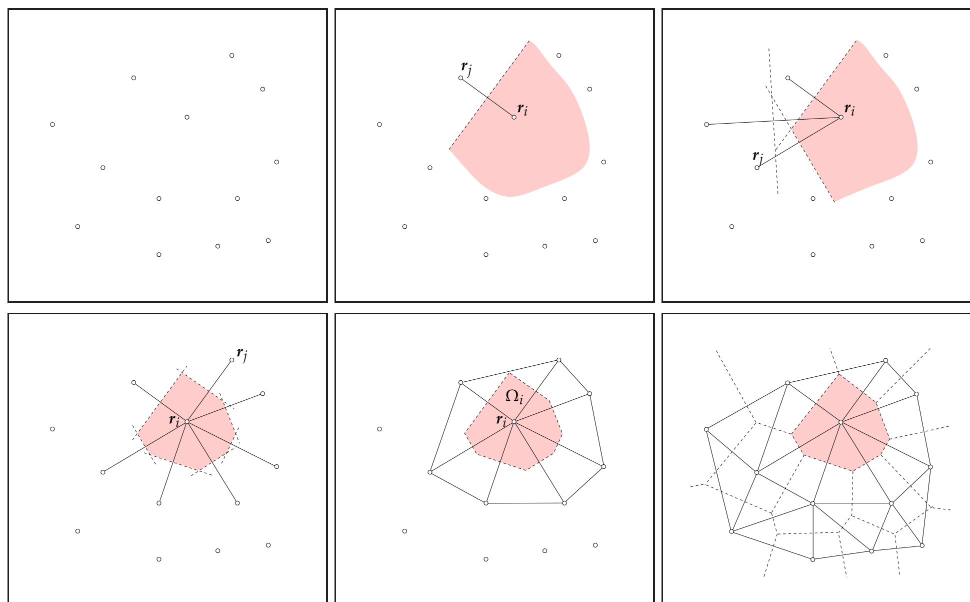


Figure 5.3: A set of 13 grid points and the associated Voronoi regions (bounded by the dashed lines).

Lemma. *If for every pair of adjacent triangles (or tetrahedrons) in a given mesh the empty sphere criterion holds, then this criterion holds globally and the mesh is a Delaunay triangulation.*

With the Empty Sphere Criterion at hand, Delaunay meshes can be constructed in many ways. One of them is the *incremental method*, where one starts with a rectangle (or any other polyhedron) made up from a few points within the simulation domain and then adds point by point (Fig. 5.6) to obtain a full mesh. Alternatively, one might take an arbitrary triangulation and swap the diagonals that violate the empty sphere criterion; this is called the *swapping method*. Moreover, as indicated in the introduction to this chapter, it is desirable that the mesh reflects the regions of interest in the device: The mesh density for example should be a function of the gradient (local change rate) of the quantities. This poses a chicken-and-egg problem: The mesh should be instantiated prior to the determination of the quantities (the simulation process it-

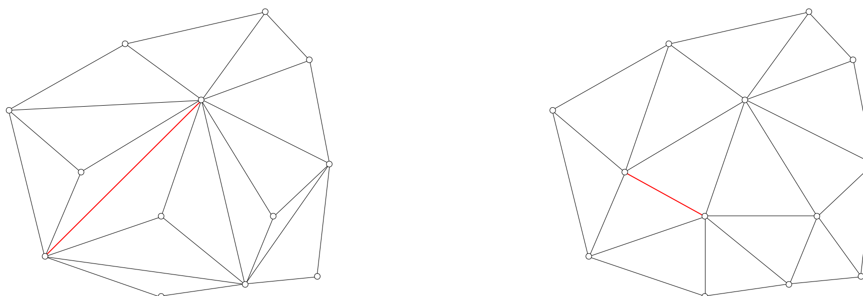


Figure 5.4: Example of two triangular grids constructed from the same point-cloud, where the right one is the dual of the Voronoi tessellation.

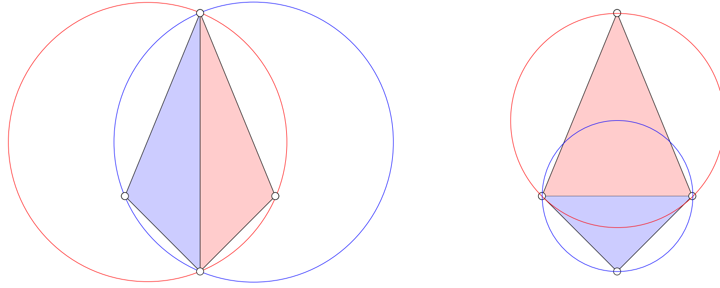


Figure 5.5: Example of two triangular meshes constructed from the same deltoid. By the swapping the diagonals in the left mesh, the mesh on the right is obtained, which is the dual of the Voronoi tessellation.

self), but the mesh generation process needs the quantities to construct a suitable mesh. This is actually the idea of adaptive schemes, where the solution is first computed on a coarse mesh, then this mesh is refined according to the behavior of the coarse solution. This procedure is then repeated on the refined mesh again, until a sufficiently good approximation to the true solution is found.

5.3 Skeleton Mesh

From the previous section it should have become clear that additional topological information has to be stored for unstructured grids. For example, the location of points cannot be **recovered**[1] by their array index anymore, as well as some kind of connectivity information must be available. In this section, the data structures used to **accomplish**[2] this are discussed.

Referring to the example of the Poisson equation in section 4.2.1, we write

$$\sum_{j \in \mathcal{N}_i} D_{i,j} A_{i,j} = - \sum_{j \in \mathcal{N}_i} \varepsilon_{i,j} \frac{\psi_j - \psi_i}{d_{i,j}} A_{i,j} = \rho_i V_i \,,$$

where $\mathcal{N}_i$ denotes the set of all nodes that are neighbors of node i. The integral is discretized by the box method using the Voronoi tessellation. In Fig. 5.8 the solid lines are the mesh edges while the dashed lines are the edges of the integration box. V_i is the volume of the box surrounding the mesh point i, $d_{i,j}$ is the length of the edge that connects i and its neighbor-point j, and $A_{i,j}$ is the length of the corresponding integration area, i.e. the area of that surface element of Ω_i that interfaces the box Ω_j at point j.

To foster the understanding of unstructured neighborhood information, a 3 times 3 rectangular example geometry is presented in Fig. 5.8. Taking a closer look at Tab. 5.1, one can verify that the only box having a volume $V_i = 1$ is box number 4 because of its location in the interior, while the corner boxes (0, 2, 6, 8) and the edge boxes (1, 3, 5, 7) have a volume of 1/4 and 1/2 respectively. The column names x_i and y_i represent the indices of the box. $A_{i,j}$ and $d_{i,j}$ have already been explained.

The information given in Tab. 5.1 is sufficient for any geometry in any dimension. It is implemented in SGFRAMEWORK using a so-called *skeleton file*. The domain specified in `sample_geometry.sk` is shown in Fig. 5.9.

[1] **to recover** [rɪˈkʌv.ɚ]: zurückgewinnen, wiedererlangen [2] **to accomplish sth.** [əˈkɑːm.plɪʃ] : etwas erreichen, etwas vollbringen

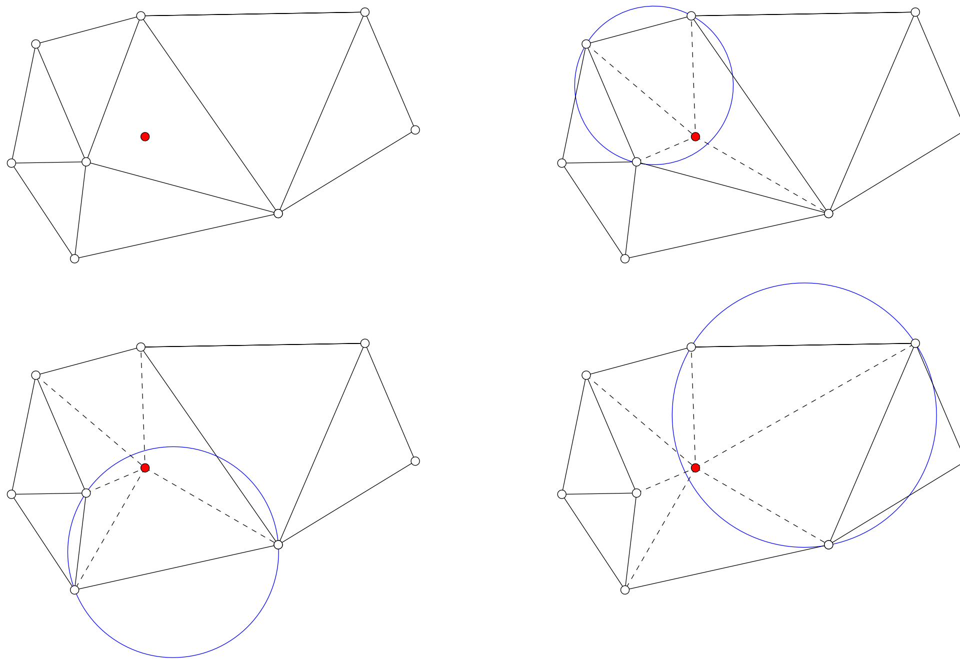


Figure 5.6: Incremental method (starting from a Delaunay mesh). First, the newly inserted point is connected with the nodes of the triangle it is located in. Then, the empty sphere criterion is checked for all adjacent triangles. If the criterion is violated for a neighbor, the interfacing edge is removed and the newly inserted point is connected with the opposite corner node of the neighbor.

```
// Mesh file for simple square mesh
// sample_geometry.sk (will be converted to sample_geometry.msh)

const dx = 1.0, W = 20.0;

point pA = (0.0,   W), pB = (  W,   W);
point pC = (  W, 0.0), pD = (0.0, 0.0);

edge eAB = METAL1 [pA,pB] (dx, 0.0);
edge eBC = METAL2 [pB,pC] (dx, 0.0);
edge eDC = METAL2 [pD,pC] (dx, 0.0);
edge eAD = METAL2 [pA,pD] (dx, 0.0);

// remove keyword 'rectangles' to get a triangular mesh
region rABCD = AIR {eAD,eDC,eBC,eAB} rectangles;

coordinates x, y;
```

source_code/sample_geometry.sk

In the first section geometrical quantities such as the device width and depth are defined as constants. Next the skeleton points are defined by two coordinates enclosed in *parentheses* each. Each point is given a name that is used for further reference. Then edges are defined using the previously set points. They are described by their two endpoints enclosed in brackets. The order of the edges (running **counterclockwise**[1]) is important later when defining regions. If the edges shall constitute a rectangular region, opposite parallel edges must point in the same

[1] **counterclockwise** [kaʊn.tə̈ˈklok.waɪz]: gegen den Uhrzeigersinn

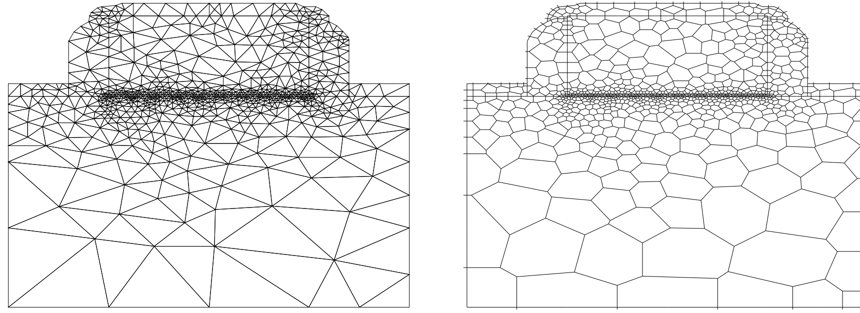


Figure 5.7: Unstructured mesh of a MOS transistor; triangular mesh (left) and the corresponding Voronoi regions (right).

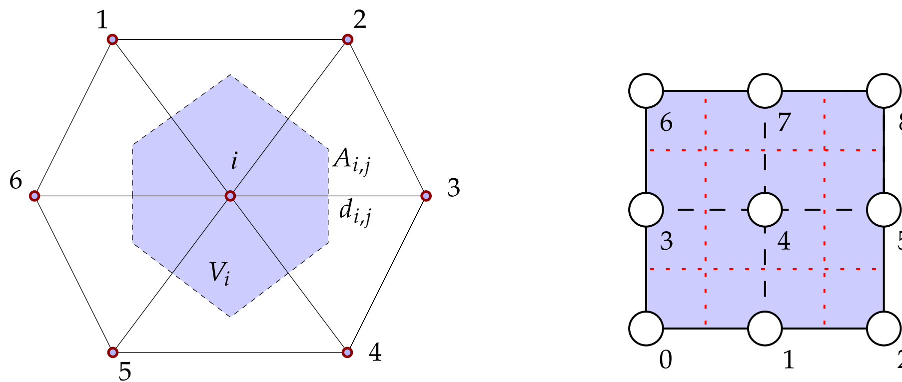


Figure 5.8: Left: The Voronoi box V_i of mesh point i is surrounded by six adjacent mesh points. For the discretization the knowledge of $A_{i,j}$ and $d_{i,j}$ is required. Right: in an ortho-grid geometry this information is implicitly available.

direction. Finally, the regions are defined in the last section of the mesh skeleton by a list of three or more edges. The list is enclosed in *braces*. The skeleton file exports all the user-defined constants plus three additional ones to the `.sg` file:

`NODES`	...	number of nodes (points)
`EDGES`	...	number of edges
`ELEMENTS`	...	number of elements (regions; triangular and rectangular)

In `sample_geometry.sk` the coordinates are identified as (x, y). The arrays $x[j]$ and $y[j]$ are automatically passed to `mesh_example1.sg` from the generated mesh file `sample_geometry.msh`. This is important, because otherwise there would be no way of finding the positions of the nodes. The labels representing a list of nodes are imported as well. The results are given in Fig. 5.10.

```
1  mesh "sample_geometry.msh";
2
3  var      psi[NODES]; // one-dimensional
4  unknown  psi[all];
5  known    psi[METAL1], psi[METAL2];
6
7  equ psi[i=AIR] -> nsum(i,j,all,
8                         ((psi[i]-psi[node(i,j)]) / elen(i,j)} * ilen(i,j))
9                        ) = 0.0;
```

Boxnumber	x_i	y_i	V_i
0	0	0	1/4
1	1	0	1/2
2	2	0	1/4
3	0	1	1/2
4	1	1	1
5	2	1	1/2
6	0	2	1/4
7	1	2	1/2
8	2	2	1/4

Connectionnumber	i	j	$A_{i,j}$	$d_{i,j}$
0	0	1	1/2	1
1	0	3	1/2	1
2	1	2	1/2	1
3	1	4	1	1
4	2	5	1/2	1
5	3	4	1	1
6	3	6	1/2	1
7	4	7	1	1
8	4	5	1	1
9	5	8	1/2	1

Table 5.1: Location and volume of Voronoi boxes (left) and connection information (right).

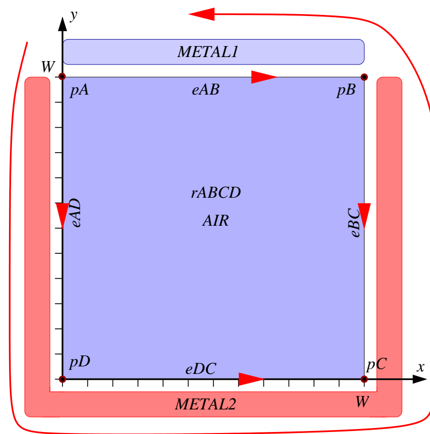


Figure 5.9: Boundary conditions have to be included into a skeleton file.

```
10
11 begin main
12  assign psi[i=METAL1] = 10.0;
13  assign psi[i=METAL2] =  0.0;
14  solve;  write;
15 end
```

source_code/mesh-example1.sg

5.4 Auxiliary Functions

Two helper functions, `nsum` and `lsum`, are described in the following as they are used in `mesh-example1.sg`. They establish a convenient means of handling neighborhood information as needed by the box integration method. The node summation function `nsum` sums over all neighbors of a given point and is declared as

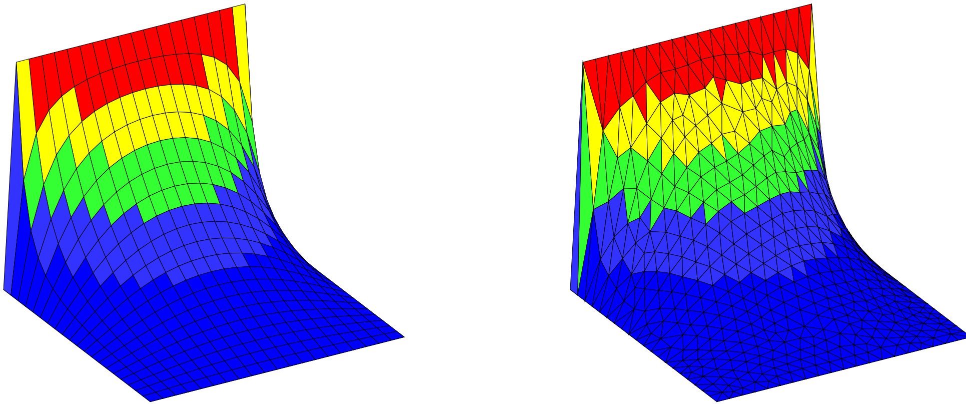


Figure 5.10: Rectangular Grid (left) and Triangular Mesh (right).

`nsum(` *nodeIndex, neighborIdentifier, label, expr* `)`

nodeIndex	...	current node
neighborIdentifier	...	identifier that will run through all neighbors of `nodeIndex`
label	...	region label (from the skeleton file), may be `all`
expr	...	the expression to evaluate

The following `nsum_example.sg` file specifies the equation for all points inside region `AIR`. For each point `i` it sums the expression for the neighbor points `j`.

```
1     equ psi[i=AIR] -> nsum(i,j,all,
2                          ((psi[i]-psi[node(i,j)]) / elen(i,j)) * ilen(i,j)
3                        ) = rho[i] * area(i) / EPS;
```

source_code/nsum_example.sg

Here `elen(i,j)` = $d_{i,j}$ is the edge length, `ilen(i,j)` = $A_{i,j}$ the integration area (3D) or integration length (2D) (which is also a length in a projection, hence the name `ilen`) and `area(i)` = V_i the box volume. The command `node(i,j)` gives the index of `j`-th neighbor of `i`. Since we deal with a two-dimensional problem, the area the flux is passing through is made up of the perpendicular bisectors given by `ilen(i,j)`.

The label summation function `lsum` sums over all nodes with a particular label and evaluates an expression on each node. The example `lsum_example.sg` demonstrates how to sum up the expression for all points `i` that are labelled `METAL1`.

`lsum(` *nodeIndex, label, expr* `)`

nodeIndex	...	current node
label	...	group of nodes (from skeleton file)
expr	...	the expression to evaluate

```
1     assign Q1 = lsum(i,METAL1,
2                   nsum(i,j,SIO2,
3                        D(EPSr,V[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)
4                       )
5                 );
```

source_code/lsum_example.sg

5.5 Skeleton Mesh – Boundaries

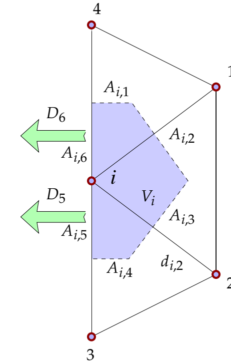


Figure 5.11: Points 4, i and 3 form a boundary.

In Section 5.3 the two-dimensional Laplace equation and its solution on an unstructured mesh including Dirichlet boundary conditions was considered. We have already given some hints on how to implement Neumann boundary conditions in Section 4.2.2, now we will look at it in more detail and also consider non-homogeneous Neumann boundary conditions. The implementation will be explained by the example of the discretized Poisson equation:

$$\sum_{j \in \mathcal{N}_i} D_{i,j} A_{i,j} = \rho_i V_i$$

In Fig. 5.11, a box at a domain boundary is illustrated. As discussed in section 4.2.2, Neumann boundary conditions with zero out-flux are implemented into SGFRAMEWORK by means of the implicit condition

$$\sum_{\substack{\text{all neighbors j} \\ \text{in current segment}}} D_{i,j} A_{i,j} = \rho_i V_i \tag{5.2}$$

which in SGFRAMEWORK reads as

```
1    equ psi[i=AIR] -> nsum(i,j,all,
2                          ((psi[i]-psi[node(i,j)]) / elen(i,j)) * ilen(i,j)
3                        ) = rho[i] * area(i) / EPS;
```

source_code/nsum_example.sg

Effectively, the code remains the same, all boundary handling is done within SGFRAMEWORK.

When dealing with non-zero out-flux, (5.2) has to be modified to take additional flux contributions at the boundary into account:

$$\sum_{\substack{\text{all neighbors j} \\ \text{in current segment}}} D_{i,j} A_{i,j} + D_5 \frac{l_5}{2} + D_6 \frac{l_6}{2} = \rho_i V_i \, .$$

```
1    nsum(i,j,all,
2          F(psi[i],psi[node(i,j),elen(i,j)) * ilen(i,j)
3        ) +
4    nsum(i,j,BND,
5          Fext * elen(i,j) / 2
6        ) = rho[i] * area(i) / EPS;
```

source_code/nsum_boundary_example.sg

File `capacitor_geometry.sg` gives an example of how the structure in Fig. 5.12 is realized in SGFRAMEWORK.

Two different metal contacts `METAL1` and `METAL2` require further extensions to our skeleton file, as can be seen in File `capacitor_geometry.sk`. More edges and regions are needed as the `METAL2` contact is area-like. The command `NOFLUX` has to be used for external Neumann boundaries, otherwise SGFRAMEWORK cannot triangulate the geometry.

```
1 const dx = 1.0, W1 = 8, W2 = 4, W3 = 8;
2 const W = W1 + W2 + W3, H = 20, d = 3;
```

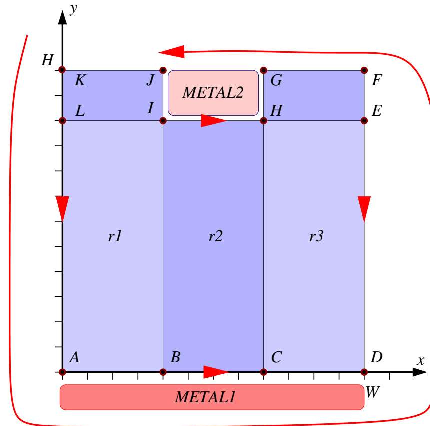


Figure 5.12: Complex geometry as METAL2 contact has three edges.

```
3
4   point pA = (0, 0);
5   point pB = (W1, 0);
6   point pC = (W1+W2, 0);
7   point pD = (W, 0);
8   point pE = (W, H-d);
9   point pF = (W, H);
10  point pG = (W1+W2, H);
11  point pH = (W1+W2, H-d);
12  point pI = (W1, H-d);
13  point pJ = (W1, H);
14  point pK = (0, H);
15  point pL = (0, H-d);
16
17  edge eAB = METAL1 [pA,pB] (dx, 0.0);
18  edge eBC = METAL1 [pB,pC] (dx, 0.0);
19  edge eCD = METAL1 [pC,pD] (dx, 0.0);
20  edge eED = NOFLUX [pE,pD] (dx, 0.0);
21  edge eFE = NOFLUX [pF,pE] (dx, 0.0);
22  edge eGF = NOFLUX [pG,pF] (dx, 0.0);
23  edge eGH = METAL2 [pG,pH] (dx, 0.0);
24  edge eIH = METAL2 [pI,pH] (dx, 0.0);
25  edge eJI = METAL2 [pJ,pI] (dx, 0.0);
26  edge eKJ = NOFLUX [pK,pJ] (dx, 0.0);
27  edge eKL = NOFLUX [pK,pL] (dx, 0.0);
28  edge eLA = NOFLUX [pL,pA] (dx, 0.0);
29
30  edge eHC =        [pH,pC] (dx, 0.0);
31  edge eHE =        [pH,pE] (dx, 0.0);
32  edge eIB =        [pI,pB] (dx, 0.0);
33  edge eLI =        [pL,pI] (dx, 0.0);
34
35  // rectangluar mesh vs. triangular mesh (without 'RECTANGLES')
36  region r1 = SIO2 {eAB,eIB,eLI,eLA} RECTANGLES;
37  region r2 = SIO2 {eBC,eHC,eIH,eIB} RECTANGLES;
38  region r3 = SIO2 {eCD,eED,eHE,eHC} RECTANGLES;
39  region r4 = SIO2 {eHE,eFE,eGF,eGH} RECTANGLES;
40  region r6 = SIO2 {eLI,eJI,eKJ,eKL} RECTANGLES;
```

source_code/capacitor_geometry.sk

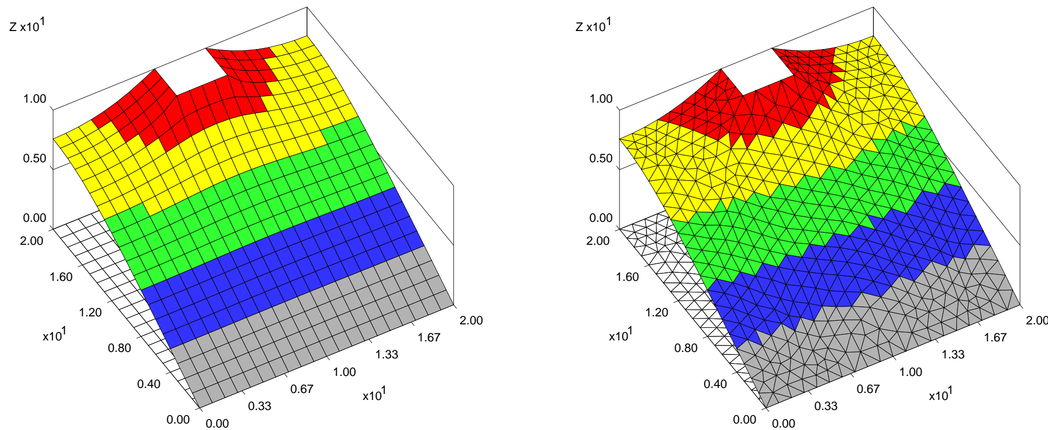


Figure 5.13: Simulation results on a rectangular grid (left) and triangular mesh (right) after a successful run of `capacitor_geometry.sg`.

```
mesh "capacitor_geometry.msh";

const    EPSr = 3.9;
const    EPSo = 8.854e-12;
var      psi[NODES], Q1, Q2;
unknown  psi[all];
known    psi[METAL1], psi[METAL2];

func E(psi1,psi2,<h>)            return  (psi1-psi2) / h;
func D(<epsr>,psi1,psi2,<h>)     return  epsr * EPSo * E(psi1,psi2,h);

equ psi[i=SIO2] ->
  nsum(i,j,all,D(EPSr,psi[i],psi[node(i,j)],elen(i,j))*ilen(i,j)) = 0.0;

begin main

  assign psi[i=METAL1] = 0.0;
  assign psi[i=METAL2] = 10.0;

solve;

  assign Q1 = lsum(i,METAL1,
    nsum(i,j,SIO2,D(EPSr,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)));
  assign Q2 = lsum(i,METAL2,
    nsum(i,j,SIO2,D(EPSr,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)));

write;

end
```

source_code/capacitor_geometry.sg

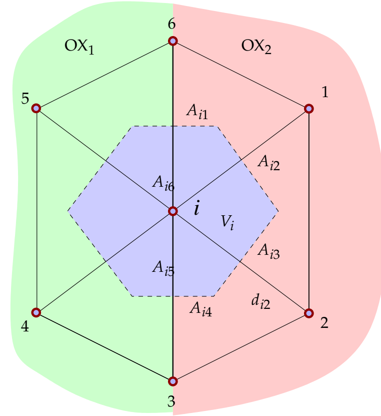


Figure 5.14: Interface conditions at the boundary between OX_1 and OX_2.

5.6 Skeleton Mesh – Interfaces

Fig. 5.14 shows an interface condition between two different oxides. This time, we have to split the flux contributions into two terms, so that

$$\sum_{j=1,2,3,6} D_{i,j} A_{i,j}^{OX_2} + \sum_{j=3,4,5,6} D_{i,j} A_{i,j}^{OX_1} = \rho_i V_i . \tag{5.3}$$

The first term represents the contributions inside OX_2 with the surrounding grid points 1, 2, 3, 6, while the second sum represents all contributions from OX_1 enclosed by 3, 4, 5, 6. An implementation in SGFRAMEWORK is

```
1     equ psi[i=INT] -> nsum(i,j,OX1,F(....) * ilen(i,j)) +
2                       nsum(i,j,OX2,F(....) * ilen(i,j)) = rho[i] * area(i) * EPS;
```

source_code/mesh_interfaces1.sg

Again, an example (Fig. 5.15 and 5.16) may provide a better understanding. Between two metal contacts we analyze the electrostatic field on unstructured mesh. The interface conditions have to be specified separately. The two contacts determine Dirichlet boundary conditions, while we assume homogeneous Neumann boundaries everywhere else on the boundary. The specifiers `INT` in the skeleton file `capacitor_example2.sk` act as a label for edges that determine the interface. `OX1` and `OX2` as region labels are used as filters for the grid points. **Pay attention**[1] to specify the interface conditions before the segment conditions!

```
1  mesh "capacitor_example2.msh";
2
3  const   EPSr1 = 3.9;
4  const   EPSr2 = 11.8;
5  const   EPSo = 8.854e-12;
6  var     psi[NODES], Q1, Q2;
7  unknown psi[all];
8  known   psi[METAL1], psi[METAL2];
9
10 func E(psi1,psi2,<h>)        return  (psi1-psi2) / h ;
11 func D(<epsr>,psi1,psi2,<h>) return  epsr * EPSo * E(psi1,psi2,h);
12
```

[1] **to pay attention to sth.** [peɪ ɛˈten.tʃən]: auf etwas achten

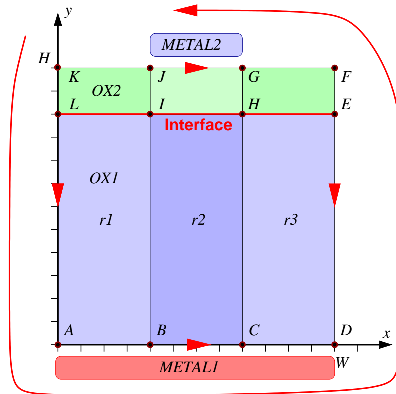


Figure 5.15: Geometry with additional interface between OX_1 and OX_2.

```
13  //interface has to come first
14  equ psi[i=INT] -> nsum(i,j,OX1,
15                        D(EPSr1,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)) +
16                    nsum(i,j,OX2,
17                        D(EPSr2,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j))
18                    = 0.0;
19
20  //Laplace for OX1
21  equ psi[i=OX1] ->
22    nsum(i,j,OX1,D(EPSr1,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)) = 0.0;
23
24  //Laplace for OX2
25  equ psi[i=OX2] ->
26    nsum(i,j,OX2,D(EPSr2,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)) = 0.0;
27
28  begin main
29
30    assign psi[i=METAL1] = 0.0;
31    assign psi[i=METAL2] = 10.0;
32
33    solve;
34
35    assign Q1 = lsum(i,METAL1,
36      nsum(i,j,OX1,D(EPSr1,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)));
37    assign Q2 = lsum(i,METAL2,
38      nsum(i,j,OX2,D(EPSr2,psi[i],psi[node(i,j)],elen(i,j)) * ilen(i,j)));
39
40    write;
41
42  end
```

source_code/capacitor_example2.sg

```
1  const dx = 1.0, W1 = 8, W2 = 4, W3 = 8;
2  const W = W1 + W2 + W3, H = 20, d = 3;
3
4  point pA = (0, 0);
5  point pB = (W1, 0);
6  point pC = (W1+W2, 0);
7  point pD = (W, 0);
8  point pE = (W, H-d);
```

```
 9  point pF = (W, H);
10  point pG = (W1+W2, H);
11  point pH = (W1+W2, H-d);
12  point pI = (W1, H-d);
13  point pJ = (W1, H);
14  point pK = (0, H);
15  point pL = (0, H-d);
16
17  edge eAB = METAL1 [pA,pB] (dx, 0.0);
18  edge eBC = METAL1 [pB,pC] (dx, 0.0);
19  edge eCD = METAL1 [pC,pD] (dx, 0.0);
20
21  edge eED = NOFLUX [pE,pD] (dx, 0.0);
22  edge eFE = NOFLUX [pF,pE] (dx, 0.0);
23  edge eLA = NOFLUX [pL,pA] (dx, 0.0);
24  edge eKL = NOFLUX [pK,pL] (dx, 0.0);
25
26  edge eHE = INT    [pH,pE] (dx, 0.0);
27  edge eIH = INT    [pI,pH] (dx, 0.0);
28  edge eLI = INT    [pL,pI] (dx, 0.0);
29
30  edge eGF = NOFLUX [pG,pF] (dx, 0.0);
31  edge eJG = METAL2 [pJ,pG] (dx, 0.0);
32  edge eKJ = NOFLUX [pK,pJ] (dx, 0.0);
33
34  edge eJI = NOFLUX [pJ,pI] (dx, 0.0);
35  edge eGH = NOFLUX [pG,pH] (dx, 0.0);
36
37  edge eHC =        [pH,pC] (dx, 0.0);
38  edge eIB =        [pI,pB] (dx, 0.0);
39
40  //rectangluar mesh vs. triangular mesh (without 'RECTANGLES')
41  region r1 = OX1 {eAB,eIB,eLI,eLA} RECTANGLES;
42  region r2 = OX1 {eBC,eHC,eIH,eIB} RECTANGLES;
43  region r3 = OX1 {eCD,eED,eHE,eHC} RECTANGLES;
44  region r4 = OX2 {eHE,eFE,eGF,eGH} RECTANGLES;
45  region r5 = OX2 {eIH,eGH,eJG,eJI} RECTANGLES;
46  region r6 = OX2 {eLI,eJI,eKJ,eKL} RECTANGLES;
```

source_code/capacitor_example2.sk

5.7 Mesh Refinement

Mesh refinement is useful to increase the resolution in certain areas of the device. Sophisticated algorithms solve a given problem on a coarse mesh, apply some error indicators to this **preliminary**[1] solution and refine the mesh in those regions. In SGFRAMEWORK, refinement has to be done manually: The role of the error estimator is assigned to the user, who has to supply a function describing the level of refinement.

The refinement criteria can be specified in the skeleton files right after the region statements in the form of one or more refinement functions. In order to determine whether a mesh element should be refined, the mesh refinement routine will evaluate the refinement function(s) at each of the element's vertices. If the measure between two grid points $M_{i,j}$ exceeds a reference distance d, the element will be refined, i.e. it will be split into two or more (hence smaller)

[1] **preliminary** [prɪˈlɪm.ɪ.nᵊr.i]: vorläufig, vorübergehend

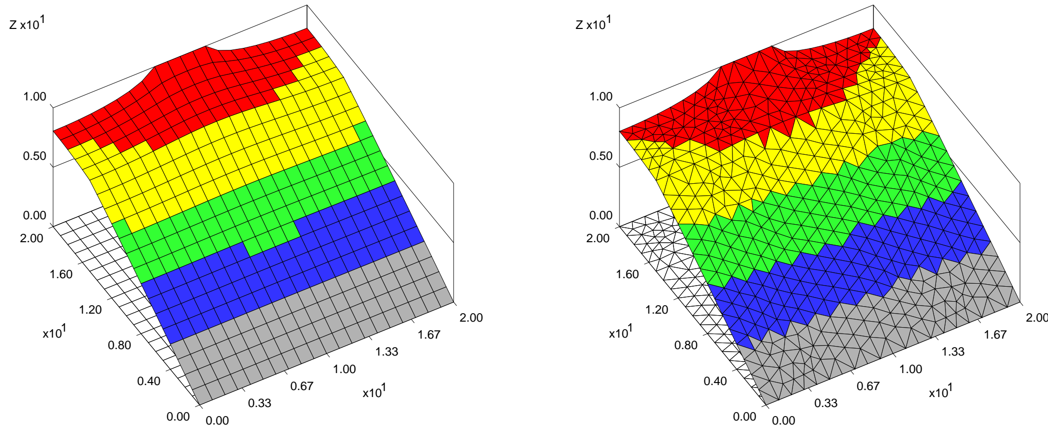


Figure 5.16: Results obtained on a rectangular grid (left) and a triangular mesh (right) for the input file `capacitor_example2.sg`.

elements. The function itself has a name, two refinement parameters and a body, using the syntax:

`refine` *name(measure, refDistance) = expr*

name	...	name of the criterion (arbitrary)
measure	...	refinement measure
refDistance	...	reference distance
expr	...	the expression to compare with the reference distance

The refinement measure describes how the current vertex distances are weighted. It may be `linear`, `logarithmic`, or `signedlog` as defined by the following expressions:

$$M_{\text{lin}} = x, \quad M_{\text{log}} = \log(x), \quad M_{\text{slog}} = \text{ sign}\,(x)\log(1+|x|).$$

For example, in order to determine whether a triangular element needs to be refined in a linear way,

$$\begin{aligned} M_{1,2} &= |M(N_1) - M(N_2)| \\ M_{2,3} &= |M(N_2) - M(N_3)| \\ M_{1,3} &= |M(N_1) - M(N_3)| \end{aligned}$$

is calculated and compared to d. If any of the $M_{i,j}$ exceeds d, i.e. the refinement function varies strongly between the edges of the triangle, the triangle is refined. Note that refinement occurs where the refinement function has a large gradient, while the magnitude of the refinement function plays a minor role.

The last section of the refinement criterion determines the minimum and maximum number of divisions and the minimum and maximum lengths of edges. Each element in the initial grid is assigned a division level of zero. At each refinement step the division level of the current element is increased by one. The mesh refinement program loops through each element and if a refinement is necessary, a check is performed on its neighbors. If a neighboring element's division level is less than that of the considered one, the neighboring element is refined first. This is done to prevent too large variations in the dimensions of two adjacent elements. One possible refinement indicator function is shown in Fig. 5.17; it can be implemented as

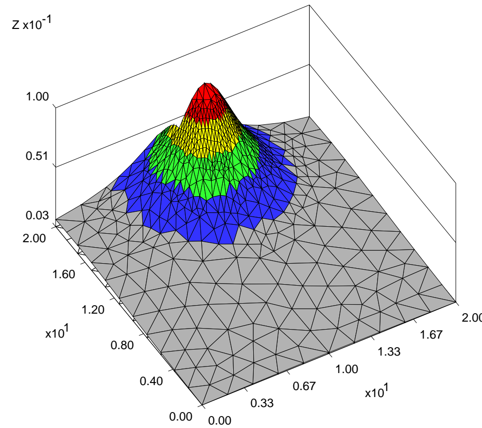


Figure 5.17: Distance Function used in `refine.sk`.

```
1    refine Dist(linear,0.01) = 1/(10 + sq(x-W/2) + sq(y-(H-d)));
2    set minimum divisions = 0;
3    set maximum divisions = 5;
```

source_code/refinement_example1.sk

The chosen refinement function results in a refinement at the edges of the upper metal contact, where the distance function has its maximum (Fig 5.18). The number of nodes increases from $N = 155$ to $N' = 541$ as the capacity $C = 34.53\,\text{pF}$ changes to $C' = 34.10\,\text{pF}$, suggesting an initial error of 1.3%.

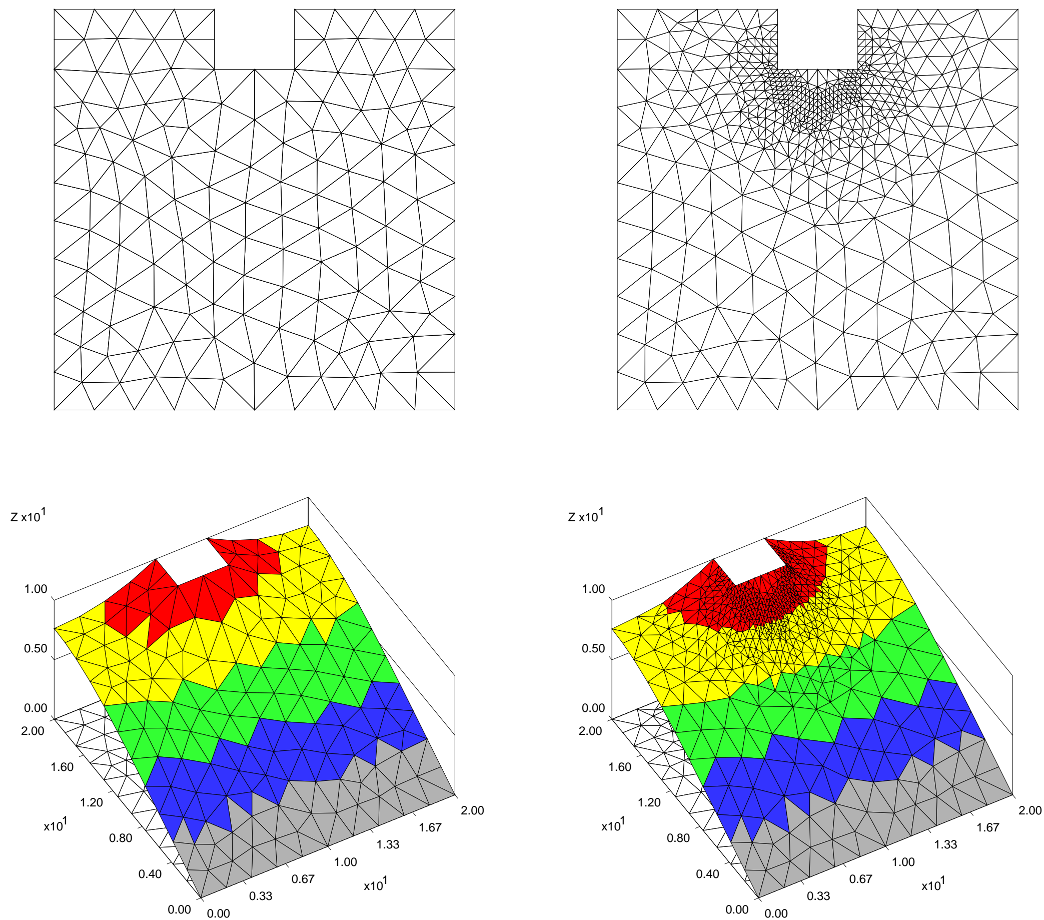


Figure 5.18: Mesh before (top left) and after refinement (top right). Potential without (bottom left) and with refinement (bottom right) using those meshes.

Chapter 6

Transport Phenomena and their Numerical Analysis

In the **preceding**[1] chapters we have learnt how to discretize derivatives and finally how to solve Poisson's equation. Poisson's equation describes an equilibrium distribution of some physical quantity and is numerically well-behaved. In this chapter we are going to simulate the transient behavior of a device, which means that we have to **cope**[2] with an additional time dependence. Our ultimate aim is the solution of the drift-diffusion model as given in (1.23) to (1.25), but before we are ready to do so, we will analyze the different contributions to each of the equations.

The equation describing conservation of a quantity n in general form is

$$\text{Increase in Time} + \text{Outflux} = \text{Production Rate} ,$$

or, mathematically

$$\frac{\partial n}{\partial t} + \nabla \cdot \Gamma = s \,. \tag{6.1}$$

The flux term Γ may consist of two distinct contributions:

- *Diffusion*: For $\Gamma = -D\nabla n$ with (constant) diffusion coefficient D, the conservation equation becomes

  $$\frac{\partial n}{\partial t} - D\nabla^2 n = s,$$

  which is a parabolic partial differential equation. In pure diffusive processes, relaxation to an equilibrium determined by the boundary conditions can be observed (cf. Fig. 6.1). For example, with homogeneous Neumann boundary conditions (i.e. no out-flux), $\int_{\mathcal{V}} n(x)\mathrm{d}V = \text{const.}$, while this is not the case with inhomogeneous boundary conditions.

- *Convection (Drift)*: $\Gamma = n\mu E$ with (constant) mobility μ. The conservation equation then is a hyperbolic partial differential equation of first order,

  $$\frac{\partial n}{\partial t} + \mu\nabla \cdot (nE) = S.$$

[1] **preceding** [prɪˈsiː.dɪŋ]: vorangegangen [2] **to cope** [koʊp]: zurechtkommen, beherrschen

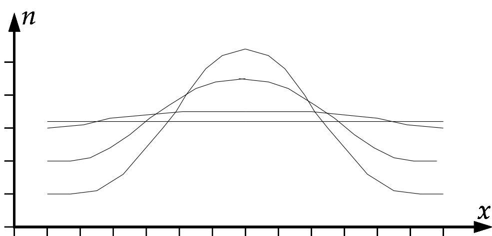


Figure 6.1: A pure diffusive process relaxes to an equilibrium over time if there is no out-flux.

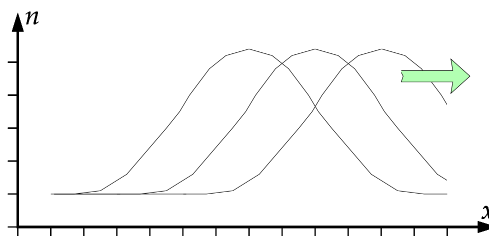


Figure 6.2: A pure convective process describes travelling waves.

Pure convective processes describe travelling waves, meaning that the initial shape of $n(x)$ remains unchanged (cf. Fig. 6.2). Because of the wave-like behavior, there is a flux through boundaries.

In most systems both contributions are present, complicating the solution process. This chapter deals with discretization in the time domain and studies the numerical stability of these schemes.

6.1 Discretization in the Time Domain

So far we have dealt with the approximation of derivatives in spatial coordinates only. For the Poisson equation, both the five-point and the nine-point stencil lead to **satisfactory**[1] results without numerical stability issues. For the time discretization of parabolic partial differential equations, this is not the case anymore.

Let us consider the ordinary differential equation

$$\frac{dn(t)}{dt} = h(n(t), t)$$

for an arbitrary, smooth function $h(\cdot, \cdot)$. Without any further assumptions on h and knowing $n(t_k)$ we have mainly two possibilities to obtain $n(t_{k+1})$ after discretization of the time-derivative: We can evaluate the right hand side at t_k or at t_{k+1}.

In the first case, the scheme is called *forward Euler scheme* (aka. *explicit Euler scheme*):

$$\frac{n(t_{k+1}) - n(t_k)}{t_{k+1} - t_k} = h(n(t_k), t_k) \quad \iff \quad n(t_{k+1}) = n(t_k) + (t_{k+1} - t_k)h(n(t_k), t_k)$$

Since $n(t_k)$ is known, we immediately obtain $n(t_{k+1})$ and are able to proceed to the next time step.

The second possibility for the evaluation of the right hand side leads to the *backward Euler scheme* (aka. *implicit Euler scheme*):

$$\frac{n(t_{k+1}) - n(t_k)}{t_{k+1} - t_k} = h(n(t_{k+1}), t_{k+1})$$

[1] **satisfactory** [sæt.ɪsˈfækt.ᵊr.i]: zufriedenstellend

The solution process for $n(t_{k+1})$ now strongly depends on h. Especially if h is non-linear, one usually has to rely on *Newton's method* (cf. Section 3.6) and solve a (system of) non-linear equation(s) within each time step.

If we denote the step size in time with $\Delta t := t_{k+1} - t_k$, then both the solutions obtained by the explicit Euler scheme and the backward Euler scheme converge to the true solution for $\Delta t \to 0$ with order $O(\Delta t)$.

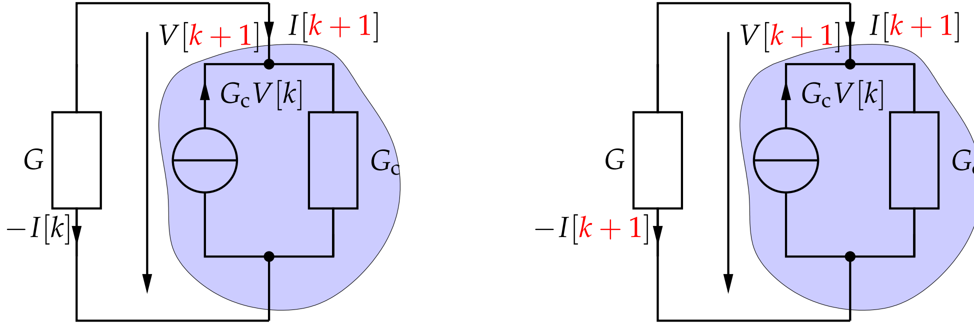


Figure 6.3: Time discretization of a discharging capacitor with Forward Euler (left) and Backward Euler (right) scheme

To become **familiar**[1] with the Euler schemes, let us have a look at an initially charged capacitor that discharges over a resistor with conductance $G = 1/R$, shown in Fig. 6.3. The governing equation for the voltage over the capacitor is

$$C\frac{\mathrm{d}V(t)}{\mathrm{d}t} = -GV(t).$$

A discretization using the explicit Euler scheme leads to

$$C\frac{V_{k+1} - V_k}{\Delta t} = -GV_k.$$

With the auxiliary conductance $G_C := C/\Delta t$ we get

$$V_{k+1} = V_k\frac{G_C - G}{G_C} = V_0\left(\frac{G_C - G}{G_C}\right)^{k+1}.$$

A discretization using the implicit Euler scheme results in

$$C\frac{V_{k+1} - V_k}{\Delta t} = -GV_{k+1} \quad \Longleftrightarrow \quad V_{k+1} = V_k\frac{G_C}{G_C + G} = V_0\left(\frac{G_C}{G_C + G}\right)^{k+1}$$

As initial condition we set $V_0 = 1$ and compare with the analytical solution $V(t) = \exp(-t/\tau)$, where $\tau = RC = \Delta t G_C/G$. Since $G_C/G = \tau/\Delta t$, we can rewrite the equations obtained by the Euler schemes as

$$V_k = V_0\left(1 - \frac{\Delta t}{\tau}\right)^k \qquad \text{(explicit Euler)}$$

$$V_k = V_0\left(1 + \frac{\Delta t}{\tau}\right)^{-k} \qquad \text{(implicit Euler)}$$

[1] **familiar** [fəˈmɪl.i.jɚ]: vertraut, geläufig

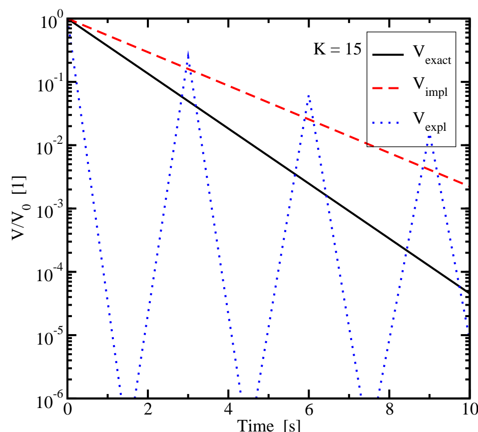


Figure 6.4: A discretization with the forward Euler scheme leads to numerical instabilities, whereas the backward Euler scheme remains stable.

On closer inspection[1], one can see that the solution obtained by the implicit Euler scheme tends to zero as $k \to \infty$ for all $\tau, \Delta t > 0$. However, the explicit Euler scheme leads to $V_1 < 0$ if $\frac{\tau}{\Delta t} > 1$, which is illustrated in Fig. 6.4. This behavior is not just specific for this example only, as the next section will show.

Note that for fixed $t = k\Delta t$ the explicit Euler scheme gives us

$$\begin{aligned} \lim_{\Delta t \to 0} V(t) &= V(0) \lim_{k \to \infty} (1 - \frac{t}{\tau k})^k , \\ &= V(0) \mathrm{e}^{-\frac{t}{\tau}} \\ &= V(t) , \end{aligned}$$

while for the implicit Euler scheme we find

$$\begin{aligned} \lim_{\Delta t \to 0} V(t) &= V(0) \lim_{k \to \infty} (1 + \frac{t}{\tau k})^{-k} && \text{(implicit Euler)} \\ &= V(0) 1/\mathrm{e}^{\frac{t}{\tau}} \\ &= V(t) . \end{aligned}$$

An implementation in SGFRAMEWORK reads

```
const C  = 0.1,    R  = 10.0,    G    = 1/R;
const K  = 1,      dt = 0.1*K,   tmax = 10;
const GC = C/dt,   N  = 100/K,   V0   = 1;

var  t[N+1], Vimpl[N+1], Vexpl[N+1], Vexact[N+1];

begin main
  assign t[i=all]        = i*dt;
  assign Vimpl[i=0]      = V0;
  assign Vimpl[i=1..N]   = Vimpl[i-1] * GC/(GC+G);

  assign Vexpl[i=0]      = V0;
  assign Vexpl[i=1..N]   = Vexpl[i-1] * (GC-G)/GC;
```

[1] **on closer inspection** [ɔn kloʊsər ɪnˈspek.ʃᵊn]: bei näherer Betrachtung

```
14
15    assign Vexact[i=0..N] = V0 * exp(-t[i]/(R*C));
16    write;
17  end
```

source_code/RC_circuit.sg

6.2 Stability of Discretization Schemes

The Nyquist-Shannon sampling theorem states that if a signal is band-limited to W, a sampling frequency of $2W$ is sufficient for exact reconstruction. A related question arises for numerical discretization schemes: Will the chosen discretization yield meaningful results? Which schemes are "better" than others? Which is the maximum allowed distance between grid points?

We will investigate stability using a technique introduced by John von Neumann and start with the continuous *Fourier transform*,

$$N(\chi, t) = \mathscr{F}\{n(x,t)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-j\chi x} n(x,t)\,\mathrm{d}x,$$
$$n(x,t) = \mathscr{F}^{-1}\{N(\chi,t)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{j\chi x} N(\chi,t)\,\mathrm{d}\xi$$

and the Fourier transform for discrete functions,

$$N(\xi,k) = \mathscr{F}\{n_{i,k}\} = \frac{1}{\sqrt{2\pi}} \sum_{i=-\infty}^{\infty} e^{-j\Delta x \xi i} n_{i,k} \Delta x\ ,$$
$$n_{i,k} = \mathscr{F}^{-1}\{N(\xi,k)\} = \frac{1}{\sqrt{2\pi}} \int_{-\pi/\Delta x}^{\pi/\Delta x} e^{ji\Delta x \xi} N(\xi,k)\,\mathrm{d}\xi\ ,$$

where $\xi \in [-\pi/\Delta x, \pi/\Delta x]$. For displacement and differentiation in space in the continuous case,

$$\mathscr{F}\{n(x+x_0,t)\} = e^{j\chi x_0} N(\chi,t), \quad \mathscr{F}\{\frac{\partial n(x,t)}{\partial x}\} = j\chi N(\chi,t)$$

holds, whereas for the discrete version we have

$$\mathscr{F}\{n(i+i_0,k)\} = e^{j\Delta x \xi i_0} N(\xi,t). \tag{6.2}$$

Let us first consider a Forward Euler discretization of the diffusion equation,

$$n_{i,k+1} = n_{i,k} + \eta D[n_{i+1,k} - 2n_{i,k} + n_{i-1,k}], \tag{6.3}$$

where $\eta = \frac{\Delta t}{(\Delta x)^2}$. We use as initial condition $n_{i,0} = N_0 \delta_{i-i_0}$ (think about why the choice $n_{i,0} = N_0$ does not yield a too meaningful example!). At $k = 1$ we get

$$\begin{aligned}
n_{i_0-1,1} &= 0 \quad + \eta D\,(N_0 \quad - 0 \quad + 0 \quad) = \eta D N_0 \\
n_{i_0\quad,1} &= N_0 \quad + \eta D\,(0 \quad - 2N_0 + 0 \quad) = N_0\,(1 - 2\eta D) \\
n_{i_0+1,1} &= 0 \quad + \eta D\,(0 \quad - 0 \quad + N_0\ ) = \eta D N_0
\end{aligned} \tag{6.4}$$

In particular, if $1 - 2\eta D < 0 \Leftrightarrow 2\eta D > 1 \Leftrightarrow \Delta t > (\Delta x)^2/2D$ we get $n_{i_0,1} < 0$, which is meaningless from a physical point of view.

A more stringent investigation of the stability properties of (6.3) is as follows: A transformation of (6.3) into the frequency domain and application of the displacement properties yields

$$\begin{aligned} N(\xi, k+1) &= N(\xi, k) + \eta D N(\xi, k) \left(e^{j\Delta x \xi} - 2 + e^{-j\Delta x \xi} \right) \\ &= N(\xi, k) \left(1 + 2\eta D \left(\cos(\Delta x \xi) - 1 \right) \right). \end{aligned}$$

If we now define the *amplification factor* $g(\Delta x \xi) := 1 + 2\eta D \left(\cos(\Delta x \xi) - 1 \right)$ we immediately see that

$$N(\xi, k+1) = N(\xi, k) g(\Delta x \xi) = N(\xi, 0) \left(g(\Delta x \xi) \right)^{k+1},$$

so if we require that the solution stays bounded as $k \to \infty$, we require

$$|g(\Delta x \xi)| \leq 1.$$

For our example, we have $g(\Delta x \xi) = 1 + 2\eta D \left(\cos(\Delta x \xi) - 1 \right) = 1 - 4\eta D \sin^2 \left(\frac{\Delta x \xi}{2} \right)$, so in order to fulfill $-1 \leq g(\Delta x \xi) \leq 1$,

$$0 \leq 4\eta D \sin^2 \left(\frac{\Delta x \xi}{2} \right) \leq 2.$$

The lower inequality is trivially fulfilled, but the upper requires

$$2\eta D \leq 1 \quad \overset{\eta = \frac{\Delta t}{(\Delta x)^2}}{\Longleftrightarrow} \quad \Delta t \leq \frac{(\Delta x)^2}{2D}. \tag{6.5}$$

Therefore, discretization in time and space are tightly connected for the Forward Euler scheme. Note that we have already found such a condition in our simple example (6.4).

Let us have a look at the Backward Euler scheme as well. We will use the general scheme of replacing $n_{i,k}$ by $(g(\Delta x \xi))^k e^{-ji\Delta x \xi}$ in

$$n_{i,k} - n_{i,k-1} = \eta D \left(n_{i+1,k} - 2n_{i,k} + n_{i-1,k} \right)$$

and obtain

$$g(\Delta x \xi) - 1 = \eta D g(\Delta x \xi) \left(e^{-j\Delta x \xi} - 2 + e^{j\Delta x \xi} \right) = 2\eta D g(\Delta x \xi) \left(\cos(\Delta x \xi) - 1 \right).$$

With this, we find $g(\Delta x \xi)$ to be

$$g(\Delta x \xi) = \frac{1}{1 + 4\eta D \sin^2 \left(\frac{\Delta x \xi}{2} \right)},$$

so $-1 \leq g(\Delta x \xi) \leq 1$ is always fulfilled. Therefore, the Backward Euler scheme is said to be *unconditionally stable*: The numerical results will be stable (which does not necessarily mean that they are good), no matter which discretization is chosen.

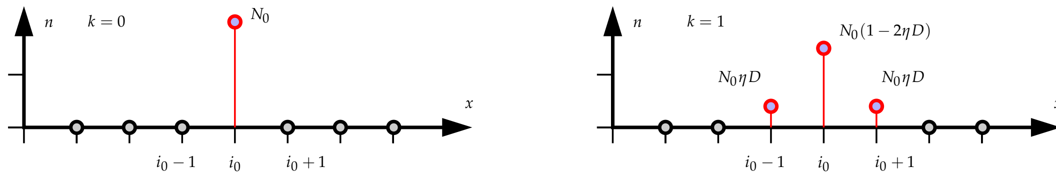


Figure 6.5: A Dirac impulse as initial condition confirms the results of the von Neumann analysis.

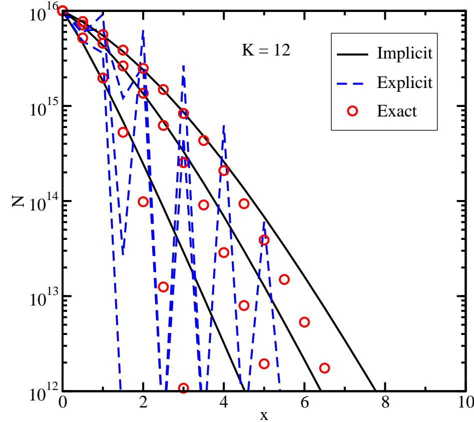


Figure 6.6: An explicit Euler scheme for the time discretization of the diffusion equation leads to an unstable numerical solution for large (time-)step sizes, whereas the implicit Euler scheme is unconditionally stable.

6.3 Diffusive Problems

For $\Gamma = -D\nabla n$ and $s = 0$ in (6.1), the *diffusion equation*

$$\frac{\partial n}{\partial t} = D\nabla^2 n \,,$$
$$n(x, t = 0) = n_0(x)$$

without source term is obtained. It is also known as *heat equation*, since the spreading of heat is also described by this equation.

Unlike the wave equation, the diffusion equation has very strong smoothing properties. This means that for $t > 0$ the solution $n(x,t)$ is smooth (infinitely differentiable), even if this is not true for n_0.

We are interested in the description of dopant diffusion, which will serve as an explanatory framework in this section. For this we assume the interface between the dopant source ($x < 0$) and the semiconductor ($x > 0$) at $x = 0$.

The diffusion coefficient D in a semiconductor is given by an *Arrhenius law*

$$D = D_0 \exp\left(-\frac{E_{\mathrm{act}}}{k_{\mathrm{B}} T_{\mathrm{L}}}\right) \,.$$

The parameters for phosphorus are $D_0 = 3.85\ \mathrm{cm}^2/\mathrm{s}$ and $E_{\mathrm{act}} = 3.66$ eV, which results in

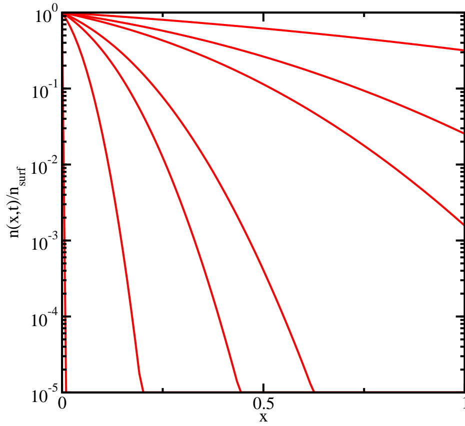


Figure 6.7: Dopant diffusion from an inexhaustible source (i.e. Dirichlet boundary condition) at $x = 0$.

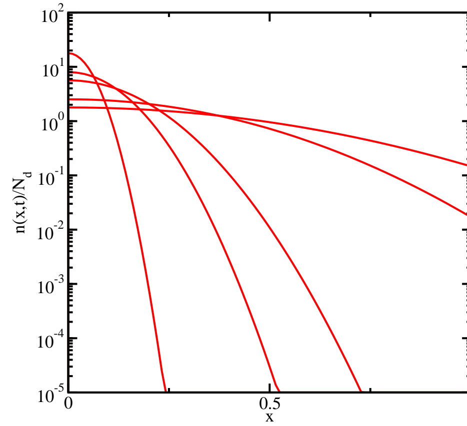


Figure 6.8: Dopant drive-in diffusion with a constant number of dopants (i.e. Neumann boundary condition) at $x = 0$.

$D = D_0 e^{-142}$ at $T_L = 300$ K. The diffusion front (which can be defined via the variance of u) at time t is located at depth $\sqrt{Dt}$. For $t > 0$, there are two types of boundary conditions possible:

- *Constant surface concentration*: The source of dopants is **inexhaustible**[1], thus $n(0,t) = n_{\text{surf}}$. Since dopants cannot spread over the whole device instantly, we have for every finite t

$$\lim_{x\to\infty} n(x,t) = 0 \,.$$

  In one spatial dimension, the analytical solution for the dopant diffusion problem is

$$n(x,t) = n_{\text{surf}} \operatorname{erfc}\left(\frac{x}{\sqrt{Dt}}\right) \,.$$

  As $t \to \infty$, the equilibrium (which is formally obtained from the solution of the equation that remains if all time derivatives are set to zero) is

$$\lim_{t\to\infty} n(x,t) = n_{\text{surf}} \,.$$

- *Constant dose (drive-in diffusion)*: Only a fixed amount of dopants is available, so

$$N_d(t) = \int_{\infty}^{\infty} n(x,t)\,\mathrm{d}x = \text{const.}$$

  As before,

$$\lim_{x\to\infty} n(x,t) = 0$$

  for every $t < \infty$. An analytical solution in one spatial dimension is

$$n(x,t) = \frac{N_d}{\sqrt{\pi D t}} \exp\left(-\frac{x^2}{4Dt}\right) \,,$$

[1] **inexhaustible** [ɪn.ɪgˈzɔː.stɪ.bl̩]: unerschöpflich

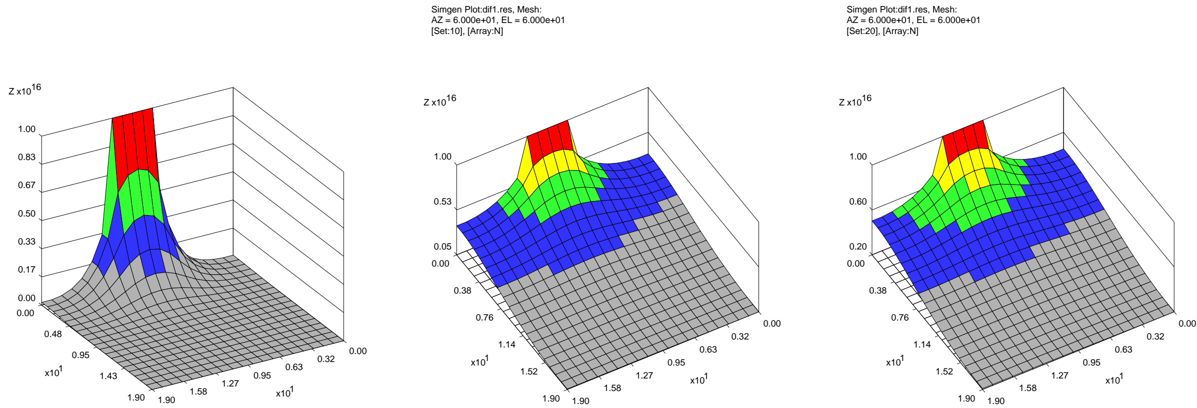


Figure 6.9: Results for dopant diffusion with constant surface concentration for three equidistant time-steps.

so the equilibrium solution for a device with infinite length is

$$\lim_{t\to\infty} n(x,t) = 0 \, .$$

However, real devices have a finite length l, so that

$$\lim_{t\to\infty} n(x,t) = \frac{N_\mathrm{d}}{l} \, .$$

Let us consider an example for each boundary condition. In the case of constant surface concentration at Γ_surf, the initial condition in our simulation domain Ω is

$$n(x,t=0) = \begin{cases} n_\mathrm{surf}, & x \in \Gamma_\mathrm{surf} \\ 0, & x \in \Omega \setminus \Gamma_\mathrm{surf} \end{cases} \, . \tag{6.6}$$

The boundary condition is $n(x,t) = n_\mathrm{surf}$ for $x \in \Gamma_\mathrm{surf}$, and there is no out-flux elsewhere, hence $\boldsymbol{\Gamma} \cdot \nabla n = 0$, where $\boldsymbol{\Gamma}$ is the outer normal vector at each point on the boundary arc $\partial\Omega \setminus \Gamma_\mathrm{surf}$.

An implementation in SGFRAMEWORK reads

```
// constants
const NX    = 20;              // number of mesh points, x direction
const NY    = 20;              // number of mesh points, y direction
const L     = 10.0e-4;         // distance scaling (cm)
const dx    = L/NX;
const dy    = L/NY;
const T     = 300.0;           // operating temperature
const D     = 1.0e-2;          // diffusion constant
const dt    = 0.1*dx*dx/(2*D); // time step -- obeys von Neumann condition
const Xmin  = 8;
const Xmax  = 12;

// declare variables and specify the unknowns
var itime;
var iwrite;
var x[NX];
var y[NY];
var N[NX,NY], Nold[NX,NY];
unknown N[all,all];
known N[Xmin..Xmax,0];
```

```
21
22  // Diffusion equation
23  equ N[i=1..NX-2,j=1..NY-2] ->
24          D*(N[i+1,j  ]-2*N[i,j]+N[i-1,j  ])/(dx*dx) +
25          D*(N[i  ,j+1]-2*N[i,j]+N[i  ,j-1])/(dy*dy) -
26          (N[i,j]-Nold[i,j])/dt = 0.0;
27
28  //boundary at i=NX-1
29  equ N[i=NX-1,j=1..NY-2] -> N[i,j] -N[i-1,j] = 0.0;
30
31  // boundary at i=0
32  equ N[i=0,j=1..NY-2] -> N[i,j] - N[i+1,j] = 0.0;
33
34  // boundary at j=NY-1
35  equ N[i=1..NX-2,j=NY-1] -> N[i,j] - N[i,j-1] = 0.0;
36
37  // first boundary at j=0
38  equ N[i=1..Xmin-1,j=0] -> N[i,j] - N[i,j+1] = 0.0;
39
40  // second boundary at j=0
41  equ N[i=Xmax+1..NX-2,j=0] -> N[i,j] - N[i,j+1] = 0.0;
42
43  // corner boundary conditions:
44  equ N[i=0,j=0]          -> N[i,j] - N[i+1,j+1] = 0.0;
45  equ N[i=0,j=NY-1]       -> N[i,j] - N[i+1,j-1] = 0.0;
46  equ N[i=NX-1,j=0]       -> N[i,j] - N[i-1,j+1] = 0.0;
47  equ N[i=NX-1,j=NY-1]    -> N[i,j] - N[i-1,j-1] = 0.0;
48
49  // set the numerical algorithm parameters
50  set NEWTON_DAMPING     = 3;
51  set NEWTON_ACCURACY    = 1.0e+4;
52  set NEWTON_ITERATIONS  = 100;
53  set LINSOL_ALGORITHM   = GAUSSELIM;
54  set LINSOL_FILL        = INFINITY;
55
56  begin InitVars
57    assign x[i=all] = i*dx;
58    assign y[i=all] = i*dy;
59    assign N[i=5..15,j=0] = 1.0e16;
60    assign Nold[i=all,j=all] = N[i,j];
61  end
62
63  begin main
64    assign itime = 0;
65    call InitVars;
66    while (itime < 2000) begin
67      assign Nold[i=all,j=all] = N[i,j];
68      solve;
69      assign itime = itime + 1;
70      assign iwrite = iwrite + 1;
71      if (iwrite == 100) begin
72        write;
73        assign iwrite = 0;
74      end
75    end
76  end
```

source_code/diffusion_example1.sg

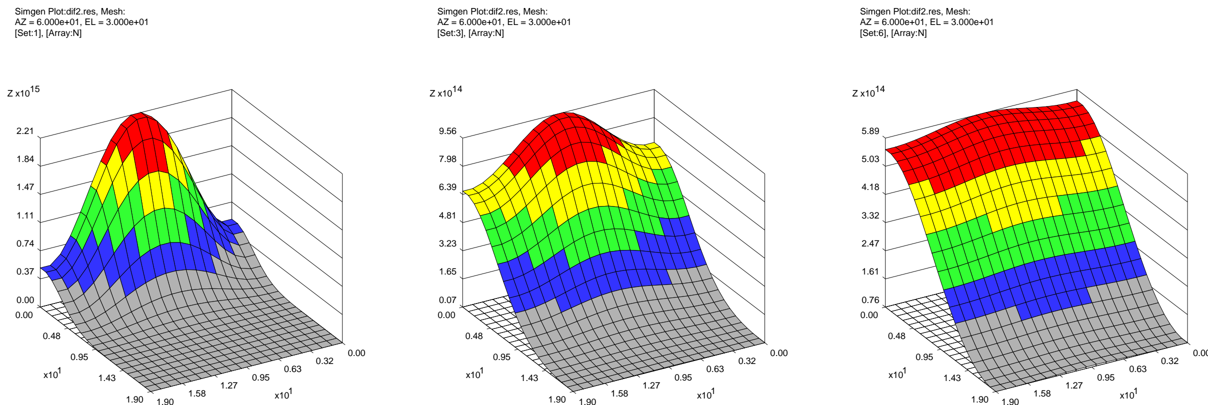


Figure 6.10: Results for drive-in dopant diffusion for three equidistant time-steps. The initial local peak concentration spreads over the whole domain and ultimately results in a constant (and smaller compared to the initial peak) concentration over the whole device.

Results are shown in Fig. 6.9. It can be seen that initially there is a steep gradient of the dopant concentration, because they only penetrate the surface. As time advances, the dopants spread over the whole simulation domain until finally after an infinite amount of time a constant dopant level n_{surf} in the whole material is reached.

For the second example, we choose the same initial conditions (6.6), but the boundary conditions are $\boldsymbol{\Gamma} \cdot \nabla n = 0$, where $\boldsymbol{\Gamma}$ is the outer normal vector at each point on the whole boundary Γ. This means that the number of dopant is fixed in the whole domain, because no particles enter or leave the simulation domain. The changes in the code for the previous example are

```
1  // changes for fixed number of dopants
2  unknown N[all,all];
3  // all variables are known now
4  // known N[Xmin..Xmax,0];
5
6  // contact side
7  // equ N[i=1..Xmin-1,     j=0] -> N[i,j] - N[i,j+1] = 0.0;
8  // equ N[i=Xmax+1..NX-2,j=0] -> N[i,j] - N[i,j+1] = 0.0;
9  // no outflux anywhere
10 equ N[i=1..NX-2,j=0] -> N[i,j] - N[i,j+1] = 0.0;
11
12 begin InitVars
13   assign N[i=5..15,j=1] = 1.0e16; // initial condition
14 end
```

source_code/diffusion_example2.sg

The numerical result are depicted in Fig. 6.10. Compared to the first example, the dopant concentration smoothes out rather quickly, because there is no more dopant source. Ultimately, the dopant concentration becomes constant in the whole domain, however, this time the equilibrium concentration is not n_{surf}, but determined by the number of particles initially available in the simulation domain.

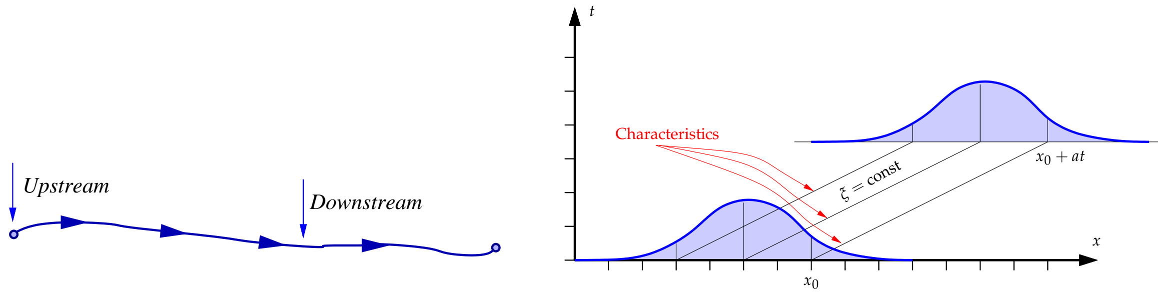


Figure 6.11: Boundary conditions have to be specified "upstream", so that they are initial conditions along a characteristic.

6.4 Convective Problems

In the introductory part of this chapter we have already discussed the convective case with $\Gamma = n\mu E$. Since the resulting partial differential equation is of first order, we have to specify one boundary condition.

The most prominent type of hyperbolic equations are wave equations, the simplest one being

$$\frac{\partial n}{\partial t} + a\frac{\partial n}{\partial x} = 0,$$
$$n(x,0) = n_0(x).$$

The solution of this equation is $n_0(x - at)$, which is a wave with initial shape $n_0(x)$ traveling with velocity a. Those lines in the (x,t)-plane with constant $x - at$ are called *characteristics* or *characteristic lines* and are illustrated in Fig. 6.11. The solution along a characteristic is determined by the initial conditions given at a single point on it. If, however, two points on a characteristic are specified by initial conditions, a solution cannot exist anymore, unless the two solutions induced by the two points happen to **coincide**[1]. This can be seen from Fig. 6.11: If we specify an initial condition at x_0 at $t = 0$, the value at all points $x_0 + at$ is determined. Thus, it is meaningless to specify at, say, $t = 2$ another condition at $x_0 + 2a$.

Let us have a closer look at the simple example

$$\frac{\partial n}{\partial t} + a\frac{\partial n}{\partial x} = 0, \quad 0 \leq x \leq 1,\ t \geq 0$$

If $a > 0$, the wave travels from left to right, therefore we specify the boundary condition on the left boundary: $n(x,0) = n_0(x)$, $n(0,t) = g(t)$. The analytical solution for $a > 0$ is then given as

$$n(x,t) = \begin{cases} n_0(x - at), & x - at > 0 \\ g(t - x/a), & x - at < 0 \end{cases}$$

and vice versa for $a < 0$.

A numerical discretization forward in time with right-sided differences in space results in

$$\frac{n_{i,k+1} - n_{i,k}}{\Delta t} + a\frac{n_{i+1,k} - n_{i,k}}{\Delta x} = 0\,,$$

[1] **to coincide** [koʊ.ɪnˈsaɪd]: übereinstimmen

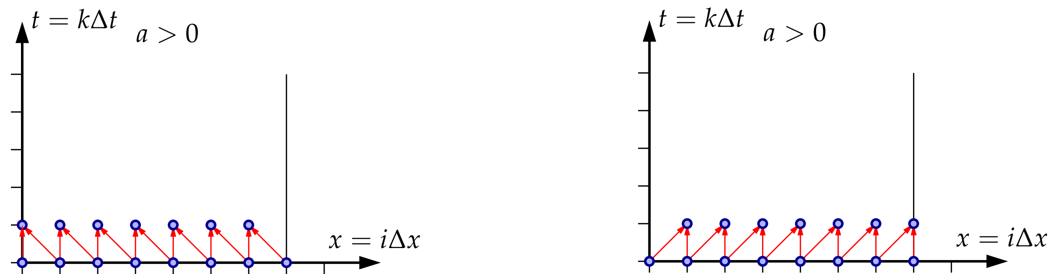


Figure 6.12: For a wave travelling from the left to the right, right-sided differences cannot reflect the wave-nature of the solution, while left-sided differences can. The opposite is true for a wave travelling from the right to the left.

leading to

$$n_{i,k+1} = n_{i,k}(1 + a\lambda) - a\lambda n_{i+1,k}\,, \qquad \lambda = \frac{\Delta t}{\Delta x}\,.$$

We see that the discretization is unable to fully reflect the wave type of the solution, because at point i no information from the left neighbor $i-1$ is used.

Using left-sided differences instead, we obtain

$$n_{i,k+1} = n_{i,k}(1 - a\lambda) + a\lambda n_{i-1,k}, \qquad \lambda = \frac{\Delta t}{\Delta x}\,. \tag{6.7}$$

Now data from the left neighbor is used leading to good results as the next example will show. Nevertheless, this does not mean that right-sided differences are worse than left-sided differences in general! Replacing a with $-a$, right-sided differences perform acceptably, while left-sided differences fail (Fig. 6.12) .

For demonstration purposes we will solve

$$\frac{\partial n}{\partial t} + a\frac{\partial n}{\partial x} = 0 \tag{6.8}$$

numerically with initial condition

$$n(x,0) = \begin{cases} 1 - |x - at - 1/2|, & |x| < 1 \\ 0, & \text{otherwise} \end{cases}$$

The results can be seen in Fig. 6.13 and illustrate the stability of left-sided differences and the instability of right-sided differences for the given equation.

```
1  const dt = 0.008, dx = 0.01, a = 1;
2  equ V[i=1..Nx] -> (V[i]     - Vold[i] ) / dt
3              + a * (Vold[i] - Vold[i-1]) / dx = 0;
4  begin calcA
5    assign A[i=0..Nx] = max(0,
6          1 - abs(i*dx - a*itime*dt - Nx/4*dx)*k);
7  end
8  begin main
9    ...
10   call calcA;
```

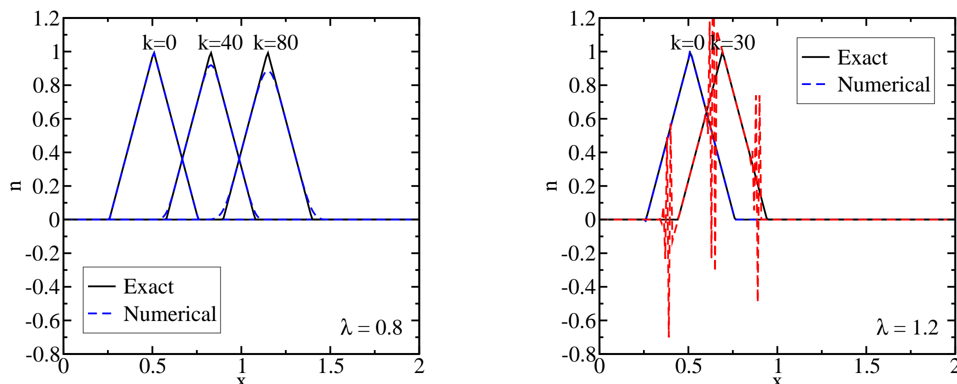


Figure 6.13: An appropriate difference scheme has to be chosen for hyperbolic equations (here: left-sided for a wave travelling from the left to the right), otherwise instabilities occur.

```
11    assign V[i=all] = A[i];
12    while (...) begin
13      assign Vold[i=all] = V[i];
14      solve;
15      call calcA;
16      assign Vex[i=all] = A[i];
17    end
18  end
```

source_code/hyperbolic_unstable.sg

We will use the von Neumann analysis to investigate the reason for instability a little bit further. Replacing $n_{i,k} = (g(\theta))^k e^{-ji\theta}$, $\theta = \Delta x \xi$ in (6.7) yields

$$g(\theta) = 1 + a\lambda(e^{-j\theta} - 1)$$

and with $e^{-j\theta} = \cos\theta - j\sin\theta$ the **modulus**[1] is

$$|g(\theta)|^2 = (1 + a\lambda\,(\cos(\theta) - 1))^2 + (a\lambda)^2 \sin^2(\theta) = 1 - 4a\lambda(1 - a\lambda)\sin^2(\theta/2)\,.$$

In order to satisfy $|g(\theta)| \leq 1$, the *Courant-Friedrichs-Lewy (CFL) Condition*

$$\boxed{|a\lambda| < 1} \tag{6.9}$$

is required for stability. This is quite intuitive: The discretized information "travels" at speed $1/\lambda$ so information at point i accesses information at point $i - 1/\lambda$. Thus, for a proper resolution of the information propagation, the grid size a should be smaller than $1/\lambda$ to resolve the flow of information.

Apart from stability issues with "wrong-sided" differences, numerical dispersion occurs even for "correct-sided" differences (cf. left illustration in Fig. 6.13). While the exact solution is a travelling hat, the numerical solution is smoothed out, which can be explained by realizing that different frequencies travel (numerically) at different speeds. This becomes – again – clearer

[1] **modulus** [mɔd.juː.ləs]: Absolutbetrag

with the von Neumann analysis. Setting $n_{i,k} = (g(\theta))^k e^{-jm\theta}$ and using (6.2) and $N(\xi, k+1) = g(\Delta x\xi)N(\xi, k)$ in (6.7) gives

$$g(\Delta x\xi) = (1 - a\lambda) + a\lambda e^{-j\Delta x\xi} .$$

The exact result $n(x,t) = n_0(x - at)$ has the Fourier transform

$$\begin{aligned} N(\chi, t) = \mathscr{F}\{u(x,t)\} &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-j\chi x} u(x,t) \mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-j\chi x} u_0(x - at) \mathrm{d}x \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-j\chi x'} e^{-j\chi at} u_0(x') \mathrm{d}x' \\ &= e^{-j\chi at} N_0(\chi) . \end{aligned}$$

Consequently, at a given time t the Fourier transform at time $t + \Delta t$ can be computed via

$$N(\chi, t + \Delta t) = e^{-j\chi a\Delta t} N(\chi, t) . \tag{6.10}$$

Since the discretized solution is an approximation to the real solution, the amplification factor $g(\Delta x\xi)$ is an approximation of $e^{-j\chi a\Delta t}$. However, the difference scheme yields

$$\begin{aligned} g(\Delta x\xi) &= (1 - a\lambda) + a\lambda e^{-j\Delta x\xi} \\ &= |g(\Delta x\xi)| e^{-j\xi\Delta t\alpha(\Delta x\xi)} \end{aligned}$$

for some α that we are not going to compute explicitly. Now, $|g(\Delta x\xi)|$ is the damping factor of a wave with frequency ξ, while $\alpha(\Delta x\xi)$ is proportional to the phase speed. Comparing with (6.10), the term $a - \alpha(\Delta x\xi)$ is responsible for dispersion. Fig. 6.14 summarizes the observed effects.

6.5 Diffusive and Convective Problems

As mentioned in the introductory part of this chapter, both diffusive and convective contributions are often present. Since a diffusive contribution leads to parabolic PDEs and convective contribution to hyperbolic PDEs, reliable numerical results are harder to obtain in the combined case. This section emphasizes the additional problems and aims at finding a robust discretization scheme for the box integration method.

In the *drift-diffusion model* (1.23) to (1.25), we have both diffusive and convective contributions:

$$\boldsymbol{J}_n = \mathrm{q} n \mu_n \boldsymbol{E} + \mathrm{q} D_n \nabla n, \tag{6.11}$$

$$\boldsymbol{J}_p = \mathrm{q} p \mu_p \boldsymbol{E} - \mathrm{q} D_n \nabla n. \tag{6.12}$$

These expressions can now be plugged into the continuity equations

$$\nabla \cdot \boldsymbol{J}_n - \mathrm{q}\frac{\partial n}{\partial t} = +\mathrm{q}R, \tag{6.13}$$

$$\nabla \cdot \boldsymbol{J}_p + \mathrm{q}\frac{\partial p}{\partial t} = -\mathrm{q}R. \tag{6.14}$$

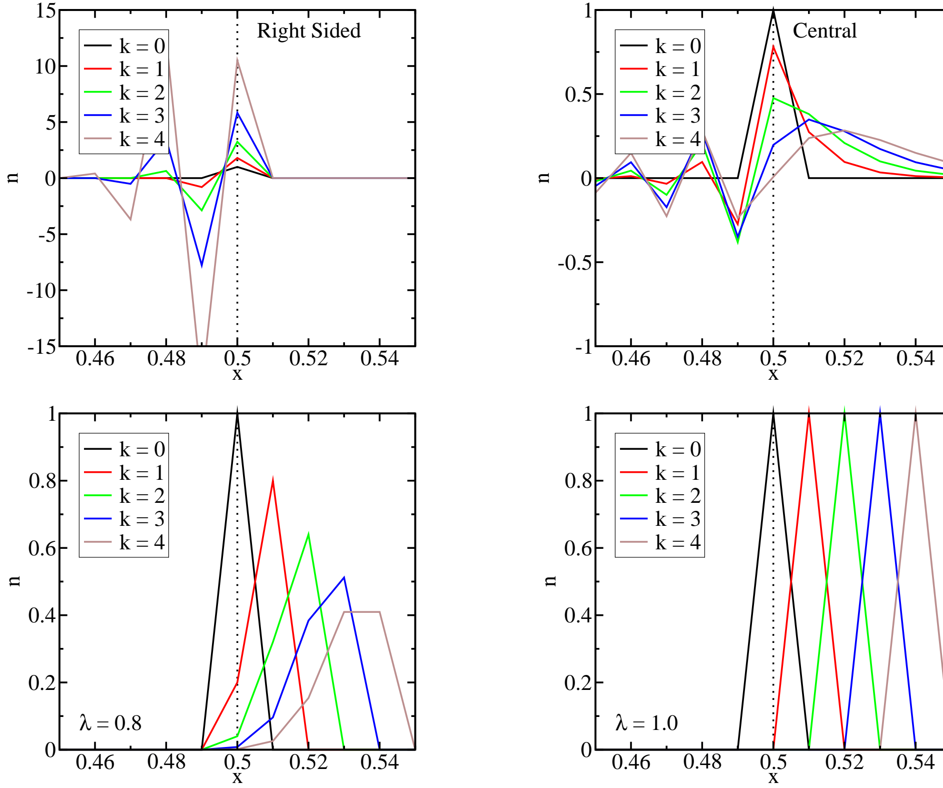


Figure 6.14: For a wave traveling to the right, neither right-sided differences (upper left) nor central differences (upper right) yield satisfactory results. Even left-sided differences may (lower left) or may not (lower right) show dispersion.

Substituting (6.11) into (6.13) leads to

$$\nabla \cdot (\mathrm{q} n \mu_n \boldsymbol{E} + \mathrm{q} D_n \nabla n) - \mathrm{q}\frac{\partial n}{\partial t} = +\mathrm{q}R.$$

Expanding terms, neglecting recombination, cancelling q and introducing the *thermal voltage* $V_T = k_\mathrm{B} T/\mathrm{q} = D_n/\mu_n$ finally results in

$$\frac{\partial n}{\partial t} - \mu_n \boldsymbol{E} \cdot \nabla n = \mu_n V_T \Delta n. \tag{6.15}$$

This is a second order parabolic partial differential equation. Depending on $\boldsymbol{E}$, the drift term can dominate in certain device regions so that one can expect a (more or less) dominant propagation in one specific direction with velocity $v = -\mu \boldsymbol{E}$. Diffusion takes place due to the term on the right hand side with coefficient $D_n = \mu_n V_T$.

For a box integration discretization, the integral form of (6.13) is needed. Therefore we integrate the equation over a box $\mathcal{V}$ and obtain

$$\int_{\mathcal{V}} \nabla \cdot \boldsymbol{J}_n \, \mathrm{d}V - \mathrm{q}\int_{\mathcal{V}} \frac{\partial n}{\partial t} \, \mathrm{d}V = \mathrm{q}\int_{\mathcal{V}} R \, \mathrm{d}V.$$

Using Gauss' integral theorem on the first integral, we find

$$\int_{\partial\mathcal{V}} \boldsymbol{J}_n \cdot \, \mathrm{d}\boldsymbol{A} - \mathrm{q}\int_{\mathcal{V}} \frac{\partial n}{\partial t} \, \mathrm{d}V = \mathrm{q}\int_{\mathcal{V}} R \, \mathrm{d}V,$$

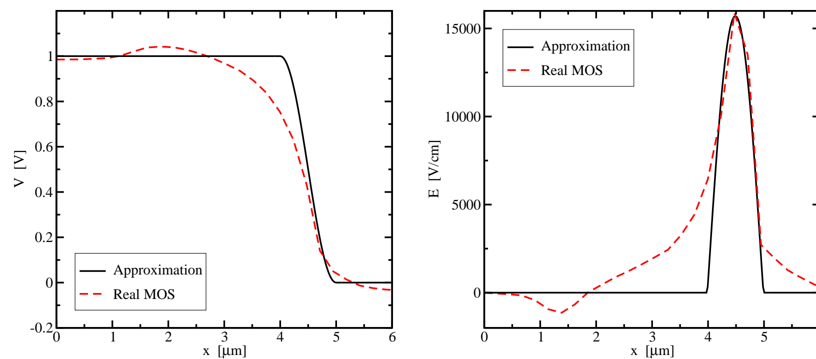


Figure 6.15: Potential (left) and electric field (right) of an approximate potential barrier (solid lines) designed to mimic the potential distribution inside a MOS transistor.

which can be approximated as

$$\sum_{j \in \mathcal{N}_i} J_{i,j} A_{i,j} - \mathrm{q} \int_{\mathcal{V}} \frac{\partial n}{\partial t} \, \mathrm{d}V = \mathrm{q} \int_{\mathcal{V}} R \, \mathrm{d}V .$$

A naive discretization for $J_{i,j}$ recalling $\boldsymbol{J}_n = \mathrm{q} n \mu_n \boldsymbol{E} + \mathrm{q} D_n \nabla n = \mathrm{q} \mu_n \left(-n \nabla V + V_T \nabla n \right)$ would be

$$J_{i,j} = \mathrm{q} \mu_n \left(-\frac{n_i + n_j}{2} \frac{V_j - V_i}{d_{i,j}} + V_T \frac{n_j - n_i}{d_{i,j}} \right) . \qquad (6.16)$$

Recall that in the box discretization the fluxes are required in the middle of a connection i, j (see Fig. 5.8), while the quantities $\boldsymbol{E}$ and ∇n can be expressed by central differences valid at this mid point. The carrier concentration is only available on the mesh points i and j. Naively, it has been assumed that

$$n_{i,j}^{\mathrm{mid}} \approx \frac{n_i + n_j}{2} .$$

The discretization (6.16) is used in the following example: A potential barrier is given (Fig. 6.15) and the resulting electron concentration has to be computed.

```
1
2    func window(min,max,x) return step(x-min)*step(max-x);
3
4    assign x[i=all]    = -12 + i*DX;
5    assign V[i=0..NX] = window(-12,   0, x[i]) * Vc +
6                        window(  0, 11 , x[i]) * Vc * sq(cos(x[i]/11*PI/2));
```

source_code/instability1.sg

The current continuity equation is implemented using the box integration method as

```
1
2    func Jsimple(ni,nj,Vi,Vj)
3      return q0 * mu * Vt / DX * (- (ni+nj)/2 * (Vj-Vi)/Vt + (nj-ni));
4
5    equ n[i=1..NX-1] -> (Jsimple(n[i],n[i-1],V[i],V[i-1]) +
6                         Jsimple(n[i],n[i+1],V[i],V[i+1]))*A = 0;
```

source_code/instability2.sg

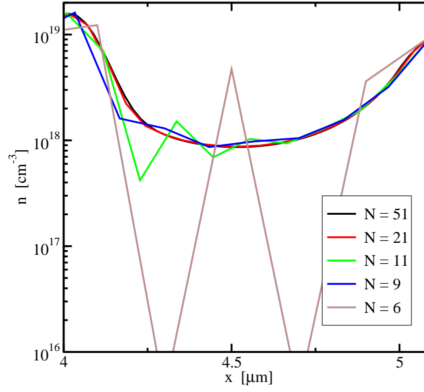


Figure 6.16: Large grid spacings show instabilities of the numerical solution.

The simulation results shown in Fig. 6.16 are poor: There is a strong dependence on the grid spacing. The scheme does not work for a large grid spacing, making simulations in two or three dimensions **unfeasible**[1].

We have to find out why the discretization (6.16) performs so badly. Let us therefore consider a one-dimensional example with constant electric field. For a conductor of width w and height h we have $V_i = \Delta x w h$, $A = wh$. Neglecting recombination, we have to solve

$$\sum_{j \in \mathcal{N}_i} J_{i,j} A = \mathrm{q} \frac{\partial n_i}{\partial t} V_i. \tag{6.17}$$

At node i, the discretization reads

$$(J_{i,i-1} + J_{i,i+1})\, A = \mathrm{q} \frac{\partial n}{\partial t} V_i \quad \Leftrightarrow \quad (J_{i,i-1} + J_{i,i+1}) = \mathrm{q} \frac{\partial n}{\partial t} \Delta x. \tag{6.18}$$

With the naive discretization (6.16) and a constant electric field we obtain

$$J_{i,i\pm1} = \mathrm{q}\mu_n \left(\frac{n_i + n_{i-1}}{2} (\pm E) + V_T \frac{n_{i\pm1} - n_i}{\Delta x} \right),$$

so all together

$$J_{i,i-1} + J_{i,i+1} = \mathrm{q}\mu_n \left(\frac{n_{i+1} - n_{i-1}}{2} E + V_T \frac{n_{i+1} - 2n_i + n_{i-1}}{\Delta x} \right).$$

Inserting into (6.18) with the time-derivative discretized with a forward Euler scheme and dividing by $\mu_n \Delta x$, (6.17) becomes

$$\frac{n_{i+1,k} - n_{i-1,k}}{2\Delta x} E + V_T \frac{n_{i+1,k} - 2n_{i,k} + n_{i-1,k}}{(\Delta x)^2} = \frac{n_{i,k+1} - n_{i,k}}{\mu_n \Delta t}. \tag{6.19}$$

Let us further analyze the discretization: With $a := -\mu_n E$, $b := \mu_n V_T$, $\eta := \Delta t/(\Delta x)^2$ and $\alpha := a\Delta x/(2b)$, equation (6.19) can be rearranged to

$$n_{i,k+1} = b\eta(1+\alpha) n_{i-1} + (1 - 2b\eta) n_i + b\eta(1-\alpha) n_{i+1}.$$

[1] **unfeasible** [ʌnˈfiː.zɪ.bl̩]: undurchführbar

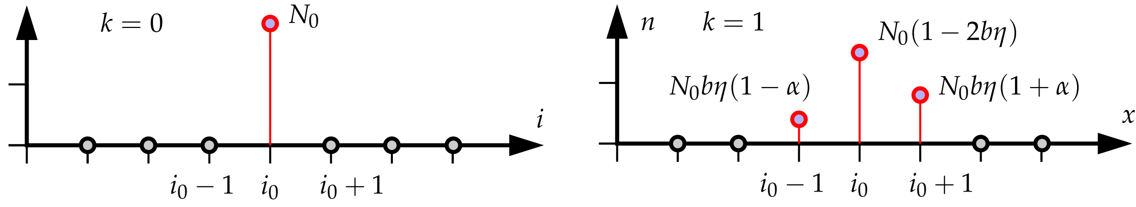


Figure 6.17: Motivation for the derivation of the Reynolds number.

We will assume $\alpha > 0$ and test with a Dirac pulse $n_{i,0} = N_0\delta_{i-i_0}$. The values for the first time step are

$$
\begin{aligned}
n_{i_0-1,1} &= 0 & &+0 & &+N_0 b\eta(1-\alpha),\\
n_{i_0\quad,1} &= 0 & &+N_0(1-2b\eta) & &+0,\\
n_{i_0+1,1} &= N_0 b\eta(1+\alpha) & &+0 & &+0.
\end{aligned}
$$

To obtain positive electron concentrations, $\alpha = a\Delta x/2b < 1$ must hold, or equivalently, $\Delta x < 2b/a = -2V_T/E$, with $E < 0$ because of $\alpha > 0$. We have just derived the *Reynolds-* or *Peclet-number* $|2V_T/E|$ of the cell. Keep in mind that it is *not* a stability criterion in the sense of (6.9), since there is *no time dependence*! Nevertheless, when Δx is larger than the Peclet-number, oscillations in the numerical solution will occur.

Please note that – just as the original PDE – the resulting discretization scheme is not symmetric: There is a net flow of electrons into one direction, driven by the electric field. Thus, for a fixed spatial point x, the concentration in upstream direction is more important than the concentration in downstream direction. Or to be (much more) pictorial: If you are sitting in a river and do not want to be hit by drifting wood, you have to look carefully for such dangerous wood in the opposite direction of the water flow.

Another problem of the naive discretization (6.16) is that the two contributions from diffusion current and drift current might be of similar magnitude and cancel each other. The main reason for the instability inherent to (6.16) is that, since the carrier concentration depends exponentially on the potential, a discretization of the electron concentration at the midpoint of an edge (i.e. the term $(n_i + n_j)/2$) is definitely a bad one. Historically, this instability prevented efficient application of numerical methods in the analysis of semiconductor devices until the pioneering work of Scharfetter and Gummel [**?**].

6.5.1 Scharfetter-Gummel Discretization

To overcome the issues mentioned above, an alternative discretization was suggested by *Scharfetter* and *Gummel*, which will be summarized in the following. Consider the projection of the current on the edge $\boldsymbol{e}_{i,j}$:

$$J_{n,i,j} = \boldsymbol{e}_{i,j} \cdot \boldsymbol{J}_n .$$

With the local coordinate $x_\mathrm{r} = (x - x_i)/d_{i,j}$ along edge $\boldsymbol{e}_{i,j}$ the current relation (6.11) becomes

$$\frac{J_{n,i,j}}{\mathrm{q}\mu_n} = nE_{i,j} + V_T \frac{\mathrm{d}n}{\mathrm{d}x_\mathrm{r}} d_{i,j} \,, \tag{6.20}$$

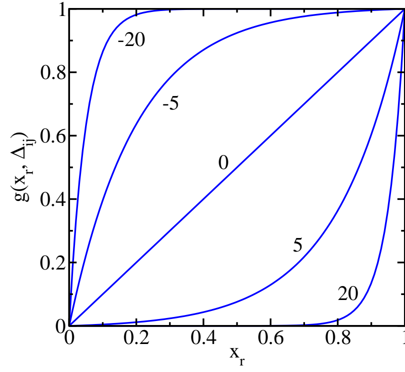


Figure 6.18: The growth function $g(x_\mathrm{r}, \Delta_{i,j})$ plotted for different values of $\Delta_{i,j}$.

where the additional term $d_{i,j}$ shows up because of the change of variables. We will now assume the three quantities $J_{n,i,j}$, $E_{i,j} = -\frac{\mathrm{d}V}{\mathrm{d}\xi}$ and μ to be constant along the edge $\boldsymbol{e}_{i,j}$, leading to an ordinary differential equation for the unknown function $n(x_\mathrm{r})$. The boundary conditions are $n(0) = n_i$ and $n(1) = n_j$. The attentive reader may object that the ODE is of first order only, still we have specified two boundary conditions. This is not **contradictory**[1], because there is an additional degree of freedom: We have chosen $J_{n,i,j}$ to be *constant*, but we can still choose its *value*!

The solution of (6.20) is

$$n(x, V) = (1 - \tilde{g}(x, V))n_i + \tilde{g}(x, V)n_j, \quad x_i \le x \le x_j \,, \tag{6.21}$$

with the growth function

$$\tilde{g}(x, V) = g(x_\mathrm{r}, \Delta_{i,j}) = \frac{1 - \exp(\Delta_{i,j} x_\mathrm{r})}{1 - \exp(\Delta_{i,j})} \,, \quad x_\mathrm{r} = \frac{x - x_i}{d_{i,j}} \,, \quad \Delta_{i,j} = \frac{V_j - V_i}{V_T} \,.$$

For the discretization, we need the values of n and $\mathrm{d}n/\mathrm{d}x_\mathrm{r}$ at the midpoint:

$$\frac{J_{n,i,j}}{\mathrm{q}\mu_n} = n|_\mathrm{midpoint} E_{i,j} + V_T \frac{n}{dx_\mathrm{r}}\bigg|_\mathrm{midpoint} \,. \tag{6.22}$$

They can be obtained from the solution of the ODE with $x_\mathrm{r} = 1/2$:

$$n|_\mathrm{midpoint} = \frac{n_i}{1 + \exp(-\Delta_{i,j}/2)} + \frac{n_j}{1 + \exp(\Delta_{i,j}/2)} \,, \tag{6.23}$$

$$\frac{\mathrm{d}n}{\mathrm{d}x_\mathrm{r}}\bigg|_\mathrm{midpoint} = \frac{\Delta_{i,j}/2}{\sinh(\Delta_{i,j}/2)} \frac{n_j - n_i}{2} \,. \tag{6.24}$$

After some formal rearrangement of (6.23) and (6.24), the *Scharfetter-Gummel discretization* for the current is finally obtained as

$$J_{n,i,j} = \frac{\mathrm{q}\mu_n V_T}{d_{i,j}} \left(n_j \mathcal{B}(\Delta_{i,j}) - n_i \mathcal{B}(-\Delta_{i,j})\right) \,, \tag{6.25}$$

$$J_{p,i,j} = -\frac{\mathrm{q}\mu_p V_T}{d_{i,j}} \left(p_j \mathcal{B}(-\Delta_{i,j}) - p_i \mathcal{B}(\Delta_{i,j})\right) \,, \tag{6.26}$$

[1] **contradictory** [kɔn.trəˈdɪk.tər.i]: widersprüchlich

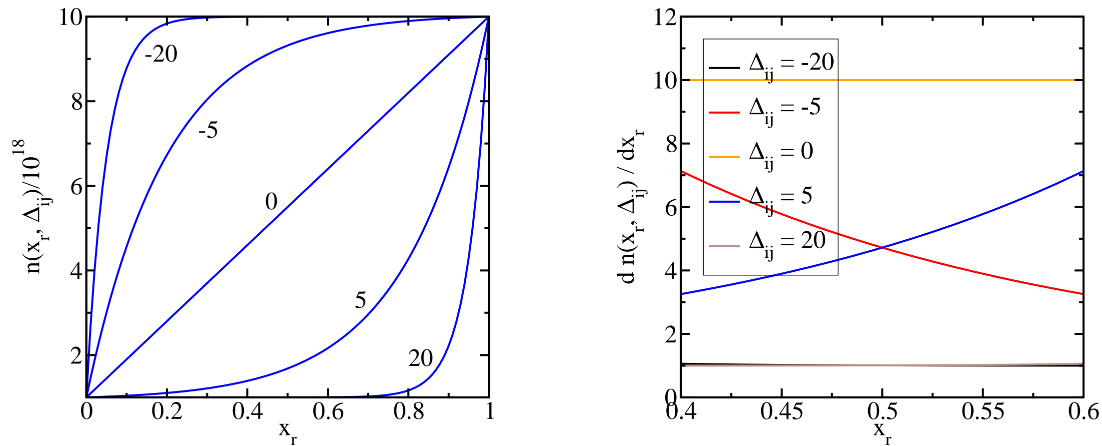


Figure 6.19: The concentration $n(x_\mathrm{r}, \Delta_{i,j})$ given by (6.23) and its derivative $\mathrm{d}n/\mathrm{d}x_\mathrm{r}$ given by (6.24).

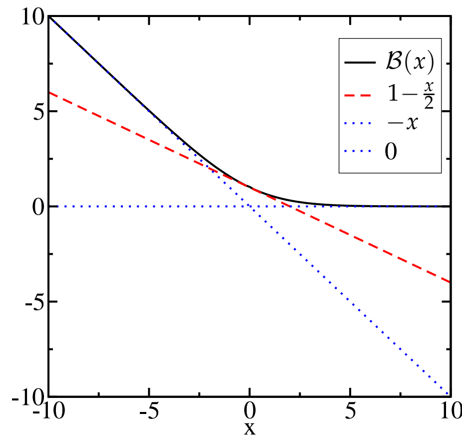


Figure 6.20: The Bernoulli function $\mathcal{B}(x)$.

with

$$\Delta_{i,j} = \frac{V_j - V_i}{V_T}, \quad \mathcal{B}(x) = \frac{x}{\mathrm{e}^x - 1}.$$

The *Bernoulli function* $\mathcal{B}(x)$ is depicted in Fig. 6.20. Its asymptotic behavior is

$$\mathcal{B}(x) \approx \begin{cases} -x, & x \ll 0, \\ 1 - \frac{x}{2}, & x \approx 0, \\ 0, & x \gg 0. \end{cases}$$

How does the Scharfetter-Gummel discretization behave for the large electric fields occuring in the previous example? Let us assume $E \gg 0$, then (6.25) used in (6.17) yields

$$J_{i,i-1} + J_{i,i+1} = \frac{\mathrm{q}\mu_n V_T}{\Delta x}\left(n_{i-1}\mathcal{B}(\Delta_{i,i-1}) - n_i\mathcal{B}(-\Delta_{i,i-1}) + n_{i+1}\mathcal{B}(\Delta_{i,i+1}) - n_i\mathcal{B}(-\Delta_{i,i+1})\right) .$$

Now,

$$\Delta_{i,i-1} = \frac{V_{i-1} - V_i}{V_T} = -E\frac{x_{i-1} - x_i}{V_T} = +E\frac{\Delta x}{V_T} \gg 0\,,$$
$$\Delta_{i,i+1} = \frac{V_{i+1} - V_i}{V_T} = -E\frac{x_{i+1} - x_i}{V_T} = -E\frac{\Delta x}{V_T} \ll 0\,,$$

so we get

$$J_{i,i-1} + J_{i,i+1} = \frac{q\mu_n V_T}{\Delta x}\left(-n_i(\Delta_{i,i-1}) + n_{i+1}(-\Delta_{i,i+1})\right)\,.$$

This finally gives for $E \gg 0$

$$\begin{aligned}
J_{i,i-1} + J_{i,i+1} &= \frac{q\mu_n V_T}{\Delta x}\left(-n_i\Delta_{i,i-1} - n_{i+1}\Delta_{i,i+1}\right) \\
&= \frac{q\mu_n V_T}{\Delta x}\left(-n_i E\frac{\Delta x}{V_T} + n_{i+1}E\frac{\Delta x}{V_T}\right) \\
&= q\mu_n E\left(n_{i+1} - n_i\right) \\
&\overset{!}{=} q\frac{\partial n_i}{\partial t}\Delta x.
\end{aligned}$$

With this,

$$\frac{\partial n_i}{\partial t} = \mu_n E\frac{n_{i+1} - n_i}{\Delta x}\,,$$

which is just the right-sided difference with the diffusive term neglected. Analogously, for $E \ll 0$, one finds the left-sided difference

$$\frac{\partial n_i}{\partial t} = \mu_n E\frac{n_i - n_{i-1}}{\Delta x}\,.$$

Comparing these results with (6.19), the diffusion term is gone and the central difference for the drift term is replaced by a right-sided difference and a left-sided difference, respectively.

However, these modifications of the simple discretization are not surprising: Since $|E| \gg 0$, the drift term dominates. Recalling the phenomena observed in the previous section about hyperbolic problems, Scharfetter-Gummel adapts itself to the dominant direction of convection and uses right-sided differences for electrons moving from the right to the left and vice versa.

Finally, let us consider the case of small electric fields: Rearranging the simple discretization (6.16) gives

$$\begin{aligned}
J_{i,j} &= q\mu_n\left(-\frac{n_i + n_j}{2}\frac{V_j - V_i}{\Delta x} + V_T\frac{n_j - n_i}{\Delta x}\right) \\
&= \frac{q\mu_n V_T}{\Delta x}\left(-\frac{n_i + n_j}{2}\frac{V_j - V_i}{V_T} + n_j - n_i\right) \\
&= \frac{q\mu_n V_T}{\Delta x}\left(-\frac{n_i + n_j}{2}\Delta_{i,j} + n_j - n_i\right) \\
&= \frac{q\mu_n V_T}{\Delta x}\left(n_j\left(1 - \frac{\Delta_{i,j}}{2}\right) - n_i\left(1 + \frac{\Delta_{i,j}}{2}\right)\right)
\end{aligned}$$

Comparison with (6.25) shows that the simple discretization can be obtained from the Scharfetter-Gummel discretization by linearizing the Bernoulli function around $E = 0$. This implies that the naive discretization is valid for **negligible**[1] fields only. Only under these cirumstances can the carrier concentration at the midpoint be expressed by their arithmetic average, see also Fig. 6.19.

[1] **negligible** [neg.lɪ.dʒə.bl̩]: vernachlässigbar

Chapter 7

Parameter Modeling

The basic semiconductor equations discussed in Chapter 1 contain several physical parameters (in particular: the mobilites μ_n and μ_p and the recombination term R) which have to be accurately modeled for the purpose of reliable device simulation. For accurate results, these parameters cannot assumed to be constant because they do not reflect the underlying physics to a sufficient degree of accuracy. Unfortunately this leads to an increased computational effort, so a good compromise between (physical) model accuracy and computational effort has to be found.

In many cases, sufficiently accurate and simple analytic models derived from theoretical considerations are not available and parameters such as the mobility have to be extracted from measurement results. One can then fit an analytic curve through the available data points and use this empirical model for the simulation.

7.1 Carrier Mobilities

Carrier mobilities in semiconductors are influenced by a variety of physical mechanisms. We will focus on the following dominant effects:

- Scattering at lattice atoms or defects: μ^{L}
- Scattering at charges or neutral impurities: μ^{I}
- Surface **roughness**[1] scattering: μ^{S}
- Increased scattering due to heating: μ^{F}

In order to obtain a **tractable**[2] model, scattering effects are frequently assumed to be independent. A common assumption is that the effective mobility is given by

$$\mu^{\mathrm{LISF}} = \mu^{\mathrm{LISF}}(\mu^{\mathrm{LIS}}(\mu^{\mathrm{LI}}(\mu^{\mathrm{L}}))) ,$$

which means that one first computes a mobility due to scattering at lattice atoms μ^{L} from a initially constant mobility μ^0. Then, a modified mobility from μ^{L} by taking a possible modification due to scattering at charges or neutral impurities into account. After that, surface roughness

[1] **roughness** [rʌf.nəs]: Rauheit, Unebenheit [2] **tractable** [træk.tə.bl̩]: handhabbar, lenkbar

Material	μ_n^0 (cm^2/Vs)	α_n (1)	μ_p^0 (cm^2/Vs)	α_p (1)
Si	1430	2.33	460	2.18
Ge	3800	1.66	1800	2.33
GaAs	8500	1	400	2.1

Table 7.1: Lattice scattering parameters for several semiconductors. Note that electron mobilities are higher than hole mobilities, therefore an n-MOS typically has a higher transconductance than a p-MOS (see Chapter 8).

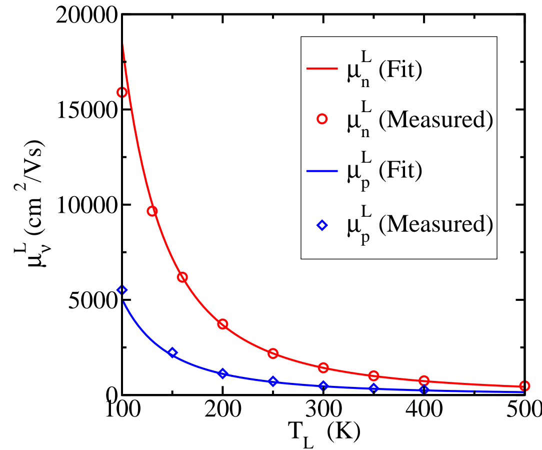


Figure 7.1: Comparison of measured and fitted mobilities due to lattice scattering in a homogeneous semiconductor.

scattering is considered to find the mobility μ^{LIS} including scattering at lattice atoms or charges and surface scattering into account. Finally, increased scattering due to heating yields a mobility μ^{LISF} that includes the physical effects mentionend above.

7.1.1 Lattice Scattering

At non-zero temperature the lattice atoms oscillate about their equilibrium sites. Even in pure, perfectly crystaline semiconductors, carriers are scattered because of their interaction with the vibrating lattice. This results in the lattice mobility μ^L which is a function of the lattice temperature. Theoretical models describing lattice scattering are complicated and results often unsatisfactory. However, for the purpose of simulation an empirical power law is conventionally assumed:

$$\mu_v^L = \mu_v^0 \cdot \left(\frac{T}{300\ \mathrm{K}}\right)^{-\alpha_v}, \quad v = n, p$$

This power law shows good agreement with experimental data (Fig. 7.1) in the temperature range between 200 K and 500 K (cf. Tab. 7.1), the typical operation temperature of semiconductor devices.

When comparing experimental data given in the literature, one finds that such data can show considerable scatter. For example, the parameters for the electron mobility a often in the (pretty

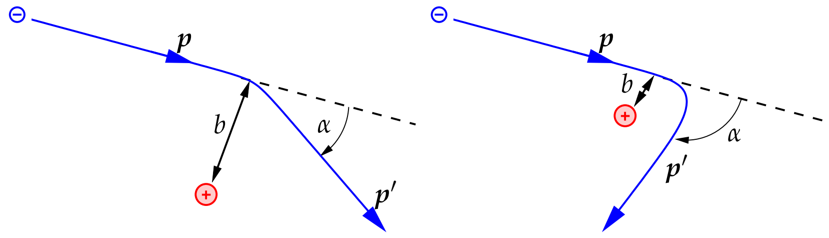


Figure 7.2: A carrier trajectory is influenced by ionized impurities. The potential (attraction or repulsion) "felt" by electrons is inversely proportional to the distance to the impurity, so trajectories far away from the impurity are less influenced (left) compared to those closer to the impurity (right).

large intervals) 1240 $\mathrm{cm}^2/\mathrm{Vs} < \mu_n^0 < 1600\ \mathrm{cm}^2/\mathrm{Vs}$ and $2.2 < \alpha_n < 2.6$ for Si. One reason is that each author uses their own measurement range, so that the available data puts some implicit weight on the measured interval. Furthermore, measurements and device fabrications are (at least to some extend) stochastic processes, resulting in many different sets of measurements.

7.1.2 Ionized Impurity Scattering

In semiconductor devices mobility reduction due to scattering by charged impurities (Fig. 7.2) is a dominant effect. The impact of lattice and impurity scattering has to be combined in some way to obtain an effective mobility. From a theoretical point of view, ionized impurity scattering is very difficult to model, because there is a strong dependence on doping concentration and mobile charges n and p that **screen**[1] the impurities.

An empirical model for the combined lattice and ionized impurity mobility was introduced by Caughey and Thomas [**?**]. To fit experimental data they used the following (empirical) expression:

$$\mu_v^{LI} = \mu_v^{\min} + \frac{\mu_v^L - \mu_v^{\min}}{1 + \left(\frac{N_I}{N_v^{\mathrm{ref}}}\right)^{\alpha_v}}, \quad v = n, p, \tag{7.1}$$

where

$$N_I = \sum_i |Z_i| N_i$$

is a sum over all charged impurities, Z_i is the charge state of the impurity. For (singly ionized) impurities (such as **boron**[2], **arsenic**[3] and **phosphorus**[4] in Si), $|Z_i| = 1$. The above expression has three free parameters that need to be calibrated to experimental data. Typical values for silicon at room temperature are $\mu_n^{\min} = 80\ \mathrm{cm}^2/\mathrm{Vs}$, $N_n^{\mathrm{ref}} = 1.12 \times 10^{17}\ \mathrm{cm}^{-3}$, $\alpha_n = 0.72$ for electrons and $\mu_p^{\min} = 45\ \mathrm{cm}^2/\mathrm{Vs}$, $N_p^{\mathrm{ref}} = 2.23 \times 10^{17}\ \mathrm{cm}^{-3}$, $\alpha_p = 0.72$ for holes.

7.1.3 Surface/Interface Scattering

The perfect periodicity of a semiconductor crystal is due to finite spatial dimensions broken by crystal surfaces. On interfaces between different materials, different lattice constants lead

[1] **to screen** [skriːn]: abschirmen, schützen, filtern [2] **boron** [ˈboʊrɔn]: Bor [3] **arsenic** [ˈɑː.sᵊn.ɪk]: Arsen [4] **phosphorus** [fɔs.fᵊr.əs]: Phosphor

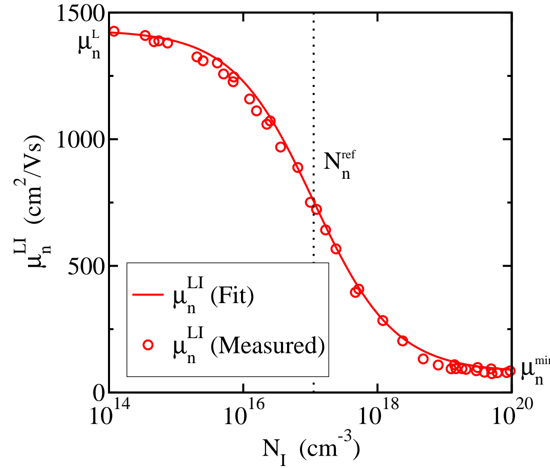


Figure 7.3: Comparison of measured and fitted mobilities μ_n^{LI}.

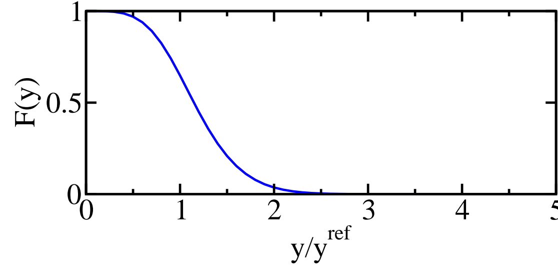


Figure 7.4: Depth dependence function $F(y)$ for modeling surface scattering in (7.2).

to unavoidable imperfections. This kind of imperfection is a severe concern if current flows primarily along the interface, as it is the case in a MOSFET. Typically, the mobility along a surface is considerably smaller than in the center of a crystal. There is no sharp transition from the high mobility in the crystal to the low mobility on the surface, instead, a smooth transition occurs.

An empirical expression for the description of such a depth dependence of the mobility suggested by Selberherr [?] is

$$\mu_\nu^{\mathrm{LIS}} = \frac{\mu_\nu^{\mathrm{ref}} + (\mu_\nu^{\mathrm{LI}} - \mu_\nu^{\mathrm{ref}})(1 - F(y))}{1 + F(y)\left(\frac{S_\nu}{S_\nu^{\mathrm{ref}}}\right)^{\gamma_\nu}}, \quad \nu = n, p. \tag{7.2}$$

The depth dependence $F(y)$ is given by

$$F(y) = \frac{2\exp\left(-\left(y/y^{\mathrm{ref}}\right)^2\right)}{1 + \exp\left(-2\left(y/y^{\mathrm{ref}}\right)^2\right)},$$

where the depth parameter y^{ref} is typically $2\ldots10$ nm. The *pressing forces* S_n and S_p are equal to the magnitude of the normal field strength at the interface ($E_\perp = \boldsymbol{E}\cdot\boldsymbol{n}$, $\boldsymbol{n}$ is the surface normal vector), if the carriers are attracted by it, otherwise they are zero. The remaining parameters are fit-parameters.

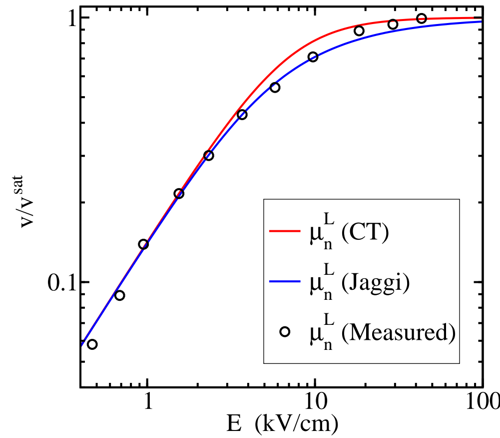


Figure 7.5: Comparison of carrier heating models. (CT = Caughey-Thomas)

7.1.4 Carrier Heating

The carrier energy consists of two contributions: one due to random thermal motion, which corresponds to the temperature, and a kinetic contribution given by the kinetic energy $\mathrm{m}v^2/2$. The average energy w per particle is thus given by

$$w = \frac{3\mathrm{k_B T}_n}{2} + \frac{\mathrm{m}v^2}{2}\,, \tag{7.3}$$

where T_n is the carrier temperature. Application of a field first increases the kinetic energy, while scattering eventually transfers the kinetic contribution to thermal energy by increasing the carrier temperature T_n.

At low electric fields the mobility is constant with respect to the field and the velocity-field relationship is consequently linear. The thermal velocity of electrons and holes is large compared to the movement due to the externally applied field and the lattice temperature matches the carrier temperature. At high fields, however, scattering events occur more frequently, converting a considerable amount of kinetic energy to thermal energy. As a result, the velocity-field relationship becomes non-linear (cf. Fig. 7.5). This effect is modeled in the framework of the drift-diffusion model by a field-dependent mobility.

Usually empirical mobility expressions are chosen where parameters are determined by fitting experimental data. One widely used expression is due to Caughey and Thomas [?]:

$$\mu_v^{\mathrm{LISF}}(E) = \frac{\mu_v^{\mathrm{LIS}}}{\left(1+\left(\frac{\mu_v^{\mathrm{LIS}}E}{v_v^{\mathrm{sat}}}\right)^{\beta_v}\right)^{1/\beta_v}}, \quad v = n, p \tag{7.4}$$

and another one is due to Jaggi [?, ?]:

$$\mu_v^{\mathrm{LISF}}(E) = \frac{2\mu_v^{\mathrm{LIS}}}{1+\left(1+\left(\frac{2\mu_v^{\mathrm{LIS}}E}{v_v^{\mathrm{sat}}}\right)^{\beta_v}\right)^{1/\beta_v}}, \quad v = n, p\,. \tag{7.5}$$

Both expressions contain as parameters the low-field mobility μ_v^{LIS} and the saturation velocity v_v^{sat}. These parameters determine the low-field and high-field limits of the $v(E)$ curve:

$$\lim_{E\to 0}\mu_v^{\mathrm{LISF}}(E) = \mu_v^{\mathrm{LIS}}, \quad \lim_{E\to\infty} v_v(E) = \lim_{E\to\infty}\mu_v^{\mathrm{LISF}}(E)\times E = v_v^{\mathrm{sat}} .$$

The following parameters can be used for silicon at room temperature: $v_n^{\mathrm{sat}} = 1\times 10^7$ cm/s, $\beta_n = 2$, $v_p^{\mathrm{sat}} = 8\times 10^6$ cm/s, $\beta_p = 1$. Note that for high electric fields, both (7.4) and (7.5) asymptotically reach $\mu_v^{\mathrm{LISF}}(E) \sim 1/E$, as expected.

7.2 Carrier Generation and Recombination

Generation-recombination phenomena determine many essential effects such as leakage current and device breakdown. For a semiconductor in thermal equilibrium there is a dynamic balance between generation and recombination processes, which leads to an equilibrium concentration n_0 for electrons and p_0 for holes:

$$n_0 = N_\mathrm{c}\exp\left(\frac{E_\mathrm{F}-E_\mathrm{c}}{k_\mathrm{B}T}\right) = n_i\exp\left(\frac{E_\mathrm{F}-E_\mathrm{i}}{k_\mathrm{B}T}\right), \tag{7.6}$$

$$p_0 = N_\mathrm{v}\exp\left(\frac{E_\mathrm{v}-E_\mathrm{F}}{k_\mathrm{B}T}\right) = n_i\exp\left(\frac{E_\mathrm{i}-E_\mathrm{F}}{k_\mathrm{B}T}\right). \tag{7.7}$$

For the product we get

$$n_0 p_0 = N_\mathrm{c}N_\mathrm{v}\exp\left(-\frac{E_\mathrm{c}-E_\mathrm{v}}{k_\mathrm{B}T}\right) = N_\mathrm{c}N_\mathrm{v}\exp\left(-\frac{E_\mathrm{g}}{k_\mathrm{B}T}\right) = n_i^2 , \tag{7.8}$$

with the intrinsic concentration

$$n_i = \sqrt{N_\mathrm{c}N_\mathrm{v}}\exp\left(-\frac{E_\mathrm{g}}{2k_\mathrm{B}T}\right) . \tag{7.9}$$

Note that these equations are valid for non-degenerate semiconductors only, because we assumed Boltzmann statistics in (7.6) and (7.7).

In the case of a deviation of the electron and hole concentrations from their equilibrium values the balance of generation and recombination rates is disturbed. In regions with **excess**[1] carriers ($np > n_\mathrm{i}^2$), recombination will **prevail**[2], whereas in the opposite case ($np < n_\mathrm{i}^2$) generation will dominate.

Generation-recombination can be induced by various physical mechanisms, such as absorption or emission of a photon, absorption or emission of a phonon (i.e. the quantum of lattice vibrations), three-particle transitions, and transitions assisted by recombination centers. The relative importance of these mechanisms depends on material properties and operating conditions.

Transition from the valence band to the conduction band and vice versa requires energy. If an electron is **elevated**[3] from the valence band to the conduction band (which corresponds to a hole going from the conduction band to the valence band), its energy increase approximately equals the band gap energy E_g. This amount of energy can be supplied by several sources:

- *Phonons*: Due to thermal lattice vibration, which is quantified by means of phonons, electrons can gain energy.

[1] **excess** [ek'ses]: überschüßig [2] **to prevail** [prɪ'veɪl]: überwiegen, vorherrschen [3] **to elevate** [el.ɪ.veɪt]: emporheben, erhöhen

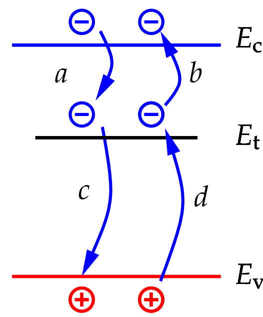


Figure 7.6: The complex process of phonon transition can be split up into four partial processes.

- *Photons*: When exposed to light, photons hit the semiconductor, each carrying an energy $\hbar\omega$. If this energy is larger or equal to the band gap, photons can be absorbed by electrons to pass the band gap.

- *Collisions*: The random motion of electrons inevitably leads to collisions among electrons. An electron from the conduction band with high energy (i.e. much higher than the conduction edge) can transfer enough energy to its collision partner in the valence band so that both electrons can finally be found in the conduction band.

The first two possibilities can also act as energy sinks: The first case usually leads to the generation of heat, while the second occurs in direct bandgap materials only and is exploited in lasers and LEDs. In the following we will look at each of these energy supplies separately.

7.2.1 Phonon Assisted Recombination and Generation

In indirect band gap semiconductors, such as silicon and germanium, it was found experimentally that generation-recombination phenomena occur primarily via trap centers. This mechanism is commonly termed Shockley-Read-Hall generation-recombination after the authors who established the theory [?, ?]. Indirect generation-recombination is a non-radiative process. In detail, four partial processes are involved (Fig. 7.6):

a) Electron capture: An electron from the conduction band is trapped by an unoccupied defect which becomes occupied.

b) Electron emission: An electron from an occupied trap moves to the conduction band. The trap becomes unoccupied.

c) Hole capture: An electron from an occupied trap moves to the valence band and neutralizes a hole. The trap becomes unoccupied.

d) Hole emission: An electron from the valence band is trapped by a defect, thus leaving a hole in the valence band and an occupied trap.

With k_a, k_b, k_c, k_d the corresponding reaction rates are given by

$$\begin{aligned} v_\mathrm{a} &= k_\mathrm{a} n N_\mathrm{t}^0 && \text{(electron capture)}\,, \\ v_\mathrm{b} &= k_\mathrm{b} N_\mathrm{t}^- && \text{(electron emission)}\,, \\ v_\mathrm{c} &= k_\mathrm{c} p N_\mathrm{t}^- && \text{(hole capture)}\,, \\ v_\mathrm{d} &= k_\mathrm{d} N_\mathrm{t}^0 && \text{(hole emission)}\,, \end{aligned}$$

where N_t^0 is the concentration of neutral traps and N_t^- is the concentration of occupied traps. The total trap concentration N_t is given by $N_\mathrm{t} = N_\mathrm{t}^0 + N_\mathrm{t}^-$. The fraction of occupied traps is given by $f_\mathrm{t} = N_\mathrm{t}^- / N_\mathrm{t}$, $1 - f_\mathrm{t} = N_\mathrm{t}^0 / N_\mathrm{t}$. The rate equation for v_a for electron capture states that the transmission rate is proportional to the number of carriers in the conduction band n and the number of neutral (free) traps N_t^0. For electron emission, the reaction rate v_b is proportional to the number of electrons N_t^- in the traps only, because most states in the conduction band are free (i.e. the distribution function f is close to zero, hence $1 - f$ is close to 1). A similar reasoning applies to holes.

In thermal equilibrium the principle of detailed balance holds, which implies in this particular case

$$v_\mathrm{a}^\mathrm{eq} = v_\mathrm{b}^\mathrm{eq}, \quad v_\mathrm{c}^\mathrm{eq} = v_\mathrm{d}^\mathrm{eq}\,.$$

Thus, the four rate constants are not independent:

$$k_\mathrm{b} = k_\mathrm{a} \underbrace{n_0 \frac{1 - f_\mathrm{t,0}}{f_\mathrm{t,0}}}_{=:n_1}\,, \tag{7.10}$$

$$k_\mathrm{d} = k_\mathrm{c} \underbrace{p_0 \frac{f_\mathrm{t,0}}{1 - f_\mathrm{t,0}}}_{=:p_1}\,, \tag{7.11}$$

with the auxiliary concentrations n_1 and p_1. Here, $f_\mathrm{t,0}$ denotes the fraction of occupied traps in thermal equilibrium. The meaning of the concentrations n_1 and p_1 will be discussed below.

With these definitions the net recombination rates become

$$\begin{aligned} R_n^\mathrm{SRH} &= v_\mathrm{a} - v_\mathrm{b} = k_\mathrm{a} N_\mathrm{t} \left(n(1 - f_\mathrm{t}) - n_1 f_\mathrm{t} \right) \\ R_p^\mathrm{SRH} &= v_\mathrm{c} - v_\mathrm{d} = k_\mathrm{c} N_\mathrm{t} \left(p f_\mathrm{t} - p_1 (1 - f_\mathrm{t}) \right)\,. \end{aligned}$$

In the general dynamic case, R_n^SRH is not necessarily equal to R_p^SRH. This is modeled by adding a conservation equation to the basic semiconductor equations:

$$\frac{\partial N_\mathrm{t}^-}{\partial \mathrm{t}} = R_n^\mathrm{SRH} - R_p^\mathrm{SRH} \tag{7.12}$$

that has to be solved in the whole domain. Under steady-state conditions, the time derivative vanishes and the net recombination rates of electrons and holes are equal. This allows the trap occupancy function to be calculated explicitly:

$$f_\mathrm{t} = \frac{k_\mathrm{a} n + k_\mathrm{c} p_1}{k_\mathrm{a}(n + n_1) + k_\mathrm{c}(p + p_1)}\,.$$

The net recombination rate after Shockley, Read [?] and Hall [?] thus evaluates to

$$\begin{aligned} R^{\mathrm{SRH}} &= N_{\mathrm{t}} k_{\mathrm{a}} k_{\mathrm{c}} \frac{np - n_1 p_1}{k_{\mathrm{a}}(n + n_1) + k_{\mathrm{c}}(p + p_1)} \\ &= N_{\mathrm{t}} k_{\mathrm{a}} k_{\mathrm{c}} \frac{np - n_{\mathrm{i}}^2}{k_{\mathrm{a}}(n + n_1) + k_{\mathrm{c}}(p + p_1)} \,. \end{aligned}$$

Introducing the carrier lifetimes $\tau_{\mathrm{n}}^{-1} = k_{\mathrm{a}} N_{\mathrm{t}}$ and $\tau_{\mathrm{p}}^{-1} = k_{\mathrm{c}} N_{\mathrm{t}}$ one finally obtains

$$\boxed{R^{\mathrm{SRH}} = \frac{np - n_{\mathrm{i}}^2}{\tau_{\mathrm{p}}(n + n_1) + \tau_{\mathrm{n}}(p + p_1)} \,.} \tag{7.13}$$

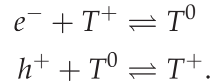
In case $np > n_{\mathrm{i}}^2$ we have $R > 0$, which means that recombination takes place until $np = n_{\mathrm{i}}^2$. On the contrary, if $np < n_{\mathrm{i}}^2$, generation is dominant, such that carrier concentrations increase until again $np = n_{\mathrm{i}}^2$. The carrier lifetimes τ_{n} and τ_{p} control the transient response of the material in non-equilibrium situations. The shorter the carrier lifetimes are, the larger is $|R|$, thus the material approaches its equilibrium more rapidly.

Traps are defects with an energy level E_{t} and a concentration N_{t}. The characteristic parameters describing the interaction of carriers and trap centers are the capture cross sections σ_{n} and σ_{p} for electrons and holes, respectively and the rate constants and the carrier lifetimes are conventionally expressed as

$$k_{\mathrm{a}} = \sigma_{\mathrm{n}} v_{\mathrm{th}}^{\mathrm{n}} \,, \quad \tau_{\mathrm{b}}^{-1} = \sigma_{\mathrm{n}} v_{\mathrm{th}}^{\mathrm{n}} N_{\mathrm{t}} \,, \quad k_{\mathrm{c}} = \sigma_{\mathrm{p}} v_{\mathrm{th}}^{\mathrm{p}} \,, \quad \tau_{\mathrm{d}}^{-1} = \sigma_{\mathrm{p}} v_{\mathrm{th}}^{\mathrm{p}} N_{\mathrm{t}} \,,$$

with the thermal velocities $v_{\mathrm{th}}^{\mathrm{n}}$ and $v_{\mathrm{th}}^{\mathrm{p}}$ given by (2.14) for electrons and holes respectively.

So far, the model was derived for *acceptor-like* traps. These can exist in a neutral or a negatively charged state $(N_{\mathrm{t}}^0, N_{\mathrm{t}}^-)$. *Donor-like* traps have a neutral and a positively charged state $(N_{\mathrm{t}}^0, N_{\mathrm{t}}^+)$. For the latter type of traps the reaction rates are of the form



$$\begin{aligned} e^- + T^+ &\rightleftharpoons T^0 \\ h^+ + T^0 &\rightleftharpoons T^+. \end{aligned}$$

Following an analogous derivation as the one leading to (7.13) it can be shown that the net recombination rate for donor-like traps is the same as for acceptor-like traps.

We still have to interpret the auxiliary concentrations n_1 and p_1. In equilibrium, the trap occupancy function $f_{\mathrm{t},0}$ is determined by the Fermi-Dirac distribution function:

$$f_{\mathrm{t},0} = \left(1 + \exp\left(\frac{E_{\mathrm{t}} - E_{\mathrm{F}}}{k_{\mathrm{B}} T}\right)\right)^{-1} \,.$$

It has to be emphasized that one must not use the Boltzmann distribution at this point, because the traps are located in the band gap, where the Boltzmann distribution is not valid anymore. We find

$$\frac{1 - f_{\mathrm{t},0}}{f_{\mathrm{t},0}} = \exp\left(\frac{E_{\mathrm{t}} - E_{\mathrm{F}}}{k_{\mathrm{B}} T}\right) \,.$$

With this we find that for the equilibrium carrier concentrations (cf. (7.6) and (7.7)), the auxiliary concentrations n_1 and p_1 introduced in (7.10) and (7.11) can be written as

$$n_1 = n_0 \frac{1 - f_{\mathrm{t},0}}{f_{\mathrm{t},0}} = n_\mathrm{i} \exp\left(\frac{E_\mathrm{t} - E_\mathrm{i}}{k_\mathrm{B} T}\right) , \tag{7.14}$$

$$p_1 = p_0 \frac{f_{\mathrm{t},0}}{1 - f_{\mathrm{t},0}} = n_\mathrm{i} \exp\left(\frac{E_\mathrm{i} - E_\mathrm{t}}{k_\mathrm{B} T}\right) . \tag{7.15}$$

Therefore, n_1 and p_1 are those carrier concentrations that correspond to a Fermi level equal to the trap level ($E_\mathrm{F} = E_\mathrm{t}$).

A trap acts most effectively as recombination center if its energy level E_t lies in the middle of the band gap ($E_\mathrm{t} \approx E_\mathrm{i}$), because in this case both n_1 and p_1 in (7.14) and (7.15) take moderate values and the recombination rate in (7.13) is large. These traps are termed *deep traps*. If the dependence of the helper concentrations n_1 and p_1 on the energy difference in (7.14) and (7.15) were linear, the trap location would be irrelevant, but since there is an exponential dependence, the convex function $n_1 + p_1$ has a unique minimum at $E_\mathrm{t} = E_\mathrm{i}$.

Dopants are impurities with energy levels close to the band edges and are termed *shallow traps*. For shallow traps $|E_\mathrm{t} - E_\mathrm{i}| \gg k_\mathrm{B} T$. Thus either n_1 or p_1 takes a large value because of the exponential dependence, which results in a negligible net recombination rate according to (7.13). Therefore, dopants are very ineffective recombination centers.

Let us consider the special case of the space charge region of a *pn*-diode under reverse bias. We can make the assumptions $n \ll n_\mathrm{i}$ and $p \ll n_\mathrm{i}$ as well as $\tau_\mathrm{n} = \tau_\mathrm{p} = \tau$. The *Shockley-Read-Hall recombination* is then computed as

$$\begin{aligned} R^{\mathrm{SRH}} &= \frac{np - n_\mathrm{i}^2}{\tau_\mathrm{p}(n + n_1) + \tau_\mathrm{n}(p + p_1)} \\ &\approx -\frac{n_\mathrm{i}^2}{\tau_\mathrm{p} n_1 + \tau_\mathrm{n} p_1} \\ &= -\frac{n_\mathrm{i}}{\tau} \frac{1}{\exp\left(\frac{E_\mathrm{t} - E_\mathrm{i}}{k_\mathrm{B} T}\right) + \exp\left(\frac{E_\mathrm{i} - E_\mathrm{t}}{k_\mathrm{B} T}\right)} \\ &= -\frac{n_\mathrm{i}}{\tau} \frac{1}{2 \cosh\left(\frac{E_\mathrm{t} - E_\mathrm{i}}{k_\mathrm{B} T}\right)} . \end{aligned}$$

Since $\cosh(x) \geq 1$ for all x, the maximum recombination rate is

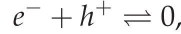
$$R^{\mathrm{SRH}}\Big|_{\max} = -\frac{n_\mathrm{i}}{2\tau}$$

for $E_\mathrm{t} = E_\mathrm{i}$. In general, the carrier lifetimes differ ($\tau_\mathrm{n} \neq \tau_\mathrm{p}$) and this energy is slightly shifted but still close to E_i under practical circumstances.

7.2.2 Photon Transition

The mechanism of direct generation/recombination can also be accompanied by photon emission or absorption. Due to the small photon momentum direct band-to-band transitions are only important in direct gap semiconductors such as GaAs. In silicon and germanium the direct generation-recombination mechanism is insignificant.

According to the reaction



$$e^- + h^+ \rightleftharpoons 0,$$

two partial processes are involved:

a) Electron-hole recombination: An electron moves from the conduction band to the valence band where it neutralizes a hole. A photon with energy of approximately the band gap energy is emitted (radiative recombination).

b) Electron-hole pair generation: An electron from the valence band absorbs a photon, whose energy is larger than the band gap energy, and moves to the conduction band. A hole is left behind in the valence band (optical generation).

With the rate constants $k_\mathrm{a}^\mathrm{opt}$ and $k_\mathrm{b}^\mathrm{opt}$ the rate equations become (note that in case a) both an electron and a hole are required)

$$v_\mathrm{a} = k_\mathrm{a}^\mathrm{opt}(T)np \,,$$
$$v_\mathrm{b} = k_\mathrm{b}^\mathrm{opt}(T) \,.$$

These two rates must be equal in thermal equilibrium:

$$v_\mathrm{a,0} = v_\mathrm{b,0} \Rightarrow k_\mathrm{a}^\mathrm{opt} n_\mathrm{i}^2 = k_\mathrm{b}^\mathrm{opt} \,.$$

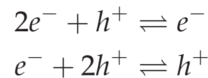
Therefore, the net recombination rate evaluates to

$$R^\mathrm{opt} = v_\mathrm{a,0} - v_\mathrm{b,0} = k_\mathrm{a}^\mathrm{opt}(np - n_\mathrm{i}^2) \,.$$

Note, again, the term $np - n_\mathrm{i}^2$, which expresses the tendency to finally reach an equilibrium, where $np = n_\mathrm{i}^2$ holds.

7.2.3 Auger Generation-Recombination

In this process three particles are involved, but only two move from one band to another. The third one, which provides or receives the excess energy, moves to another energy level within the same band, where it ultimately loses (in case of recombination) its energy to thermal vibrations. We consider the direct band-to-band Auger process which is sometimes also referred to as *phonon-assisted* Auger process. Four partial processes are involved:



$$2e^- + h^+ \rightleftharpoons e^-$$
$$e^- + 2h^+ \rightleftharpoons h^+$$

a) Electron capture: An electron from the conduction band moves to the valence band. The excess energy is transmitted to another conduction electron. In the valence band the electron neutralizes a hole.

b) Electron emission: A valence electron moves to the conduction band by consuming the energy of a high energetic electron in the conduction band. A hole is left behind in the valence band.

c) Hole capture: An electron from the conduction band moves to the valence band. The excess energy is transmitted to another hole. In the valence band the electron neutralizes a hole.

d) Hole emission: A valence electron moves to the conduction band by consuming the energy of a high energetic hole in the valence band. A hole is left behind in the valence band.

With the rate constants c_n, e_n, c_p and e_p the reaction rates read

$$v_\mathrm{a} = c_\mathrm{n} n^2 p \qquad \text{(electron capture)}\,, \tag{7.16}$$

$$v_\mathrm{b} = e_\mathrm{n} n \qquad \text{(electron emission)}\,, \tag{7.17}$$

$$v_\mathrm{c} = c_\mathrm{p} n p^2 \qquad \text{(hole capture)}\,, \tag{7.18}$$

$$v_\mathrm{d} = e_\mathrm{p} p \qquad \text{(hole emission)}\,. \tag{7.19}$$

The term $n^2 p$ in (7.16) is due to the need for two electrons and one hole from the conduction and valence band respectively. Even though there are two electrons involved in the electron emission process in (7.17), only one electron from the conduction band (recall that n represents the number of electrons in the conduction band only) participates. A similar reasoning applies to (7.18) and (7.19).

In thermal equilibrium the principle of detailed balance holds, which implies

$$v_{\mathrm{a},0} = v_{\mathrm{b},0} \Rightarrow c_\mathrm{n} n_\mathrm{i}^2 = e_\mathrm{n}\,,$$
$$v_{\mathrm{c},0} = v_{\mathrm{d},0} \Rightarrow c_\mathrm{p} n_\mathrm{i}^2 = e_\mathrm{p}\,.$$

The remaining rate constants c_n and c_p are referred to as *Auger coefficients*. The net recombination rate for the Auger process becomes

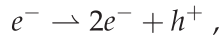
$$R^\mathrm{AU} = v_\mathrm{a} - v_\mathrm{b} + v_\mathrm{c} - v_\mathrm{d} = (c_\mathrm{n} n + c_\mathrm{p} p)(np - n_\mathrm{i}^2)\,.$$

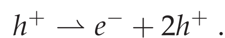
Note again the term $np - n_\mathrm{i}^2$, which expresses the tendency to finally reach an equilibrium. The Auger coefficients vary slightly with temperature between 77 K and 400 K. Typical values for silicon at room temperature are: $c_\mathrm{n} = 2.9 \times 10^{-31}\ \mathrm{cm^6/s}$ and $c_\mathrm{p} = 9.9 \times 10^{-32}\ \mathrm{cm^6/s}$.

7.2.4 Impact Ionization

Impact ionization is a pure generation process in which a high energetic carrier generates an electron-hole pair. From a microscopic point of view there is no difference between impact ionization and Auger generation. The only difference lies in the energy sources of the two processes: the Auger generation rate was evaluated by making use of the principle of detailed balance, which holds in equilibrium. Impact ionization, however, is a typical non-equilibrium process requiring large electric fields.

Two partial processes have to be considered:





a) Electron emission: A valence electron moves to the conduction band by consuming the energy of a high energetic electron in the conduction band. A hole is left behind in the valence band.

b) Hole emission: A valence electron moves to the conduction band by consuming the energy of a high energetic hole in the valence band. A hole is left behind in the valence band.

Obviously, these two partial processes are identical to the Auger partial processes b) and d). However, for impact ionization the reaction rates are modeled differently in the framework of the drift-diffusion model:

$$v_a = \alpha_n \frac{|\boldsymbol{J}_n|}{q} ,$$
$$v_b = \alpha_p \frac{|\boldsymbol{J}_p|}{q} .$$

Here, α_n and α_p are the ionization coefficients for electrons and holes, respectively. They are defined as the reciprocals of the average distances traveled by the carriers between consecutive ionization events. For instance, over a distance $1/\alpha_n$ an electron generates on average one electron-hole pair. The total generation rate can thus be written as

$$G^{II} = v_a + v_b = \alpha_n \frac{|\boldsymbol{J}_n|}{q} + \alpha_p \frac{|\boldsymbol{J}_p|}{q} . \tag{7.20}$$

Generation due to impact ionization is proportional to the current densities, while Auger generation is proportional to the carrier concentrations (cf. (7.17) and (7.19)). This means that Auger generation may take place in regions with a high concentration of mobile carriers with negligible current flow, whereas impact ionization requires non-negligible current flow as **prerequisite**[1].

Both theoretical and experimental investigations indicate an exponential dependence of the ionization coefficients on the electric field:

$$\alpha_n = A_n \exp\left(-\left(B_n/E\right)^{\beta_n}\right) , \quad \alpha_p = A_p \exp\left(-\left(B_p/E\right)^{\beta_p}\right) .$$

$E = \boldsymbol{E} \cdot \boldsymbol{J}/|\boldsymbol{J}|$ is the field component in direction of current flow. By interpreting a lot of experimental data, Chynoweth found the exponents β_n and β_p to be unity. Theoretical investigations by Shockley predicted the same exponents. A different treatment by Wolff predicts the exponents to be two. Fig. 7.2.4 shows the good agreement of theoretical and measurement results.

Practically, β_n and β_p are chosen between one and two in order to obtain good agreement with experimental data. For silicon at room temperature typical parameters are $A_n = 7.03 \times 10^5$ cm^{-1}, $B_n = 1.231 \times 10^6$ V/cm, $\beta_n = 1$, $A_p = 6.71 \times 10^5$ cm^{-1}, $B_p = 1.693 \times 10^6$ V/cm and $\beta_p = 1$.

[1] **prerequisite** [priːˈrek.wɪ.zɪt]: Voraussetzung, Bedingung

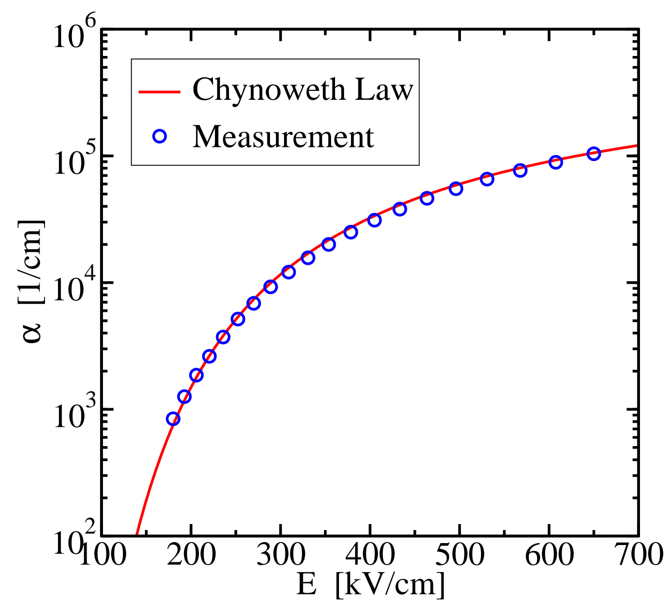


Figure 7.7: Verification of lattice scattering parameters for a homogeneous (bulk) sample.

Chapter 8

Devices in Detail

While the 1950s and 1960s were dominated by bipolar transistor technology, the 1970s saw Metal-Oxide-Semiconductor (MOS) technology begin to overtake bipolar technology in terms of functional complexity and level of integration. Geometrical scaling in MOS technology made it possible to increase chip complexity and to challenge bipolar circuits in the area of high-speed applications. In the 1980s, complementary MOS (CMOS) technology began to replace bipolar technologies such as transistor-transistor-logic (TTL) in many system applications [**?**, ?].

This chapter starts with the analytical description of a diode, after that the bipolar transistor is discussed. The main motivation is to highlight the approximations necessary to obtain closed form results. These approximations are not required in a numerical scheme and the results will be compared. The second half of this chapter deals with the MOS capacitor, leading to the MOS field effect transistor (MOSFET). Finally, scaling issues in CMOS technology will be discussed.

8.1 Analytical Diode Model

In 1944, Shockley theoretically investigated the junction of two semiconductor segments with different doping. When a p- and an n-doped region are joined, a *space-charge* will build up, resulting from diffusion of carriers from one material into the other caused by the difference of Fermi levels, leaving behind their ionized atoms they originate from. This results in an electric field that opposes the diffusion and drives the carriers back to their origin, until an equilibrium is reached (Fig. 8.1). In the space charge region, carriers of opposite charge recombine, so that the region is **depleted**[1] of carries, therefore this region is also called *depletion region*.

Inside the space charge region, we assume $\rho = \mathrm{q}C$, while $\rho = 0$ in the outside. The width of transition between these two regions is assumed to be negligible for analytical purposes. This *depletion approximation* is justified if the width of the space charge region is much larger than the Debye length $L_\mathrm{D} = \sqrt{\varepsilon V_\mathrm{T}/\mathrm{q}N}$. With this we can immediately solve Poisson's equation, because it is now decoupled from the carrier concentrations: Outside the depletion region, $\rho = 0$ holds and thus E is constant. Inside the space-charge region, $\rho = \mathrm{q}C$ holds, so

$$\frac{\mathrm{d}E}{\mathrm{d}x} = \frac{\mathrm{q}}{\varepsilon}(p - n + C) \approx \frac{\mathrm{q}C}{\varepsilon}\,. \tag{8.1}$$

[1] **to deplete** [dɪˈpliːt]: verringern, aufbrauchen, verarmen

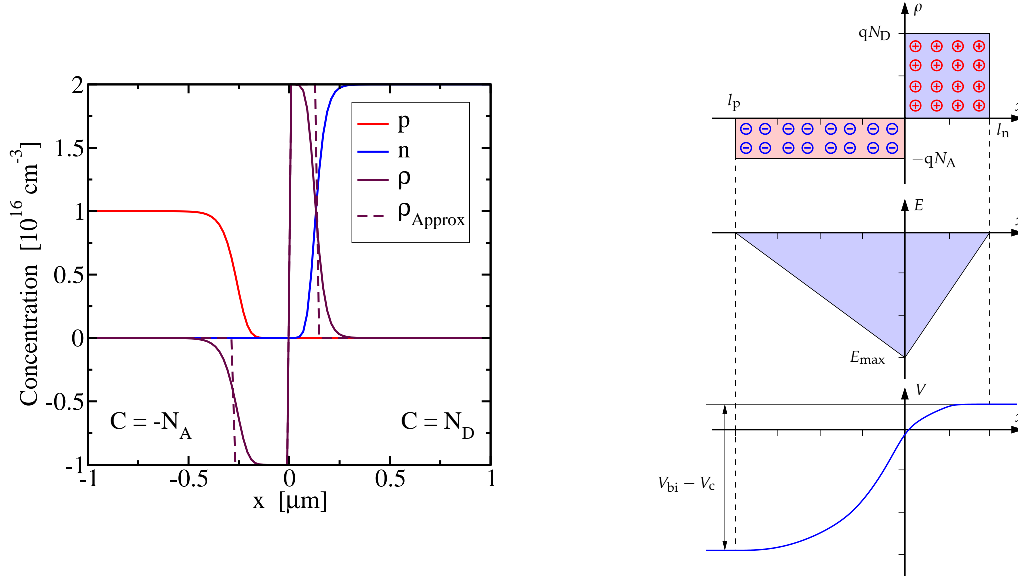


Figure 8.1: Concentrations of electrons and holes around the *pn*-junction of a diode (left). The depletion approximation enables an analytical calculation of the electric field and the potential.

This gives

$$E(x) = \begin{cases} -\frac{q}{\varepsilon} N_A (x + l_p) , & -l_p \le x \le 0 , \\ \frac{q}{\varepsilon} (N_D x - N_A l_p) , & 0 \le x \le l_n . \end{cases}$$

The maximum electric field is found at the interface:

$$E_{max} = E(0) = -\frac{q}{\varepsilon} N_A l_p,$$

so the width of the negatively charged space-charge region l_p is

$$l_p = -\frac{\varepsilon}{q} \frac{E_{max}}{N_A} .$$

Since $E = -dV/dx$, the potential is found as (left contact at potential V_c)

$$V(x) = \begin{cases} V_c & x \le -l_p , \\ V_c + \frac{q}{2\varepsilon} N_A \left(x - l_p\right)^2 , & -l_p \le x \le 0 , \\ V_c + \frac{q}{2\varepsilon} \left(N_A l_p^2 - N_D x^2 + N_A x l_p\right) , & 0 \le x \le l_n . \end{cases}$$

The total charge must vanish due to charge neutrality, therefore

$$N_A l_p = N_D l_n \quad \Rightarrow \quad l_n = -\frac{\varepsilon}{q} \frac{E_{max}}{N_D} .$$

At $x = l_n$, we find the potential

$$\begin{aligned} V(l_n) &= V_c + \frac{q}{2\varepsilon} \left(N_A l_p^2 - N_D l_n^2 + N_A l_n l_p\right) \\ &= V_c + E_{max}^2 \frac{\varepsilon}{2q} \left(\frac{1}{N_A} + \frac{1}{N_D}\right) \\ &= \psi_{bi} . \end{aligned}$$

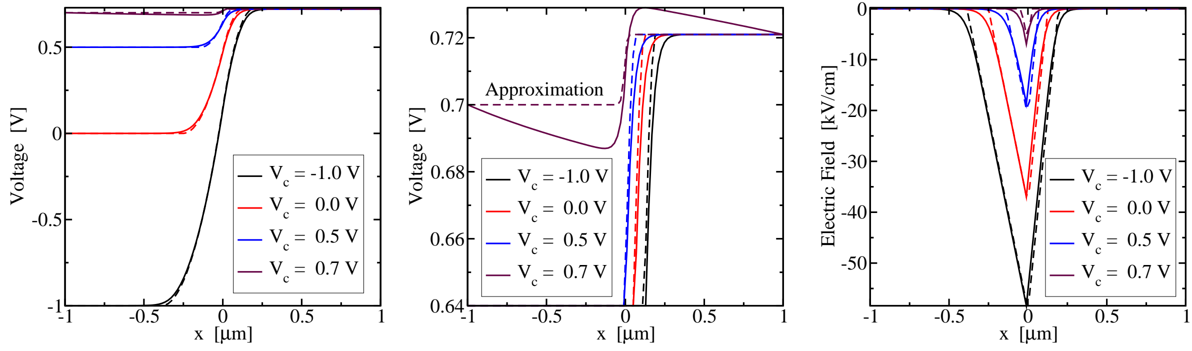


Figure 8.2: Comparison of the electrostatic potential and the electric field obtained by numerical simulation with the (analytical) depletion approximation. Results are poor for $V_\mathrm{c} \gtrsim \psi_\mathrm{bi}$ and there is a considerable voltage drop outside the space charge region.

At this point, the built-in potential from Section 2.3 turns up again: If two semiconductors with different Fermi level are joined, we must not forget about the potential difference at the contacts. We find

$$\psi_\mathrm{bi} - V_\mathrm{c} = E_\mathrm{max}^2 \frac{\varepsilon}{2\mathrm{q}} \left(\frac{1}{N_\mathrm{A}} + \frac{1}{N_\mathrm{D}} \right) .$$

Conversely, the space charge region is defined via

$$E_\mathrm{max} = -\sqrt{\frac{2\mathrm{q}}{\varepsilon} \frac{\psi_\mathrm{bi} - V_\mathrm{c}}{\frac{1}{N_\mathrm{A}} + \frac{1}{N_\mathrm{D}}}} ,$$

which yields

$$l = \sqrt{\frac{2\varepsilon}{\mathrm{q}} (\psi_\mathrm{bi} - V_\mathrm{c}) (\frac{1}{N_\mathrm{A}} + \frac{1}{N_\mathrm{D}})} , \quad l_\mathrm{p} = l N_\mathrm{D}/(N_\mathrm{A} + N_\mathrm{D}) , \quad l_\mathrm{n} = l N_\mathrm{A}/(N_\mathrm{A} + N_\mathrm{D}) . \tag{8.2}$$

The expression in the square root has to be non-negative, so we can immediately see that this model fails for the case $V_\mathrm{c} > \psi_\mathrm{bi}$! A more physical explanation is that as V_c approaches ψ_bi, the opposing electric field becomes smaller and smaller. Therefore, diffusion acting on the carriers is only partially compensated by the force resulting from the junction potential variation. Thus, if $V_\mathrm{c} \approx \psi_\mathrm{bi}$, there is no clearly defined space-charge region anymore, so that our model assumptions cannot be justified anymore.

For $V_\mathrm{c} \ll \psi_\mathrm{bi}$, the simulation results for the potential and the electric field (cf. Fig. 8.3) are in good agreement with our analytical results. Let us consider the equilibrium case first: In the p region, outside the space charge region, there holds

$$p_{p0} = N_\mathrm{A}, \quad n_{p0} = n_\mathrm{i}^2 / N_\mathrm{A} ,$$

while in the n region we have

$$n_{n0} = N_\mathrm{D}, \quad p_{n0} = n_\mathrm{i}^2 / N_\mathrm{D} .$$

Away from thermal equilibrium we have according to Chapter 2

$$n = n_\mathrm{i} \exp\left(\frac{E_{\mathrm{F}n} - E_\mathrm{i}}{k_\mathrm{B} T_\mathrm{L}} \right) , \qquad p = n_\mathrm{i} \exp\left(\frac{E_\mathrm{i} - E_{\mathrm{F}p}}{k_\mathrm{B} T_\mathrm{L}} \right) ,$$

so we get

$$np = n_\mathrm{i}^2 \exp\left(\frac{E_{\mathrm{F}n} - E_{\mathrm{F}p}}{\mathrm{k_B} T_\mathrm{L}}\right) = n_\mathrm{i}^2 \exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right). \tag{8.3}$$

Shockley assumed in his derivation (1949) a so called *low-level injection*, which means that the excess minority carrier concentrations injected in a quasi-neutral region are low compared to the majority carrier concentration. With this assumption, the majority carrier concentrations remain at their equilibrium values:

$$n_\mathrm{n} = n_\mathrm{n0}\,, \qquad p_\mathrm{p} = p_\mathrm{p0}\,. \tag{8.4}$$

From (8.3) we deduce

$$n_\mathrm{p} = n_\mathrm{p0} \exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right), \qquad p_\mathrm{n} = p_\mathrm{n0} \exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right), \tag{8.5}$$

for the distribution of minority carriers concentrations at the edge of the space charge region.

Since the electric field outside the space-charge region is neglected, the resulting current in these neutral regions is purely diffusive. The majority carrier concentrations are due to (8.4) constant in the quasi-neutral regions, but (8.5) tells us that this is not true for the minority concentrations. Consequently, the minority carrier concentrations determine the (purely diffusive) current.

Thus, the minority carrier current inside the p-region is

$$J_\mathrm{n} \approx \mu_\mathrm{n}\, \mathrm{k_B}\, T_\mathrm{L}\, \nabla n. \tag{8.6}$$

We will further neglect recombination both inside and outside the space-charge region, thus $J_\mathrm{n} \approx$ const. With $n(-l_\mathrm{p}) = n_\mathrm{p}$ as first boundary condition we can write

$$n(x) = n_\mathrm{p} + \frac{J_\mathrm{n}}{\mu_\mathrm{n}\, \mathrm{k_B}\, T_\mathrm{L}}(x + l_\mathrm{p})$$

Let L denote the width of the p-region and with $n(-L) = n_\mathrm{p0}$ and $L \gg l_\mathrm{p}$ we obtain for the current

$$J_\mathrm{n} = \mu_n \mathrm{k_B} T_\mathrm{L} \frac{n_\mathrm{p} - n_\mathrm{p0}}{L - l_\mathrm{p}} \approx \mu_\mathrm{n} \mathrm{k_B} T_\mathrm{L} \frac{n_\mathrm{p} - n_\mathrm{p0}}{L} = \frac{\mu_\mathrm{n} \mathrm{k_B} T_\mathrm{L} n_\mathrm{i}^2}{N_\mathrm{A} L}\left(\exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right) - 1\right).$$

An analogous procedure for holes yields

$$J_\mathrm{p} = \mu_\mathrm{p} \mathrm{k_B} T_\mathrm{L} \frac{p_\mathrm{n} - p_\mathrm{n0}}{L - l_\mathrm{n}} \approx \mu_\mathrm{p} \mathrm{k_B} T_\mathrm{L} \frac{p_\mathrm{n} - p_\mathrm{n0}}{L} = \frac{\mu_\mathrm{p} \mathrm{k_B} T_\mathrm{L} n_\mathrm{i}^2}{N_\mathrm{D} L}\left(\exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right) - 1\right).$$

The total diode current is the sum of the electron and hole currents (which do not interact because we have neglected R):

$$J = J_\mathrm{n} + J_\mathrm{p} = \frac{\mu_\mathrm{n} \mathrm{k_B} T_\mathrm{L} n_\mathrm{i}^2}{N_\mathrm{A} L}\left(\exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right) - 1\right) + \frac{\mu_\mathrm{p} \mathrm{k_B} T_\mathrm{L} n_\mathrm{i}^2}{N_\mathrm{D} L}\left(\exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right) - 1\right) \tag{8.7}$$

$$= \underbrace{\frac{\mathrm{k_B} T_\mathrm{L} n_\mathrm{i}^2}{L}\left(\frac{\mu_\mathrm{n}}{N_\mathrm{A}} + \frac{\mu_\mathrm{p}}{N_\mathrm{D}}\right)}_{=:J_\mathrm{s}}\left(\exp\left(\frac{V_\mathrm{c}}{V_\mathrm{T}}\right) - 1\right), \tag{8.8}$$

which is the characteristic current relation for the diode.

Let us recapitulate the assumptions leading to the analytic diode model:

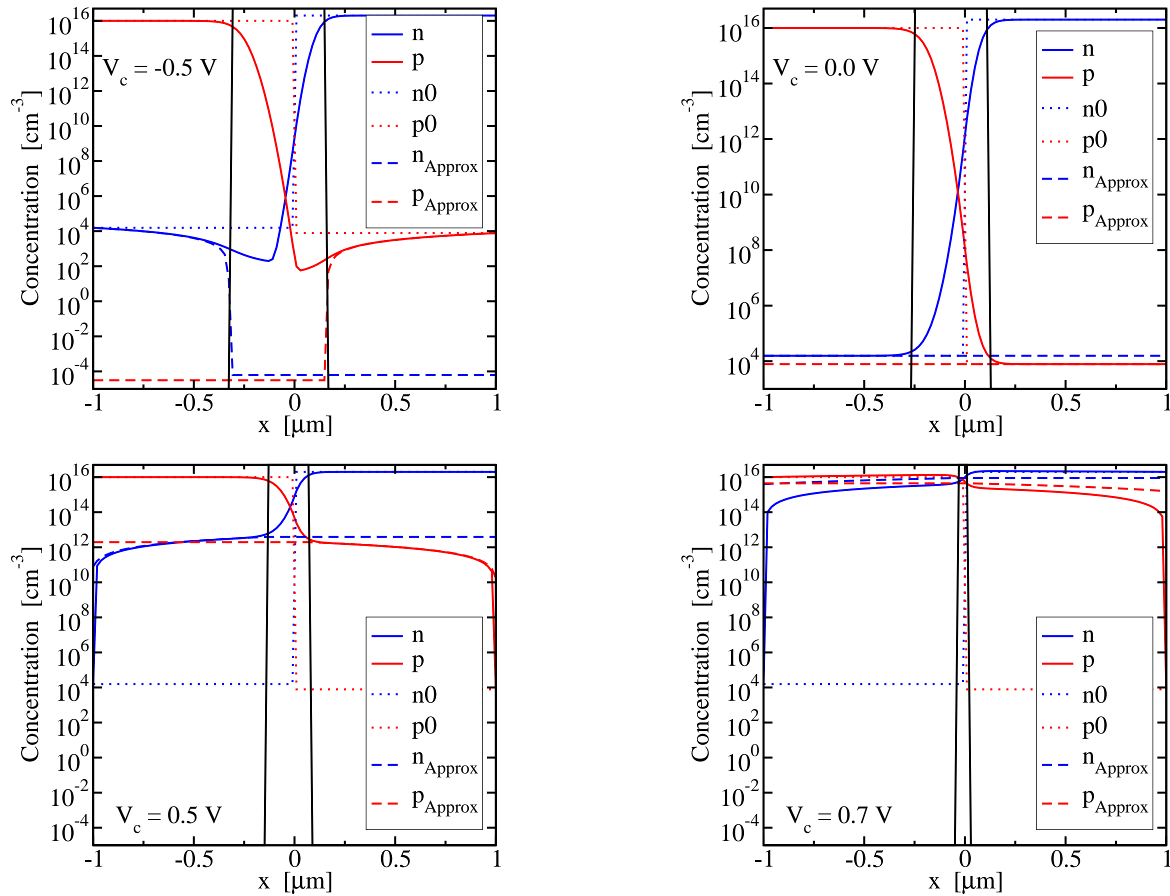


Figure 8.3: Comparison of the compact model with numerical results. At $V_c \neq 0.0\,\mathrm{V}$ we observe a slope in the minority carrier concentrations, which determines the current-voltage relation. At $V_c \approx 0.7\,\mathrm{V}$, the space charge region becomes very thin and the compact model does not show good agreement with the simulation results anymore.

- We have assumed *abrupt doping profiles*. In reality, such profiles do not exist. This is not a problem as long as the doping transition length is small compared to the space charge region. For small diodes with high (asymmetric) doping concentrations, space charge regions become very thin (cf. Eq. 8.2) and realistic shapes of doping profiles have to be considered.

- The *depletion approximation* assumes an abrupt space charge region. In reality, there is a smooth (but still steep) transition between the quasi neutral and the space charge regions in the order of the Debye length, which has to be considered for highly asymmetric doping profiles.

- *No recombination* within the space charge region was assumed. Real diodes are, however, non-ideal and the effects of generation and recombination have to be taken into account. Doing so, the exponential in (8.7) has to be extended by an *ideality factor* $1 \leq n \leq 2$, so that the current relation becomes

$$J = J_\mathrm{s}\left(\exp\left(\frac{V_\mathrm{c}}{nV_\mathrm{T}}\right) - 1\right) .$$

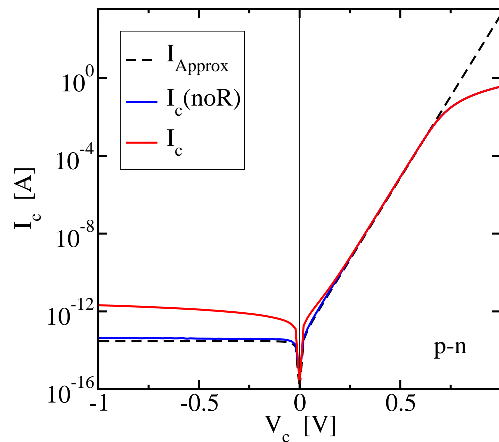


Figure 8.4: A real diode differs from the compact model especially through high-level injection and increased reverse-bias current due to generation and recombination.

- *Short diode*: On the one hand, recombination outside the space charge region can be neglected (diode is relatively short), but the diode is still long enough to neglect the space charge width in the gradient at (8.6).
- *Low-level injection*: We assumed that majority carriers were not influenced and that a pure minority diffusion current dominated.

A comparison of the compact model with numerical results in Fig. 8.3 shows good agreement of the minority concentrations outside the space charge region. However, as the applied voltage approaches ψ_{bi}, the approximation becomes poorer.

Finally, some further effects in real diodes shall be mentioned: First, *high-level injection* decreases the current for higher bias voltages. Second, the voltage drop in the quasi neutral regions decreases the current at high bias voltages (*high bias effect*). The third effect we mention here is a higher current under reverse-bias compared to the predictions of our model. This can be traced back to generation in the space charge region.

Apart from the physical effects mentioned, a real diode is at least two-dimensional, resulting in additional corner effects (larger electric fields!) that cannot be tackled by a simple one-dimensional approximation. Furthermore, doping profiles are more complex and the mobility is not constant either. Fig. 8.4 compares numerically simulated diode characteristics with the results from the compact model.

8.2 Small Signal Analysis of a Diode

In (8.7) the DC-characteristic of a diode is given. The transient description, however, requires additional considerations. In particular, charge storage effects have to be taken into account, otherwise the device would be infinitely fast, which contradicts the **inertia**[1] of charges.

There are two dominant forms of charge-storage:

- Charge is stored in the depletion region due to the dopants, resulting in a *junction capacitance* (or *depletion capacitance*). The junction capacitance is caused by charge dipoles

[1] **inertia** [ɪˈnɜː.ʃə]: Trägheit

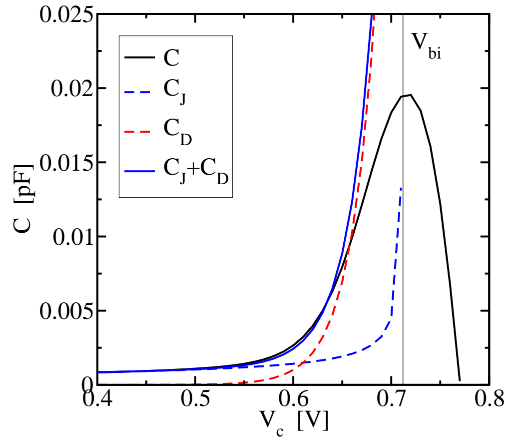


Figure 8.5: Comparison of compact model with numerical simulation with small-signal mode of *Minimos – NT*. There is a good agreement for V_c not too close to ψ_{bi}.

formed by ionized dopants in the depletion region so the depletion region width and thus the charge changes with the applied voltage. Let the contact potential change by a small amount dV_c. Then the depletion region charge per unit area changes by dQ'_J. The junction capacitance per unit area is therefore defined as

$$C'_J = \frac{dQ'_J}{dV_c}.$$

Since $Q'_J = qN_D l_n = qN_A l_p$ we get with (8.2)

$$C'_J(V_c) = qN_D \frac{dl_n}{dV_c} = \sqrt{\frac{q\varepsilon}{2(\psi_{bi} - V_c)(1/N_A + 1/N_D)}} = \frac{C'_J(0)}{\sqrt{1 - V_c/\psi_{bi}}}$$

Using again (8.2), we get

$$C'_J(V_c) = \frac{\varepsilon}{l},$$

which is just the same as for a parallel-plate capacitor!

- Charge is stored due to injected minority-carriers in the space-charge regions, leading to the *diffusion capacitance* (or *storage capacity*) C'_D.
  The hole charge stored in the n-region is

$$Q'_p = q\int_{l_n}^{L} (p_n(x) - p_{n0})\, dx = \frac{qL}{2} p_{n0} \left(\exp\left(\frac{V_c}{V_T}\right) - 1\right) \qquad \text{for } L \gg l_n.$$

  Similarly, the electron charge stored in the p-region is

$$Q'_n = q\int_{-l_p}^{-L} (n_p(x) - n_{p0})\, dx = \frac{qL}{2} n_{p0} \left(\exp\left(\frac{V_c}{V_T}\right) - 1\right) \qquad \text{for } L \gg l_p.$$

  In total we have

$$C'_D = \frac{dQ'_D}{dV_c} = \frac{dQ'_p + dQ'_n}{dV_c} = \frac{qLn_i^2}{2V_T} \left(\frac{1}{N_A} + \frac{1}{N_D}\right) \left(\exp\left(\frac{V_c}{V_T}\right) - 1\right).$$

  Due to the exponential term, the contribution is negligible under reverse bias. However, as V_c approaches ψ_{bi}, the capacitance is overestimated (cf. Fig. 8.5).

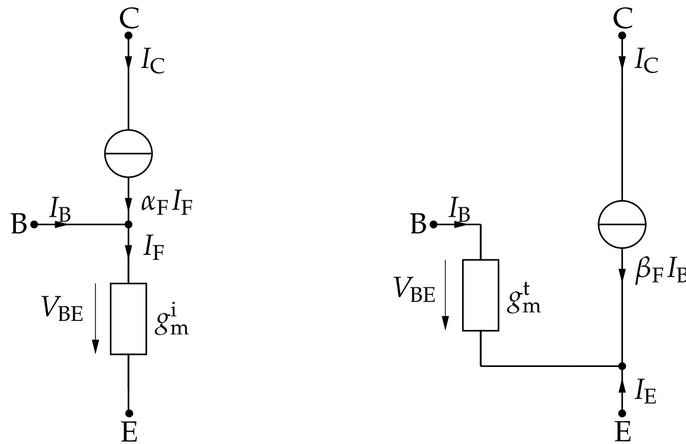


Figure 8.6: Frequently used equivalent circuit diagrams for a BJT.

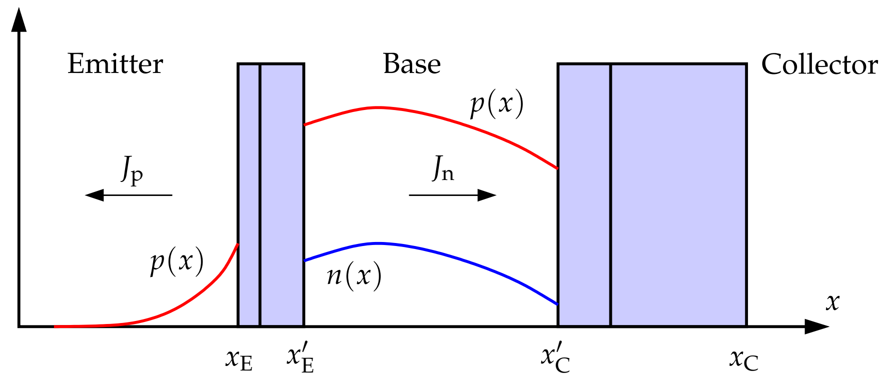


Figure 8.7: Electron and hole densities in a npn-transistor.

8.3 Analytical Bipolar Junction Transistor Model

As for the diode, we will derive a compact model for *bipolar junction transistors* (BJTs). In the literature, two equivalent circuit diagrams for such a model are often used, shown in Fig. 8.6. They are indeed equivalent, which is not obvious at first sight, but shall be shown now.

We start with the electron current relations in the base of a npn-transistor:

$$J_n = q\mu_n n E + qD_n \nabla n, \tag{8.9}$$

$$\nabla \cdot J_n = qR. \tag{8.10}$$

The Shockley-Read-Hall recombination rate (7.13) in the base with $p \gg n$ and $p \approx N_A$ becomes

$$R^{SRH} \approx \frac{nN_A - n_i^2}{\tau_n N_A} = \frac{n - n_{p0}}{\tau_n} =: \frac{\Delta n}{\tau_n},$$

where $n_{p0} = n_i^2/N_A$ is the (constant) equilibrium concentrations of electrons in the p-type base.

Assuming that the diffusion current dominates in (8.9), we obtain

$$\mathrm{q}D_\mathrm{n}\nabla^2 n = \mathrm{q}\frac{\Delta n}{\tau_\mathrm{n}}.$$

Cancelling q and using the identity $\nabla^2 n = \nabla^2(n - n_{\mathrm{p0}}) = \nabla^2(\Delta n)$ (because we assumed n_{p0} to be constant) leads to

$$D_\mathrm{n}\nabla^2(\Delta n) = \frac{\Delta n}{\tau_\mathrm{n}}.$$

In one dimension, the general solution of this differential equation is

$$\Delta n = C_1 \exp\left(\frac{x}{L_\mathrm{B}}\right) + C_2 \exp\left(-\frac{x}{L_\mathrm{B}}\right),$$

where $L_\mathrm{B} = \sqrt{D_\mathrm{n}\tau_\mathrm{n}}$ is the diffusion length. From the base-emitter-diode we find as boundary condition

$$n(0) = n_{\mathrm{p0}} \exp\left(\frac{V_{\mathrm{BE}}}{V_\mathrm{T}}\right), \quad \text{hence} \quad \Delta n(0) = n_{\mathrm{p0}}\left(\exp\left(\frac{V_{\mathrm{BE}}}{V_\mathrm{T}}\right) - 1\right) \tag{8.11}$$

and from the base-collector-diode

$$n(W) = n_{\mathrm{p0}} \exp\left(V_{\mathrm{BC}}/V_\mathrm{T}\right), \quad \text{hence} \quad \Delta n(W) = n_{\mathrm{p0}}\left(\exp\left(V_{\mathrm{BC}}/V_\mathrm{T}\right) - 1\right). \tag{8.12}$$

The recombination in the base is small, therefore L_B is large, $L_\mathrm{B} \gg W$, where W is the base width. For this reason, we can linearize $\exp(x) \approx 1 + x$ around zero and can now write

$$\Delta n \approx \Delta n(0)(1 - \frac{x}{W}) + \Delta n(W)\frac{x}{W}.$$

The current density at the beginning of the base is

$$J_\mathrm{n}(0) = \mathrm{q}D_\mathrm{n}\frac{\mathrm{d}\Delta n}{\mathrm{d}x} = \frac{\mathrm{q}D_\mathrm{n}}{W}\left(\Delta n(W) - \Delta n(0)\right) \tag{8.13}$$

and with (8.11) and (8.12) this becomes

$$I_\mathrm{n} = I_{\mathrm{n0}}\left(\left(\exp\left(V_{\mathrm{BE}}/V_\mathrm{T}\right) - 1\right) - \left(\exp\left(V_{\mathrm{BC}}/V_\mathrm{T}\right) - 1\right)\right) = I_{\mathrm{n1}} - I_{\mathrm{n2}},$$

where $I_{n0} = \mathrm{q}D_\mathrm{n}An_\mathrm{i}^2/(N_\mathrm{A}W)$. Note that the minority carriers in the emitter and collector are not affected:

$$I_{\mathrm{Ep}} = \frac{\mathrm{q}D_\mathrm{p}A}{W}\frac{n_\mathrm{i}^2}{N_{\mathrm{DE}}}\left(\exp\left(V_{\mathrm{BE}}/V_\mathrm{T}\right) - 1\right), \quad I_{\mathrm{Cp}} = \frac{\mathrm{q}D_\mathrm{p}A}{W}\frac{n_\mathrm{i}^2}{N_{\mathrm{DC}}}\left(\exp\left(V_{\mathrm{BC}}/V_\mathrm{T}\right) - 1\right).$$

For later use we define:

$$\frac{I_{\mathrm{n2}}}{I_{\mathrm{Ep}}} = \frac{D_\mathrm{n}}{D_\mathrm{p}}\frac{L_\mathrm{p}}{W}\frac{N_{\mathrm{DE}}}{N_\mathrm{A}} =: g_\mathrm{F}, \quad \frac{I_{\mathrm{n1}}}{I_{\mathrm{Cp}}} = \frac{D_\mathrm{n}}{D_\mathrm{p}}\frac{L_\mathrm{p}}{W}\frac{N_{\mathrm{DC}}}{N_\mathrm{A}} =: g_\mathrm{R}.$$

The total emitter current is then

$$I_\mathrm{E} = -I_\mathrm{n} + I_{\mathrm{Ep}} = -I_{\mathrm{n1}} + I_{\mathrm{n2}} + I_{\mathrm{Ep}} = -I_{\mathrm{Cp}}g_\mathrm{R} + \underbrace{I_{\mathrm{Ep}}(1 + g_\mathrm{F})}_{I_\mathrm{F}};$$

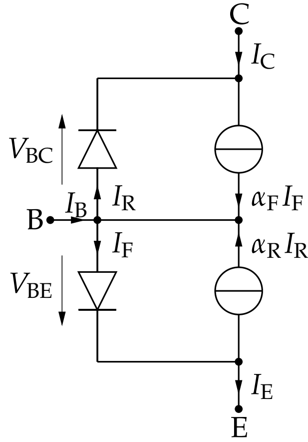


Figure 8.8: Injection version of the Ebers-Moll model.

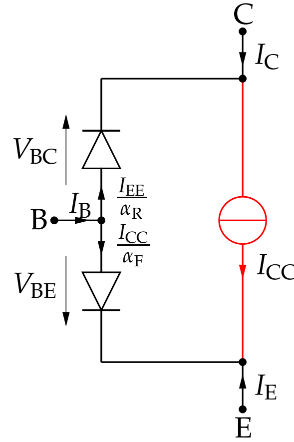


Figure 8.9: Transport version of the Ebers-Moll model.

Similarly, the collector current can be written as

$$I_C = -I_n - I_{Cp} = -I_{n1} + I_{n2} - I_{Cp} = -\underbrace{I_{Cp}(1+g_R)}_{I_R} + I_{Ep} g_F \, .$$

This gives for the controlled currents

$$g_F I_{Ep} = \frac{1+g_F}{g_F} I_F =: \alpha_F I_F, \quad g_R I_{Cp} = \frac{1+g_R}{g_R} I_R =: \alpha_R I_R$$

and so we arrive at

$$I_E = -\alpha_R I_R + I_F \, , \quad I_C = -I_R + \alpha_F I_F.$$

This is the *injection version* of the *static Ebers-Moll model*, where

$$I_F = I_{ES}\left(\exp\left(\frac{V_{BE}}{V_T}\right) - 1\right) \, , \quad I_R = I_{CS}\left(\exp\left(\frac{V_{BC}}{V_T}\right) - 1\right) \, . \tag{8.14}$$

If we define

$$I_{CC} = \alpha_F I_F = I_S\left(\exp\left(\frac{V_{BE}}{V_T}\right) - 1\right) \, , \tag{8.15}$$

$$I_{EE} = \alpha_R I_R = I_S\left(\exp\left(\frac{V_{BC}}{V_T}\right) - 1\right) \, , \tag{8.16}$$

with $I_S = \alpha_F I_{ES} = \alpha_R I_{CS}$, we arrive at the *transport version* of the static Ebers-Moll model with

$$I_{CT} = I_{CC} - I_{EE} \, , \quad \beta_F = \frac{1-\alpha_F}{\alpha_F} \, , \quad \beta_R = \frac{1-\alpha_R}{\alpha_R} \, .$$

Let us come back to the injection version of the model. In the active region there holds $I_R \approx 0$, so linearization of (8.14) around the operating point (V_{BE}^0, I_F^0) yields

$$I_F(V_{BE}) = I_F(V_{BE}^0) + \underbrace{\left.\frac{\partial I_F}{\partial V_{BE}}\right|_{V_{BE}^0}}_{=:g_m^i} \left(V_{BE} - V_{BE}^0\right)$$
$$= I_F^0 + g_m^i \left(V_{BE} - V_{BE}^0\right) ,$$

so in small signal analysis (i.e. $V_{BE} = V_{BE}^0 + v_{BE}$, $I_F = I_F^0 + i_F$) this results in

$$i_F = g_m^i v_{BE}, \tag{8.17}$$

which results in the equivalent circuit shown on the left of Fig. 8.6.

We can do similar simplifications for the transport version of the model: In the active region, $I_{EE} \approx 0$ and with (8.15) we can write the current at the basis as

$$I_{CC} = I_{BS} \left(\exp\left(\frac{V_{BE}}{V_T}\right) - 1 \right) .$$

Linearization around the operating point yields

$$I_B(V_{BE}) = I_B(V_{BE}^0) + \underbrace{\left.\frac{\partial I_B}{\partial V_{BE}}\right|_{V_{BE}^0}}_{=:g_m^t} \left(V_{BE} - V_{BE}^0\right)$$
$$= I_B^0 + g_m^t \left(V_{BE} - V_{BE}^0\right) ,$$

so in small signal analysis we find in analogy to (8.17) the relation

$$i_B = g_m^t v_{BE},$$

thus the equivalent circuit shown on the right of Fig. 8.6 is justified.

Comparing both models, the conductivities $g_m^i = \partial I_F / \partial V_{BE}$ and $g_m^t = \partial I_B / \partial V_{BE}$ are not the same. However, they are related via

$$g_m^t = (1 - \alpha_F) g_m^i , \qquad \alpha_F \leq 1 ,$$

as can easily be seen from Fig. 8.6 after expressing I_B as a function of I_F.

How well do the derived compact models reflect a real transistor? The most obvious difference is that the *transfer characteristics* $I_C(V_{BE})$ of our compact model show an output conductivity of zero. The reason for this is that we assumed a constant width of the base in (8.13). However, this is not true: The neutral base width is modulated by the applied collector-base reverse bias V_{CB}, which gives rise to the so-called *Early effect*: The off-state current I_S as well as the amplification factor β_F depend on V_{CB}, in linear approximation

$$I_S(V_{CB}) = I_S(0) \left(1 - \frac{V_{CB}}{V_A}\right) ,$$
$$\beta_F(V_{CB}) = \beta_F(0) \left(1 - \frac{V_{CB}}{V_A}\right) ,$$

where V_A is the *Early voltage*. The output conductance is then

$$g_c = \left.\frac{\partial I_C}{\partial V_{CE}}\right|_{V_{BE}=\text{const.}} \approx \frac{I_C(0)}{V_A} .$$

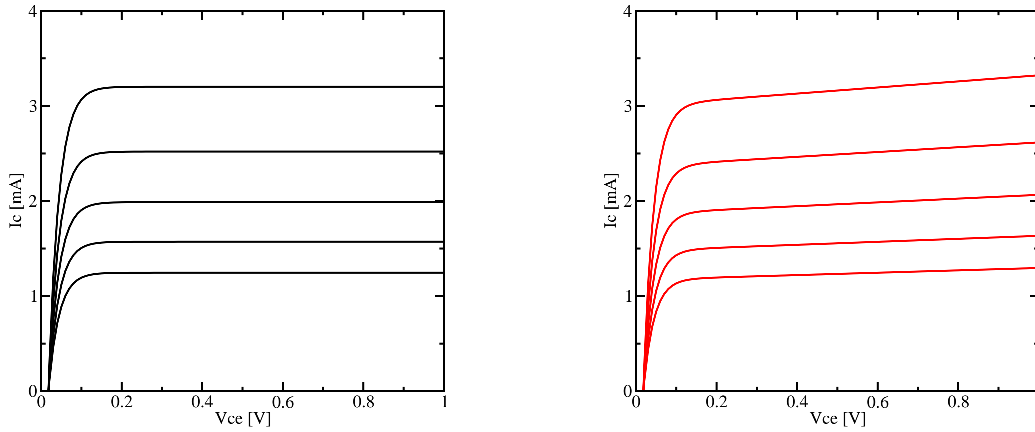


Figure 8.10: Transfer characteristic of our compact model (left) and after taking the Early effect into account (right).

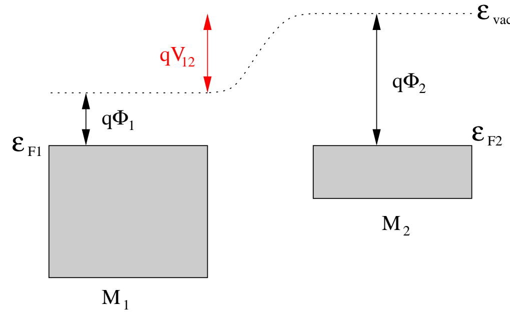


Figure 8.11: Work functions are used to describe a potential drop between two materials.

8.4 The Metal-Oxide-Semiconductor Capacitor

The work function of a material is the potential difference from the vacuum level to the Fermi level. Thus, for two materials

$$q\Phi_1 = E_{\text{vac}} - E_{\text{F1}}\,, \qquad q\Phi_2 = E_{\text{vac}} - E_{\text{F2}}\,.$$

The potential drop between material M_1 and material M_2 is the *contact potential*

$$\psi_{12} = \Phi_2 - \Phi_1.$$

With the concept of work functions in hand, we proceed to the simplest layout of a *MOS capacitor* (Fig. 8.12). Going from the gate to the substrate, the potentials are

$$\psi_{M1} + \psi_{1S} = (\Phi_1 - \Phi_M) + (\Phi_S - \Phi_1) = \Phi_S - \Phi_M = \Phi_{MS}. \tag{8.18}$$

Metal	Al	Pt	W	Mg	Ag	Au	Cu	Cr
$q\Phi_M$ (eV)	4.28	5.65	4.63	3.66	4.30	4.80	4.25	4.50

Table 8.1: Work function values for several metals.

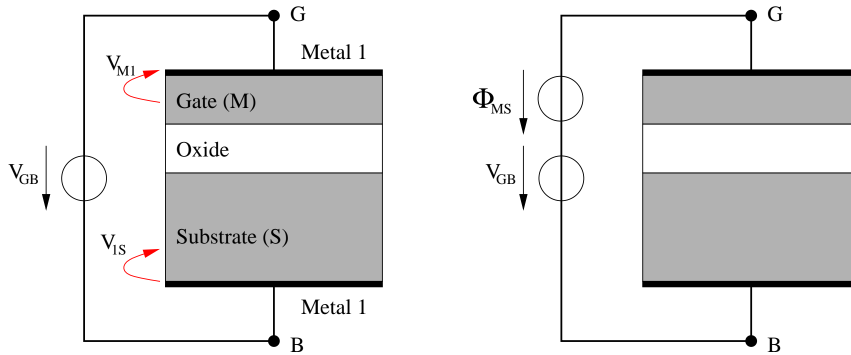


Figure 8.12: Schematics of a MOS capacitor.

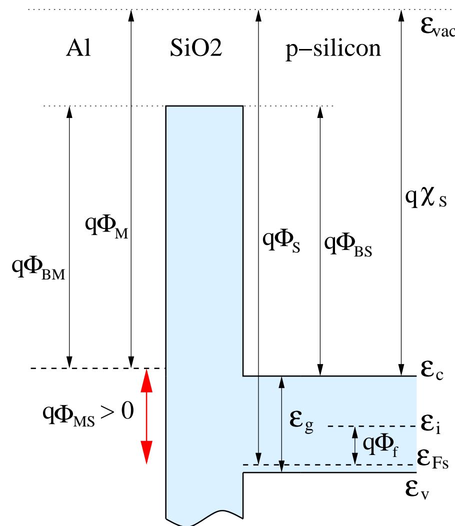


Figure 8.13: Band diagram of an ideal MOS capacitor.

Taking the work function difference Φ_{MS} into account allows us to assume ideal contacts in the sense that no additional built-in potentials occur. Conversely, if the voltage applied from the outside equals the work function difference, carrier concentrations are at their equilibrium values, so there is no voltage drop. From Kirchhoff's voltage law, $\Phi_{\mathrm{MS}} + V_{\mathrm{GB}} = 0$, so the *flat band voltage* is $\psi_{\mathrm{FB}} = -\Phi_{\mathrm{MS}}$.

Let us consider a more realistic setting, where the gate is made of aluminum, the oxide of SiO_2 and the substrate is p-doped silicon (Fig. 8.13). For the flat band voltage there holds $V_{\mathrm{FB}} = -\Phi_{\mathrm{MS}} > 0$. Similar to the work function in metals, we introduce the *electron affinity* χ_{S} of a semiconductor: It is the energy required to detach an electron from a singly charged negative ion. For our purposes, it is proportional to the difference between the vacuum potential and the conductance band edge: $\mathrm{q}\chi_{\mathrm{S}} = E_{\mathrm{vac}} - E_{\mathrm{c}}$.

The *Fermi potential* is given as $\mathrm{q}\Phi_{\mathrm{F}} = E_{\mathrm{i}} - E_{\mathrm{Fs}}$, thus we can write the work function as

$$\mathrm{q}\Phi_{\mathrm{S}} = E_{\mathrm{vac}} - E_{\mathrm{Fs}} = \mathrm{q}\chi_{\mathrm{S}} + E_{\mathrm{g}} - (E_{\mathrm{i}} - E_{\mathrm{v}}) + \mathrm{q}\Phi_{\mathrm{F}}. \tag{8.19}$$

Semiconductor	Si	Ge	GaAs	GaP	GaSb	InAs	InP	InSb
χ_S [V]	4.05	4.00	4.07	3.80	4.06	4.90	4.38	4.59

Table 8.2: Electron affinity for several semiconductors.

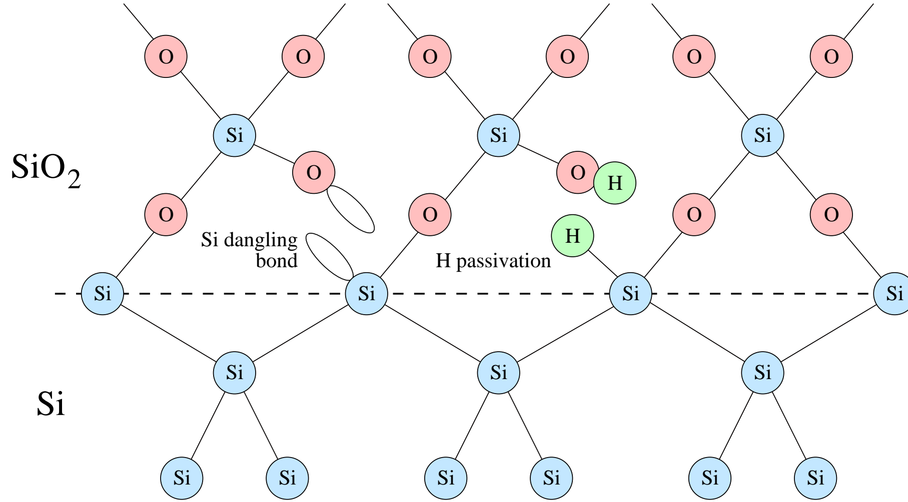


Figure 8.15: Dangling bonds on the interface can be passivated with hydrogen.

In Fig. 8.13 all quantities are illustrated as well as the barrier heights $q\Phi_{BM}$ and $q\Phi_{BS}$. The Fermi potentials depend on the doping level:

$$p\text{-type} : \Phi_F = +\frac{k_B T}{q} \ln\left(\frac{N_A}{n_i}\right) > 0,$$
$$n\text{-type} : \Phi_F = -\frac{k_B T}{q} \ln\left(\frac{N_D}{n_i}\right) < 0.$$

Since the Fermi potential is the only contribution that depends on the doping, a doping-independent part

$$q\Phi'_{MS} = q\Phi_M - q\chi_S - E_g + E_i - E_v.$$

Typical values are $\Phi'_{MS} = -0.6$ V for an Al-gate and a silicon substrate and $\Phi'_{MS} = 0.3$ V for an Au-gate and a silicon substrate. For a silicon gate, one finds $\Phi_{MS} = \Phi_F^G - \Phi_F^B$. In the case of a n^+-gate, $\Phi_F^G = -0.56$ V.

In a non-ideal MOS capacitor, positively charged *dangling bonds* at the Si-SiO$_2$-interface occur (cf. Fig. 8.15). Most of them are passivated by hydrogen. Apart from charges caused by dangling bonds, there may be interface trapped charges, oxide trapped charges or mobile ionic charges such as sodium, but they are usually negligible compared to $10^9 \ldots 10^{10}\ \mathrm{m}^{-2}$ caused by fixed oxide charges (that normally carry a positive charge) and

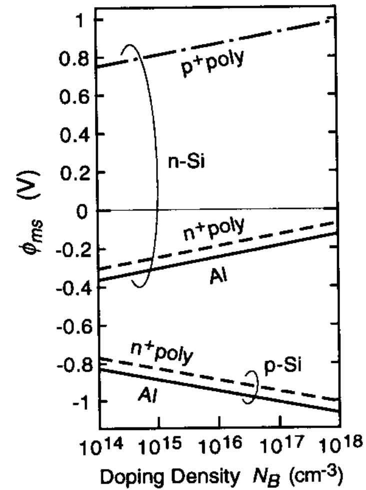


Figure 8.14: Work functions for several materials and doping densities.

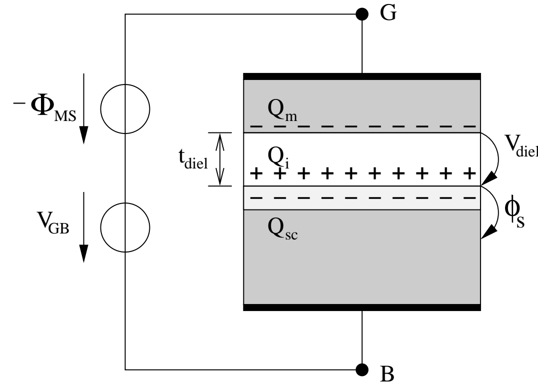


Figure 8.16: Charges in a MOS capacitor.

dangling bonds (positive charges in a pMOS and negative charges in an nMOS). These charges cause a shift in the flat band voltage. With the notations used in Fig. 8.16, the balance of charges requires

$$Q_\mathrm{m} + Q_\mathrm{i} + Q_\mathrm{sc} = 0\,. \tag{8.20}$$

For the dielectric displacement, $\varepsilon_\mathrm{diel} E_\mathrm{diel} = Q_\mathrm{m}$ holds. Using Kirchhoff's voltage law again, one finds

$$\psi_\mathrm{GB} - \Phi_\mathrm{S} - \psi_\mathrm{diel} + \Phi_\mathrm{MS} = 0. \tag{8.21}$$

The flat band condition requires $Q_\mathrm{sc} = 0$ and $\Phi_\mathrm{S} = 0$. Putting all equations together, we find

$$\psi_\mathrm{diel} = E_\mathrm{diel} t_\mathrm{diel} = \frac{Q_\mathrm{m}}{\varepsilon_\mathrm{m}} t_\mathrm{diel} = -\frac{Q_\mathrm{i}}{C_\mathrm{diel}}. \tag{8.22}$$

So, the new flat band voltage is

$$\psi_\mathrm{FB} = \frac{Q_\mathrm{i}}{C_\mathrm{diel}} - \Phi_\mathrm{MS}. \tag{8.23}$$

Next we are going to investigate the case $V_\mathrm{GB} \neq V_\mathrm{FB}$. In this case, a space-charge region forms in the semiconductor near the interface. The total potential drop caused by the space-charge region is called the *surface potential* ϕ_S and induces a shift of the band edges:

$$E_\mathrm{c}(x) = E_\mathrm{c,0} - \mathrm{q}\phi(x),$$
$$E_\mathrm{v}(x) = E_\mathrm{v,0} - \mathrm{q}\phi(x),$$

where the potential $\phi|_\mathrm{bulk} = 0$ and the surface potential is $\phi_\mathrm{S} = \phi(0)$.

In Tab. 8.3 and Fig. 8.17 the band edge energies and electrostatic potential for a nMOS capacitor for several bias voltages are shown. Very important with respect to the MOSFET is the case of inversion: In equilibrium, the carrier concentrations are given by (cf. Chapter 2)

$$n = n_\mathrm{i} \exp\left(\frac{E_{\mathrm{F}n} - E_\mathrm{i}}{\mathrm{k_B} T_\mathrm{L}}\right), \quad p = n_\mathrm{i} \exp\left(\frac{E_\mathrm{i} - E_{\mathrm{F}p}}{\mathrm{k_B} T_\mathrm{L}}\right). \tag{8.24}$$

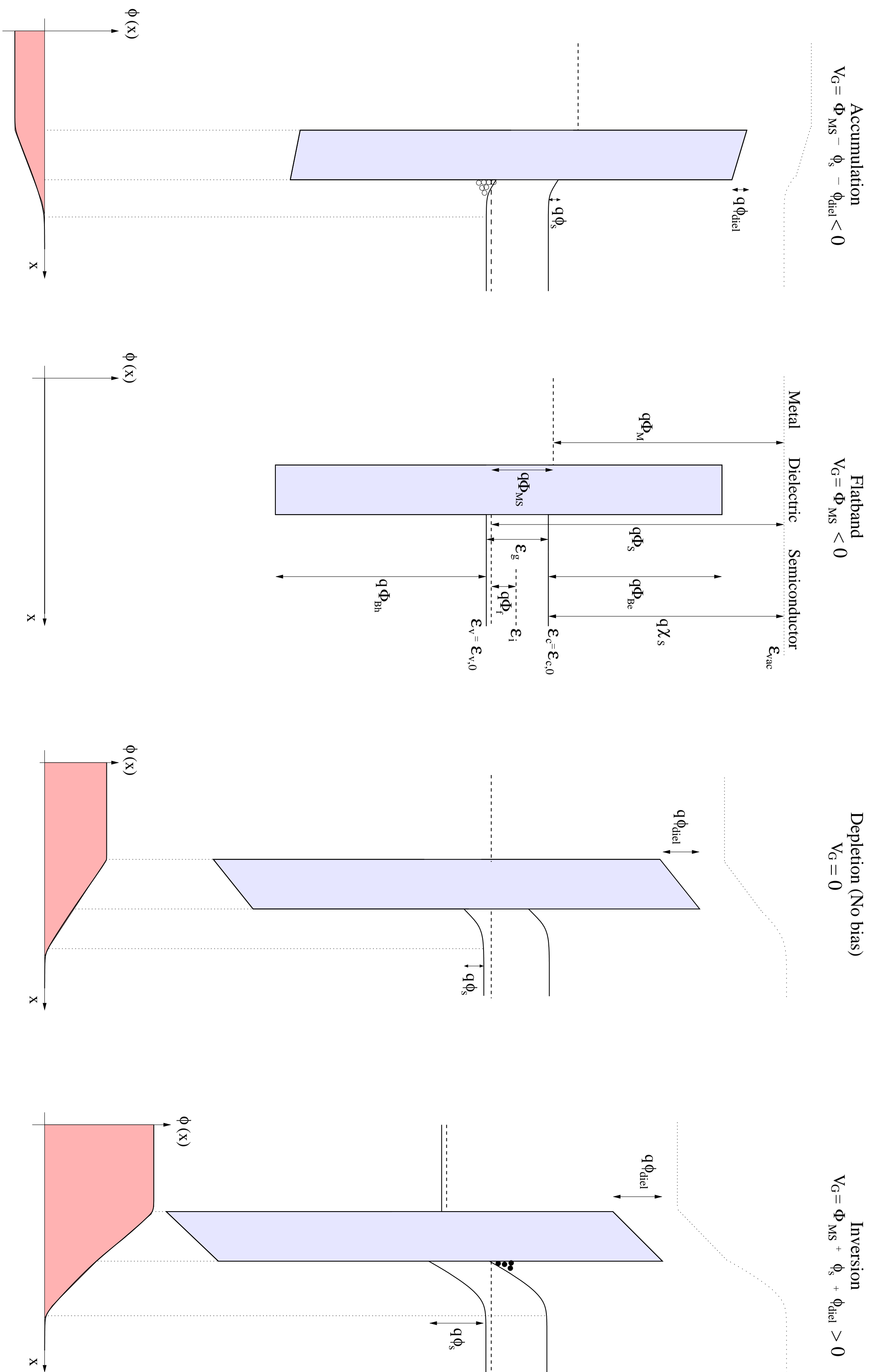


Figure 8.17: Band energies and electrostatic potential for an nMOS in different operation conditions.

	Bias	Surface potential	Charges in substrate
Accumulation	$V_{GB} < V_{FB}$	$\phi_S < 0$	$Q_{sc} > 0$
Flat band	$V_{GB} = V_{FB}$	$\phi_S = 0$	$Q_{sc} = 0$
Depletion	$V_{GB} > V_{FB}$	$\phi_S > 0$	$Q_{sc} < 0$
Inversion	$V_{GB} \gg V_{FB}$	$\phi_S > 0$	$Q_{sc} < 0$

Table 8.3: Operation ranges of a nMOS (i.e. p-type substrate, $V_{FB} < 0$).

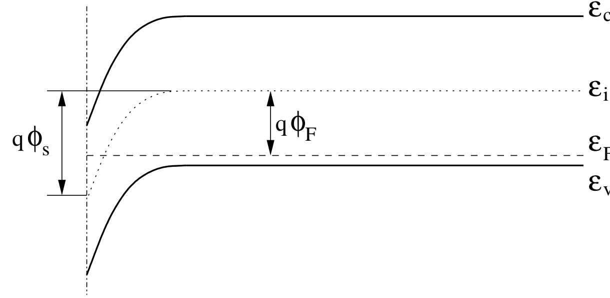


Figure 8.18: Inversion occurs when the conduction band-edge falls below the Fermi level.

The onset of strong inversion is (somewhat arbitrarily) defined as the point at which $n_{surf} = p_{bulk}$, where

$$n_{surf} = n_i \exp\left(\frac{E_{Fn} - E_{i,surf}}{k_B T_L}\right), \quad E_{i,surf} = E_i - q\phi_S. \tag{8.25}$$

According to Fig. 8.18, the conduction band-edge falls below the Fermi-level as soon as $\phi_S = 2\Phi_F$.

Still, charge neutrality has to hold in inversion as well. According to Fig. 8.19, charges in the space charge region consist of a depletion layer charge Q_d and an inversion (channel) charge Q_{ch}, thus

$$Q_m + Q_i + Q_{ch} + Q_d = 0. \tag{8.26}$$

Kirchhoff's voltage law requires $V_{GB} = \Phi_{MS} + V_{diel} + \phi_S$. As before, the dielectric displacement is $\varepsilon_{diel} E_{diel} = Q_m$, which together with the conditions $\phi_S = 2\Phi_{MS}$ and $Q_{ch} \approx 0$ for strong inversion yields the *threshold voltage*

$$V_t = \Phi_{MS} - \frac{Q_i}{C_{diel}} + 2\Phi_F - \frac{Q_d}{C_{diel}}. \tag{8.27}$$

For n-channel devices (i.e. p-bulk), normally $V_t \geq 0$, while for p-channel devices (i.e. n-bulk), $V_t < 0$. One can express Q_d in (8.27) by

$$Q_d = \begin{cases} -qN_A d, & p\text{-type}, \\ qN_D d, & n\text{-type}. \end{cases}$$

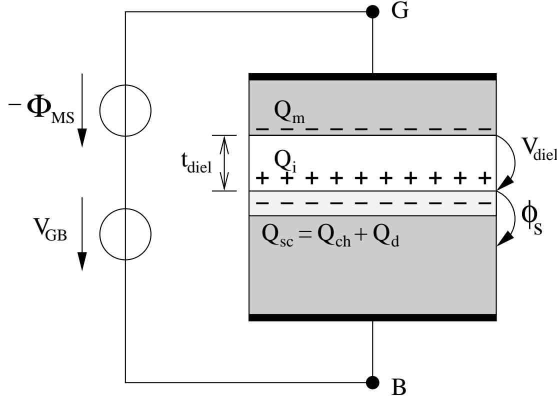


Figure 8.19: Charges in a MOS channel in case of inversion.

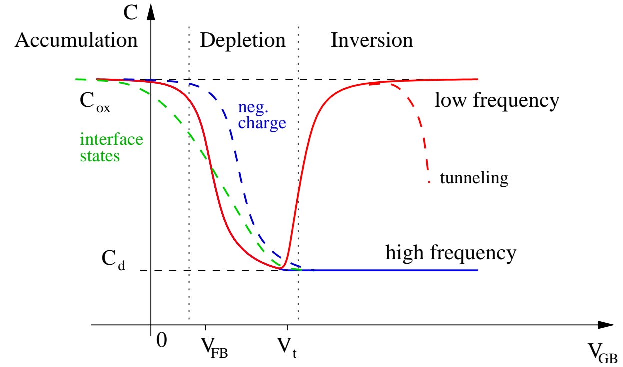


Figure 8.20: C(V)-curve of a MOS capacitor.

With the *depletion layer width*

$$d = \begin{cases} \sqrt{\frac{2\varepsilon_S \phi_S}{qN_A}}, & p\text{-type}, \\ \sqrt{\frac{2\varepsilon_S |\phi_S|}{qN_D}}, & n\text{-type}, \end{cases} \tag{8.28}$$

one finally obtains

$$V_t = \begin{cases} \Phi_{MS} - \frac{Q_i}{C_{diel}} + 2\Phi_F - \frac{\sqrt{4\varepsilon_S q N_A \Phi_F}}{C_{diel}}, & p\text{-MOS } (\Phi_F > 0), \\ \Phi_{MS} - \frac{Q_i}{C_{diel}} + 2\Phi_F - \frac{\sqrt{4\varepsilon_S q N_D |\Phi_F|}}{C_{diel}}, & n\text{-MOS } (\Phi_F < 0). \end{cases} \tag{8.29}$$

The main electrical measurement to determine the insulator quality in a MOSFET is the *capacity-voltage curve (C(V)-curve)* of the MOS-capacitor between gate and bulk. It enables the determination of insulator thickness (magnitude of capacity in accumulation), the flat band voltage as well as threshold voltage, the bulk doping (from the inversion capacitance) and the interface trap density (shape of C(V)-curve), as can be seen in Fig. 8.20. It is important to note that there are two distinct behaviors at inversion, depending on the frequency of the applied signal. For high frequencies, the inversion channel cannot built up properly, thus only C_d contributes. In a real MOSFET, the high frequency branch is not observed as carriers are supplied by the source and drain regions.

Despite all the details we put into our model, it does not include any quantum-mechanical effects, which leads to significant deviations as devices are scaled down to several nanometers. In Fig. 8.21 a comparison of classical numerical simulation with quantum-mechanical simulation for an nMOS with $t_{diel} = 15$Å is shown.

8.5 The Metal-Oxide-Semiconductor Field-Effect-Transistor (MOSFET)

Already back in 1926, Julius Edgar Lilienfeld proposed a *Method and Apparatus for Controlling Electric Currents* (Fig. 8.22). However, fabrication was not possible due to material-related problems. In 1960, Kahng and Attala realized the first field-effect transistor in MOS technology. We will summarize the analytical compact model given by Sah in 1964 [?].

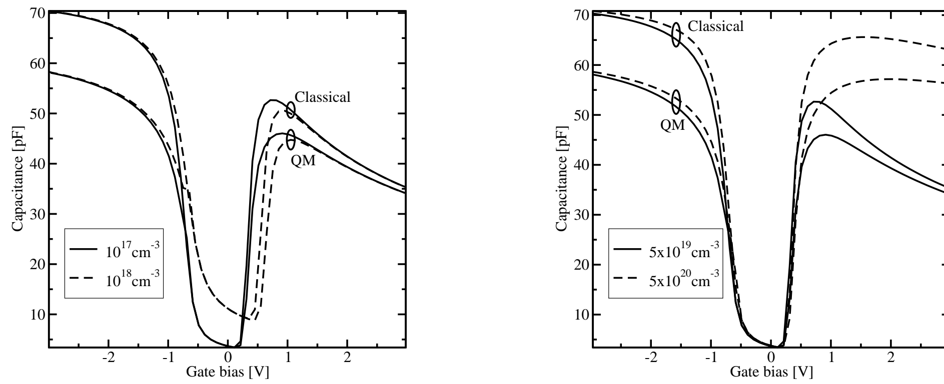


Figure 8.21: Influence on substrate doping (left) and polydepletion (right) using a classical and quantum-mechanical simulation of a nMOS with $t_{\mathrm{diel}} = 15$Å.

In the *linear region*, where $V_{\mathrm{GS}} > V_{\mathrm{t}}$ and $V_{\mathrm{GD}} > V_{\mathrm{t}}$, the drain current is given as

$$I_{\mathrm{D}} = \frac{W}{L}\mu_{\mathrm{n}}C_{\mathrm{diel}}\left(\left(V_{\mathrm{GS}} - V_{\mathrm{t}}\right)V_{\mathrm{DS}} - \frac{1}{2}V_{\mathrm{DS}}^2\right),$$

with the *threshold voltage*

$$V_{\mathrm{t}} = V_{\mathrm{FB}} + 2\Phi_{\mathrm{F}} + \gamma\sqrt{2\Phi_{\mathrm{F}} - V_{\mathrm{BS}}} \tag{8.30}$$

including the *body factor*

$$\gamma = \frac{1}{C_{\mathrm{diel}}}\sqrt{2\varepsilon_{\mathrm{Si}}\mathrm{q}N_{\mathrm{A}}}.$$

This is the level 1 MOSFET model of the circuit simulator SPICE.

In the *saturation region* (also referred to as *pinch-off region*), $V_{\mathrm{GS}} > V_{\mathrm{t}}$, but $V_{\mathrm{GD}} < V_{\mathrm{t}}$. The current remains at constant level as soon as the saturation voltage is reached:

$$\frac{\mathrm{d}I_{\mathrm{D}}}{\mathrm{d}V_{\mathrm{DS}}} = 0 \Longrightarrow V_{\mathrm{DS}}^{\mathrm{sat}} = V_{\mathrm{GS}} - V_{\mathrm{t}}.$$

The idealized MOSFET then behaves like an ideal current source (i.e. no dependence on V_{DS}):

$$I_{\mathrm{D}} = \frac{W}{L}\mu_{\mathrm{n}}C_{\mathrm{diel}}\frac{1}{2}\left(V_{\mathrm{GS}} - V_{\mathrm{t}}\right)^2,$$

For example, with technological parameters $N_{\mathrm{A}} = 5 \times 10^{17}\ \mathrm{cm}^{-3}$, $T = 300$ K, $Q_{\mathrm{i}} = 10^{10}\mathrm{q}/\ \mathrm{cm}^2$, $t_{\mathrm{ox}} = 10$ nm, $\mu_{\mathrm{n}} = 300\ \mathrm{cm}^2/\mathrm{V}$, $W/L = 1$ and $\Phi_{\mathrm{F}}^{\mathrm{G}} = -0.56$ V the calculated parameters that determine the output characteristics are $\phi_{\mathrm{S}} = 2\Phi_{\mathrm{F}} = 0.896$ V, $V_{\mathrm{FB}} = -1.01$ V, $V_{\mathrm{t}} = 1.00$ V and $\gamma = 1.18\ \mathrm{V}^{1/2}$. In Fig. 8.23 a plot of the resulting output characteristics is shown.

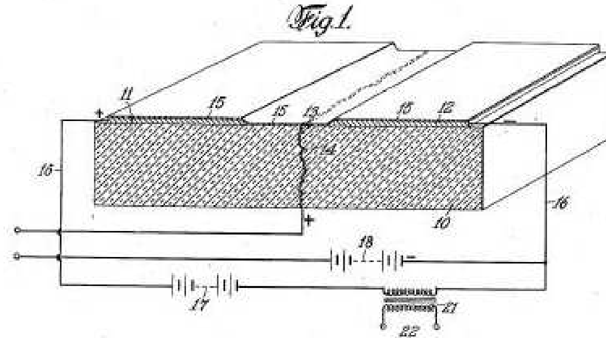


Figure 8.22: Schematic of a *Method and Apparatus for Controlling Electric Currents*, proposed by Lilienfeld in 1926.

Even though the output characteristics reflect the physical device to a certain extent, the model still yields infinite output conductance. In real MOSFETs, this is not the case, because the channel is pinched off at $x = L - l_\mathrm{d}$, where l_d depends on V_DS (*channel length modulation*). The drain saturation current now becomes

$$I_\mathrm{D}^\mathrm{sat} = \frac{W}{L - l_\mathrm{d}} \mu_\mathrm{n} C_\mathrm{diel} \frac{1}{2} \left(V_\mathrm{GS} - V_\mathrm{t}\right)^2 .$$

Therefore, with the approximation

$$I_\mathrm{D} = I_\mathrm{D}^\mathrm{sat} \frac{L}{L - l_\mathrm{d}},$$

a linearization (recall $1/(1-x) \approx 1 + x$ for small x) yields

$$I_\mathrm{D} \approx \left(1 + \frac{l_\mathrm{d}}{L}\right) I_\mathrm{D}^\mathrm{sat} \approx \left(1 + \lambda V_\mathrm{DS}\right) I_\mathrm{D}^\mathrm{sat},$$

where the *channel length parameter* λ $[\mathrm{V}^{-1}]$ was introduced. This approximation now results in a finite output conductance.

In Fig. 8.24, simulation results with MINIMOS-NT for a MOSFETs are shown in two different scales. In the subthreshold region, where $V_\mathrm{GS} < V_\mathrm{t}$, the *subthreshold slope* is

$$S = \frac{\mathrm{d}V_\mathrm{GS}}{\mathrm{d}(\log I_\mathrm{D})} \leq \ln(10) \frac{k_\mathrm{B} T}{\mathrm{q}}.$$

This raises the question of a proper definition of the *threshold current*, so that values for the threshold voltage are comparable. A practical definition (or convention) is

$$I_\mathrm{D}^\mathrm{th} = 0.1 \mu\mathrm{A} \frac{W}{L}.$$

Finally, we must not skip the *body effect*: The threshold voltage V_t depends on the substrate bias V_SB, but we are not going into further details.

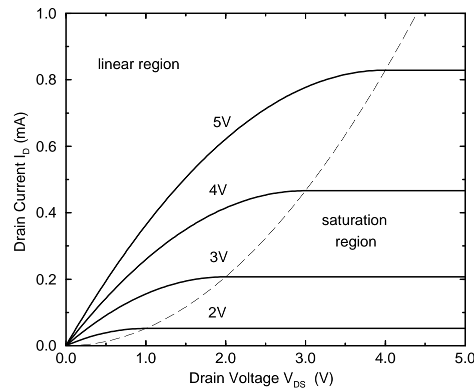


Figure 8.23: Characteristics of a MOSFET in the linear and in the saturation region.

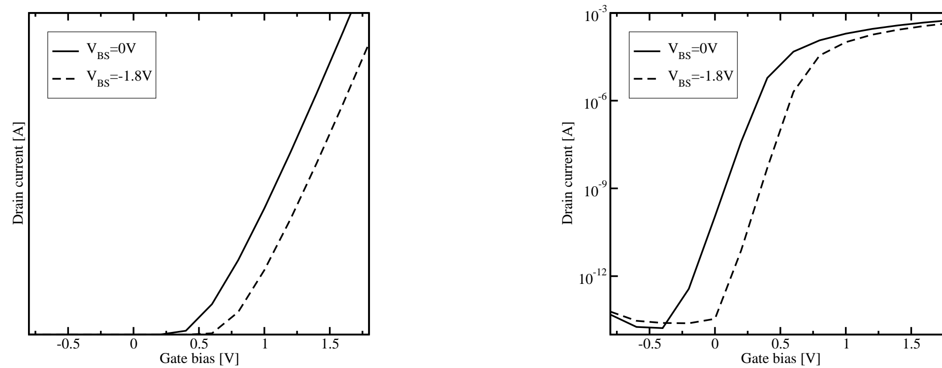


Figure 8.24: Transfer characteristics of a MOSFET simulated with MINIMOS-NT and plotted in linear scale (left) and logarithmic scale (right) for the drain current.

8.6 CMOS Design Issues

Current MOSFETs are scaled down to gate lengths of several nanometers only. In 1974, Dennard analyzed the effects of scaling on physical and electrical quantities in devices. We will have a look at constant-field scaling, where design parameters are chosen such that the electrical field within the device remains the same.

In Fig. 8.25, the effects of *(constant-field) scaling* of a device by a factor $s > 1$ are demonstrated. Some problems and deviations from the realizations in practice are as follows:

- Within one product family, supply voltages remain constant for reasons of compatibility with other peripheral devices. Thus, the electric field inside a device actually increases even for a constant-field scaling.
- In modern MOSFETs, the gate oxide thickness approaches one nanometer, which corresponds to a few atomic layers only. The atomic structure of matter comes into play: It is not possible to decrease the oxide thickness by a constant factor anymore. Instead,

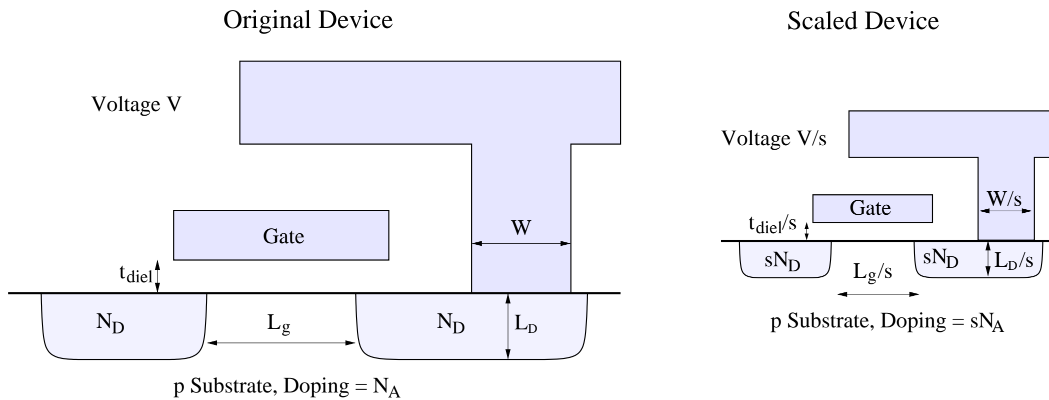


Figure 8.25: Effect of constant-field scaling on device dimensions.

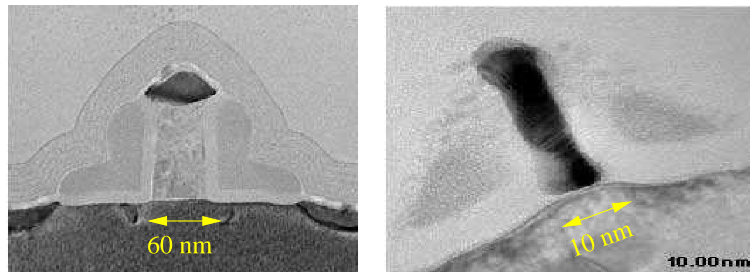


Figure 8.26: Two MOSFETs (Intel) with 60nm (left) and 10nm (right) gate length.

there are only discrete multiples of the width of one atomic layer possible. Additionally, quantum mechanical effects have to be taken into account.

- A decrease of the diameter of metallic interconnects leads to an increase of the current densities in the interconnect. High currents may destroy the metallic lattice of the interconnect over time, leading to a breakdown of the circuit. This effect is known as *electromigration*.

- Doping concentrations face natural limits: The impurities have to be incorporated into the semiconductor lattice. If their concentration is too high, this is not possible anymore. For instance, it is difficult in practice to obtain electrically active concentrations larger than $e \times 10^{20}$ cm^{-3}.

In Fig. 8.26, pictures of real MOSFETs from Intel are shown. In contrast to clearly-defined rectangular domains in a two-dimensional cross-section of a MOSFET, the real device does not have such clearly defined sharp transitions between different materials. One has to keep in mind that such small structures are currently fabricated using ultra-violet (UV) light with a wavelength of 193nm (extreme UV light with smaller wavelengths is in preparation, but not technically mature yet), thus posing highest quality requirements (and costs) on the fabrication equipment.

For a standard CMOS inverter, Fig. 8.27 shows some of the requirements and problems that

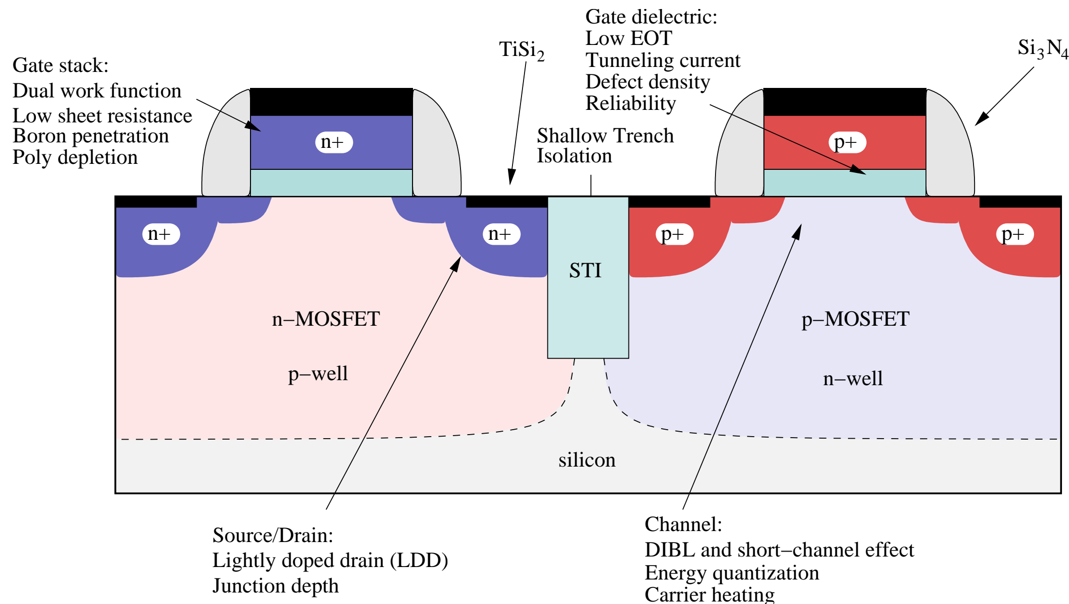


Figure 8.27: Requirements and problems with CMOS scaling.

arise from scaling. We will list some of the electrical and fabrication issues in the following:

- *Shallow trench isolation*: As devices are packed closer to each other, the surface leakage current between adjacent wells increases dramatically. This effect can be suppressed (though not completely eliminated) by fabricating shallow trenches filled with a dielectric between wells.

- *Threshold voltage roll-off*: The threshold voltage of a MOSFET is (besides other effects) determined by the depletion charge Q_d as given in (8.27). The zone of the gate-controlled depletion charge is determined by the gate length and has a trapezoidal shape due to the **encroachment**[1] of the depletion regions from the source and drain reversed-bias junctions into the depletion zone created by the gate electrode. If the channel is long, this encroachment can be neglected. However, for short channels, the gate charge is considerably reduced and the threshold voltage changes (cf. Fig. 8.28). The problem associated with the short-channel effect is not that devices with different channel lengths have different threshold voltages, rather, the problem is that in short devices small statistical variations in the gate length give rise to larger statistical variations of the threshold voltage, which poses a reproducibility problem.

- *Drain-induced barrier lowering* (DIBL): When the drain voltage of a turned-on short-channel MOSFET increases from the linear region toward the saturation region, its threshold voltage roll-off becomes larger (cf. Fig. 8.28, where the conduction band edge is shown on the left). In a turned-on MOSFET, carriers face an energy barrier between source and drain that is controlled by V_{GS}. In short-channel MOSFETs, the barrier height is additionally influenced by V_{DS}: In the saturation region, the depletion-layer width of the reverse-biased *pn*-junctions increases and reduces the effective potential barrier width. In long-channel devices, this reduction of the barrier width is not significant, but at short channels, the

[1] **encroachment** [ɪnˈkrəʊtʃ.mənt]: Beeinträchtigung

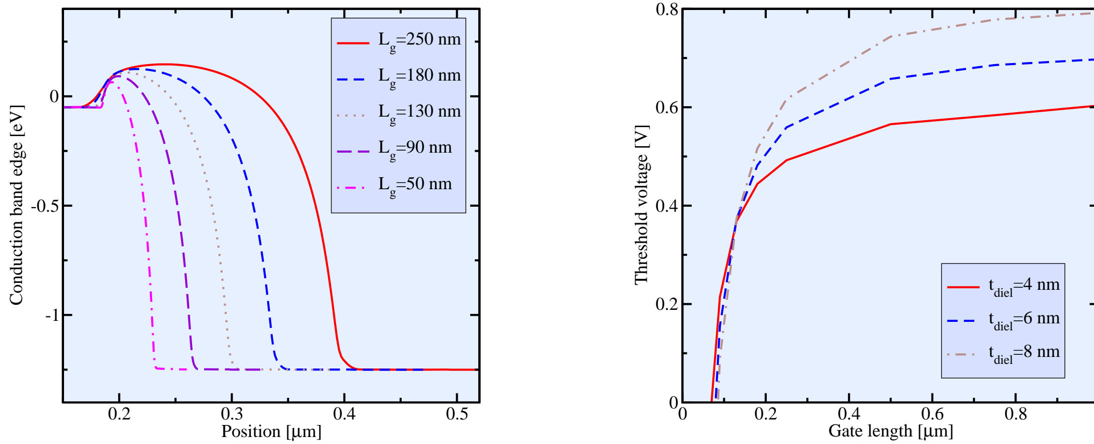


Figure 8.28: Drain induced barrier lowering (left) and roll-off-curve resulting from short-channel *n*-MOSFETS (right).

maximum barrier height is also reduced and leads to a substantial increase in electron injection from the source to the drain. As a result, the subthreshold current increases. In other words: An increase of the drain voltage leads to a decrease of the threshold-voltage. Therefore, the threshold voltage increases rapidly for small gate lengths, so that the MOSFETs cannot be turned off properly anymore.

- *Bulk punch-through*: DIBL causes the formation of a leakage path at the SiO_2/Si interface. If the drain voltage is large enough, significant leakage current may also flow from drain to source via the bulk of the substrate in a short-channel MOSFET due to the increasing depletion-layer width at the drain junction. Consequently, the gate can no longer turn the device completely off and loses control of the drain current at high drain voltages. Even worse, when two adjacent depletion regions touch, *punchthrough* occurs, which will destroy the device. To overcome this problem, anti-punchthrough implants (*retrograde wall*, pocket, halo) exist.

In CMOS inverters with small size, the problem of *polydepletion* comes into play: Polysilicon gates allow the adjustment of work functions and hence threshold voltages by doping. However, a depletion layer at the interface to the dielectric can show up at low electron concentrations (Fig. 8.30), resulting in a voltage drop

$$V_{\mathrm{poly}} \approx \frac{\varepsilon_{\mathrm{diel}}^2 E_{\mathrm{diel}}^2}{2q\varepsilon_{\mathrm{Si}} N_{\mathrm{poly}}},$$

This voltage drop leads to an effective increase of the dielectric thickness, which is not at all desired. However, the problem can be avoided by the use of metal gates, which, unfortunately, introduce a **plethora**[1] of additional problems into the process.

Another quantum-mechanical issue is *carrier quantization*. The distribution of carriers in the channel taking quantum mechanics into account differs compared to classical models. In a classical view, the peak concentration of carriers is directly at the interface to the gate-dielectric,

[1] **plethora** [ˈpleθ.ɚ.ə]: Fülle, Vielzahl

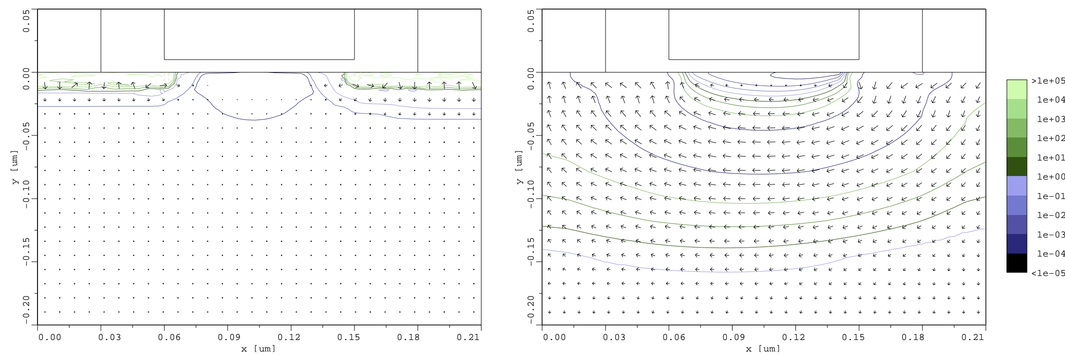


Figure 8.29: Current density in a turned-off MOSFET with (left) and without (right) retrograde wall. In the latter case the gate cannot control the current flow anymore, thus the device cannot be switched off properly.

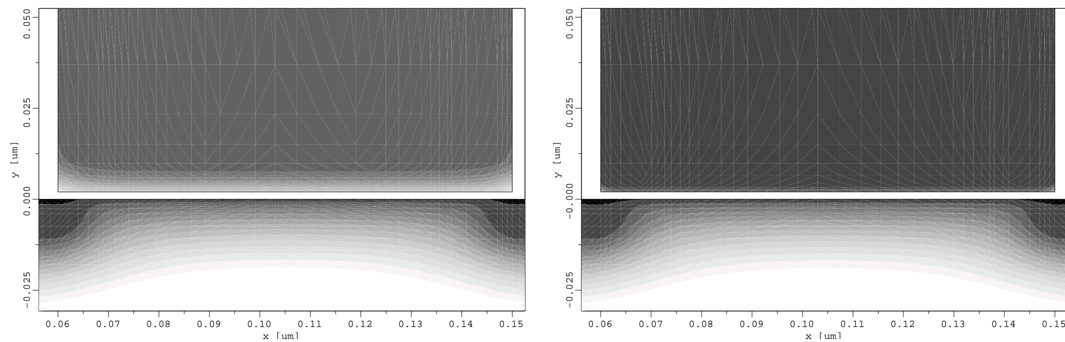


Figure 8.30: Electron concentrations for $N_{\mathrm{poly}} = 10^{19}\mathrm{cm}^{-3}$ (left) and $N_{\mathrm{poly}} = 10^{20}\mathrm{cm}^{-3}$. In the former case, polydepletion in the gate occurs.

but quantum-mechanics **prohibits**[1] a peak at the interface, so that the concentration maximum is in fact inside the channel (Fig. 8.31). This modifies the output-characteristics (cf. Fig. 8.32)

Single- and double-gate SOI (silicon on insulator) MOSFETs offer a superior control of the inversion charge. In single-gate SOI MOSFETs, the undoped channels offer high mobility (recall that impurity scattering at dopants decreases the mobility!), whereas the insulator at the bottom prevents punchthrough. Double-gate SOI result in a symmetric concentration profile of the inversion charge thanks to their symmetric layout with a gate electrode on both ends of the channel. However, SOI devices behave worse when it comes to heat conductance, because the insulator typically has a much lower heat conductivity than the bulk silicon.

The scaling issues discussed in this section are only a selection of the challenges. For more details, the reader is referred to the literature [?, ?].

[1] **to prohibit** [prəˈhɪb.ɪt]: verhindern, unterbinden

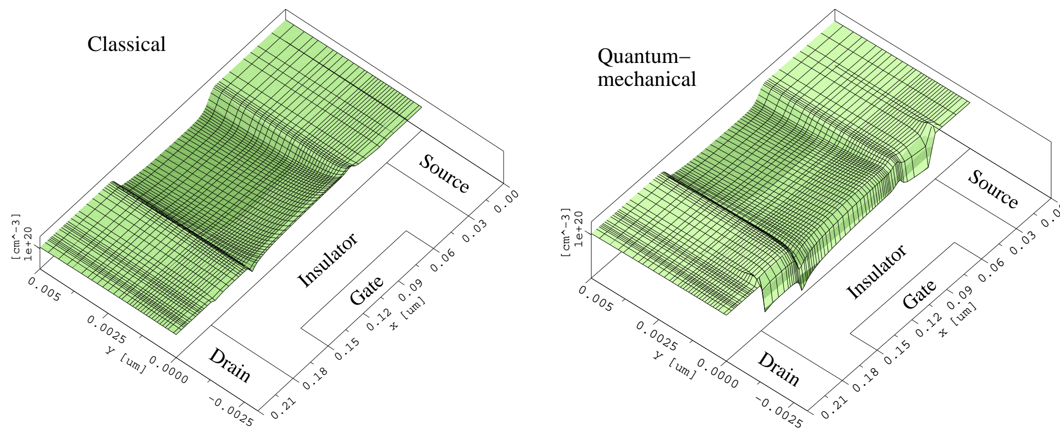


Figure 8.31: Carrier quantization is a quantum-mechanical effect and results in a peak concentration in the interior of the channel.

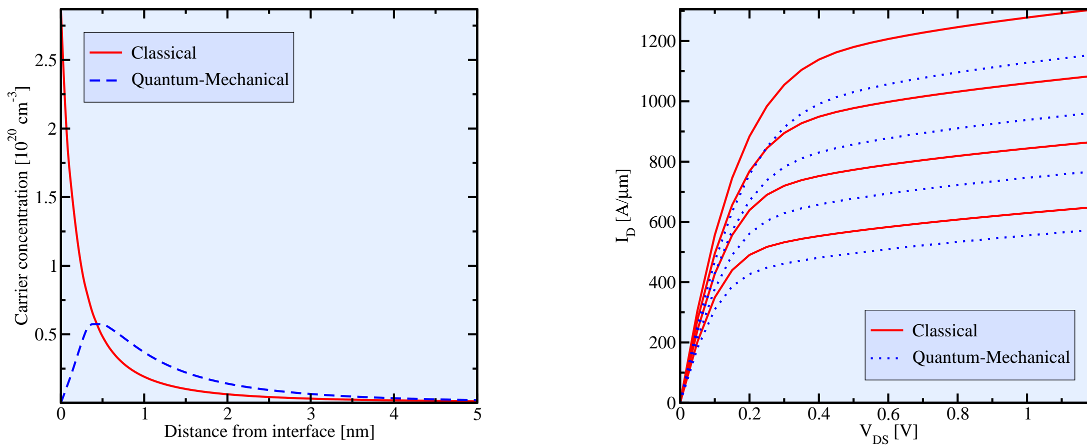


Figure 8.32: Carrier quantization and its effect on output-characteristics.

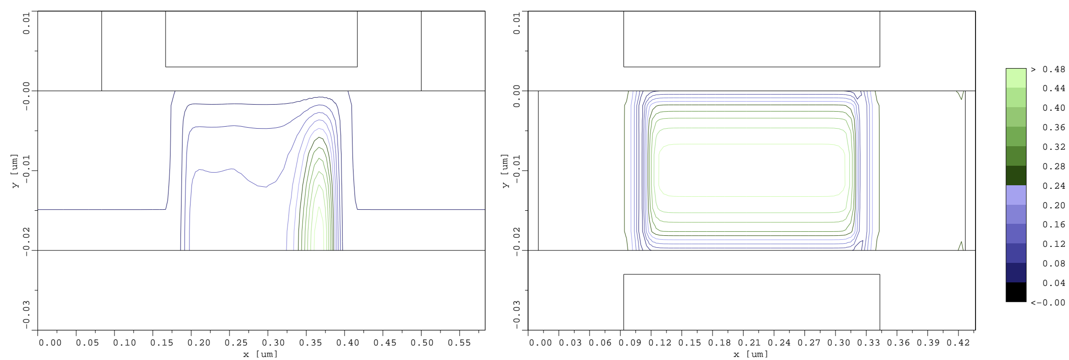


Figure 8.33: Inversion charge in a single-gate (left) and a double-gate (right) SOI MOSFET, illustrating the superior control of the inversion charge compared to traditional bulk-MOSFETs.

Appendix A

Partial Differential Equations

As we have seen in the introductory chapter, semiconductors can be described by a nonlinear set of coupled partial differential equations (PDEs), the drift-diffusion model given in (1.23), (1.24) and (1.25). In contrast to systems of *linear* equations, a more sophisticated solution procedure is required here, which must be carefully chosen depending on the type of the underlying PDEs. Governed by the structure of these equations, **archetypical**[1] behavior of the field quantities can be observed.

A.1 Boundary and Initial Conditions

In this lecture we will be dealing mainly with *partial differential equations* of the form

$$F(u, \partial u/\partial x_1, \partial u/\partial x_2, \partial u/\partial x_3, \ldots) = G(\boldsymbol{x}) \ .$$

It is a difficult topic to prove the existence of a solution of equations of this type. Similarly to ordinary differential equations, there are typically degrees of freedom in the solution of partial differential equations (provided it exists). To narrow the solution down, boundary and initial conditions must be provided. For the purpose of device simulation, usually a finite problem domain is assumed. Denoting the problem domain with Ω and its boundary with $\partial\Omega$, either the values of u at $\partial\Omega$ or the values of the normal derivative of u at $\partial\Omega$ may be given for a second order partial differential equation.

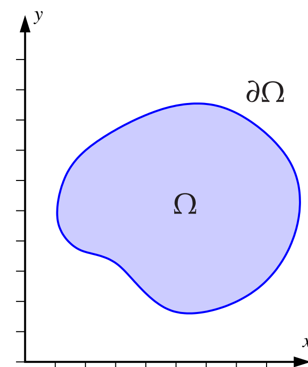


Figure A.1: A region Ω and its boundary $\partial\Omega$.

In the first case, the boundary condition is termed *Dirichlet condition*, and the whole problem is formulated as

> "For a given problem space Ω and a given function ϕ, find a suitable function u that fulfills $F(u, \partial u/\partial x_1, \partial u/\partial x_2, \partial u/\partial x_3, \ldots) = G(\boldsymbol{x})$ in the interior of Ω with $u(\boldsymbol{x}) = \phi(\boldsymbol{x})$ on the boundary $\partial\Omega$."

Dirichlet conditions are quite intuitive when considering for example Poisson's equation $\nabla^2\psi = \rho/\varepsilon$: The function u is then identified with the electrostatic potential ψ, the boundary $\partial\Omega$ is formed by electrodes, and the function ϕ is the voltage at these electrodes.

[1] **archetypical** [ɑː.kɪˈtaɪ.p.ᵊl]: urbildlich, musterhaft, typisch

The second important type are *Neumann conditions*, where the change of the field in the direction perpendicular to the boundary is given,

$$\frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}} := \boldsymbol{n} \cdot \nabla u(\boldsymbol{x}) = \phi(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \partial\Omega \,,$$

with the unit vector $\boldsymbol{n}$ locally perpendicular to $\partial\Omega$. Continuing the example from above, where u is the electrostatic potential ψ, prescribing the normal projection of ∇u effectively prescribes the electric *flux* coming out of Ω (remember, ∇u reflects the electric field, which is proportional to the flux). In most cases, $\partial u / \partial \boldsymbol{n} = 0$, i.e. *no flux* out of Ω at all, is prescribed. This is, however, in many cases true for infinitely large simulation domains only, while usually only finite simulation domains are realized numerically. As a **remedy**[1], a sufficiently large simulation domain is chosen with (at least to some extent) artificial homogeneous Neumann boundary conditions.

Be aware that it is not sufficient for Laplace's (and Poisson's) equation to prescribe Neumann boundary conditions only: For a solution $u_0(x)$ of the system

$$\begin{aligned} \nabla^2 u &= 0, && \text{in } \Omega, \\ \frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}} &= \phi(\boldsymbol{x}), && \text{on } \partial\Omega, \end{aligned}$$

every function $u(\boldsymbol{x}) = u_0(\boldsymbol{x}) + C$ with constant C is also a solution. Therefore, in order to assure uniqueness of a solution of Laplace's (or Poisson's) equation, Dirichlet boundary conditions on at least part of the boundary $\partial\Omega$ must be given.

As indicated, it is also possible to have the boundary $\partial\Omega$ decomposed into several (**disjoint**[2]) parts $\Gamma_1, \Gamma_2, \ldots \Gamma_r$ and impose boundary conditions of different types on each part:

$$\begin{aligned} u(\boldsymbol{x}) &= g_1(\boldsymbol{x}) && \text{on } \Gamma_1, \\ \frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}} &= g_2(\boldsymbol{x}) && \text{on } \Gamma_2, \\ &\vdots \\ u(\boldsymbol{x}) &= g_r(\boldsymbol{x}) && \text{on } \Gamma_r. \end{aligned}$$

Actually, this is the regular case; think of a simple parallel-plate capacitor: The electrodes, of course, form a Dirichlet boundary, while at the remaining boundaries the potential can not possibly be fixed. Instead, it is assumed that no electric flux leaves the problem space at these boundaries, an approximation which has less impact the farther the boundaries are away from the electrodes. Such a combination of boundary conditions is called *mixed boundary conditions*

There is a third type of boundary conditions possible, called *Robin conditions*. They can be considered a linear combination of Dirichlet and Neumann boundary conditions:

$$\alpha(\boldsymbol{x}) u(\boldsymbol{x}) + \beta(\boldsymbol{x}) \frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}} = \gamma(\boldsymbol{x}) \quad \text{on } \partial\Omega.$$

Depending on the problem formulation and the chosen solution method, it can be quite difficult to incorporate this type of boundary conditions into numerical simulations.

If the region Ω is unbounded (e.g. $\Omega = \mathbb{R}^3$), other conditions must be given, e.g. the value of $u(\boldsymbol{x})$ for $|\boldsymbol{x}| \to \infty$. But these unbounded problem spaces pose another problem when tackled by numeric simulation: How should an infinite region be represented in a finite computer

[1] **remedy** [rem.ə.di]: Abhilfe, Behelf [2] **disjoint** [dɪsˈdʒɔɪn.t]: disjunkt, einander nicht überschneidend

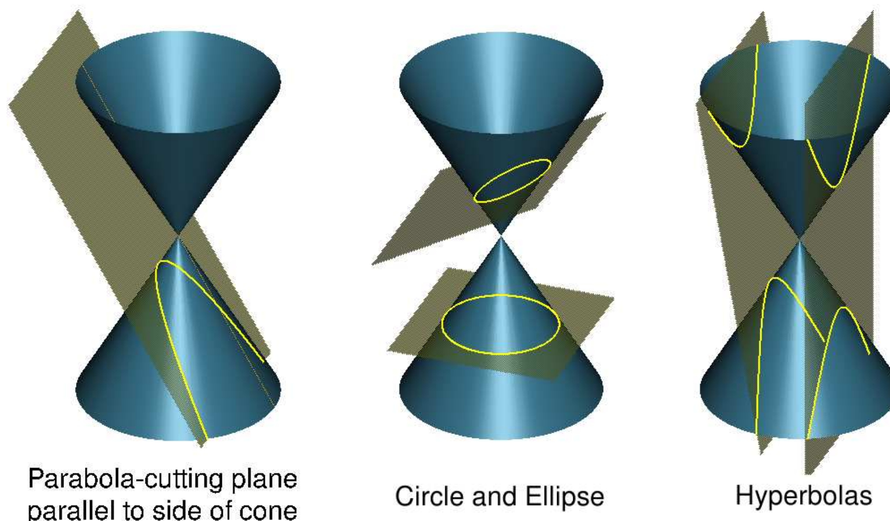


Figure A.2: Conic sections [?].

memory? Therefore, this case will not be discussed in this lecture (although, of course, there are methods around to solve even this kind of problem).

Note that if the stated problem is non-stationary (time-dependent), the time t is one of the independent variables x_i. In this case, the boundary condition is also termed *initial condition*. But from a mathematical point of view it is in no way special—time is a variable just as the spatial coordinates are.

A.2 Classification

Every linear, second order partial differential equation with constant coefficients belongs to one of three groups, depending on the coefficients in the equation. The members of these groups share a lot of common properties and have the same **peculiarities**[1] when it comes to discretization and numerical solution.

Specializing the general equation scheme from the beginning of this chapter to a linear, second order type in two variables x and y we have

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = G \quad , \tag{A.1}$$

where a through f are constants and G is a function in x and y. Now, every differentiation with respect to x is formally replaced by a multiplication with α, every $\partial/\partial y$ by β, yielding the **bivariate**[2] polynomial

$$P(\alpha, \beta) = a\alpha^2 + 2b\alpha\beta + c\beta^2 + d\alpha + e\beta + f \quad . \tag{A.2}$$

The equation $P(\alpha, \beta) = 0$ describes *conic sections*. Which conic section it describes is determined by the **discriminant**[3] $ac - b^2$: For $ac - b^2 > 0$ the equation describes an **ellipse**[4], for $ac - b^2 = 0$

[1] **peculiarity** [pɪˌkjuː.liˈer.ə.ţi]: Eigenheit, Ausprägung [2] **bivariate** [baɪvəriɑːt]: von zwei Variablen abhängig
[3] **discriminant** [diskrimɪnənt]: Diskriminante [4] **ellipse** [ɪˈlɪps]: Ellipse

a **parabola**[1] and for $ac - b^2 < 0$ a **hyperbola**[2]. For partial differential equations in three or more independent variables, the above scheme can be generalized analogously; the derivatives with respect to the different spatial coordinates represent the variables α, β and so on in our polynomial.

Moreover, if the coefficients a through f are allowed to be functions of x and y instead of constants, the type of the partial differential equation may change over the domain Ω, leading to additional numerical subtleties.

- *Elliptic Partial Differential Equations*: The elliptic type is the 'nicest' type in terms of sensitivity to discretization errors and numerical stability. A typical example is the Poisson equation $\nabla^2 u = \phi$. In two dimensions it has the form $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \phi$, giving the characteristic equation's coefficients $a = c = 1$ and $b = d = e = f = 0$, which results in a discriminant greater than zero. Elliptic differential equations describe stationary processes and hence do not have time derivatives.

- *Parabolic Partial Differential Equations*: The diffusion equation $\partial u/\partial t - \partial^2 u/\partial x^2 = \phi$ is a typical example for a parabolic differential equation. It contains a first order derivative with respect to time and a second order derivative with respect to the spatial coordinate, giving $a = b = e = f = 0$, $c = -1$, and $d = 1$, which results in a discriminant equal to zero. The combined drift-diffusion and continuity equations (1.24) and (1.25) on page 5 as well as the heat-flow equation (1.26) are examples for parabolic differential equation.

- *Hyperbolic Partial Differential Equations*: Describing **propagating**[3] 'distortions' like waves, the hyperbolic type is the most cumbersome for numerical simulation. First of all, in contrast to the parabolic type, where discontinuous distortions are smoothed out over time, discontinuities ('shocks') propagate *as they are* through a hyperbolic system. Secondly, the fact that propagation takes place lets us observe a preferred direction of propagation in the system, making the discretization more complicated. (Imagine a plane wave propagating in a medium. It obviously does not make sense to discretize the space along the direction of propagation in the same way as perpendicular to the direction of propagation.) Unfortunately, the hyperbolic differential equations associated with semiconductors may change their direction of propagation, requiring an adaptive discretization scheme.

  The homogeneous wave equation $\partial^2 u/\partial t^2 - \partial^2 u/\partial x^2 = 0$, with $a = 1$, $b = d = e = f = 0$, and $c = -1$ results in a discriminant of -1, and is therefore classified as a hyperbolic differential equation.

[1] **parabola** [pəræb.ᵊl.ə]: Parabel [2] **hyperbola** [haɪˈpɛː.bᵊl.ə]: Hyperbel [3] **to propagate** [prɔp.ə.geɪt]: ausbreiten, fortpflanzen

Appendix B

Vector Analysis and Its Implementation in SGFramework

One quantity of interest in the simulation of a semiconductor device is the electromagnetic field inside the device. This field is **governed**[1] by Maxwell's equations, which are coupled with the semiconductor equations. Since the mathematical representation of scalar and vectorial fields, i.e. mappings from $\mathbb{R}^3 \mapsto \mathbb{R}$ and $\mathbb{R}^3 \mapsto \mathbb{R}^3$, respectively, a few things you actually should remember from your mathematics- or electrodynamics lectures are summarized in this section.

B.1 Divergence

The *divergence* of a vector field $\boldsymbol{v}(\boldsymbol{r})$ is a scalar field, which, if $\boldsymbol{v}$ is interpreted as a physical flow, is a measure for the source density of this flow. If

$$\boldsymbol{v}(\boldsymbol{r}) = \begin{pmatrix} v_x(\boldsymbol{r}) \\ v_y(\boldsymbol{r}) \\ v_z(\boldsymbol{r}) \end{pmatrix} \quad , \qquad \boldsymbol{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad , \tag{B.1}$$

then in Cartesian coordinates in three dimensions,

$$\operatorname{div} \boldsymbol{v}(\boldsymbol{r}) = \frac{\partial v_x(\boldsymbol{r})}{\partial x} + \frac{\partial v_y(\boldsymbol{r})}{\partial y} + \frac{\partial v_z(\boldsymbol{r})}{\partial z} \quad . \tag{B.2}$$

An example is $\operatorname{div} \boldsymbol{D} = \rho$, where $\boldsymbol{D}$ is the electric flux density (electric displacement field) and ρ is the charge density.

The physical meaning of the divergence can be understood as follows: Imagine a spatial point $\boldsymbol{r}$ and a small volume $\Delta\mathcal{V}$ (with surface $\partial\Delta\mathcal{V}$) around $\boldsymbol{r}$. Moreover, imagine a vector field $\boldsymbol{A}$, defined at least in the point $\boldsymbol{r}$ and its vicinity. Then, the *flux* F of $\boldsymbol{A}$ leaving $\partial\Delta\mathcal{V}$ is

$$F = \int_{\partial\Delta\mathcal{V}} \boldsymbol{A} \cdot \mathrm{d}\boldsymbol{S} = \int_{\partial\Delta\mathcal{V}} \boldsymbol{A} \cdot \boldsymbol{n}\, \mathrm{d}S \quad . \tag{B.3}$$

Many semiconductor devices can be characterized by a two-dimensional layout, for example a MOSFET. This allows a two-dimensional simulation such that the quantities of interest in the real device are found by scaling along the third axis, because the variation along the third axis

[1] **to be governed** [gʌv.ᵊnd]: bestimmt sein

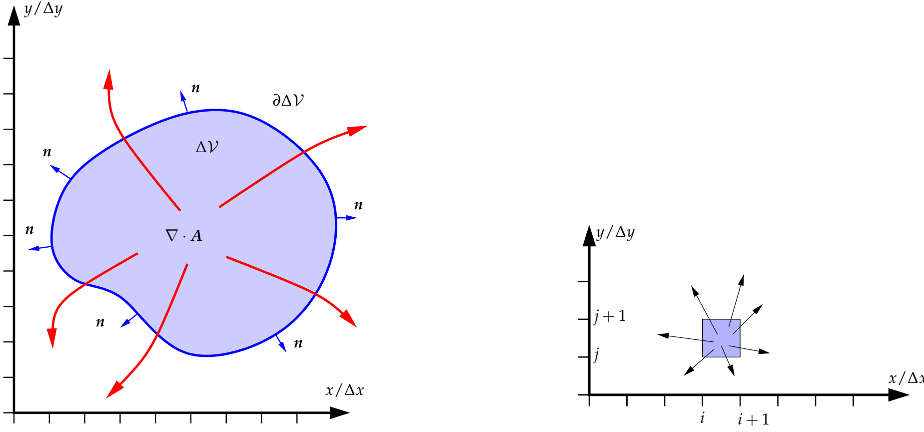


Figure B.1: Flux of vector field $\boldsymbol{A}$ through the surface $\partial\Delta\mathcal{V}$ (left) and its box discretization (right).

is small. In mathematical terms, we will compute an averaged flux $\bar{F}$ in two dimensions, which is linked to the true flux F by a multiplication with the width w along the third dimension:

$$\bar{F} := \frac{F}{w} = \int_{\partial\Delta\mathcal{V}_{2d}} \boldsymbol{A}\cdot\boldsymbol{n}\,\mathrm{d}s \ . \tag{B.4}$$

In order to discretize the divergence operator in two dimensions, we assume a box between $(i\Delta x, j\Delta y)$ and $((i+1)\Delta x, (j+1)\Delta y)$ (cf. Fig. B.1). With $\boldsymbol{A} = (A_x, A_y)^T$, the fluxes leaving the top and the bottom of the box are

$$\bar{F}^{\text{top}} = \int\limits_{i\Delta x}^{(i+1)\Delta x} A_y(x, y=(j+1)\Delta y)\,\mathrm{d}x \ , \qquad \bar{F}^{\text{bottom}} = -\int\limits_{i\Delta x}^{(i+1)\Delta x} A_y(x, y=j\Delta y)\,\mathrm{d}x \ . \tag{B.5}$$

The net flux leaving in y-direction thus is

$$\bar{F}^y = \bar{F}^{\text{top}} + \bar{F}^{\text{bottom}} = \int\limits_{i\Delta x}^{(i+1)\Delta x} (A_y(x, y=(j+1)\Delta y) - A_y(x, y=j\Delta y)]\,\mathrm{d}x \ . \tag{B.6}$$

For sufficiently small Δy we can approximate

$$A_y(x, y=(j+1)\Delta y) \approx A_y(x, y=j\Delta y) + \frac{\partial A_y}{\partial y}\Delta y \ . \tag{B.7}$$

Inserting (B.7) into (B.6) yields

$$\begin{aligned}\bar{F}^y &= \int\limits_{i\Delta x}^{(i+1)\Delta x} (A_y(x, y=(j+1)\Delta y) - A_y(x, y=j\Delta y))\,\mathrm{d}x \\ &\approx \int\limits_{i\Delta x}^{(i+1)\Delta x} \frac{\partial A_y}{\partial y}\Delta y\,\mathrm{d}x \\ &\approx \frac{\partial A_y}{\partial y}\Delta x\Delta y \ .\end{aligned} \tag{B.8}$$

Approaching the x-direction analogously,

$$\bar{F}^x \approx \frac{\partial A_x}{\partial x} \Delta x \Delta y \ , \tag{B.9}$$

and using $\Delta V = w \Delta x \Delta y$ we finally get

$$\frac{F^x + F^y}{\Delta V} = \frac{\bar{F}^x + \bar{F}^y}{\Delta x \Delta y} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} A_x \\ A_y \end{pmatrix} = \nabla \cdot \boldsymbol{A} \ . \tag{B.10}$$

We have just confirmed Gauss' theorem, which states the boundary integral can translated into an integral over the domain:

$$F = \int_{\partial \Delta \mathcal{V}} \boldsymbol{A} \cdot \boldsymbol{n} \, \mathrm{d}S = \int_{\Delta \mathcal{V}} \operatorname{div} \boldsymbol{A} \, \mathrm{d}V \ . \tag{B.11}$$

Thus, div $\boldsymbol{A}$ can be interpreted as a sink/source density. The above equation can be recast in a more physical setting as follows: The net number of particles leaving the volume $\Delta\mathcal{V}$ has to be equal to the number of particles generated within $\Delta\mathcal{V}$ to obtain an equilibrium situation.

As an example, for $\boldsymbol{A} = (3x - 0.2y^2)\boldsymbol{e}_x + (y + 4y^2)\boldsymbol{e}_y$ we get $\nabla \cdot \boldsymbol{A} = 4 + 8y$, displayed in the following program and visualized in Figure B.2.

```
1  const DIM = 30;          // number of mesh points in x and y
2  const DX = 0.2;          // mesh spacing in x
3  const DY = 0.1;          // mesh spacing in y
4  var    x[DIM], y[DIM], Ax[DIM,DIM], Ay[DIM,DIM],
5         divA[DIM-1,DIM-1];
6
7  begin main
8
9    // initialize the x and y components of vector A
10   assign   x [i=all]        = i*DX;
11   assign   y [j=all]        = j*DY;
12   assign   Ax[i=all,j=all]  = 3.0*x[i]-0.2*sq(y[j]);
13   assign   Ay[i=all,j=all]  = y[j]+4.0*sq(y[j]);
14
15   // compute the divergence and write the results
16   assign divA[i=all,j=all] = (Ax[i+1,j]-Ax[i,j])/DX +
17                              (Ay[i,j+1]-Ay[i,j])/DY;
18   write;
19 end
```

source_code/divergence_example.sg

B.2 Curl (Rotation)

Just like the divergence, the curl operator (aka. rot operator) is applied to vector fields; but it results in a vector field whose magnitude is a measure for the rate of rotation, and whose direction is perpendicular to the rotation plane of the operand.

Unlike the divergence, the curl operator is only meaningful for three-dimensional spaces,[1]

[1] For spaces of dimension $n > 3$, the framework of *alternating differential forms*, of which the vector calculus presented here is a special case, has to be employed.

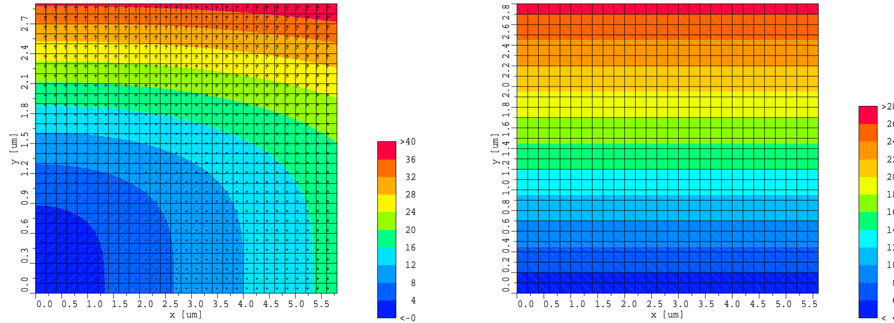


Figure B.2: The vector field $\boldsymbol{A}$ (left) and its divergence $\nabla \cdot \boldsymbol{A}$ (right).

where it is defined as

$$\operatorname{curl} \boldsymbol{v}(\boldsymbol{r}) = \begin{pmatrix} \dfrac{\partial v_z(\boldsymbol{r})}{\partial y} - \dfrac{\partial v_y(\boldsymbol{r})}{\partial z} \\ \dfrac{\partial v_x(\boldsymbol{r})}{\partial z} - \dfrac{\partial v_z(\boldsymbol{r})}{\partial x} \\ \dfrac{\partial v_y(\boldsymbol{r})}{\partial x} - \dfrac{\partial v_x(\boldsymbol{r})}{\partial y} \end{pmatrix} . \tag{B.12}$$

B.3 Gradient

While the curl and divergence operators act on vector fields, the *gradient* has to be used on scalar fields, returning a vector field that points in the direction of the **steepest**[1] **ascent**[2] of the operand. The definition reads

$$\operatorname{grad} f(\boldsymbol{r}) = \begin{pmatrix} \dfrac{\partial f(\boldsymbol{r})}{\partial x} \\ \dfrac{\partial f(\boldsymbol{r})}{\partial y} \\ \dfrac{\partial f(\boldsymbol{r})}{\partial z} \end{pmatrix} . \tag{B.13}$$

Program `exgrad.sg` gives a short example of how (B.13) is implemented in SGFRAMEWORK, while Fig. B.3 displays the result.

```
1  const DIM = 30;          // number of mesh points in x and y
2  const DX  = 1.0;         // mesh spacing in x
3  const DY  = 1.0;         // mesh spacing in y
4
5  var x[DIM], y[DIM], phi[DIM,DIM];
6  var gphix[DIM,DIM], gphiy[DIM,DIM], magphi[DIM,DIM];
7
8  const A = 1.0, B = 2.0, X0 = 50.0, Y0 = 50.0;
```

[1] **steep** [stiːp]: steil [2] **ascent** [əˈsent]: Anstieg

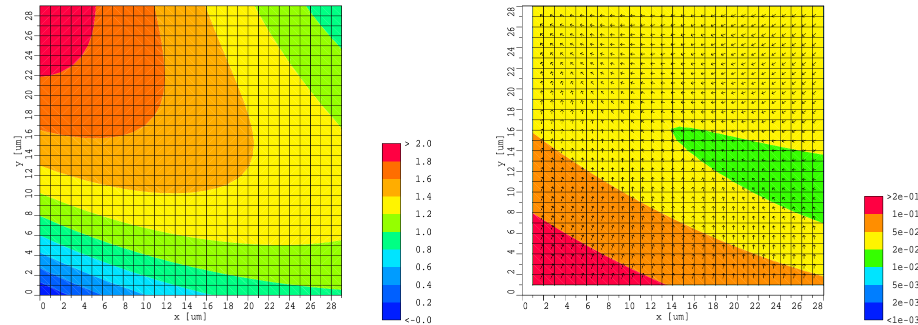


Figure B.3: Solution of `exgrad.sg`: Φ (left), $\nabla\Phi$ (right). The arrows in the right figure are perpendicular to isolines (i.e. the interfaces between different colors) on the left, showing the direction of steepest ascent.

```
 9  begin main
10    assign x[i=all] = i*DX;
11    assign y[j=all] = j*DY;
12    assign phi[i=all,j=all]=
13     {A*[1.0-sq(x[i]/X0-1.0)]+B*[1.0-sq(y[j]/Y0-1.0)]} *
14     {sq(x[i]/X0-1.0)+sq(y[j]/Y0-1.0)};
15
16    // determine the x and y components of phi and its magnitude
17    assign gphix[i=1..DIM-2,j=1..DIM-2] = (phi[i+1,j]-phi[i-1,j])/(2.0*DX);
18    assign gphiy[i=1..DIM-2,j=1..DIM-2] = (phi[i,j+1]-phi[i,j-1])/(2.0*DY);
19    assign magphi[i=all,j=all] = sqrt(sq(gphix[i,j])+sq(gphiy[i,j]));
20
21    // write the results
22    write;
23  end
```

source_code/grad_example.sg

B.4 Nabla

As an aid when manipulating expressions using the differential operators above, the linear differential operator *Nabla* is defined as

$$\nabla = \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} \end{pmatrix} . \tag{B.14}$$

Using the Nabla operator and the two basic product operations for vectors, the previously discussed operators can simply be written as

$$\operatorname{grad} f(\boldsymbol{r}) = \nabla f(\boldsymbol{r}) \ , \qquad \operatorname{div} \boldsymbol{v}(\boldsymbol{r}) = \nabla \cdot \boldsymbol{v}(\boldsymbol{r}) \ , \quad \text{and} \quad \operatorname{curl} \boldsymbol{v}(\boldsymbol{r}) = \nabla \times \boldsymbol{v}(\boldsymbol{r}) \ . \tag{B.15}$$

The fact that the curl is restricted to three-dimensional spaces can be **attributed**[1] to the fact that the cross product of two vectors is only defined for these spaces.

B.5 Manipulating Expressions

When dealing with more complicated expressions involving vector calculus operators, the Nabla-concept **comes in handy**[2]. But one has to be careful and keep in mind that Nabla is *both a vector and a differential operator*. From the former it follows that all identities known for vectors can be applied to expressions using Nabla, but the latter tells us that if such an identity involves rearranging the variables following Nabla, the product-rule of differentiation must be obeyed, since Nabla's differentiating nature acts on everything following it!

For example, if we were to simplify $\operatorname{div}(\boldsymbol{u} \times \boldsymbol{v})$ (the dependence on $\boldsymbol{r}$ will be suppressed from now on **for the sake of**[3] a **concise**[4] notation), we translate

$$\operatorname{div}(\boldsymbol{u} \times \boldsymbol{v}) = \nabla \cdot (\boldsymbol{u} \times \boldsymbol{v}) \tag{B.16}$$

and use the fact that the triple product $\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c})$ allows **cyclic**[5] permutation:

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b}) = \boldsymbol{b} \cdot (\boldsymbol{c} \times \boldsymbol{a}) = -\boldsymbol{b} \cdot (\boldsymbol{a} \times \boldsymbol{c}) = \cdots \ . \tag{B.17}$$

Before applying (B.17) to (B.16), however, one has to take care of Nabla's differentiating property, which says that a variable can only be moved out of Nabla's scope, i.e. pulled in front of Nabla, if it is constant (cf. the rule of ordinary differentiation, where $\mathrm{d}(cf) = c\mathrm{d}f$ if and only if c is constant). By writing $\nabla_{\boldsymbol{u}}$ and $\nabla_{\boldsymbol{v}}$ for Nablas that **act only upon**[6] $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively, and therefore treat all other variables as constants, the product rule of differentiation yields

$$\nabla \cdot (\boldsymbol{u} \times \boldsymbol{v}) = \nabla_{\boldsymbol{u}} \cdot (\boldsymbol{u} \times \boldsymbol{v}) + \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{u} \times \boldsymbol{v}) \ . \tag{B.18}$$

Now the vector identities from above can be used to rearrange the individual terms,

$$\nabla_{\boldsymbol{u}} \cdot (\boldsymbol{u} \times \boldsymbol{v}) = \boldsymbol{v} \cdot (\nabla_{\boldsymbol{u}} \times \boldsymbol{u}) \ , \qquad \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{u} \times \boldsymbol{v}) = -\boldsymbol{u} \cdot (\nabla_{\boldsymbol{v}} \times \boldsymbol{v}) \ , \tag{B.19}$$

leading to the result

$$\operatorname{div}(\boldsymbol{u} \times \boldsymbol{v}) = \boldsymbol{v} \cdot \operatorname{curl} \boldsymbol{u} - \boldsymbol{u} \cdot \operatorname{curl} \boldsymbol{v} \ . \tag{B.20}$$

Caveat[7]: This identity is part of the derivation of the lemma of Poynting. You may enjoy guessing what $\boldsymbol{u}$ and $\boldsymbol{v}$ stand for in this case, and how the expression is manipulated further. Of course, you may also look it up in the appropriate lecture notes or books, or even don't bother thinking about it at all—but that way, you'll miss a lot of fun!

[1] **to attribute** [ˈæt.rɪ.bjuːt]: zuschreiben, zurückführen auf [2] **to come in handy** [kʌm ɪn hæn.di]: sich gut treffen, gelegen kommen [3] **for the sake of** [fɔːʳ ðe seɪk ɔv]: um … Willen [4] **concise** [kənˈsaɪs]: übersichtlich [5] **cyclic** [ˈsai.klɪ.k]: zyklisch, periodisch [6] **act upon** [ækt əˈpɒn]: einwirken [7] **caveat** [ˈkæv.i.æt]: Hinweis, Warnung

B.6 Identities

The most important identity in vector calculus involves the curl of a gradient field,

$$\operatorname{curl}\operatorname{grad} f = \nabla \times (\nabla f) = \begin{pmatrix} \frac{\partial}{\partial y}\frac{\partial f}{\partial z} - \frac{\partial}{\partial z}\frac{\partial f}{\partial y} \\ \frac{\partial}{\partial z}\frac{\partial f}{\partial x} - \frac{\partial}{\partial x}\frac{\partial f}{\partial z} \\ \frac{\partial}{\partial x}\frac{\partial f}{\partial y} - \frac{\partial}{\partial y}\frac{\partial f}{\partial x} \end{pmatrix} \ . \tag{B.21}$$

If f is sufficiently smooth (which will be implicitly assumed, otherwise the whole analysis would be pointless), the lemma of Schwartz states that the order of differentiation in the mixed second partial derivatives is arbitrary. That means that all terms cancel. A gradient field therefore never has a curl component,

$$\operatorname{curl}\operatorname{grad} f = \nabla \times (\nabla f) \equiv 0 \ . \tag{B.22}$$

Even more important is the reverse statement: If a vector field is rotation-free, it can always be represented as the gradient of a suitable scalar field, called a *scalar potential*. The use of a scalar potential not only makes life a bit easier (remember, a scalar field is *one* function dependent on three coordinates, while a vector field is *three* functions dependent on three coordinates), but also ensures that the claim of being rotation-free is automatically fulfilled. A scalar potential is never unique, since the gradient of a constant field c vanishes, $\operatorname{grad}(f + c) = \operatorname{grad} f + \operatorname{grad} c = \operatorname{grad} f$, and therefore also $f + c$ is a valid potential.

A similar observation (with basically the same proof) can be made with the divergence of a curl field,

$$\operatorname{div}\operatorname{curl} \boldsymbol{v} = \nabla \cdot (\nabla \times \boldsymbol{v}) \equiv 0 \ . \tag{B.23}$$

This identity enables the representation of a field that is free of sources by the curl of a vector field, called its *vector potential*. In terms of complexity, nothing is gained, but again the requirement of being source-free is automatically fulfilled. Note that **analogous**[1] to the scalar potential, a vector potential can always be 'shifted': Following (B.22), the identity $\operatorname{div}\operatorname{curl}(\boldsymbol{v} + \operatorname{grad} f) = \operatorname{div}(\operatorname{curl} \boldsymbol{v} + \operatorname{curl}\operatorname{grad} f) = \operatorname{div}\operatorname{curl} \boldsymbol{v}$ shows that it is possible to add an arbitrary gradient field to a vector potential.

The last identity makes use of the rules stated in the last section. By noting

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{c} \ , \tag{B.24}$$

the 'double curl' $\operatorname{curl}\operatorname{curl} \boldsymbol{v}$ is transformed into

$$\operatorname{curl}\operatorname{curl} \boldsymbol{v} = \nabla \times (\nabla \times \boldsymbol{v}) = \nabla(\nabla \cdot \boldsymbol{v}) - (\nabla \cdot \nabla)\boldsymbol{v} = \nabla(\nabla \cdot \boldsymbol{v}) - \nabla^2 \boldsymbol{v} = \operatorname{grad}\operatorname{div} \boldsymbol{v} - \operatorname{div}\operatorname{grad} \boldsymbol{v} \ . \tag{B.25}$$

This identity can be used to derive the electromagnetic wave equations from Maxwell's equations. We won't make use of it, though.

B.7 Integral Theorems of Stokes and Gauss

The theorems of Stokes and Gauss are used to transform integrals of the divergence or curl of vector fields into integrals of the fields themselves. A particularly interesting point in these transformations is how the integration domain is transformed.

[1] **analogous** [əˈnæl.ə.gəs]: analog, entsprechend, sinngemäß

Consider a surface $\mathcal{A}$[1] and a vector field $\boldsymbol{v}$ defined at least on every point that is a member of $\mathcal{A}$. The surface $\mathcal{A}$ is confined by its boundary curve $\partial\mathcal{A}$; the field $\boldsymbol{v}$ needs to be defined on the boundary, too. When integrating over surfaces, the differential in the integrand is vector-valued; its magnitude is the area of the infinitesimal area element, and its direction is perpendicular to the plane spanned by the area element. Integrals along a curve also involve vector-valued differentials, but in this case their direction is the tangential to the curve. Denoting the differential area element by $\mathrm{d}\boldsymbol{A}$ and the differential curve element by $\mathrm{d}\boldsymbol{s}$, the theorem of Stokes states that

$$\int_{\mathcal{A}} \operatorname{curl} \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{A} = \int_{\mathcal{A}} (\nabla \times \boldsymbol{v}) \cdot \mathrm{d}\boldsymbol{A} = \int_{\partial\mathcal{A}} \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{s} \ . \tag{B.26}$$

Similarly, the integral theorem of Gauss states that the volume integral of a divergence is equal to the integral of the original field over the surface of the volume:

$$\int_{\mathcal{V}} \operatorname{div} \boldsymbol{v} \, \mathrm{d}V = \int_{\mathcal{V}} \nabla \cdot \boldsymbol{v} \, \mathrm{d}V = \int_{\partial\mathcal{V}} \boldsymbol{v} \cdot \mathrm{d}\boldsymbol{A} \tag{B.27}$$

[1] Be careful not to mix up the *surface* $\mathcal{A}$, which is an abstract geometrical object, and its *surface area* $A(\mathcal{A})$, which is a physical quantity, usually measured in square meters! The same applies to *volumes*.

Appendix C

Basics of Electromagnetism – Maxwell's Equations

Maxwell's equations describe the *structure* of the electromagnetic field. In differential form they are formulated as

$$\operatorname{curl}\boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \qquad \text{Faraday's Law of Induction} \tag{C.1}$$

$$\operatorname{curl}\boldsymbol{H} = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t} \qquad \text{Ampère's Circuital Law with Maxwell's Extension} \tag{C.2}$$

$$\operatorname{div}\boldsymbol{D} = \rho \qquad \text{Gauss' Law} \tag{C.3}$$

$$\operatorname{div}\boldsymbol{B} = 0 \qquad \text{Gauss' Law for Magnetism} \tag{C.4}$$

There is another form, the integral form, which some find easier to interpret, because in this form the equations are directly related to physically observable global quantities like charge, voltage and current. By using Stoke's integral theorem, Ampère's law is integrated over a surface $\mathcal{A}$ fixed in space to yield the magnetic voltage ψ_m

$$\psi_\mathrm{m}(\partial\mathcal{A}) = \int_{\partial\mathcal{A}} \boldsymbol{H}\cdot\mathrm{d}\boldsymbol{s} = \int_{\mathcal{A}} \operatorname{curl}\boldsymbol{H}\cdot\mathrm{d}\boldsymbol{A} = \int_{\mathcal{A}} \left(\boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t}\right)\,\mathrm{d}\boldsymbol{A} = I(\mathcal{A}) + \frac{\partial \Psi(\mathcal{A})}{\partial t} \quad . \tag{C.5}$$

The law of induction is transformed into

$$\psi(\partial\mathcal{A}) = \int_{\partial\mathcal{A}} \boldsymbol{E}\cdot\mathrm{d}\boldsymbol{s} = \int_{\mathcal{A}} \operatorname{curl}\boldsymbol{E}\cdot\mathrm{d}\boldsymbol{A} = -\int_{\mathcal{A}} \frac{\partial \boldsymbol{B}}{\partial t}\,\mathrm{d}\boldsymbol{A} = \frac{\partial \Phi(\mathcal{A})}{\partial t} \quad . \tag{C.6}$$

The two divergence equations thus read

$$\Psi(\partial\mathcal{V}) = \int_{\partial\mathcal{V}} \boldsymbol{D}\cdot\mathrm{d}\boldsymbol{A} = \int_{\mathcal{V}} \operatorname{div}\boldsymbol{D}\,\mathrm{d}V = \int_{\mathcal{V}} \rho\,\mathrm{d}V = Q(\mathcal{V}) \tag{C.7}$$

and

$$\Phi(\partial\mathcal{V}) = \int_{\partial\mathcal{V}} \boldsymbol{B}\cdot\mathrm{d}\boldsymbol{A} = \int_{\mathcal{V}} \operatorname{div}\boldsymbol{B}\,\mathrm{d}V = 0 \quad . \tag{C.8}$$

Maxwell's equations introduce six quantities: A flux density and field strength for the electric and magnetic part of the field, each, an electrical current density, and an electrical charge density. Therefore, to **unambiguously**[1] solve a problem a total of six equations is required. But

[1] **unambiguously** [ʌn.æmˈbɪg.ju.ə.sli]: eindeutig

Maxwell himself only provided four equations, and they are only 'half-equations' in the sense that they only assert properties of either the divergence- or curl part of the respective field, so we are in need for another four relations. Moreover, we have not considered the presence of matter so far, which is composed of a large number of charged particles that may either be bound to the matter's crystal structure or may possibly move around freely. All charge carriers, bound or free, are subject to the Lorentz force

$$\boldsymbol{F} = \mathrm{q}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \ , \tag{C.9}$$

where $\boldsymbol{v}$ is the velocity of the individual particle (referenced to the same coordinate system where $\boldsymbol{E}$ and $\boldsymbol{B}$ are referenced to). Depending on their bond state, they contribute to different quantities of the electromagnetic field.

The bound charge particles that **constitute**[1] matter (the protons and the electrons in the inner hulls) may be 'electromagnetically active' by contributing to the electrical *(polarization)* or magnetical *(magnetization)* flux. In simple matter (homogeneous, linear, and isotropic), the flux densities of the two field components are proportional to the respective field strengths,

$$\boldsymbol{D} = \varepsilon \boldsymbol{E} \quad \text{and} \quad \boldsymbol{B} = \mu \boldsymbol{H} \ . \tag{C.10}$$

Bound charge carriers inserted artificially cause a **net**[2] charge of the body, which is represented by a non-vanishing charge density ρ.

The free charges are the ones responsible for conduction. The Lorentz force fully acts on them, but frequent collisions with other particles ensure that their velocity does not exceed a certain material-dependent limit. In simple matter, this behavior is expressed as a linear relation between the electric field strength and the electric current density, which is known as *Ohm's law:*

$$\boldsymbol{J} = \sigma \boldsymbol{E} \ . \tag{C.11}$$

Note that depending on the properties of the matter in question, other transport mechanisms may need to be added to the model, effectively *replacing* Ohm's law. The semiconductors the whole lecture is about are an example for this.

C.1 Interface Conditions

When considering the interface of two adjacent bodies with different material parameters, it is obvious that not all field quantities can be continuous; e.g. consider two bodies composed of simple matter in the sense of the previous section having different permittivities. Since in both bodies the respective linear relations between $\boldsymbol{D}$ and $\boldsymbol{E}$ are valid, both fields can not be continuous simultaneously across the interface.

The remedy is to allow step-like discontinuities in the fields at the interface. Clearly, at these discontinuities the fields are not differentiable any more, but the integral formulation of Maxwell's equations provides clues on how the discontinuities behave. When integrating over a surface or volume that includes an interface, the integral is divided in two parts, one 'before' and one 'after' the interface, and the integral theorems of Stokes and Gauss are extended by expressions that handle the discontinuities at the interface. For the electric field, the relations are

$$\boldsymbol{D}_2 \cdot \boldsymbol{n} - \boldsymbol{D}_1 \cdot \boldsymbol{n} = \rho_\mathrm{s} \quad \text{and} \quad \boldsymbol{E}_2 \times \boldsymbol{n} - \boldsymbol{E}_1 \times \boldsymbol{n} = 0 \ , \tag{C.12}$$

[1] **to constitute** [ˈkɑːn.stɪ.tuːt]: bilden, ausmachen [2] **net** [net]: netto (auch: Netz)

where $\boldsymbol{n}$ is the unit vector locally perpendicular to the interface when **traversing**[1] from body '1' (with $\boldsymbol{D}_1$ and $\boldsymbol{E}_1$) to body '2' (with $\boldsymbol{D}_2$ and $\boldsymbol{E}_2$). The equations state that the electric flux component *perpendicular to the interface* has a discontinuity through the sheet charge density ρ_s at the interface,[2] and the electric field strength's component *tangential to the interface* is continuous. The respective other component (the tangential flux and the perpendicular field strength) can be calculated using the respective permittivity relation in the body in question,

$$\boldsymbol{D}_{1,2} \times \boldsymbol{n} = \varepsilon_{1,2}\boldsymbol{E}_{1,2} \times \boldsymbol{n} \quad \text{and} \quad \boldsymbol{E}_{1,2} \cdot \boldsymbol{n} = \frac{1}{\varepsilon_{1,2}}\boldsymbol{D}_{1,2} \cdot \boldsymbol{n} \ . \tag{C.13}$$

C.2 Continuity Equation

One of the most fundamental axioms in physics is the assumption that electric charges can not be generated or destroyed. Being a prerequisite to Maxwell's equations, it must be possible to deduce it from them. In fact, taking the divergence of Ampère's Law,

$$\operatorname{div}\operatorname{rot}\boldsymbol{H} = \operatorname{div}\boldsymbol{J} + \operatorname{div}\left(\frac{\partial \boldsymbol{D}}{\partial t}\right) = \operatorname{div}\boldsymbol{J} + \frac{\partial \operatorname{div}\boldsymbol{D}}{\partial t} = \operatorname{div}\boldsymbol{J} + \frac{\partial \rho}{\partial t} \ . \tag{C.14}$$

But at the other hand, $\operatorname{div}\operatorname{rot} \equiv 0$, and therefore we get the so called *charge continuity equation*

$$\operatorname{div}\boldsymbol{J} = -\frac{\partial \rho}{\partial t} \ . \tag{C.15}$$

[1] **to traverse** [trəˈvɜːs]: überschreiten, durchlaufen [2] Since at interfaces the crystal structure of matter is perturbed, a net charge is assumed to be present there in general.

Appendix D

Vocabulary

abbreviation	[ə.briː.viˈeɪ.ʃᵊn]	Abkürzung, Kurzwort
accordance	[əˈkɔːr.dᵊnts]	Übereinstimmung
accurate	[æk.jʊ.rət]	genau
act upon	[ækt əˈpɒn]	einwirken
adjacent	[əˈdʒeɪ.sᵊnt]	benachbart, angrenzend
aka, also known as	[ɔːlsou noʊŋ æz]	auch bekannt unter, so genannt
analogous	[əˈnæl.ə.gəs]	analog, entsprechend, sinngemäß
arbitrary	[ˈɑːrbətreri]	willkürlich, beliebig
archetypical	[ɑː.kɪˈtaɪ.p.ᵊl]	urbildlich, musterhaft, typisch
arsenic	[ˈɑː.sᵊn.ɪk]	Arsen
ascent	[əˈsent]	Anstieg
assumption	[əˈsʌmp.ʃᵊn]	Annahme
big picture	[bɪg pɪk.tʃɚ]	Das große Ganze, ein erster Einstieg
bivariate	[baɪvəriɑːt]	von zwei Variablen abhängig
boron	[ˈboʊrɔn]	Bor
brevity	[ˈbrev.ə.t̬i]	Kürze
calculation	[kæl.kjʊˈleɪ.ʃᵊn]	Berechnung
Cartesian	[kɑːˈti.zi.ən]	kartesisch
caveat	[ˈkæv.i.æt]	Hinweis, Warnung
cf, to confer	[kənˈfɜːʳ]	vergleichen, konsultieren
concise	[kənˈsaɪs]	übersichtlich
contradictory	[kɒn.trəˈdɪk.tᵊr.i]	widersprüchlich
counterclockwise	[kaʊn.təˈklok.waɪz]	gegen den Uhrzeigersinn
crude	[kruːd]	grob, ungehobelt
cyclic	[ˈsai.klɪ.k]	zyklisch, periodisch
Delaunay	[deloʊneɪ]	Delaunay
deliberately	[dɪˈlɪb.ᵊr.ət.li]	absichtlich
denominator	[dɪˈnɑː.mə.neɪ.t̬ɚ]	Nenner
derivative	[dɪˈrɪv.ə.t̬ɪv]	Ableitung
deviation	[diː.viˈeɪ.ʃᵊn]	Abweichung

dipole	[ˈdaɪpoʊl]	Dipol
discretization	[dɪˈskriːtɪˈseɪ.ʃᵊn]	Diskretisierung
discriminant	[diskrimɪnənt]	Diskriminante
disjoint	[dɪsˈdʒɔɪn.t]	disjunkt, einander nicht überschneidend
distinction	[dɪˈstɪŋk.ʃᵊn]	Unterscheidung
distribution	[dɪˈstrɪb.juː.ʃᵊn]	Verteilung
dumb	[dʌm]	einfältig, primitiv
electron	[ɪˈlektrɑːn], NOT [ˈəlektrɑːn]	Elektron
ellipse	[ɪˈlɪps]	Ellipse
encroachment	[ɪnˈkrəʊtʃ.mənt]	Beeinträchtigung
evenly	[iː.vᵊn.li]	gleichmäßig
excess	[ekˈses]	überschüßig
expansion	[ikˈspæn.tʃᵊn]	hier: Entwicklung
familiar	[fəˈmɪl.i.jɚ]	vertraut, geläufig
fictitious	[fɪkˈtɪʃ.əs]	fiktiv
for the sake of sth.	[fɔːʳ ðe seɪk ɔv]	um ... Willen
handy	[hæn.di]	praktisch, geschickt
hyperbola	[haɪˈpɛː.bᵊl.ə]	Hyperbel
impurity	[ɪmˈpjʊəriti]	Störstelle, Störatom
inertia	[ɪˈnɜː.ʃə]	Trägheit
inexhaustible	[ɪn.ɪgˈzɔː.stɪ.bl̩]	unerschöpflich
intersection	[ɪn.təˈsek.ʃᵊn]	Schnittmenge, Schnittpunkt
ionic conductor	[aɪˈɔn.ɪk kənˈdʌk.tɚ]	Ionenleiter
junction	[ˈdʒʌŋkʃᵊn]	die Sperrschicht
Laplacian	[laplasɪæn]	Laplace-Operator
lattice	[læt.ɪs]	Kristallgitter
modulus	[mɔd.juː.ləs]	Absolutbetrag
negligible	[ˈneg.lɪ.dʒə.bl̩]	vernachlässigbar
net	[net]	netto (auch: Netz)
no matter	[noʊ ˈmæt̬.ɚ]	ganz egal
obsolete	[ɔb.sᵊlˈiːt]	hinfällig
on closer inspection	[ɔn kloʊsər ɪnˈspek.ʃᵊn]	bei näherer Betrachtung
overshoot	[əʊ.vəˈʃuːt]	die Überhöhung
painstakingly	[peɪnzˌteɪ.kiŋ.li]	sorgfältig
parabola	[pəræb.ᵊl.ə]	Parabel
peculiarity	[pɪˌkjuː.liˈer.ə.t̬i]	Eigenheit, Ausprägung
phosphorus	[fɔs.fᵊr.əs]	Phosphor
pitfall	[pɪt.faːl]	Fallgrube, Fallstrick, Fehler

plethora	[ˈpleθ.ɚ.ə]	Fülle, Vielzahl
preceding	[prɪˈsiː.dɪŋ]	vorangegangen
preface	[ˈprefɪs], NOT [ˈprɪːfeɪs]	Vorwort
preliminary	[prɪˈlɪm.ɪ.nᵊr.i]	vorläufig, vorübergehend
prerequisite	[priːˈrek.wɪ.zɪt]	Voraussetzung, Bedingung
principal minors	[prɪnt.sɪ.pᵊl maɪ.nəʳ]	Hauptminoren
pronunciation	[prəˌnʌntsiˈeɪʃᵊn]	Aussprache
quantity	[ˈkwɑːn.t̬ə.t̬i]	Größe, hier speziell: Matrixelemente
reentrant corners	[riː.en.trənt]	einspringende Ecken
remedy	[rem.ə.di]	Abhilfe, Behelf
root	[ruːt]	Wurzel, Ursprung, hier: Nullstelle
roughness	[rʌf.nəs]	Rauheit, Unebenheit
rudimentary	[ruːdɪˈmen.tᵊr.i]	elementar
satisfactory	[sæt.ɪsˈfækt.ᵊr.i]	zufriedenstellend
solely	[soʊl.li]	lediglich
sound	[saʊnd]	auch: vernünftig
sophistication	[səˌfɪs.tɪˈkeɪ.ʃᵊn]	Raffinesse
steep	[stiːp]	steil
stencil	[stent.sᵊl]	Vervielfältigungsmatrix
straightforward	[streɪtˈfɔː.wəd]	unkompliziert
subtlety	[sʌt.l.ti]	Schwierigkeit, Raffinesse
sufficient	[səf.ɪʃ.ᵊnt]	ausreichend
tessellation	[tes.ᵊlˈeɪ.ʃᵊn]	Mosaik
throughout	[θruːˈaʊt]	durchweg, hindurch
to accomplish	[əˈkɑːm.plɪʃ]	etwas erreichen, etwas vollbringen
to attribute	[ˈæt.rɪ.bjuːt]	zuschreiben, zurückführen auf
to be governed	[gʌv.ᵊnd]	bestimmt sein
to be subjected to sth.	[sʌb.dʒekt]	etwas ausgesetzt werden
to cope	[koʊp]	zurechtkommen, beherrschen
to coincide	[koʊ.ɪnˈsaɪd]	übereinstimmen
to come in handy	[kʌm ɪn hæn.di]	sich gut treffen, gelegen kommen
to constitute	[ˈkɑːn.stɪ.tuːt]	bilden, ausmachen
to decompose	[diː.kəmˈpəʊz]	aufteilen, spalten
to decouple	[dɪkʌp.l]	entkoppeln
to deplete	[dɪˈpliːt]	verringern, aufbrauchen, verarmen
to determine	[dɪˈtɛː.mɪn], NOT [determaɪn]	bestimmen, festlegen
to devote	[dɪˈvoʊ.t̬ɪd]	widmen
to diminish	[dɪˈmɪn.iʃ]	abnehmen, abklingen
to disturb	[dɪˈstɜːb]	stören, durcheinanderbringen
to elevate	[el.ɪ.veɪt]	emporheben, erhöhen
to employ	[ɪmˈplɔɪ]	einführen, einsetzen
to emphasize	[emp.fə.saɪz]	betonen, hervorheben
to encourage	[ɪnˈkʌr.ɪdʒ]	animieren, ermuntern
to evaluate	[ɪˈvæl.ju.eɪt]	etwas auswerten

to justify	[dʒʌs.tɪ.faɪ]	rechtfertigen
to neglect	[nɪˈglekt]	vernachlässigen
to omit	[oʊˈmɪt]	auslassen
to pay attention to sth.	[peɪ ɛˈten.tʃᵊn]	auf etwas achten
to perpendicular bisect	[pɛː.pᵊnˈdɪk.jʊ.lər baɪˈsekt]	in zwei gleich große Teile teilen
to prevail	[prɪˈveɪl]	überwiegen, vorherrschen
to prohibit	[prəˈhɪb.ɪt]	verhindern, unterbinden
to propagate	[prɔp.ə.geɪt]	ausbreiten, fortpflanzen
to provoke	[prəˈvoʊk]	auslösen, bewirken
to readdress	[ˌriːəˈdres]	sich nocheinmal zuwenden
to reason	[riː.zᵊn]	begründen, überlegen
to recover	[rɪˈkʌv.ɚ]	zurückgewinnen, wiedererlangen
to resort to sth.	[rɪˈzɔːrt]	auf etw. zurückgreifen
to scatter	[skæt̬.ɚ]	streuen, zerstreuen
to screen	[skriːn]	abschirmen, schützen, filtern
to seek sth.	[siːk]	etwas suchen
to surmount	[səˈmaʊnt]	bewältigen, überwinden
to traverse	[trəˈvɜːs]	überschreiten, durchlaufen
to vanish	[væn.ɪʃ]	verschwinden
to yield sth.	[jiːld]	etwas ergeben, etwas hervorbringen
tractable	[træk.tə.bl̩]	handhabbar, lenkbar
truncation error	[trʌŋˈkeɪ.ʃᵊn ˈer.ɚ]	Abschneidefehler
unambiguously	[ʌn.æmˈbɪg.ju.ə.sli]	eindeutig
unfeasible	[ʌnˈfiː.zɪ.bl̩]	undurchführbar
vacancy	[veɪ.kᵊnt.si]	freie Stelle, insbes. Gitterfreistelle
vicinity	[vəˈsɪnət̬i]	Umgebung
worth noticing	[wɜːθ noʊ.t̬ɪsɪŋ]	erwähnenswert

Bibliography

[1] D.M. Caughey and R.E. Thomas. Carrier mobilities in silicon empirically related to doping and field. *Proceedings of the IEEE*, 55(12):2192–2193, Dec. 1967.

[2] J.-P. Colinge and C. A. Colinge. *Physics of Semiconductor Devices*. Springer Science+Business Media, Inc., New York, 2002.

[3] Conic Sections — Wikipedia. http://en.wikipedia.org/wiki/Conic_sections.

[4] R. N. Hall. Electron-Hole Recombination in Germanium. *Phys. Rev.*, 87(2):387, Jul 1952.

[5] R. Jaggi. High-Field Drift Velocities in Silicon and Germanium. *Helvetia Physica Acta*, 42:941–943, 1969.

[6] R. Jaggi and H. Weibel. High-Field Electron Drift Velocities and Current Densities in Silicon. *Helvetia Physica Acta*, 42:631–632, 1969.

[7] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling*. McGraw-Hill Professional, 1998.

[8] E. H. Nicollian and J. R. Brews. *MOS (Metal Oxide Semiconductor) Physics and Technology*. Wiley-Interscience, 1982.

[9] C. T. Sah. Characteristics of the metal-Oxide-semiconductor transistors. *IEEE Transactions on Electron Devices*, 11:324–345, 1964.

[10] D. L. Scharfetter and H. K. Gummel. Large-SignalAnalysis of a Silicon Read Diode Oscillator. *IEEE Transactions on Electron Devices*, 16:64–77, 1969.

[11] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer-Verlag, New York, 1984.

[12] W. Shockley and W. T. Read. Statistics of the Recombinations of Holes and Electrons. *Phys. Rev.*, 87(5):835–842, Sep 1952.

[13] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Mathematics Series. Wadsworth & Brooks/Cole, Pacific Grove, California, 1989.

[14] Simon M. Sze. *Physics of Semiconductor Devices*. Wiley-Interscience, November 1981.

[15] W. V. Van Roosbroeck. Theory of Flow of Electrons and Holes in Germanium and other Semiconductors. *Bell Syst. Techn. J.*, 29:560–607, 1950.