

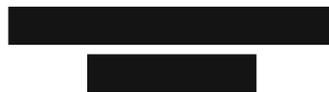
DISSERTATION

Three-Dimensional Mesh Generation for Device and Process Simulation

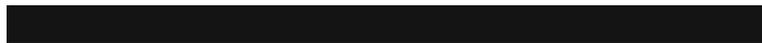
ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
von

Johann Cervenka



Matrikelnummer 8826391



Wien, im Oktober 2004

Abstract

Due to the progressive miniaturization of integrated circuits the exact and fast simulation of physical processes becomes more and more important. On the one hand, due to the miniaturization, parasitic effects begin to influence the device characteristics. Therefore, existing models must be extended. On the other hand, due to the reduction of the device dimension, a limited spatial expansion in the third dimension cannot be neglected any longer. At geometry corners, parasitic effects can occur and may even dominate the device behavior severely. Since these effects are not describable in two dimensions, the models must be extended to three dimensions and the development and simulation tools have to be adapted to three-dimensional requirements. The extension of the simulation tools involves the extension of the simulators, which contains the adaption for three-dimensional structure descriptions and the embedding of the new and extended models. Additionally, the mesh generators must be extended for three dimensions. Since the necessary amount of data and computing time that is needed for the simulation increases enormously, it is inevitable to adapt the simulation meshes to the given requirements in order to obtain accurate simulation results even with limited resources. As will be shown this task is far from trivial.

In the context of this work the issue of three-dimensional mesh generation for specific simulation problems in microelectronics is outlined. Chapter 4 describes the development of a method for the fast computation of diffusion processes in simple semiconductor structures. Since diffusion processes are transient procedures, an exact three-dimensional simulation needs a set of simulations on partially very dense meshes. Therefore the required computing time can increase enormously. With the developed method, a Green's Function approach of the diffusion equation can be used if complicated nonlinear models can be neglected, which offers the advantage to supply the final diffusion profile after only one simulation step. It can be performed without the frequent computation of only temporarily used distributions. Moreover, with this method the result is independent of the simulation mesh. Since the meshes, holding the initial and the final distribution, are independent, both of them can be adapted to their respective requirements.

An adapted Delaunay mesh generation approach for the electrical simulation of semiconductor devices is described in the following chapter. Since particularly with the simulation of MOS transistors a very high resolution of the mesh is necessary below the gate oxide, global mesh refinement methods are impracticable due to high resource consumption. The frequently used ortho grids, where an anisotropic grid density can be relatively easily obtained, cannot be used with non-axes-parallel and non-planar geometries. With the developed method, the grid points are placed along computed equipotential surfaces. Since the positive characteristics of the ortho grids are preserved, the mesh lines near the surface match the contours of the geometry edges and no restriction on planar structures exists. Along these equipotential faces a high point density can be selected within desired regions. A further advantage of this method is that the point density can be tuned in relation to the direction, along three almost orthogonal axial directions, which results in controllable anisotropy. Similar procedures are well-known as elliptical grid generation.

Finally, the relevance of each topic is clarified by accomplished applications. The boundary region of a power field-effect transistor is optimized concerning its electrical characteristics with the help of the diffusion simulation. The developed potential method is used for the development of a simulation mesh of a FinFET structure. A final example shows a complete process simulation of an EEPROM memory cell, whereby also the potential method is used in a somewhat modified operational area.

Kurzfassung

Durch die fortschreitende Miniaturisierung integrierter Schaltkreise nimmt die genaue und schnelle Simulation physikalischer Vorgänge einen immer höheren Stellenwert ein. Einerseits wirken sich aufgrund der Miniaturisierung parasitäre Effekte in verstärktem Maße aus und bestehende Modelle müssen dahingehend erweitert werden, andererseits kann durch die Verkleinerung der Bauteilabmessungen eine beschränkte räumliche Ausdehnung in die dritte Raumrichtung nicht mehr vernachlässigt werden. Ebensolche parasitäre Effekte können genau an Geometriecken auftreten und das Bauteilverhalten gravierend beeinflussen. Da diese Effekte nicht zweidimensional beschreibbar sind, müssen sowohl die Modelle auf drei Dimensionen erweitert werden, als auch die Entwicklungs- und Simulationswerkzeuge den dreidimensionalen Anforderungen angepasst werden. Die Erweiterung der Simulationswerkzeuge umfasst die Erweiterung der eigentlichen Simulatoren, die ebenfalls für dreidimensionale Strukturbeschreibungen angepasst werden müssen und in denen die neuen und erweiterten Modelle eingebettet werden müssen. Darüberhinaus müssen auch die Gittergeneratoren für drei Dimensionen erweitert werden. Da das erforderliche Datenvolumen und auch die zur Simulation benötigten Rechenzeiten enorm anwachsen, ist es unumgänglich, die Rechengitter den gegebenen Anforderungen anzupassen, um bei eingeschränkten Ressourcen dennoch genaue Simulationsergebnisse zu erzielen.

Im Rahmen dieser Arbeit wurde speziell auf die dreidimensionale Gittergenerierung für spezielle Anwendungsgebiete der Simulation in der Mikroelektronik eingegangen. In Kapitel 4 wird ein entwickeltes Verfahren zur schnellen Berechnung von Diffusionsprozessen in einfachen Halbleiterstrukturen vorgestellt. Da Diffusionsprozesse transiente Vorgänge sind, benötigt eine genaue dreidimensionale Simulation eine Reihe von Einzelsimulationen auf teilweise sehr dichten Rechengittern und die Rechenzeiten können enorm anwachsen. Beim entwickelten Verfahren wird auf komplizierte nichtlineare Modelle verzichtet. Die Berechnung erfolgt anhand Green'scher Funktionen der Diffusionsgleichung, die den Vorteil besitzen, einen Endwert des Diffusionsprozesses nach nur einem Simulationsschritt liefern zu können. Es kann auf die wiederholte Auswertung zeitlich diskretisierter Diffusionsvorgänge verzichtet werden. Desweiteren konnte bei diesem Verfahren eine gewisse Unabhängigkeit

des Ergebnisses vom Rechengitter erzielt werden. Da Anfangsverteilung und Endergebnis auf voneinander unabhängigen Gittern vorliegen dürfen, können auch beide ihren jeweiligen Anforderungen angepasst werden.

Der zweite Teil der Arbeit beschäftigt sich mit der Delaunay Gittergenerierung für die Simulation elektronischer Halbleiterbauteile. Da speziell bei der Simulation von MOS Transistoren eine sehr hohe Gitterauflösung unter dem Gateoxid notwendig ist, versagen globale Gitterverfeinerungsmethoden ebenfalls aufgrund des hohen Ressourcenverbrauchs. Ebenso versagen die vielfach verwendeten Orthogitter, bei denen eine anisotrope Gitterdichte relativ leicht eingestellt werden kann, weitläufig bei nichtachsenparallelen und nichtplanaren Geometrien. Bei der hier entwickelten Methode werden die Gitterpunkte entlang berechneter Isopotenzialflächen gesetzt. Hierdurch wird erreicht, dass die positiven Eigenschaften der Orthogitter erhalten bleiben, sich die oberflächennahen Gitterlinien den Geometriekanten anschmiegen und keine Einschränkung auf planare Strukturen besteht. Entlang dieser Isopotenzialflächen können nun in gewünschten Bereichen die Punktdichten höher gewählt werden. Ein weiterer Vorteil dieser Methode ist, dass die Punktdichte auch richtungsabhängig, entlang dreier nahezu normal stehender Achsenrichtungen, das heißt anisotrop, eingestellt werden kann. Allgemein sind ähnliche Verfahren als elliptische Gittergenerierung bekannt.

Abschließend ist jeder Themenbereich anhand eines praktisch durchgeführten Beispiels verdeutlicht. Der Randbereich eines Leistungsfeldeffekttransistors wird unter Zuhilfenahme der Diffusionssimulation bezüglich seiner elektrischen Eigenschaften optimiert. Die entwickelte Potenzialmethode findet anhand der Entwicklung eines Rechengitters einer MOS Transistorstruktur Anwendung. Ein abschließendes Beispiel zeigt eine komplette Prozesssimulation einer EEPROM Speicherzelle mit den entstehenden Gittern, wobei auch die Potenzialmethode in einem etwas abgewandelten Einsatzgebiet zur Anwendung kommt.

Acknowledgment

First and foremost I have to thank the people from the Institut für Mikroelektronik. Not only for conveying the experience and knowledge to solve my work, also for forming new friendships. Foremost, my gratitude goes to Prof. Siegfried Selberherr who directed me to my thesis. With the required strength he showed me the way to finish my work. Thanks to Ewald, Manfred, and Frau Winkler for the excellent infrastructure.

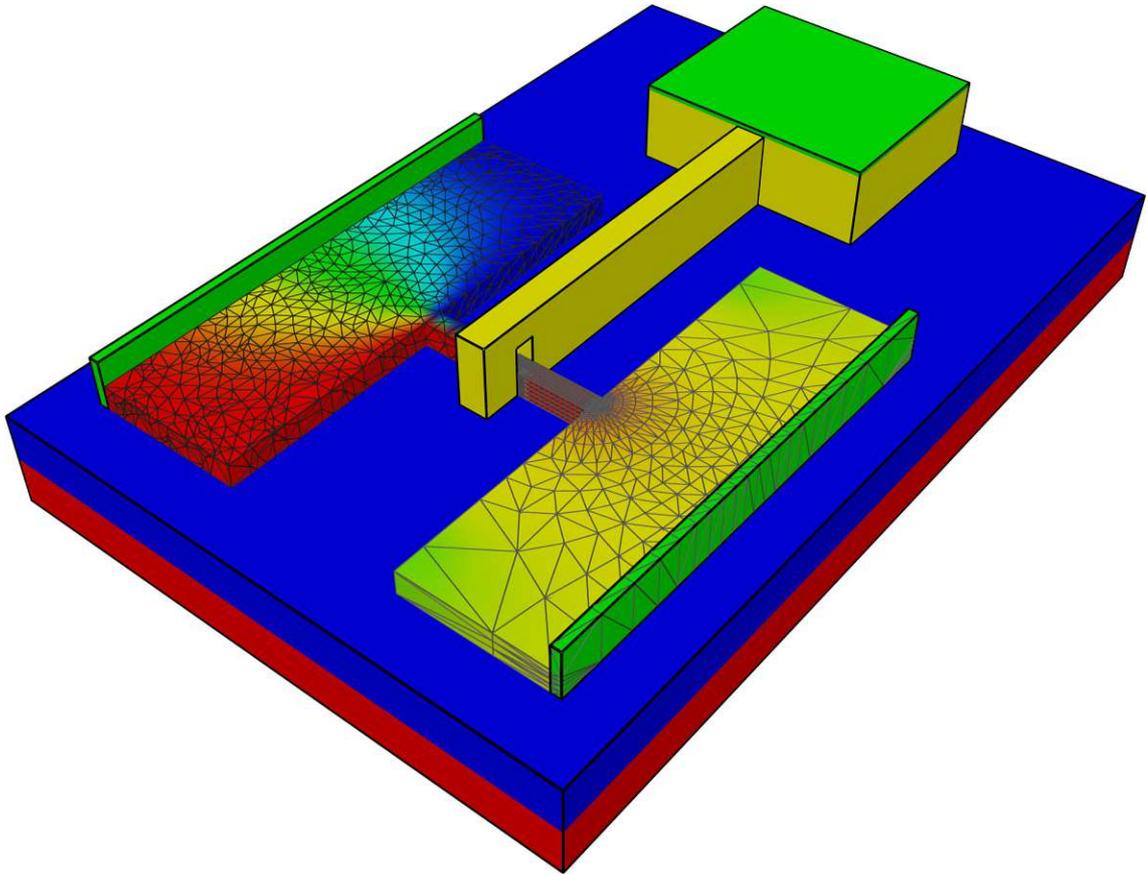
I am very grateful to the Christian Doppler Forschungsgesellschaft, Wien, Austria as well as our industrial partners, Austriamicrosystems AG, Unterpremstätten, Austria and Infineon Technologies Villach AG, Villach, Austria, for funding my studies. Beyond the context of this partnership, it was possible to build up a friendly working atmosphere. Thank also goes to Prof. Tibor Grasser who led me during parts of the work. Thanks for the guidance concerning the concept of my thesis. Thanks to all those who spent a lot of time with proofreading. Especially I want to thank Prof. Herbert Haas for the inspiring advice, which helped to improve the mathematical quality of the work.

During my work, I found a lot of friends at the institute. Primary thanks to my former room colleagues, Saba, Highlander, Andy, Wolfi, and Robert. We spent nice times — not mandatorily at the institute. Thanks to the vi- and scriptking, adminking, networktestking, and mathking, for the inspiring high-level talks, problem solutions, mathematical discussions, nightly sessions, and culinary support. Naturally, also to all the colleagues I made friends with.

Special regards to all my friends, who watched me many years studying. And last but not least I want to thank my parents, my brother, and my relatives for supporting me during the whole time of my long-lasting education.

As a final remark to all my friends who are still studying or preparing a final work, good luck

and someday, it is finished.



Different Grids in the Silicon Segment of a Device Structure.

Contents

1	Introduction	1
1.1	Grid Requirements for Numerical Discretization Methods	2
1.2	Influence of the Simulation Grid on the Solution	3
2	Grid Types	9
2.1	Ortho Grids	10
2.1.1	Mixed Grids	11
2.1.2	Providing Attributes	12
2.2	Tetrahedral Grids	12
2.3	Grid Requirements	13
2.4	Demands for the Box Integration Method	14
2.4.1	Voronoi Tessellation — Delaunay Mesh	15
2.4.2	Two-Dimensional Criterion	16
2.4.3	Three-Dimensional Criterion	20
2.5	Demands for Finite Elements	23
2.5.1	Two-Dimensional Criterion	25

2.6	Grid Refinement — Adaptive Meshes	26
2.6.1	Hierarchical Refinement	27
2.6.2	Moving Boundaries	33
2.6.3	Cellular Data	35
3	The Box Integration Method	38
3.1	The Poisson Equation	38
3.1.1	Approximation of E_n and ε	42
3.1.2	Continuation of the Discretization	46
3.1.3	Boundary Conditions	48
3.2	The Diffusion Equation	51
3.2.1	Initial Conditions	52
3.3	The Basic Semiconductor Equations	53
4	An Alternative Approach for Diffusion Simulation	56
4.1	Diffusion and Green's Functions	57
4.2	Prerequisites for Applying the Green's Functions Method	59
4.3	Advantages of the Green's Functions Method	60
4.4	Device Optimization by Three-Dimensional Diffusion Simulation	61
4.4.1	Simulated Structure	62
4.4.2	Comparison of Simulation Approaches	63
4.4.3	Calibration and Evaluation	63
4.4.4	Results	65

5	Grid Generation for Device Simulation	70
5.1	Grid Requirements for Device Simulation	71
5.2	Adapted Grid Generation	73
5.3	Potential-Based Grid Generation	77
5.3.1	Two-Dimensional Behavior	77
5.3.2	Extension to Three Dimensions	79
5.3.3	Calculating the Equipotential Surfaces of the Capacitor	81
5.3.4	Topological Cuboids	84
5.3.5	Proof of the Two-Dimensional Duality	85
5.3.6	The Three-Dimensional Capacitor Model	89
5.3.7	Evaluation of the Electric Potential of the Capacitor	94
5.4	Alternative Approaches for the Potential Method	98
5.5	Example for the Potential Method and the Electrode Placement	99
6	Applications of the Potential Based Method	103
6.1	Device Simulation of a FinFET	103
6.2	Development of an EEPROM Memory-Cell	109
6.2.1	Process Simulation of the Memory-Cell	109
6.2.2	Electrical Analysis of the Device	111
7	Conclusion and Outlook	115
	Bibliography	118

List of Figures	125
List of Tables	129
List of Publications	130
Curriculum Vitae	131

Chapter 1

Introduction

FOR THE DEVELOPMENT of today's highly-integrated electrical devices, the use of numerical simulation tools has become indispensable. Details in the manufacturing procedure of these devices cannot be described by simple design rules and the layout of device or interconnect structures cannot be predicted by simple considerations. Electrical characteristics, the influence of the interconnect structures on the capacitances and inductances between the metal layers, delay times of the signals, and further effects have to be accounted for during the development process. The detailed simulation of etching, deposition, implantation, and diffusion processes have to be analyzed. Finally, the electrical characteristics of the devices have to be investigated.

Coming along with the rapidly rising complexity and down-scaling of electronic devices, many effects have to be considered. Effects near corners of the devices become more dominant and physical simulation models have to be expanded. On the one hand more complex models have to be used, on the other hand the existing models, respectively their numerical solutions, have to be expanded from two spatial dimensions to their three-dimensional representations.

Usually the applied models have an analytical representation, for instance in the form of partial differential equations. Only in the most trivial cases, it is possible to find an analytical solution for those problems. Even for simple linear partial differential equations closed solutions cannot be found, because of the complex geometries and resulting boundary conditions.

1.1 Grid Requirements for Numerical Discretization Methods

Therefore, numerical methods have to be applied. Typically, a mesh is spanned over the simulation domain and a numerical solution is sought, which approximates the solution in the sampling points. The calculation is done by discretizing the differential equations by the methods of Finite Differences [66], Finite Elements [74][41] or Finite Boxes [16]. Usually this discretization results in a large equation system, which has to be solved. Each differential equation typically delivers an equation system with the rank of the system being in the same size as the number of grid points. Normally the given differential equations are not separable and in sum, an equation system with the number of unknowns being equal the number of differential equations multiplied with the number of grid points must be solved. For nonlinear systems, the final solution can only be found iteratively by applying Newton's method [59]. For time dependent problems, the solution of each time-step has to be calculated iteratively until the desired point of time is reached. For implicit methods each time-step requires the solution of the equation system. Therefore, these iterations can be very expensive in terms of calculation time, reaching the maximum in nonlinear time-variant problems.

Ideally the mesh generation process and the simulation process should be separated. This procedure eases the development of the simulation tools and guarantees the modularity and flexibility of the tools. However, these steps are not necessarily separable. For one-dimensional problems, the methods of Finite Differences, Finite Elements (assuming linear Ansatzfunctions) and Finite Boxes (also known as Box Integration) result in the same solution. Within two-dimensional simulations, Finite Difference schemes exist in practicable form only for ortho-product grids where they deliver the same equation-system as for Finite Boxes. Moreover, Finite Elements and Finite Boxes also deliver the same solution. However, in three dimensions discretization becomes more difficult and Finite Boxes and Elements differ in their solution [13][14].

Additionally the different simulation methods have different demands on the grid. In two dimensions, these criteria lead to the same geometrical grid criteria. However, for three-dimensional problems, different criteria have to be fulfilled, depending on the simulation method. On the one hand, Box Integration requires the well-known Delaunay grids, and on the other hand, Finite Element analysis should be based on a grid fulfilling a "Cotangent Criterion", where many questions for grid generation are still unresolved. Details of the discretization process and the resulting grid criterions will be shown in Section 2.2. That is the reason why good Box Integration grids will not necessarily be good Finite Element grids [19].

Finally, a specifically designed grid adaptation method can dramatically reduce the calculation effort and the memory consumption. These specially adapted grids require less grid points and produce more compact equation systems for reaching the same accuracy. Even within one equation class, the requirement on the grid can vary dynamically. A simple example, frequently used within electrical device manufacturing, is a diffusion simulation of implanted dopants. This problem requires a transient simulation where a relatively sharp implanted dopant profile diffuses to a smooth profile over time. To achieve accurate simulation results the grid density must be high in areas with high concentration curvatures [72]. In addition, locations that require high grid-resolution will move within the material regions during diffusion time. While moving, the dopant distribution is also smoothed (except for some special applications). This leads to the requirement of adaptive mesh refinement. In view of calculation times, this refinement cannot be performed by frequent regriding. Usually an approach of hierarchical refinement and coarsening will be chosen (see Section 2.15).

Basically, an overall increase of sampling points will also increase the accuracy of the simulation (until numerical instabilities limit a continuative increase of sampling points [3]). Despite the limited machine resources, namely CPU time and memory, this fact cannot be exploited indefinitely. Consider the case where a two-dimensional grid with N grid points is expanded to three dimensions. Based on the consideration that the two-dimensional grid is spanned by about $N^{1/2} \times N^{1/2}$ grid points, the estimated number of points of the three-dimensional grid will lead to an increase of computation points to $N^{3/2}$. The increase of memory consumption is of the same magnitude, not including the storage required for the third coordinate values of the grid points plus additionally used point references of the grid elements. The computation times will rise by the same amount $O(N^{3/2})$ for the best fitted solvers available like multigrid solvers, which have nearly linear complexity, $O(N^{2.3/2})$ for solvers which account for the sparsity of the equation systems, like Conjugate Gradient solvers, and up to an amount of $O(N^{3.3/2})$ for simple Gaussian solvers. The limited calculation accuracy of the floating-point units of the simulation machines also poses a limit to the equation system size.

1.2 Influence of the Simulation Grid on the Solution

In the following the influence of two different grid refining methods will be compared. The first method refines the grid by a global unstructured increase of grid points. This is done by down-scaling the maximum grid element area. This approach will

be compared to a more sophisticated point placement algorithm which, as will be shown, also achieves more accurate results.

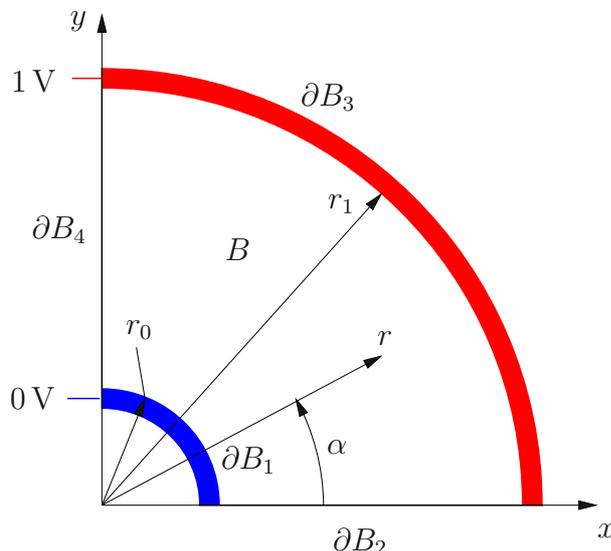


Figure 1.1: The simulation domain of the coaxial capacitor.

In Figure 1.1 a quarter of the cross-section of a coaxial capacitor is shown. The electrode areas ∂B_1 and ∂B_3 are biased at 0 V and 1 V, respectively. The whole capacitor area B is filled by an insulator with constant permittivity ε .

Because of the axial symmetry of the device, this simulation could be performed in one dimension. However, to compare the influence of the grid adaption methods, the simulation is performed in two dimensions. This artificial example is chosen, because the analytical solution of this problem can be easily found and can be used to calculate the error of the two grid adaption methods.

For the analytical and numerical evaluation of this problem, the Laplace equation [22]

$$-\operatorname{div}(\varepsilon \operatorname{grad} \varphi) = 0 \quad (1.1)$$

has to be solved within the domain B . The boundary conditions are

$$\varphi = 0 \text{ V} \quad \text{on boundary } \partial B_1, \quad (1.2)$$

$$\varphi = 1 \text{ V} \quad \text{on boundary } \partial B_3. \quad (1.3)$$

Since only a quarter of the whole coaxial capacitor is simulated, vanishing normal derivatives at the radial boundaries have to be appended

$$\partial_n \varphi = 0 \quad \text{on boundary } \partial B_2 \text{ and } \partial B_4. \quad (1.4)$$

For this simple structure, an analytical solution can be found and the electrostatic potential results to

$$\varphi(r, \alpha) = \frac{\ln \frac{r}{r_0}}{\ln \frac{r_1}{r_0}}, \quad (1.5)$$

only depending on the distance r from the center and independent of the radial angle α . The symbol r_0 denotes the radius of the inner electrode and r_1 of the outer one.

Within the following Box Integration simulations, the error of the discrete approximations is defined as:

$$f = |\varphi - \bar{\varphi}|_{\max}. \quad (1.6)$$

The symbol φ represents the exact analytic solution. In consistency with the Box Integration method, the numerical approximation $\bar{\varphi}$ is derived by linear interpolation inside the triangles of the grid.

The first meshing method places the grid points with nearly constant grid density. The only constraint for the grid density is a maximum allowable triangle area, to achieve the different grid densities. A typical grid produced by the Delaunay triangulator *triangle* [61] is shown in Figure 1.2.

The adapted grid generation method places the grid points quasi-orthogonally (Figure 1.3), which means a constant number of points m with equidistant point spacings along the expected equipotential surfaces (lines with constant radius) and a constant number of grid points n with radially increasing point distances in the perpendicular direction (along the diametric lines). This way, the number of grid points along and across the radial lines can be varied arbitrarily. The resulting grid is a triangular grid with $m \times n$ grid points. Figure 1.2 and Figure 1.3 contain approximately the same number of grid points.

In Figure 1.4 the error of the numerical solution on the adapted grid with 10, 20, 30, . . . , 60 ticks in radial direction, as a function of the total number of grid points is displayed. An interesting aspect derived by this error analysis is that the error reaches a minimal value, if the tangential and radial number of grid ticks lie within the same value. In this example for instance exactly at 30×34 grid ticks, but also verified by trials with other numbers of radial and tangential ticks. An additional increase of tangential sampling points does not improve the solution.

By varying the number of ticks radially and axially, the error depending on the total number of grid points is compared to the unstructured method. This can be seen in

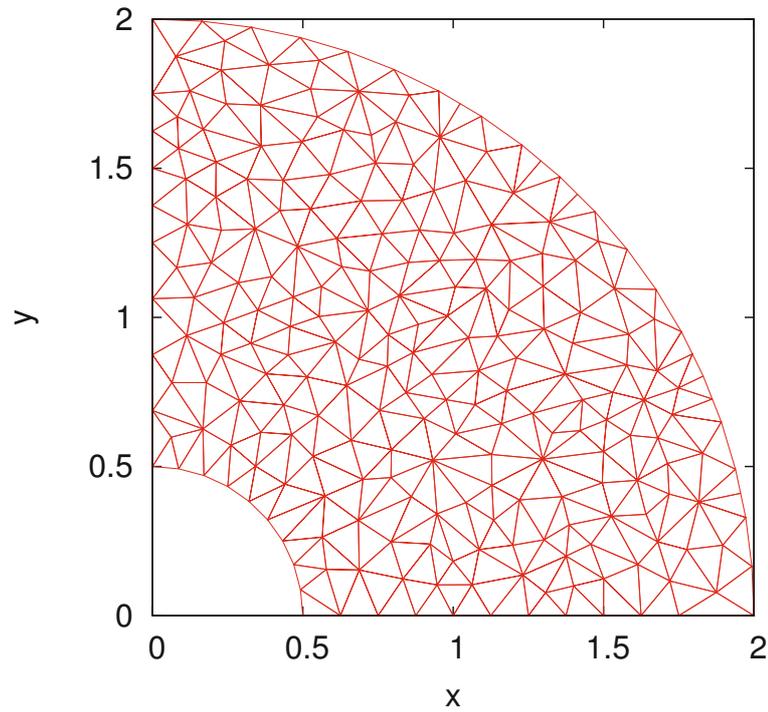


Figure 1.2: Unstructured grid refinement, 237 grid points. The resulting maximal discretization error is 134 mV.

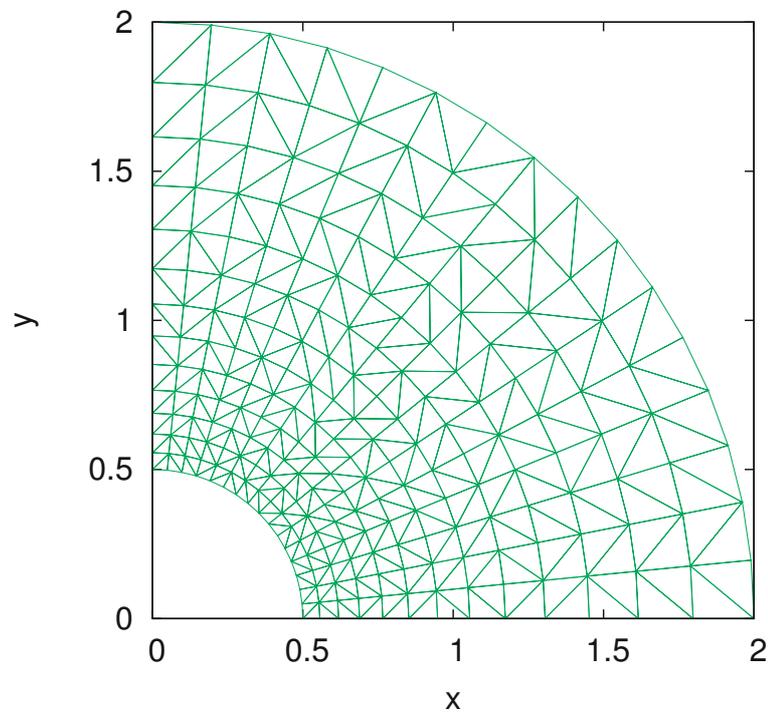


Figure 1.3: Refinement, which is adapted to the given problem, 17×14 grid points. The maximal discretization error is reduced to 8.7 mV.

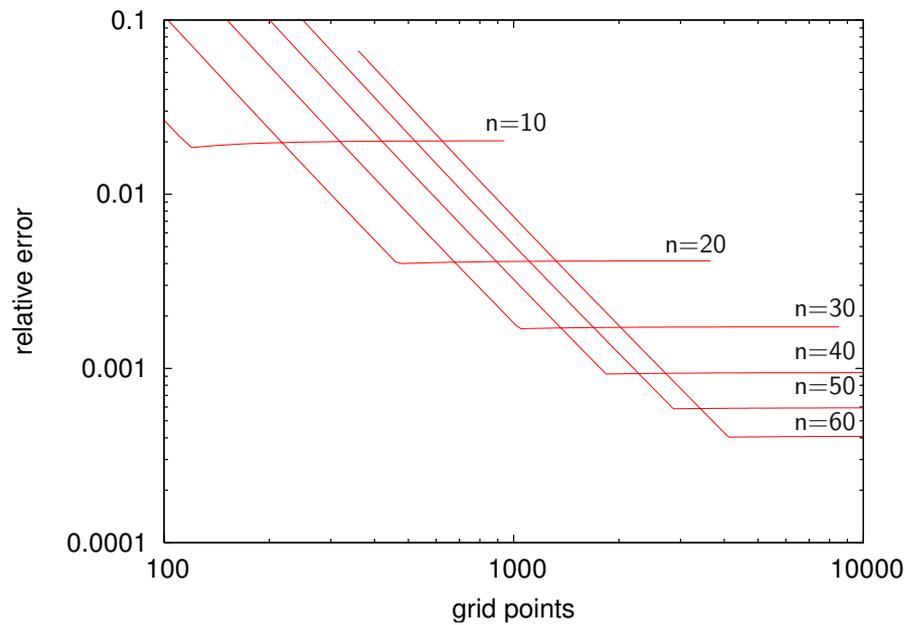


Figure 1.4: Error of the adapted grid. Each curve results from a grid with n ticks in radial direction. The total number of grid points is varied. A minimal error is reached, if the number of radial and tangential ticks are nearly of equal size.

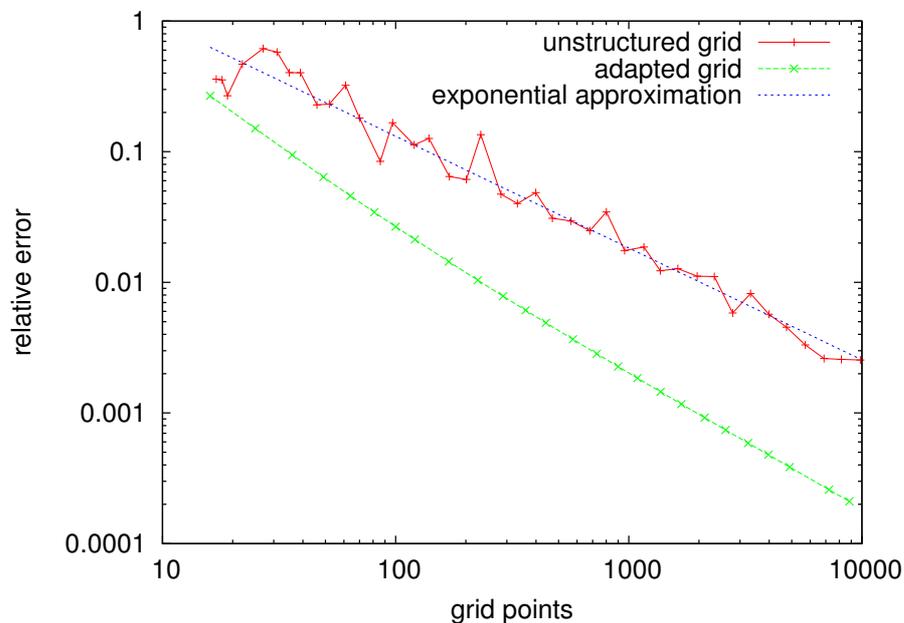


Figure 1.5: Errors of the unstructured versus the adapted grid. The unstructured grid is refined by a maximum area constraint. The adapted grid generation is built with the same number of grid ticks in radial and tangential direction.

Figure 1.5. The conclusion is, as expected, that within the adapted point placement the error differs by orders of magnitude. Within the simulation range of grid points an error dependency of $f \propto n^{-0.8}$ can be expected for the unstructured method and about $f \propto n^{-1.1}$ for the more sophisticated method.

Thus, the evaluation of the numerical solution on the unstructured grid needs about 10 times the number of grid points than the improved version to achieve equal accuracy — also implicating the same factor for memory and time consumption. Within three-dimensional simulations this aspect becomes even more severe.

Chapter 2

Grid Types

WHEN DISCRETIZING the analytical problem the chosen grid type must be accounted for. Different numerical methods require special kinds of grids and the discretization process depends on the grid type, too.

One way to classify grids is by their basic grid elements. Grids with only one element class have to be distinguished from grids with multiple element classes. It can often be an advantage to allow differently shaped elements, since the use of different element types may result in grids that are more flexible. The boundary approximation of these elements can sometimes be better and the total number of grid elements is usually reduced. However, those mixed element grids are often not easy to handle with regard to both grid generation as well as numerical discretization. The numerical discretization method must be applicable for all types of elements that are used in the grid, which limits the set of usable element types and reduces the flexibility of this method. Therefore, in most simulators only mixed element grids with some basic element types, such as rectangles and triangles or tetrahedrons and cuboids, are implemented.

Within this work, we will only consider simple element grids, and discriminate between triangular shaped elements, which are triangles in two dimensions and tetrahedrons for three dimensions, and rectangle shaped elements, rectangles for two dimensions and cuboids for three dimensions. The second kind of grids are usually known as ortho-product or ortho grids.

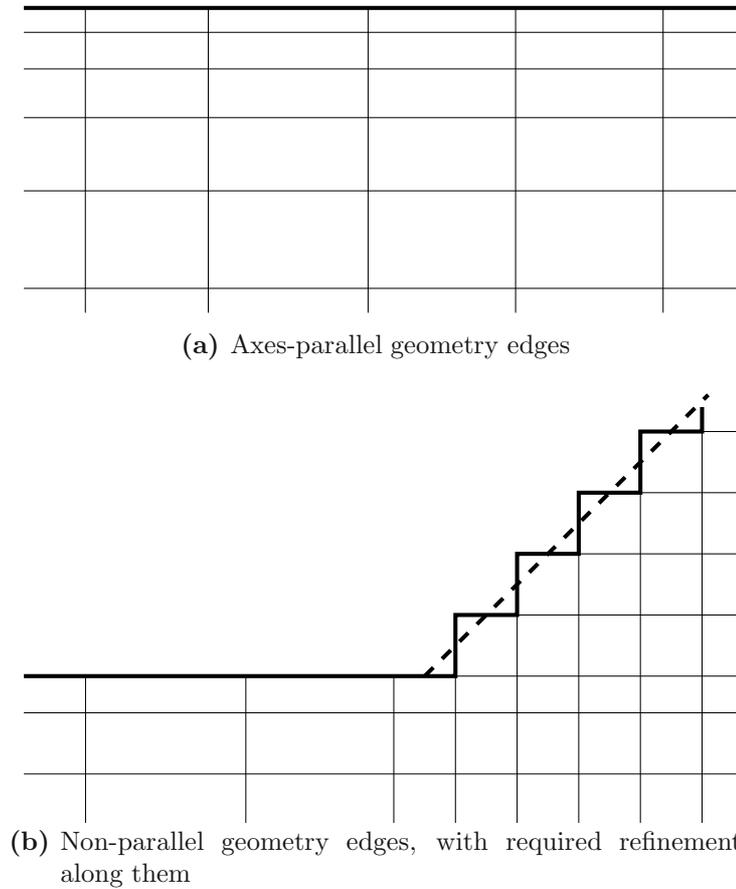


Figure 2.1: Two-dimensional ortho grids along axes-parallel and non-parallel geometry edges.

2.1 Ortho Grids

The simplest way for meshing a region is to cut the simulation area into layers of different thickness. After cutting in all three directions of space, we get an ortho grid. For the grid representation, only the elements that lie within the simulation domain are relevant (Figure 2.1).

If all edges of the geometry of the structure are parallel to the coordinate axes, we will find an acceptable mesh for this domain. However, with complex geometrical structures the limits of this method are quickly reached. Basically, inserting a new point in an $[m \times n \times o]$ points wide grid, where m , n , and o denote the number of ticks in the three coordinate directions, will produce a grid with $[(m+1) \times (n+1) \times (o+1)]$ points. This may pose an unacceptable computational burden in terms of CPU-time and memory consumption.

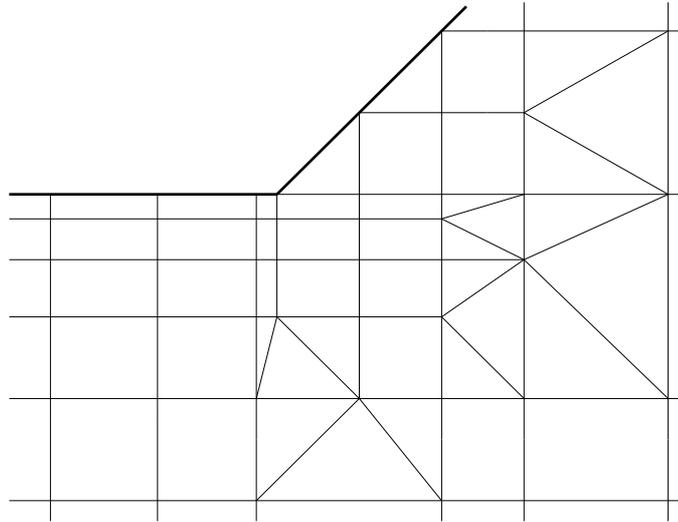


Figure 2.2: A two-dimensional mixed grid with rectangular and triangular elements to approximate the boundary. Terminating lines reduce the number of grid points.

Especially with non-planar geometries, there is no way to reproduce such non-planar surfaces. They must be approximated by steps in the geometry, as sketched in Figure 2.1 for a two dimensional example of an ortho grid. To achieve the desired resolution of those edges which are non-parallel to the coordinate axes these lines must be approximated by small steps. A negative side effect of this method is that the amount of grid points will rise dramatically. Each additional coordinate tick will result in an unacceptable number of unwanted grid points.

2.1.1 Mixed Grids

As a solution to this dilemma, it can be useful to allow triangular or tetrahedral elements, too. This is shown in Figure 2.2. Grid lines, which are needed in one region only, can terminate inside the simulation domain. At the termination points, rectangles (cuboids) with additional points on the sides arise. As such rectangles (cuboids) usually are not supported, they must be split into triangles (tetrahedrons) to satisfy the connectivity. However, if triangles or tetrahedrons are provided, they can also be provided at the geometry edges, which is shown in the region near the fat boundary edges of Figure 2.2. This generally will reduce the amount of grid points while better approximating the geometry.

Furthermore, since ortho grids can be split into triangles (or tetrahedra for three dimensions), there is no need to store rectangles any longer. Each rectangle can be

split in two triangles, each cuboid can be split into five or six tetrahedrons. As a side effect, this decomposition fulfills the Delaunay criterion, which is relevant for certain numerical schemes as mentioned previously. Adding special terminating lines criteria within these generated triangles, a valid triangular (tetrahedral) Delaunay mesh can be built.

2.1.2 Providing Attributes

Ortho grids are a common way to provide attributes, such as dopant concentrations. On each point of the $[m \times n \times o]$ ortho grid several attribute values can be stored. The actual simulation is performed on a separate simulation grid, which can be handled independently from these attribute grids. This has the big advantage that the layout of the attribute and simulation grid can be customized to their individual requirements. Regridding, coarsening, and refining the simulation grid will not alter the attributes and no information loss of the primary attribute data is caused by a change of the simulation grid. Otherwise, the resolution of the attributes can be tuned arbitrarily, without an increase of the simulation times. Before starting the simulation, the attribute values of the ortho grid are interpolated to the simulation grid. In general, attributes are often provided on different grids than the simulation grid.

2.2 Tetrahedral Grids

A flexible way for meshing geometries is possible by tetrahedral meshing. There are almost no limitation on the geometries and the direction of the geometry edges is not subject to any restrictions.

The most important discretization methods in semiconductor process and device simulation are Box Integration and the Finite Elements method. The problems under consideration, such as electrical device simulations of semiconductor devices and diffusion simulations of implanted dopants, impose certain requirements on the grids. Basically it seems possible that both simulation problems can be solved by both discretization methods. However, most device simulators (MINIMOS [6][11], DESSIS [33], MEDICI [68]) use the Box Integration method, which will be examined in Chapter 3, whereas diffusion simulation is usually done by Finite Elements.

2.3 Grid Requirements

To obtain quality criteria for the grids, the differential equations of the models must be taken into account. For the solution of the frequently required Laplace equation and the diffusion equation, an additionally required grid-condition must be formulated. It will be helpful to illustrate it by an example. When solving the electrostatic potential distribution inside a capacitor it is physically plausible that, if the voltages on the contacts lie in the range of φ_0 to φ_1 , all potential values inside the capacitor also must lie within the same range.

Mathematically, this is known as the maximum principle [28][45]. In its simplest form it states that both the maximum and the minimum values occur on the boundary or at initial time. Since the numerical solution must be a good approximation of the exact solution and unphysical solutions must be prevented, this maximum principle has to be satisfied also by the numerical method. For the diffusion equation, this principle implies that unphysical negative concentrations may not appear in the solution (as negative concentrations do not appear in the initial condition). This principle implies also another physical interpretation, which is known as the Positive Transmissibility Condition. It states, that the physical flow has to be directed from higher to lower concentrations.

Within the discretization methods, which are in particular the Box Integration, as well as the Finite Element discretization with a standard Galerkin weighted residual approach, the Laplace equation

$$\operatorname{div}(\varepsilon \operatorname{grad} x) = 0 \quad (2.1)$$

and the diffusion equation

$$\operatorname{div} \operatorname{grad} x = \frac{1}{D} \frac{\partial x}{\partial t} \quad (2.2)$$

result in the same type of equation system

$$\begin{pmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & & \ddots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (2.3)$$

$$\mathbf{K} \cdot \mathbf{x} = \mathbf{b}.$$

The maximum principle is satisfied if \mathbf{K} becomes an M-matrix [43][45].

$\mathbf{A} = (a_{ij})$ is an M-matrix [75], if \mathbf{A} is a real and nonsingular $n \times n$ matrix, where

$$a_{ij} \leq 0 \quad \forall i, j : i \neq j, \quad (2.4)$$

$$a_{ii} > 0 \quad \forall i, \quad (2.5)$$

$$a_{ii} \geq \sum_{\forall j : i \neq j} |a_{ij}| \quad \forall i, \quad (2.6)$$

$$a_{ii} > \sum_{\forall j : i \neq j} |a_{ij}| \quad \text{for at least one } i. \quad (2.7)$$

The entries of $\mathbf{K} = (k_{ij})$ differ in the composition of left-hand and right-hand terms of equation (2.1), or (2.2) with backward Euler time discretization [66].

$$\mathbf{K} = \mathbf{S} \quad \text{for (2.1) and} \quad (2.8)$$

$$\mathbf{K} = D \mathbf{S} + \frac{1}{\Delta t} \mathbf{M} \quad \text{for (2.2).} \quad (2.9)$$

$\mathbf{M} = (m_{ij})$ denotes the mass matrix, which arises from the time variant right-hand side, $\mathbf{S} = (s_{ij})$ is the stiffness matrix, which depends only on the left-hand side of equation (2.1) or (2.2). For this composition we know that \mathbf{K} is an M-matrix if \mathbf{M} is a positive definite diagonal matrix and \mathbf{S} also is an M-matrix. For Box Integration (refer Section 3.2), the positive definite diagonal characteristic of \mathbf{M} is satisfied. For Finite Elements, mass matrix lumping has to be performed to achieve a positive definite diagonal matrix \mathbf{M} [25][46]. Since \mathbf{S} only depends on the mesh, this condition translates to a grid constraint. Off-diagonal elements s_{ij} of \mathbf{S} must not be positive

$$s_{ij} \leq 0. \quad (2.10)$$

This constraint also fulfills the Positive Transmissibility Condition [28].

Consequently the requirements for Box Integration and Finite Elements follow.

2.4 Demands for the Box Integration Method

In association with the Box Integration discretization (refer Chapter 3, equations (3.32) and (3.57)), equation (2.10) can be rewritten as

$$s_{ij} = -f_{ij} \frac{A_{ij}}{d_{ij}} \leq 0, \quad (2.11)$$

where f_{ij} denotes a positive material parameter (the permittivity for the Laplace equation, 1 for the diffusion equation). The parameters d_{ij} and A_{ij} are geometrical values which are determined by the geometry of the grid elements only. d_{ij} is the length of the edge $\langle p_i p_j \rangle$ between the points p_i and p_j (positive), and A_{ij} the coupling area between the two points. As a geometrical consequence the coupling areas and point distances are symmetrical

$$A_{ij} = A_{ji} \quad \text{and} \quad d_{ij} = d_{ji}, \quad (2.12)$$

and it is also of advantage and plausible to use a symmetrical material parameter

$$f_{ij} = f_{ji}, \quad (2.13)$$

which finally delivers a symmetrical system matrix \mathbf{S} . Consequentially, relation (2.11) is only satisfied if the coupling area is positive

$$A_{ij} \geq 0. \quad (2.14)$$

However, such positive coupling areas are guaranteed by a Delaunay tessellation of the point set. In general, the definition of a Delaunay triangulation is based on the Voronoi diagram by the principle of duality [42] and will be introduced in the following section.

2.4.1 Voronoi Tessellation — Delaunay Mesh

Let $\mathcal{P} = \{p_1, p_2, \dots, p_r\}$ be a finite set of points in a sub-domain Ω^n of the n -dimensional space \mathcal{R}^n .

A Voronoi region $V(p_i)$ is the set of all points of Ω^n that are closer to p_i than to any other point of \mathcal{P} .

$$V(p_i) = \{p \mid p \in \Omega^n \wedge \|p - p_i\| < \|p - p_j\|, \forall j \neq i\} \quad (2.15)$$

The resulting Voronoi regions $V(p_i)$ form a Voronoi tessellation of Ω^n (without overlap or exclusion)

$$\Omega^n = \bigcup_{i=1}^r V(p_i). \quad (2.16)$$

By connecting the vertices p_i, p_j of two touching Voronoi regions $V(p_i)$ and $V(p_j)$, a Delaunay edge $\langle p_i p_j \rangle$ is constructed. The pool of all Delaunay edges builds the

Delaunay mesh of the point set \mathcal{P} . This Delaunay graph shows the following properties:

Two points p_i and p_j form a Delaunay edge $\langle p_i p_j \rangle$ if and only if there exists an n -dimensional sphere which passes p_i and p_j and contains no other points of \mathcal{P} .

($n \geq 2$) Three non-collinear points p_i, p_j , and p_k form a Delaunay triangle $\langle p_i p_j p_k \rangle$ if and only if there exists an n -dimensional sphere which passes p_i, p_j , and p_k and contains no other points of \mathcal{P} .

($n \geq 3$) Four non-coplanar points p_i, p_j, p_k , and p_l form a Delaunay tetrahedron $\langle p_i p_j p_k p_l \rangle$ if and only if there exists an n -dimensional sphere which passes p_i, p_j, p_k , and p_l and contains no other points of \mathcal{P} .

These formulations can be expressed in a mathematical way:

$$\begin{aligned} \langle p_i p_j p_k p_l \rangle_{\text{Delaunay}} : \quad & \forall m \neq i, j, k, l \\ & \exists p_m \in \Omega^n \quad \wedge \\ & \|s - p_i\| = \|s - p_j\| = \|s - p_k\| = \|s - p_l\| \quad \wedge \\ & \|s - p_i\| < \|s - p_m\| \end{aligned}$$

Here, s is the center of the n -dimensional sphere. Analogous formulations can be found for Delaunay triangles and lines. A Delaunay tetrahedron implies that it must consist of Delaunay triangles and Delaunay edges.

2.4.2 Two-Dimensional Criterion

A two-dimensional Voronoi box is shown in Figure 2.3. In two dimensions the coupling area A_{ij} degenerates to the length of the Voronoi edge, which bisects the edge $\langle p_i p_j \rangle$ and connects the two center-points of the outer-circles of the triangles $\langle p_i p_j p_k \rangle$ and $\langle p_i p_j p_l \rangle$.

By splitting this area A_{ij} into the parts $A_{ij,k}$ and $A_{ij,l}$ arising from the involved triangles $\langle p_i p_j p_k \rangle$ and $\langle p_i p_j p_l \rangle$, respectively, the situation depicted in Figure 2.4(a) is obtained. In this context, for each triangle $\langle p_i p_j p_k \rangle$ the possibilities shown in Figure 2.4(c) and Figure 2.4(d) exist. The portion $A_{ij,k}$ can lie inside or outside the half plane which is spanned by the straight line, built by p_i and p_j , and the

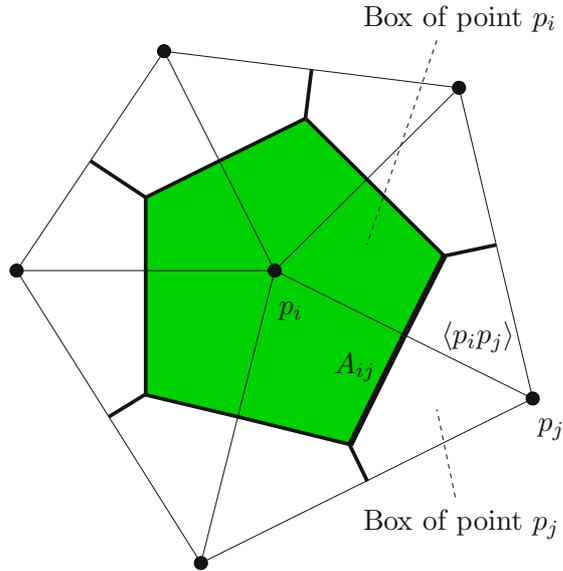


Figure 2.3: A two-dimensional Voronoi box of the point p_i

point p_k . By definition an inside-portion $A_{ij,k}$ is signed positive (Figure 2.4(c)) and signed negative if it lies outside (Figure 2.4(d)). In consideration of non-overlapping Voronoi boxes, the sum of both portions has to be positive. And with geometrical considerations (refer Figure 2.4(b))

$$\Theta_k + \Theta_l \leq 180^\circ, \quad (2.17)$$

or equivalently,

$$\Theta_k \leq 180^\circ - \Theta_l \quad (2.18)$$

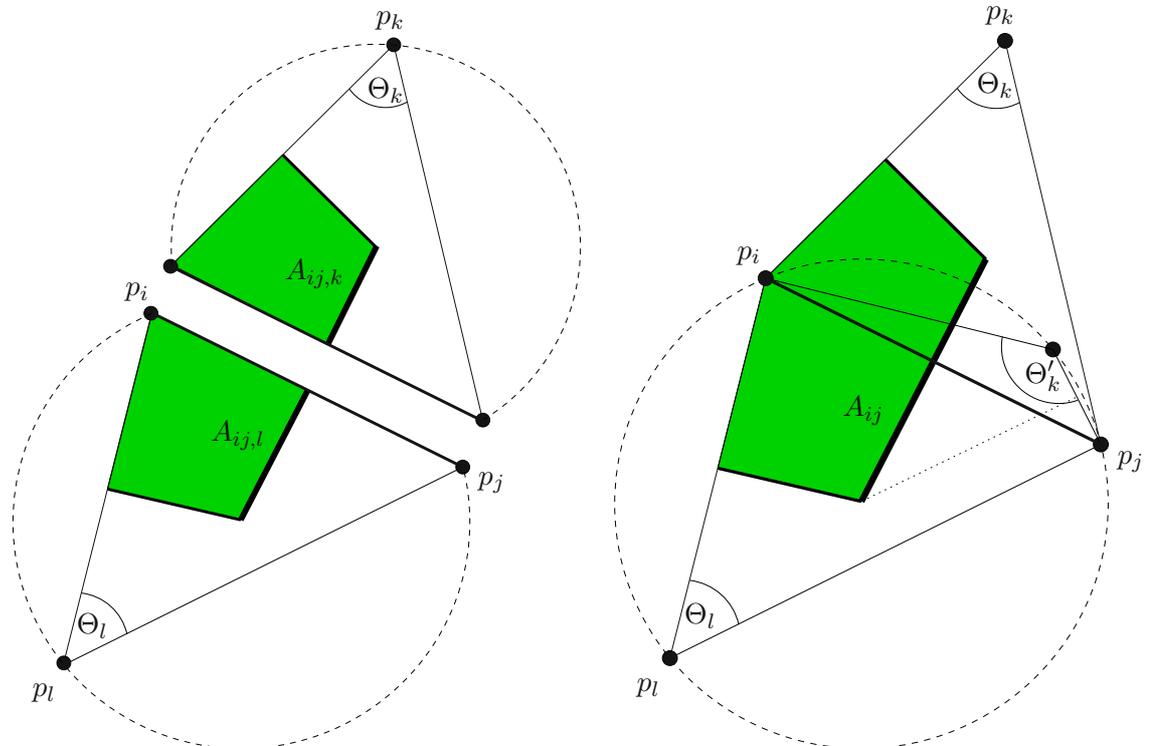
has to be satisfied.

Consequently it follows that if the point p_k lies inside the outer-circle of the triangle $\langle p_i p_j p_l \rangle$, relation (2.18) is violated. If p_k lies outside or directly on the circle (marginal case), the relation is satisfied. This criterion must be fulfilled for each neighboring triangle-pair of the mesh. Examples for valid and invalid (overlapping) two-dimensional Voronoi edges are shown in Figure 2.5.

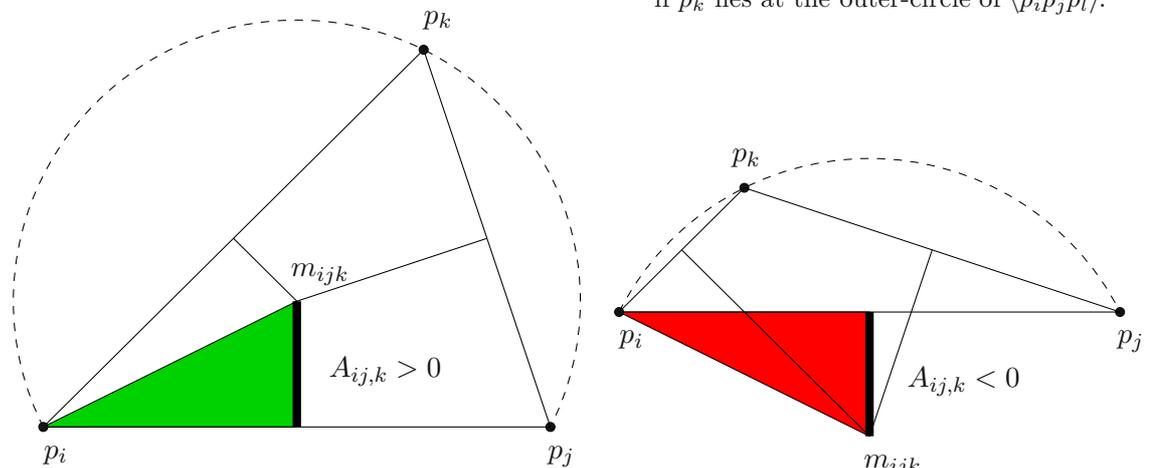
With other words:

There must not be any grid point, which lies inside the outer-circle of each triangle.

This is known as the Delaunay criterion for two dimensions.

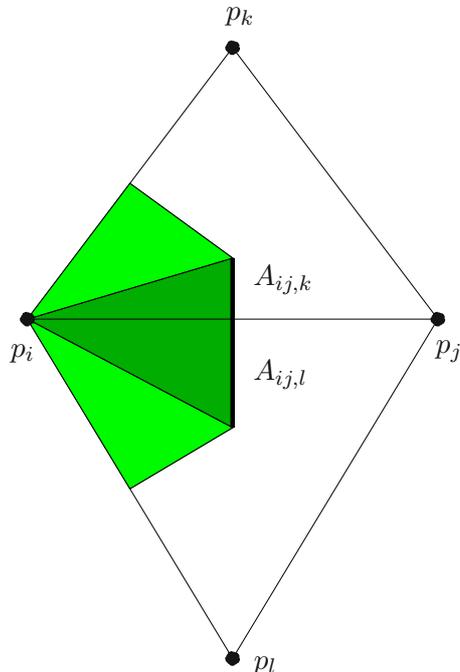


(a) The components of the Voronoi regions of the triangles $\langle p_i p_j p_k \rangle$ and $\langle p_i p_j p_l \rangle$. (b) The angle criterion for two dimensions. The marginal case with $A_{ij} = 0$ is reached if p_k lies at the outer-circle of $\langle p_i p_j p_l \rangle$.

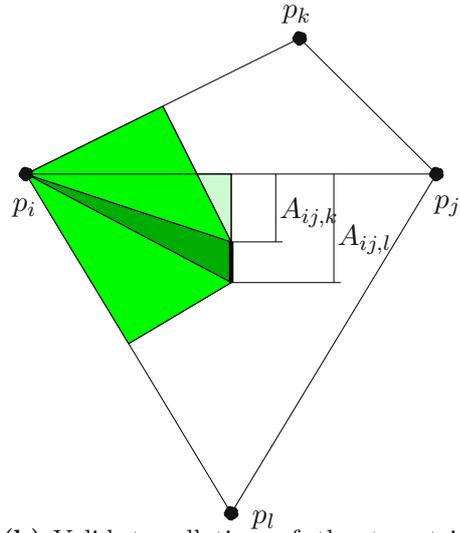


(c) The center m_{ijk} of the outer-circle of the triangle $\langle p_i p_j p_k \rangle$ lies in the half plane $\langle p_i p_j \rangle, p_k$. The portion of the coupling area is positive $A_{ij,k} > 0$. (d) The center m_{ijk} of the outer-circle of the triangle $\langle p_i p_j p_k \rangle$ lies outside of the half plane $\langle p_i p_j \rangle, p_k$. The portion of the coupling area is negative $A_{ij,k} < 0$.

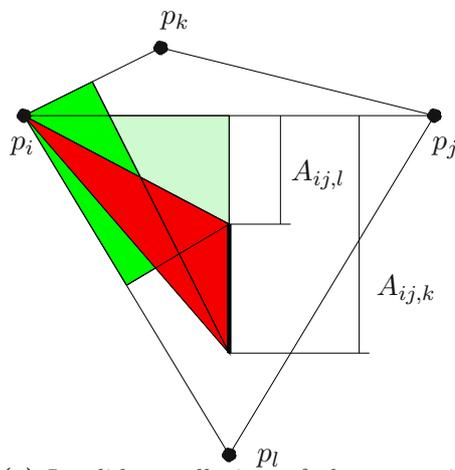
Figure 2.4: A detail of the Voronoi Region shown before.



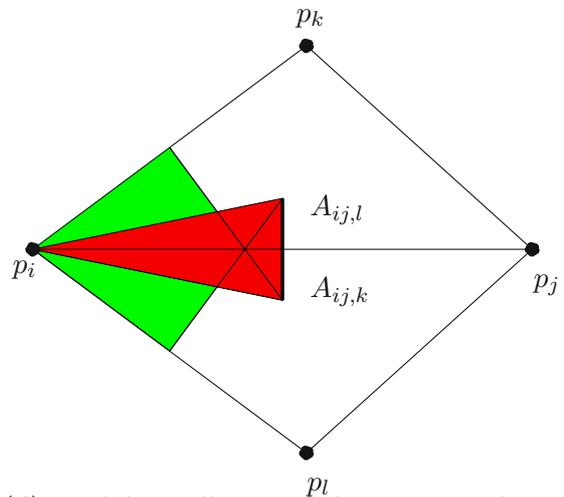
(a) Valid tessellation of two triangles, $A_{ij,k} > 0$ and $A_{ij,l} > 0$



(b) Valid tessellation of the two triangles, $A_{ij,k} < 0$, $A_{ij,l} > 0$ and in sum $A_{ij} > 0$, a non-overlapping Voronoi box remains



(c) Invalid tessellation of the two triangles, $A_{ij,k} < 0$, $A_{ij,l} > 0$, but $A_{ij} = A_{ij,k} + A_{ij,l} < 0$, an overlapping Voronoi box remains



(d) Invalid tessellation of the two triangles, $A_{ij,k} < 0$ and $A_{ij,l} < 0$, also an overlapping Voronoi box is built

Figure 2.5: Valid and invalid tessellations of two triangles.

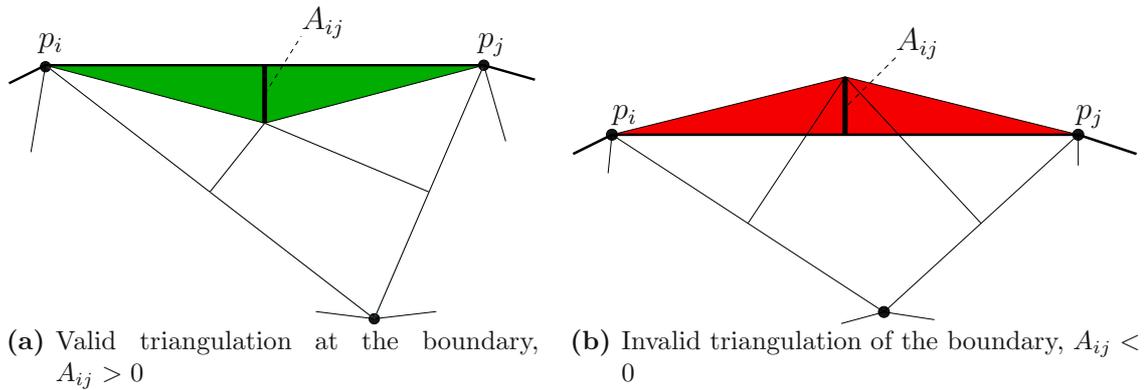


Figure 2.6: Valid and invalid Voronoi edges at the surface. At the marginal case, the center of the outer-circle lies at the surface.

Addendum for Boundary Points

While the above criterion can be satisfied for each point set, for given geometry and boundary constrains the criterion must be formulated more strictly.

In that case, (2.14) must also be satisfied on the boundaries. The coupling area A_{ij} between two boundary points p_i and p_j must not be negative (see Figure 2.6). This means, all centers of the outer-circles of the boundary triangles must lie within the boundaries. This type of triangulation is called a *Constrained Delaunay Triangulation* and the criterion can be formulated as:

There must not be any grid point, which lies inside the half circles constructed by the boundary lines.

2.4.3 Three-Dimensional Criterion

For the three-dimensional case such an area A_{ij} is shown in Figure 2.7(b), which is based on a typical Voronoi box as shown in Figure 2.7(a). This area can be split into the different components due to each tetrahedron (shown in Figure 2.7(c) and the separated situation in Figure 2.7(d)). The different parts are spanning triangles. Each triangle is defined by the center of the examined edge, the center of the outer-circle of the appropriate triangle and the center of the outer-sphere of the tetrahedron.

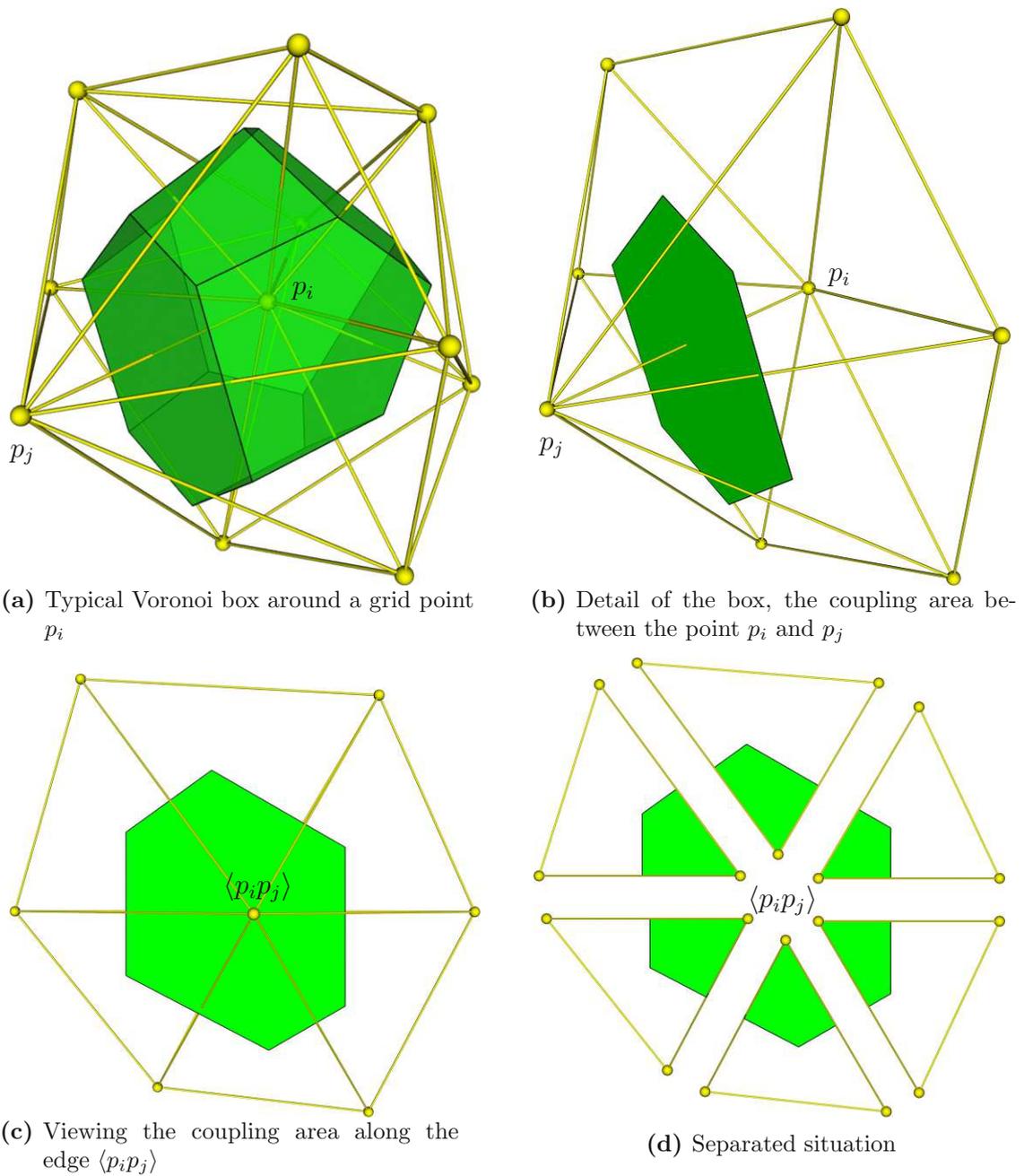
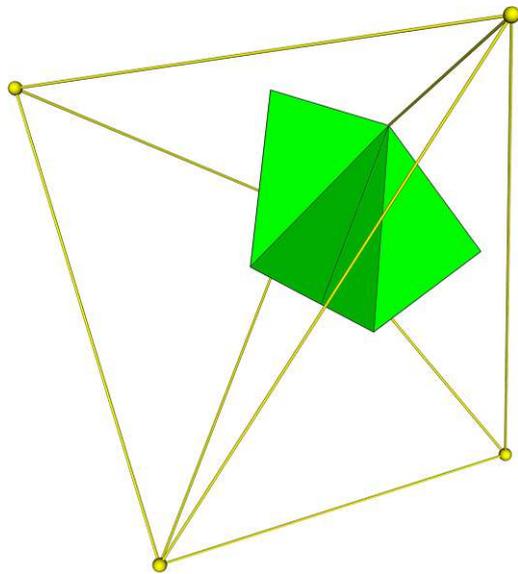
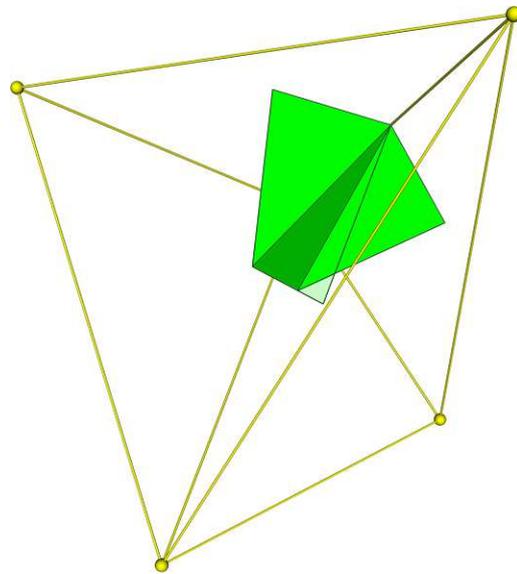


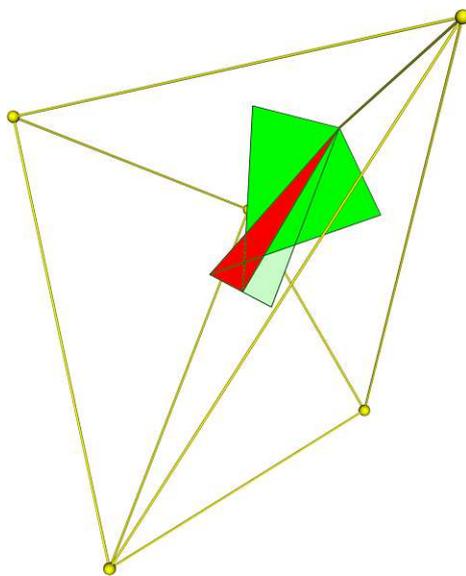
Figure 2.7: The coupling areas of a three-dimensional tetrahedral Delaunay grid around the point p_i .



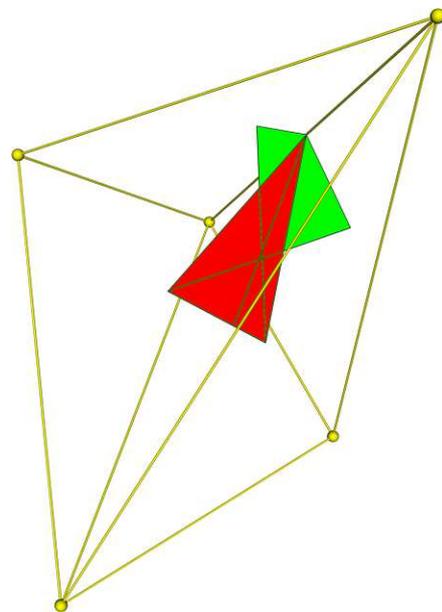
(a) Valid coupling area, the portions of both tetrahedrons are positive



(b) Valid coupling area, one positive and one negative portion, in sum a positive, non-overlapping area remains



(c) Invalid coupling, one positive and one negative coupling portion, an overlapping area is constructed



(d) Invalid coupling, both portions are negative, a negative and overlapping area remains

Figure 2.8: The coupling areas between two points of different tetrahedrons.

Analogous to two dimensions, the sign of such an area of a box part that lies inside the tetrahedrons must be positive, outside areas have a negative sign, and generally, the areas are not allowed to overlap.

Such details of valid boxes are shown in Figure 2.8(a) and 2.8(b). In the first example, both area parts are positive. In the second example, one area is negative, but smaller than the other one, which also leads to a valid grid. Invalid details with a resulting negative sum of the areas are shown in Figure 2.8(c) and 2.8(d).

Such a tessellation of a point set is also a Delaunay tessellation and the criterion reads as follows:

There must not be any grid point, which lies inside the outer-sphere of each tetrahedron.

But different to the two-dimensional case, no simplified angle criterion can be declared for three dimensions.

Addendum for Boundary Points

Also for three-dimensional Delaunay grids, the coupling area A_{ij} between two boundary points p_i and p_j must not be negative. The three-dimensional behavior is shown in Figure 2.9 and the criterion can be formulated:

There must not be any grid point, which lies inside the half spheres constructed by the boundary triangles.

2.5 Demands for Finite Elements

Within Finite Elements the coefficients of the matrix \mathbf{S} shown in (2.8) and (2.9) can be generally expressed as [46][54][75]

$$s_{ij} = \sum_{\text{elements}} \int_{V_{\text{element}}} (\nabla N_i) \underline{\underline{\epsilon}} (\nabla N_j) dv, \quad (2.19)$$

respectively

$$s_{ij} = \sum_{\text{elements}} \int_{V_{\text{element}}} (\nabla N_i) \cdot (\nabla N_j) dv \quad (2.20)$$

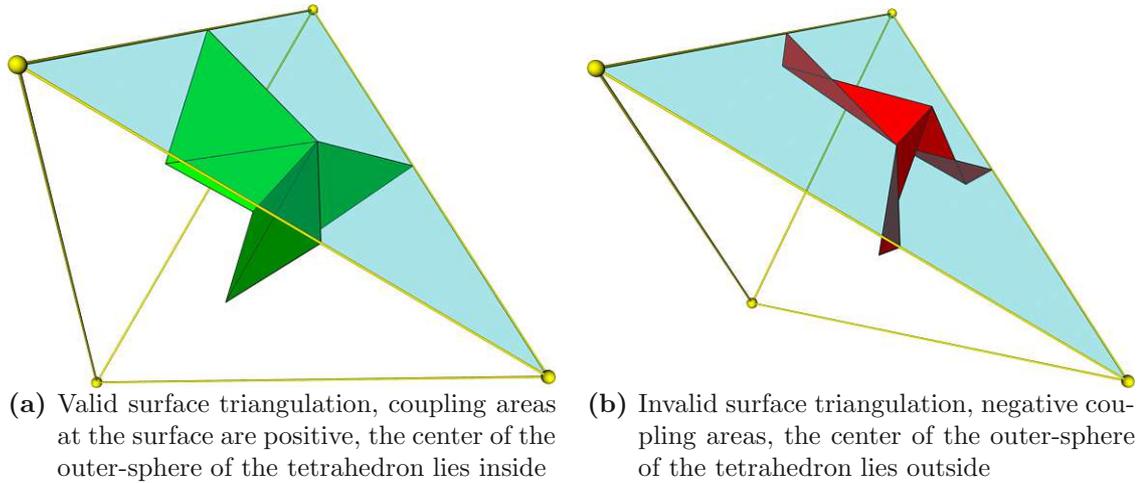


Figure 2.9: Valid and invalid Voronoi faces at the surface. At the marginal case, the center of the outer-sphere lies at the surface.

for the element volumes V_{element} . The symbols N_i and N_j denote the Ansatzfunktionen of the points p_i and p_j . If $\underline{\epsilon}$ is a constant scalar within the tetrahedrons, this in-product has a simple geometrical meaning and the compliance (2.10)

$$s_{ij} \leq 0 \quad (2.21)$$

leads to an angle criterion for each edge of the mesh [75]

$$\sum_{k=1}^n l_k \cot \Theta_k \geq 0. \quad (2.22)$$

Sum of dihedral angles:

Let $\langle p_i p_j \rangle$ be an edge with n adjacent tetrahedra t_k . For each tetrahedron t_k two planes exist which do not contain $\langle p_i p_j \rangle$ and which span a dihedral angle Θ_k . The two planes share an edge with length l_k . The sum over $k = 1 \dots n$ of the cotangent of Θ_k weighted by l_k must be greater or equal than zero.

Within Figure 2.10 the criterion is clarified. In Figure 2.10(a) the dihedral angle of the edge $\langle p_i p_j \rangle$ is shown. It is the angle between the faces which do not contain the edge $\langle p_i p_j \rangle$. Here, l is the length of the edge opposing to $\langle p_i p_j \rangle$. In Figure 2.10(b) all the tetrahedrons connected to $\langle p_i p_j \rangle$ are shown. Each of the participating tetrahedrons t_k spans its own dihedral angle Θ_k to $\langle p_i p_j \rangle$.

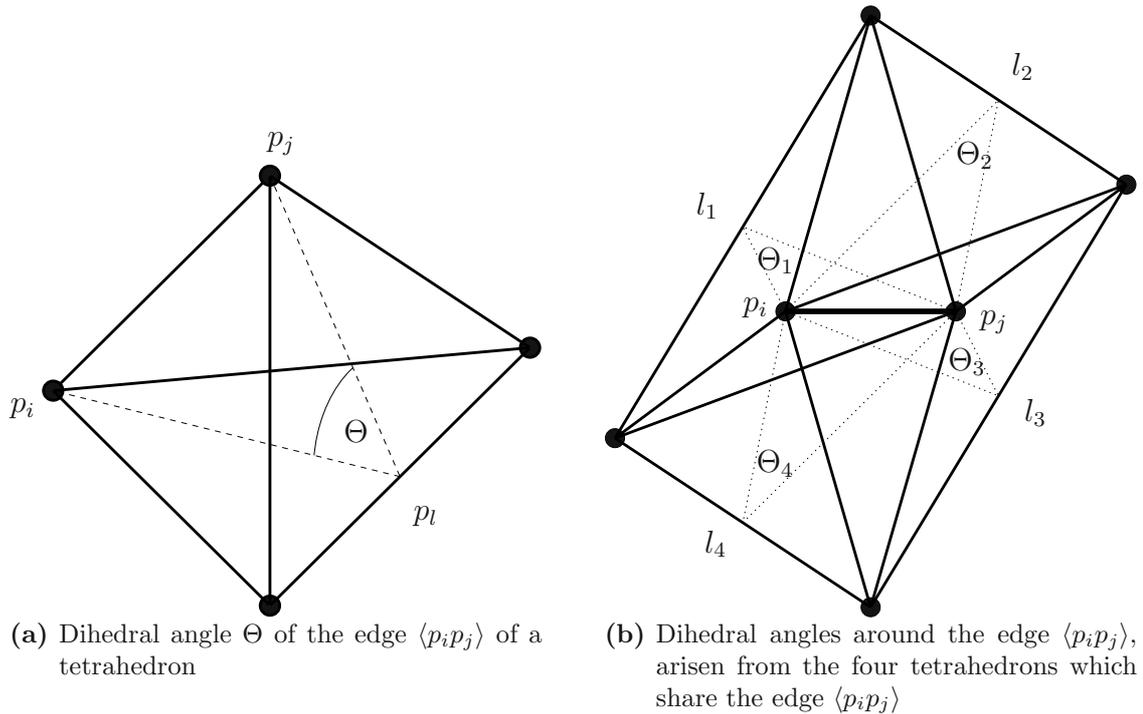


Figure 2.10: Three-dimensional grid criterion for Finite Elements.

2.5.1 Two-Dimensional Criterion

In two dimensions the criterion (2.22) is also valid. The two triangles connected to $\langle p_i p_j \rangle$ can be considered as tetrahedrons with the same (small) edge l_k , which can be canceled out of the summation, and the dihedral angle of the edge $\langle p_i p_j \rangle$ simplifies to the angle between the two edges of the triangle which do not contain this edge.

The dihedral angle reduces to

$$\Theta_k = \angle(\langle p_i p_k \rangle, \langle p_k p_j \rangle). \quad (2.23)$$

As shown in Figure 2.11 in two dimensions exactly two triangles are connected by each edge and (2.22) simplifies to

$$\cot \Theta_k + \cot \Theta_l \geq 0. \quad (2.24)$$

With the following transformation and knowing that

$$0 < \Theta \leq 180^\circ \quad \text{or} \quad \sin \Theta > 0 \quad (2.25)$$

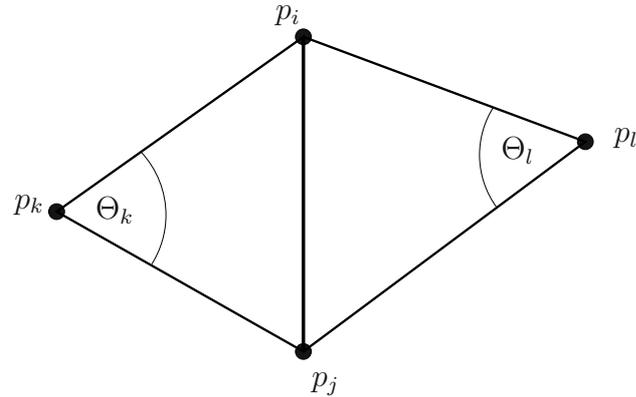


Figure 2.11: Two-dimensional mesh criterion for Finite Elements

we get

$$\sin \Theta_l \cos \Theta_k + \sin \Theta_k \cos \Theta_l \geq 0, \quad (2.26)$$

which is equivalent to

$$\sin(\Theta_k + \Theta_l) \geq 0. \quad (2.27)$$

Therefore,

$$\Theta_k + \Theta_l \leq 180^\circ \quad (2.28)$$

is the well-known formulation for two dimensions. This is the same criterion as for Box Integration, given in (2.17). Therefore, the general two-dimensional grid criterion for Finite Boxes and Finite Elements results in the same formulation which is satisfied if the tessellation of the grid points is a Delaunay tessellation. In three dimensions, the criterion for Finite Boxes and Finite Elements differ. Detailed information of the three-dimensional differences can be found in [13][14][19][28].

2.6 Grid Refinement — Adaptive Meshes

During the simulation process, often locally adapted grids are required. We can distinguish two types of adaptation:

- Due to calculation errors a grid refinement at local regions becomes necessary. This can be within time invariant simulations where a criterion indicates a

large discretization error in some regions.

Alternatively, within transient simulations and the areas of interest move inside the device, which is caused by the gradients of the distributions. During a diffusion simulation, for instance, initially narrow and steep dopant profiles become wider. In addition, the profile edge becomes weaker and propagates across the device. Such an one-dimensional diffusion profile is shown in Figure 2.12. The grid resolution in areas of high concentration curvatures and gradients must be high to achieve a good accuracy. Initial areas of high grid density become useless during simulation and the grid density can be reduced in these areas, while in regions where the dopant edge sweeps through the grid density must be increased. This enforces transient grid refinement and coarsening during the simulation. It has to be noted that full regridding is not practicable with regards to calculation time.

- By special kinds of simulations even boundaries may be deformed or new material segments can be created — for example within oxidation — or a material segment can be etched away. Concerning grid generation, these applications require a broad pool of grid adaptation techniques. Some of them are not trivially solvable. Local deformation can be solved relatively simply by local regridding or movement and stretching of grid elements, whereas the creation of new segments is not trivial. As several segments can merge during the simulation, one segment can also be split into several parts.

Usually additional knowledge of the simulation problem is required. A general refinement approach has not been found yet and grid refinement must be adapted to the given problem. The methods described in this section have not been realized in the scope of this work, but show an overview of investigations and developments which are still under construction.

2.6.1 Hierarchical Refinement

For frequently performed local grid refinement and coarsening, the complete regeneration of the grid of the simulation domain is not acceptable due to the enormous time consumption of each new regridding step. Local regridding will be a better solution, but most of the grid generators are not designed for local regridding. The grid generator must support functionality for the removal of grid elements and points, the insertion of new points, and regridding. Local regridding of an area often requires the removal of neighboring elements and an inclusion of these regions in the regridding process. Most grid generators do not supply these functionalities.

$$\begin{aligned} \text{Differential Equation: } & \frac{\partial^2 c}{\partial z^2} = \frac{1}{D} \frac{\partial c}{\partial t} \\ \text{Boundary Condition: } & \frac{\partial c}{\partial z}(0, t) = 0 \\ \text{Initial Distribution: } & c(z, 0) = c_0 \left(e^{-\frac{(z-d)^2}{k^2}} + e^{-\frac{(z+d)^2}{k^2}} \right) \\ \text{Solution: } & c(z, t) = c_0 \sqrt{\frac{k^2}{k^2 + 4 D t}} \left(e^{-\frac{(z-d)^2}{k^2 + 4 D t}} + e^{-\frac{(z+d)^2}{k^2 + 4 D t}} \right) \end{aligned}$$

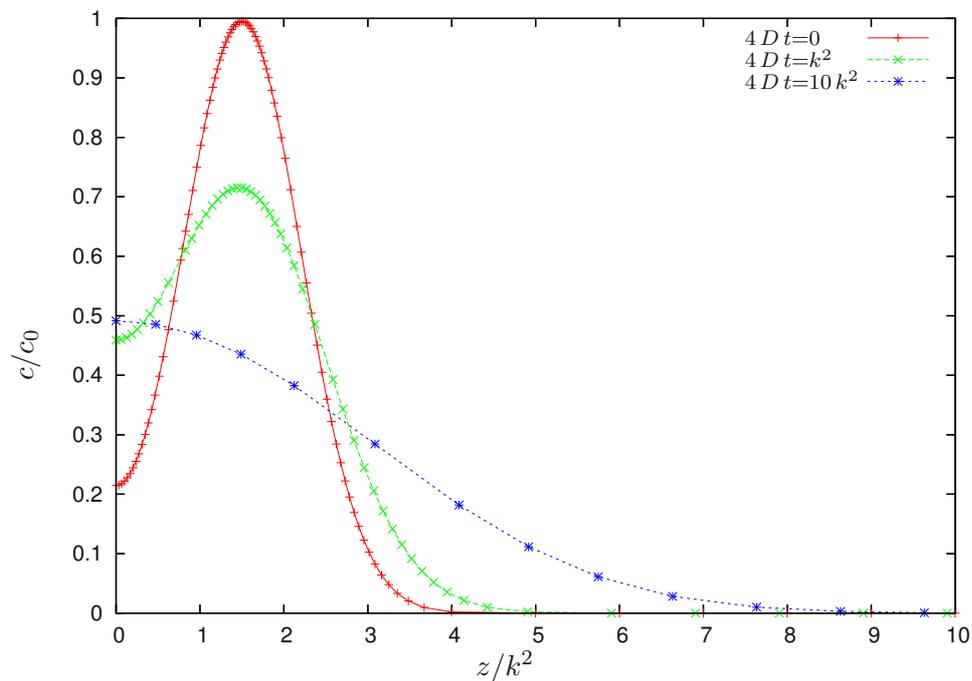


Figure 2.12: One-dimensional diffusion, diffusion constant D , $d = 1.5k$, no out-diffusion across $z = 0$. The initial concentration profile at $4Dt = 0$ and simulated distributions at $4Dt = k^2$ and $4Dt = 10k^2$ are shown.

A commonly used method of local adaptation is based on hierarchical mesh refinement. A given (relatively crude) grid is refined by splitting some grid elements into smaller ones. In frequent refinement steps, these elements can be split themselves. If the hierarchical splitting information is stored, this method offers also the possibility of hierarchical coarsening of the meshes, but this functionality results in some additional hierarchical information, which must be stored in the data structures. Additionally, care has to be taken that certain grid criteria must be fulfilled also for the refined grids — for instance the Delaunay criterion. Usually splitting a grid element into smaller ones may also cause splitting of neighboring grid elements and it must be guaranteed by the algorithm that this neighbor splitting is done only

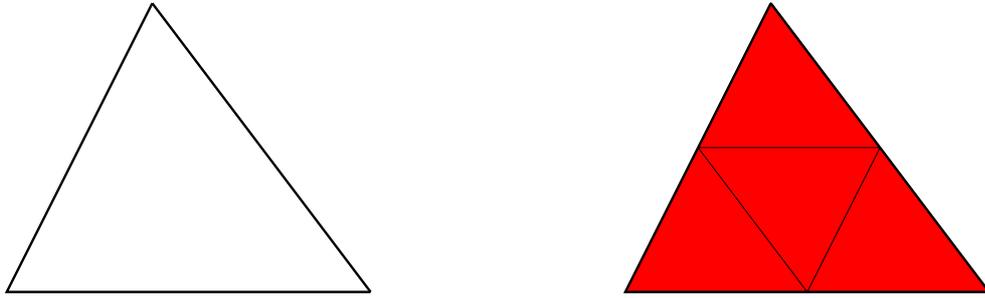


Figure 2.13: Red-refinement of a triangle.

locally and terminates, and that refining of a region does not induce a global split of all the grid elements, resulting in a never terminating splitting process.

Red-Green-Refinement

One method to perform such a refinement, is called red-green-refinement. This refinement technique is a commonly used refinement method for two-dimensional applications, especially for Delaunay grids. A triangle to be refined is split into four smaller children — so called red triangles, as shown in Figure 2.13. At the intersections of the triangle edges, new points are inserted. Thereby four triangles are generated, whose edges are connected to these points. A pleasant property of this method is that all new triangles are geometrically similar to the original one and thereby the element quality of the new triangles is the same as the quality of the original one. A positive aspect is that usually the condition of the resulting equation system will not become worse by the refinement — but it is also not improved.

Using this method, any neighboring triangles can be red-refined (Figure 2.14) and any of the red triangles can be red-refined itself. But red-refined triangles will produce extra points along an edge of neighboring triangles that are not refined to the same level. These triangles must be refined irregularly, which are labeled as

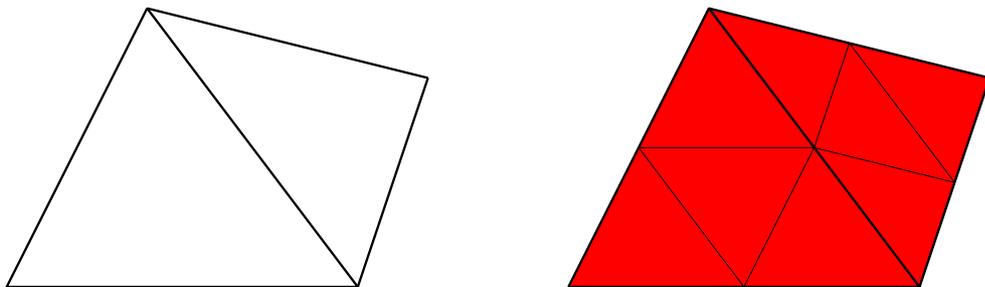


Figure 2.14: A red-refined triangle with a red-refined neighbor.

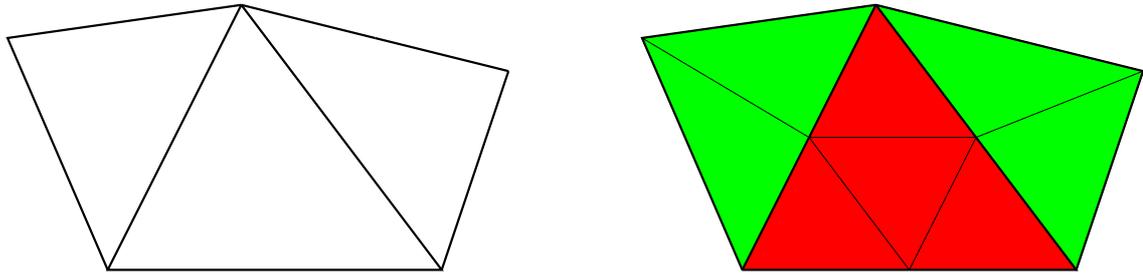


Figure 2.15: A red-refined triangle with green-refined neighbors.

green triangles and are of lower quality (shown in Figure 2.15). Green-refinement is only performed if a single point is inserted. If more points are inserted or higher refinement of this triangle is wanted, the green triangles are removed and the parent triangle is red-refined, too.

By continuation of these two strategies, a hierarchical mesh is produced. Red triangles can be surrounded by red triangles as well as green triangles, and red triangles can also be further red-refined, which may also induce the generation of surrounding green triangles. With a hierarchical memory representation, refinement and also coarsening of already refined triangles can be handled easily.

With the development of three-dimensional grids this technique has also been adapted to three dimensions [31]. A tetrahedron is red-refined by inserting the intersection points of the tetrahedron edges. By the insertion of these six points, the parent tetrahedron is split into two kinds of elements; four tetrahedrons and one octahedron (see Figure 2.16). On the one hand the octahedron can be split into four tetrahedrons (Figure 2.17). There are three possibilities according to the diagonal along which the octahedron is divided. Unlike for two-dimensional applications, not all the tetrahedrons are of the same geometrical shape. The tetrahedrons, which are created by splitting the octahedron, are spikier than the parents and therefore, of lower quality. Best results are obtained if the octahedron is split along its shortest diagonal. Consecutive refinement splits the resulting tetrahedrons. On the other hand this effect can be handled with mixed elements. A subsequent red-refinement of the octahedron delivers six smaller octahedrons and eight tetrahedrons. The resulting elements are geometrically similar to their parents or grandparents. After final refinement, the remaining octahedrons are usually decomposed to tetrahedrons by the common method (Figure 2.17).

Furthermore, for the handling of the surrounding green tetrahedrons, more cases must be considered. Theoretically, up to five points can be inserted (neglecting six point insertion, where this tetrahedron also is a red-refined tetrahedron). Only the three cases shown in Figure 2.19 are usually accounted for. Otherwise, the tetrahedron also is red-refined [39].

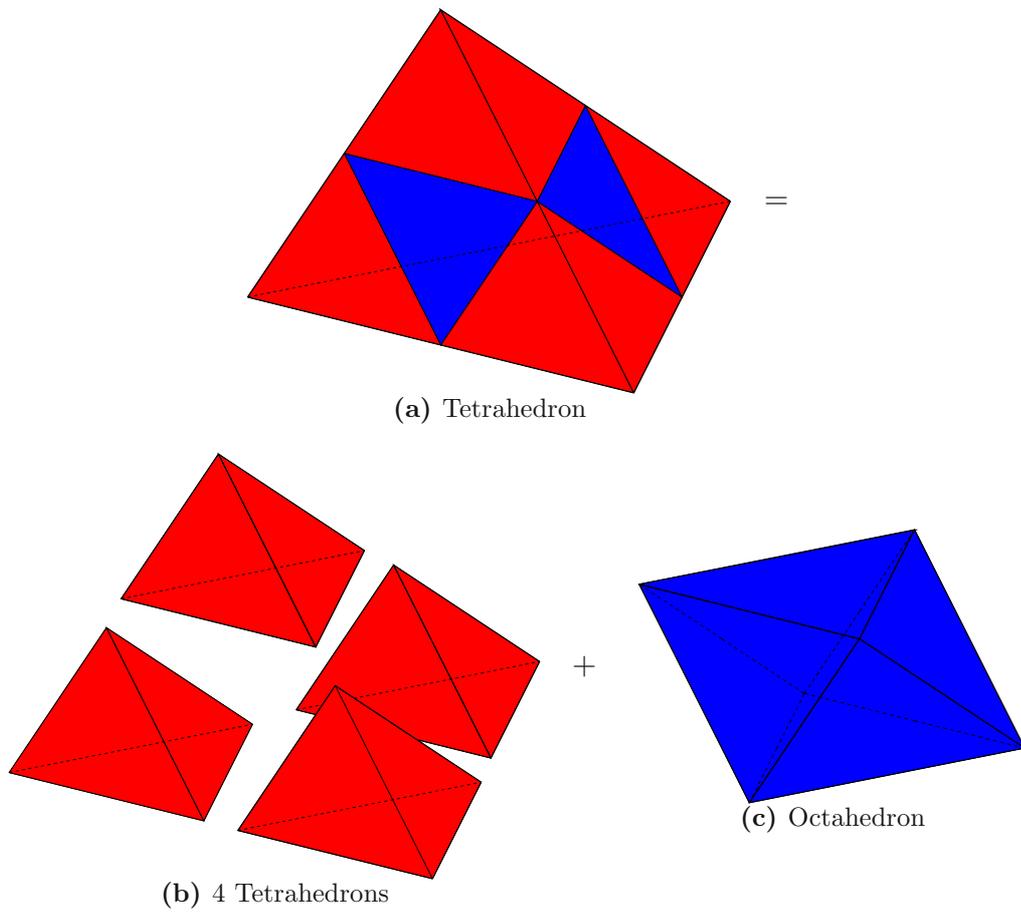


Figure 2.16: Red-refinement of a tetrahedron into children, 4 tetrahedrons and an octahedron.

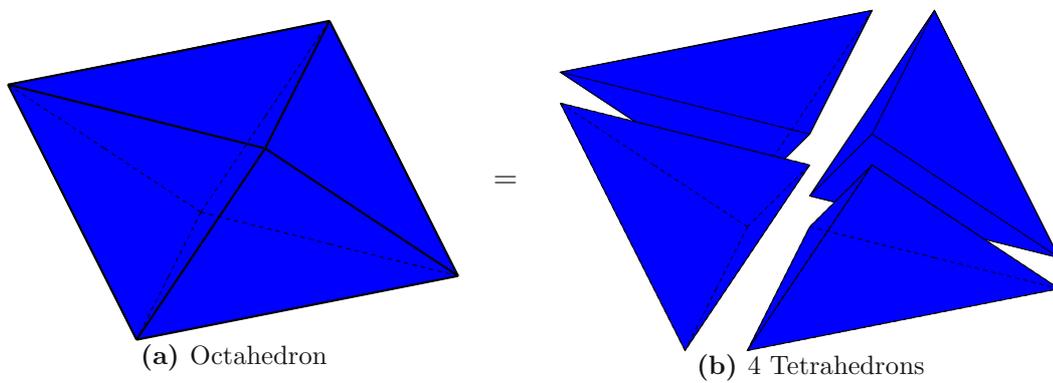


Figure 2.17: Final decomposition of the octahedron into 4 tetrahedrons.

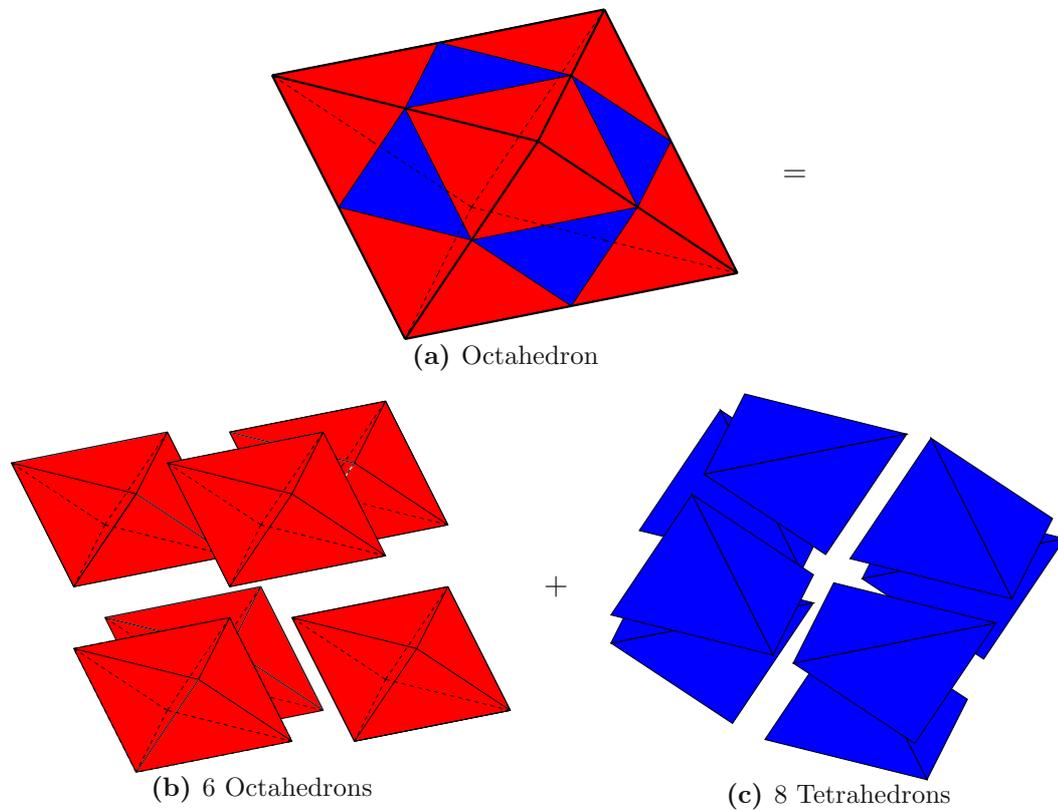


Figure 2.18: Subsequent red-refinement of the octahedron into 6 parent-similar octahedrons and 8 grandparent-similar tetrahedrons.

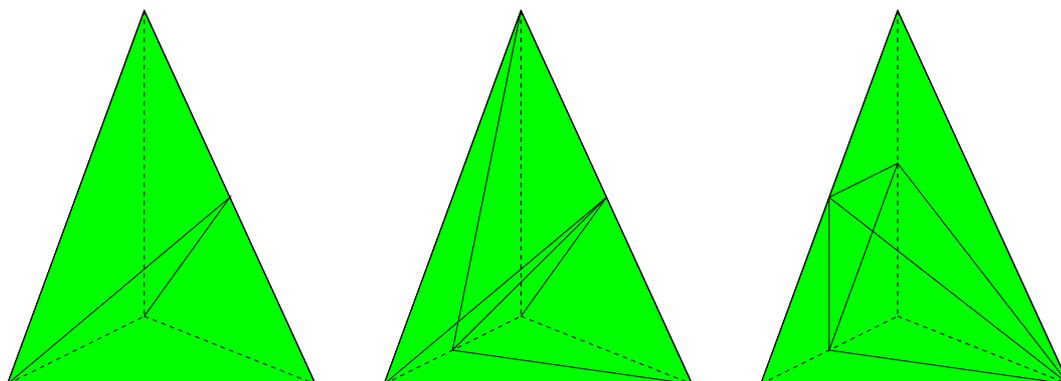


Figure 2.19: Possible green neighbors of a red-refined triangle.

Bisection and Projection

Other methods for grid refinement also involve the bisection of mesh edges. Different to red-refinement, the edges are not mandatorily split in the middle and only one edge of a tetrahedron is split. Each tetrahedron sharing this edge is decomposed into two tetrahedrons. Further refinement operates on the newly generated tetrahedrons.

Equally bisecting mesh edges prevents that the grid spacing changes too rapidly while the triangle angles can become more obtuse. Obtuse angles may be removed in later repairing steps by edge-flipping or similar. Special care must be taken, not to lose the hierarchical grid information potentially required for coarsening. Orthogonally projecting the split points from the opposite triangle vertex prevents the generation of obtuse triangles, but may change the grid spacing dramatically.

A new method for bisection was introduced by Rivara and Hitschfeld [21][50][51]. In the two-dimensional method, only the longest triangle edge is intersected. Assuming that the longest triangle edge is on the opposite side of the biggest triangle angle, also the biggest angle is split into nearly two equal parts. Doing so, a shrinking of small angles can be prevented. Within this algorithm, also surrounding triangles may split and care has to be taken that this cascading effect does not result in a never-ending bisecting process and a never terminating algorithm. Within three dimensions, this technique is extended by several refinement steps and the termination of this process is influenced by the order of the different refinement steps. A lot of research still has to be done for three-dimensional applications.

A two-dimensional example of a grid refined by line splitting can be seen in Figure 2.20. This version of the algorithm introduced in [29] is based on recursive bisecting of the longest edges of triangles and is capable of refinement and coarsening of previously refined areas upon to the base grid, too. The refinement is started on a relatively coarse basegrid. Fragments of the basegrid can be seen in the regions which surround the circular region. This grid is refined in areas of interest. These areas are marked by a criterion that is adapted to this problem. In the above example the refinement criterion is tailored for the simulation of electromigration [7].

2.6.2 Moving Boundaries

The deformation of mesh elements and moving mesh points occurs for instance during oxidation simulation. Often only local movement of the boundaries is expected while other portions of the mesh remain unchanged. Therefore, it is not feasible to start a new mesh generation every time a small portion of the grid moves slightly.

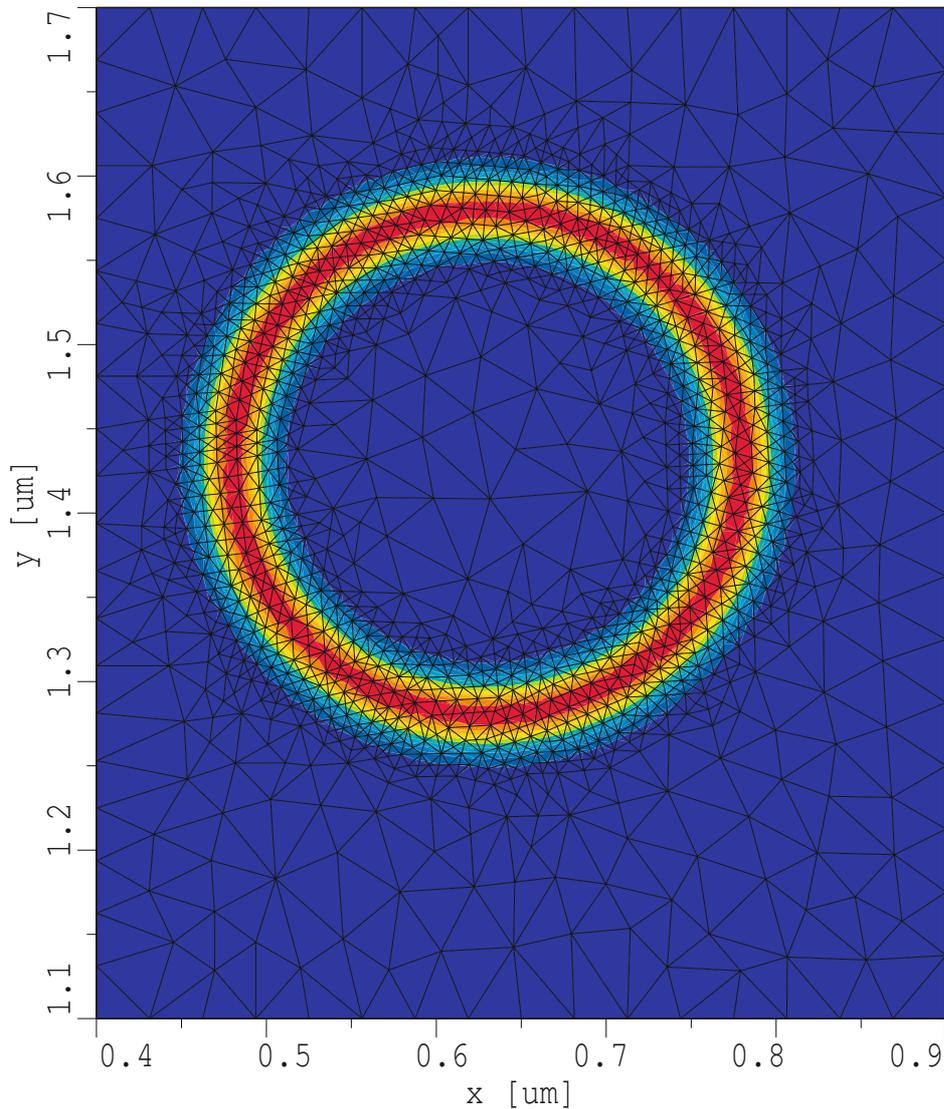


Figure 2.20: Two-dimensional grid refinement by line splitting.

During particular transient simulation steps of an oxidation simulation, the grid movement might be only small. However, difficulties are to be expected when the boundaries change qualitatively. If boundaries start to overlap a fusion of segment portions will arise.

When the mesh distortion is only local, the grid points can be moved and the involved grid elements are deformed. To avoid a too strong deformation, annealing techniques can be performed on the grid or a full regriding must be started. Within these methods, it is expensive to perform a collision detection of the growing grid elements. Further development of efficient methods will be necessary.

Other methods try to solve this problem by using diffuse boundaries. Thereby the grid points have an additional parameter that decides how much this point is part of the segment or the surrounding segment. With linear interpolation inside the tetrahedrons the exact boundary can be evaluated. These methods are known as levelset algorithms. After finishing the whole simulation, a global regridding for following simulations, which eliminates this fuzziness, must be performed. But within this method, the influence of the diffuse boundary must be accounted for within the differential equations to be solved. For two-dimensional problems, some approaches are available. For three dimensions, however, some difficulties still exist [1][8][32][47].

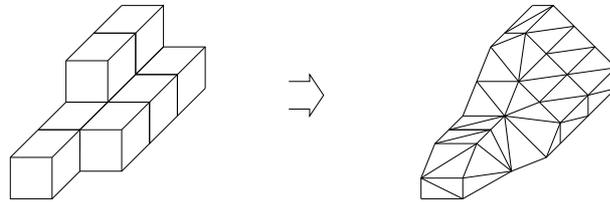
2.6.3 Cellular Data

For the simulation of etching and deposition processes, efficient algorithms have been developed which work with cellular data representations [38]. In contrast to a representation via meshes, the structure is discretized into small portions of matter. These are pixels within two dimensions and voxels in three dimensions (usually cubes). The device structures can easily be modified by adding and removing volume parts of these cubes, which makes this method of discretization perfectly suitable for performing process simulation which changes the topology. These topological modifications are managed in a straightforward manner by image processing algorithms. The major advantage is that topological changes of the structure can be handled without intensive surface tracing algorithms which avoid the formation of surface loops.

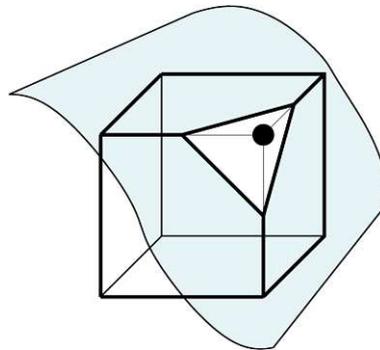
Unfortunately, for a good resolution of the structure, these voxels must be very small in size and the amount of stored data increases enormously. However, usually there is not much overhead in the amount of stored data as compared to more flexible grid representations. Within the commonly used device structures, normally some material layers are of such small thickness that the grid representation also requires a large number of grid elements to resolve the structures.

In order to transform the cellular representation to a polygonal surface description usually marching cube algorithms are applied [34]. The cube data only delivers a rippled (manhattan type) surface representation. Their description can be improved by handling the material type as integer values of the cube points so that a fairly smooth surface representation can be generated (see Figure 2.21(a) and Figure 2.21(b)) [30]. By allowing only bisectional splitting of the cube edges, the number of pathological cases can be limited; an excerpt of the remaining cases is shown in Figure 2.21(c).

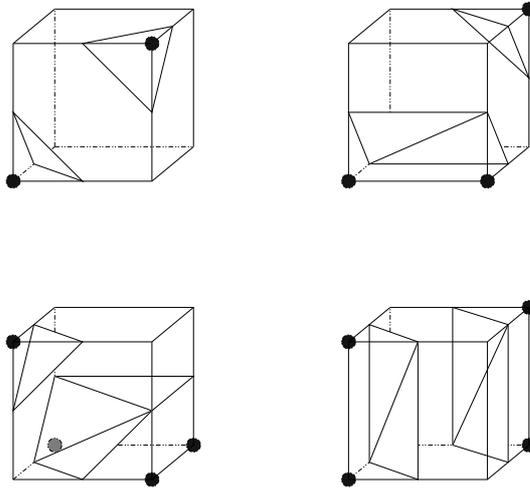
While a similarity of this method to the levelset algorithm can be observed, algorithms for extracting the surfaces have been developed that are more general and which work in a manner similar to levelset algorithms on ortho grids.



(a) Marching cubes — The surface is approximated by planes which result from edge bisectioning of the cubes.



(b) Approximation of the surface, the black point lies outside of the segment



(c) Some elementary cases which can result for cut cubes, the black points are cut away.

Figure 2.21: Overview of the marching cubes algorithm [30].

Considering that the extracted surface triangles are of the same size as the sampling cubes, the extracted surface data often consists of a large number of very small triangles. Therefore data reduction is necessary. While triangles lying in the same plane can be removed easily, the removal of ripples and staircases is more critical because structural corners and edges should be preserved. Certain surface coarsening and smoothing algorithms can be applied [30][58]. They may be based only on the surface representations or may be applied to solid modeled data of the already regridded representation. Special care must be taken of collision detection. An overlap of thin material layers must be prevented by the algorithm. Several techniques are known, their amount of smoothening and data reduction varies and strongly depends on the structures and methods. A suitable way must be found individually for each application [23][58].

Chapter 3

The Box Integration Method

A WIDELY USED METHOD for numerical discretization is the method of Box Integration, also known as Finite Boxes, Finite Volumina method for three dimensions, or Finite Areas for two dimensions. The applicability of this methodology is closely coupled to Delaunay grids, described before [35].

In the field of electrical device simulation, several differential equations are applied. The first group is derived by stationary electric field calculations as a result of applying the Maxwell equations. A similar differential equation type is obtained by observing diffusion or thermal conductance processes, with the difference of an additional time dependent term. These types of differential equations are described first. The second class results from the semiconductor equations, in its simplest form the drift-diffusion model. This type will be described later.

3.1 The Poisson Equation

The Poisson equation is fundamental for all electrical applications. The derivation is shown for a stationary electric field [26]. For the derivation, the material parameters may be inhomogeneous, locally dependent but not a function of the electric field. In this section, the principle of the discretization is demonstrated. In the case of a low field dependency, a recursive reinsertion of newly derived material parameters may result in stable solutions [54]. A highly nonlinear behavior has to be managed by linearization and Newton iterations, for instance [2][4][60].

The Maxwell equations for a stationary electric field reduce to

$$\operatorname{curl} \mathbf{E} = 0, \quad (3.1)$$

$$\operatorname{div} \mathbf{D} = \rho. \quad (3.2)$$

\mathbf{E} and \mathbf{D} are related by the permittivity $\underline{\epsilon}$

$$\mathbf{D} = \underline{\epsilon} \mathbf{E}. \quad (3.3)$$

Physical properties require that the permittivity tensor $\underline{\epsilon}$ is symmetric and positive definite for lossless media (or a positive scalar for isotropic media). Equation (3.1) is satisfied implicitly by introducing the electric potential φ

$$\mathbf{E} = -\operatorname{grad} \varphi. \quad (3.4)$$

Combing (3.1)–(3.4), the Maxwell equations can be rewritten as

$$\operatorname{div} (\underline{\epsilon} \operatorname{grad} \varphi) = -\rho. \quad (3.5)$$

This can be transformed to its integral formulation. By applying the Gaussian integral formula (3.5) results in

$$\int_V \operatorname{div} (\underline{\epsilon} \operatorname{grad} \varphi) \, dv = \oint_{\partial V} (\underline{\epsilon} \operatorname{grad} \varphi) \cdot d\mathbf{A} = - \int_V \rho \, dv. \quad (3.6)$$

Equation (3.6) must be fulfilled within any arbitrary volume V , with ∂V being the surface of this volume. While performing Box Integration, this formula must be satisfied in the Voronoi boxes of each grid point. In Figure 3.1 a typical Voronoi box of a point p_i , belonging to all the tetrahedrons connected to the point p_i , can be seen. The integration can be split into the integration over the different box parts, caused by all the tetrahedrons $tets_i$ which share the same point p_i

$$\sum_{t \in tets_i} \int_{A_{t,i}} (\underline{\epsilon} \operatorname{grad} \varphi) \cdot d\mathbf{A} = - \int_{V_i} \rho \, dv. \quad (3.7)$$

Such a part of the Voronoi box within the tetrahedron $\langle p_i p_j p_k p_l \rangle$ is shown in Figure 3.2. Each part around the point p_i can also be split into the three area contributions that are caused by the different edges $\langle p_i p_j \rangle$, $\langle p_i p_k \rangle$ and $\langle p_i p_l \rangle$ (see Figure 3.3)

$$\sum_{\substack{t \in tets_i, \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} (\underline{\epsilon} \operatorname{grad} \varphi) \cdot d\mathbf{A} = - \int_{V_i} \rho \, dv. \quad (3.8)$$

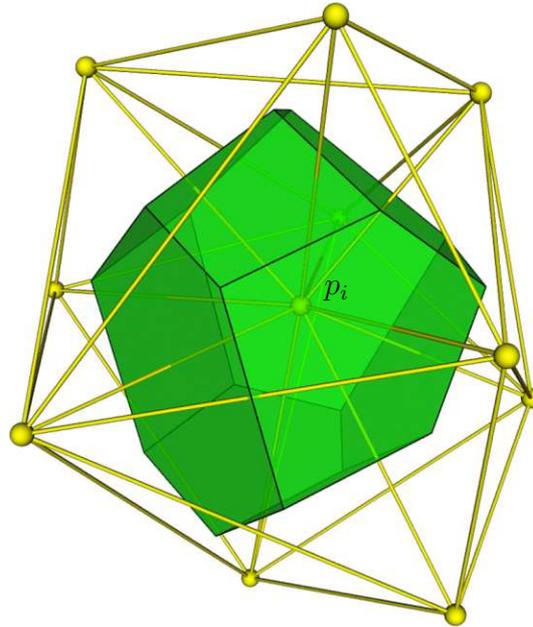


Figure 3.1: Tetrahedrons around point p_i with the resulting Voronoi box constructed by these tetrahedrons.

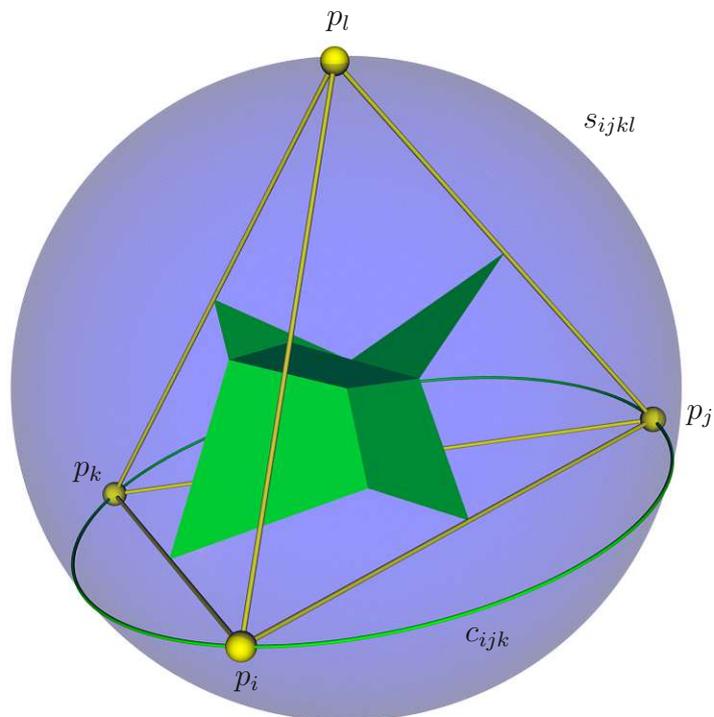


Figure 3.2: Box parts of the tetrahedron $\langle p_i p_j p_k p_l \rangle$ with drawn outer-sphere s_{ijkl} of the tetrahedron and outer-circle c_{ijk} of the triangle $\langle p_i p_j p_k \rangle$.

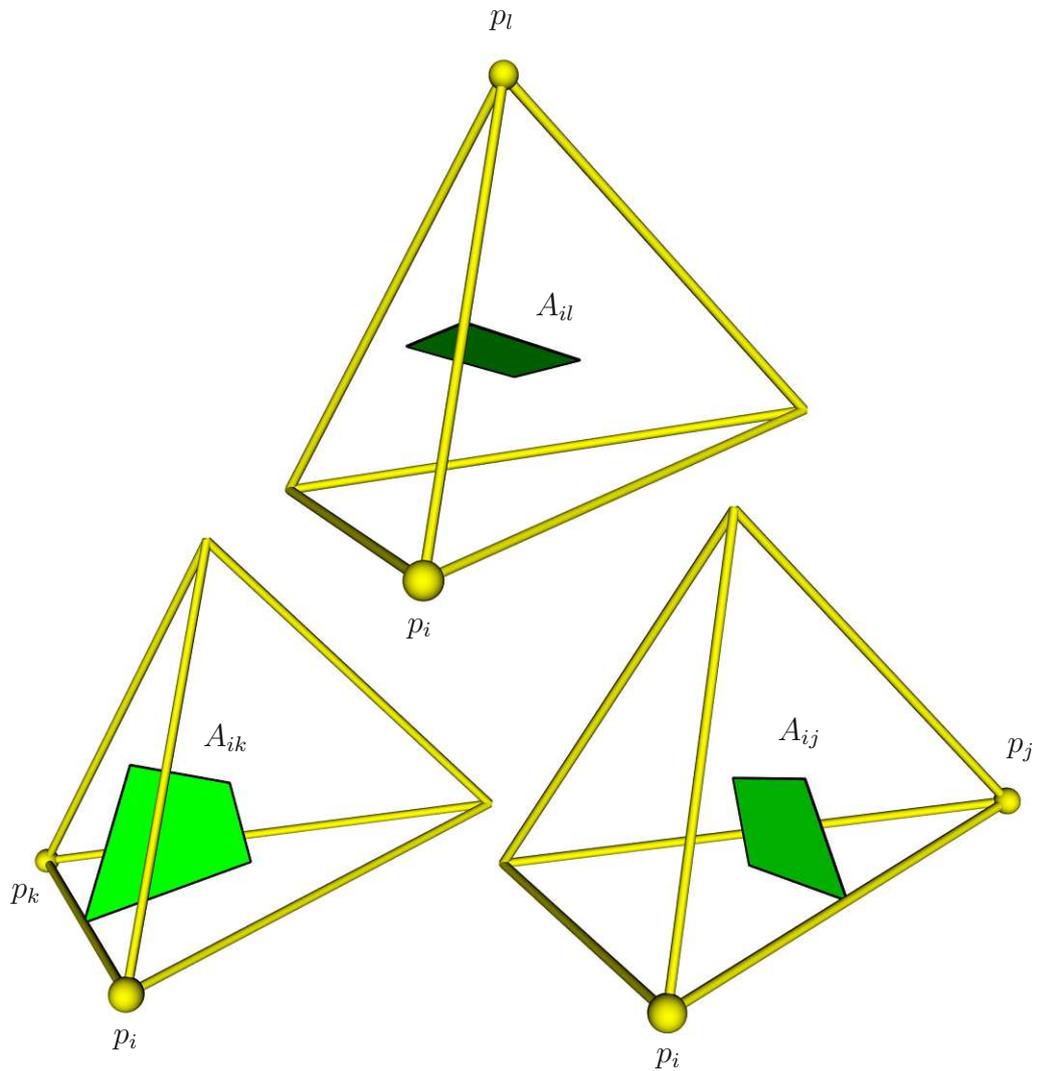


Figure 3.3: Different Voronoi faces which share the point p_i of a tetrahedron.

Using Box Integration, an approximation of first order is usually assumed, which means that the electric potential φ within one tetrahedron is a linear function along the axes or, in other words, the electric field \mathbf{E} is constant within each tetrahedron. Material parameters are assumed to be constant within the tetrahedron, too. The electric charge density ρ on the right-hand-side is assumed to be constant within the whole Voronoi box as well and can therefore be analytically calculated and written

as the electric charge q_i of the Voronoi box of the point p_i , which results in

$$-\sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} (\underline{\varepsilon} \mathbf{E}) \cdot d\mathbf{A} = -\int_{V_i} \varrho_i dv = -q_i. \quad (3.9)$$

As in the usual applications the permittivity tensor reduces to a scalar, which means the electric displacement and the electric field have the same direction, it follows

$$(\varepsilon \mathbf{E}) \cdot d\mathbf{A} = \varepsilon (\mathbf{E} \cdot d\mathbf{A}) = \varepsilon (\mathbf{E} \cdot \mathbf{n}) dA = \varepsilon E_n dA \quad (3.10)$$

and equation (3.9) can be simplified to

$$-\sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \varepsilon E_n dA = -\int_{V_i} \varrho_i dv = -q_i. \quad (3.11)$$

3.1.1 Approximation of E_n and ε

Usually the integral (3.11) is approximated by

$$-\sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \varepsilon_{ij} E_{ij} A_{t,ij} = -q_i, \quad (3.12)$$

with

$$E_{ij} \approx \frac{\varphi_i - \varphi_j}{d_{ij}} \quad \text{and} \quad \varepsilon_{ij} \approx \frac{\varepsilon_i + \varepsilon_j}{2}. \quad (3.13)$$

An estimation of the validity of this approximation based on the homogenous Poisson equation (Laplace equation) will be explained under the conditions shown in Figure 3.4. (ξ, η, ζ) spans a local Cartesian coordinate system with origin p_i . In the one-dimensional case and with constant permittivity the potential varies linearly with position as

$$\varphi(\xi) = \varphi_i - E_{ij} \xi. \quad (3.14)$$

Whereas with non-constant permittivity we have at first order

$$\tilde{\varepsilon}(\xi) = \varepsilon_i + \frac{\Delta\varepsilon}{d_{ij}} \xi, \quad (3.15)$$

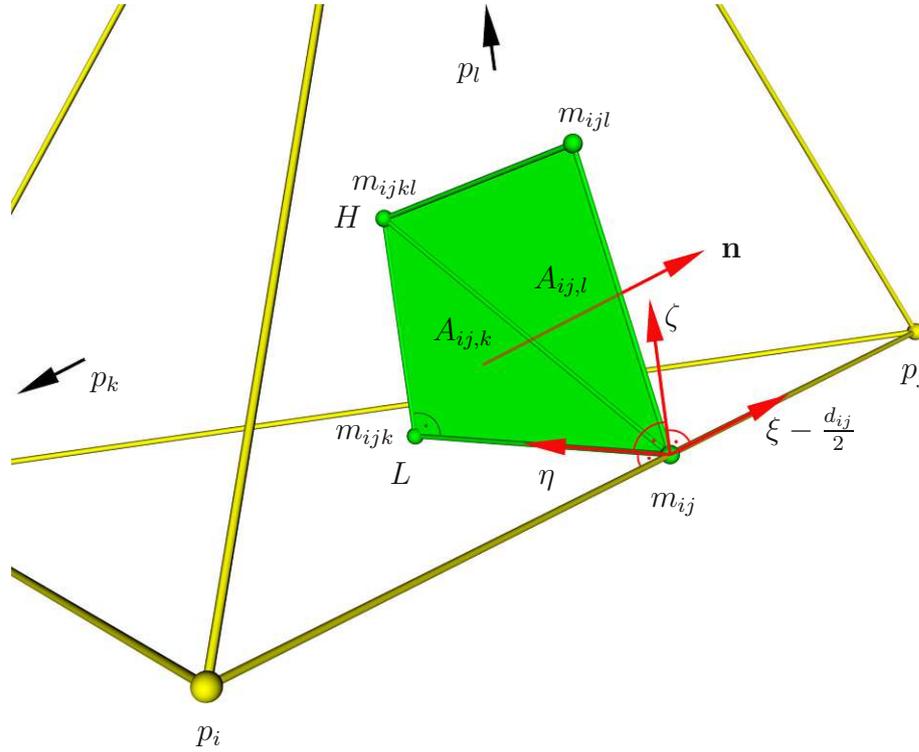


Figure 3.4: A detail of the tetrahedron $\langle p_i p_j p_k p_l \rangle$ shown in Figure 3.3 which shows the coupling Voronoi region between the point p_i and p_j . This region can be split into the two triangular components $A_{ij,k}$ and $A_{ij,l}$ which are spawned by the midpoint m_{ij} of the edge $\langle p_i p_j \rangle$, the midpoint m_{ijk} of the outer-circle of the triangle $\langle p_i p_j p_k \rangle$ and the midpoint m_{ijl} of the outer-sphere of the tetrahedron $\langle p_i p_j p_k p_l \rangle$, or m_{ij} , m_{ijl} , and m_{ijkl} , respectively.

with

$$\tilde{\varepsilon}(0) = \varepsilon_i \quad \text{and} \quad \tilde{\varepsilon}(d_{ij}) = \varepsilon_j \quad (3.16)$$

and therefore,

$$\Delta\varepsilon = \varepsilon_j - \varepsilon_i. \quad (3.17)$$

The continuity of the electric displacement

$$\tilde{\varepsilon}(0) \tilde{E}(0) = \tilde{\varepsilon}(\xi) \tilde{E}(\xi) = \tilde{\varepsilon}(d_{ij}) \tilde{E}(d_{ij}), \quad (3.18)$$

with

$$\tilde{E} = -\frac{\partial \tilde{\varphi}}{\partial \xi} \quad (3.19)$$

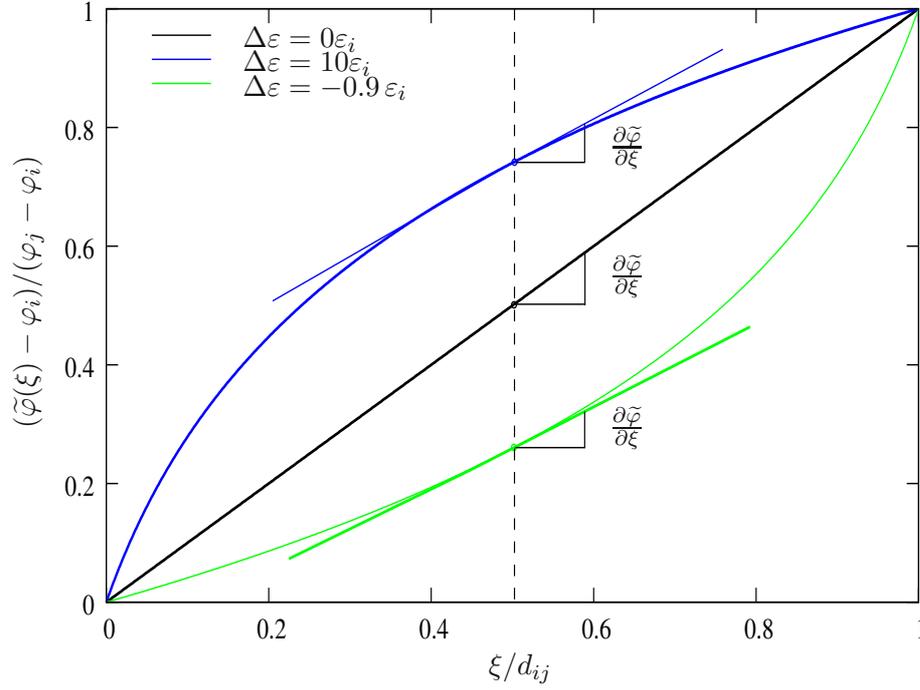


Figure 3.5: Potential distribution between two grid points p_i and p_j with different permittivities.

delivers the deformed distribution of $\tilde{\varphi}(\xi)$

$$\varphi_i - \tilde{\varphi}(\xi) = (\varphi_i - \varphi_j) \frac{\ln\left(1 + \frac{\Delta\varepsilon}{\varepsilon_i} \frac{\xi}{d_{ij}}\right)}{\ln\left(1 + \frac{\Delta\varepsilon}{\varepsilon_i}\right)} \quad (3.20)$$

and therefore,

$$\tilde{E}\left(\frac{d_{ij}}{2}\right) = -\frac{\partial\tilde{\varphi}}{\partial\xi}\Bigg|_{\frac{d_{ij}}{2}} \neq \frac{\varphi_i - \varphi_j}{d_{ij}} = E_{ij}. \quad (3.21)$$

The resulting potential distributions $\tilde{\varphi}(\xi) - \varphi_i$ between p_i and p_j for different values of $\Delta\varepsilon$ are shown in Figure 3.5. The overestimation of the electric field $E_{ij}/\tilde{E}(d/2)$ in the middle of the grid element in dependence of the permittivity change inside the element is shown in Figure 3.6. According to this figure, a limit for the validity of approximation (3.13) can be defined by

$$|\varepsilon_i - \varepsilon_j| < k \min(\varepsilon_i, \varepsilon_j) \quad \forall i, j : \exists \text{ edge } \langle p_i p_j \rangle \quad (3.22)$$

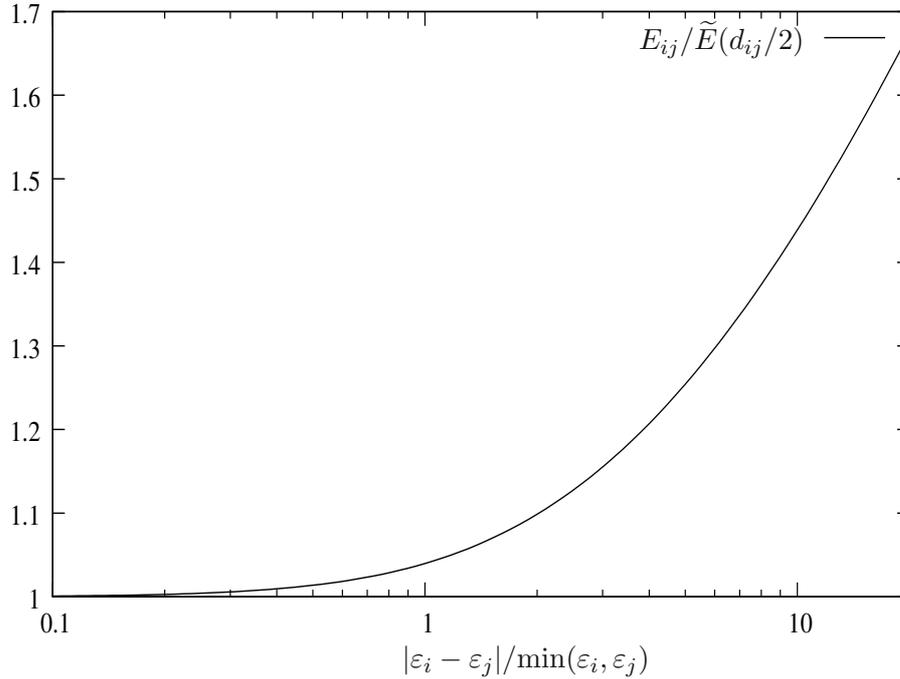


Figure 3.6: Ratio of the approximated electric field E_{ij} to the exact solution $\tilde{E}(d_{ij}/2)$ at the center of the control volume.

or a limit for continuously defined permittivity

$$|\text{grad } \varepsilon_i| d_{ij} < k \varepsilon_i \quad \forall i, j : \exists \text{ edge } \langle p_i p_j \rangle. \quad (3.23)$$

With $k \approx 2$, the estimation error of E_{ij} is limited to 10%.

On a three-dimensional grid (refer to Figure 3.4), the spatial derivatives of ε cause an additional contribution to the discretization error. With linear interpolation of ε and φ , equation (3.11) results in

$$\begin{aligned} & - \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \tilde{\varepsilon} \tilde{E}_n \, dA = - \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \tilde{\varepsilon} \begin{pmatrix} \tilde{E}_\xi \\ \tilde{E}_\eta \\ \tilde{E}_\zeta \end{pmatrix} \cdot \begin{pmatrix} dA \\ 0 \\ 0 \end{pmatrix} = \\ & - \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \tilde{\varepsilon} \tilde{E}_\xi \, dA = \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \tilde{\varepsilon} \frac{\partial \tilde{\varphi}}{\partial \xi} \, dA = \\ & \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_0^{L_{ij}} \int_0^{\eta \frac{H_{ij}}{L_{ij}}} \left(\varepsilon_i + \frac{\Delta \varepsilon_\xi}{d_{ij}} \xi + \frac{\Delta \varepsilon_\eta}{d_{ij}} \eta + \frac{\Delta \varepsilon_\zeta}{d_{ij}} \zeta \right) \frac{\Delta \varphi_\xi}{d_{ij}} \, d\eta \, d\zeta = -q_i \end{aligned} \quad (3.24)$$

and with $LH = 2A_{ij}$

$$\begin{aligned} & \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \left(\varepsilon_i + \frac{\Delta \varepsilon_\xi}{d_{ij}} \frac{d_{ij}}{2} \right) \frac{\Delta \varphi_\xi}{d_{ij}} \frac{LH}{2} + \frac{\Delta \varepsilon_\eta}{d_{ij}} \frac{\Delta \varphi_\xi}{d_{ij}} \frac{L^2 H}{3} + \frac{\Delta \varepsilon_\zeta}{d_{ij}} \frac{\Delta \varphi_\xi}{d_{ij}} \frac{LH^2}{6} = \\ & \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \frac{\varepsilon_i + \varepsilon_j}{2} \Delta \varphi_\xi \frac{A_{ij}}{d_{ij}} + \Delta \varepsilon_\eta \Delta \varphi_\xi \frac{A_{ij}}{d_{ij}} \frac{2L}{3} + \Delta \varepsilon_\zeta \Delta \varphi_\xi \frac{A_{ij}}{d_{ij}} \frac{H}{3} = -q_i. \end{aligned} \quad (3.25)$$

The first addend of equation (3.25) equals the value given by discretization (3.12) and (3.13) whereas the remaining terms cause an additional discretization error. Often the derivative components of neighboring tetrahedrons compensate each other, but not always. Even the lengths of L and H are not limited to lie within the tetrahedron ranges (for instance, obtuse angled triangles for two dimensions) and a limitation for this approximation is difficult. For the whole simulation domain the estimation

$$|\text{grad } \varepsilon_i|_{\max} d_{ij, \max} < k \varepsilon_{i, \min} \quad \forall i, j : \exists \text{ edge } \langle p_i p_j \rangle \quad (3.26)$$

seems plausible, but is too pessimistic as in most cases L and H have the same order of magnitude as their based tetrahedrons. The previously derived estimation (3.23) is therefore also applicable to the three-dimensional case.

Another common approximation method for the effective permittivity ε_{ij} is [27]

$$\varepsilon_{ij} \approx \frac{2 \varepsilon_i \varepsilon_j}{\varepsilon_i + \varepsilon_j} \quad (3.27)$$

which shows the behavior of a serial connection of two capacitors (or resistors for electrical flow fields). This approximation has the advantage that if the permittivity (more likely for the conductivity) is zero in a Voronoi box, no flux enters the box. Nevertheless, this zero flux leaves the electric potential of the included grid point undefined and may cause problems for the equation solver, applied later on.

3.1.2 Continuation of the Discretization

However, equation (3.11) is discretized as

$$- \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \int_{A_{t,ij}} \varepsilon_{ij} E_{ij} \, dA = \sum_{\substack{t \in \text{tets}_i \\ \forall \langle p_i p_j \rangle}} \varepsilon_{ij} \frac{\varphi_j - \varphi_i}{d_{ij}} A_{t,ij} = -q_i. \quad (3.28)$$

In compliance with the decomposition of the Voronoi regions in Section 2.4.3, $A_{t,ij}$ is the right signed value of the Voronoi area between p_i and p_j of the tetrahedron t . Which means in detail, area portions with

- outer-circle inside the appropriate triangle and outer-sphere inside the tetrahedron counted positive,
- outer-circle outside the appropriate triangle and outer-sphere outside the tetrahedron counted positive, and
- one of them inside and the other one outside counted negative.

Combining the portions of the tetrahedrons which share the point p_i delivers the commonly used notation

$$\sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \varepsilon_{ij} \frac{A_{ij}}{d_{ij}} (\varphi_i - \varphi_j) = q_i \quad (3.29)$$

with A_{ij} the Voronoi area of the edge $\langle p_i p_j \rangle$ (coupling area between p_i and p_j)

$$A_{ij} = \sum_{\substack{t \in \text{tets}_i \\ \langle p_i p_j \rangle \in t}} A_{t,ij}. \quad (3.30)$$

Equation (3.29) must be fulfilled for each grid point p_i inside the simulation domain, which results in an equation system

$$\mathbf{S} \cdot \mathbf{x} = \mathbf{B} \quad (3.31)$$

with the system matrix $\mathbf{S} = (s_{ij})$

$$s_{ij} = -\varepsilon_{ij} \frac{A_{ij}}{d_{ij}} \quad \forall i, j : \exists \text{ edge } \langle p_i p_j \rangle, \quad (3.32)$$

$$s_{ii} = \sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \varepsilon_{ij} \frac{A_{ij}}{d_{ij}} = - \sum_{\forall j : i \neq j} s_{ij} \quad \forall i, \quad (3.33)$$

the unknowns

$$\mathbf{x} = (\varphi_i), \quad (3.34)$$

and the right-hand side

$$\mathbf{B} = (q_i), \quad (3.35)$$

which is defined for all inner points p_i of the domain.

3.1.3 Boundary Conditions

For solving this time-invariant elliptical partial differential equation, boundary conditions have to be applied to all boundaries. For each grid point lying at the boundary of the domain one of the two following additional equations or a linear-combination of them must be added for the discrete version, which fulfill the boundary conditions.

Each grid point of the boundary has to be defined by exactly one of the different conditions. Additionally, at least one grid point must be described by a Dirichlet condition to define a definite potential distribution. With this set of equations, the discrete Poisson equation can be solved.

Dirichlet Conditions

The first class of boundary conditions is an imposed potential Φ

$$\varphi = \Phi \quad \text{on boundary } \partial B_\Phi. \quad (3.36)$$

The discrete formulation of this boundary condition simply replaces the respective rows of equation (3.29) by

$$\varphi_i = \Phi_i \quad \forall i : p_i \in \partial B_\Phi \quad (3.37)$$

or within the matrix representation

$$s_{ij} = 0 \quad \forall i, j : i \neq j, p_i \in \partial B_\Phi, \quad (3.38)$$

$$s_{ii} = 1 \quad \forall i : p_i \in \partial B_\Phi, \quad (3.39)$$

$$b_i = \Phi_i \quad \forall i : p_i \in \partial B_\Phi, \quad (3.40)$$

which at first delivers an asymmetric system matrix \mathbf{S} . Since most equation solvers require a symmetric system matrix, it may be necessary to transform the matrix to a symmetric one, which can be simple obtained by summation of adequate rows.

Requirements for an M-Matrix

As shown in Section 2.3, for satisfying the maximum principle the system matrix \mathbf{S} has to be an M-matrix. Under the precondition of a valid Delaunay grid, the validity of the M-matrix will be shown.

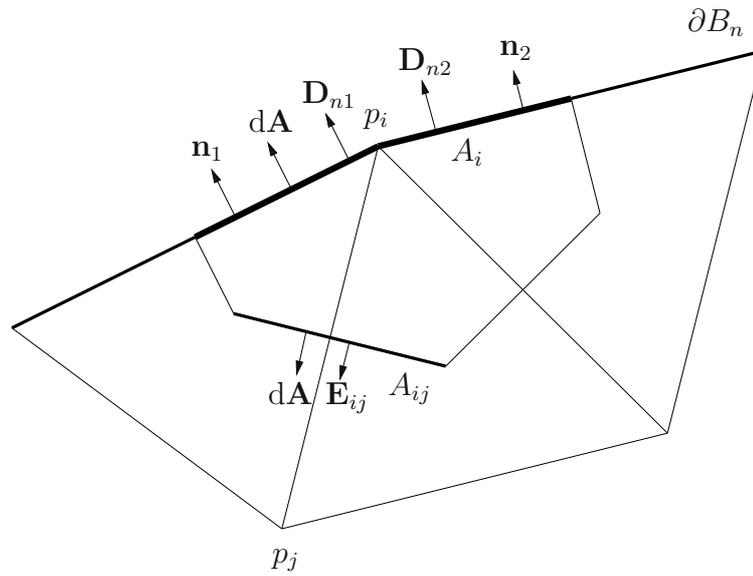


Figure 3.7: A Voronoi box of the point p_i at a boundary.

As the grid is Delaunay, it follows that the coupling areas are positive

$$A_{ij} \geq 0$$

and therefore,

$$(3.32) \quad s_{ij} \leq 0 \quad \text{satisfies (2.4),}$$

$$(3.33) \quad - \sum_{\forall j: i \neq j} s_{ij} = s_{ii} \geq 0 \quad \text{satisfies (2.5),}$$

$$(3.33) \quad - \sum_{\forall j: i \neq j} s_{ij} = s_{ii} = \sum_{\forall j: i \neq j} |s_{ij}| \quad \text{satisfies (2.6),}$$

and finally by defining at least one boundary point $p_{\Phi,k}$ as a Dirichlet point

$$(3.38)-(3.39) \quad 1 = s_{kk} > \sum_{\forall j: k \neq j} |s_{kj}| = 0 \quad \text{satisfies (2.7).}$$

Neumann Conditions

The second type of boundary condition specifies the imposed normal component $\mathbf{n} \cdot \mathbf{D} = D_n$ on the boundaries ∂B_n (see Figure 3.7), which occurs between different segments. As shown in Figure 3.7, the normal vector \mathbf{n} may have various directions on different boundary pieces around p_i . This behavior could be handled by splitting

the boxes (and also the point p_i itself) into several pieces. Usually this approach is prevented and the normal derivatives are combined to one value. In this case, the boundary conditions result in a distortion of (3.29)

$$\sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \varepsilon_{ij} \frac{A_{ij}}{d_{ij}} (\varphi_i - \varphi_j) + D_{n,i} A_i = q_i \quad \forall i : p_i \in \partial B_n \quad (3.41)$$

with $D_{n,i}$ the assigned normal component at the boundary at the grid point p_i . It must be distinguished between A_{ij} which represents the area of the Voronoi box between p_i and p_j , and A_i as the assigned boundary area of the Voronoi box around the grid point p_i .

In the majority of cases, a special kind of this boundary condition is used, which shows a vanishing flux across the boundary (homogenous Neumann condition)

$$D_n = 0. \quad (3.42)$$

This type of condition specifies the artificial boundaries which terminate the simulation domains. As no distortion of equation (3.41) appears, these boundaries do not require a special treatment and the equations can be left unchanged as in (3.31)–(3.35).

Cauchy Boundary Conditions

The linear-combination

$$D_n + \alpha \varphi = \beta \quad (3.43)$$

combines the two previous conditions and results in the discrete formulation

$$\sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \varepsilon_{ij} \frac{A_{ij}}{d_{ij}} (\varphi_i - \varphi_j) + (\beta_i - \alpha_i \varphi_i) A_i = q_i \quad \forall i : p_i \in \partial B_c. \quad (3.44)$$

3.2 The Diffusion Equation

In this section, the discretization of parabolic time-variant problems is described. In its simplest representation, the right-hand side of the diffusion equation is time-invariant. Solving a diffusion problem the diffusion equation becomes time-variant. It describes the out-diffusion of matter, driven by its own concentration gradient. The diffusion flux \mathbf{J} can be written as

$$\mathbf{J} = -D \operatorname{grad} c \quad (3.45)$$

where D denotes the diffusion coefficient and c is the concentration of the diffusing material.

Additionally, the conservation of material must be fulfilled

$$\operatorname{div} \mathbf{J} = -\frac{\partial c}{\partial t}. \quad (3.46)$$

After insertion of (3.46) in (3.45), the diffusion equation can be reformulated as

$$\operatorname{div} \operatorname{grad} c = \frac{1}{D} \frac{\partial c}{\partial t}. \quad (3.47)$$

As described in the previous section, this equation can be discretized as

$$\sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \frac{A_{ij}}{d_{ij}} (c_j - c_i) = \frac{1}{D} \left(\frac{\partial}{\partial t} c_i \right) V_i \quad (3.48)$$

The time derivative in this formula can be discretized by several methods. By the backward Euler method, the time derivative is approximated by [66]

$$\frac{\partial}{\partial t} c(t + \Delta t) \approx \frac{c(t + \Delta t) - c(t)}{\Delta t} = \frac{c^{k+1} - c^k}{\Delta t}, \quad (3.49)$$

with Δt the sampling interval. The discrete notation by backward Euler time discretization follows by

$$\sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \frac{A_{ij}}{d_{ij}} (c_j^{k+1} - c_i^{k+1}) = \frac{1}{D} \frac{c_i^{k+1} - c_i^k}{\Delta t} V_i. \quad (3.50)$$

By separating the unknowns to the left-hand side, this expression becomes

$$\frac{1}{\Delta t} c_i^{k+1} V_i + D \left(- \sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \frac{A_{ij}}{d_{ij}} c_j^{k+1} + \sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \frac{A_{ij}}{d_{ij}} c_i^{k+1} \right) = \frac{1}{\Delta t} c_i^k V_i. \quad (3.51)$$

In matrix notation, (3.51) can be written as

$$\mathbf{K} \cdot \mathbf{x} = \mathbf{B}, \quad (3.52)$$

with

$$\mathbf{K} = \frac{1}{\Delta t} \mathbf{M} + D \mathbf{S}, \quad (3.53)$$

$$x_i = c_i^{k+1} \quad \forall i, \quad (3.54)$$

$$m_{ij} = 0 \quad \forall i, j : i \neq j, \quad (3.55)$$

$$m_{ii} = V_i \quad \forall i, \quad (3.56)$$

$$s_{ij} = -\frac{A_{ij}}{d_{ij}} \quad \forall i, j : \exists \text{ edge } \langle p_i p_j \rangle, \quad (3.57)$$

$$s_{ii} = \sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} \frac{A_{ij}}{d_{ij}} = - \sum_{\forall j : \exists \text{ edge } \langle p_i p_j \rangle} s_{ij} \quad \forall i, \quad (3.58)$$

$$b_i = \frac{1}{\Delta t} V_i c_i^k \quad \forall i. \quad (3.59)$$

The necessary boundary conditions for this parabolic equation can be handled as shown in the previous section. Also the requirements for an M-matrix (see Section 2.3) are satisfied. The conditions for \mathbf{S} can be handled as in the previous section and \mathbf{M} consists of positive diagonal entries only. Additionally, an initial condition is required.

3.2.1 Initial Conditions

The initial concentration distribution at initial time τ is defined as

$$c(\mathbf{x}, \tau) = C(\mathbf{x}). \quad (3.60)$$

The discrete system is satisfied by the discrete formulation

$$c_i(\tau) = c_i^0 = C_i \quad \text{for all grid points } p_i. \quad (3.61)$$

The concentration distributions $c_i^1, c_i^2, \dots, c_i^{k+1}$ can be computed by an sequential evaluation of (3.52)–(3.59).

3.3 The Basic Semiconductor Equations

The classic drift-diffusion equations for electrons and holes read¹

$$\mathbf{J}_n = -q \mu_n n \text{ grad } \varphi + q D_n \text{ grad } n, \quad (3.62)$$

$$\mathbf{J}_p = -q \mu_p p \text{ grad } \varphi - q D_p \text{ grad } p, \quad (3.63)$$

with $\mathbf{J}_{n,p}$ the electron or hole current densities, n, p the electron or hole concentrations, $\mu_{n,p}$ and $D_{n,p}$ the carrier mobilities and diffusivities, and finally q the unit charge of the electron (positive). These equations are connected via the recombination rate R

$$\text{div } \mathbf{J}_n - q \frac{\partial n}{\partial t} = q R, \quad (3.64)$$

$$\text{div } \mathbf{J}_p + q \frac{\partial p}{\partial t} = -q R \quad (3.65)$$

and coupled to the already discussed Poisson equation, where the charge is determined via the difference between the ionized impurities N_C and the electron and hole concentrations

$$\text{div } (\varepsilon \text{ grad } \varphi) = q (n - p - N_C). \quad (3.66)$$

The continuity equations (3.64) and (3.65) are converted to integral formulation

$$\oint_{\partial V} \mathbf{J}_n \cdot d\mathbf{A} = q \int_V \left(R + \frac{\partial n}{\partial t} \right) dv, \quad (3.67)$$

$$\oint_{\partial V} \mathbf{J}_p \cdot d\mathbf{A} = -q \int_V \left(R + \frac{\partial p}{\partial t} \right) dv, \quad (3.68)$$

which must be satisfied for each Voronoi box of the tessellation. In the discrete form they can be rewritten as in (3.28)-(3.29)

$$\sum_{\forall j : \exists \text{ edge } \langle x_i x_j \rangle} \mathbf{J}_{n,ij} \cdot \mathbf{A}_{ij} = q \left(R_i + \frac{\partial n_i}{\partial t} \right) V_i, \quad (3.69)$$

$$\sum_{\forall j : \exists \text{ edge } \langle x_i x_j \rangle} \mathbf{J}_{p,ij} \cdot \mathbf{A}_{ij} = -q \left(R_i + \frac{\partial p_i}{\partial t} \right) V_i. \quad (3.70)$$

¹To avoid symbol clashes between grid points and the discrete hole concentration, in this section the grid points are labeled x_i .

The material constants, the electric field \mathbf{E} and the currents \mathbf{J}_n and \mathbf{J}_p are assumed to be constant along each edge and therefore,

$$\mathbf{J}_{n,ij} \cdot \mathbf{A}_{ij} = (-q \mu_n n \text{ grad } \varphi + q D_n \text{ grad } n) \Big|_{ij} \cdot \mathbf{A}_{ij}, \quad (3.71)$$

$$\mathbf{J}_{p,ij} \cdot \mathbf{A}_{ij} = (-q \mu_p p \text{ grad } \varphi - q D_p \text{ grad } p) \Big|_{ij} \cdot \mathbf{A}_{ij}, \quad (3.72)$$

where, because of the inner product, only the components of the values along $\mathbf{A}_{i,ij}$ or along the edge $\langle x_i x_j \rangle$ remain

$$J_{n,ij} = -q \mu_n n \partial_{ij} \varphi + q D_n \partial_{ij} n = q \mu_n (-n \partial_{ij} \varphi + V_{th} \partial_{ij} n), \quad (3.73)$$

$$J_{p,ij} = -q \mu_p p \partial_{ij} \varphi + q D_p \partial_{ij} p = q \mu_p (p \partial_{ij} \varphi + V_{th} \partial_{ij} p), \quad (3.74)$$

with

$$\mu_{n,p} V_{th} = D_{n,p} \quad \text{and} \quad V_{th} = \frac{k T}{q}. \quad (3.75)$$

The discretization of the transport equations is more crucial, since they are of convection-diffusion type [66]. Because the carrier concentrations change exponentially, a discretization scheme analogously to (3.13) would require an extremely dense mesh. Therefore, a better approximation for the electron and hole concentrations is chosen. As equations (3.73) and (3.74) are defined along the edge $\langle x_i x_j \rangle$, these equations are ordinary differential equations. With the introduction of a local coordinate ξ along the edge $\langle x_i x_j \rangle$, assumption of constant current density $J_{n,ij}$, constant mobility μ_n , and constant electric field

$$-\partial \varphi_{ij} = -\frac{d\varphi}{d\xi} \approx -\frac{\varphi_j - \varphi_i}{d_{ij}} \quad (3.76)$$

along the edge, a linear differential equation of first order is derived, which can be solved for $n(\xi)$

$$n(\xi) = K e^{\Delta_{ij} \frac{\xi}{d_{ij}}} + \frac{J_{n,ij} d_{ij}}{q \mu_n V_{th} \Delta_{ij}}, \quad (3.77)$$

with

$$\mathcal{B}(\Delta) = \frac{\Delta}{e^\Delta - 1} \quad \text{and} \quad \Delta_{ij} = \frac{\varphi_j - \varphi_i}{V_{th}}. \quad (3.78)$$

The general solution of the differential equation shows two unknowns K and $J_{n,ij}$ which have to be fixed by two boundary conditions (even if it is a differential equation of first order)

$$n(\xi_i) = n_i \quad \text{and} \quad n(\xi_j) = n_j. \quad (3.79)$$

The result for the hole current density can be evaluated analogously. Insertion of the boundary conditions and solving for $J_{n,ij}$ and $J_{p,ij}$ delivers

$$J_{n,ij} = \frac{q \mu_n V_{th}}{d_{ij}} (n_j \mathcal{B}(\Delta_{ij}) - n_i \mathcal{B}(-\Delta_{ij})), \quad (3.80)$$

$$J_{p,ij} = \frac{q \mu_p V_{th}}{d_{ij}} (p_j \mathcal{B}(-\Delta_{ij}) - p_i \mathcal{B}(\Delta_{ij})), \quad (3.81)$$

which results in the final discrete formulation of the semiconductor equations

$$\mathbf{S}_n \cdot \mathbf{n} = \mathbf{B}_n, \quad (3.82)$$

$$\mathbf{S}_p \cdot \mathbf{p} = \mathbf{B}_p. \quad (3.83)$$

Here

$$s_{n,ij} = \mu_n V_{th} \frac{A_{ij}}{d_{ij}} \mathcal{B}(\Delta_{ij}) \quad \forall i, j : \exists \text{ edge } \langle x_i x_j \rangle, \quad (3.84)$$

$$s_{n,ii} = - \sum_{\forall j : \exists \text{ edge } \langle x_i x_j \rangle} s_{n,ji} \quad \forall i, \quad (3.85)$$

$$b_{n,i} = (R_i + \frac{\partial}{\partial t} n_i) V_i \quad \forall i, \quad (3.86)$$

$$s_{p,ij} = \mu_p V_{th} \frac{A_{ij}}{d_{ij}} \mathcal{B}(-\Delta_{ij}) \quad \forall i, j : \exists \text{ edge } \langle x_i x_j \rangle \quad (3.87)$$

$$s_{p,ii} = - \sum_{\forall j : \exists \text{ edge } \langle x_i x_j \rangle} s_{p,ji} \quad \forall i, \quad (3.88)$$

$$b_{p,i} = -(R_i + \frac{\partial}{\partial t} p_i) V_i \quad \forall i. \quad (3.89)$$

This method is referred as Scharfetter-Gummel discretization [17][44][57]. The resulting expressions are linear equation systems in n_i or p_i where the boundary conditions can be set as in the previous sections. If calculating a non-stationary problem the time discretization can be performed as shown in Section 3.2.

The major difference to the field or diffusion equation is that the system matrices \mathbf{S}_n and \mathbf{S}_p are no longer symmetrical, because of $\mathcal{B}(\Delta_{ij}) \neq \mathcal{B}(\Delta_{ji})$. As the semiconductor equations for electrons (3.62) and holes (3.63) depend on the field equation (3.66) and this equation depends on the carrier concentrations itself, the whole differential equation system must be solved. While the dependency is nonlinear in φ_i , a recursive solution mechanism for solving this system is inevitable. As the coefficients of the matrices are influenced by $\mathcal{B}(\Delta_{ij})$, a small potential change will influence the carrier concentrations exponentially, which influences the potential distribution itself. A simple iterative approach, which evaluates each equation back-to-back and reinserts the updated values in the next iteration, does not deliver stable states. Newton or even specially damped Newton algorithms are required [5][67].

Chapter 4

An Alternative Approach for Diffusion Simulation

THE DOPANT ATOMS inside of semiconductor devices are usually introduced by ion implantation in several processing steps, with different masks, to restrict the implantation to certain regions, and with various implant energies and doses. Low energetic ion implantation results in very sharp profiles located only a few nanometers below the surfaces [24][64]. To spread the distributions and to activate the dopants, the device structures are exposed to high temperature annealing steps. During these high temperature processes, the dopants diffuse. As the final dopant distributions determine the locations of the pn-junctions and therefore determine the device behavior, an accurate simulation of these effects is important [18][52][69].

When the doping concentrations are low, the diffusion process follows the linear diffusion law with constant diffusivity. This diffusivity depends on the involved materials and the processing temperatures. For higher concentrations, the behavior differs from the linear model. The diffusivity depends on the values of the concentrations. In this case, other models must be applied [40].

Even with the linear diffusion equation, the simulation procedure is computationally expensive. The differential equation is time variant and therefore a transient simulation must be applied. The time steps of these simulations may be very small, at least at the beginning of the simulation when high diffusion gradients are expected. As time passes, the distributions smooth out and the time steps can also be enlarged. Concurrently, at regions with high doping gradients the grid density must be high at least in the directions of the gradients. And as the domains of high gradients move, the refined domains must also move. This refinement is often done by red-green refinement or directly by hierarchical splitting of grid edges.

The possibility of coarsening the refined areas is eased by the hierarchical methods. This is necessary as the high gradient locations move and temporarily produced high point densities can be removed. Under certain circumstances, another discretization method than Box Integration must be selected because the Delaunay criterion is not fulfilled. Box Integration with Delaunay grids gives the advantage that caused by the maximum principle negative concentrations can never appear [66]. With other discretization methods this non-physical behavior must be prevented by applying special grid criteria [19][48].

However, often quick and simple predictions of layout problems have to be chosen. Especially within three-dimensional simulations, the time horizon of a transient diffusion simulation may be exceeded, difficulties in the solving procedure, or plausibility flaws may occur, such as negative concentrations by applying Finite Element methods on badly fitted grids. Simplifications concerning the models and geometries have to be assumed. As the simulation domains for dopant diffusion, usually the silicon segment, can be simplified, the use of a Green's Function approach may be a possible solution to model the diffusion process, without negligence of a three-dimensional simulation. This approach has the advantage that it requires significantly less computational time, while by simulating the time variant differential equation a lot of time is wasted for computing the dopant distributions at several time steps. But in fact, only the final distribution at the end of the high temperature process is of interest. With the use of Green's Functions, it is possible to calculate the dopant distribution at any time step of interest by only one iteration. No transient simulation is necessary [10].

The requirements for the applicability of this method will be examined later on after the method has been explained and the final algorithm has been derived.

4.1 Diffusion and Green's Functions

The simple linear diffusion differential equation reads

$$\frac{\partial c(\mathbf{x}, t)}{\partial t} = D \nabla^2 c(\mathbf{x}, t), \quad (4.1)$$

where c denotes the time and space dependent concentration, with a constant diffusion coefficient D and the initial condition

$$c(\mathbf{x}, 0) = C_0(\mathbf{x}). \quad (4.2)$$

With the theory of Green's Functions the solution $c(\mathbf{x}, t)$ of this differential equation problem for one, two, and three dimensions is obtained as

$$c(\mathbf{x}, t) = \frac{1}{(4\pi Dt)^{n/2}} \int_{\mathcal{V}'} C_0(\mathbf{x}') e^{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{4Dt}} dx'^n, \quad (4.3)$$

where n is the dimensionality of the problem.

The discretization of this problem is performed by the following method: After the simulation domain is meshed and the Delaunay boxes are determined, the initial distribution $c(\mathbf{x}, 0)$ is discretized on the grid points by Dirac functions

$$C_0(\mathbf{x}) \approx \tilde{C}_0(\mathbf{x}) = \sum_i D_0(\mathbf{x}_i) \delta(\mathbf{x} - \mathbf{x}_i) \quad (4.4)$$

with

$$D_0(\mathbf{x}_i) = \int_{\mathcal{V}_i} C_0(\mathbf{x}) dv. \quad (4.5)$$

The preservation of the total implantation dose in the simulation domain

$$\begin{aligned} \int_{\mathcal{V}} \tilde{C}_0(\mathbf{x}) dv &= \int_{\mathcal{V}} \sum_i D_0(\mathbf{x}_i) \delta(\mathbf{x} - \mathbf{x}_i) dv = \sum_i \int_{\mathcal{V}} D_0(\mathbf{x}_i) \delta(\mathbf{x} - \mathbf{x}_i) dv \\ &= \sum_i D_0(\mathbf{x}_i) = \sum_i \int_{\mathcal{V}_i} C_0(\mathbf{x}) dv = \int_{\mathcal{V}} C_0(\mathbf{x}) dv = N \end{aligned} \quad (4.6)$$

is guaranteed by the discretization of the initial distribution.

As a consequence by applying (4.3), the approximated transient concentration distribution can be evaluated by

$$\begin{aligned} c(\mathbf{x}, t) \approx \tilde{c}(\mathbf{x}, t) &= \frac{1}{\sqrt{4\pi Dt}^3} \int_{\mathcal{V}'} \tilde{C}_0(\mathbf{x}') e^{-\frac{|\mathbf{x}-\mathbf{x}'|^2}{4Dt}} dv' \\ &= \frac{1}{\sqrt{4\pi Dt}^3} \sum_i D_0(\mathbf{x}'_i) e^{-\frac{|\mathbf{x}-\mathbf{x}'_i|^2}{4Dt}}. \end{aligned} \quad (4.7)$$

The total dose in the whole space \mathcal{R}

$$\begin{aligned}
\int_{\mathcal{R}} \tilde{c}(\mathbf{x}, t) dv &= \int_{\mathcal{R}} \frac{1}{\sqrt{4\pi Dt}^3} \sum_i D_0(\mathbf{x}'_i) e^{-\frac{|\mathbf{x}-\mathbf{x}'_i|^2}{4Dt}} dv \\
&= \frac{1}{\sqrt{4\pi Dt}^3} \sum_i \int_{\mathcal{R}} D_0(\mathbf{x}'_i) e^{-\frac{|\mathbf{x}-\mathbf{x}'_i|^2}{4Dt}} dv \\
&= \frac{1}{\sqrt{4\pi Dt}^3} \sum_i D_0(\mathbf{x}'_i) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{|\mathbf{x}-\mathbf{x}'_i|^2}{4Dt}} dx dy dz \\
&= \frac{1}{\sqrt{4\pi Dt}^3} \sum_i D_0(\mathbf{x}'_i) \sqrt{4\pi Dt}^3 = \sum_i D_0(\mathbf{x}'_i) = N
\end{aligned} \tag{4.8}$$

is preserved, too. The preservation of the implanted dose is a significant and necessary condition for the quality of the method. In total, the amount of implanted ions must not change during the diffusion process. Therefore, the discrete formulation must fulfill this preservation, too.

Since the implantation distribution is delivered on a grid, this grid can be used for calculating the entire dopant diffusion process. In this case, the initial dopant concentration $D_0(\mathbf{x}_i)$ is defined on the grid points p_i and therefore, the integration of (4.5) can be omitted and replaced by

$$D_0(\mathbf{x}_i) = C_0(\mathbf{x}_i) V_i, \tag{4.9}$$

which is the initial concentration at the grid point, weighted by the control volume of the Voronoi box.

4.2 Prerequisites for Applying the Green's Functions Method

The Green's Functions Method can only be applied if the diffusivity is constant in the whole simulation domain. This condition is usually satisfied, if only one segment has to be simulated. Normally different segments consist of different materials where the dopants also have different diffusivities. A further fundamental assumption of the chosen method is that the diffusion domain must be unbounded. This is usually not satisfied and every segment boundary will violate this condition. However, when the boundaries are far away from domains of interest, their influence is only marginal and the existence of these boundaries can be neglected. Neglecting the influence is

usually possible for the boundaries of the silicon wafer except the top boundary where the initial concentration was implanted. At this boundary, the distance of the implanted ions to the top of the wafer is not large enough. However, if the top of the wafer is flat and if it can be postulated that no dopant diffusion will occur through this boundary, the wafer can be mirrored above its top. With this trick, it can be achieved that no conduction of dopants will occur at the top and this boundary can be removed. Because the partial conduction coming from the original wafer and the conduction resulting from the mirrored wafer will cancel each other due to symmetry considerations. This implies, if possible simplifications make it feasible to place the dopant ions directly on the top of the wafer, that the entire dopant concentrations must be set twice than the original concentration to account for the dopants diffusing to the upper and lower half of the wafer.

Certainly, it must be guaranteed that the grid density of the initial distribution or the desired time where the final distribution will be calculated are convenient in terms of the relaxation processes between the grid points. If the final time is not far enough and the diffusion between the grid points is high, the discretization of the initial distribution must be denser. However, if the final time and the diffusion ranges become wider, the initial discretization of the distribution can be chosen cruder. Even by this diffusion of dopants, the influence of marginal areas must be accounted for in the calculation of the active areas. Which means that the initial dopant concentration of a larger domain, than where the final distribution has to be evaluated, has to be known and included in the diffusion process.

4.3 Advantages of the Green's Functions Method

The Green's Functions method gives a relatively simple and fast tool for calculating diffusion processes. Compared to direct simulation of the transient diffusion process, this method has the advantage that no iteration over small time steps is necessary until the final result is reached. Therefore, the frequent solution of a large equation system is not necessary. The final concentration can be directly written as given by (4.7). With this method the value of the concentration can be evaluated at any point in time *and* at any point in space.

Additional simplifications can be introduced, which speed up the calculation time extremely. Every partial initial Dirac like concentration sustains a broadening during time. The speed of broadening is proportional to $\sqrt{D/t}$ and implies that in the first few moments the distribution broadens fastest. If the initial distribution is steep delimited within a certain area the error of replacing this distribution by a layer of Dirac functions will be marginal.

For ion implantation with low implantation energies, the concentration maxima that are located close to the surface flow away due to diffusion. If the time interval is long and the broadening of the initial distribution is large relative to the initial thickness, the initial distribution can be simplified as an areal distribution. The Dirac like initial distributions are placed along surfaces inside the wafer, at little distance from the top of the wafer or certainly direct at the top. In this simplified case, only the top of the wafer has to be meshed. Utilizing with the adequate dose the final distribution can be achieved by only a single sweep over the surface of the wafer, which is much faster than iterating within a three-dimensional grid.

If the implantation doses are known and the diffusion ranges are large compared to the transversal implantation depths and expansions widths a relatively simple mask based tool for implantation and following diffusion based on this method can be developed. The dopant concentration can be evaluated and provided to the device simulation in every desired point.

As an additional feature of this method, the grids used for the initial distribution and the final profile can be designed independently and may be adapted to their individual requirements. The final concentration distribution is not mandatorily connected to a grid, as the concentration can be calculated in any place of interest, which may be an advantage. However, most of the tools applied later on require a grid. Even visualization tools often require a grid to display the dopant profiles or calculate iso-surfaces (as shown in the following example). As the demands on the grids for electrical simulation and visualization may differ seriously, an adapted dopant profile on the desired grid can be easily obtained by reapplying the diffusion simulation on the new grid.

4.4 Device Optimization by Three-Dimensional Diffusion Simulation

A major goal in the process development of high voltage processes is the design of devices with given breakdown voltages and low on-resistances. To reach this goal it is necessary to optimize the space charge regions of the device. Unfortunately these effects are three-dimensional and a device optimization needs the support of accurate three-dimensional process and device simulation.

The requirement for a low on-resistance (R_{on}) is to design a single device as small as possible. The reduction of space charge regions is limited by the dopant surface concentration of the wells which may result in impact ionization effects in case of

too high doping concentrations. On the other hand, lowering the doping concentrations is limited by the required punch-through voltage. To fulfill these conflicting criterions the doping concentrations must be optimized.

4.4.1 Simulated Structure

The investigated device is the tip of a drain finger of a high voltage PMOS transistor. For proper insulation of a p-type MOS transistor in a p-doped substrate, the whole transistor structure must be placed in a n-doped well located inside the substrate (NWell). The resulting pn-junction blocks the two different regions. Under the drain and source regions of the transistor, higher doping concentrations for electrical strength are required, whereas under the gate and near the surface lower doping concentrations are required to prevent impact ionization. These effects are combined in a deep NWell under the active drain and source regions and a shallow NWell in peripheral regions of the drain and source contact, called SDNTUB. The main drain and source regions of the transistor have to be constructed by a p-doped region under the contacts, located close to the surface of the wafer. This region is realized by a p-doped tub (PTUB). In combination with the SDNTUB a second pn-junction is built. Overall, a pnp-structure is formed by the PTUB, SDNTUB, and the substrate and under normal operation the PTUB/SDNTUB junction is biased in reverse direction.

To optimize the main characteristic of the PMOS transistor it is necessary to ensure that no three-dimensional effects dominate the device behavior. The optimal drain finger layout ensures that applying maximal V_{dd} causes no punch-through between PTUB and substrate and no avalanche breakdown occurs at the surface of the wells.

The anticipated netto doping distribution of the three-dimensional structure can be seen in Figure 4.1. For better understandability of the nomenclature of the layers, the pn-junctions and the names of the layers are plotted.

The complete device is embedded in the SNTUB so that there is no direct connection between PTUB and substrate. Only in the area of the PTUB, the DNTUB determines the distance between the pn- and the np-junctions. The PTUB/DNTUB mask layout is given in Figure 4.2, which shows that the DNTUB mask is enclosed by the PTUB mask. To enlarge the distance between the two junctions it is necessary to use a long DNTUB diffusion time so that the DNTUB dopants nearly diffuse spherically at the tip of the drain finger. This long DNTUB diffusion finally leads to a DNTUB formation which starts outside of the PTUB mask. The three-dimensional consideration is necessary because the spherical diffusion of the DNTUB dilutes the DNTUB concentration in the area of the finger tip and thus reduces the punch-through voltage of the PMOS device.

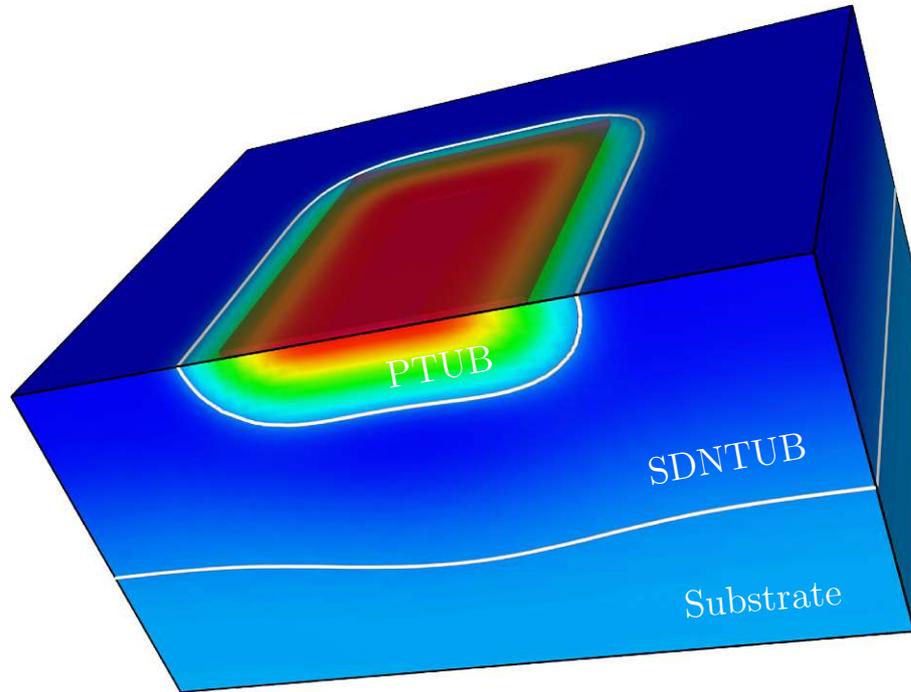


Figure 4.1: Netto doping of the semiconductor segment.

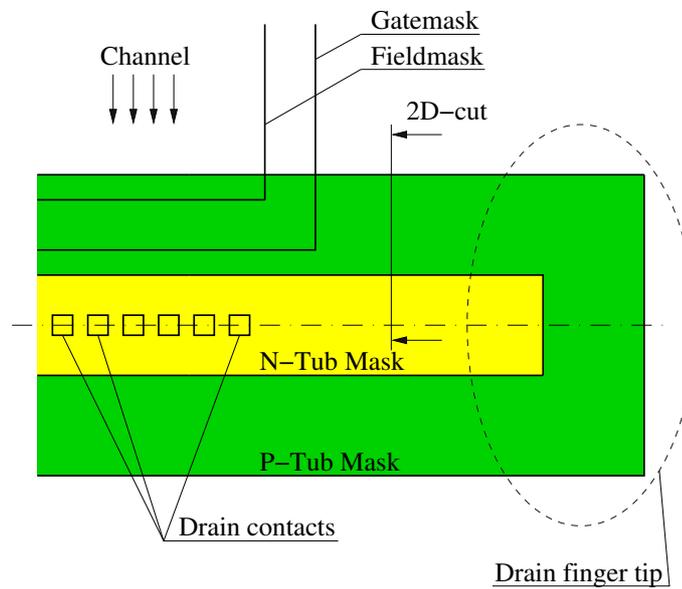
4.4.2 Comparison of Simulation Approaches

The conventional procedure is to simulate the whole ion implantation process first [24] and then the three-dimensional transient diffusion [48]. Thereby both steps require a particularly fine grid to achieve appropriate accuracy [15] and, therefore, the vast amount of memory and huge calculation times constitute prohibitive demands in practice.

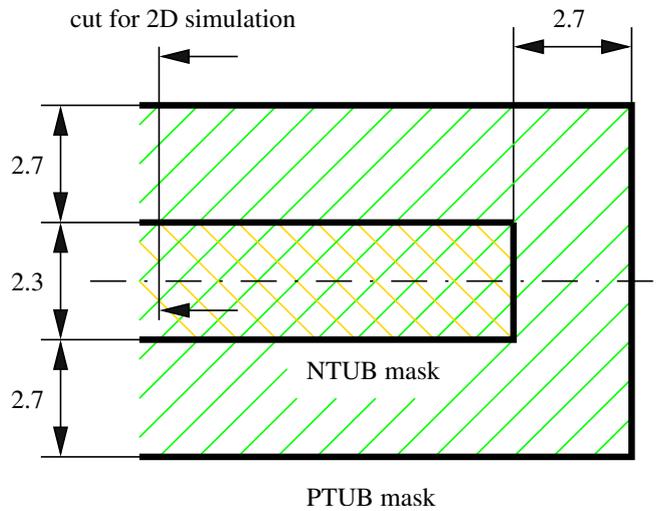
Because of the long diffusion ranges, the exact simulation of the ion implantation process can be neglected and the implanted ions were assumed only located at the top of the wafer. With this simplification the final diffusion profile inside the wafer can be calculated by the Green's Function method. A grid is only necessary at the surface of the wafer and the resulting doping distribution can be calculated at any point of interest.

4.4.3 Calibration and Evaluation

The diffusion model has to be calibrated by the two-dimensional simulation results which are available far away from the tip of the finger. The simulated drain current versus the drain-substrate-voltage of the two-dimensional cut is shown in Figure 4.3.



(a) Mask layout of the drain region of the transistor cut for 2D simulation



(b) Mask layout of the finger tip

-  mask for boron implantation
-  mask for phosphorus implantation

Figure 4.2: Well mask layout of the high voltage device.

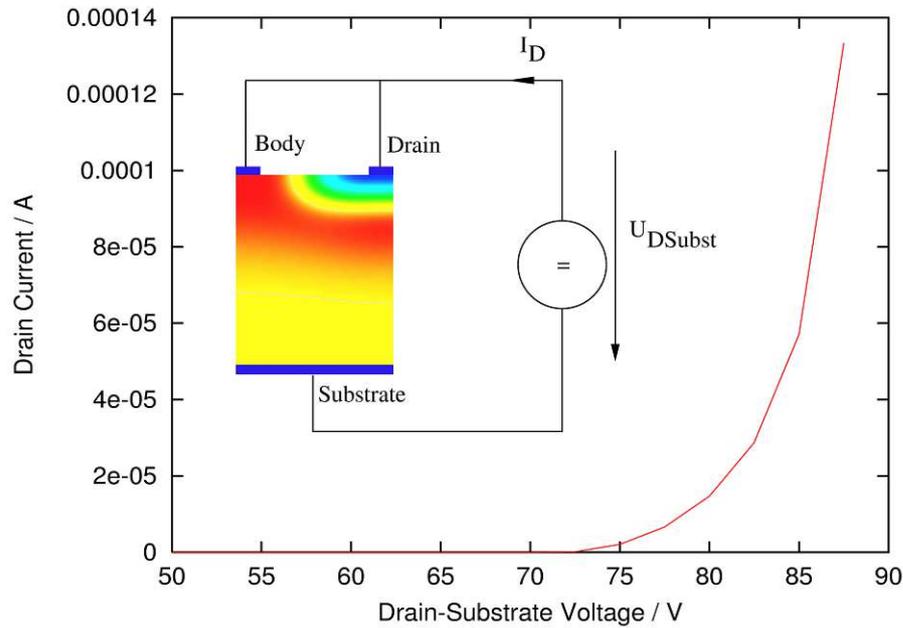


Figure 4.3: Two-dimensional simulation of the drain-current far away from the tip of the finger.

The assessment criterion of the new layout parameters is the fact that the dopant concentration of the PTUB/SDNTUB junction at the surface of the wells is the same for the two-dimensional case and the three-dimensional finger case. This ensures that the breakdown at the surface in the three-dimensional structure takes place in the same voltage range as compared with the two-dimensional structure.

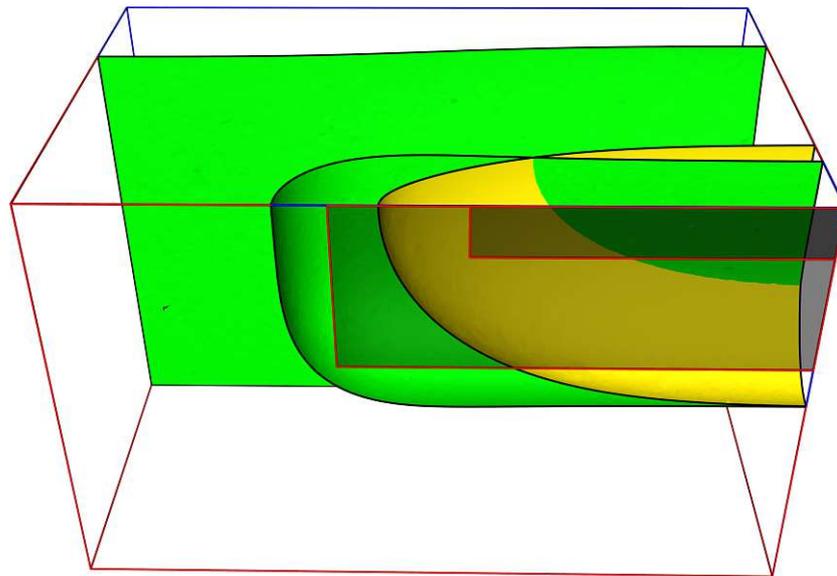
4.4.4 Results

The simulation results show that the spherical out-diffusion of the DNTUB is larger than expected due to the large NTUB depth. This depth is about 7.5 micron in the two-dimensional simulation. The spherical diffusion length is also of the same size from the top of the DNTUB finger to the direction of the two-dimensional case. In fact, the two-dimensional situation is given when the DNTUB mask is enlarged by about 7 micron as compared to Figure 4.2. This means that the DNTUB mask can even exceed the PTUB mask. However an enlargement of 7 microns would cause impact ionization near the top of the PTUB surface. So the limiting case of the DNTUB enlargement is the dopant concentration of the two-dimensional simulation at the surface of the junction. This critical concentration is given when the DNTUB mask is shifted by 2 microns towards the PTUB mask. The different surfaces of the pn-junctions are shown in Figure 4.4(a) for the original mask setting and Figure 4.4(b) for the improved finger design.

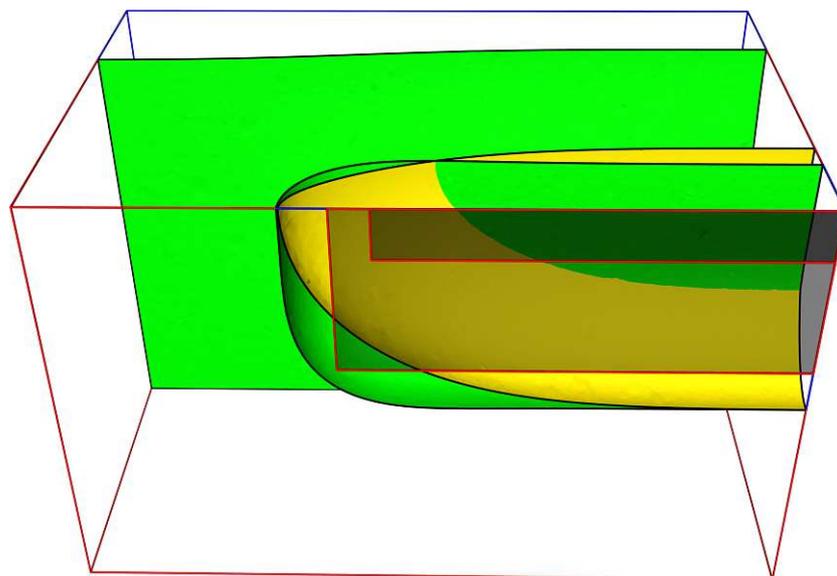
Another interesting effect is that the punch-through in the three-dimensional case does not occur directly under the symmetry line of the finger (see Figure 4.5(a)). The explanation is that the DNTUB dopants diffuse spherically while the PTUB dopants diffuse cylindrically. Therefore, the punch current has its maximum density near the edge of the PTUB mask. For comparison the space charge region of the enlarged finger is shown in Figure 4.5(b).

The simulation results are validated by a set of test devices. Figure 4.6 shows the punch current dependency of the finger elongation starting with the initial layout shown in Figure 4.2.

With these careful considerations the device has been optimized to fulfill the electrical requirements, particularly with respect to punch-through between the junctions and breakdown by impact ionization. Without the outlined simulation methodology it would not have been possible to fully optimize the structure.

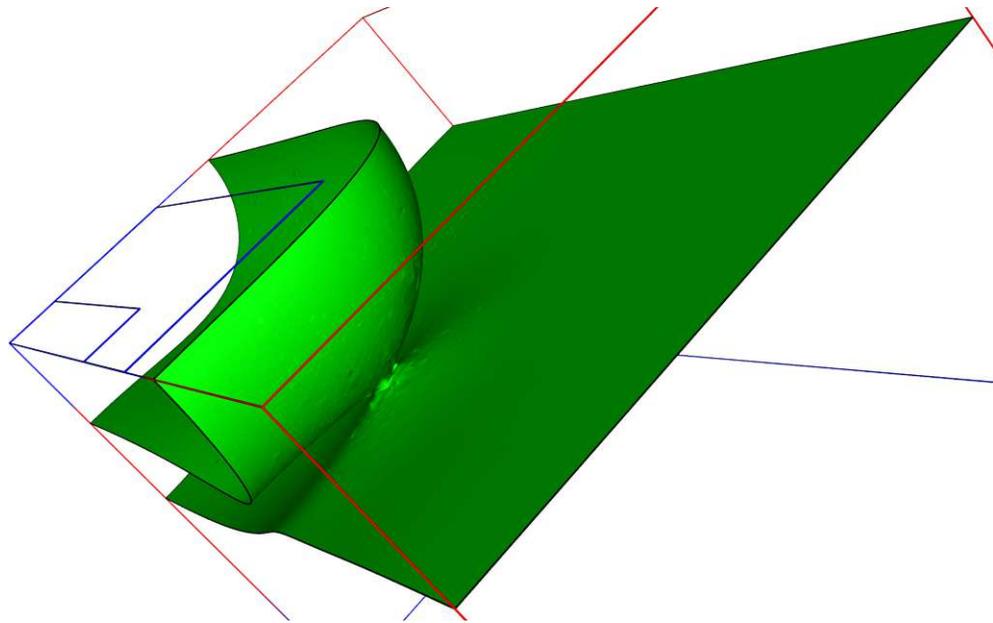


(a) The original finger layout.

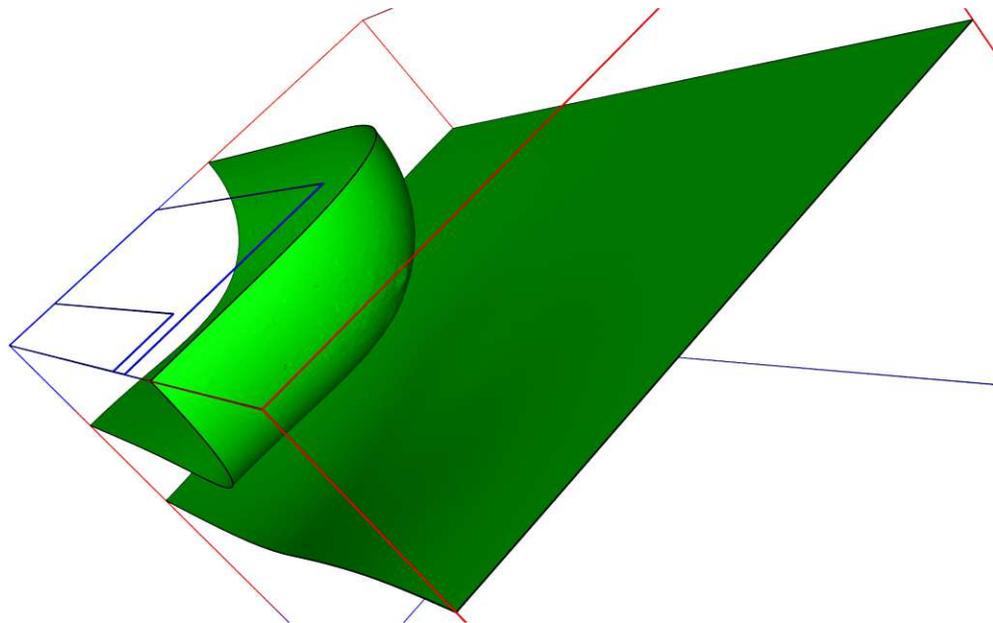
(b) The $2\ \mu\text{m}$ enlarged finger layout.

- pn-junction, where the upper one also represents a boron iso-surface
- phosphorus iso-surface

Figure 4.4: Relevant iso-surfaces of the phosphorus doping and the pn-junctions of the finger. The relevant boron surface is approximately represented by the pn-junction which is located under the surface of the wafer.



(a) The original finger layout.



(b) The $2\ \mu\text{m}$ enlarged finger layout.

Figure 4.5: Surfaces surrounding the space charge regions between both pn-junctions.

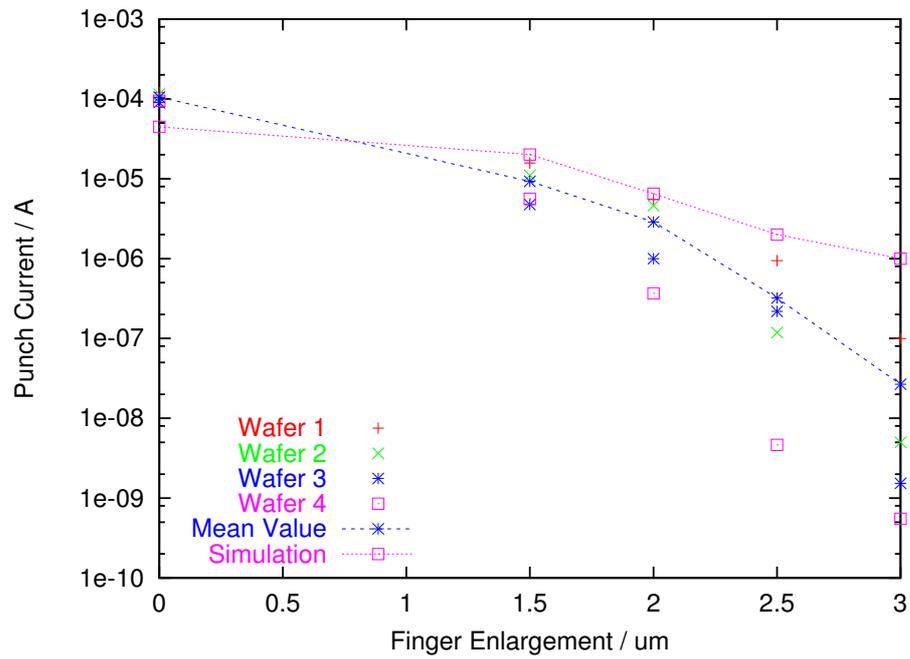


Figure 4.6: Simulation of the original finger structure compared to the measurement of four test wafers.

Chapter 5

Grid Generation for Device Simulation

FOR DEVICE SIMULATIONS, the demands on grid generators are very high. Coming along with the growing complexity and miniaturization of the devices two-dimensional models are often not sufficient. These models cannot describe parasitic effects near corners of the structures. Due to the shrinking aspect ratios of the devices, the limited device extension along the third coordinate direction cannot be neglected. However, these parasitic effects are able to change the device characteristics dramatically. Neglecting the third dimension can even cause an invalid simulation result. Therefore, the models must be extended to three dimensions and also three-dimensional simulation grids have to be used.

Simple electric field calculations or their assigned capacitance, inductance and resistance calculations of interconnect structures show a relatively well-tempered behavior. The underlying differential equations are often linear, such as the Laplace equation. Additional nonlinearities such as a temperature-dependent resistance can be handled by post-iterations: The simulation is started with an initial value of the temperature dependent resistance. After evaluation of the current densities, the heating of the device regions is calculated, which changes the specific resistance. These new values will be inserted in the next field evaluation. This procedure is continued until the resistance, current density, and temperature do not change significantly. The prerequisites on the meshes are not very high, already relatively crude grids will deliver good results. Only the convergence of the solvers must be guaranteed by the mesh type in combination with the discretization method. In contrast, within electric calculations of semiconductor devices the demands on the grids grow rapidly, because semiconductor equations are highly nonlinear and show a strong coupling; this basically due to the exponential dependence of the carrier concentration on the potential.

5.1 Grid Requirements for Device Simulation

As an example, consider the long channel nMOSFET shown in Figure 5.1, operating in inversion mode. We assume no recombination and stationary conditions. No voltage is connected between source and drain, and therefore the device is current-free. To compensate the electric field in the channel arising from the applied gate voltage, a carrier displacement will occur in that area.

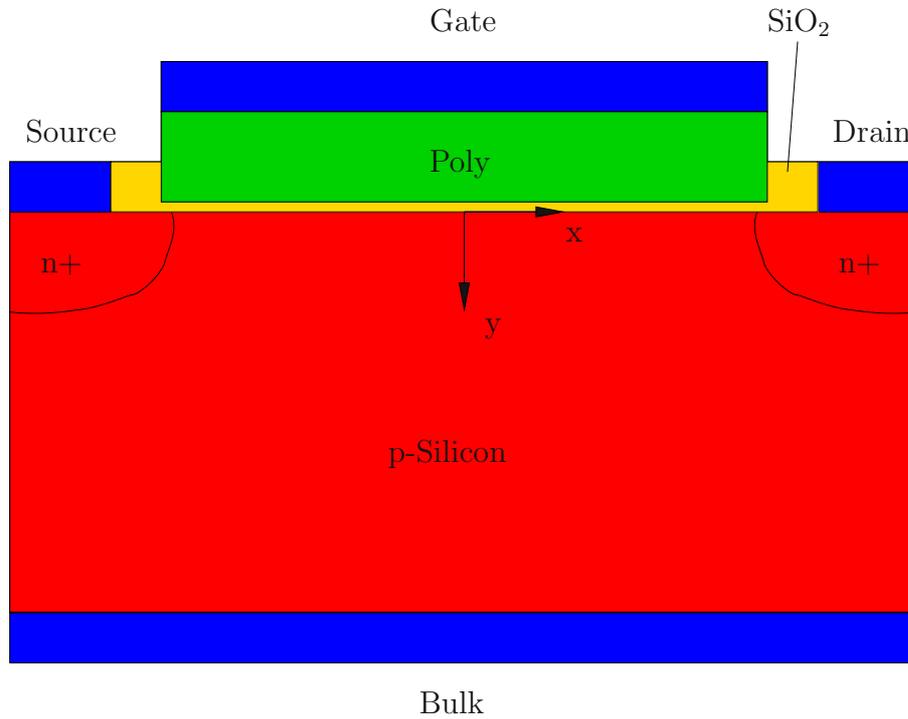


Figure 5.1: Structure of a two-dimensional MOS transistor.

The set of equations are the Poisson equation and the semiconductor equations

$$\operatorname{div} (\varepsilon \operatorname{grad} \varphi) = q (n - p - (N_A - N_D)), \quad (5.1)$$

$$\operatorname{div} \mathbf{J}_n - q \frac{\partial n}{\partial t} = q R, \quad (5.2)$$

$$\operatorname{div} \mathbf{J}_p + q \frac{\partial p}{\partial t} = -q R, \quad (5.3)$$

$$\mathbf{J}_n = -q \mu_n n \operatorname{grad} \varphi + q D_n \operatorname{grad} n, \quad (5.4)$$

$$\mathbf{J}_p = -q \mu_p p \operatorname{grad} \varphi - q D_p \operatorname{grad} p. \quad (5.5)$$

In inversion mode, the holes under the gate are displaced and the following relation holds

$$n \gg N_A - N_D \gg p. \quad (5.6)$$

The holes become negligible and only the drift-diffusion current relation for the electrons has to be considered. Without recombination and in stationary mode the required equations simplify to ($\mu_n V_{th} = D_n$)

$$\text{div} (\varepsilon \text{ grad } \varphi) = q n, \quad (5.7)$$

$$\text{div } \mathbf{J}_n = 0, \quad (5.8)$$

$$\mathbf{J}_n = -q \mu_n n \text{ grad } \varphi + q \mu_n V_{th} \text{ grad } n. \quad (5.9)$$

It is obvious that carrier displacement occurs only along the y-axis, in the direction of the electric field and therefore only the y-component of equations (5.8) and (5.9) is crucial

$$J_{n,y} = q \mu_n n E_y + q \mu_n V_{th} \frac{\partial n}{\partial y} = 0. \quad (5.10)$$

This equation system can be solved by differentiating equation (5.7) (only the y-component of the electric field is present). We assume constant permittivity and mobility, and obtain

$$\varepsilon \frac{\partial^2 E_y}{\partial y^2} = -q \frac{\partial n}{\partial y}. \quad (5.11)$$

Inserting (5.7) and (5.11) into (5.10) gives

$$-\varepsilon \frac{\partial E_y}{\partial y} E_y - V_{th} \varepsilon \frac{\partial^2 E_y}{\partial y^2} = 0 \quad (5.12)$$

which is an ordinary differential equation in E_y ($E'_y = \frac{\partial}{\partial y} E_y$)

$$E''_y + \frac{1}{V_{th}} E_y E'_y = 0. \quad (5.13)$$

This equation shows an unphysical solution $E_y \propto \tanh(y+d)$, which leads from the additional differentiation (5.11), and the solution $E_y = A/(y+d)$, where both the electric field and the carrier concentration decay to zero. With the evaluation of A the final solution is

$$E_y = 2 V_{th} \frac{1}{y+d}. \quad (5.14)$$

The parameter d is affected by the thickness of the field-oxide, the applied voltage, and the dopant concentrations. By insertion into (5.7), the distribution of the charge carriers follows as

$$n = \frac{2 V_{\text{th}} \varepsilon}{q} \frac{1}{(y + d)^2}. \quad (5.15)$$

Here it can be seen that the electric field and even more, the carrier concentration changes rapidly along the y -axis, whereas along the x -axis the values remain unchanged. In Figure 5.2 the resulting carrier concentration of such a device is shown. Here the silicon segment has a constant donor background doping of 10^{15} cm^{-3} and the highly doped areas under the source and drain contacts have a constant acceptor doping of 10^{20} cm^{-3} . The oxide thickness under the gate contact is 20 nm. By applying a drain-source voltage and without a gate-source voltage, one of the two source/silicon or drain/silicon pn-junctions are in reverse direction and the device is blocked. With a gate bias of 10 V and all other contacts grounded, the carrier concentrations raise at the gate regions. The device is in inversion and the carrier concentrations under the gate contact are higher than the concentrations in the source and drain regions. The pn-junctions are not reverse biased any longer.

With introducing a voltage between source and drain, a current will arise. As the free charge carriers are responsible for the current density, a relatively high current density change along the y -axis, combined with nearly constant current density along the x -axis will follow. In Figure 5.3, the current density, with a voltage of 1 V between source and drain, is shown. Along the channel, the current density is almost constant. This simple device with planar silicon surfaces has been simulated based on an ortho grid. The comparison of different grid approaches, with a dense grid with a grid spacing of 0.01 nm along the y -axis and a coarse grid with minimum grid distance of 20 nm under the gate, and the result of the analytical solution is shown in Figure 5.4. The analytical solution loses its validity since relation (5.6) is violated. Comparing the simulations, an underestimation of the current density under the gate contact of about 1 : 30 can be detected, which may decide about an breakdown of the device in critical cases.

5.2 Adapted Grid Generation

A high grid density in directions where simulation values change rapidly must be guaranteed to achieve an accurate numerical solution. As in the above example, the resolution of the $1/(y + d)^2$ dependency of the current density must be guaranteed and therefore the grid density under the gate oxide must be a fraction of the thickness

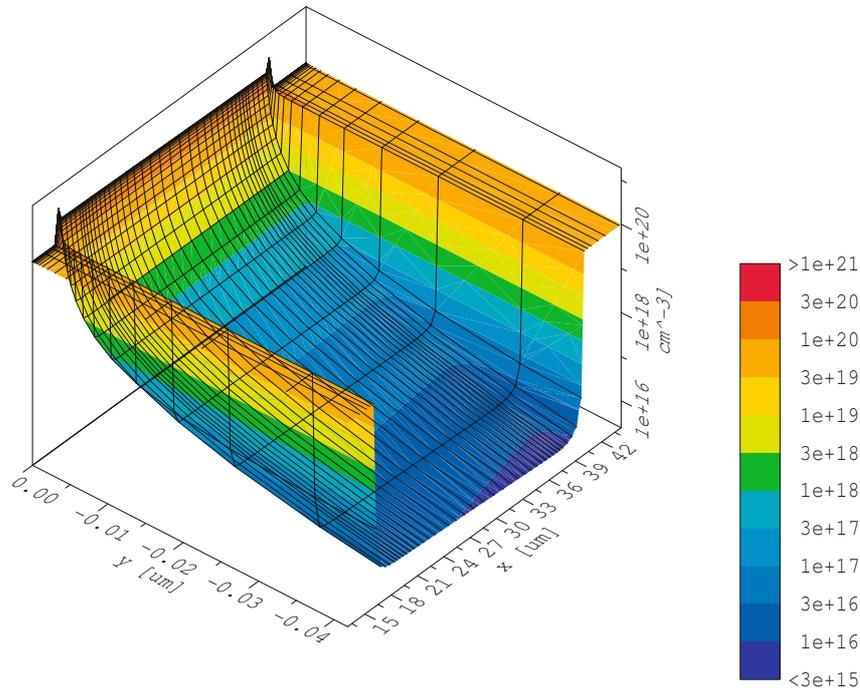


Figure 5.2: The carrier concentration of the two-dimensional device simulation with a gate-source and gate-substrate voltage of 10 V.

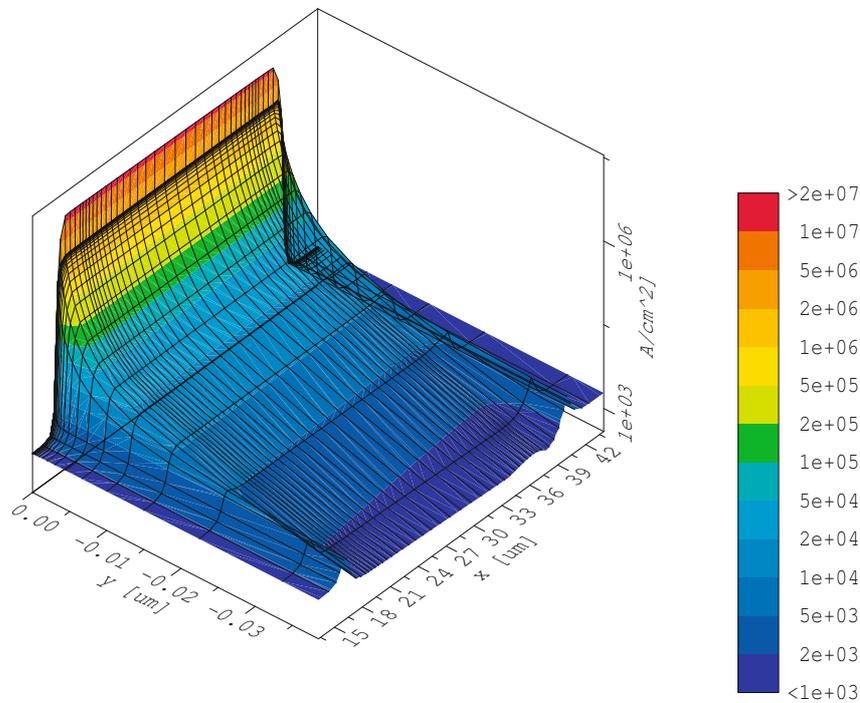


Figure 5.3: The current density of the two-dimensional device simulation at $V_{DS} = 1$ V.

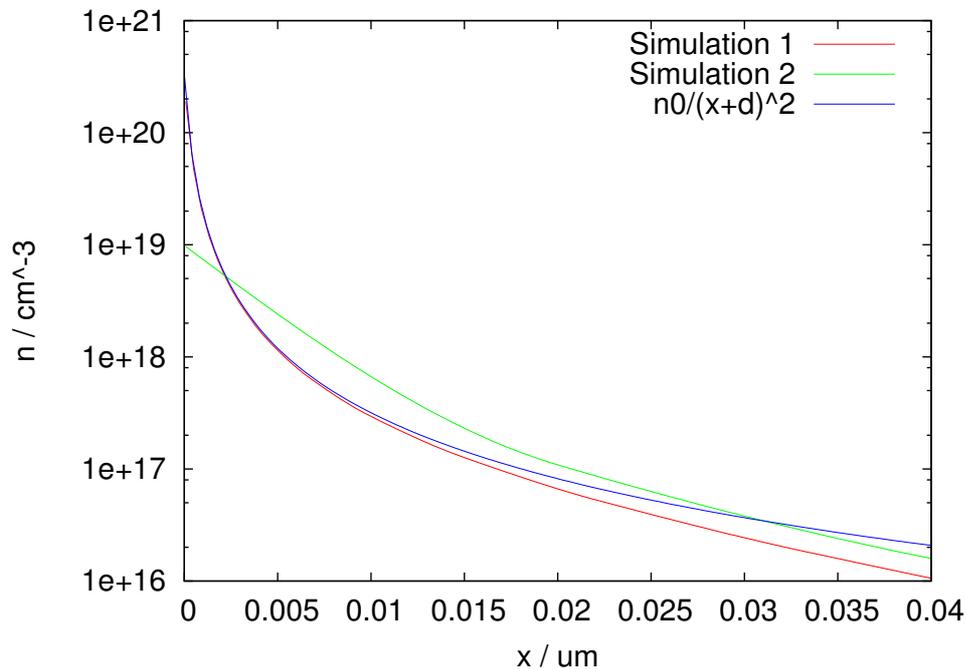


Figure 5.4: Comparison of the simulation approaches. The first two-dimensional simulation is performed on a dense grid with a grid spacing of 0.01 nm under the gate. The coarse grid has a spacing of 20 nm under the gate. These simulations are compared with the derived analytical solution (5.15), $n_0 = 3.38 \times 10^5 \text{ cm}^{-1}$, $d = 0.32 \text{ nm}$.

of the oxide, whereas along the channel the grid density can be much cruder. An aspect ratio of about 1 : 100 is not rare.

In general, increasing the number of points increases the accuracy of the solution (neglecting numerical errors). However, especially within three-dimensional simulations an isotropic increase of grid points will cause intolerable memory consumption and calculation times. Therefore, a compromise of these effects must be a main goal.

In areas where simulated values are strongly affected (also affected with high non-linearities), the point density must be high and so the tetrahedrons have to be split into smaller ones. A two-dimensional grid example, which shows the refinement procedure, can be seen in Figure 5.5. The right grid is refined with a global grid density criterion turned on. An example for such a global criterion is a maximal area constrain for all grid elements. As seen in Figure 5.5(b), the resulting triangles become nearly equilateral. Applying the same grid refinement method for three dimensions, the expected tetrahedrons will become nearly equilateral, too.

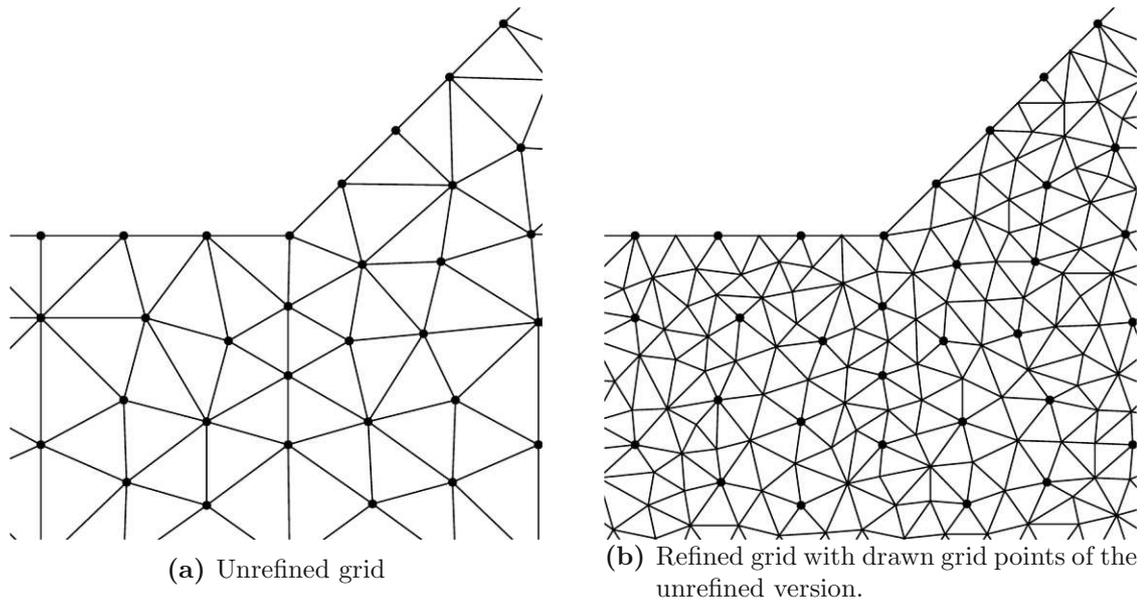


Figure 5.5: Refinement of a two-dimensional triangular grid.

As the required grid density is directionally dependent (anisotropic) — like in the transistor example — the density in one direction is sufficient, but in another, it can be too high. In this direction the material parameters and unknowns do not change so rapidly and the density of discretization along this direction does not need to be so fine. With global refinement we have an undesirable high amount of grid points where the simulation must be performed and the memory and time consumption will be unnecessarily high. Therefore, investigations of producing grids of desired structure are necessary.

Due to the discretization of the differential equations and especially of the boundary conditions, orthogonal and boundary conform meshes are desired. Matrix entries and the discretization errors caused by Neumann conditions are decimated, which are achieved for instance between the gate oxide and the semiconductor segment. In general, orthogonal grid components in and across main directions of the current densities also decimate the errors caused by the Box Integration method inside the structures.

Therefore, a new method for three-dimensional grid generation was developed in the scope of this thesis. The methodologies of this method are described in the following sections.

5.3 Potential-Based Grid Generation

In the particular case of MOS structures, a point distribution with low point density along the channel and high point density towards the bulk seems feasible. Nevertheless, to fulfill the Delaunay criterion, the grid generator sometimes has to insert additional grid points. Therefore, a method for point insertion which does not violate the Delaunay criterion must be found.

For ortho-product grids, the aspect ratio of the cuboids can be varied arbitrarily. As mentioned before, an ortho grid can be transformed easily to a tetrahedral grid by splitting each cuboid into five or six tetrahedrons. The Delaunay criterion will never be violated, since all resulting tetrahedrons have the same outer-sphere and neighboring grid points can never lie within. A pleasant side effect of this method is that the coupling areas of the tetrahedrons inside the original parallelepiped will become zero, the resulting equation system will have fewer off-diagonal entries and the assembled equation system will be the same as the one obtained from the original ortho grid.

However, this approach is difficult when cuboids have to be fitted along geometry lines that are not in the same direction as the edges of the cuboids (refer Section 2.1). For maintaining the accuracy, the number of cuboids and, therefore, also the number of grid points will increase rapidly. In this case, the general use of tetrahedral grids is required. There are fewer geometry related limitations and a Delaunay grid can be found in most cases. But even an enhancement of the grid density will cause the tetrahedrons to become nearly equilateral (see Figure 5.5(b)) and a directional dependent grid density cannot be tuned arbitrarily. In the method developed here, the advantages of ortho and tetrahedral grids are combined.

5.3.1 Two-Dimensional Behavior

The basic idea of the newly developed method can be explained by considering a nearly cuboidal capacitor with two electrodes. The electric field outside the capacitor is neglected by assuming a high-k dielectric with a permittivity ε much higher than the vacuum permittivity. A sketch of such a capacitor is shown in Figure 5.6.

After applying a voltage between the electrodes, the equipotential lines and the field lines can be evaluated. The field lines are orthogonal to the potential lines. Additionally, either the equipotential or field lines are boundary-conform or orthogonal to the boundaries of the capacitor. If the spacing of selected potential and field lines of the capacitor is sufficiently small, two potential and two field lines lying next

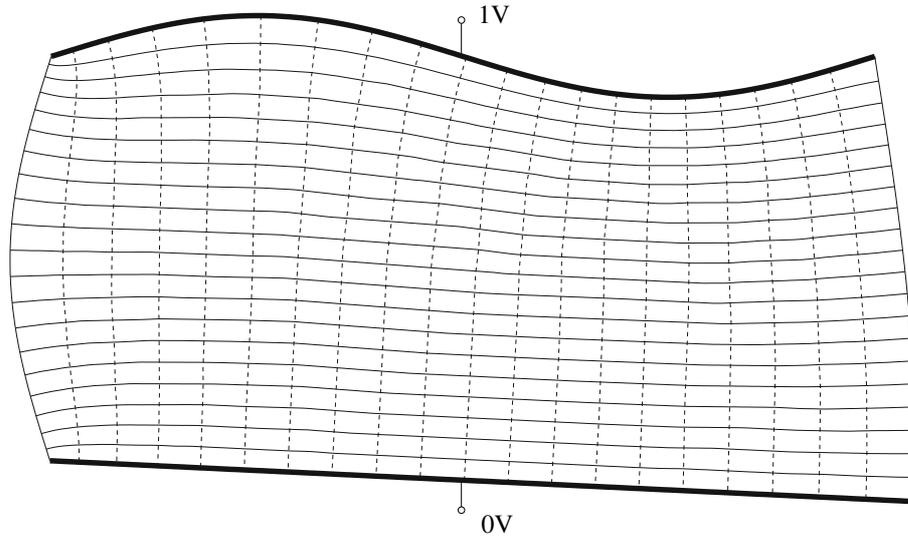


Figure 5.6: Two-dimensional capacitor with the electrodes and drawn field and equipotential lines.

to each other approximately form rectangles, as shown in the figure. The spacing between the selected field and potential lines can be varied independently. To create a grid suitable for device simulation, these rectangles must be split into triangles by a mesh generator.

This method can be used for creating a set of grid points in a given domain. Four points at the boundary of this domain are placed at the corners of the capacitor. Between the corners, the upper and lower electrodes are located. After the evaluation of the field distribution, the grid points can be set at arbitrary intersections of the equipotential lines and field lines.

The field lines are not evaluated directly. Rather, they are evaluated by a duality, which can be applied if the following prerequisites are satisfied:

- Constant permittivity ε inside the capacitor.
- Four corner points are placed on the boundary.
- Between each two corners the electrodes are placed where the voltage is connected (Dirichlet boundary conditions).
- The remaining opposite boundaries show a vanishing normal component of the electric field (homogenous Neumann conditions).

After applying a voltage between the electrodes, the electric potential distribution ($\mathbf{E} = -\text{grad } \varphi$) inside the capacitor can be calculated and the equipotential lines ($\varphi = \text{const}$) can be evaluated.

The field lines are evaluated via a duality in the specified way:

- The dual capacitor, with congruent shape, replaces the original capacitor.
- Constant permittivity ε inside the capacitor.
- The same four corner points on the boundary.
- The original boundaries with vanishing normal component of the field are replaced by electrodes — with imposed voltage.
- The original electrodes are replaced by boundaries with vanishing normal component of the electric field.
- The field lines of the original capacitor are the equipotential lines of the dual capacitor.

The dual electric field ($\mathbf{F} = -\text{grad } \psi$) and its associated equipotential lines ($\psi = \text{const}$) are calculated.

Now each point (x, y) inside the capacitor can be represented by its potential representation (φ, ψ) .

5.3.2 Extension to Three Dimensions

In the two-dimensional case, there are four electrodes at the boundary of the capacitor — two opposite electrodes of the original capacitor and the remaining two opposite boundaries as the electrodes for the dual capacitor. The extension of this method to three dimensions is as:

Prerequisites:

- Constant permittivity ε inside the capacitor.
- There are exactly six boundary groups — “sides”,
- Which must not overlap and
- In general, all six “sides” must cover the whole surface.
- Two opposite “sides” act as electrodes,

- The other four “sides” act as Neumann boundaries.

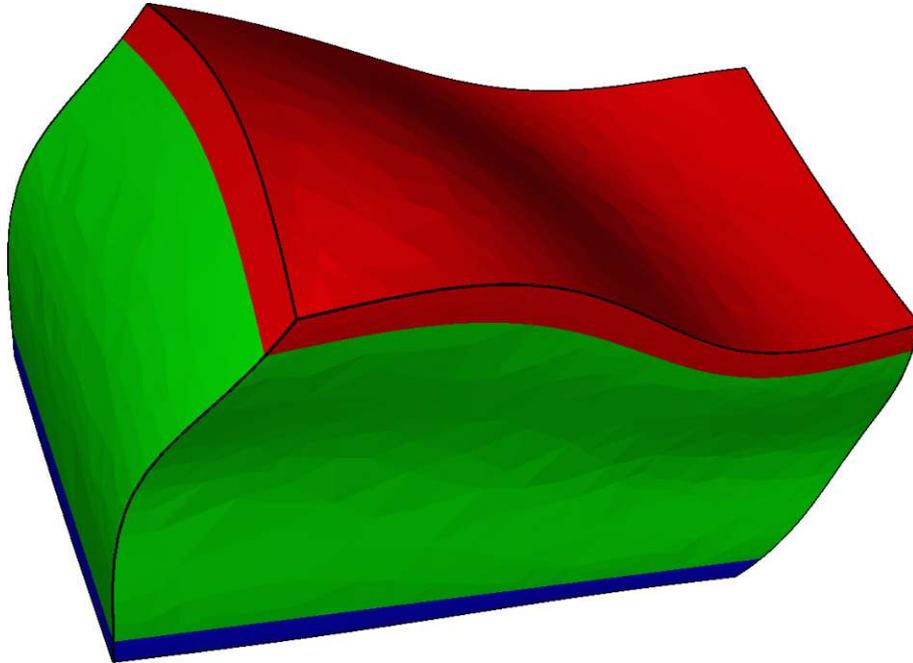


Figure 5.7: Capacitor with electrodes at top and bottom.

Such a capacitor can be seen in Figure 5.7. The original capacitor delivers the field distribution $\mathbf{E} = -\text{grad } \varphi$ with the equipotential surfaces $\varphi = \text{const.}$

Methodology of the Capacitor Model

The electric field \mathbf{F} and its electric potential ψ , and as a consequence of this three-dimensional expansion, the electric field \mathbf{G} and the potential ϑ , are derived as the resulting field inside a capacitor of the following shape:

- The new capacitor region is the same as the original one.
- Constant permittivity.
- Two opposing Neumann surfaces are replaced by electrodes where a voltage is connected.
- The former electrodes are replaced by homogenous Neumann boundaries.

- The remaining two surfaces are left as homogenous Neumann surfaces.

With this new placement of the electrodes, the electric potential $\mathbf{F} = -\text{grad } \psi$ can be derived. By an additional replacement of the remaining two “sides” by electrodes and all others as insulator surfaces, the third potential $\mathbf{G} = -\text{grad } \vartheta$ can be calculated.

5.3.3 Calculating the Equipotential Surfaces of the Capacitor

Inside the capacitor, the Laplace equation has to be solved (as shown in Section 3.1). The boundary conditions are

$$\varphi(x, y, z) = \begin{cases} \Phi_0 & \text{for } (x, y, z) \in \partial V_1 \\ \Phi_1 & \text{for } (x, y, z) \in \partial V_2, \end{cases} \quad (5.16)$$

for the electrodes where the voltage is connected and vanishing normal component

$$\partial_n \varphi(x, y, z) = \begin{cases} 0 & \text{for } (x, y, z) \in \partial V_3 \text{ or } (x, y, z) \in \partial V_4 \\ 0 & \text{for } (x, y, z) \in \partial V_5 \text{ or } (x, y, z) \in \partial V_6, \end{cases} \quad (5.17)$$

on the other two opposing boundary groups. With this set of boundary conditions, the equipotential faces of the electric field can be evaluated.

It can be seen that the elements defined by two appropriate equipotential lines of φ , ψ , and ϑ are spanning the desired cuboids for the tetrahedrization. The different electrode placements with their assigned field distribution are illustrated in Figure 5.8 (original distribution), 5.9 (first electrode-replacement), and 5.10 (second electrode-replacement). Resulting equipotential surfaces of all three field calculations are shown in Figure 5.11. The full set of differential equations (assuming constant permittivity) with their assigned boundary conditions can be seen in Table 5.1.

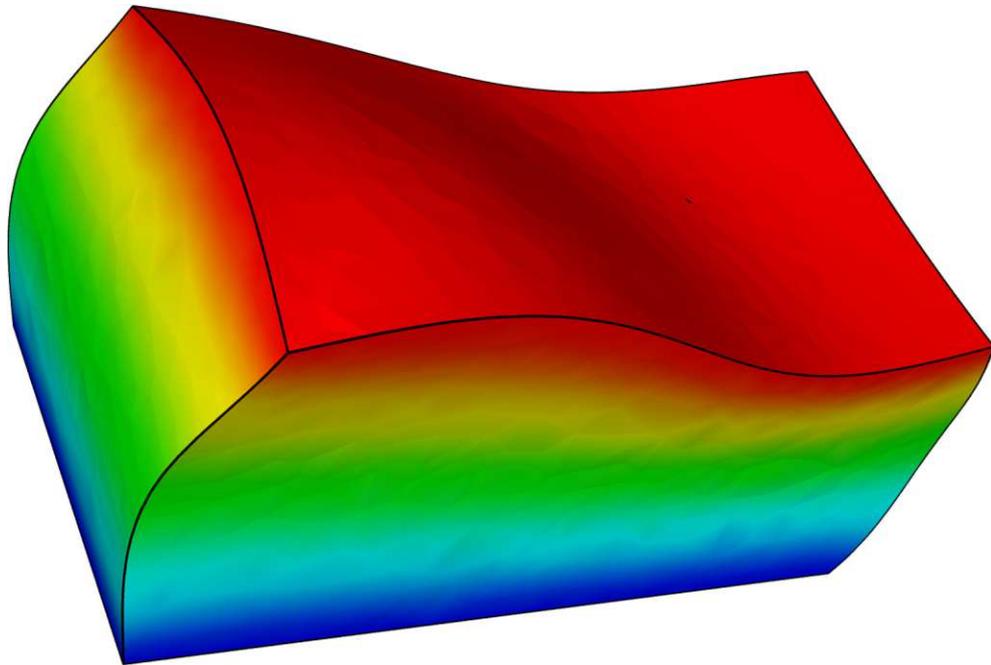


Figure 5.8: Cuboidal capacitor, original electrodes, potential distribution.

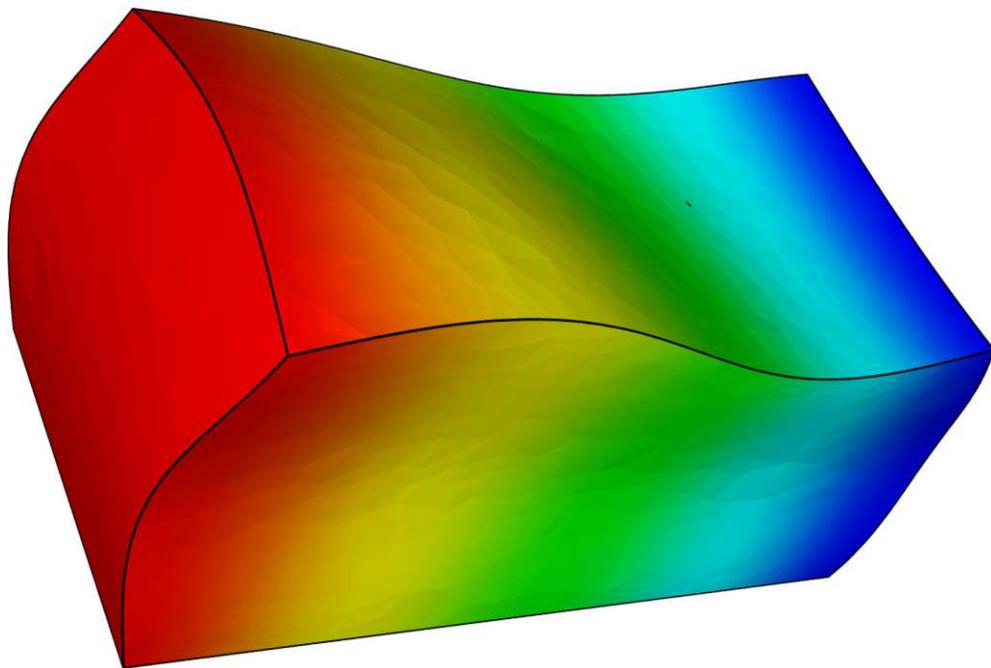


Figure 5.9: Cuboidal capacitor, first electrode-replacement, potential distribution.

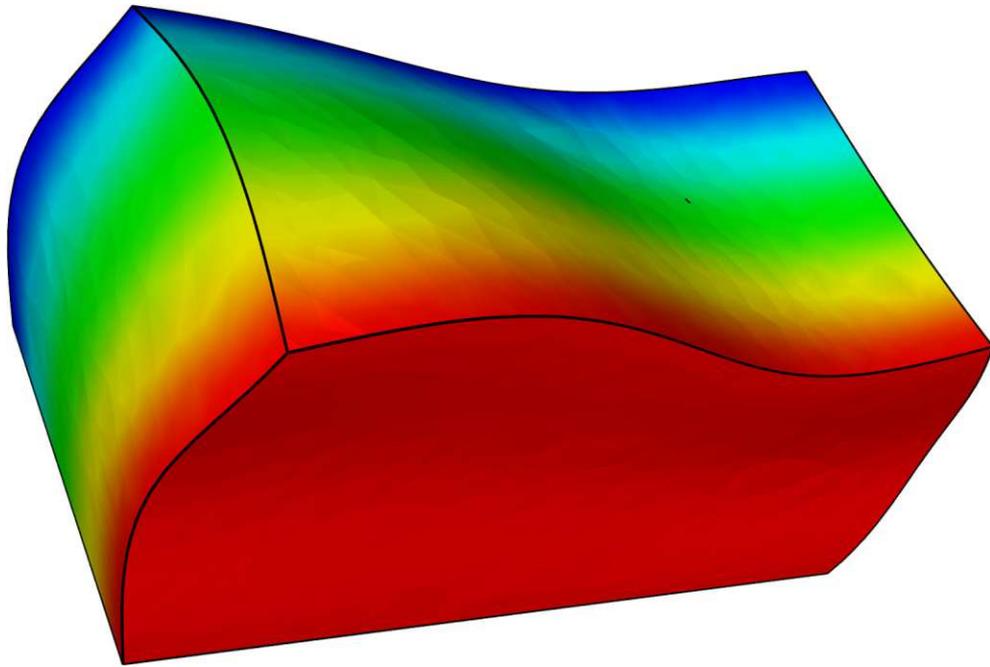


Figure 5.10: Cuboidal capacitor, second electrode-replacement, potential distribution.

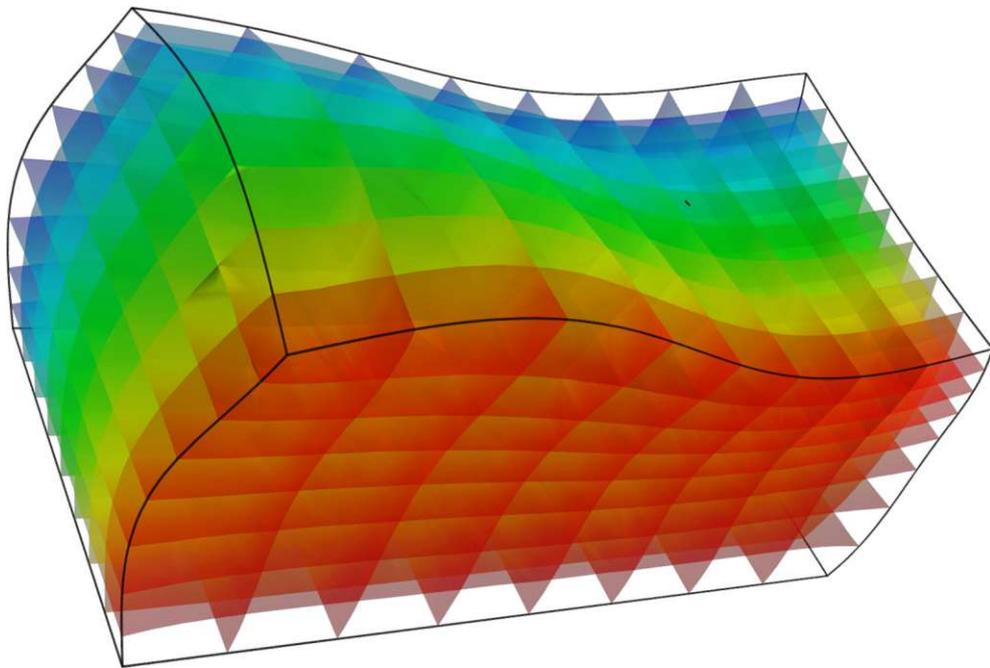


Figure 5.11: Cuboidal capacitor, with the equipotential surfaces of the three electric fields that are spanning the cuboids.

$\operatorname{div} \mathbf{E} = 0$	$\operatorname{div} \mathbf{F} = 0$	$\operatorname{div} \mathbf{G} = 0$	
$\operatorname{curl} \mathbf{E} = \mathbf{0}$	$\operatorname{curl} \mathbf{F} = \mathbf{0}$	$\operatorname{curl} \mathbf{G} = \mathbf{0}$	
$\mathbf{E} = -\operatorname{grad} \varphi$	$\mathbf{F} = -\operatorname{grad} \psi$	$\mathbf{G} = -\operatorname{grad} \vartheta$	
$\varphi = \Phi_1$	$\partial_n \psi = 0$	$\partial_n \vartheta = 0$: for $(x, y, z) \in \partial V_1$
$\varphi = \Phi_2$	$\partial_n \psi = 0$	$\partial_n \vartheta = 0$: for $(x, y, z) \in \partial V_2$
$\partial_n \varphi = 0$	$\psi = \Psi_3$	$\partial_n \vartheta = 0$: for $(x, y, z) \in \partial V_3$
$\partial_n \varphi = 0$	$\psi = \Psi_4$	$\partial_n \vartheta = 0$: for $(x, y, z) \in \partial V_4$
$\partial_n \varphi = 0$	$\partial_n \psi = 0$	$\vartheta = \Theta_5$: for $(x, y, z) \in \partial V_5$
$\partial_n \varphi = 0$	$\partial_n \psi = 0$	$\vartheta = \Theta_6$: for $(x, y, z) \in \partial V_6$

Table 5.1: Differential equations and boundary conditions of the capacitor and its dual capacitors.

5.3.4 Topological Cuboids

In general, a realistic devices are more complex than such capacitors and therefore, the whole simulation area must be split into segments of usually different materials. Not all of them will look like cuboids, either. However, if the following conditions are fulfilled, the segments can be treated as warped cuboids. In this case, the calculation of the equipotential surfaces can be performed as described.

Like the sides of a cuboid, it must be possible to define such six “sides” for the segment — an upper, lower, left, right, top, and bottom “side”. Usually the semiconductor segment has such a shape: The simulation domain of the silicon wafer with planar surfaces, except the upper face with the oxide-interface, for instance. In this case it is easy to extract the six “sides” of the warped cuboid. Even other regions with four surrounding flat faces and two warped faces, each at the top and the bottom, can be handled easily. Additionally other criteria for finding the “sides” of the cuboids are conceivable, even though there might be problems when these “sides” have to be detected automatically.

Such a segment with warped six “sides” gives a topological cuboid. By applying a voltage at two of the opposing “sides”, the electric field distribution can be evaluated. By connecting a voltage to each of the remaining opposite side pairs and also by calculating the potential distribution we get three potential distributions.

With this methodology, the topological cuboids of the grid can be calculated. Doing so, the distances between the selected potential ticks can be tuned arbitrarily and the aspect ratios of the resulting cuboids can be varied independently.

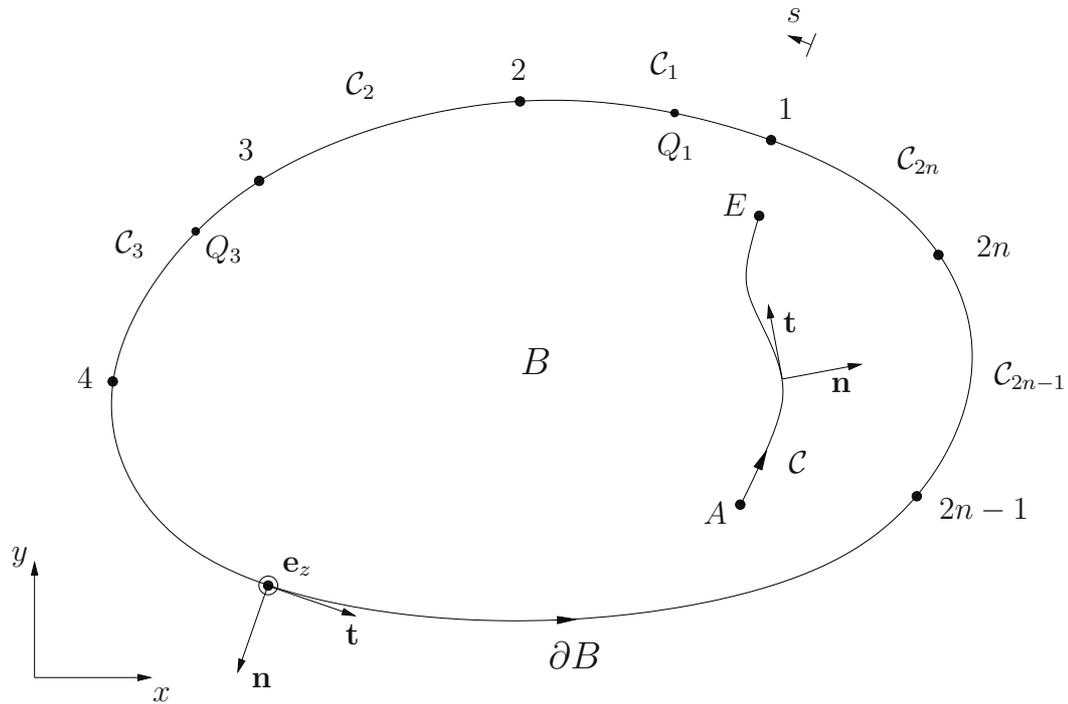


Figure 5.12: Domain for the electric field calculation.

To achieve a valid Delaunay grid, the grid points produced by this method are inserted in the global point set of the geometry and are meshed by the grid generator. The demands on Delaunay grids are nearly fulfilled by the orthogonality of these cuboids. As a result of the given boundary points, a distortion of these cuboids may be caused.

5.3.5 Proof of the Two-Dimensional Duality

First we start with the stationary field equations. With constant permittivity ε the equations reduce to

$$\operatorname{div} \mathbf{E} = 0 \quad (5.18)$$

$$\operatorname{curl} \mathbf{E} = \mathbf{0} \quad (5.19)$$

which must be satisfied in the capacitor domain B , shown in Figure 5.12. The contacts of the capacitor are the boundaries

$$C_{2i}, \quad i = 1(1)n, \quad (5.20)$$

which alternate with the Neumann boundaries

$$\mathcal{C}_{2i-1}, \quad i = 1(1)n. \quad (5.21)$$

The tangent and normal vectors \mathbf{t} and \mathbf{n} are defined as

$$\mathbf{n} = \mathbf{t} \times \mathbf{e}_z \quad (5.22)$$

and

$$\mathbf{t} = \mathbf{e}_z \times \mathbf{n}. \quad (5.23)$$

Solution with a Scalar Potential φ

The first way for solving the partial differential equation system (5.18) and (5.19) is performed by a gradient field

$$\mathbf{E} = -\text{grad } \varphi, \quad (5.24)$$

which satisfies (5.19) implicitly and delivers the differential equation

$$\text{div grad } \varphi = \Delta \varphi = 0 \quad \text{in } B \quad (5.25)$$

with the boundary conditions

$$\text{Dirichlet:} \quad \varphi = \Phi_{2i} = \text{const} \quad \text{on } \mathcal{C}_{2i}, \quad i = 1(1)n \quad (5.26)$$

$$\text{hom. Neumann:} \quad \partial_n \varphi = 0 \quad \text{on } \mathcal{C}_{2i-1}, \quad i = 1(1)n \quad (5.27)$$

The boundary problem (5.25), (5.26) and (5.27) has a well-defined solution. Additionally the following integrals will be defined:

$$\Lambda_{2i} = \int_{\mathcal{C}_{2i}} E_n \, ds = - \int_{\mathcal{C}_{2i}} \partial_n \varphi \, ds, \quad i = 1(1)n \quad (5.28)$$

and the trivial portions on the homogenous Neumann boundaries

$$\Lambda_{2i-1} = \int_{\mathcal{C}_{2i-1}} E_n \, ds = - \int_{\mathcal{C}_{2i-1}} \partial_n \varphi \, ds = 0, \quad i = 1(1)n. \quad (5.29)$$

Integration of (5.25) over the whole region B and applying Gauss' integral theorem delivers

$$0 = \int_B \Delta \varphi \, dA = \int_B \text{div grad } \varphi \, dA = \int_{\partial B} \text{grad } \varphi \cdot \mathbf{n} \, ds = \int_{\partial B} \partial_n \varphi \, ds = - \sum_{i=1}^n \Lambda_{2i} \quad (5.30)$$

Solution with an (Electric) Vector Potential \mathbf{A}^e

Different to the previous way, the basic approach is a vector potential

$$\mathbf{E} = \text{curl } \mathbf{A}^e, \quad \mathbf{A}^e = \psi(x, y) \mathbf{e}_z \quad (5.31)$$

which satisfies (5.18). Because of the two-dimensionality of the problem, the vector potential shows only a z-component. Transformation delivers

$$\begin{aligned} \mathbf{E} &= \text{curl}(\psi \mathbf{e}_z) = \nabla \times (\psi \mathbf{e}_z) = -\mathbf{e}_z \times \nabla \psi \\ &= \text{grad } \psi \times \mathbf{e}_z = \partial_y \psi \mathbf{e}_x - \partial_x \psi \mathbf{e}_y \end{aligned} \quad (5.32)$$

$$\mathbf{t} \cdot \mathbf{E} = \mathbf{t} \cdot (\text{grad } \psi \times \mathbf{e}_z) = \text{grad } \psi \cdot (\mathbf{e}_z \times \mathbf{t}) = -\text{grad } \psi \cdot \mathbf{n} = -\partial_n \psi \quad (5.33)$$

$$\mathbf{n} \cdot \mathbf{E} = \mathbf{n} \cdot (\text{grad } \psi \times \mathbf{e}_z) = \text{grad } \psi \cdot (\mathbf{e}_z \times \mathbf{n}) = \text{grad } \psi \cdot \mathbf{t} = \partial_t \psi \quad (5.34)$$

$$\begin{aligned} \text{curl } \mathbf{E} &= \nabla \times (\text{grad } \psi \times \mathbf{e}_z) = (\mathbf{e}_z \nabla) \text{grad } \psi - \mathbf{e}_z (\nabla \text{grad } \psi) \\ &= \underbrace{\partial_z \text{grad } \psi}_0 - \mathbf{e}_z (\nabla \nabla \psi) = -\mathbf{e}_z \Delta \psi = \mathbf{0} \end{aligned} \quad (5.35)$$

$$\implies \Delta \psi = 0 \quad \text{in } B. \quad (5.36)$$

The boundary conditions (5.26) and (5.27) result to

$$\begin{aligned} \mathcal{C}_{2i} : \quad \varphi = \text{const} &\implies E_t = 0 \implies \mathbf{t} \cdot \mathbf{E} = 0 \\ \xrightarrow{(5.33)} \partial_n \psi = 0 &\quad \text{on } \mathcal{C}_{2i}, \quad i = 1(1)n \end{aligned} \quad (5.37)$$

\implies **homogenous Neumann conditions**

$$\begin{aligned} \mathcal{C}_{2i-1} : \quad \partial_n \varphi = 0 &\implies E_n = 0 \implies \mathbf{n} \cdot \mathbf{E} = 0 \\ \xrightarrow{(5.34)} \psi = \Psi_{2i-1} = \text{const} &\quad \text{on } \mathcal{C}_{2i-1}, \quad i = 1(1)n \end{aligned} \quad (5.38)$$

\implies **Dirichlet conditions**

The previously defined integral is carried out as

$$\int_{\mathcal{C}} E_n \, ds = \int_{\mathcal{C}} \mathbf{E} \cdot \mathbf{n} \, ds \stackrel{(5.34)}{=} \int_{\mathcal{C}} \partial_t \psi \, ds = \int_A^E d\psi = \psi|_E - \psi|_A \quad (5.39)$$

$$(5.40)$$

and further on by defining the arbitrary points

$$Q_{2i-1}(s_{2i-1}) \in \mathcal{C}_{2i-1} \quad (5.41)$$

follows

$$\int_{Q_{2n-3}}^{Q_{2n-1}} E_n \, ds = \int_{s_{2i-3}}^{s_{2i-1}} E_n \, ds = \psi(s_{2i-1}) - \psi(s_{2i-3}) = - \int_{s_{2i-3}}^{s_{2i-1}} \partial_n \varphi \, ds = \Lambda_{2i}. \quad (5.42)$$

or

$$\Psi_{2i-1} = \sum_{j=1}^{i-1} \Lambda_{2j} + \Psi_1, \quad i = 1(1)n \quad (5.43)$$

where the value of Ψ_1 can be chosen arbitrarily.

Summary of Both Solutions

$$\begin{array}{lll} \text{I:} & \begin{array}{l} \Delta \varphi = 0 \\ \varphi = \Psi_{2i} = \text{const} \\ \partial_n \varphi = 0 \end{array} & \begin{array}{l} \text{in } B \\ \text{on } \partial B_{2i}, \quad i = 1(1)n \\ \text{on } \partial B_{2i-1}, \quad i = 1(1)n \end{array} \end{array} \quad (5.44)$$

and

$$\begin{array}{lll} \text{II:} & \begin{array}{l} \Delta \psi = 0 \\ \partial_n \psi = 0 \\ \psi = \Psi_{2i-1} = \text{const} \end{array} & \begin{array}{l} \text{in } B \\ \text{on } \partial B_{2i}, \quad i = 1(1)n \\ \text{on } \partial B_{2i-1}, \quad i = 1(1)n \end{array} \end{array} \quad (5.45)$$

which describe the same fields, if Ψ_{2i-1} satisfy (5.28) and (5.43).

Field Lines of \mathbf{E}

The field lines of \mathbf{E} are defined by the differential equation

$$\frac{dy}{dx} = \frac{E_y}{E_x}. \quad (5.46)$$

By insertion of (5.32) it follows

$$0 = dy E_x - dx E_y = dy \partial_y \psi + dx \partial_x \psi = d\psi \implies \psi = \text{const} \quad (5.47)$$

which are the equipotential lines of ψ . Additionally, as the field lines of \mathbf{E} are orthogonal to the equipotential lines of φ , it follows that the set of equipotential lines $\varphi = \text{const}$ and $\psi = \text{const}$ are orthogonal.

This derivation has shown that the duality of the field lines and equipotential lines within the given set of differential equations can be obtained by a replacement of Dirichlet (homogenous Neumann) by homogenous Neumann (Dirichlet) boundary conditions. The values of Ψ_{2n-1} have to be evaluated by (5.28) and (5.43).

In this generalization of the capacitor problem, even more than two electrodes in the original constellation could be handled. In this case, if the original field problem has n boundaries with imposed voltage, it requires also n Neumann boundaries (electrodes for the dual system) which alternate with the Dirichlet boundaries along the whole surface.

As neither a bias of Ψ_1 nor an arbitrary constant scaling factor of ψ change the shape of the field lines (equipotential lines of ψ), in case of exactly two contacts, the actual values of the two dual boundary voltages can be set to two different arbitrary values, most notably to values suitable for the computation.

5.3.6 The Three-Dimensional Capacitor Model

Unfortunately such a duality is not guaranteed in three dimensions. The theoretical expansion of the two-dimensional capacitor model delivers a three-dimensional capacitor with six sides where two of them act as electrodes. To determine the field lines, three potentials have to be calculated: one with the original electrode setting, one with two other opposite electrodes and the third one with the remaining two opposite electrodes.

One might expect that the expanded two-dimensional duality formulation can be expressed as follows:

The three fields build an orthogonal system. The field lines of each electric field lie within the equipotential surfaces of each of the remaining potentials and the conclusion is that the intersections of two equipotential surfaces deliver the field lines of the third potential.

Nevertheless, this might not hold under certain circumstances. In three dimensions, the duality formulation is not guaranteed, because the three potentials, which can be

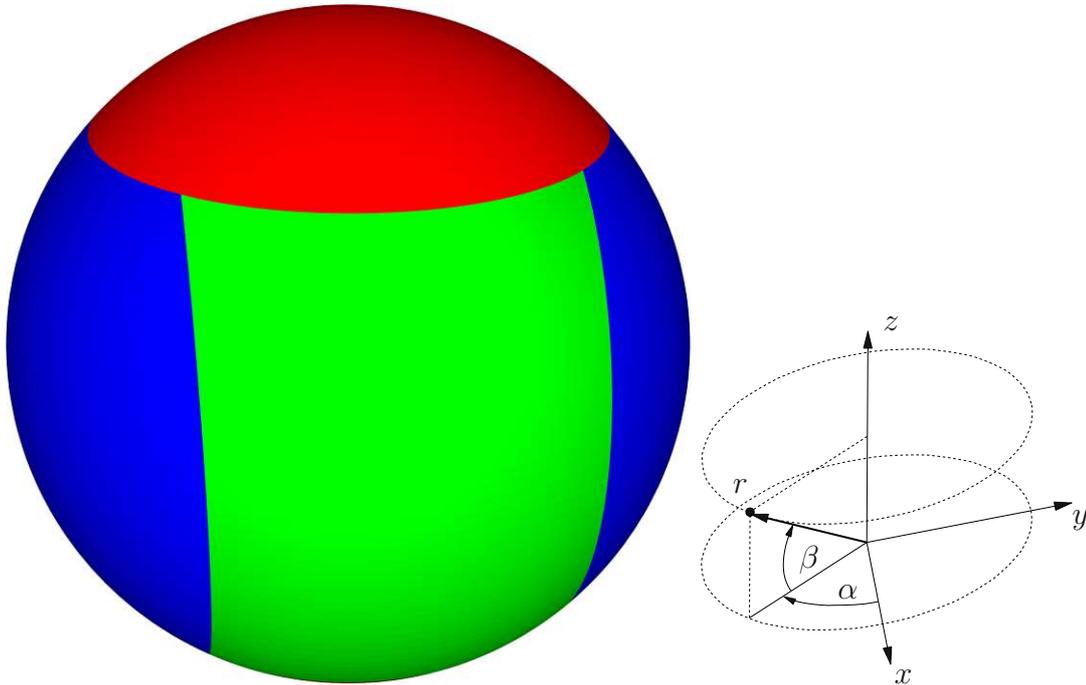


Figure 5.13: Spherical capacitor with electrodes placed at the top and bottom (hidden) cups. The electrodes for the second potential calculation are placed at the remaining cups at the left and right side. The electrodes for the third potential calculation are the remaining areas of the surface.

considered as a curvilinear coordinate system, usually do not represent an orthogonal system. Particularly, it is not even guaranteed that the field lines on the surface of the simulation domain are the equipotential lines of one of the other potentials. Additionally, the different electric fields are not perpendicular [70].

The orthogonality of the equipotential surfaces only exists, if the coordinate lines on the surface follow lines of curvature. This happens, when the coordinate lines follow surface lines of maximum or minimum curvature. However, this is only fulfilled for simple geometries and usually not satisfied [70].

The following example should clarify the situation. Consider a spherical device region as shown in Figure 5.13. The coordinates of the device are given in spherical coordinates by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = r \begin{pmatrix} \cos \alpha \cos \beta \\ \sin \alpha \cos \beta \\ \sin \beta \end{pmatrix}. \quad (5.48)$$

Constant radius $r = r_0 = \text{const}$ delivers the surface of this domain. The meridians of

the sphere are formed by $\alpha = \text{const}$ and $\beta = \text{const}$ and represent circles of constant height. The upper and lower contacts of the domain of the capacitor are cups of the sphere and will be limited by the circles at $\beta = \beta_0$ and $\beta = \beta_1$. The extracted potential is φ . With this contact setting the solution is rotationally symmetric and it follows that the equipotential lines at the surface of the sphere are circles of constant height $\beta = \beta_i = \text{const}$ and the field lines at the surface are meridians between β_0 and β_1 with $\alpha = \alpha_i = \text{const}$.

Inside the device, the equipotential surfaces are perpendicular to the surface of the sphere and the field lines are also perpendicular to the former. Fortunately, a complete analytical solution of the potential distribution is not necessary. The major prediction is that the field lines at the surface of the sphere \mathcal{C} are represented by meridians

$$\mathcal{C} = \mathcal{C}(\beta) \Big|_{\substack{r=r_0 \\ \alpha=\alpha_i}} = r_0 \begin{pmatrix} \cos \alpha_i \cos \beta \\ \sin \alpha_i \cos \beta \\ \sin \beta \end{pmatrix} \quad (5.49)$$

and the equipotential lines \mathcal{P} at the surface

$$\mathcal{P} = \mathcal{P}(\alpha) \Big|_{\substack{r=r_0 \\ \beta=\beta_i}} = r_0 \begin{pmatrix} \cos \alpha \cos \beta_i \\ \sin \alpha \cos \beta_i \\ \sin \beta_i \end{pmatrix} \quad (5.50)$$

are lines of constant height. At an intersection point, the tangential vectors of these curves are perpendicular which can be expressed by a vanishing inner product

$$\frac{d\mathcal{C}}{d\beta} \Big|_{\substack{r=r_0 \\ \alpha=\alpha_i \\ \beta=\beta_i}} \cdot \frac{d\mathcal{P}}{d\alpha} \Big|_{\substack{r=r_0 \\ \alpha=\alpha_i \\ \beta=\beta_i}} = r_0 \begin{pmatrix} -\cos \alpha_i \sin \beta_i \\ -\sin \alpha_i \sin \beta_i \\ \cos \beta_i \end{pmatrix} \cdot r_0 \begin{pmatrix} -\sin \alpha_i \cos \beta_i \\ \cos \alpha_i \cos \beta_i \\ 0 \end{pmatrix} \equiv 0. \quad (5.51)$$

So far, there was no specification of the other four electrodes. However, any possible specification of them does not change the layout of the original field lines. The second potential ψ is defined by the left and right electrodes as the cup of the remaining part of the sphere at $y < y_0$ and $y > y_1$. The front and back contacts are the remaining parts of the surface and deliver the potential ϑ . The borderline of the left contact and the front contact is a line of constant y -coordinate

$$\mathcal{Q} = \mathcal{Q}(\beta) \Big|_{\substack{r=r_0 \\ y=y_0}} = \begin{pmatrix} \pm \sqrt{r_0^2 \cos^2 \beta - y_0^2} \\ y_0 \\ r_0 \sin \beta \end{pmatrix} = \begin{pmatrix} f(\beta) \\ y_0 \\ r_0 \sin \beta \end{pmatrix} \quad (5.52)$$

which is an equipotential line of ψ as well as an equipotential line of ϑ and should be expected to be a field line of φ . However, it can be seen that the curves $\mathcal{C}(r_0, \alpha_i, \beta)$ and $\mathcal{Q}(r_0, y_0, \beta)$ are not identical. The derivative

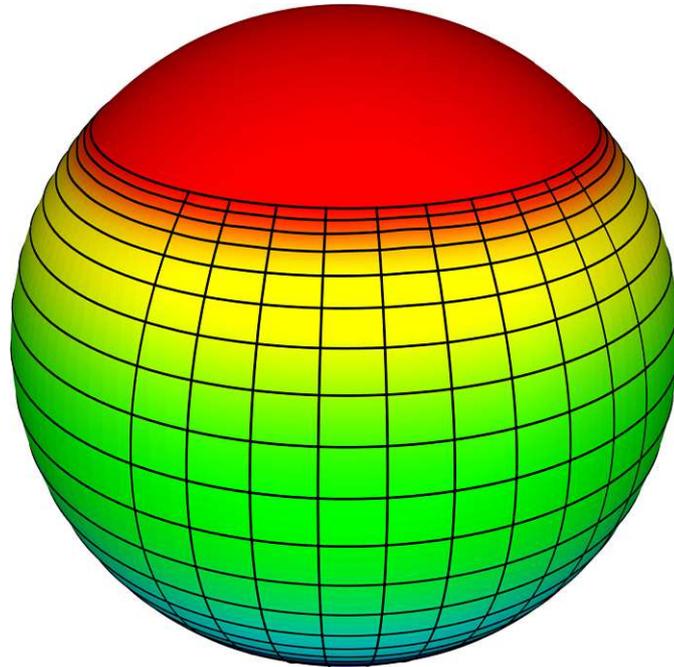
$$\frac{d\mathcal{Q}}{d\beta}(r_0, y_0, \beta) = \begin{pmatrix} f'(\beta) \\ 0 \\ r_0 \cos \beta \end{pmatrix} \quad (5.53)$$

has no y-component and the inner product at an intersection point (remember $y_0 = r_0 \sin \alpha_i \cos \beta_i$) usually does not give zero.

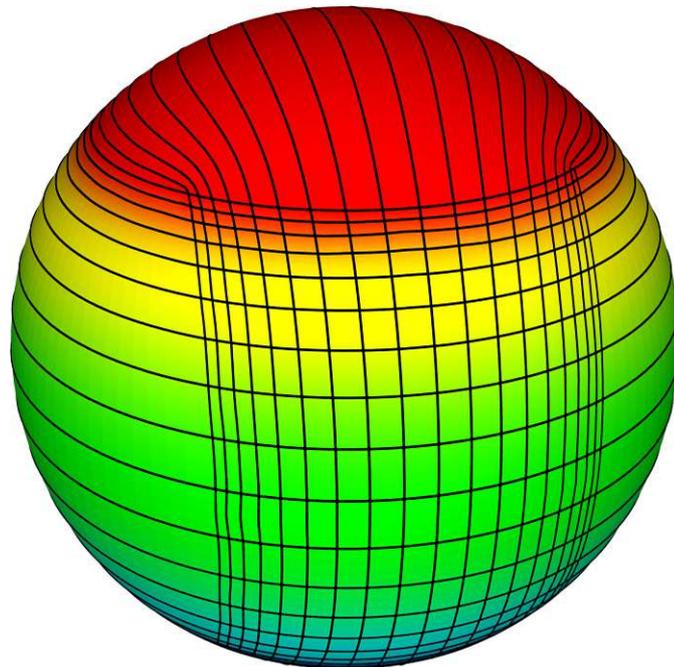
$$\begin{aligned} \left. \frac{d\mathcal{Q}}{d\beta} \right|_{\substack{r=r_0 \\ y=y_0 \\ \beta=\beta_i}} \cdot \left. \frac{d\mathcal{P}}{d\alpha} \right|_{\substack{r=r_0 \\ \alpha=\alpha_i \\ \beta=\beta_i}} &= \begin{pmatrix} f'(\beta_i) \\ 0 \\ r_0 \cos \beta_i \end{pmatrix} \cdot r_0 \begin{pmatrix} -\sin \alpha_i \cos \beta_i \\ \cos \alpha_i \cos \beta_i \\ 0 \end{pmatrix} \\ &= -\frac{r_0^2 \cos \beta_i \sin \beta_i}{\sqrt{r_0^2 \cos^2 \beta_i - y_0^2}} \cdot r_0 \sin \alpha_i \cos \beta_i \\ &= -\frac{r_0^3 \cos \beta_i \sin \beta_i \sin \alpha_i \cos \beta_i}{r_0 \cos \beta_i \cos \alpha_i} \\ &= -\frac{1}{2} r_0^2 \tan \alpha_i \sin 2\beta_i \neq 0 \end{aligned} \quad (5.54)$$

Therefore, these lines are not orthogonal and cannot be field lines of φ . In Figure 5.14(a) the equipotential lines and field lines of the original electrode placement are shown. Figure 5.14(b) shows the original equipotential lines and the equipotential lines of the switched electrode placement. These equipotential lines do not represent the field lines. However, the resulting quadrangles at the surface appear to be nearly orthogonal. Therefore, for three dimensions the field lines of φ cannot be formed by the intersections of the equipotential surfaces ψ and ϑ . Moreover, the equipotential surfaces of φ , ψ , and ϑ are not orthogonal any longer.

Nevertheless, usually even these surfaces will be nearly orthogonal, within the geometries considered and due to the smoothness of the resulting potential lines arising from the Laplace equation. Therefore, nearly cuboidal elements and smooth grids will be produced by the equipotential surfaces. The equipotential surfaces conform to the boundaries. As a result all points inside the simulation area will be reached by the equipotential surfaces of $\varphi = \text{const}$, $\psi = \text{const}$ and $\vartheta = \text{const}$ — a geometrical point (x, y, z) inside the simulation domain can be described by a triple $(\varphi, \psi, \vartheta)$. Only in rare cases, an ambiguous mapping, a folding of the curvilinear coordinate lines, may arise [36][70].



(a) Field lines and equipotential lines defined by the original electrode setting



(b) Equipotential lines defined by the replaced electrodes. Additionally the equipotential lines of the original electrode placement are drawn.

Figure 5.14: Potential and field lines of the spherical capacitor.

5.3.7 Evaluation of the Electric Potential of the Capacitor

By selecting several potential value ticks of the three different potentials and calculating the intersection points of the equipotential surfaces, the point set for the grid generation is derived. With the selection of the tick values of the potentials, the grid density can be controlled in various ways.

To simplify the expressions, the boundary values of the potentials are set to 0 and 1. Each geometrical point (x, y, z) inside the capacitor has a potential representation $(\varphi, \psi, \vartheta)$ within the potential range $([0, 1] \times [0, 1] \times [0, 1])$. As the discrete maximum principle is satisfied, it seems plausible that the potential values raise monotonously along the field lines, going from one electrode to the opposite one. Therefore, except for folding it is a one-one mapping of geometrical points to potential coordinates. In this way, a selected potential triple $(\varphi, \psi, \vartheta)$ delivers a distinct geometrical point (x, y, z) inside the capacitor.

In a first step it is necessary to find an initial Delaunay tetrahedrization of the capacitor domain. This grid generation step does not need a special grid refinement. All material properties are linear, in particular a constant permittivity is used, and the Laplace equation as an elliptic partial differential equation is numerically well behaved. In addition, as a big advantage different from generating a grid for the whole device, only one capacitor segment has to be meshed. This grid can be produced regardless of other segments. This means that no additional grid points are induced by geometry and grid constrains of neighboring segments. A relatively crude grid can be used. Only for preserving the quality of the resulting cuboids, it is useful to introduce a refinement criterion based on the directions of the electric fields.

The solutions of the field equations is usually obtained by the Box Integration method (refer Section 3.1). The differential equation systems for solving the three potential distributions can be seen in Figure 5.1. The resulting equation system is of the form

$$\mathbf{A} \cdot \boldsymbol{\varphi} = \mathbf{b}. \quad (5.55)$$

with

$$\boldsymbol{\varphi} = (\varphi_i) \quad (5.56)$$

the unknown potential values on the grid points p_i .

The elements of the system matrix $\mathbf{A} = (a_{ij})$ and the right-hand side $\mathbf{b} = (b_i)$ are set to

$$a_{ij} = -\varepsilon_{ij} \frac{A_{ij}}{d_{ij}} \quad (5.57)$$

$$a_{ii} = \sum_{\forall i \neq j} a_{ij} \quad (5.58)$$

$$b_i = 0 \quad (5.59)$$

$\forall i, j : \exists \langle p_i p_j \rangle$ and p_i lying inside ∂V or on $\partial V_3, \partial V_4, \partial V_5$ or on ∂V_6 ,

$$a_{ii} = 1 \quad (5.60)$$

$$b_i = \Phi_1 \quad (5.61)$$

$\forall i, j : \exists \langle p_i p_j \rangle$ and p_i lying on ∂V_1 ,

$$a_{ii} = 1 \quad (5.62)$$

$$b_i = \Phi_2 \quad (5.63)$$

$\forall i, j : \exists \langle p_i p_j \rangle$ and p_i lying on ∂V_2 and

$$a_{ij} = 0 \quad (5.64)$$

for all other i, j .

These equations are used for calculating for the potential values φ_i on the grid points p_i . For the potentials ψ and ϑ an similar equation system is set up. Basically, all three potential evaluations use the same system matrix \mathbf{A} , except the lines and rows resulting from the boundary points where the adequate boundary conditions have to be inserted. For the evaluation of the potential values, a Conjugate Gradient Solver is advisable [71].

Why Choosing a Conjugate Gradient Solver

The matrix \mathbf{A} , for the equation system $\mathbf{A} \cdot \boldsymbol{\varphi} = \mathbf{b}$, is a sparse $n \times n$ matrix. Applying a Gaussian Solver destroys the sparse system, because by eliminating the entries under the diagonal a lot of entries above the diagonal become non-zero. Therefore, the memory usage for storing the system matrix elements increases to $O(n^2)$. Additionally the elimination of a fully occupied row requires $O(n^2)$

arithmetic operations and eliminating all rows is of $O(n^3)$. Therefore the use of a solver, which preserves the sparsity of the system and does not show a complexity of $O(n^3)$ is a better choice. With simple modifications, the system matrix \mathbf{A} can be made suitable for a Conjugate Gradient solver. Since a CG-solver requires a symmetrical system matrix, the original matrix must be transformed to become symmetric. The symmetry of the system matrix is violated in rows and columns where the values of φ_i are imposed. One possible method is to pre-eliminate these rows and columns and move the adequate entries to the right-hand side. This method requires a row number to point number remapping and each potential system delivers a completely different system matrix.

Another way is to eliminate asymmetrical matrix entries a_{ji} on the left-hand side by

$$a'_{jk} = a_{jk} - a_{ik} \frac{a_{ji}}{a_{ii}} \quad \forall k, \quad (5.65)$$

$$b'_j = b_j - b_i \frac{a_{ji}}{a_{ii}}, \quad (5.66)$$

$\forall i, j$ with $i < j, a_{ji} \neq a_{ij}$ and p_i on ∂V_1 or on ∂V_2 .

As all off-diagonal elements a_{ik} with $i \neq k$ of the concerned rows are zero, the transformation does not change any other entries than a_{ji} and simplifies to

$$a'_{ji} = a_{ji} - a_{ii} \frac{a_{ji}}{a_{ii}}, \quad (5.67)$$

$$b'_j = b_j - b_i \frac{a_{ji}}{a_{ii}}, \quad (5.68)$$

$\forall i, j$ with $i < j$ and p_i on ∂V_1 or on ∂V_2 .

For an efficient memory representation of the equations, it is necessary to use a sparse matrix representation. The chosen method is only practicable, if the matrix values are left unchanged during the whole solving process and direct access to the row entries can be omitted. This representation can be assembled easily by iteration over all points and insertion of the couplings of the edges that are connected with the iteration point. The storage representation of a matrix works with four arrays `diag[]`, `offdiag[]`, `colidx[]`, `startcolidx[]`, where the entries are defined as [49]:

Algorithm 5.3.1 *Sparse representation of $\mathbf{A} = (a_{ij})$*

```

 $a_{ii} = \text{diag}[i]$ 
startcolidx[i] <= k <= startcolidx[i+1]
j = colidx[k]
 $a_{ij} = \text{offdiag}[k]$ 

```

With this method, matrix-vector products can be performed directly on this data structure. Direct access to the row indices can be omitted by using the following algorithm:

Algorithm 5.3.2 *Matrix-vector product $\mathbf{A} \cdot \mathbf{x} = \mathbf{c}$*

```

for j in all point indices
  c[j]=diag[j] · x[j]
  for k=startcolidx[j] to startcolidx[j+1]-1
    c[j]=c[j] + offdiag[k] · x[colidx[k]]

```

No special sorting of the column indices is necessary. Matrix-matrix products, however, require direct access to the row entries within the above matrix representation and thus a search algorithm within the row indices becomes necessary. This kind of products has to be avoided. Therefore, a simple CG-algorithm without pivoting and without preconditioning is used. The algorithm looks as follows:

Algorithm 5.3.3 *CG-Method to solve $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$*

1. set

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \cdot \mathbf{x}_0, \quad \mathbf{d}_0 = \mathbf{r}_0$$

2. $k = 0, 1, \dots$ until \mathbf{r} is sufficiently set

$$\lambda_k = \frac{\mathbf{r}_k^T \cdot \mathbf{r}_k}{\mathbf{d}_k^T \cdot \mathbf{A} \cdot \mathbf{d}_k},$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k,$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \lambda_k \mathbf{A} \cdot \mathbf{d}_k,$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \cdot \mathbf{r}_{k+1}}{\mathbf{r}_k^T \cdot \mathbf{r}_k},$$

$$\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k.$$

In this case, only vector-vector products ($O(n)$) and matrix-vector products are needed. If only the non-zero entries of the matrix \mathbf{A} are stored, the memory usage is of the same order as the number of existing edges m . The number of arithmetic operations for a matrix-vector product is of $O(m)$, too. With the use of exact arithmetic, the equation solver delivers the solution at least after n recursion steps. With non-exact arithmetic a border for the residuum \mathbf{r} must be given and usually this border is reached with less than n steps [20][63][71].

While the mean value of edges in a tetrahedral mesh is of the same order than the number of grid points (usually multiplied with a typical factor, but surely not of quadratic or higher order), the matrix-vector product is also of $O(n)$. Therefore the memory usage is of $O(n)$ and the amount of arithmetic operations for solving the equation system is of order $O(n^2)$, which is a huge improvement compared to a Gaussian solver.

5.4 Alternative Approaches for the Potential Method

When applying the potential method, not all of the device segments can be used as such capacitors. Because of the strong nonlinearities of the coupled differential equations (refer Section 5.1), the grid resolution on the semiconductor segment must be high. While this method works well on the semiconductor segment of a device, on other segments it may be impossible to define the six sides of the cuboid. Sometimes an automatic search for the sides will not be possible and, due to the complexity of user interaction on three-dimensional data, this way would not seem feasible anyway.

An alternative variant of this method can be derived by only selecting two opposite electrodes of the capacitor. These electrodes can be selected in a more general way, for example by selecting interface boundaries between different segments. The equipotential surfaces are evaluated similarly, but different to the previously described method, the surface of these electrodes is triangulated and the intersections of equipotential surfaces and field lines starting from the boundary points are delivered to the grid generator. The field lines have to be evaluated directly, not indirectly via the dual electric fields, but by walking along the directions of the electric field.

$$\begin{pmatrix} x_{\text{new}} \\ y_{\text{new}} \\ z_{\text{new}} \end{pmatrix} = \begin{pmatrix} x_{\text{old}} \\ y_{\text{old}} \\ z_{\text{old}} \end{pmatrix} + \delta \cdot \mathbf{E}(x_{\text{old}}, y_{\text{old}}, z_{\text{old}}). \quad (5.69)$$

The stepping factor δ must be chosen to take on appropriate small value. As the discrete electric field is constant inside each tetrahedron, best results are obtained

if the stepping takes place from one tetrahedron to the next one. This procedure is continued until the next desired potential tick is reached and the resulting point is delivered to the grid generator. Because of the orthogonality and the previously performed triangulation of the electrodes, the resulting grid elements are nearly straight trilateral prisms. They are tetrahedrized and can be split easily into three tetrahedrons. A disadvantage of this method is, that due to numerical errors while walking along the field lines, the resulting prisms may be twisted and may cause problems during grid generation.

Unfortunately these methods have a fundamental disadvantage. If a grid point is inserted at the electrode surface (maybe a geometry point), it will cause many grid points at the intersections with the equipotential planes. Moreover, inserting an additional potential tick causes a lot of grid points along the field lines. For preventing this induced point insertion, the use of point deletion algorithms similar to a terminating lines algorithm for ortho grids is necessary.

The previously described methods disperse grid points by a selected method and the grid generation is done by a Delaunay grid generator. The satisfaction of the Delaunay criterion is necessary for the following device simulation that is based on Finite Boxes. Other methods of directly splitting edges of an initial grid are also feasible. To prevent obtuse angles in the tetrahedrons, usually the longest edges of the tetrahedrons are split. Especially with a combination with the potential method, the lengths of the edges can be weighted directionally dependent on size and direction of the electric field. Using this method also an anisotropic grid can be obtained. However, as the grid lines are split directly to obtain the desired grid density and no afterwards Delaunay tetrahedrization is performed, the satisfaction of the Delaunay criterion cannot be guaranteed by this method and it cannot be used for Box Integration. Applications for oxidation simulation using Finite Elements had still been performed by this method [73].

5.5 Example for the Potential Method and the Electrode Placement

The structure given in Figure 5.15 should be meshed by the original potential method. On the basis of this structure an electrode placement, different than selecting the physical top plane as the upper electrode is shown. As an interactive description of placing the electrodes is quite complicated, this artificial geometry with some simple and flat surfaces is selected.

The top electrode will not surround the whole top of the structure; it is rather placed along the hole in the middle of the structure. The whole bottom side of the

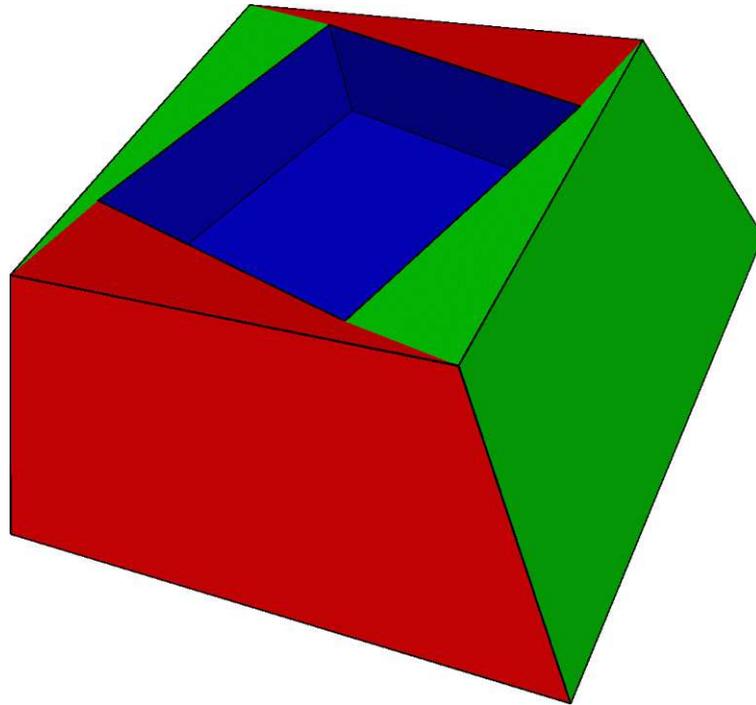


Figure 5.15: Three-dimensional structure with the placed electrodes.

structure is selected as its opposite electrode. In sum, all six electrodes must cover the entire surface. The chosen electrode placement can be seen in Figure 5.15 where opposite electrodes are shown in the same color.

A relatively crude grid with low demands on the quality is generated by the Delaunay grid generator *DeLink* [12] for evaluating the potential distributions. This initial grid has about 4000 grid points. The resulting potential distribution, derived by this grid is presented in Figure 5.16.

After calculating the three different potential distributions, the grid points are placed along the equipotential surfaces. A potential distribution with the resulting equipotential surfaces of all three potentials is shown in Figure 5.17.

Along the upper electrode surface, the distances of the potential values will be equidistant. The selected potential ticks start with a dense spacing at the upper electrode and grow monotonically to the lower electrode. The final grid can be seen in Figure 5.18. To illustrate the anisotropy of the generated grid, a quarter of the structure is cut away. Under the top surface, the dense grid spacing can be seen, while in lateral direction the density is lower.

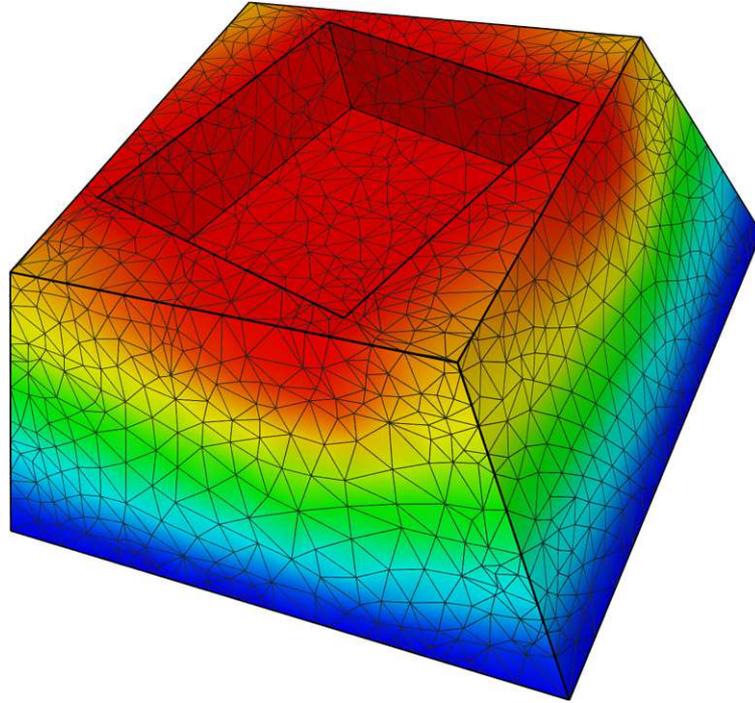


Figure 5.16: Three-dimensional structure with a potential distribution and shown base grid, on which the potential evaluation is performed.

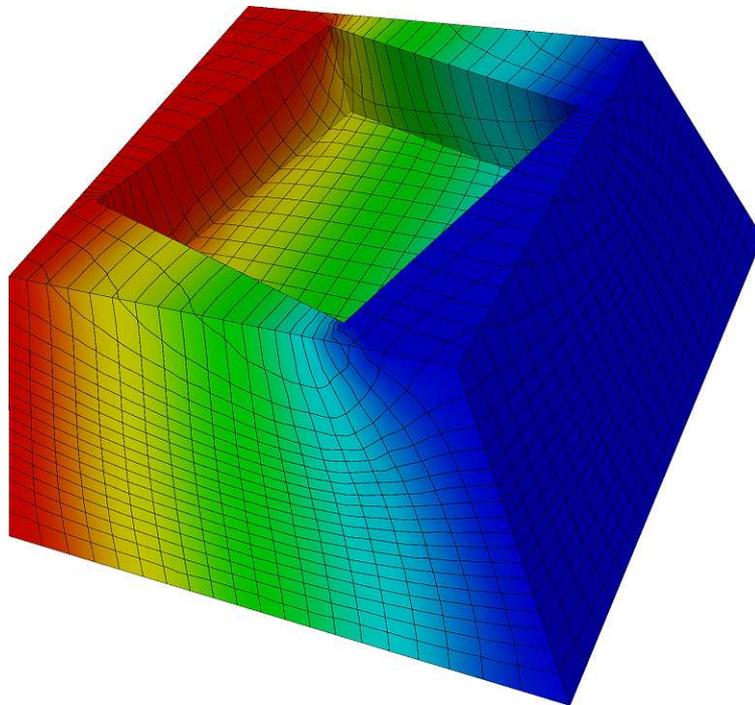


Figure 5.17: Final grid, developed by the potential method with shown potential distribution, contacts at left and right.

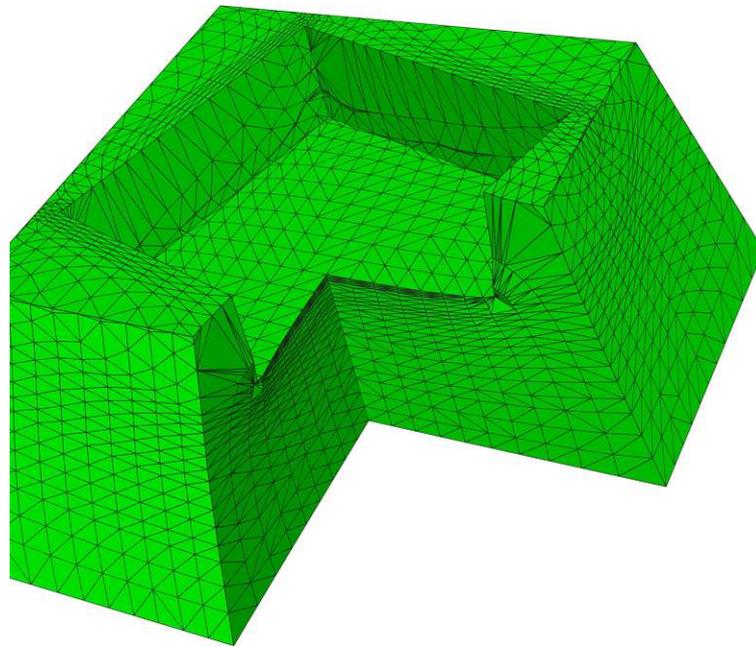


Figure 5.18: Final grid, a quarter of the device is cut away for illustration purposes.

Chapter 6

Applications of the Potential Based Method

IN THIS CHAPTER two applications of the potential based grid generation method are shown. The first example describes a FinFET structure which achieved attention in the last time. The examination of an EEPROM memory cell, which was of major interest in a device fabrication, is described in the second example.

6.1 Device Simulation of a FinFET

The basic structure of a FinFET published in [9] is shown in Figure 6.1. The silicon layer is formed as silicon on insulator (SOI). The fin is formed as a small silicon finger. For a better carrier absorption the source and drain regions are realized as large silicon pads. The active device area is the small fin region under the polysilicon gate. To obtain higher drive currents, additional fins can be applied in parallel. As the polysilicon is separated from the silicon by a thin oxide layer, the current density inside the silicon fin can be controlled by the gate voltage.

Depending on the device design we can distinguish between different kinds of FinFETs. Every side of the fin surrounded by the gate can be used as a control contact. If the surrounding oxide is thin, the charge carriers are controlled by the gate voltage. Using a thick oxide, the gate voltage does not influence the carriers. Generally there are two possibilities:

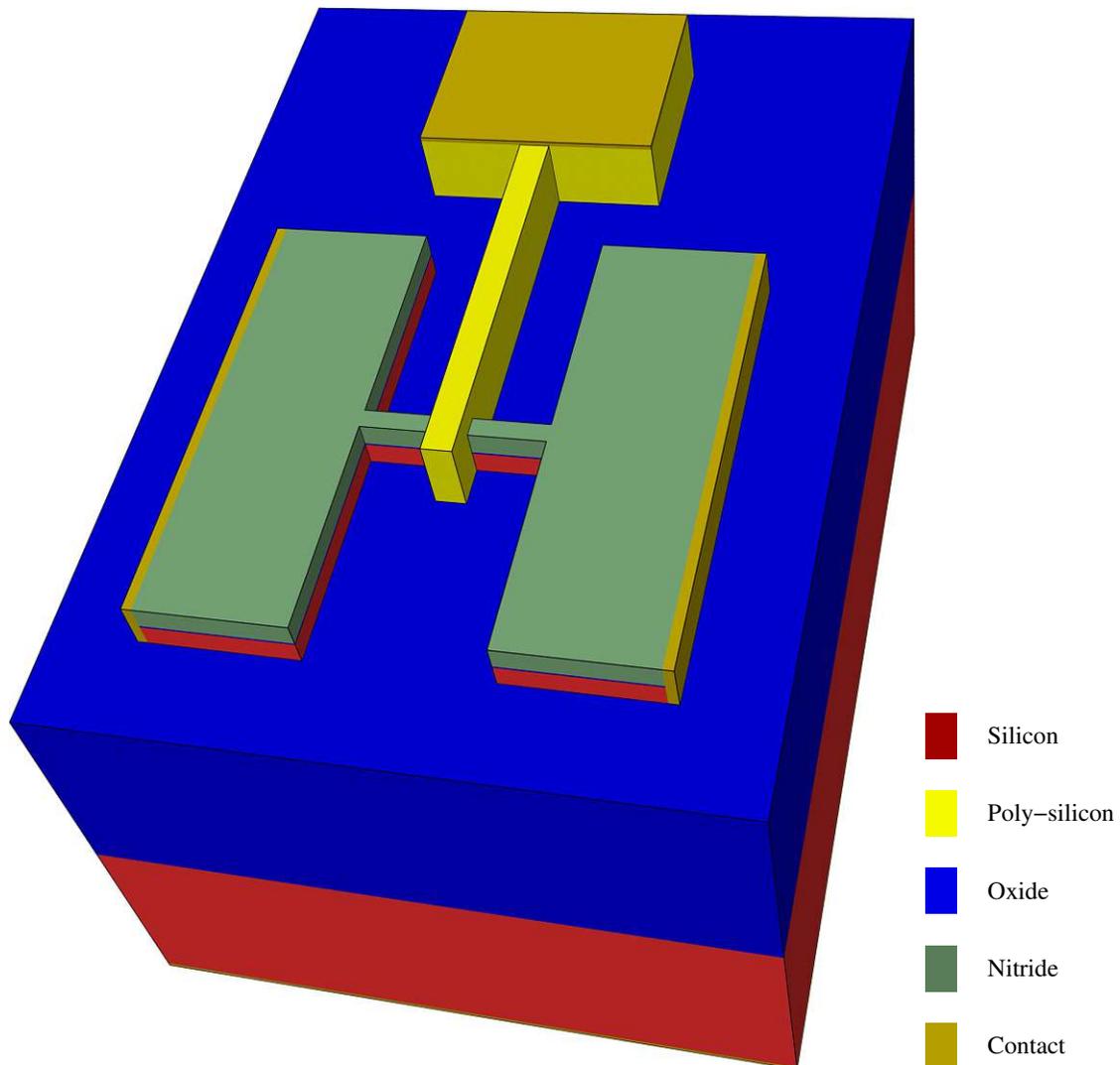


Figure 6.1: Geometry of the simulated FinFET structure.

- In double gate FinFET only two sides of the fin are enclosed by a thin oxide. The upper face has a thick oxide. Here the carriers on both sides of the fin can be controlled. This device is usually simulated by a two-dimensional simulation. However, effects at the corners of the fin can there not be accounted for. Actually at these regions the electric field is much higher and a higher current density will be obtained. These effects can only taken into account by three-dimensional simulations.
- In triple gate FinFET all three sides of the fin are enclosed by a thin oxide. Therefore, the carriers on all three sides can be controlled. In this case, the upper plane cannot be described by two-dimensional geometries and the device simulation has to be performed three-dimensional.

In Figure 6.2 an ortho grid is used for the simulation. Therefore, every grid line continues through the whole device. The active area is only the small part of the fin surrounded by the polysilicon. There the grid density must be high. As a negative sideeffect, the high grid line density propagates through all other segments. A unnecessarily high grid line density is produced in these other segments and the global amount of grid points is too high. Especially in the insulator segments, the grid density could be much cruder. Fragments of this grid line propagation can be seen as dense clustered grid lines on the surface of the oxide layer, which are caused by the dense grid lines in the fin.

The grid resulting from the potential method is shown in Figure 6.3. The grid density can be controlled for each segment and the amount of grid points is much smaller. A detail of the structure is shown in the next figures. Here only the silicon layer is drawn. Figure 6.4 shows the ortho grid, Figure 6.5 the potential based grid. The silicon segment can be split into three different regions:

- The active area in the middle of the fin. The grid density must be high in all three directions of space. The carrier concentration and current density distribution must be properly resolved in all three directions.
- The outer fin areas, which are attached on the left and right sides of the active area. The current density does not change much along the fin. The grid density along the fin can be cruder. No radial imposed electric field is present and the differences in the current density settle down, therefore, across the fin the current must be pretty good resolved.
- In the source and drain region the current lines spread out. Miscellaneous current density differences are settled and a crude grid can be chosen. The current lines inside the fin spread out in the regions of the contacts. In terms of the current flow, the grid lines are a good approximation of the current flow.

The different approaches are compared in Table 6.1. They were performed by the device simulator *MINIMOS-NT* [11] on an IBM p-Series machine with 1400 MHz Power 4+ processors. A third simulation was performed on a potential based grid, which is refined in the the active areas of the fin. The according output characteristics of the potential based simulations are shown in Figure 6.6. No significant change of the characteristic is determined by the different grids. The high ratio of grid elements to grid points of the potential method based grids compared to the ortho grid is caused by the different type of grid elements. Contour lines of the current density in the active fin area are shown in Figure 6.1.

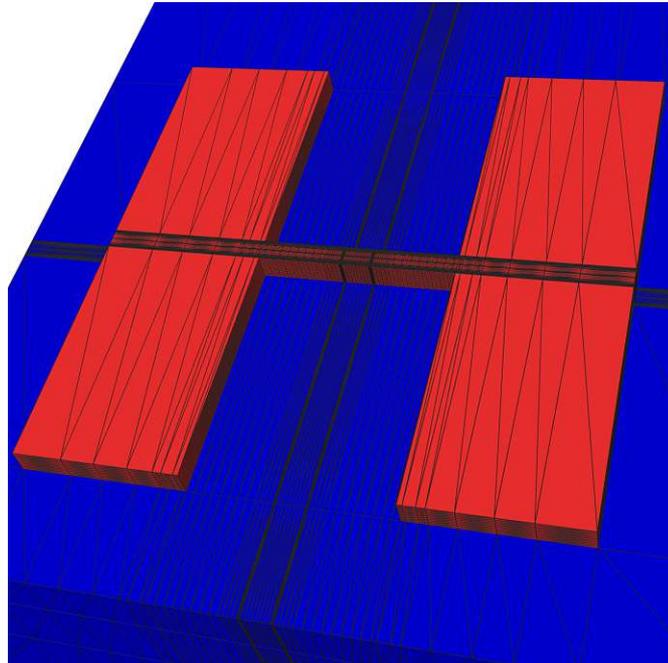


Figure 6.2: The oxide and silicon layers of the FinFET structure. The simulation grid is an ortho grid. The grid elements are split to tetrahedrons only for visualization purposes.

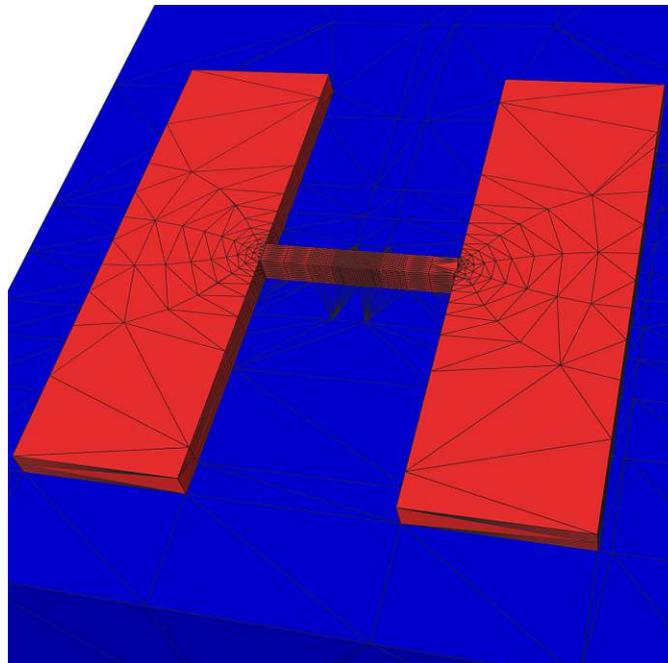


Figure 6.3: The oxide and silicon layers of the FinFET. The simulation grid is generated by the potential method. In the source and drain regions the approximation of the grid lines to the current lines can be seen.

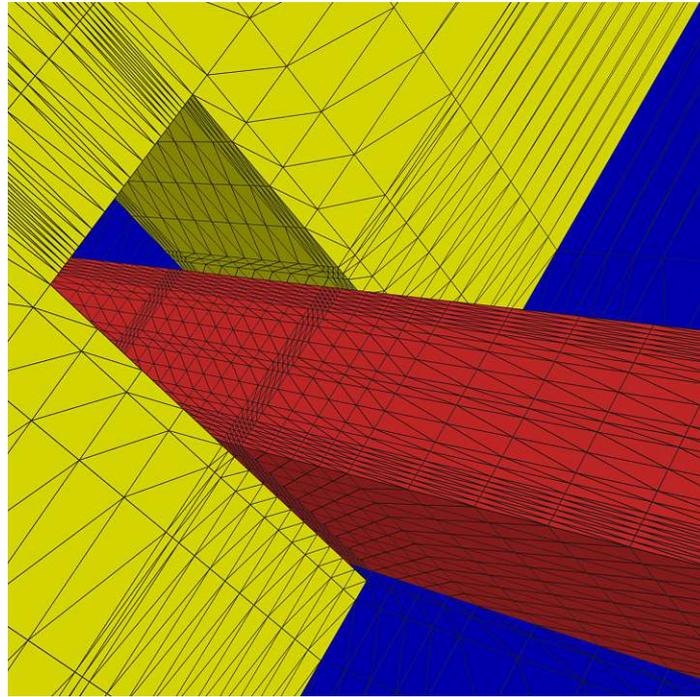


Figure 6.4: Generated ortho grid in the active regions of the transistor. Only the oxide, silicon and polysilicon segments are shown.

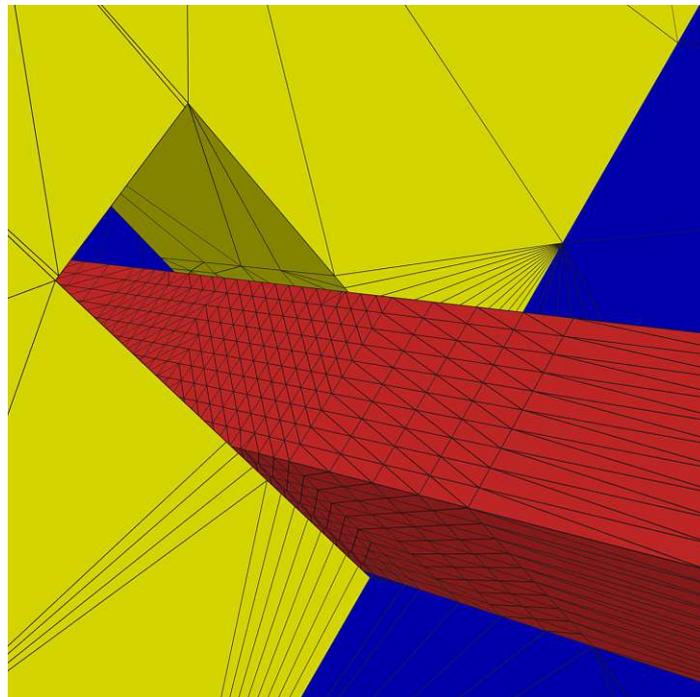


Figure 6.5: Potential based tetrahedral grid in the active regions of the transistor. Only the oxide, silicon and polysilicon segments are shown.

Grid type	Points	Elements	Elem. type	Rank	Sim. time
Ortho grid	48,480	42,840	Cuboids	83,998	124m30s
Potential method 1	11,186	32,666	Tetrahedrons	16,238	38m39s
Potential method 2	42,404	128,819	Tetrahedrons	65,132	390m10s

Table 6.1: Comparison of the different grids. Rank describes the rank of the system matrix of the equation system assembled by *MINIMOS-NT*, which was used for the electrical simulation.

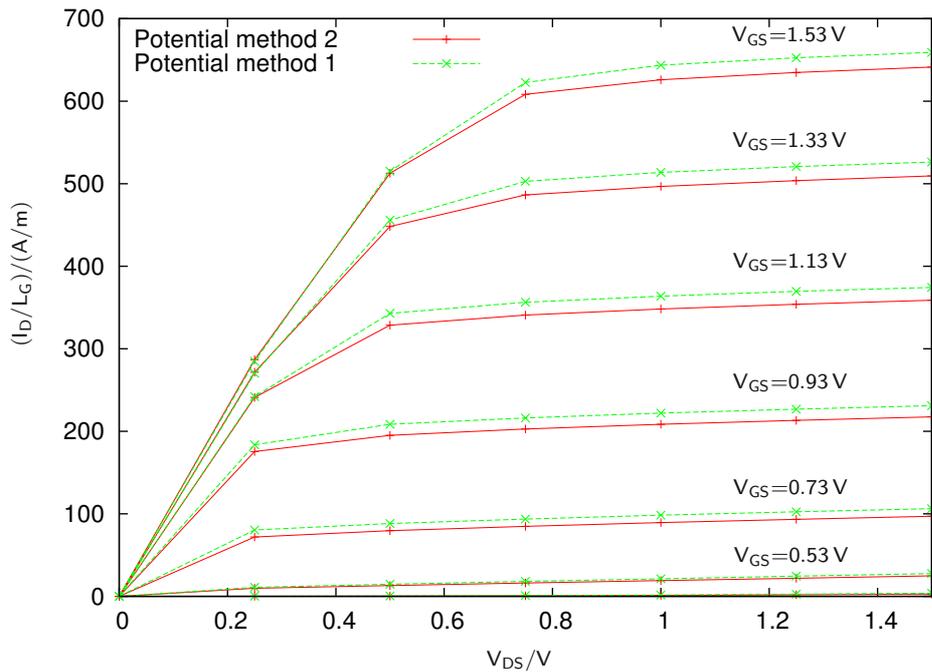


Figure 6.6: Simulated output characteristic of the FinFET. The drain current I_D is standardized by the gate length L_G .

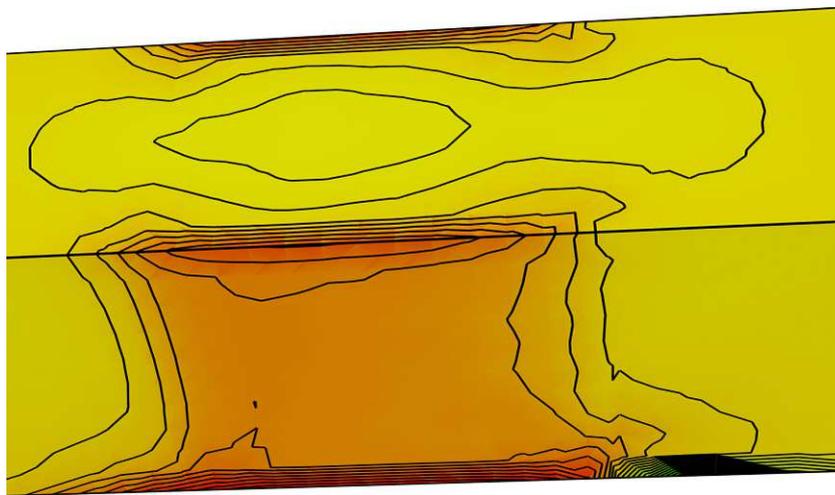


Figure 6.7: Contour lines of the electron current density in the channel area of the fin at $V_{GS} = V_{DS} = 1.5$ V.

6.2 Development of an EEPROM Memory-Cell

In the next example, the interaction of the different tools for the development of an EEPROM memory cell will be demonstrated.

The goal was to extract characteristic values of an EEPROM memory cell. For timing analysis, the capacitances between the semiconductor segment and the control- and floating-gate are crucial. To allow the extraction of these capacitances the three-dimensional extraction of the electric field is necessary. As it is impossible to describe such a complex device structure manually, the device structure had to be generated by a fully three-dimensional simulation of all the manufacturing processes of the device. The cross section of the memory cell and its typical layer thicknesses are shown in Figure 6.8.

6.2.1 Process Simulation of the Memory-Cell

While initial process steps like the oxidation of the silicon wafer for generating the field-oxide can be performed by a two-dimensional analysis, the following process steps have to be performed fully three-dimensional:

- The process simulation starts with the two-dimensional DIOS [65] simulation of the generation of the field-oxide on the silicon wafer.
- The two dimensional description is expanded to three dimensions.

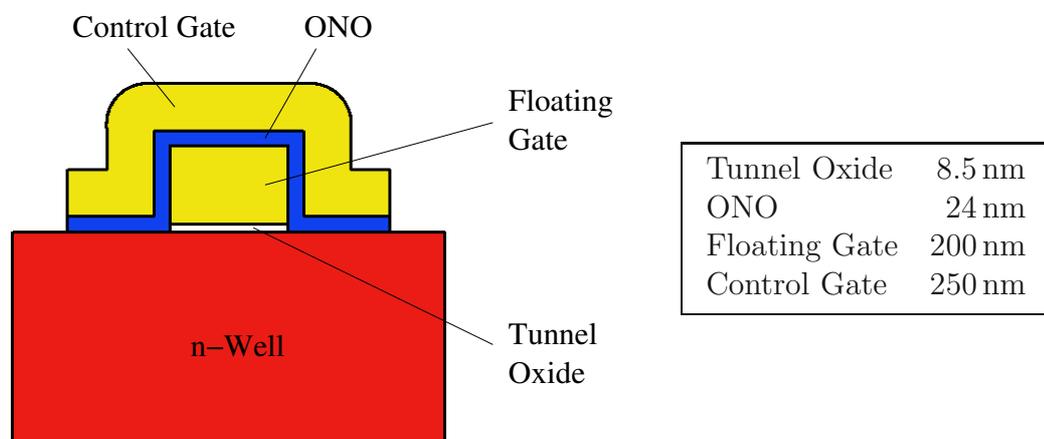


Figure 6.8: The cross section and typical layer thicknesses of the memory cell.

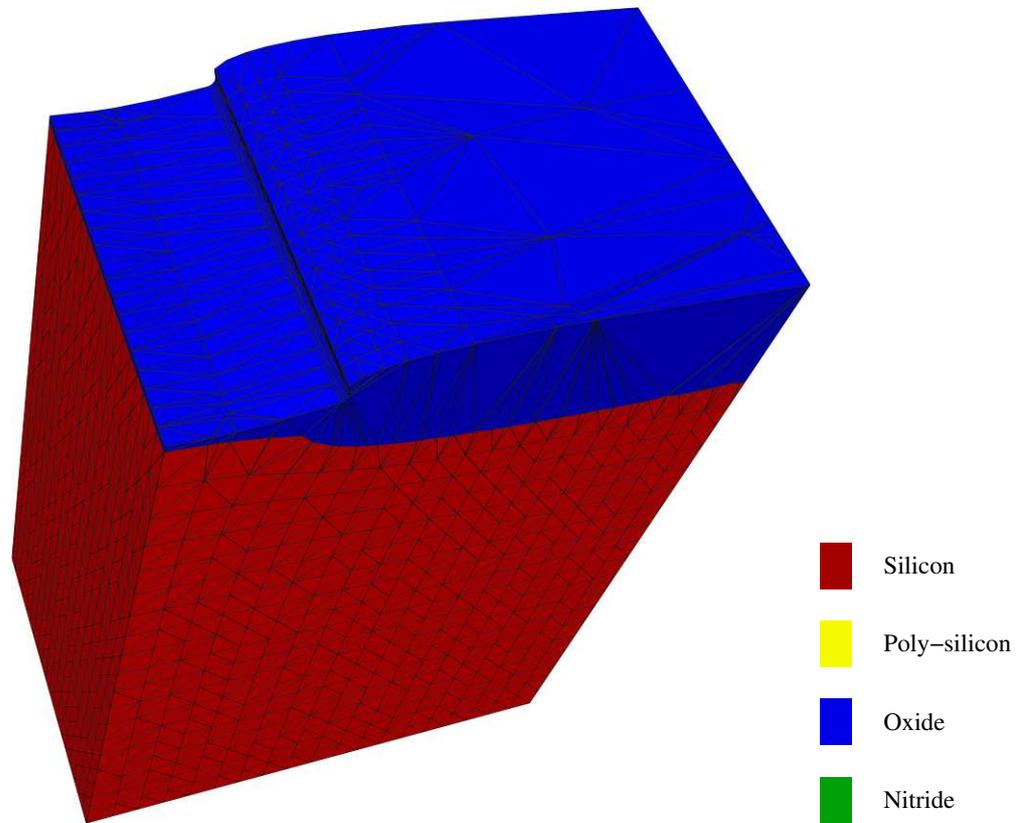


Figure 6.9: Expanded two-dimensional structure — 8,800 points, 51,000 tetrahedrons.

- Since the expansion tool delivers only a three-dimensional surface representation of the device and the tools for further process simulation require a three-dimensional grid representation, a Delaunay mesh-generation of the surface structure is required (Figure 6.9). As the potential method delivers fairly smooth grids, the silicon segment is meshed by this method. The use of a dense grid can be avoided, because no detailed electrical device simulation is necessary. A calculation of the capacitances can be performed on an even cruder grid.
- Addition of the floating-gate-mask, deposition of the floating-gate, and removal of the gate-mask.
- The deposition of the ONO-layers, as a sequence of three deposition simulation steps, namely for an **O**xide, a **N**itride, and again an **O**xide layer. Figure 6.10 shows the first added oxide layer, Figure 6.11 the complete ONO layers.
- The addition of the control-gate-mask, deposition of the control-gate, and removal of the gate-mask. The resulting device structure can be seen in Fig-

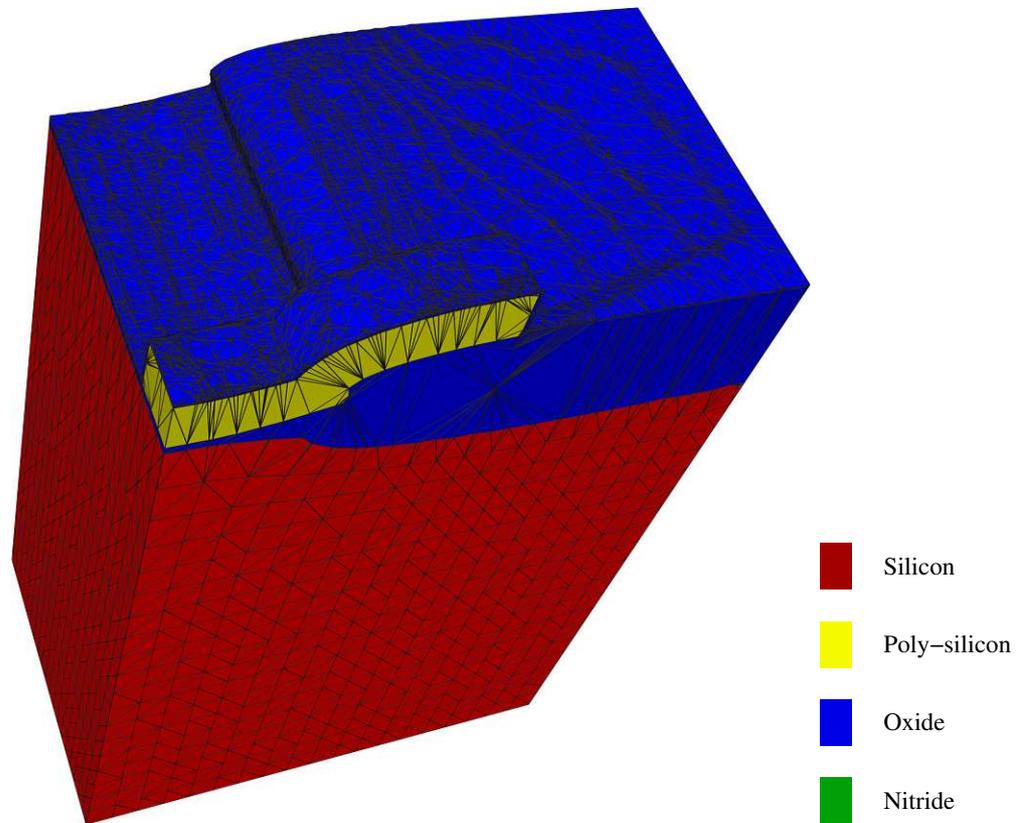


Figure 6.10: Floating-gate and first oxide layer — 18,300 points, 69,000 tetrahedrons.

ure 6.12.

- For the final electric field calculation it is necessary to account for the electric field in the surrounding material. This is done by a layer of an insulating material with relative permittivity $\epsilon_r = 1$ enveloping the device.

6.2.2 Electrical Analysis of the Device

The final electrical simulation and capacitance extraction is performed with the simulator STAP, part of the *Smart Analysis Programs* [55][56]. This is a three-dimensional interconnect simulator, which computes the field distribution inside the simulation area using Finite Elements. The capacitances are calculated via the energy of the electric field [54]. The result of the field calculation can be seen in Figure 6.13. An explanation for the large number of grid points and tetrahedrons is a global grid refinement of the program STAP, where each tetrahedron is split into

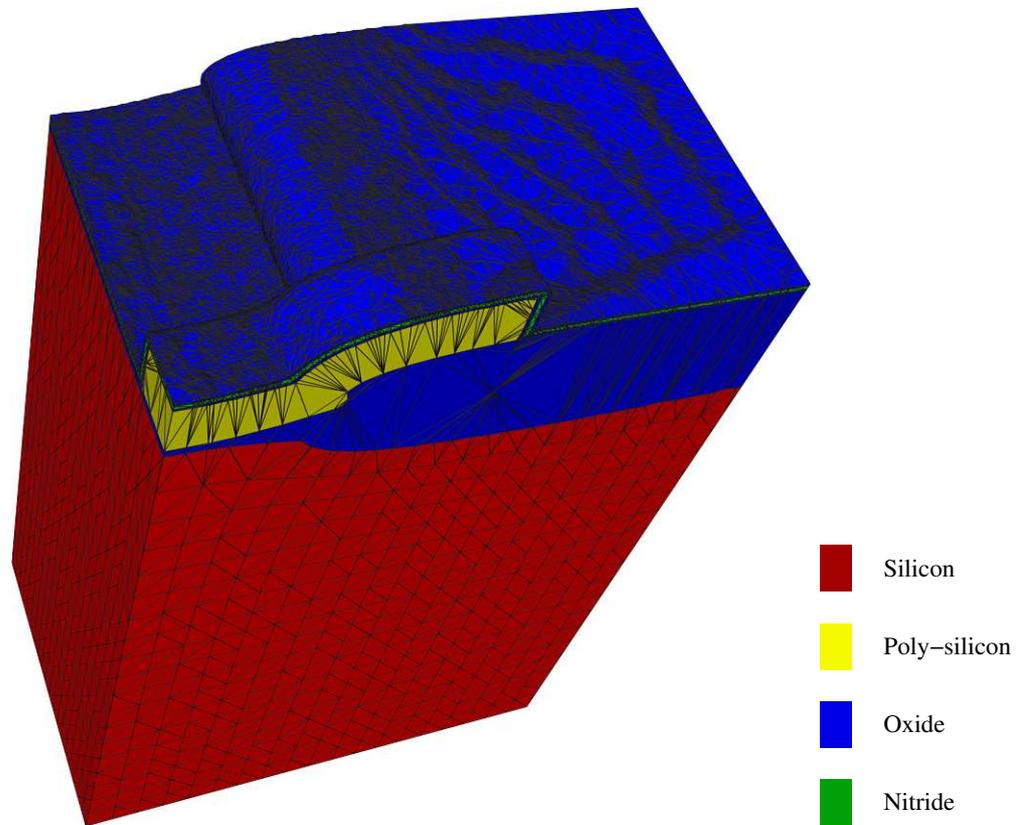


Figure 6.11: ONO layers added — 70,300 points, 341,000 tetrahedrons.

8 smaller ones. The floating-gate and the control-gate segments are connected to constant potential, 0 V and 1 V, respectively. The influence of the silicon segment is taken into account as a ground plane (connected to 0 V). Within this figure, the contact regions, which are the floating-gate, the control-gate, and the silicon segment, have been removed. Results of the capacitance extraction are shown in Table 6.2.

It is remarkable that the basically simple silicon/oxide-structure, starting with 51,000 tetrahedrons, increases to 400,000 tetrahedrons and the memory limits during process simulation are reached. However, the field extraction is performed by 3,400,000 tetrahedrons and the memory consumption is not of this magnitude. The enormous process simulation overhead in memory consumption is caused by the required functionality of the process simulation tools. In detail, for deposition and etching at least two grids, the original and the modified grid, are stored during each simulation step. Additional functionality, such as neighborhood information, surface information, an octree for point location and additional attributes have to be stored in the data structures. For performing the interconnect simulation, only

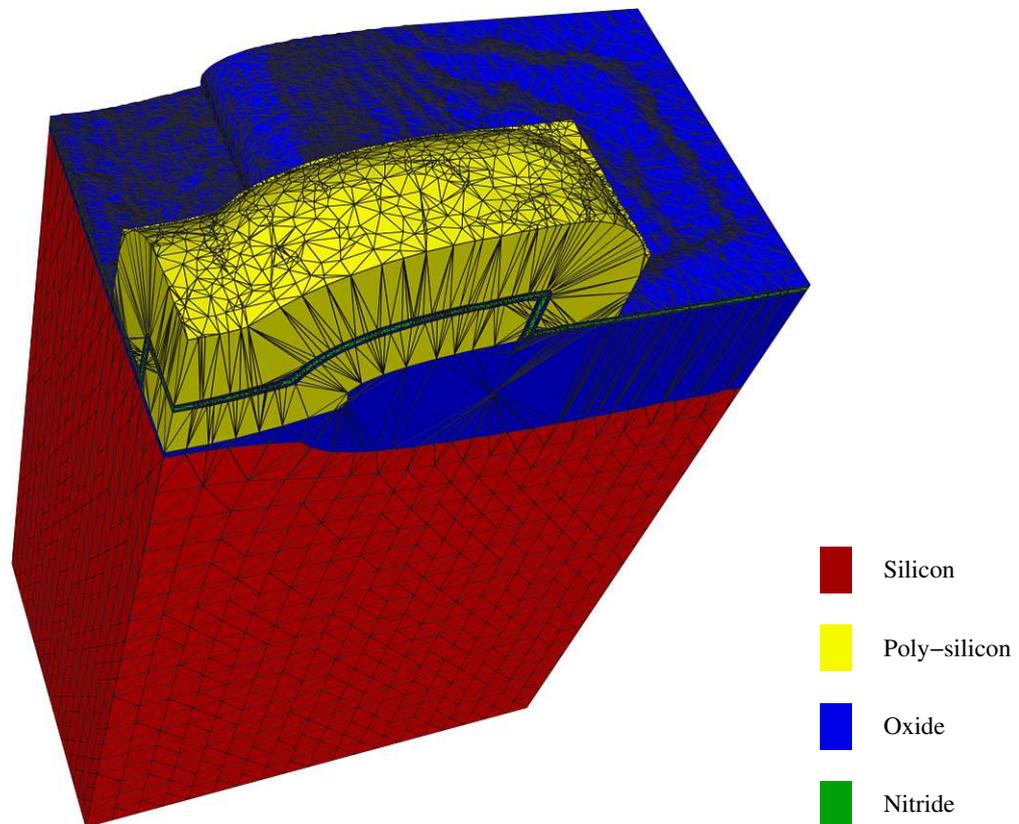


Figure 6.12: Final structure of the EEPROM cell — 76,700 points, 402,000 tetrahedrons.

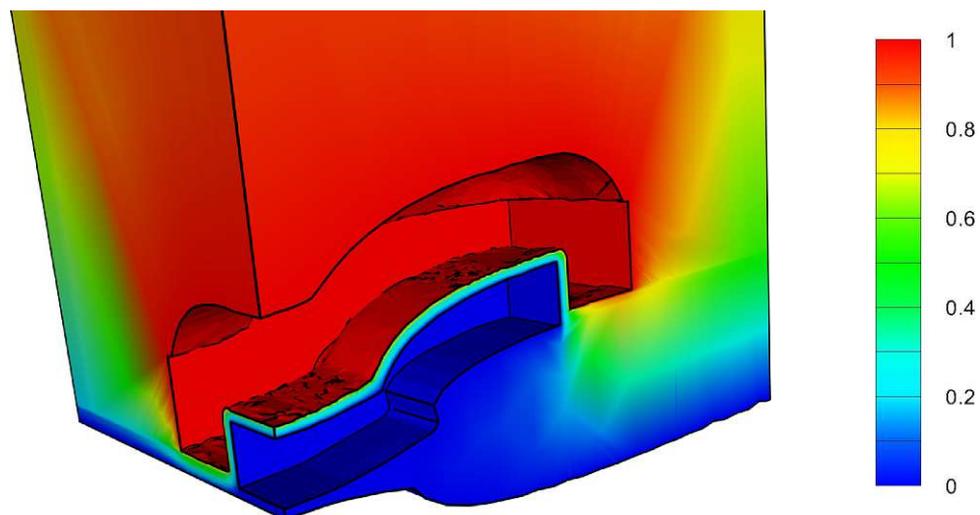


Figure 6.13: Distribution of the electric potential, 0 V at the floating-gate, 1 V at the control-gate — 640,000 points, 3,400,000 tetrahedrons.

$C_{\text{Si-Cg}}$	Capacity silicon – control-gate	$0.82 \times 10^{-16} \text{ F}$
$C_{\text{Si-Fg}}$	Capacity silicon – floating-gate	$1.1 \times 10^{-16} \text{ F}$
$C_{\text{Cg-Fg}}$	Capacity control-gate – floating-gate	$2.3 \times 10^{-16} \text{ F}$

Table 6.2: Results of the capacitance simulation.

the final grid and potential attributes are necessary. For future simulations of more complex three-dimensional structures further work on data reduction and surface smoothing will be necessary.

Chapter 7

Conclusion and Outlook

This work focused on some issues for accurate numerical simulations of electronic devices with the main attention on three-dimensional concepts. Due to the rapidly increasing miniaturization and complexity of the devices, the use of three-dimensional simulations is often inevitable. Starting with the diffusion simulation of dopants, three-dimensional effects, which are often parasitic, cannot be neglected any longer. With the developed tool for diffusion simulation the fast computation of simplified diffusion processes was enabled. On the basis of those, statements about device characteristics can be chosen. Additionally, the three-dimensional diffusion simulation is a fundamental basis for three-dimensional electrical simulations of semiconductors. The second part of this work deals with the simulation meshes which are required for device simulation. The developed method allows to tune the grid density in wide ranges. Due to the almost orthogonal potential distributions, which are the basis for the placement of the grid points, the density of the grids can be independently controlled in three directions of space. Since the equipotential surfaces are conforming to the geometry surfaces, next to the surfaces of the geometry, grid elements with certain preferred directions can be obtained. And as a result of nearly cuboidal grid elements, the Delaunay criterion is almost fulfilled and the following tetrahedrization does not cause difficulties. With this method, the automatic generation of specially suited grids for three-dimensional semiconductor device simulation was enabled.

There still remain unresolved issues in the field of fully automatic grid generation. Especially for three-dimensional grid generation a wide range of questions is left open and needs a lot of further work. Usually a general approach of meshing cannot be given and the development of the grid generators follows different ways.

Several anisotropic grid generation methods are under development, with the main goal to deliver grids with as few as possible grid points, but not neglecting the

necessity of an accurate solution on the based model. That means that different kinds of grids are necessary for process simulation, diffusion simulation, and device simulation.

In summary, further development steps are:

- Surface coarsening algorithms: Often between two kinds of simulation, the surface representation must be adapted to the new demands, for example between a topological and a diffusion simulation. As the physically based addition and deletion of material influences the surface directly and the grid in the inner of the regions is of minor interest, the demands for a diffusion or electrical simulation shift to the interior grid. The surface grid should be coarsened as much as possible, but structural edges must be detected and preserved automatically.
- Implementation of grid refinement strategies: The grids should be adapted to the different discretization methods and different simulation models. Especially edge splitting algorithms which produce anisotropic grid densities must be implemented.
- Development of coarsening algorithms of the volume mesh: On the one hand in combination with refinement strategies, the generation of adaptive meshes for transient simulations must be improved. On the other hand, the matrix sizes can be reduced by coarsening the grid in regions of minor interest.
- Finally the handling of thin material layers must be improved. Such thin segment structures are found frequently within electronic semiconductor device structures, for example the field oxide layer under the polysilicon segment of the gate contact of MOS transistors. For the generation of a Delaunay grid, point insertion on the surface is often necessary. With standard point insertion algorithms a lot of additional grid points are generated on the surface of thin layers. To prevent this effect, a projection of grid points from one to the other thin layer surface gives better results.

For further extensions of the potential method described in this work, investigations on point reduction algorithms are required. As described before, an additional equipotential surface causes additional grid points at all intersections with the other existing potential ticks. Therefore an algorithm, which works similarly to a terminating lines algorithm based on ortho-grids is inevitable. These intersectional points in regions of small interest must be prevented. Additionally, grid points at the surface, which arise from structural edges or are eventually inserted by the grid generator to satisfy the Delaunay criterion, affect the grid quality near the boundaries.

To prevent this usually adverse influence on the grid quality, a continuation of these surface points as potential ticks inside the segment region keeps the orthogonality of the grid lines. In addition to this continuation, the number of grid points inside the structures caused by these potential ticks must be limited by terminating lines algorithms which maintain the Delaunay criterion.

Bibliography

- [1] D. Adalsteinsson and J.A. Sethian, “A Fast Level Set Method for Propagating Interfaces,” *J.Comput.Phys.*, vol. 118, no. 1, pp. 269–277, April 1995.
- [2] O.E. Akcasu, “Convergence Properties of Newton’s Method for the Solution of the Semiconductor Transport Equations and Hybrid Solution Techniques for Multidimensional Simulation of VLSI Devices,” *Solid-State Electron.*, vol. 27, no. 4, pp. 319–328, 1984.
- [3] V. Axelrad, “Grid Quality and Its Influence on Accuracy and Convergence in Device Simulation,” *IEEE Trans.Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 2, pp. 149–157, 1998.
- [4] R.E. Bank and D.J. Rose, “Parameter Selection for Newton-like Methods Applicable to Nonlinear Partial Differential Equations,” *SIAM J.Numer.Anal.*, vol. 17, no. 6, pp. 806–822, 1980.
- [5] D.A. Bell, “Improved Formulation of Gummel’s Algorithm for Solving the 2-Dimensional Current-Flow Equations in Semiconductor Devices,” *Electron.Lett.*, vol. 8 pp. 536–538, 1972.
- [6] T. Binder, J. Cervenka, K. Dragosits, A. Gehring, T. Grasser, M. Gritsch, R. Klima, M. Knaipp, H. Kosina, R. Mlekus, V. Palankovski, R. Rodriguez-Torres, M. Rottinger, G. Schrom, S. Selberherr, M. Stockinger, and S. Wagner, *Minimos-NT, Device and Circuit Simulator, Version 2.0*, Institut für Mikroelektronik, 2002.
- [7] H. Ceric and S. Selberherr, “An Adaptive Grid Approach for the Simulation of Electromigration Induced Void Migration,” *IEICE Trans.Electronics*, vol. E86-C, no. 3, pp. 421–426, 2003.
- [8] T. Chen, D.W. Yergeau, and R.W. Dutton, “A Common Mesh Implementation for Both Static and Moving Boundary Process Simulations,” In Meyer and Biesemans [37], pp. 101–104.

- [9] Y.-K. Choi, T.-J. King, and C. Hu, "Nanoscale CMOS Spacer FinFET for the Terabit Era," *IEEE Electron Device Lett.*, vol. 23, no. 1, pp. 25–27, January 2002.
- [10] H.J. Dirschmid, *Einführung in die mathematischen Methoden der theoretischen Physik*, Vieweg, 1976.
- [11] C. Fischer, P. Habaš, O. Heinreichsberger, H. Kosina, Ph. Lindorfer, P. Pichler, H. Pötzl, C. Sala, A. Schütz, S. Selberherr, M. Stiftinger, and M. Thurner, *MINIMOS 6 User's Guide*, Institut für Mikroelektronik, Technische Universität Wien, Austria, 1994.
- [12] P. Fleischmann, *Mesh Generation for Technology CAD in Three Dimensions*, Dissertation, Technische Universität Wien, 2000 <http://www.iue.tuwien.ac.at/phd/fleischmann>.
- [13] P. Fleischmann, B. Haindl, R. Kosik, and S. Selberherr, "Investigation of a Mesh Criterion for Three-Dimensional Finite Element Diffusion Simulation," In *SISPAD'99* [62], pp. 71–74.
- [14] P. Fleischmann, R. Kosik, B. Haindl, and S. Selberherr, "Simple Mesh Examples to Illustrate Specific Finite Element Mesh Requirements," In *8th International Meshing Roundtable*, pp. 241–246, 1999.
- [15] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr, "Grid Generation for Three-Dimensional Process and Device Simulation," In *Simulation of Semiconductor Processes and Devices*, pp. 161–166, Business Center for Academic Societies Japan, Tokyo, Japan, 1996.
- [16] Ch. Großmann and H. G. Roos, *Numerik partieller Differentialgleichungen*, Teubner Studienbücher Mathematik, Teubner-Verlag, Stuttgart, 1994.
- [17] H.K. Gummel, "A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations," *IEEE Trans. Electron Devices*, vol. ED-11 pp. 455–465, 1964.
- [18] K. Habberger, H. Ryssel, G. Prinke, and K. Hoffmann, "Diffusionsmechanismen in Halbleitern," Techn. Report RY1/6, Institut für Festkörpertechnologie, München, 1979.
- [19] B. Haindl, R. Kosik, P. Fleischmann, and S. Selberherr, "Comparison of Finite Element and Finite Box Discretization for Three-Dimensional Diffusion Modeling Using AMIGOS," In *SISPAD'99* [62], pp. 131–134.
- [20] M.R. Hestenes and E. Stiefel, "Method of Conjugate Gradients for Solving Linear Systems," *J. Res. Nat. Bur. Stand.*, vol. 49 pp. 409–436, 1952.

- [21] N. Hitschfeld, M.C. Rivara, and M. Palma, “Improving the Quality of Delaunay Triangulations for the Control Volume Discretization Method,” In Meyer and Biesemans [37], pp. 189–192.
- [22] H. Hofmann, *Das elektromagnetische Feld*, Springer, Wien New-York, 1974.
- [23] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface Reconstruction from Unorganized Points,” *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.
- [24] A. Hössinger, S. Selberherr, M. Kimura, I. Nomachi, and S. Kusanagi, “Three-Dimensional Monte-Carlo Ion Implantation Simulation for Molecular Ions,” In *Electrochemical Society Proceedings*, vol. 99-2, pp. 18–25, 1999.
- [25] T. Ikeda, *Maximum Principle in Finite Element Models for Convection-Diffusion Phenomena*, North Holland, Amsterdam, 1983.
- [26] J.D. Jackson, *Klassische Elektrodynamik*, John Wiley & Sons, 1999.
- [27] I. Kaufman, “Finite Volume Poisson Solver for Ionic Channels,” Techn. report, Department of Physics, Lancaster University, Lancaster <http://www.lancs.ac.uk/~kaufmani/psn/fvm.pdf>.
- [28] R. Kosik, P. Fleischmann, B. Haindl, P. Pietra, and S. Selberherr, “On the Interplay Between Meshing and Discretization in Three-Dimensional Diffusion Simulation,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 11, pp. 1233–1240, 2000.
- [29] I. Kossacky, “A Recursive Approach to Local Mesh Refinement in Two and Three Dimensions,” *J.Comp.Appl.Math.*, vol. 55 pp. 275–288, 1994.
- [30] C. Ledl, “Konvertierung rasterorientierter Geometrien in polygonal begrenzte,” Diplomarbeit, Technische Universität Wien, 1994.
- [31] E. Leitner and S. Selberherr, “Three-Dimensional Grid Adaptation Using a Mixed-Element Decomposition Method,” In Ryssel and Pichler [53], pp. 464–467.
- [32] K. Lilja, V. Moroz, and D. Wake, “A 3D Mesh Generation Method for the Simulation of Semiconductor Processes and Devices,” In *Proc. International Conference on Modeling and Simulation of Microsystems Semiconductors, Sensors and Actuators*, pp. 334–338, Santa Clara, CA, USA, April 1998.
- [33] J. Litsios, B. Schmithüsen, U. Krumbein, A. Schenk, E. Lyumkis, B. Polsky, and W. Fichtner, *DESSIS 3.0: Manual*, ISE Integrated Systems Engineering, Zürich, Switzerland, 1996 release 3.0 edition.

- [34] W.E. Lorensen and H.E. Cline, “Marching Cubes: A High Resolution 3D Surface Construction Algorithm,” *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [35] P.A. Markowich, C.A. Ringhofer, S. Selberherr, and M. Lentini, “A Singular Perturbation Approach for the Analysis of the Fundamental Semiconductor Equations,” *IEEE Trans. Electron Devices*, vol. ED-30, no. 9, pp. 1165–1180, 1983.
- [36] C.W. Mastin and J.F. Thompson, “Quasiconformal Mappings and Grid Generation,” *SIAM J.Sci.Stat.Comput.*, vol. 5, no. 2, pp. 305–316, 1984.
- [37] K. De Meyer and S. Biesemans, editors, *Simulation of Semiconductor Processes and Devices*, Springer, Wien, New York, 1998.
- [38] R. Mlekus, Ch. Ledl, E. Strasser, and S. Selberherr, “Polygonal Geometry Reconstruction after Cellular Etching or Deposition Simulation,” In Ryssel and Pichler [53], pp. 50–53.
- [39] N. Molino, R. Bridson, J. Teran, and R. Fedkiw, “A Crystalline, Red Green Strategy for Meshing Highly Deformable Objects with tetrahedra,” In *12th International Meshing Roundtable*, pp. 103–114. Sandia National Laboratories, 2003.
- [40] Y. Nishi and R. Doering, *Handbook of Semiconductor Manufacturing Technology*, Marcel Dekker AG, 2000.
- [41] J.T. Oden and J.N. Reddy, *An Introduction to the Mathematical Theory of Finite Elements*, Wiley, 1976.
- [42] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations – Concepts and Applications of Voronoi Diagrams*, Wiley, 1992.
- [43] C. Pao, *Nonlinear Parabolic and Elliptic Equations*, Plenum, New York, 1992.
- [44] A.L. Pardhanani and G.F. Carey, “A Mapped Scharfetter-Gummel Formulation for the Efficient Simulation of Semiconductor Device Models,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 10, pp. 1227–1233, 1997.
- [45] M.H. Protter and H.F. Weinberger, *Maximum Principles in Differential Equations*, Springer, 1999.
- [46] M. Putti and Ch. Cordes, “Finite Element Approximation of the Diffusion Operator on Tetrahedra,” *SIAM J.Sci.Comput.*, vol. 19, no. 4, pp. 1154–1168, 1998.

- [47] M. Radi, *Three-Dimensional Simulation of Thermal Oxidation*, Dissertation, Technische Universität Wien, 1998 <http://www.iue.tuwien.ac.at/phd/radi>.
- [48] M. Radi, E. Leitner, E. Hollensteiner, and S. Selberherr, “Analytical Partial Differential Equation Modeling Using AMIGOS,” In *Proc. IASTED International Conference Artificial Intelligence and Soft Computing*, pp. 423–426, Banff, Canada, July 1997.
- [49] J.K. Reid, “On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations,” In J.K. Reid, editor, *Large Sparse Sets of Linear Equations*. Academic Press, 1971.
- [50] M.C. Rivara, “Mesh Refinement Processes Based on the Generalized Bisection of Simplices,” *SIAM J.Numer.Anal.*, vol. 21, no. 3, pp. 604–613, 1984.
- [51] M.C. Rivara and P. Inostroza, “A Discussion on Mixed (Longest Side Midpoint Insertion) Delaunay Techniques for the Triangulation Refinement Problem,” In *4th International Meshing Roundtable*, pp. 335–346, Sandia National Labs., Albuquerque, New Mexico, 1995.
- [52] H. Ryssel, “Implantation and Diffusion Models for Process Simulation,” In K.M. DeMeyer, editor, *VLSI Process and Device Modeling*, pp. 1–41. Katholieke Universiteit Leuven, 1983.
- [53] H. Ryssel and P. Pichler, editors, *Simulation of Semiconductor Devices and Processes*, vol. 6, Springer, Wien, Austria, 1995.
- [54] R. Sabelka, *Dreidimensionale Finite Elemente Simulation von Verdrahtungsstrukturen auf Integrierten Schaltungen*, Dissertation, Technische Universität Wien, 2001 <http://www.iue.tuwien.ac.at/phd/sabelka>.
- [55] R. Sabelka, R. Martins, and S. Selberherr, “Accurate Layout-Based Interconnect Analysis,” In Meyer and Biesemans [37], pp. 336–339.
- [56] R. Sabelka and S. Selberherr, “SAP — A Program Package for Three-Dimensional Interconnect Simulation,” In *Proc. Intl. Interconnect Technology Conference*, pp. 250–252, Burlingame, California, June 1998.
- [57] D.L. Scharfetter and H.K. Gummel, “Large-Signal Analysis of a Silicon Read Diode Oscillator,” *IEEE Trans.Electron Devices*, vol. ED-16, no. 1, pp. 64–77, 1969.
- [58] W.J. Schroeder, J.A. Zarge, and W.E. Lorenson, “Decimation of Triangle Meshes,” *Computer Graphics*, vol. 26, no. 2, pp. 65–70, 1992.
- [59] G. Schumicki and P. Seegebrecht, *Prozeßtechnologie*, Springer, 1991.

- [60] A.H. Sherman, “On Newton-Iterative Methods for the Solution of Systems of Nonlinear Equations,” *SIAM J.Numer.Anal.*, vol. 15 pp. 755–771, 1978.
- [61] Jonathan Richard Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator,” In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of *Lecture Notes in Computer Science*, , pp. 203–222, Springer-Verlag, May 1996 From the First ACM Workshop on Applied Computational Geometry.
- [62] *Proc. Simulation of Semiconductor Processes and Devices*, Kyoto, Japan, September 1999.
- [63] P. Stein and R.L. Rosenberg, “On the Solution of Linear Simultaneous Equations by Iteration,” *J.London Math.Soc.*, vol. 23 pp. 111–118, 1948.
- [64] J.L. Stone and J.C. Plunkett, “Ion Implantation Processes in Silicon,” In F.F.Y. Wang, editor, *Impurity Doping Processes in Silicon*, pp. 56–146, North-Holland, Amsterdam, 1981.
- [65] N. Strecker, *ISE TCAD Manuals vol. 5*, ISE Integrated Systems Engineering, Zürich, Switzerland, 1997 release 4.
- [66] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Chapman & Hall, New York, 1989.
- [67] A.D. Sutherland, “On the Use of Overrelaxation in Conjunction with Gummel’s Algorithm to Speed the Convergence in a Two-Dimensional Computer Model for MOSFET’s,” *IEEE Trans.Electron Devices*, vol. ED-27 pp. 1297–1298, 1980.
- [68] Technology Modeling Associates, Inc., Sunnyvale, California *TMA Medici, Two-Dimensional Device Simulation Program, Version 4.0 User’s Manual*, October 1997.
- [69] N.D. Thai, “Concentration Dependent Diffusion of Boron and Phosphorus in Silicon,” *J.Appl.Phys.*, vol. 41, no. 7, pp. 2859–2866, 1970.
- [70] J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin, *Numerical Grid Generation*, North Holland, 1985.
- [71] University of Nijmegen, NL *Short Course on Preconditioned Conjugate Gradient Methods*, 1989.
- [72] W. Wessner, C. Heitzinger, A. Hössinger, and S. Selberherr, “Error Estimated Driven Anisotropic Mesh Refinement for Three-Dimensional Diffusion Simulation,” In *Simulation of Semiconductor Processes and Devices*, pp. 109 – 112, Boston,USA, September 2003.

- [73] W. Wessner, C. Hollauer, A. Hssinger, and S. Selberherr, “Anisotropic Laplace Refinement for Three-Dimensional Oxidation Simulation,” In *Simulation of Semiconductor Processes and Devices*, Munich, Germany, September 2004.
- [74] R.E. White, *An Introduction to the Finite Element Method with Applications to Nonlinear Problems*, John Wiley & Sons, New York, 1985.
- [75] J. Xu and L. Zikatanov, “A Monotone Finite Element Scheme for Convection-Diffusion Equations,” *Mathematics of Computation*, vol. 68, no. 228, pp. 1429–1446, October 1999.

List of Figures

1.1	The simulation domain of the coaxial capacitor.	4
1.2	Unstructured grid refinement, 237 grid points. The resulting maximal discretization error.	5
1.3	Refinement, which is adapted to the given problem, 17×14 grid points. The maximal discretization error.	6
1.4	Error of the adapted grid. Each curve results from a grid with n ticks in radial direction.	7
1.5	Errors of the unstructured versus the adapted grid. The unstructured grid is refined by n ticks.	8
2.1	Two-dimensional ortho grids along axes-parallel and non-parallel geometry edges.	10
2.2	A two-dimensional mixed grid with rectangular and triangular elements to approximate a circular geometry.	11
2.3	A two-dimensional Voronoi box of the point p_i	17
2.4	A detail of the Voronoi Region shown before.	18
2.5	Valid and invalid tessellations of two triangles.	19
2.6	Valid and invalid Voronoi edges at the surface. At the marginal case, the center of the circle is on the surface.	20
2.7	The coupling areas of a three-dimensional tetrahedral Delaunay grid around the point p_i	21
2.8	The coupling areas between two points of different tetrahedrons.	22
2.9	Valid and invalid Voronoi faces at the surface. At the marginal case, the center of the circle is on the surface.	23
2.10	Three-dimensional grid criterion for Finite Elements.	25

2.11	Two-dimensional mesh criterion for Finite Elements	26
2.12	One-dimensional diffusion, diffusion constant D , $d = 1.5 k$, no out-diffusion across $z = 0$	
2.13	Red-refinement of a triangle.	29
2.14	A red-refined triangle with a red-refined neighbor.	29
2.15	A red-refined triangle with green-refined neighbors.	30
2.16	Red-refinement of a tetrahedron into children, 4 tetrahedrons and an octahedron.	31
2.17	Final decomposition of the octahedron into 4 tetrahedrons.	31
2.18	Subsequent red-refinement of the octahedron into 6 parent-similar octahedrons and 8 gr	
2.19	Possible green neighbors of a red-refined triangle.	32
2.20	Two-dimensional grid refinement by line splitting.	34
2.21	Overview of the marching cubes algorithm [30].	36
3.1	Tetrahedrons around point p_i with the resulting Voronoi box constructed by these tetra	
3.2	Box parts of the tetrahedron $\langle p_i p_j p_k p_l \rangle$ with drawn outer-sphere s_{ijkl} of the tetrahedron	
3.3	Different Voronoi faces which share the point p_i of a tetrahedron. . .	41
3.4	A detail of the tetrahedron $\langle p_i p_j p_k p_l \rangle$ shown in Figure 3.3 which shows the coupling Vor	
3.5	Potential distribution between two grid points p_i and p_j with different permittivities. 4	
3.6	Ratio of the approximated electric field E_{ij} to the exact solution $\tilde{E}(d_{ij}/2)$ at the center	
3.7	A Voronoi box of the point p_i at a boundary.	49
4.1	Netto doping of the semiconductor segment.	63
4.2	Well mask layout of the high voltage device.	64

- 4.3 Two-dimensional simulation of the drain-current far away from the tip of the finger. 65
- 4.4 Relevant iso-surfaces of the phosphorus doping and the pn-junctions of the finger. The
- 4.5 Surfaces surrounding the space charge regions between both pn-junctions. 68
- 4.6 Simulation of the original finger structure compared to the measurement of four test wa
- 5.1 Structure of a two-dimensional MOS transistor. 71
- 5.2 The carrier concentration of the two-dimensional device simulation with a gate-source a
- 5.3 The current density of the two-dimensional device simulation at $V_{DS} = 1\text{ V}$. 74
- 5.4 Comparison of the simulation approaches. The first two-dimensional simulation is perfo
- 5.5 Refinement of a two-dimensional triangular grid. 76
- 5.6 Two-dimensional capacitor with the electrodes and drawn field and equipotential lines.
- 5.7 Capacitor with electrodes at top and bottom. 80
- 5.8 Cuboidal capacitor, original electrodes, potential distribution. 82
- 5.9 Cuboidal capacitor, first electrode-replacement, potential distribution. 82
- 5.10 Cuboidal capacitor, second electrode-replacement, potential distribution. 83
- 5.11 Cuboidal capacitor, with the equipotential surfaces of the three electric fields that are s
- 5.12 Domain for the electric field calculation. 85
- 5.13 Spherical capacitor with electrodes placed at the top and bottom (hidden) cups. The el
- 5.14 Potential and field lines of the spherical capacitor. 93
- 5.15 Three-dimensional structure with the placed electrodes. 100
- 5.16 Three-dimensional structure with a potential distribution and shown base grid, on whic
- 5.17 Final grid, developed by the potential method with shown potential distribution, contac

- 5.18 Final grid, a quarter of the device is cut away for illustration purposes. 102
- 6.1 Geometry of the simulated FinFET structure. 104
- 6.2 The oxide and silicon layers of the FinFET structure. The simulation grid is an ortho g
- 6.3 The oxide and silicon layers of the FinFET. The simulation grid is generated by the pot
- 6.4 Generated ortho grid in the active regions of the transistor. Only the oxide, silicon and
- 6.5 Potential based tetrahedral grid in the active regions of the transistor. Only the oxide, s
- 6.6 Simulated output characteristic of the FinFET. The drain current I_D is standardized by
- 6.7 Contour lines of the electron current density in the channel area of the fin at $V_{GS} = V_{DS}$
- 6.8 The cross section and typical layer thicknesses of the memory cell. . . 109
- 6.9 Expanded two-dimensional structure — 8,800 points, 51,000 tetrahedrons. 110
- 6.10 Floating-gate and first oxide layer — 18,300 points, 69,000 tetrahedrons. 111
- 6.11 ONO layers added — 70,300 points, 341,000 tetrahedrons. 112
- 6.12 Final structure of the EEPROM cell — 76,700 points, 402,000 tetrahedrons. 113
- 6.13 Distribution of the electric potential, 0 V at the floating-gate, 1 V at the control-gate —

List of Tables

5.1	Differential equations and boundary conditions of the capacitor and its dual capacitors.	
6.1	Comparison of the different grids. Rank describes the rank of the system matrix of the	
6.2	Results of the capacitance simulation.	114

List of Publications

- [1] J. Cervenka, M. Knaipp, A. Hössinger, and S. Selberherr, “Green’s Function Approach for Three-Dimensional Diffusion Simulation of Industrial High-Voltage Applications,” In *Proceedings SISPAD 2001*, pp. 408–411, Athens, Greece, September 2001.
- [2] J. Cervenka, P. Fleischmann, S. Selberherr, M. Knaipp, and F. Unterleitner, “Optimization of Industrial High Voltage Structures by Three-Dimensional Diffusion Simulation,” In *Proceedings ESSDERC 2001*, pp. 227–230, Nuremberg, Germany, 2001.
- [3] V. Palankovski, G. Röhrer, E. Wachmann, J. Kraft, B. Löffler, J. Cervenka, R. Quay, T. Grasser, and S. Selberherr, “Optimization of High-Speed SiGe HBTs,” In *Proceedings Intl. Symposium on Electron Devices for Microwave and Optoelectronic Applications*, pp. 187–191, Vienna, Austria, 2001.
- [4] J. Cervenka, R. Klima, M. Knaipp, and S. Selberherr, “Three-Dimensional Device Optimization by Green’s Functions,” *European Physical Journal - Applied Physics*, vol. 21 pp. 103–106, 2003.
- [5] A. Hössinger, J. Cervenka, and S. Selberherr, “A Multistage Smoothing Algorithm for Coupling Cellular and Polygonal Datastructures,” In *2003 Intl. Conf. on Simulation of Semiconductor Processes and Devices*, pp. 259–262, Boston, USA, 2003.
- [6] J. Cervenka, A. Hössinger, R. Minixhofer, T. Grasser, and S. Selberherr, “Drei-dimensionale Modellierung Elektronischer Bauelemente,” In *Beiträge der Informationstagung Mikroelektronik 2003*, pp. 377–382, Vienna, Austria, 2003.

Curriculum Vitae

- December 11th 1968 Born in Schwarzach, Austria
- 1975 – 1979 Primary school, Vienna, Austria
- 1979 – 1983 Secondary school, Vienna, Austria
- 1983 – 1988 Technical college, Vienna, Austria
- June 1988 Final examination
- 1988 – 1993 Vienna University of Technology, Austria, “1. Diplomprüfung” in Electrical Engineering
- 1993 – 1999 Vienna University of Technology, Austria, specialization in Industrial Electronics and Control Engineering
- 1996 – 1997 Community Service
- October 1999 Masters Degree, “Diplomingenieur” in Electrical Engineering
- 1999 – 2002 Vienna University of Technology, Austria. Enrolled in a full Ph.D. program at the Christian Doppler Laboratory at the Institute for Microelectronics under the supervision of Prof. Siegfried Selberherr.
- 2003 – 2004 Vienna University of Technology, Austria. Enrolled in a full Ph.D. program at the Christian Doppler Laboratory at the Institute for Microelectronics under the supervision of Prof. Tibor Grasser.