Bachelor Thesis

Topology Optimization of Metalization Grid Patterns to Improve the Power Conversion Efficiency of Thin-Film Solar Cells

Benedikt Braun
Matriculation Number 3413157

02.12.2021

Supervisors:
Prof. Dr. rer. nat. Dominik Göddeke
Prof. Dr.-Ing. Michael Powalla
M. Sc. Mario Zinßer

Institute of Applied Analysis and Numerical Simulation,
University of Stuttgart

Photovoltaics: Materials Research,
Center for Solar Energy and Hydrogen Research Baden-Württemberg

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Die Arbeit oder wesentliche Bestandteile davon sind weder an dieser noch an einer anderen Bildungseinrichtung bereits zur Erlangung eines Abschlusses eingereicht worden. Das elektronische Exemplar stimmt mit den anderen Exemplaren überein.

Ich erkläre weiterhin, bei der Erstellung der Arbeit die einschlägigen Bestimmungen zum Urheberschutz fremder Beiträge entsprechend den Regeln guter wissenschaftlicher Praxis eingehalten zu haben. Soweit meine Arbeit fremde Beiträge (z.B. Bilder, Zeichnungen, Textpassagen etc.) enthält, habe ich diese Beiträge als solche gekennzeichnet (Zitat, Quellenangabe) und eventuell erforderlich gewordene Zustimmungen der Urheber zur Nutzung dieser Beiträge in meiner Arbeit eingeholt. Mir ist bekannt, dass ich im Falle einer schuldhaften Verletzung dieser Pflichten die daraus entstehenden Konsequenzen zu tragen habe.

Statement of Originality

This thesis has been performed independently. It contains no material that has been accepted for the award of a degree at this or any other university. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text.

I further declare that I have performed this thesis according to the existing copyright policy and the rules of good scientific practice. In case this works contains contributions of someone else (e.g. pictures, drawings, text passages etc.), I have clearly identified the source of these contributions, and, if necessary, have obtained approval from the originator for making use of them in this thesis. I am aware that I have to bear the consequences in case I have contravened these duties.

B. Braun

....

Ort, Datum, Unterschrift

Deutsche Zusammenfassung

Der metallische Leiter, welcher in Form eines Gitters auf der Oberfläche einer Solarzelle angebracht ist, heißt Grid. Die Funktion dieses Grids ist es, den in der Absorberschicht einer Solarzelle erzeugten Strom, ohne große Verluste, an der Oberfläche zum externen Abgreifpunkt zu leiten.

Durch die sehr gute Leitfähigkeit des Grids wird ein verlustarmer Ladungstransport ermöglicht. Allerdings bewirkt das für Lichtstrahlen undurchdringbare Grid eine Abschattung der Absoberschicht und verhindert, dass an dieser Stelle Strom erzeugt werden kann. Wenn kein Grid angebracht ist, fließt die Ladung durch die oberste Schicht einer Solarzelle. Diese besteht aus transparenten leitfähigen Oxiden (engl. transparent conducting oxides (TCO)). Das TCO lässt Lichtstrahlen durch und dadurch kann Strom erzeugt werden. Obwohl die Schicht den Strom leiten kann, besitzt sie denoch einen sehr hohen elektrischen Widerstand. Das bedeutet, eine geeignete Wahl des Gridmusters verschattet möglichst wenig Fläche der Solarzelle und bietet trotzdem einen flächendeckenden, verlustarmen Ladungsabtransport. Ein Gridmuster, welches beide Anforderungen bestens erfüllt, soll in dieser Bachelorarbeit mithilfe von Topologie-Optimierung gefunden werden.

Topologie-Optimierung ist eine mathematische Optimierungsmethode, mit der, innerhalb eines Gebietes, eine optimale Materialverteilung gefunden werden kann, um eine hohe, strukturbedingte Leistung zu erzielen. Im Zuge dieser Arbeit ist dieses Gebiet die Oberfläche einer Solarzelle und das Material, welches auf der Oberfläche verteilt werden soll, ist das Metall, welches das Gridmuster bildet. Die Leistung einer Solarzelle wird mit dem Wirkungsgrad angegeben. Der Wirkungsgrad ist die Effizienz, mit der die Solarenergie in elektrische Energie umgewandelt werden kann.

Zur Berechnung des Wirkungsgrades wird das Gebiet mit einem Voronoi-Diagramm in Simplizes unterteilt. Basierend auf der Poisson-Gleichung für elektrische Leitfähigkeit, kann die Ladung, die durch ein Simplex fließt, mit einer Finite-Elemente-Methode berechnet werden. Aus den einzelnen generierten Strömen lässt sich ein Gesamtstrom berechnen, mit welchem die erzeugte, elektrische Energie berechnet werden kann.

Der einzige Parameter, welcher zur Berechnung der Effizienz einer Solarzelle benötigt und in dieser Arbeit variiert wird, ist das Gridmuster. Die Komponenten des

Dichtevektors geben dabei die Metalldichte eines jeden Simplexes an. Zur Optimierung dieses Dichtevektors werden in dieser Arbeit Optimierungsverfahren verglichen, die in Richtung des steilsten Abstiegs optimieren. Mit einem dieser Verfahren werden weitere Modifizierungen des Dichtevektors getestet.

Eine der Modifizierungen betrifft dabei die Umgebung des externen Abgreifpunktes. Die aufgebrachte Gridfläche muss an dieser Stelle groß genug sein, damit ein externer Kontakt ohne Probleme angebracht werden kann.
Die nächste Modifizierung, die verwendet wird, ist eine Methode zur lokalen Optimierung. Dabei werden die durch die Diskretisierung entstandenen Simplizes zufällig in mehrere lokale Teilgebiete eingeteilt und der Reihe nach optimiert.
Besitzt eine Komponente des Dichtevektors einen Wert von 0 steht dies für kein Grid, während ein Wert von 1 für das vorhanden sein von Grid steht. Die Komponenten des Dichtevektors repräsentieren dabei jeweils ein Simplex und damit eine Teilfläche der Solarzelle. Eine Modifizierung ermöglicht außer den Werten 0 (kein Grid, schlecht leitend, Strom wird erzeugt) und 1 (Grid, gut leitend, kein Strom wird erzeugt) Zwischenwerte. Mit diesen Zwischenwerten kann eine kontinuierliche Optimierung durchgeführt werden. Die Leitfähigkeit bzw. die Möglichkeit Strom zu generieren, wird dabei für Zwischenwerte interpoliert. Je nach Wahl der Interpolationsfunktion, kann der Wert der Leitfähigkeit für Zwischenwerte gut oder schlecht sein. Ebenso für die Menge an generiertem Strom. Sowohl niedrige als auch hohe Werte kommen mit Vorteilen, weshalb eine geschickte Kombination zu einem verbesserten Optimierungsverhalten führen kann.
Die letzte Modifizierung, die eine Rolle spielt, ist das Gridmuster, von welchem ausgehend optimiert wird. Dabei wird, unter anderem, das im Labor vom Zentrum für Sonnenenergie- und Wasserstoffforschung Baden-Württemberg (ZSW) verwendete Gridmuster optimiert.

Das Ziel dieser Arbeit ist es, mit den kombinierten Methoden und den Ergebnissen der damit durchgeführten Optimierungen ein neues Gridmuster zu konstruieren, welches dem bisher verwendeten Gridmuster überlegen ist.

Abstract

The metallic conductor, which is attached on the surface of a solar cell, is called grid. The function of this grid is to conduct the current generated in the absorber layer of a solar cell to the external front contact at the surface without large losses.

The very good conductivity of the grid enables low-loss charge transport. However, the grid, which is impenetrable to light rays, causes shading on the absorber layer and thus prevents current from being generated. When no grid is in place, charge flows through the top layer of a solar cell. The top layer consists of transparent conducting oxides (TCO). The TCO allows light rays to pass through and current can be generated. Although the layer can conduct electricity, the layer has a very high electrical resistance. Meaning that an appropriate choice of grid pattern will shade as little area of the solar cell as possible while still providing area-wide, low-loss charge dissipation. A grid pattern, which fulfills both requirements in the best possible way, is to be found in this bachelor thesis. For this purpose, topology optimization is used.

Topology optimization is a mathematical optimization method, to find an optimal distribution of material within a specific domain in order to achieve a high, structure-dependent performance. Within the framework of this thesis, the domain is the surface of a solar cell and the material to be distributed on the surface is the metal that forms the grid pattern. The performance of a solar cell is given by efficiency. The Power Conversion Efficiency (PCE) is the efficiency with which solar energy can be converted into electrical energy.

To calculate the efficiency, the domain is divided into simplices using a Voronoi diagram. Based on the Poisson equation for electrical conductivity, the charge flowing through a simplex can be calculated using a finite element method. With the individual currents generated, a total current can be calculated, which can be used to calculate the electrical power generated.

Within this thesis, the only changeable parameter that is required for the calculation of the solar cell efficiency is the grid pattern. The components of the density vector indicate the metal density of each simplex. To optimize this density vector, optimization methods are compared, which optimize in the direction of the steepest descent. With one of these methods further modifications of the density vector are tested.

One of the modifications concerns the surroundings of the external front contact. The applied grid area must be large enough at this point to apply an external contact without occuring problems.

The next modification used is a method for local optimization. With this method, the simplices created by the discretization are randomly divided into several local subareas and optimized one by one.

If the components of the density vector has a value of 0 it stands for no grid, while a value of 1 stands for the presence of grid. The components of the density vector each represent a simplex and thus a subarea of the solar cell. A modification allows intermediate values besides 0 (no grid, poorly conducting, current is generated) and 1 (grid, well conducting, no current is generated). With these intermediate values, continuous optimization can be performed. The conductivity or the possibility to generate current is interpolated for intermediate values. Depending on the choice of the interpolation function, the value of the conductivity for intermediate values can be good or bad. Likewise for the amount of generated current. Both good and bad values come with advantages, so a clever combination can lead to improved optimization behavior.

The last modification that plays a role is the grid pattern to start the optimization from. Among others, the grid pattern used by the Center for Solar Energy and Hydrogen Research Baden-Württemberg (ZSW) is optimized.

The aim of this work is to construct a new grid pattern with the combined methods and the results of the optimizations carried out with it, which is superior to the previously used grid pattern.

Contents

1 Introduction

The first signs of global warming were already noted around the middle of the 20th century. It has not been excluded that human civilization is responsible for it. For this reason, the Intergovernmental Panel on Climate Change (IPCC) was founded in 1988. The mission of the institution is to investigate the dangers of climate change and to prepare appropriate responses. The results of the investigations form the basis for science-based political decisions, while the IPCC does not provide concrete proposals for solutions or recommendations for action. Only options for action and their implications are identified [1].

Photovoltaics is one of the fields that the IPCC is investigating. A model was developed to predict how much energy can be generated annually by photovoltaics. According to this model, photovoltaics will provide $12.5 \cdot 10^{12}$ kilowatt hours per year worldwide in 2050. The ICPP estimates that the share of solar energy in 2050 will be 5 to 17 percent. A study by the German Mercator Research Institute on Global Commons and Climate Change (MCC) found that the IPCC significantly underestimated the growth of photovoltaics. The authors of the study calculated a share of 30 to 50 percent [2].

At the Center for Solar Energy and Hydrogen Research Baden-Württemberg (ZSW), research is carried out to improve the efficiency of photovoltaics. In order to reduce resource consumption and costs of laboratory experiments, a simulation of the efficiency of a solar cell was created in collaboration of Mario Zinßer at ZSW [3]. With the simulation it is possible to consider the optical and electrical properties of a solar cell reliably. According to [4], the calculations are accurate and so individual components of a solar cell can be optimized independently with little time required to obtain more efficient solar cell designs.

One of the components that make up a solar cell is the metallic conductor on the top of a solar cell. This conductor is the metalized grid and the whole structure is called a grid pattern. Thus, the problem that is addressed in this thesis can be formulated as follows:

How can an optimal metalization grid pattern of a thin film solar cell be found that maximizes the power conversion efficiency for a given set of external parameters?

To make the mathematical part of the thesis more accessible, the rough structure of a solar cell is described at the beginning, as well as its functionality. Based on this, the model for calculating the power conversion efficiency η is established within the framework of a topology optimization. The model for calculating the efficiency is based on the results of Mario Zinßer [4] and were extended together.

Subsequently, an optimizer is determined which is able to adjust the grid pattern to maximize the power conversion efficiency η. For the selection of the optimizer, any grid pattern may be created without restriction to a practicable result. Only in later optimizations, the resulting grid pattern must meet certain requirements. To achieve this, filters are used in addition to a contact pad to reduce inaccuracies within the grid pattern. The contact pad is an area of the solar cell which is coated with metal in order to allow the connection of an external contact. Since adding the contact pad results in limitations of the optimization behavior, a filter method is presented to overcome these limitations.

Through an interpolation it is possible that the question whether an area on the solar cell should get grid can not only be answered with 'yes' or 'no'. The effects that such an interpolation has on the optimization behavior are investigated and then used to obtain the best possible optimization result.

Each optimization starts at a given design and improves it. In the beginning, therefore, a simple grid pattern is sought, which already provides a good starting position for the optimization.

Finally, at the end of the thesis, the individual methods studied are to be combined. The optimization that takes place ends up, in the best case, in a grid pattern that leads to a higher power conversion efficiency η than the grid pattern, Labgrid, currently in use.

1.1 Structure of a Solar Cell

In the past decades, many different solar cells have been developed. They can be classified based on the semiconductor material and the cell technology used to construct the solar cells.
Generally, solar cells can be categorized into two cell technologies. The first generation cells, called wafer-based solar cells, are made of crystalline silicon. The second generation cells are thin-film solar cells. In this thesis, the latter are discussed.
Solar cells are often named after the absorbing semiconductor material used in the solar cell. Some typical semiconductors of thin film solar cells are CdTe, $CuIn_{1-x}Ga_xSe_2$ (CIGS) and metal halide perovskite (PSC).

A solar cell comprises several layers with different properties. As an example, the structure of a CIGS solar cell is shown in Figure 1.1 and CIGS is the semiconductor used in this thesis.

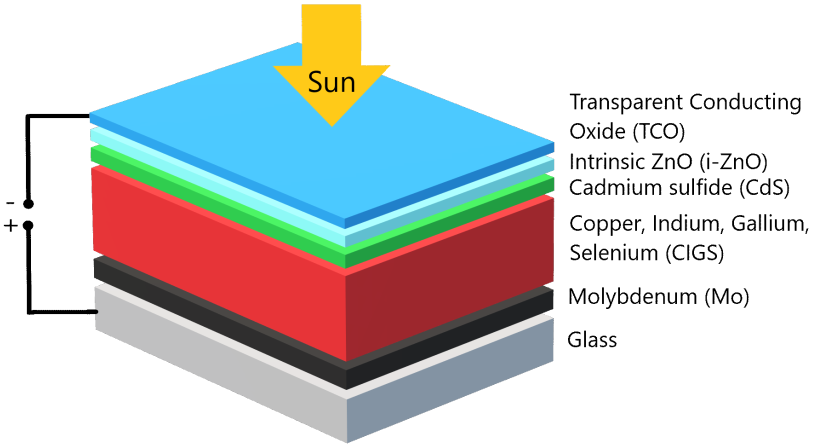

Sun
Transparent Conducting Oxide (TCO)
Intrinsic ZnO (i-ZnO)
Cadmium sulfide (CdS)
Copper, Indium, Gallium, Selenium (CIGS)
Molybdenum (Mo)
Glass
-
+

Figure 1.1: Layer model of solar cell, inspired by [5].

The upper layers are very thin and therefore, they require mechanical support. To provide the necessary support the bottom of a solar cell is often made of glass. Alternatively, flexible metal or polymer foil can be used. A back contact is placed on top of the glass pane. The back contact is a metal layer which has to meet certain material requirements and is connected to the positive pole of an electric circuit. For CIGS, molybdenum as a back contact leads to the best results [5].

Using vacuum-based processes like sputtering or thermal evaporation or using liquid-based processes such as a chemical bath or an ink-like coating procedure, different layer stacks are coated on the substrate [5]. One of these layers is a semiconductor, also called absorber, which absorbs the incoming photons, whose energy is higher than the band gap, see Section 1.2. The thickness of the layer is typically less than $3\,\mu$m [5]. The absorber releases electrons and thus, generates current.

Between the top layer and the semiconductor is a buffer layer. The role of the thin buffer layer is still under discussion, but the high efficiencies that can be achieved with it are undeniable. However, the buffer layer is not necessary for high efficiencies [6]. If a buffer layer is present, it creates a uniform coverage and protects the absorber from damage in the manufacturing process.

On top of the solar cell is the external front contact which is connected to the negative pole of the electric circuit. This completes the electric circuit through the solar cell. That implies that the top layer has to conduct the current but also has to be transparent, so the sunlight can reach the absorber. This is why transparent conducting oxides (TCOs) are used.

To allow more sunlight to pass through the TCO, the thickness of the TCO can be reduced. As a result, the sheet resistance of the TCO increases [7]. To minimize the ohmic power loss [8], a metallized grid is added. After adding a grid, the generated current is lower because the sunlight is blocked in a few places. The problem that is addressed in this thesis is how to optimize the trade-off between generating current and ohmic power loss, i.e., improve the efficiency of a solar cell with the placement of such a grid without changing the structure of the whole cell. Figure 1.2 shows an example of the overall structure of a solar cell.

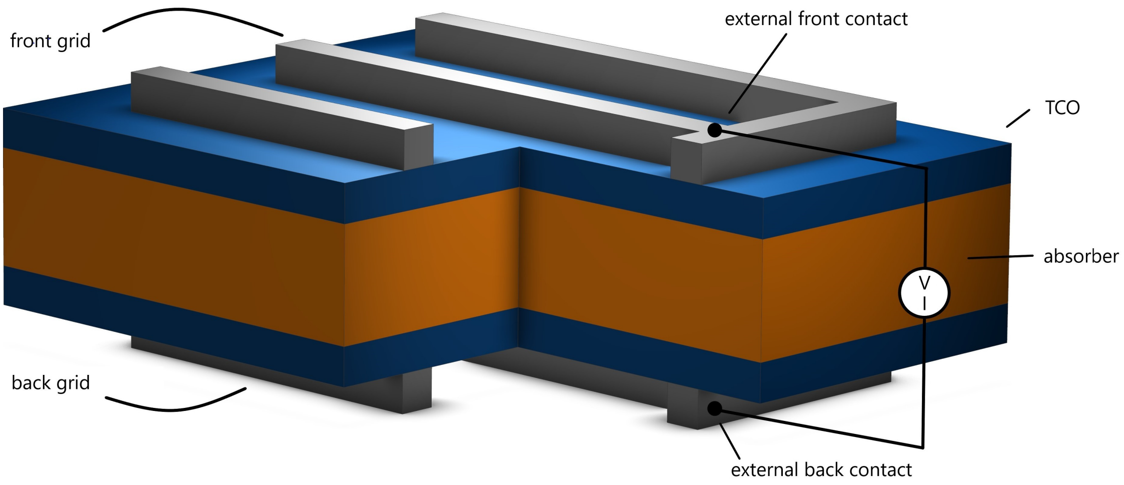


Figure 1.2: Front grid and external front contact, adapted from Mario Zinßer [4].

1.2 Mechanism of a Solar Cell

A solar cell generates current $\mathrm{i}^{\mathrm{total}}$ when sunlight shines on it. The operating voltage V_{op} of the solar cell is determined by the generated current and the connected load resistance. The fraction of incident power of the sunlight converted into electricity defines the power conversion efficiency (PCE) η [9]. This is the overall efficiency of a solar cell, which depends on the optical and electrical properties of the cell. The greater the power conversion efficiency η, the better the solar energy can be converted into electric power. The quality of a solar cell is measured by this efficiency. Moreover, in the context of this thesis, the power conversion efficiency η is considered to evaluate the grid patterns.

The optical properties determine how much of the sunlight reaching the surface of the solar cell hits the absorber. The sunlight falling on the TCO layer is refracted at the surface. This leads to a loss of solar energy. Unfortunately, the sunlight is refracted each time it reaches a new layer. In the simulation used for this thesis, the optical properties are modeled by a transfer-matrix method [4, 10].

After the sunlight has passed through the upper layers, the photons of the sunlight strike the absorber layer. Now three things can happen:

1. the photon is reflected at the surface of the absorber
2. the photon passes the absorber material (the photon has low energy)
3. the photon is absorbed (the photon has a sufficiently high energy)

While the first scenario depends on the angle at which the photon hits the surface, the other two scenarios can be explained with the band gap value of the semiconducting material. Within the band gap, no electric states exist. In the second case, the photon energy is too low to excite an electron and the photon just passes the absorber. The last case is the most interesting one. The photon has enough energy to excite an electron, i.e., the electron gets shifted from the valence band to the conduction band and an electron-hole pair is generated [11].

However, if the energy of the photon is much higher than the band gap value, the energy difference is converted into heat. This effect is called thermal relaxation. When optimizing the efficiency of a solar cell by modifying the band gap of the absorber material, it is important to allow as many electrons as possible to jump into the conduction band. At the same time, not too much energy should be wasted converting into heat. There are several approaches, e.g., a tandem solar cell [12], which are not dealt with in this thesis.

Assuming no external circuit is connected to the solar cell, the generated electron-hole pair recombines after its specific lifespan. If the anode of an external electric circuit is connected to the external back contact and the cathode to the external front contact, a current can flow between them. The electron flow starting at the absorber layer and going to the TCO layer of the solar cell is called photocurrent $\mathbf{i}^{\mathrm{ph}}$. The current of the solar cell that can be measured is called total generated current $\mathrm{i}^{\mathrm{total}}$.

The voltage is the difference of the front and the back potential. Due to the metallic back contact, a constant potential can be assumed on the back side. The front potential gets larger the further the current is generated from the external front contact as shown in Figure 1.3 (left). This can be explained by Ohm's law and thus, the proportionality between resistance and voltage. Therefore, the voltage at the external front contact is just the operating voltage V_{op}. Adding a metalization grid minimizes the resistances and consequently the voltage drop and as a result the ohmic power loss is reduced, see Figure 1.3 (right).

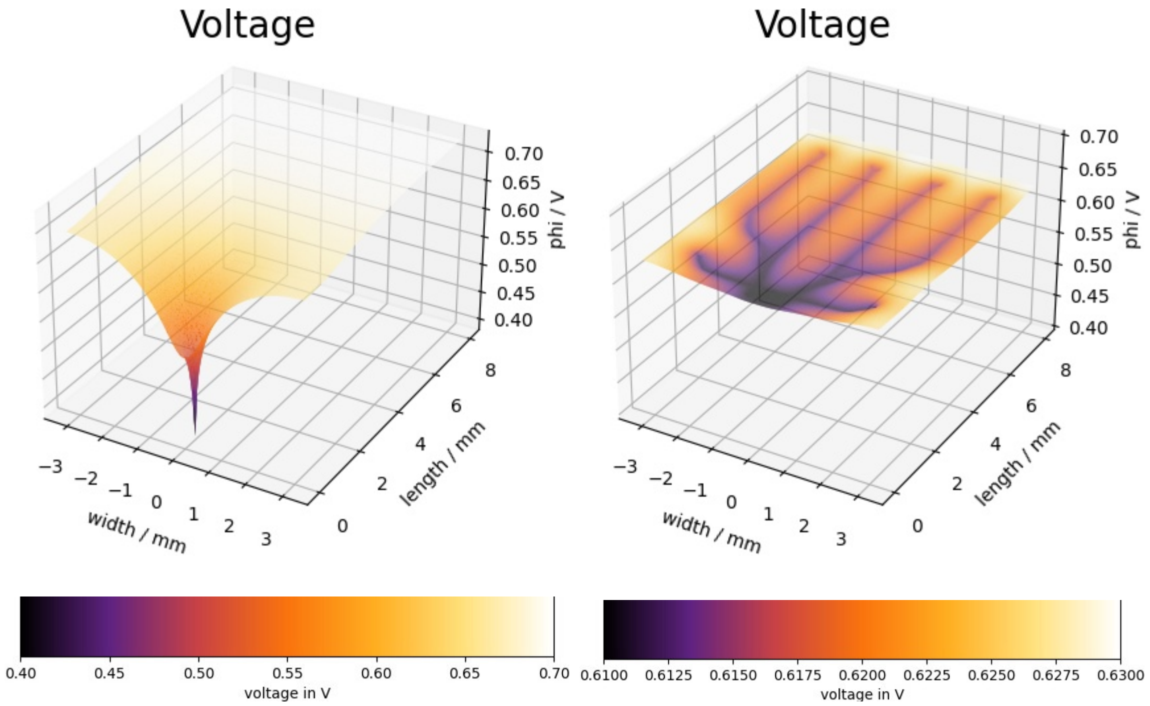


Figure 1.3: Voltage without (left) and with grid (right). The external front contact is located at (0,0.5).

If a subarea of the solar cell has a grid, electrons flow not only through the TCO but also across the grid. A parallel connection can be observed. In summary, the generated current depends on the photocurrent and the operating voltage, that is applied to the solar cell.

2 Topology Optimization

In this thesis, the grid pattern of a solar cell is to be optimized to maximize the resulting power conversion efficency η. In other words, the distribution of well-conducting metal on the surface of the solar cell is sought, with which it is possible to generate a large amount of current, but also to dissipate it efficiently. An optimization in which the distribution of material on a domain is optimized is called topology optimization [13, Chap. 1]. In the context of this thesis, the domain is the solar cell and the material is metal.

First, in Section 2.1, the basic terminology of topology optimization is explained in general and then applied specifically to the problem of the thesis. The basis for such a topology optimization is an objective function, which calculates the quality of the distribution. In the case of this thesis, the objective function is the power conversion efficiency η.

How to calculate the power conversion efficiency η is explained in Section 2.2. In order to be able to consider the entire area of the solar cell in the numerical calculation, the domain of the solar cell is divided into simplices with a Voronoi diagram. In the next step, the differential equation describing the current flow within the solar cell is solved using a finite element method. In the last step of the calculation, the calculated currents generated in the area of a simplex must be added and multiplied by the applied operating voltage to calculate the power of the solar cell. The efficiency of a solar cell is the ratio of the sun light power that is supplied into the solar cell to the electrical power that is subsequently received back [9].

Now that it is known how to evaluate a grid pattern, the distribution of metal is to be varied while keeping the solar cell structure the same otherwise, in order to iteratively optimize the grid. To ensure that the variation is not random but results in better power conversion efficiencies η, it is best to leave the variation to an optimizer. The last Section 2.3 of this chapter focuses on finding a suitable optimizer. A small experiment with the Nelder-Mead optimizer shows that the use of a gradient-free optimizer, which requires many function evaluations, is unsuitable for a complex and time-consuming function as in the case of the objective function at hand. Gradient-based optimizers are investigated for this purpose. Likewise, at the latest after Section 2.3.1 where the gradient is calculated, it becomes clear that it will not be efficient to use second order optimizers.

2.1 General Topology Optimization Approach

Topology optimization is a mathematical method to find, in the context of this thesis, the optimal material distribution within a prescribed design domain in order to achieve high structure-dependent performance. This could be, e.g., the task of minimizing the material use and strain energy of structures while maintaining their mechanical strength [13, Chap. 1] or in this thesis, to find a grid that optimizes the power conversion efficiency η of a solar cell. In general, the optimization problem can be written as:

$$\min_{\rho} \quad F(\omega, \rho(\omega), \psi(\omega, \rho(\omega))) = \int_{\Omega} f(\omega, \rho(\omega), \psi(\omega, \rho(\omega)))\, \mathrm{d}\omega \tag{2.1}$$
$$\text{subject to constrains on } \begin{cases} \rho(\omega) \\ \psi(\omega, \rho(\omega)) \\ \text{else} \end{cases}$$

The problem of topology optimization is thus defined by the six terms:

- Design domain $\Omega \subset \mathbb{R}^d$
- Design function
  $\rho \colon \Omega \to \{0,1\}, \quad \omega \mapsto \rho(\omega)$
- State function
  $\psi \colon \Omega \times \{0,1\} \to \mathbb{R}, \quad (\omega, \rho(\omega)) \mapsto \psi(\omega, \rho(\omega))$
- Local objective function
  $f \colon \Omega \times \{0,1\} \times \mathbb{R} \to \mathbb{R}, \quad (\omega, \rho(\omega), \psi(\omega, \rho(\omega))) \mapsto f(\omega, \rho(\omega), \psi(\omega, \rho(\omega)))$
- Objective function
  $F \colon \Omega \times \{0,1\} \times \mathbb{R} \to \mathbb{R}, \quad (\omega, \rho(\omega), \psi(\omega, \rho(\omega))) \mapsto F(\omega, \rho(\omega), \psi(\omega, \rho(\omega)))$
- Constraints

Thereby, the design domain Ω defines the geometry in which the material distribution is to be optimized. The vector ω is a specific position within the design domain. The design function ρ is a discontinous function and represents the material distribution. An evaluation of the function at the position ω describes whether material is currently used at this position.

Based on the position and material properties at this location, several states can be created. For example, if a water channel is considered and an obstacle (material) is placed in the flow of water, a change of state can be achieved. Where the obstacle is placed, water no longer flows. The states at different positions ω within the design domain Ω, depending on the material property there, are represented by the state function ψ. Often, the state function satisfies a linear or nonlinear equation of state and the evaluation of the state function is mostly done by solving a differential equation.

A change of state at a position has the consequence that the problem, which is optimized, is either better or worse solved at this position. The local objective function f evaluates exactly that and has the same unit as the objective function. The objective function F is an indicator of how good the structure-dependent performance is. This is composed of the total return of the performance achieved at each position. Accordingly, the integral is formed over the individual evaluations f at each position ω of the design domain, as shown in Equation (2.1).

In the case of the example task mentioned at the beginning, to minimize the strain energy of structures while maintaining their mechanical strength, the objective function F would reflect the strain energy of the entire geometry, just as the local objective function f would reflect the strain energy for individual positions. The strain energy is based on the state function ψ. The state function ψ represents the displacements depending on the element thicknesses given by the design function ρ [13, Chap. 1]. In this task, other requirements must also be satisfied at the same time, such as maintaining their mechanical strength. These requirements can be subordinated to the concept of constraints.

The constraints on the topology optimization model are chosen so that the model reflects reality as accurately as possible. Constraints, for instance, ensure that the material distribution in some parts of the domain cannot change at all or can only change in a certain way. Several constraints are possible while performing topology optimization. The constraints can directly affect the design function ρ, the state function ψ or other parts of the model. In the case of the design function ρ, there can be a limit how much material can be used within the design domain and constraints affecting the state vector ψ, e.g., can be boundary conditions for the differential equation.

Discretization

In order to transform the continuous problem into a numerically solvable problem, the design domain Ω is discretized into several subdomains. For this purpose, the geometry is divided into equidistant mesh points ω_d. This way the design domain is represented by mesh points. Around the mesh points, new simplices Ω_i are created, e.g., by a Voronoi diagram [14, 15], to perform a finite element method. For each mesh point there is exactly one simplex.
Due to the discretization, the design function ρ is only evaluated at the mesh points. This leads to a vector, whose components represent the evaluation of the design function ρ at the corresponding mesh point. The vector is called density vector ρ_d.

As mentioned before, evaluating the state function ψ includes a differential equation. To obtain the solution of the differential equation after discretization, it can be approximated using a finite element method (FEM), on the mesh described above. In general, a FEM gives the following system of equations with the stiffness matrix $\mathbf{K}$ which is a matrix valued function and the right-hand side $\mathbf{f}$:

$$\mathbf{K}(\rho_d)\psi_d(\rho_d) = \mathbf{f} \tag{2.2}$$

The stiffness matrix $\mathbf{K}$ is composed of the properties of the materials used and thus depends directly on the density vector ρ_d. The assembly of $\mathbf{K}$ and the definition of ψ_d in the context of this thesis will be discussed in detail later.

Due to the discretization, the integral in Equation (2.1) can be approximated by a sum,

$$F(\omega, \rho, \psi) = \int_{\Omega} f(\omega, \rho, \psi)\ \mathrm{d}\omega \approx \sum_{i=1}^{n} \int_{\Omega_i} f_i(\rho_{d_i}, \psi_{d_i}), \tag{2.3}$$

where n is the number of different simplices and the functions f_i describe how good the performance of the associated subdomain is.

Topology Optimization

Based on [16] an algorithm of the topology optimization can be created. The steps are listed in Algorithm 1.

Algorithm 1 Topology optimization algorithm

```
1. Initialize simplices $\Omega_i$ on design domain $\Omega$
2. Set initial guess of density vector $\rho_d$
3. Start optimization loop until convergence
    a) Calculate objective function F$(\omega_d, \rho_d, \psi_d)$ for given density vector $\rho_d$
    b) Update density vector $\rho_d$ with an optimizer
    c) Modify density vector $\rho_d$ in order to get better results
```

Once the model for the topological optimization is completed, the search for a suitable optimization method starts.

Due to the binary distributed density vector it seems obvious to solve the task with an optimization approach which is suitable for optimizing binary vectors. However, solving the problem, as long as it involves a binary vector, becomes extremely difficult. So far only very small problems have been solved to global optimality [17]. Since this does not really work, a different approach is used. Instead of a binary density vector $\rho_d \in \{0, 1\}^n$, a continuous vector $\tilde{\rho}_d \in \mathbb{R}^n$ is used. This allows the use of a continuous optimizer to find global optimality. Then, using a projection, the vector $\tilde{\rho}_d$ is mapped back to a binary density vector ρ_d. But even this approach does not guarantee that it will find a converged solution. The problem is that these approaches are often sensitive to parameter fluctuations and therefore end up in oscillating and non-converging solutions [18].

Topology Optimization Model for Solar Cell

On the basis of the general procedure, topology optimization is performed on a solar cell within this thesis. For this purpose, the geometry of the solar cell, which serves as the desgin domain, is discretized into simplices from the beginning and only considered in this form. This will remain for the rest of the thesis. The optimization problem can be formulated as finding the density vector ρ_d representing the metalization grid pattern that improves the power conversion efficiency η of a thin-film solar cell.

The terms of topology optimization are defined as shown in Table 2.1:

General	**Thesis**	**Description**
Design domain Ω	Ω	The geometry of a solar cell [-3.2,3.2]×[0,7.95] $\subset \mathbb{R}^2$ used in [3]
Density vector ρ_d	$\mathbf{x}$	Describes the material distribution whether a simplex Ω_i gets metalization grid or not
State vector ψ_d	$\mathbf{u}$	Represents the electric potential of each simplex Ω_i
Stiffness matrix $\mathbf{K}$	$\mathbf{G}$	Represents the electrical conductivities between two simplices Ω_i
Right-hand side $\mathbf{f}$	$\mathbf{i}^{\text{gen}}$	Represents the generated currents in each simplex Ω_i
Local objective function f_i	$\mathbf{i}_i^{\text{gen}}$	The indicator for the quality of a simplex Ω_i is the generated current
Objective function F	η	The indicator for the quality of a design is the power conversion efficiency
Constraints	$\mathbf{u}_j = \text{V}_{\text{op}}$ $\mathbf{x}_j = 1$	Where j is the index of the subdomain, where the external front contact is applied

Table 2.1: Terminologies for the topology optimization of a solar cell.

2.2 Calculation of Power Conversion Efficiency

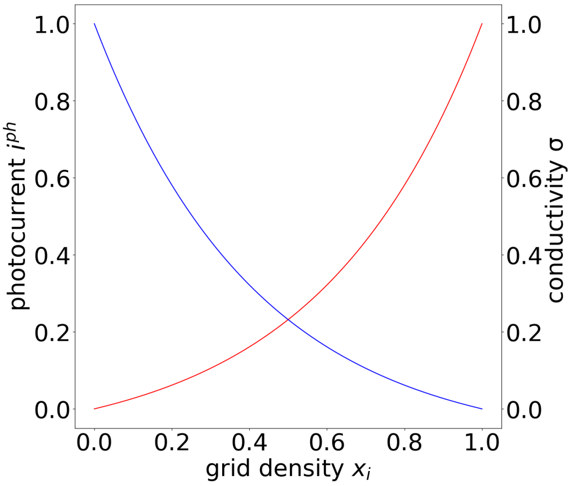
The power conversion efficiency η of a solar cell depends on several factors and is the objective function in the framework of topology optimization of the metalization coating of a solar cell. In order to obtain a proper approximation of the efficiency, a discretization of the solar cell is performed as explained in Section 2.1. For this purpose, the domain of the solar cell gets divided in multiple subdomains to perform a FEM simulation. The center of a subdomain is the mesh point and they are arranged within an equidistant mesh. To obtain a plain rectangular boundary, additional pairs of mesh points are added so that the connecting line of a pair of mesh points would be orthogonal to the boundary and the boundary is between those two points. By placing these pairs close enough to each other, the desired boundary is obtained by a Voronoi diagram [4]. Due to this procedure, the edge subdomains have a non-equidistant form, as can be seen in Figure 2.2.

From now on, an *element* of the solar cell is an object whose shape and position on the solar cell is defined by the subdomain. In addition, an element contains information about the voltage and the generated current in the specific part of the solar cell and this part can also have a metalization coating. In short, the elements have the same properties as a mini solar cell.

Introduction of the Density Vector representing the Metalization Grid

As explained in Section 1.1, the generated current of an element depends on the photocurrent generated in the element and the conductivity to the neighboring elements. So that sunlight reaches the absorber and photocurrent is generated, the absorbing material has a TCO layer on top. To achieve a better conductivity some elements of the solar cell get a metalization grid. Furthermore a better conductivity implies a lower electrical resistance. Unfortunately, the elements with grid generate no current, so most elements do not have a grid. This trade-off will be optimized in this thesis.

Normally, the components of the density vector $\mathbf{x}$ (see Table 2.1) have a value of zero if the corresponding element should have no grid or a value of one to cover the element by grid. For the optimization of grid patterns, a continuous function is used to allow values between zero and one, as explained in the last section. These functions are based on the SIMP (Solid Isotropic Material with Penalization) approach and are called density functions [19, 20, 21]. Basically, the density function can have any shape [22]. However, the focus in this thesis is on exponential functions. First, a cubic interpolation was used according to Gupta [23]. Because of the flexibility and the easy way to calculate the gradient of an exponential function, the exponential function is used for interpolation instead. At the end of the optimization, all values of $\mathbf{x}$ should be zero or one and this is given by a suitable choice of the function. The first optimizations are performed with the functions $SIMP_{\text{con}}$ for the conductivity and $SIMP_{\text{gen}}$ for the amount of generated current in Figure 2.1, and for later use their gradients are provided, too. Different exponential functions are discussed in Section 3.3. The remaining intermediate values of $\mathbf{x}$ are set at the end of the optimization either to 0 or to 1 depending on a threshold value.



Red:
$$SIMP_{\text{gen}}(\mathbf{x}_i) = 0.1 \cdot e^{\ln(11)\cdot(1-\mathbf{x}_i)} - 0.1.$$
$$\frac{\mathrm{d}SIMP_{\text{gen}}(\mathbf{X}_i)}{\mathrm{d}\mathbf{X}_i} = -0.1 \cdot \ln(11) \cdot e^{\ln(11)\cdot(1-\mathbf{x}_i)}$$

Blue:
$$SIMP_{\text{con}}(\mathbf{x}_i) = 0.1 \cdot e^{\ln(11)\cdot\mathbf{x}_i} - 0.1$$
$$\frac{\mathrm{d}SIMP_{\text{con}}(\mathbf{X}_i)}{\mathrm{d}\mathbf{X}_i} = 0.1 \cdot \ln(11) \cdot e^{\ln(11)\cdot\mathbf{x}_i}$$

Figure 2.1: Initial SIMP functions used in the optimization.

Calculation of the Power Conversion Efficiency

The power conversion efficiency η is the proportion of the incident illumination power $\mathrm{p}_{\text{light}}$ that is converted into electrical power p_{cell} within the solar cell:

$$\eta_{\text{total}}^{\text{cell}} = -\frac{\mathrm{p}_{\text{cell}}}{\mathrm{p}_{\text{light}}} \tag{2.4}$$

The minus before the fraction is needed because incoming power is positive while outgoing power is negative. The electrical power p_{cell} is calculated with the generated current I in the domain Ω by the expression

$$\mathrm{p}_{\text{cell}} = \mathrm{V}_{\text{op}} \cdot I(U, I^{\text{ph}}), \tag{2.5}$$

where V_{op} is the applied operating voltage at the external front contact. A closer look at Equation (2.5) gives two different parts. The first part, V_{op}, is constant in the whole calculation. The second part is the generated current of a solar cell depending on the photocurrent I^{ph} and the voltage U. The relationship between these terms can be derived from Maxwell's equations [24, 25]. The relevant Maxwell's equations are given by

$$\nabla \cdot D = \tilde{\rho}, \tag{2.6}$$
$$\nabla \times H = \frac{\partial D}{\partial t} + j, \tag{2.7}$$

where

electric displacement field	D	
enclosed charge density	$\tilde{\rho}$	
current density	$j = \frac{1}{\rho^{\text{sr}}} \cdot E$	(2.8)
material dependent specific resistance	ρ^{sr}	(2.9)
electric field in Ω	$E = -\nabla U.$	(2.10)

Due to the divergence-free nature of the total current density (2.11), the following applies

$$\begin{aligned} 0 &= \nabla \cdot (\nabla \times H) && (2.11) \\ &= \nabla \cdot \left(\frac{\partial D}{\partial t} + j \right) \\ &= \nabla \cdot \frac{\partial D}{\partial t} + \nabla \cdot j \\ &= \frac{\partial (\nabla \cdot D)}{\partial t} + \nabla \cdot j \\ &= \frac{\partial \tilde{\rho}}{\partial t} + \nabla \cdot j && \text{Continuity equation} \qquad (2.12) \end{aligned}$$

Poisson's Equation of Charge Conservation during Electrical Conductivity

Based on the formulation (2.12) and use of Equations (2.8) and (2.10) the problem to calculate the voltage can be set up:

$$\begin{aligned} 0 &= \frac{\partial \tilde{\rho}}{\partial t} + \nabla \cdot j \\ &\overset{(2.8)}{=} \frac{\partial \tilde{\rho}}{\partial t} + \nabla \cdot \left(\frac{1}{\rho^{\text{sr}}} \cdot E \right) \\ &\overset{(2.10)}{=} \frac{\partial \tilde{\rho}}{\partial t} + \nabla \cdot \left(\frac{1}{\rho^{\text{sr}}} \cdot (-\nabla U) \right) \\ &= \frac{\partial \tilde{\rho}}{\partial t} - \frac{1}{\rho^{\text{sr}}} \cdot \Delta U \end{aligned}$$

From this the problem to be considered is derived. The problem is based on Poisson's equation (2.13) of charge conservation during electrical conductivity with $\tilde{\Omega} \subset \Omega$,

$$\begin{aligned} \frac{1}{\rho^{\text{sr}}(x,y)} \cdot \Delta U(x,y) &= \frac{\partial \tilde{\rho}}{\partial t} && \forall (x,y) \in \Omega \setminus \tilde{\Omega} \qquad (2.13) \\ U(x,y) &= \text{V}_{\text{op}}, && \forall (x,y) \in \tilde{\Omega} \end{aligned}$$

with mixed Dirichlet and Neumann boundary conditions,

$$\begin{aligned} U(x,y) &= \text{V}_{\text{op}} && \forall (x,y) \in \partial \tilde{\Omega} \qquad (2.14) \\ \mathbf{n} \cdot \left(\frac{1}{\rho^{\text{sr}}(x,y)} \cdot \nabla U(x,y) \right) &= 0, && \forall (x,y) \in \partial \Omega \qquad (2.15) \end{aligned}$$

where $\tilde{\Omega}$ is the to the external front contact connected area of the solar cell and ρ^{sr} is a piecewise constant (p.c.) function of the specific sheet resistance at the position $(x, y) \in \Omega$. Assuming that no Neumann condition is placed on $\partial\tilde{\Omega}$, the solution space and the test space with test functions φ are chosen according to [26]:

$$U \in H^1_{D_{\mathrm{V_{op}}}} := \left\{ U \in H^1(\Omega) \mid U = \mathrm{V_{op}} \text{ on } \partial\tilde{\Omega} \right\} \tag{2.16}$$

$$\varphi \in H^1_{D_0} := \left\{ \varphi \in H^1(\Omega) \mid \varphi = 0 \text{ on } \partial\tilde{\Omega} \right\} \tag{2.17}$$

The difference between the two spaces is that the Dirichlet condition (2.14) is built into the definition of the solution space $H^1_{D_{\mathrm{V_{op}}}}$ (2.16), while the functions φ are zero in the test space $H^1_{D_0}$ (2.17) on the Dirichlet portion of the boundary. Furthermore,

$$H^1_{\mathrm{V_{op}}} \subseteq H^1_{D_{\mathrm{V_{op}}}} \subseteq H^1 \text{ and } H^1_0 \subseteq H^1_{D_0} \subseteq H^1.$$

This differential equation can be reformulated into a weak formulation of the general elliptic problem. To do this, a test function $\varphi \in H^1_{D_0}$ is multiplied on both sides and then integrated over the domain $\Omega \setminus \tilde{\Omega}$

$$\int_{\Omega\setminus\tilde{\Omega}} \left(\frac{1}{\rho^{\mathrm{sr}}} \cdot \Delta U \right) \cdot \varphi \;\; \mathrm{d}(x,y) = \int_{\Omega\setminus\tilde{\Omega}} \frac{\partial\tilde{\rho}}{\partial t} \cdot \varphi \;\; \mathrm{d}(x,y). \tag{2.18}$$

By partial integration of the left-hand side of Equation (2.18), the following terms are obtained:

$$\underbrace{\int_{\partial(\Omega\setminus\tilde{\Omega})} \nabla U \cdot \mathbf{n} \cdot \left(\frac{1}{\rho^{\mathrm{sr}}} \cdot \varphi \right) \;\; \mathrm{d}(x,y)}_{L} - \underbrace{\int_{\Omega\setminus\tilde{\Omega}} \nabla U \cdot \nabla \left(\frac{1}{\rho^{\mathrm{sr}}} \cdot \varphi \right) \;\; \mathrm{d}(x,y)}_{K}$$

For the sake of simplicity, the integrals are discussed separately:

$$K = \int_{\Omega\setminus\tilde{\Omega}} \nabla U \cdot \frac{1}{\rho^{\mathrm{sr}}} \cdot \nabla\varphi \;\; \mathrm{d}(x,y) + \int_{\Omega\setminus\tilde{\Omega}} \nabla U \cdot \underbrace{\nabla \frac{1}{\rho^{\mathrm{sr}}}}_{=0,\ \rho^{\mathrm{sr}} \text{ p.c.}} \cdot \varphi \;\; \mathrm{d}(x,y) \tag{2.19}$$

$$L = \int_{\partial\Omega} \underbrace{\nabla U \cdot \mathbf{n} \cdot \frac{1}{\rho^{\mathrm{sr}}}}_{=0, \text{ see (2.15)}} \cdot \varphi \;\; \mathrm{d}(x,y) + \int_{\partial\tilde{\Omega}} \nabla U \cdot \mathbf{n} \cdot \frac{1}{\rho^{\mathrm{sr}}} \cdot \underbrace{\varphi}_{=0, \text{ see (2.17)}} \;\; \mathrm{d}(x,y) \tag{2.20}$$

Combined, this results in the weak formulation:

$$\underbrace{-\int_{\Omega\setminus\tilde{\Omega}} \frac{1}{\rho^{\mathrm{sr}}} \cdot \nabla U \cdot \nabla\varphi \;\; \mathrm{d}(x,y)}_{a(u,\varphi)} = \underbrace{\int_{\Omega\setminus\tilde{\Omega}} \frac{\partial\tilde{\rho}}{\partial t} \cdot \varphi \;\; \mathrm{d}(x,y)}_{l(\varphi)} \tag{2.21}$$

With the relation

$$\frac{\partial\tilde{\rho}}{\partial t} = \jmath^{\mathrm{ph}} \tag{2.22}$$

the weak formulation can be transformed to

$$\underbrace{\int_{\Omega\setminus\tilde{\Omega}} \frac{1}{\rho^{\mathrm{sr}}} \cdot \nabla U \cdot \nabla\varphi \quad \mathrm{d}(x,y)}_{a(u,\varphi)} = \underbrace{-\int_{\Omega\setminus\tilde{\Omega}} j^{\mathrm{ph}} \cdot \varphi \quad \mathrm{d}(x,y)}_{l(\varphi)}. \tag{2.23}$$

A unique solution to the weak formulation of the problem exists [27]. An approximated solution itself can be found in a finite dimensional subspace of Ω with basis $\{\varphi_j\}_{j=1}^{n}$ by a finite element method. The approximate solution $\tilde{U}$ for the basis is given by

$$\tilde{U} = \sum_{j=1}^{n} \mathbf{u}_j \varphi_j, \tag{2.24}$$

where $\mathbf{u}$ is the solution vector of the system of equations

$$\mathbf{G} \cdot \mathbf{u} = -\mathbf{i}^{\mathrm{gen}}(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}), \tag{2.25}$$

where $\mathbf{G}$ is the stiffness matrix, representing the conductivities between elements, $\mathbf{u}$ the voltage vector and $\mathbf{i}^{\mathrm{gen}}$ the generated current vector.
Thereby, the entries g_{ij} of the stiffness matrix $\mathbf{G}$ can be determined as

$$g_{ij} = a(\varphi_j, \varphi_i)$$

and the entries $\mathbf{i}_j^{\mathrm{gen}}$ of the generated current vector $\mathbf{i}^{\mathrm{gen}}$ as

$$\mathbf{i}_j^{\mathrm{gen}} = l(\varphi_j).$$

The solution $U(x,y)$ of the differential equation (2.13) is defined in the domain Ω as

$$U(x,y) = \begin{cases} \tilde{U} = \sum\limits_{j=1}^{n} \mathbf{u}_j \varphi_j & (x,y) \in \Omega \setminus \tilde{\Omega} \\ \mathrm{V}_{\mathrm{op}} & (x,y) \in \tilde{\Omega} \end{cases} \tag{2.26}$$

and thus, the total generated current I of the solar cell can be calculated with

$$I = \frac{U}{\rho^{\mathrm{sr}}} = \begin{cases} -\sum\limits_{j=1}^{n} \mathbf{i}_j^{\mathrm{gen}} \varphi_j & (x,y) \in \Omega \setminus \tilde{\Omega} \\ \\ \dfrac{\mathrm{V}_{\mathrm{op}}}{R_{\mathrm{contact}}} & (x,y) \in \tilde{\Omega}. \end{cases} \tag{2.27}$$

Deduction of System of Equations

In the previous section, a possibility to assemble a stiffness matrix has been shown using the weak theory to a mixed Dirichlet-Neumann boundary value problem. Furthermore, this approach makes it possible to prove that the problem is well-posed. Independently of the weak formulation, the system of equations in Equation

(2.25) can also be set up in another way. Thereby, a more intuitive approach is used to solve the problem, avoiding having to make a concrete choice of test functions $\varphi \in H^1_{D_0}$. The basic idea behind the calculation is Ohm's law, which describes the relation between the voltage U, the resistance R and the current I. With the relation in Equation (2.12) and Equation (2.22) follows

$$-j^{\mathrm{ph}} = \nabla \cdot j. \tag{2.28}$$

By integrating on both sides over the domain Ω the following transformations are obtained:

$$-\int_{\Omega} j^{\mathrm{ph}} \ \ \mathrm{d}(x,y) = \int_{\Omega} \nabla \cdot j \ \ \mathrm{d}(x,y) \tag{2.29}$$

On the left side, the generated current I at position (x,y) within the solar cell is achieved. On the right side, Gauss's theorem can be used. Thus,

$$I(x,y) = -\int_{\partial\Omega} j \ \ \mathrm{d}(x,y). \tag{2.30}$$

With the simplices Ω_i created according to Section 2.1 the domain Ω can be decomposed and thus also the integral:

$$I(x,y) = \int_{\Omega} j^{\mathrm{ph}} \ \ \mathrm{d}(x,y) = \sum_{i=1}^{n} \int_{\Omega_i} j^{\mathrm{ph}} \ \ \mathrm{d}(x,y) \tag{2.31}$$

Analogous to Equation (2.29) and Equation (2.30), the same can be done for a simplex Ω_i. This leads to

$$I(x,y) = \sum_{i=1}^{n} \int_{\Omega_i} j^{\mathrm{ph}} \ \ \mathrm{d}(x,y) = \sum_{i=1}^{n} I_i(x,y) = -\sum_{i=1}^{n} \int_{\partial\Omega_i} j \ \ \mathrm{d}(x,y), \tag{2.32}$$

where $I_i(x,y) = \mathbf{i}_i^{\mathrm{gen}}$. The boundary of a simplex, by construction, consists of linear sections connected to each other. The integral over the boundary of a simplex $\partial\Omega_i$ is thus the sum over the integrals of the individual linear segments $\partial\Omega_{i_k}$:

$$\mathbf{i}_i^{\mathrm{gen}} = -\int_{\partial\Omega_i} j \ \ \mathrm{d}(x,y) = -\sum_{k \in \mathrm{N}_i} \int_{\partial\Omega_{i_k}} j \ \ \mathrm{d}(x,y) \tag{2.33}$$

Thereby, N_i is the index set of neighbor elements of the i^{th} element. Furthermore with Ohm's law

$$\int_{\partial\Omega_{i_k}} j \ \ \mathrm{d}(x,y) = \frac{\mathbf{u}_k - \mathbf{u}_i}{\mathbf{R}_{ik}}, \tag{2.34}$$

where $\mathbf{u}_i$ and $\mathbf{u}_k$ are the voltages in the corresponding simplex.
Thus, for each element, according to Equation (2.33) and Equation (2.34), the following correlation can be created:

$$0 = \mathbf{i}_i^{\mathrm{gen}}(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}) + \sum_{k \in \mathrm{N}_i} \frac{\mathbf{u}_k - \mathbf{u}_i}{\mathbf{R}_{ik}} = \mathbf{i}_i^{\mathrm{gen}}(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}) + \left(\sum_{k \in \mathrm{N}_i} \frac{\mathbf{u}_k}{\mathbf{R}_{ik}} - \mathbf{u}_i \cdot \sum_{k \in \mathrm{N}_i} \frac{1}{\mathbf{R}_{ik}} \right) \tag{2.35}$$

From the relationship between conductivity σ_{ik} and resistance $\mathbf{R}_{ik}$, it follows with $\sigma_{ik} = \dfrac{1}{\mathbf{R}_{ik}}$, that

$$0 = \mathbf{i}_i^{\text{gen}}(\mathbf{i}^{\text{ph}}, \mathbf{u}) + \left(\sigma_{i1}, ..., -\sum_{k \in \mathrm{N}_i} \sigma_{ik}, ..., \sigma_{in} \right) \cdot (\mathbf{u}_1, ..., \mathbf{u}_i, ..., \mathbf{u}_n)^T , \tag{2.36}$$

where $\sigma_{ik} = 0$, if $k \notin \mathrm{N}_i$. The conductivity σ_{ik} from the i^{th} to the k^{th} element is 0, since there is no direct connection between these two elements and therefore there is no current between them. Thus, for all elements Equation (2.36) can be interpreted as a matrix vector multiplication and is equivalent to Equation (2.25):

$$0 = \mathbf{i}^{\text{gen}}(\mathbf{i}^{\text{ph}}, \mathbf{u}) + \mathbf{G} \cdot \mathbf{u} \tag{2.37}$$

The total generated current I in a solar cell is calculated according to Equation (2.32).

Assembly of the Stiffness Matrix

Regarding the deduction in Equation (2.35) and (2.36), the entries $\mathrm{g}_{ij} \in \mathbf{G}$ look as follows:

$$\mathrm{g}_{ij} = \begin{cases} \sigma_{ij} & j \in \mathrm{N}_i \\ -\sum\limits_{j \neq i} \sigma_{ij} & j = i \\ 0 & \text{else} \end{cases} \tag{2.38}$$

The conductivity σ_{ij} between two adjacent elements can be calculated by the relation to the resistance $\mathbf{R}_{ij}$ mentioned above. $\mathbf{R}_{ij}$ is the resistor between the i^{th} and the j^{th} element. Thus, the resistance $\mathbf{R}_{ij}$ is composed of the resistance $\mathbf{R}_{ij}^{\text{toEdge}}$ of the i^{th} element and the resistance $\mathbf{R}_{ji}^{\text{toEdge}}$ of the j^{th} element [4]:

$$\mathbf{R}_{ij} = \mathbf{R}_{ij}^{\text{toEdge}} + \mathbf{R}_{ji}^{\text{toEdge}} = \mathbf{R}_{ji} \tag{2.39}$$

The resistance $\mathbf{R}_{ij}^{\text{toEdge}}$ depends mainly on the material used in the element i^{th} element. If the element has a grid coating in addition to the transparent conducting oxide, electrons can flow through both materials. In this case, the resistance is calculated using the formula for parallel resistance. Otherwise, the resistor is simply the resistance of the TCO:

$$\mathbf{R}_{ij}^{\text{toEdge}} = \begin{cases} \mathbf{R}_{ij}^{\text{TCO,toEdge}} & \mathbf{x}_i = 0 \\ \dfrac{\mathbf{R}_{ij}^{\text{grid,toEdge}} \cdot \mathbf{R}_{ij}^{\text{TCO,toEdge}}}{\mathbf{R}_{ij}^{\text{grid,toEdge}} + \mathbf{R}_{ij}^{\text{TCO,toEdge}}} & \mathbf{x}_i \neq 0 \end{cases} \tag{2.40}$$

In general, the resistance of the grid coating $\mathbf{R}_{ij}^{\text{grid,toEdge}}$ or the one of the TCO layer $\mathbf{R}_{ij}^{\text{TCO,toEdge}}$ depends on the specific resistance ρ of the material used in the element, the Euclidean distance of the mesh points, the length of the common edge $\partial\Omega_i \cap \partial\Omega_j$ of the i^{th} and the j^{th} simplex and the thickness d of the material layer. Due to its construction, the electrons flow exactly half the distance between the mesh points in the material of the i^{th} element and half the distance in the j^{th} element [4]. The thickness and the common edge define the area in which electrons can flow from element to element. While the thickness of the TCO layer is uniform across the solar cell, the thickness of the grid layer varies in each element. The resistance of the grid layer also depends on the entry of the density vector $\mathbf{x}$. For intermediate values of the density vector $\mathbf{x}$, the SIMP function for the conductivity $SIMP_{\text{con}}$ presented in Figure 2.1 is used:

$$\mathbf{R}_{ij}^{\text{grid,toEdge}} = \frac{\rho^{\text{grid,toEdge}} \cdot \frac{\left\|\mathbf{x}_{pos_i} - \mathbf{x}_{pos_j}\right\|_2}{2}}{d^{\text{grid,toEdge}} \cdot |\partial\Omega_i \cap \partial\Omega_j|} \cdot SIMP_{\text{con}}(\mathbf{x}_i) \tag{2.41}$$

$$\mathbf{R}_{ij}^{\text{TCO,toEdge}} = \frac{\rho^{\text{TCO,toEdge}} \cdot \frac{\left\|\mathbf{x}_{pos_i} - \mathbf{x}_{pos_j}\right\|_2}{2}}{d^{\text{TCO,toEdge}} \cdot |\partial\Omega_i \cap \partial\Omega_j|}$$

The constructed stiffness matrix $\mathbf{G}$ is sparse, i.e., apart from the diagonal entry, the i^{th} row has one entry for each neighbor of the i^{th} element. Within the simulation software twinPV [3], this property is used to store the entries efficiently and thus, calculations are faster, because zero entries are ignored. All matrix entries are stored in the Compressed Row Storage (CRS) format.

Before Equation (2.25) can be solved, the constraint, set in Table 2.1, has to be added. If an external front contact is applied to the i^{th} element, the voltage $\mathbf{u}_i$, at this element, is equal to the operating voltage V_{op}. Hence, the i^{th} row of the stiffness matrix $\mathbf{G}$ is a unit line. The corresponding entry of the right-hand side is the operating voltage V_{op}. Thus, the desired result of $\mathbf{u}_i = \text{V}_{\text{op}}$ follows from the matrix-vector multiplication.
Hence, the system of equations (2.25) has this specific form in the i^{th} row:

$$\begin{pmatrix} g_{11} & \cdots & g_{1i-1} & g_{1i} & g_{1i+1} & \cdots & g_{1n} \\ \vdots & & & \vdots & & & \vdots \\ g_{i-11} & \cdots & g_{i-1i-1} & g_{i-1i} & g_{i-1i+1} & \cdots & g_{i-1n} \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ g_{i+11} & \cdots & g_{i+1i-1} & g_{i+1i} & g_{i+1i+1} & \cdots & g_{i+1n} \\ \vdots & & & \vdots & & & \vdots \\ g_{n1} & \cdots & g_{ni-1} & g_{ni} & g_{ni+1} & \cdots & g_{nn} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \mathbf{u}_{i+1} \\ \vdots \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} -\dot{\mathbf{i}}_1^{\text{gen}} \\ \vdots \\ -\dot{\mathbf{i}}_{i-1}^{\text{gen}} \\ \text{V}_{\text{op}} \\ -\dot{\mathbf{i}}_{i+1}^{\text{gen}} \\ \vdots \\ -\dot{\mathbf{i}}_n^{\text{gen}} \end{pmatrix} \tag{2.42}$$

Solving the nonlinear System of Equations

So far, only the stiffness matrix $\mathbf{G}$ is known. To be able to solve the system of equations (2.25), the right side must also be known. The right side of the equation system is the generated current $\mathbf{i}^{\text{gen}}$ that depends on the voltage $\mathbf{u}$ and the photocurrent $\mathbf{i}^{\text{ph}}$.

To calculate the photocurrent in the i^{th} element Equation (2.43) is used:

$$\mathbf{i}_i^{\text{ph}} = j^{\text{ph}} \cdot A_i \cdot SIMP_{\text{gen}}(\mathbf{x}_i) \qquad \begin{array}{c}\text{current density } j \\ \text{element area } A_i\end{array} \tag{2.43}$$

If the voltage $\mathbf{u}$ and the photocurrent $\mathbf{i}^{\text{ph}}$ is known, the generated current can be calculated with the equation for the single diode model [28]. The problem is that the equation is an implicit equation. Also, to calculate the generated current $\mathbf{i}^{\text{gen}}$ of an element, the voltage $\mathbf{u}$ of that element is needed. But that is the solution vector of the system of equations. For this reason, there is no way to solve it directly.

To solve this problem, Equation (2.37) is used. A function $\boldsymbol{f}$ can be defined whose root is the solution vector $\mathbf{u}$.

$$\boldsymbol{f}(\mathbf{z}) = \mathbf{i}^{\text{gen}}(\mathbf{i}^{\text{ph}}, \mathbf{z}) + \mathbf{G} \cdot \mathbf{z} \tag{2.44}$$

The problem to solve is a root-finding problem. In principle, any root-finding algorithm can be used to solve this problem. In the case of the thesis, the Newton-Raphson method is used. The linear system of equations within a Newton iteration is preconditioned with incomplete LU factorization and then solved with the biconjugated gradient method. The methods implemented in the Extreme Numerics library are used for this purpose [29]. With this, the roots of (2.44) can be determined. Now all inputs of $\mathbf{i}^{\text{gen}}(\mathbf{i}^{\text{ph}}, \mathbf{u})$ are defined and the calculation of the electrical power p_{cell} as in Equation (2.5) is possible. The illumination power p_{light} is the power of the incoming light spectrum.

Perturbance of the equidistant Mesh

At the beginning of this chapter, an equidistant mesh has been created and the subdomains of the solar cell have a square shape. This square shape of the elements leads to problems, which are explained below.

A characteristic of electricity is that the current mainly takes the path of least resistance. Hence, the current paths are a line to the center of the adjacent subdomains. Therefore, the current path between two diagonal neighbors runs exactly through the common corner of the subdomains.

In the calculation, this behavior is expressed by a divisor of the length of the common edge of two subdomains $\partial\Omega_i \cap \partial\Omega_j$, as shown in Equation (2.41). When the current flows through the corner, the length of the common edge is zero. To avoid the zero division error, the mesh points are randomly perturbed. This leads to different subdomain shapes and to current paths which only go through edges and not through corners. A checkerboard pattern is created when the current flows only through diagonal elements. Changing the shape of the subdomains also avoids a checkerboard pattern of the grid.

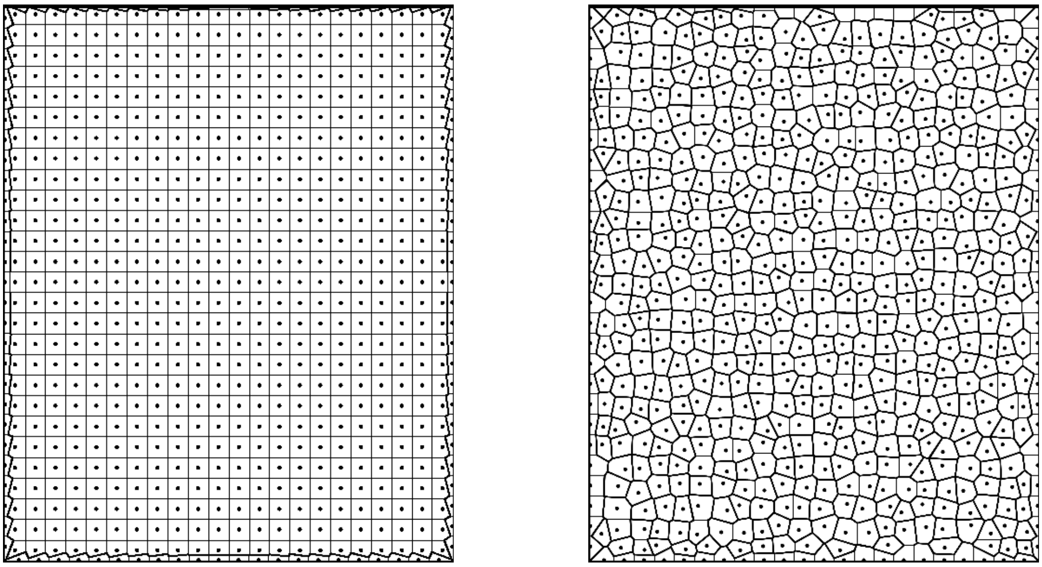


Figure 2.2: Comparison of equidistant and perturbed mesh.

Before creating the irregular subareas with a Voronoi diagram, the positions (x_i, y_i) of each mesh point are randomly shifted with a maximum change of half of the average distance of two mesh points. Figure 2.2 visualizes both meshes and the advantage of the subdomains based on the perturbed approach becomes visible.

Configuration of Solar Cell

For the calculation of the power conversion efficiency η some data of the configuration of a solar cell are needed. These would be, on the one hand, the materials used for the layers, the resulting material-dependent sheet resistances and the parameters needed for the equation of the single diode model. On the other hand, a light spectrum which shines on the solar cell is also required. Except for the grid pattern, these data remain constant throughout the thesis. The values used for the configuration of the solar cell can be found in Table 2.2.

Layer	**Material**	**Thickness d**	**Sheet Resistance ρ**
Back contact	Mo	$500{\cdot}10^{-9}$ m	$53.4{\cdot}10^{-9}\,\Omega$m
Front contact TCO	Al:ZnO	$250{\cdot}10^{-9}$ m	$1.3{\cdot}10^{-5}\,\Omega$m
Front grid	Ni/Al/Ni	$2500{\cdot}10^{-9}$ m	$27{\cdot}10^{-9}\,\Omega$m

Parameters for Single Diode Model	
Saturation current I_0	$2.9991{\cdot}10^{-6}$ A
Photocurrent I_{ph}	411.7050 A
Thermal voltage V_T	0.0259 V
Serial resistance R_s	$1.1254{\cdot}10^{-11}\,\Omega$
Shunt resistance R_{sh}	$0.06667\,\Omega$
Ideality factor of the diode a	1.5093
Number of series-connected cells n	1

Light spectrum	AM 1.5G

Table 2.2: Configuration of the solar cell as in the simulation platform twinPV [3].

2.3 Optimization with Gradient

A topology optimization always includes an optimization, as the name implies. It has been already indicated in Section 2.1 that the choice of a capable optimizer is not easy. For this reason, different optimizers are tested in this section and in the end it is decided which one can best solve the grid optimization problem. Furthermore, the approach presented in Section 2.1 allows a continuous density vector **x** and thus continuous optimizers.

As explained in the last paragraph of Section 2.1, the objective function to be optimized in this thesis is the power conversion efficiency η, see Equation (2.4), of a solar cell depending on the given grid pattern as input parameter **x**. This means the grid pattern that leads to the highest efficiency is the desired return vector.

Thus, the optimization problem can be defined as follows:

$$\begin{aligned} \textbf{max} \quad & \eta_{\text{total}}^{\text{cell}}(\mathbf{x}), \\ \text{subject to} \quad & 0 \leq \mathbf{x}_i \leq 1, \quad \forall_i \in \text{I}, \end{aligned} \tag{2.45}$$

where I = 1,..., n is the set of indexes and n is the number of mesh points.

The following problem is to be optimized:

According to Table 2.1, the design domain Ω of the surface of a solar cell is Ω = [-3.2,3.2]×[0,7.95] $\in \mathbb{R}^2$. Since, for physical reasons, no charge flows from the left half to the right half of the solar cell, it is possible to consider only one side of the solar cell for optimization and then mirror the resulting grid pattern. Thus, the new design domain Ω is Ω = [0,3.2]×[0,7.95] $\in \mathbb{R}^2$. The consequence of this is that with the same number of mesh points a twice finer discretization of the design domain Ω is possible. The design domain Ω is divided into n different simplices Ω_i with the Voronoi diagram depending on a given number of mesh points n as described in Section 2.2. For the optimizations in this chapter and in Chapter 3 20389 mesh points are used. The external front and the back contact are located at position $(0{,}0.5) \in \Omega$.

The threshold used for binarization at the end of the optimizations is based on an empirical observation and therefore, the threshold value is 0.1. The power conversion efficiency η does not change very much, whether an element, which should be coated with grid, see Figure 2.3 (right), has a density of 0.1 or 1. On the other hand, if an element gets grid when it should not get grid, see Figure 2.3 (left), the loss of conversion efficiency η is negligible compared to the other case.

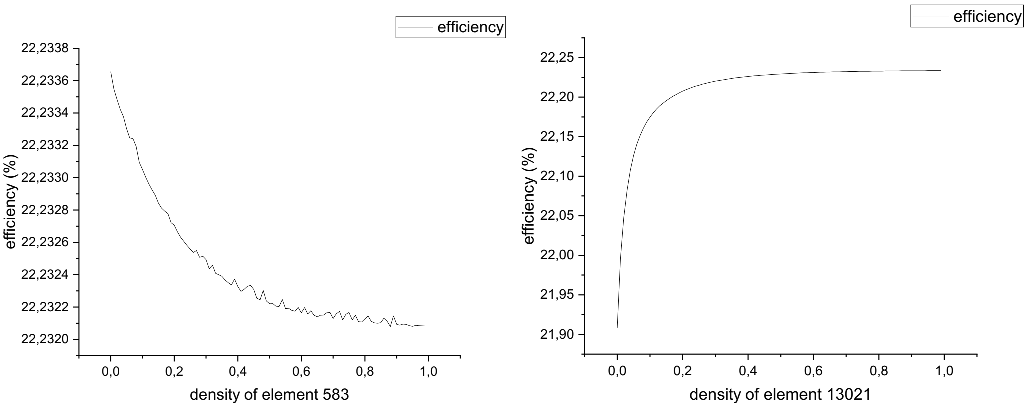


Figure 2.3: Efficiency depending on the density variation of elements without grid (left) and with grid (right).

The computer system on which the optimizations are performed is as in Table 2.3.

Operating system	Microsoft Windows 10 Pro
Processor	AMD Ryzen 7 5800X 8-Core Processor, 3801MHz
RAM	32GB
IDE	Microsoft Visual Studio
Code	Simulation Platform twinPV [3]

Table 2.3: Hardware and software for gradient-based optimization with parallel computing.

Since the calculation of the power conversion efficiency η is very complex and many iterations are required to find a maximum, derivative-free optimizers take a long time to find the optimal input parameters. This is also demonstrated in the paragraph below with a small experiment.

Optimization of Grid Pattern with the Nelder Mead Method

A frequently used optimizer, when the calculation of the derivatives is too costly, is the algorithm of Nelder-Mead. With the help of the optimizer, a grid pattern is to be found with which a high power conversion efficiency η can be calculated. For this optimization, the implementation of the Nelder Mead algorithm from Extreme Numerics [29] is used.

To perform the optimization, the design domain Ω is discretized with 200 mesh points. This leads to 200 different simplices and therefore also to 200 elements, which can have no grid or a grid.

The update scheme of the Nelder Mead method for the density vector $\mathbf{x} \in [0,1]^{100}$ is described in [30]. Starting from the initial guess, the density vector $\mathbf{x}$ is optimized in each iteration. The initial guess for the optimization is the i^{th} unit vector, with the external front contact attached to the i^{th} element.

After 1119 function evaluations in 54.2 sec, a maximum power conversion efficiency η of 16.56 % has been found with a tolerance of 0.4. The resulting grid pattern of this optimization is inefficient, but the main problem is that already for a few mesh points a huge number of function evaluations are necessary.

Gradient-Based Approach

In order to make the optimization more efficient, the gradient of the objective function can be additionally employed.

In general, if a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, the gradient of the function f exists as a vector-valued function $\nabla f \colon \mathbb{R}^n \to \mathbb{R}^n$. The associated components ∇f_i are partial derivatives of f with respect to the components of the input vector $\mathbf{y} \in \mathbb{R}^n$. In particular, the gradient determines how the objective function will change, if $\mathbf{y}$ undergoes a small change. A gradient-based optimizer can use this information to calculate the next iteration point.

This is useful, because:

$$f(\tilde{\mathbf{y}}) = \max_{\mathbf{y}\in\mathbb{R}^n} f(\mathbf{y}) \Leftrightarrow \frac{df(\tilde{\mathbf{y}})}{d\mathbf{y}} = 0 \land \exists_{\varepsilon>0} \forall_{\mathbf{y}\in B_\varepsilon(\tilde{\mathbf{y}})\cap\mathbb{R}^n} : f(\mathbf{y}) \neq f(\tilde{\mathbf{y}}) \tag{2.46}$$

Thus, if $\mathbf{y}$ gets changed marginally to $\tilde{\mathbf{y}}$ and $f(\tilde{\mathbf{y}})$ nearly stays the same, there is no reason to make a bigger change of $\mathbf{y}$ in the next iteration to get closer to the optimum. On the other hand, if $\mathbf{y}$ gets changed marginally to $\tilde{\mathbf{y}}$ and $f(\tilde{\mathbf{y}})$ is substantially different from before, the next iteration point is usually further away. In nearly all cases the negative gradient indicates how to choose the next point, so the value of the next initial point is higher.

But the gradient alone is often not enough to find the global optimum. One of the main reasons why optimizers generally fail to find it is that they get stuck in local optima. Unwanted local extrema can be skipped if a sufficiently large step size is involved in an optimization [31, Chap. 3.5].

How different optimization methods implement this and how well they solve the problem will be presented later on in Section 2.3.2. To perform gradient-based optimization, the question of how to determine the directional derivatives of the power conversion efficiency η with respect to the density vector $\mathbf{x}$ will be discussed in the next section.

2.3.1 Gradient Calculation

The objective function is the power conversion efficiency η, calculated in Equation (2.4), and the gradient with respect to the density vector $\mathbf{x}$ of this function is required. The gradient can be expressed as

$$\frac{\mathrm{d}\eta_{\mathrm{total}}^{\mathrm{cell}}}{\mathrm{d}\mathbf{x}} = -\frac{1}{\mathrm{p}_{\mathrm{light}}} \cdot \frac{\mathrm{dp}_{\mathrm{cell}}}{\mathrm{d}\mathbf{x}}, \tag{2.47}$$

since $\mathrm{p}_{\mathrm{light}}$ is a constant value. The derivation of the electrical power $\mathrm{p}_{\mathrm{cell}}$ is given by:

$$\frac{\mathrm{dp}_{\mathrm{cell}}}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{d}\left(\mathrm{V}_{\mathrm{op}} \cdot I\right)}{\mathrm{d}\mathbf{x}}$$

Using the product rule and Equation (2.32), the following term is obtained:

$$\frac{\mathrm{dp}_{\mathrm{cell}}}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{dV}_{\mathrm{op}}}{\mathrm{d}\mathbf{x}} \cdot \sum_{i\in \mathrm{I}} \mathbf{i}_i^{\mathrm{gen}}\left(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}\right) + \mathrm{V}_{\mathrm{op}} \cdot \frac{\mathrm{d}\left(\sum_{i\in \mathrm{I}} \mathbf{i}_i^{\mathrm{gen}}\left(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}\right)\right)}{\mathrm{d}\mathbf{x}}$$

The generated current $\mathbf{i}^{\mathrm{gen}}$ depends on $\mathbf{x}$, because $\mathbf{x}$ is used to calculate the photocurrent $\mathbf{i}^{\mathrm{ph}}$ and the voltage $\mathbf{u}$. V_{op} is a constant value, so its derivative vanishes. Thus,

$$\frac{\mathrm{dp}_{\mathrm{cell}}}{\mathrm{d}\mathbf{x}} = \mathrm{V}_{\mathrm{op}} \cdot \frac{\mathrm{d}\left(\sum_{i\in \mathrm{I}} \mathbf{i}_i^{\mathrm{gen}}\left(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}\right)\right)}{\mathrm{d}\mathbf{x}}.$$

The chain rule is used to rewrite the term to

$$\frac{\mathrm{dp}_{\mathrm{cell}}}{\mathrm{d}\mathbf{x}} = \underbrace{\mathrm{V}_{\mathrm{op}} \cdot \frac{\mathrm{d}\left(\sum_{i\in \mathrm{I}} \mathbf{i}_i^{\mathrm{gen}}\left(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}\right)\right)}{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}}_{\frac{\partial \mathrm{p}_{\mathrm{cell}}}{\partial \mathrm{d}\mathbf{i}^{\mathrm{gen}}}} \cdot \frac{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathrm{p}_{\mathrm{cell}}}{\partial \mathbf{i}^{\mathrm{gen}}} \cdot \frac{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}{\mathrm{d}\mathbf{x}}.$$

The partial derivative of the power $\mathrm{p}_{\mathrm{cell}}$ with respect to $\mathbf{i}^{\mathrm{gen}}$ is the operating voltage V_{op} in each component. As mentioned earlier, the generated current depends on the photocurrent $\mathbf{i}^{\mathrm{ph}}$ and the voltage $\mathbf{u}$. Hence, the total derivative can be written as

$$\frac{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{i}^{\mathrm{gen}}}{\partial \mathbf{i}^{\mathrm{ph}}} \cdot \frac{\mathrm{d}\mathbf{i}^{\mathrm{ph}}}{\mathrm{d}\mathbf{x}} + \frac{\partial \mathbf{i}^{\mathrm{gen}}}{\partial \mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}}. \tag{2.48}$$

The partial derivatives of $\mathbf{i}^{\mathrm{gen}}$ with respect to $\mathbf{i}_k^{\mathrm{ph}}$ and $\mathbf{u}_k$ in Equation (2.48) are determined in Equation (2.49) with the notation of [28], $\mathbf{u}_k = \mathrm{V}$ and $W_0(h(\mathbf{u}_k))$ is the Lambert W-function [32]. Hence,

$$\begin{aligned} \frac{\partial \mathbf{i}^{\mathrm{gen}}}{\partial \mathbf{i}_k^{\mathrm{ph}}} &= \frac{\mathrm{n} \cdot \mathrm{a} \cdot V_T \cdot \frac{R_s \cdot R_{sh} \cdot h(\mathbf{u}_k)}{(R_s + R_{sh}) \cdot \mathrm{n} \cdot \mathrm{a} \cdot V_T} \cdot W_0(h(\mathbf{u}_k))}{R_s \cdot (h(\mathbf{u}_k) \cdot W_0(h(\mathbf{u}_k)) + h(\mathbf{u}_k))} - \frac{R_{sh}}{R_s + R_{sh}} \\ \frac{\partial \mathbf{i}^{\mathrm{gen}}}{\partial \mathbf{u}_k} &= \frac{\left(\frac{1}{R_s} - \frac{1}{R_s + R_{sh}}\right) \cdot W_0(h(\mathbf{u}_k))}{W_0(h(\mathbf{u}_k)) + 1} + \frac{1}{R_s + R_{sh}} \end{aligned} \tag{2.49}$$

where

$$h(\mathbf{u}_k) = \frac{I_0 \cdot R_s \cdot R_{sh}}{(R_s + R_{sh}) \cdot \mathrm{n} \cdot \mathrm{a} \cdot V_T} \cdot e^{\frac{R_s \cdot (R_{sh} \cdot (I_{ph} + I_0) - \mathbf{u}_k)}{(R_s + R_{sh}) \cdot \mathrm{n} \cdot \mathrm{a} \cdot V_T} + \frac{\mathbf{u}_k}{\mathrm{n} \cdot \mathrm{a} \cdot V_T}}.$$

The partial derivative of the components of $\mathbf{i}^{\mathrm{ph}}$, calculated in Equation (2.43), can be determined as

$$\frac{\mathrm{d}\mathbf{i}_k^{\mathrm{ph}}}{\mathrm{d}\mathbf{x}_i} = \begin{cases} j^{\mathrm{ph}} \cdot A_i \cdot \dfrac{\mathrm{d}SIMP_{\mathrm{gen}}(\mathbf{x}_k)}{\mathrm{d}\mathbf{x}_i} & i = k \\ 0 & i \neq k. \end{cases} \tag{2.50}$$

Thus, the derivative of $\mathbf{i}^{\mathrm{ph}}$ is given by a diagonal matrix, where the diagonal entries are the calculated derivatives $\frac{\mathrm{d}\mathbf{i}_i^{\mathrm{ph}}}{\mathrm{d}\mathbf{x}_i}$ in Equation (2.50). Based on the system of equations (2.25)

$$\mathbf{G} \cdot \mathbf{u} = -\mathbf{i}^{\mathrm{gen}}(\mathbf{i}^{\mathrm{ph}}, \mathbf{u}),$$

the total derivative with respect to $\mathbf{x}$ leads to

$$\frac{\mathrm{d}\mathbf{G}}{\mathrm{d}\mathbf{x}} \cdot \mathbf{u} + \mathbf{G} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} = -\frac{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}{\mathrm{d}\mathbf{x}}. \tag{2.51}$$

This expression can be rearranged as

$$\mathbf{G} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} = -\frac{\mathrm{d}\mathbf{i}^{\mathrm{gen}}}{\mathrm{d}\mathbf{x}} - \frac{\mathrm{d}\mathbf{G}}{\mathrm{d}\mathbf{x}} \cdot \mathbf{u}. \tag{2.52}$$

If the right side of the Equation (2.52) is known, then a system of linear equations must be solved to obtain the derivative of the voltage $\mathbf{u}$. One of the missing values on the right side is the derivative of the generated current $\mathbf{i}^{\text{gen}}$. This derivative is calculated in (2.48), but cannot be determined because the derivative of the voltage $\mathbf{u}$ with respect to $\mathbf{x}$ is unknown. In order to avoid a recursive problem, a simplification is made:

$$\frac{\mathrm{d}\mathbf{i}^{\text{gen}}}{\mathrm{d}\mathbf{x}} \approx \frac{\partial \mathbf{i}^{\text{gen}}}{\partial \mathbf{x}} \tag{2.53}$$

Furthermore, the derivative of the stiffness matrix $\mathbf{G}$ is missing. Each entry has to be differentiated. This leads to

$$\frac{\mathrm{d}\mathrm{g}_{ij}}{\mathrm{d}\mathbf{x}_i} = \begin{cases} \dfrac{\mathrm{d}\sigma_{ij}}{\mathrm{d}\mathbf{x}_i} & j \in \mathrm{N}_i \\ -\displaystyle\sum_{j \neq i} \dfrac{\mathrm{d}\sigma_{ij}}{\mathrm{d}\mathbf{x}_i} & j = i \\ 0 & \text{else,} \end{cases}$$

where

$$\frac{\mathrm{d}\sigma_{ij}}{\mathrm{d}\mathbf{x}_i} = -\frac{1}{\mathbf{R}_{ij}^2} \cdot \frac{\mathrm{d}\mathbf{R}_{ij}}{\mathrm{d}\mathbf{x}_i}.$$

The partial derivative of the resistance $\mathbf{R}_{ij}$ with respect to the density vector $\mathbf{x}_i$ is given by:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{R}_{ij}}{\mathrm{d}\mathbf{x}_i} &= \frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{R}_{ij}^{\text{toEdge}}} \cdot \frac{\partial \mathbf{R}_{ij}^{\text{toEdge}}}{\partial \mathbf{R}_{ij}^{\text{grid,toEdge}}} \cdot \frac{\mathrm{d}\mathbf{R}_{ij}^{\text{grid,toEdge}}}{\mathrm{d}\mathbf{x}_i} \\ &= 1 \cdot \left(\frac{\mathbf{R}_{ij}^{\text{TCO,toEdge}}}{\mathbf{R}_{ij}^{\text{grid,toEdge}} + \mathbf{R}_{ij}^{\text{TCO,toEdge}}} \right)^2 \cdot \frac{\rho^{\text{grid,toEdge}} \frac{\left\| \mathbf{x}_{pos_i} - \mathbf{x}_{pos_j} \right\|_2}{2}}{d^{\text{grid,toEdge}} |\partial\Omega_i \cap \partial\Omega_j|} \frac{\mathrm{d}SIMP_{\text{con}}(\mathbf{x}_i)}{\mathrm{d}\mathbf{x}_i} \end{aligned}$$

Due to the constraint presented in Table 2.1, the partial derivative of the stiffness matrix $\mathbf{G}$ with respect to the element that is the external front contact is zero in each component. Now the right side of Equation (2.52) is known, but the system cannot be solved directly, because the right side is a matrix. Therefore, the first column of the solution matrix $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}}$ is determined by the solution vector $\mathbf{y}_0$ of

$$\mathbf{G} \cdot \mathbf{y}_0 = \mathbf{b}_0,$$

where $\mathbf{b}_0$ is the first column of the right side matrix. The remaining columns are filled analogously. In the end, the derivative of the power p_{cell} is given by

$$\frac{\mathrm{d}\mathrm{p}_{\text{cell}}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathrm{p}_{\text{cell}}}{\partial \mathrm{d}\mathbf{i}^{\text{gen}}} \cdot \left(\frac{\partial \mathbf{i}^{\text{gen}}}{\partial \mathbf{i}^{\text{ph}}} \cdot \frac{\mathrm{d}\mathbf{i}^{\text{ph}}}{\mathrm{d}\mathbf{x}} + \frac{\partial \mathbf{i}^{\text{gen}}}{\partial \mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} \right), \tag{2.54}$$

where Equations (2.50) and (2.52) are used. The total derivative of the efficiency η can be determined by substituting Equation (2.54) into Equation (2.47).

Verification of Gradient

Since the simplification of the analytical calculation of the gradient, in the following nevertheless called analytical gradient, is complex and requires several matrix multiplications in addition to solving a system of equations, the question arises why the more accurate numerical gradient, calculated with the difference quotient, is not used right away. The difference quotient would require $(n+1)$ function evaluations, where n is the number of elements. Since the calculation of the power conversion efficiency η has to be evaluated expensively, this leads to enormous run times. The runtimes of the two variants can be seen in Figure 2.4.

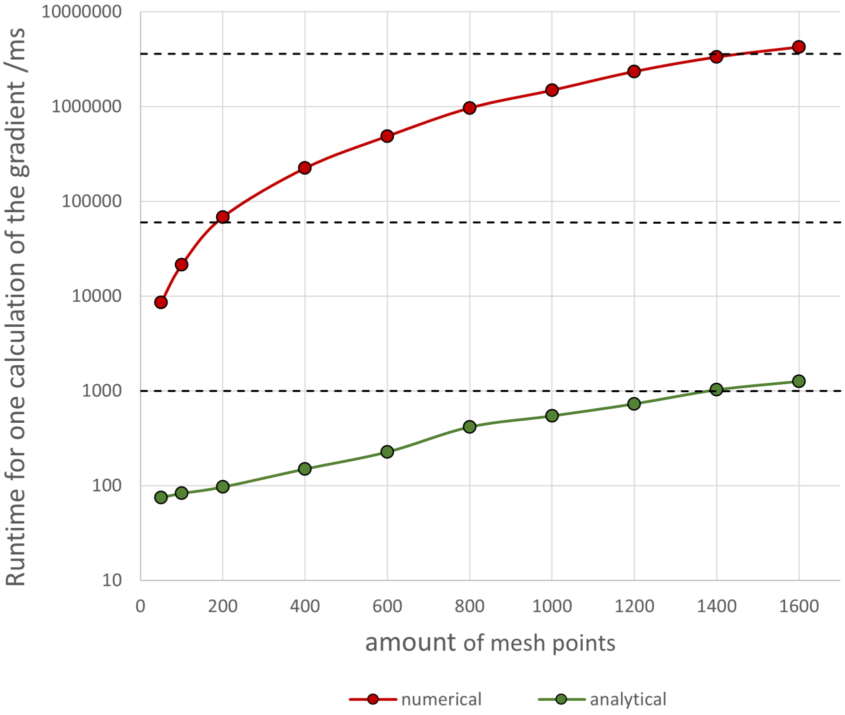


Figure 2.4: Runtime of the numerical and the analytical gradient.

To check whether the analytical gradient with the simplification made in Equation (2.53) is applicable at all, the values of the analytical gradient are compared with those of the numerical gradient. For the validation, the gradient, the subdomains of the solar cell are colored according to the values of the gradient. If the analytical and the numerical gradients match, the images should be nearly identical. This is the case in Figure 2.5. Thus, the analytical and the numerical gradients have almost everywhere the same sign. However, the values of the gradients are in different orders of magnitude. The maximum value of the analytical gradient is approximately $4.8 \cdot 10^5$ and the minimum value is -0.003143. On the other hand, the numerical gradient is within [-0.0954,0.0177]. The problem of scaling can be solved in the optimization by controlling the step size of an optimizer. Therefore, the analytical gradient with the simplification gives the direction of the steepest descent and can be used for optimization.

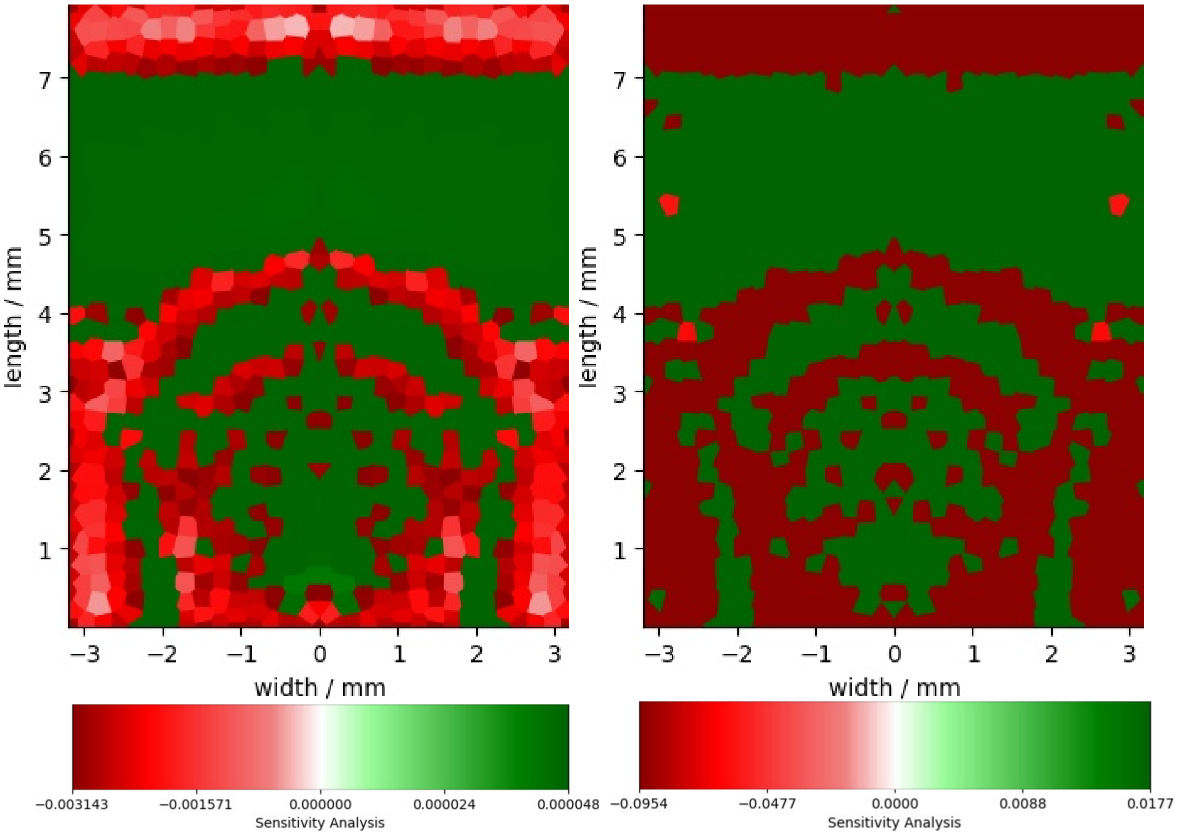


Figure 2.5: Comparison of the analytical (left) and the numerical (right) gradient.

2.3.2 Optimizers

The optimization in this section is performed with gradient based optimizers. It is well known that steepest descent methods are inefficient in material distribution problems, as the topology optimization of a bone remodeling with a steepest descent method from 2010 shows [33]. Nevertheless, in this thesis it will be shown that these optimization methods based on steepest descent can also be efficient. The gradient descent method and its variants, which have been successfully applied to large-scale optimization problems for machine learning [34, 31, Chap. 3.5], are characterized by a simple update rule as well as a high convergence rate. On the other hand, often, only small steps are done and this results in a slower convergence speed. To reduce the number of iterations required, the optimizer can be improved. While some of these possible adjustments will be discussed directly following the introduction of the optimizers, other methods will be considered later in Chapter 3.

Furthermore, the optimizers used to train neural networks are suitable for problems with a high number of degrees of freedom. Along with this, the optimizers also perform well on non-convex problems.

This is also true for the problem that is to be optimized. For this reason, gradient descent and various adaptations such as the adaptive moment estimation (Adam) algorithm [35] are discussed. In contrast, the Broyden-Fletcher-Goldfarb-Shanno

(BFGS) algorithm [36, 37, 38, 39] is presented, which is often used in the context of topology optimization, e.g., see Abert [40].

The individual optimizers are presented below. The first optimizer considered is the comparatively simple to implement gradient descent method. Subsequently, this method is further developed by simple adjustments such as normalization or component-wise normalization of the gradient. This is continued until one ends up with the Adam method [31, Chap. 3]. In comparison, the BFGS method is tested as an optimizer using an approximation for the Hessian matrix.

For each optimizer, the algorithm used to obtain a better grid pattern in each iteration is presented. Then, using these update rules for the density vector $\mathbf{x}$, an optimization is performed to maximize the power conversion efficiency η. The results of the individual optimizers are then compared on the basis of the quality, i.e., the power conversion efficiency η, the number of iterations required until convergence and an empirical rating of the design of the resulting grid patterns. It should be mentioned that there is no convergence criterion. The number of iterations is set high enough so that convergence occurs in any case. However, if the number of iterations is too large, this optimizer will not be considered in the further decision, which optimizer should be developed in the next chapter, since its effort is significantly higher than that of the others. In retrospect, the number of required iterations until convergence is set to the iteration from which on there are no significant changes of the density vector $\mathbf{x}$ as well as the power conversion efficiency η. The initial guess for the optimization is the i^{th} unit vector, with the external front contact attached to the i^{th} element.

Furthermore, the problem to be optimized is a non-convex problem. This means that not every local maximum is also a global maximum. This is also accompanied by the fact that changes to parameters can lead to a different local maximum. And thus also the optimizers all generate different grid patterns. The color code for each figure of grid patterns within this thesis is the one defined by the colorbar in Figure 2.6.

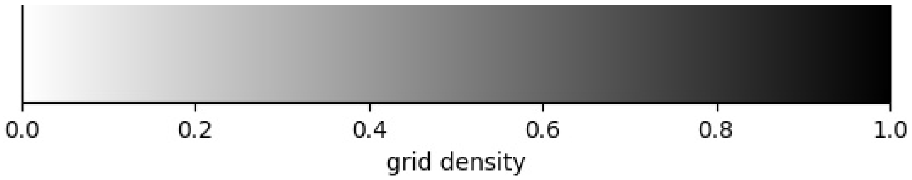


Figure 2.6: Colorbar for figures of grid patterns.

At the end of this chapter, the implementation of the optimizer that performed best in the comparison is presented.

2.3.2.1 Gradient Descent

One of the simplest gradient-based optimizers is the gradient descent method. Gradient descent is an algorithm that is one of the most popular algorithms to perform optimization [34, 31, Chap. 3.5].

In general, gradient descent is an iterative optimizer and in each iteration the input parameter $\mathbf{z}$ is updated based on this scheme,

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \lambda_i \cdot \nabla f(\mathbf{z}_i), \tag{2.55}$$

where λ_i is an parameter to adjust the step size, i.e., how big the change of $\mathbf{z}$ should be in each iteration. In using the negative gradient to update the input parameter, one moves in any case in the direction of the steepest descent, i.e., toward the nearest minimum [31, Chap. 3].

Since the optimization problem in the context of this thesis to find the best density vector $\mathbf{x}$ for a high power conversion efficiency η is a maximization problem, the gradient descent algorithm has to be adjusted. Instead of finding a minimum a maximum is required. To achieve this, the step size λ_i must be multiplied by -1. This leads to following update scheme:

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda_i \cdot \nabla f(\mathbf{z}_i) \tag{2.56}$$

In general, if a fixed step size λ is chosen too small, the risk of ending up in a local maximum for a non-convex problem increases. For both convex and non-convex problems, optimization with a small step size can require many iterations. If the step size is too large, the global maximum may be skipped each time and results in oscillation [31, Chap. 3]. These problems are visualized based on the maximization of an example function in Figure 2.7.

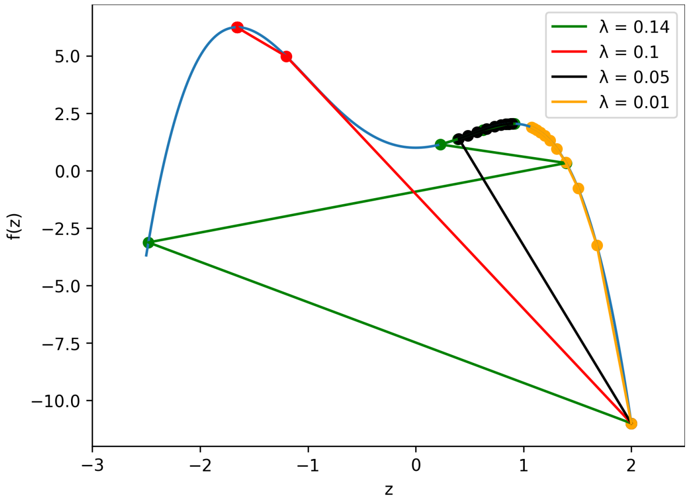


Figure 2.7: Comparison of different fixed step sizes λ used to optimize f$(z) = -z^4 - z^3 + 3z^2 + 1$: $\lambda = 0.01$: Converges too slow. $\lambda = 0.05$: Converges in local maximum. $\lambda = 0.10$: Converges in global maximum. $\lambda = 0.14$: Skips global maximum and converges in local maximum.

Near stationary points such as a maximum or a saddle point, the values of the gradient becomes very small. This leads to a slower and slower convergence to these points. In order to prevent the ever slower convergence rate, the gradient can be normalized before it is employed. Thus it is achieved that in each iteration the total change for a fixed step size remains the same. The change is the step size λ_i [31, Chap. A] and the update rule is

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda_i \cdot \frac{\nabla f(\mathbf{z}_i)}{\left\|\nabla f(\mathbf{z}_i)\right\|_2}. \tag{2.57}$$

At the beginning of the optimization, this adaptation stops the optimizer from moving too fast in one direction as long as a stationary point is still further away. Towards the end of the optimization, however, the stationary point can be skipped with this adjustment because of the step length remaining the same. Therefore, an adaptive step length should be used [31, Chap. A].

For some objective functions it is useful to enhance the smaller values of the gradient. If the objective function contains regions that are flat and is steep only with respect to some components and the gradient is normalized as in Equation (2.57), some values of the gradient can be nullified [31, Chap. A]. This can lead to a local maximum, because the optimizer first considers only the steepest descent.

Instead of normalizing each partial derivative by the magnitude of the entire gradient, each partial derivative can be normalized only with respect to itself:

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda_i \cdot \operatorname{sgn}\left(\nabla f(\mathbf{z}_i)\right) \tag{2.58}$$

Then the density vector $\mathbf{z}$ is always adjusted as it would currently fit best for each component [31, Chap. A].

The results of the self implemented gradient descent variations optimizing the power conversion efficiency η of a solar cell are illustrated in Figure 2.8. The top row shows the grid patterns into which the optimization methods have converged. The figures in the bottom row of Figure 2.8 represent the final grid patterns after setting the values of the density vector $\mathbf{x}$ to 0 and 1 depending on the threshold value of 0.1.

The step size for the gradient descent algorithm is $\lambda_i = 100000$. The final result after 200 iterations can be seen in the upper left. Due to the large step size λ, there is a very large change in the density vector $\mathbf{x}$ in each iteration. The oscillation of the density vector $\mathbf{x}$ is reflected in the grid pattern. The power conversion efficiency η that is achieved with this grid pattern is 21.95 %.

In the center of the upper row, the final result of the normalized gradient descent optimization can be seen. The number of iteration is 200 and the step size used here is $\lambda_i = 4$. In contrast to the basic gradient descent method, a clear structure can already be guessed. However, there are also many gray areas in this image, which show that even with this method the optimum has not yet been found. The achieved power conversion efficiency η is 22.03 %.

The stright grid arms achieved with the componentwise normalized gradient descent method stands out compared to the other results. Unfortunately, around 300 iterations were also necessary for it. The resulting grid pattern can be found in the upper right image. In each iteration, each component changed by $\lambda_i = 0.005$ depending on the gradient's sign function. It is also noticeable that the grid arms are shorter than those of the normalized gradient descent. Overall, this structure has a power conversion efficiency η of 21.79 %.

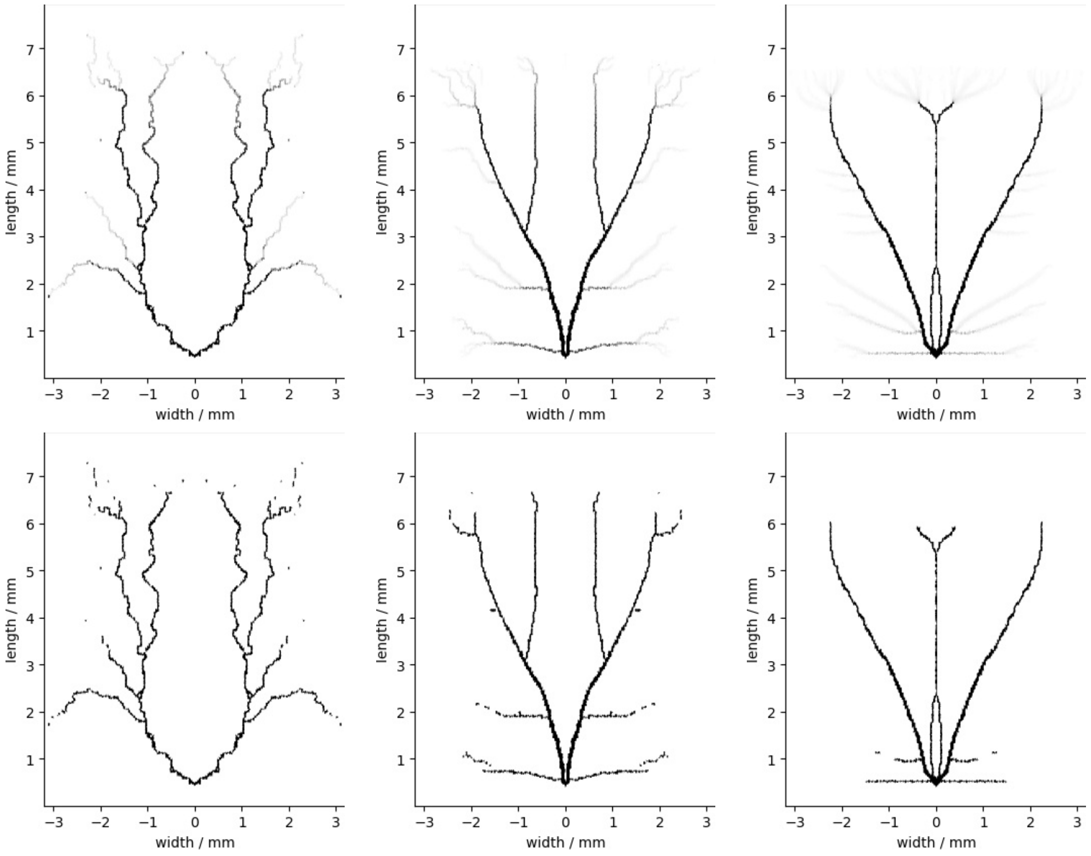


Figure 2.8: Left: Gradient Descent, center: Normalized Gradient Descent, right: Componentwise Normalized Gradient Descent, top: Optimization results, bottom: Result after binarization.

2.3.2.2 Adaptive Moment Estimation

The adaptive moment estimation (Adam) algorithm [35] is an algorithm that is based on the properties and methods of already known optimizers. The Adam method owes its origin to the AdaGrad method [41] and the RMSProp method [42]. However, the basic idea remains that of the gradient descent method, introduced in Section 2.3.2.1. Accordingly, the update scheme to maximize a general objective function $f(\mathbf{z})$ looks similar to

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda \cdot \alpha_i, \tag{2.59}$$

where λ is the step size and α_i is an update parameter which replaces the gradient. The update parameter α_i depends on the exponential moving averages of the gradient and the squared gradient [35].

The moving average $\mathbf{m}$ of the gradient, that is used within the update rule, is an estimate of the first moment. Basically, this is a concept that includes the gradients from the previous iterations. The estimate of the first moment $\mathbf{m}_{i+1}$ is given by a convex combination of the previous first moment $\mathbf{m}_i$ and the current gradient

$$\mathbf{m}_{i+1} = \beta_1 \cdot \mathbf{m}_i + (1 - \beta_1) \cdot \nabla f(\mathbf{z}_i), \tag{2.60}$$

where $\beta_1 \in [0, 1)$ is a parameter to control the exponential decay rates of the moving average of the gradient [35]. As already said, with the convex combination one achieves that the gradients, which one has calculated in previous iterations, are also taken into account. In the first iteration the estimate of the first moment is initialized as zero vector:

$$\begin{aligned} &\mathbf{m}_0 = \mathbf{0} \\ &\mathbf{m}_{i+1} = \beta_1 \cdot [\beta_1 \cdot \mathbf{m}_{i-1} + (1 - \beta_1) \cdot \nabla f(\mathbf{z}_{i-1})] + (1 - \beta_1) \cdot \nabla f(\mathbf{z}_i) \qquad i \geq 1 \end{aligned}$$

This modification is made to help traverse flatter areas of the objective function and not coming to a stop on a short flat section in between due to its moment. Moreover, the problem of 'zig-zagging' behavior of the gradient descent method can be solved with the use of the moment [31, Chap. A].

Due to the initialization of the estimated first moment, the moving average is biased towards zero [35]. This initialization bias can be counteracted by following transformation:

$$\hat{\mathbf{m}}_{i+1} = \frac{\mathbf{m}_{i+1}}{(1 - \beta_1)^{i+1}} \tag{2.61}$$

As mentioned in the beginning, to calculate the update parameter α, the moving average $\mathbf{s}$ of the squared gradient is also used. This corresponds to the second moment. The calculation of the second moment $\mathbf{s}_i$ in each iteration is similar to the calculation of the first moment in Equations (2.60) and (2.61) [35]:

$$\begin{aligned} &\mathbf{s}_0 = \mathbf{0} \\ &\mathbf{s}_{i+1} = \beta_2 \cdot \mathbf{s}_i + (1 - \beta_2) \cdot (\nabla f(\mathbf{z}_i))^2 \end{aligned} \tag{2.62}$$

As before for the first moment, the initialization bias of the second moment has to be counteracted:

$$\hat{\mathbf{s}}_{i+1} = \frac{\mathbf{s}_{i+1}}{(1 - \beta_2)^{i+1}} \tag{2.63}$$

All together, the update rule of the Adam method is given by Equation (2.59), where α_i is composed of the first ($\hat{\mathbf{m}}$) and second ($\hat{\mathbf{s}}$) moment:

$$\alpha_i = \frac{\hat{\mathbf{m}}_{i+1}}{\sqrt{\hat{\mathbf{s}}_{i+1}} + \varepsilon}, \tag{2.64}$$

where ε is a perturbation so the denominator is non zero.

In the end, the Adam optimization method combines ideas of its precedents and evolves into a better version.

The result of the optimization with a self implemented algorithm of the Adam method is shown in Figure 2.9. In total, the optimization was performed with 200 iterations. As with the gradient descent method, the grid pattern of the penultimate iteration is shown on the left and the binary grid pattern obtained by the threshold of 0.1 is shown on the right.

The parameters used for this optimization are as follows:

$$\lambda = 0.03$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$\varepsilon = \text{1e-9}$$

The special feature of the grid pattern obtained by the Adam method is the extensive coverage of the solar cell with grid. There is barely a spot on the solar cell that has a further path than 2 mm to the next grid arm. This feature pays off with a high power converison efficiency η of 22.08 %.

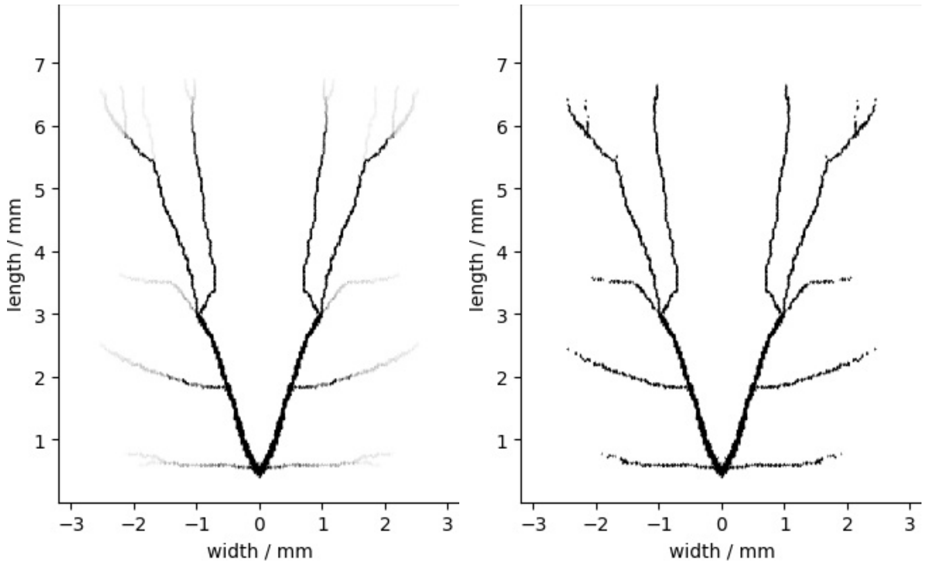


Figure 2.9: Results of optimization with the Adam method. On the left is shown the grid pattern before and on the right after binarization.

2.3.2.3 Broyden-Fletcher-Goldfarb-Shanno

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [36, 37, 38, 39] is an optimizer that is using an approximated second derivative of a general objective function $f(\mathbf{z})$. Thus it belongs to the family of Newton-method optimization algorithms. Since the second derivative is only approximated, BFGS is a Quasi-Newton method.

In general, a quasi-Newton method for a maximization problem has the update scheme [39],

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \lambda \cdot \mathbf{H}_i \cdot \nabla f(\mathbf{z}_i), \tag{2.65}$$

where λ is a scalar to adjust the step size and $\mathbf{H}_i$ is the approximation of the inverse Hessian at $\mathbf{z}_i$. The inverse Hessian matrix is replaced with a secant approximation. The secant method is an iterative algorithm that often has the identity matrix as initial guess and then applies updates with each iteration [43, Chap. 6]. In the BFGS algorithm, the Hessian matrix is updated according to the following scheme [37],

$$\begin{aligned}\mathbf{H}_i^{k+1} = \mathbf{H}_i^k &- \frac{\delta_{k+1} \cdot \gamma_{k+1}^T \cdot \mathbf{H}_i^k + \mathbf{H}_i^k \cdot \gamma_{k+1} \cdot \delta_{k+1}^T}{\delta_{k+1}^T \cdot \gamma_{k+1}} \\ &+ \left(1 + \frac{\gamma_{k+1}^T \cdot \mathbf{H}_i^k \cdot \gamma_{k+1}}{\delta_{k+1}^T \cdot \gamma_{k+1}}\right) \cdot \frac{\delta_{k+1} \cdot \delta_{k+1}^T}{\delta_{k+1}^T \cdot \gamma_{k+1}},\end{aligned} \tag{2.66}$$

where $\quad \delta_{k+1} = \mathbf{z}_{i+1} - \mathbf{z}_i$

$\qquad\quad \gamma_{k+1} = \nabla f(\mathbf{z}_{i+1}) - \nabla f(\mathbf{z}_i).$

To calculate the step size λ, for example, backtracking line search or the line search with Wolfe condition can be used [43, Chap. 4].

In the context of this thesis, to optimize the density vector $\mathbf{x}$ to maximize the power conversion efficiency η, the implemented version of the Extreme Numerics library [29] is tested. The default parameters of Extreme Numerics are used for the optimization, where the convergence criterion is defined by a maximum number of 200 iterations. The optimized grid pattern before binarization can be seen in Figure 2.10 (left) and after binarization on the right. With this grid pattern, a power conversion efficiency η of 20.25 % is obtained.

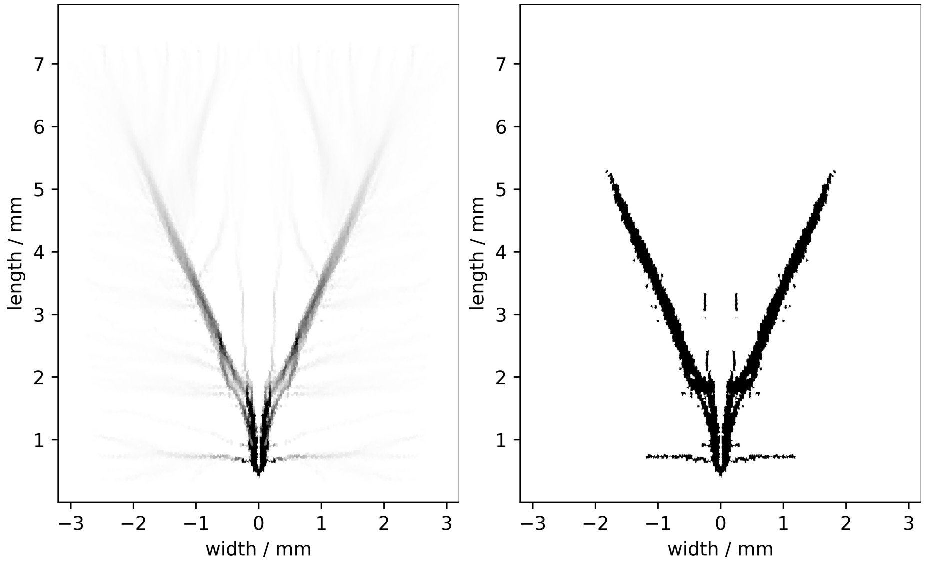


Figure 2.10: Results of optimization with the BFGS method. On the left is shown the grid pattern before and on the right after binarization.

2.3.3 Comparison of Optimizers

In the further course of this thesis, only one optimizer will be used. The decision which optimizer to use depends on the quality of the resulting grid patterns (PCE η), the runtime per iteration (∅ Runtime/It.), the number of iterations needed (No. of It.), and the potential that lies in an optimizer. The gradient descent method (GD), the normalized GD method (NGD), the componentwise NGD method (CNGD), the Adam method and the BFGS method are compared in Table 2.4.

Optimizer	**PCE η**	**∅ Runtime/It.**	**No. of It.**	**Total Runtime**
GD	21.95 %	41.6 sec	195	8112 sec
NGD	22.03 %	35.6 sec	159	5660.4 sec
CNGD	21.79 %	37.1 sec	300+ it.	11130+ sec
Adam	22.08 %	37.1 sec	140	5194 sec
BFGS	20.25 %	49.34 sec	114	5624.76 sec

Table 2.4: Comparison of optimizer: Gradient descent (GD), normalized GD, componentwise NGD, Adam and BFGS, see Table 2.2 for the configuration of the solar cell and see Table 2.3 for the system used to calculate the results.

Since the BFGS method gives by far the worst result, this method will not be considered further in the following.

It is remarkable that the average runtime per iteration of the optimizers are almost the same. The reason for this is that the effort for all calculations that have to be done in one update step is almost equal to the effort for calculating the gradient. For each optimizer the gradient must be calculated in each iteration, which is why the similar runtimes occur. This means that only the number of required iterations is decisive for the evaluation of the total time of an optimization. As explained in Section (2.3.2), the optimization is terminated after a fixed number of iteration, given by the maximum iteration. Subsequently, the change of the density vector $\mathbf{x}$ as well as the change of the power conversion efficiency $\mathbf{x}$ have been taken as an indicator for the convergence of a method. The iteration from which there is no relevant change is given as the number of iterations needed until convergence. This prevents too early termination and thus minimizes the termination error.

Looking at the resulting grid pattern created with the componentwise normalized gradient descent method in Figure 2.8 before setting the density vector values to 0 and 1, there are hardly any gray areas to be seen. Accordingly, there are no more elements in the rear part of the solar cell that would perform better with grid. And also the lengths of the grid arms compared to those received by the Adam method and the normalized gradient descent method are significantly shorter. It can be concluded that the shorter grid arms lead to a worse power conversion efficiency η. Since the componentwise normalized gradient descent method is converged, it can be assumed that it is stuck in a local maximum. Based on the update rule, it follows that a larger step size is needed to skip the local maximum. However, increasing the step size leads to an oscillating density vector $\mathbf{x}$. Thus, a better result is not to be expected.

Unlike the other optimization methods, the grid pattern achieved with the gradient descent method has a different basic structure. Hence, it can be concluded that the gradient descent method finds with a large step size another local maximum, while the others all land in almost the same local maximum. Unfortunately, this local maximum is worse in terms of power conversion efficiency η as well as in terms of the shape of the grid pattern.

This makes the decision between the normalized gradient descent method and the Adam method. Both can be considered due to the almost similar high power conversion efficiency η achieved with their grid patterns. But since the Adam method converges 466.4 sec faster, this optimizer is used for further optimizations.

2.3.4 Implementation of the Adam Method

In the following, the implementation of the Adam method and the framework of the optimization is given, which is used to optimize the density vector $\mathbf{x}$ for which a maximum power conversion efficiency η is obtained. Extreme Numerics [29] library has been used for mathematical operations.

The implementation of the Adam method in Listing 2.1 is based on the update rule introduced in Section 2.3.2.2. Within the implementation, the through optimization received density vector $\mathbf{x}$ is in $\mathbb{R}^n$. So that the density vector $\mathbf{x}$ is in $[0,1]^n$, the value of components less than 0 are set to 0 and the value of components greater than 1 are set to 1. In Listing 2.2, the Adam method is included in the framework of optimization. Within the framework of optimization, important parameters such as the maximum number of iterations or the number of mesh points are specified.

Then, the discretization discussed in Section 2.1 is performed. The initial guess becomes the i^{th} unit vector, where the external front contact is applied to the i^{th} element. According to the maximum number of iterations, the Adam method updates the density vector $\mathbf{x}$ in each iteration.

Listing 2.1: Adam Method

```
private (Vector<double>, Vector<double>) Optimizer_Adam(
    densityVector, gradient, previousGradient, iteration)
{
    var d = 0.9 * previousGradient + 0.1 * gradient;
    var h = 0.999 * previousGradient.ElementwisePow(2)
            + 0.001 * gradient.ElementwisePow(2);
    d /= (1 - Math.Pow(0.9, iteration + 1));
    h /= (1 - Math.Pow(0.999, iteration + 1));

    var stepSize = 0.03;
    densityVector = (densityVector + stepSize
            * d.ElementwiseDivideInPlace(h.ElementwisePow(0.5)
            + 1e-9)).MinInPlace(1).MaxInPlace(0);

    return (densityVector, gradient);
}
```

Implementation of the Adam method in C# with Extreme Numerics Library [29].

Listing 2.2: Optimization Loop

```
//set max iteration
var maxIteration = 200;

//initialize mesh points
var numberPoints = 20000;
var points = EquidistantDistribution(numberPoints);

//create discretization
var mesh = VoronoiDiagram(points);

//set initial guess (only grid at external front contact)
var densityVector = new Vector.Create(numberPoints)
    {0,...,0,1,0,...,0};

//initialize previousGradient
var gradient = new Vector<double>(numberPoints)

//Optimization loop, ends after max iteration
for (int iteration = 0; iteration < maxIteration; iteration++)
    var efficiency = FunctionValue(densityVector);
    var gradient = CalculateGradient(densityVector);
    (densityVector, previousGradient) =
            Optimizer_Adam(densityVector, gradient,
                previousGradient, iteration)
```

Implementation of Optimization in C# with source code of simulation platform twinPV [3].

3 Modifications

At this point, one could claim to be done with the optimization. It is known how to calculate the efficiency η, and it is also known how to determine the gradient in order to optimize the density vector $\mathbf{x}$, on which the grid pattern is based, with gradient-based optimizers. For further optimization the Adam method is used with the parameters from Section 2.3.2.2.

The first question that arises when considering the results from Section 2.3.2 is whether the grid proposed there can really be used. The answer to this question is most likely no. On the one hand, it must be possible to connect an external front contact to the solar cell. This is provided by a contact pad discussed in Section 3.1.1. On the other hand, the grid must not have any interruptions, entanglements or islands, i.e., isolated spots without a connection to the main structure. In the Sections 3.1.2.1 and 3.1.2.2 a density filter and a method of image processing, the Gaussian blur, are explained with which one obtains smoother grid patterns. [44]

To enable local optimization, it is common to keep certain elements constant and to change only a fraction of all elements at once. The elements that are optimized are located in a so-called batch. The batch size can be adjusted adaptively and the selection criterion can vary. The performance of the selected optimizer presented in Section 2.3.2 is compared in Section 3.2 with the same optimizer but this time using the batch approach.

As mentioned in Chapter 2.2, the Solid Isotropic Material with Penalization (SIMP) approach is used to interpolate the intermediate values of the density vector $\mathbf{x}$ during the optimization process. In Section 3.3 different density functions are tested and the advantages and disadvantages are discussed.

Usually, the initial guess of an optimization algorithm already determines the number of iteration needed until convergence and also the quality of the optimization. There are many possibilities to set the density vector $\mathbf{x}$ as initial guess. One could be the added contact pad or the currently used grid pattern. A good initial guess can also be achievable by a nonlinear localization strategy. The results of setting the initial guess with and without a local preoptimization are presented in Section 3.4.

3.1 Technical Restrictions

As is so often the case, many simulation models cannot be completely transferred to reality. This is because models often only cover a certain part of the problem. For example, the previous optimizations of the grid pattern have only taken into account where the external front contact has been located and how the power conversion efficiency η has been calculated on a specified geometry. Apart from that, there have been no technical restrictions in the optimizations carried out so far.

First, the model of a solar cell should be completed. To do this, a contact pad is added. It will be seen that such a restriction is a big intervention in the optimization. Furthermore, it must be possible to apply the grid to a solar cell in the laboratory. For this reason, the grid components must meet certain criteria. Since these criteria are not necessarily met even before adding the contact pad, a solution must be found. An approach to solve this problem will be presented afterwards.

3.1.1 Contact Pad

It has been assumed that the external front contact can be represented by an element obtained through the discretization. However, the discretization can be chosen arbitrarily fine to get closer to reality. But the possibility to get the current from outside exactly at this point becomes more improbable. In order to have enough space to place the contact without damaging the solar cell, according to [45] a grid area of approximately 800x500 μm^2 is placed at the location of the external front contact, see Section 2.3 for geometry. Figures 3.1 and 3.2 show that an optimization where the contact pad is added from the beginning and cannot be changed during the optimization leads to a large intervention of the optimization behavior. In Figure 3.1, the grid structures in the surroundings of the external front contact are compared. On the left side the optimized surroundings of the external front contact from Section 2.3.2 is shown and on the right side the contact pad as initial guess.

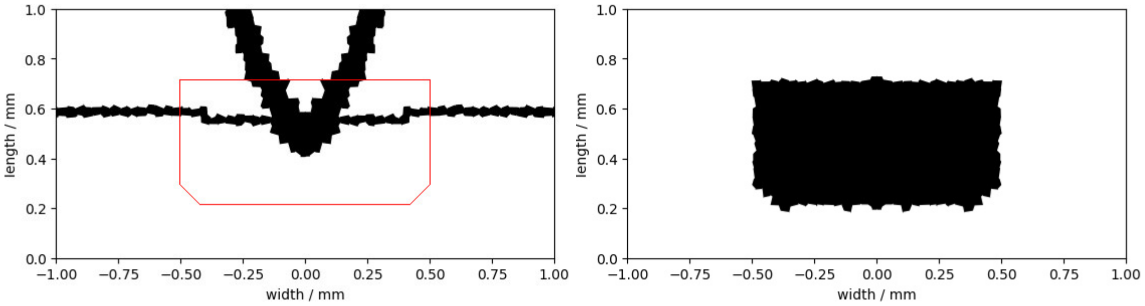


Figure 3.1: Comparison of optimized external front contact and contact pad.

Since the contact pad requires significantly more grid in this area, it is not surprising that the optimization with the contact pad is different. To ensure that the contact pad is not changed during optimization, the components of the density vector **x** representing the contact pad are set constant to 1. Thus also the partial derivatives with respect to these components are 0. Apart from the contact pad, the framework of optimization from Section 2.3.2 remains the same.

The result of this optimization can be seen in Figure 3.2. The obtained grid structure unexpectedly looks similar to the one before adding the contact pad. Both patterns have more or less three main grid arms from which further grid arms branch off. This also means that the property of covering the solar cell with grid over a large area and thus efficiently conducting the current from every corner has been preserved.

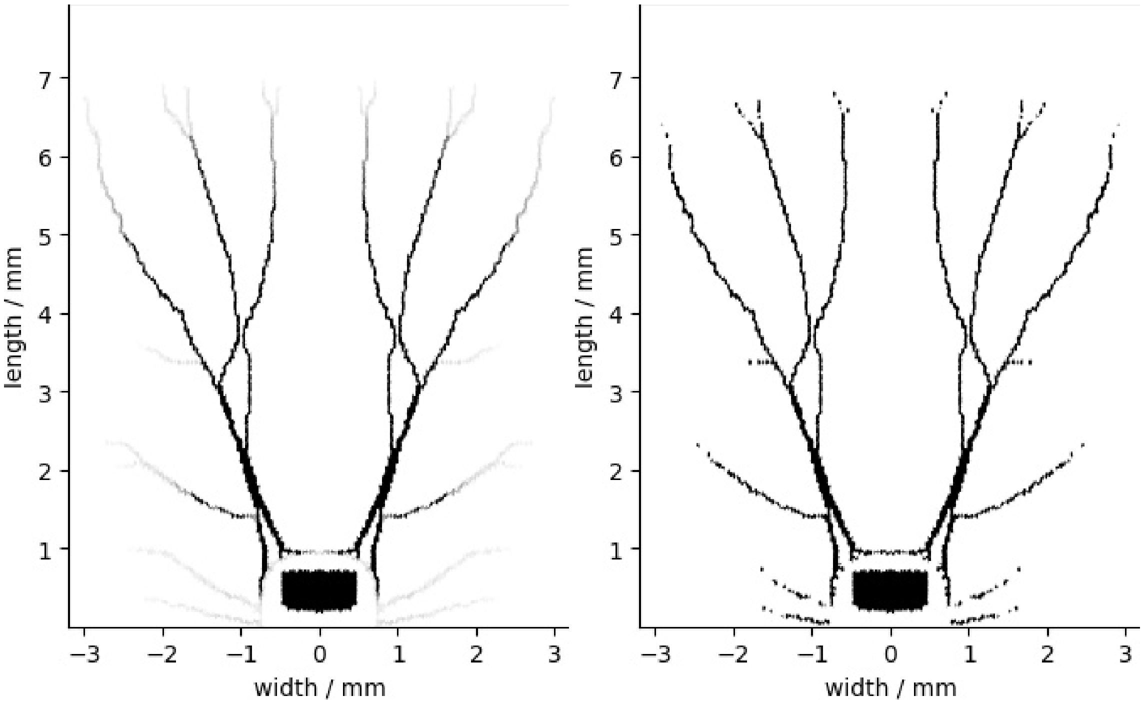


Figure 3.2: Result of the Adam method with contact pad before binarization on the left and after binarization on the right.

Apart from the fact that the grid arms are much closer together in comparison to the grid patterns in Section 2.3.2, another problem has arisen. The grid arms, which should bring the current from the more distant parts of the solar cell to the external front contact, do not have a direct connection. The contact pad is completely isolated.

The power conversion efficiency η achieved with this grid pattern is 19.09 %. And so, after adding the contact pad, the efficiency is approximately 3 percentage points worse than before.

The next step in the optimization would be to solve the problem with the isolated contact pad. Fortunately, there are filters for this that can close the gaps.

3.1.2 Filtering

So far another problem of the optimized grid patterns is the way a solar cell gets its grid coating. To apply the grid structure on a solar cell, masks are created through which the material is coated. The manufacturing of these masks has its own limitations. One of these limitations is that it is not possible to put on the mask in several parts. This means that the mask must be available as a complete component. As a result, no areas enclosed by grid are allowed. Likewise, it is not practical to apply grid to areas that have no connection to the external front contact. In addition, mask production is only feasible with great effort, which also results in high costs. For this reason, grid structures are kept simple. However, the structures obtained by optimization are anything but trivial.

By filtering, the grid structure can be smoothed. This can be done in two ways. On the one hand, the filter can be included in the optimization, i.e., in each iteration, and on the other hand, the density vector $\mathbf{x}$ can be manipulated between iterations. In the first case, the derivative of the filter operation is included in the gradient calculation. In the second case, the filtering is applied and the modified grid structure is used as a new initial guess in the next iteration. Both filter methods are used during optimization. This allows post optimization of the filtered grid pattern.

Hereafter, two filters are presented. One has been developed in the framework of topology optimization, while the other comes from the field of image processing.

3.1.2.1 Density Filter

In the framework of topology optimization, density filtering was introduced by Bruns and Tortorelli [46]. The basic idea of filtering is that the density of each element is redefined as a weighted average of the densities in a neighborhood of the element [44]. In the case of this thesis, it means that if one element on the solar cell has grid, then the neighboring elements will be affected by it and their value in the density vector $\mathbf{x}$ will increase. On the one hand, this is to prevent elements with a grid from being enclosed by elements without a grid. In short, grid should only be added where there is already grid in the surrounding area. On the other hand, sometimes it can be enough to develop only one branch in an area of the solar cell. An example are the dense main branches from Figure 3.2. In principle, it would have been even better if there had been one main branch, which might be thicker, but would develop more reasonable side branches. Depending on how the grid has evolved in an area of the solar cell, a grid branch may be very thin if many independent branches have evolved or more developed if there are few branches within the area. Another reason to use filters is the possibility to create patterns without squiggles.

The filter is applied before the density vector $\mathbf{x}$ is used for the calculation in each iteration. When filtering, each component of the density vector $\mathbf{x}$ is reweighted according to the following scheme [23],

$$\tilde{\mathbf{x}}_i = \frac{1}{\sum\limits_{e \in \mathrm{N}_i} w_{ie}} \cdot \sum_{e \in \mathrm{N}_i} \left(w_{ie} \cdot \mathbf{x}_e \right),$$

where w_{ie} is a weighting function using the Euclidean distance of the i^{th} element to its e^{th} neighbor. The effect of filtering depends on a given filterRadius. The density of the i^{th} element is affected by the density of a neighboring element mainly if the distance between these elements is smaller than the filterRadius. In this thesis two weighting functions are tested:

Linearly decaying function [46] $\quad w_{ie} = \text{filterRadius} - \|\mathbf{x}_{pos_e} - \mathbf{x}_{pos_i}\|_2$

Gaussian distribution function [47] $\quad w_{ie} = \exp\left[-\frac{1}{2} \left(\frac{\|\mathbf{x}_{pos_e} - \mathbf{x}_{pos_i}\|_2^2}{\sigma^2} \right) \right]$

In the case of the Gaussian function σ^2 is the variance. According to [48], the full width at half maximum FWHM for a Gaussian function is calculated by

$$\text{FWHM} = 2 \cdot \sqrt{2 \cdot \ln(2)} \cdot \sigma. \tag{3.1}$$

The FWHM is invariant under translations and so it does not depend on the expected value $\mathbf{x}_{pos_i}$. The elements within the FWHM have a larger influence on the distribution. Thus, the filterRadius can be defined as

$$\text{filterRadius} = \frac{\text{FWHM}}{2}.$$

After reformulation of equation (3.1), the variance σ^2 can be written as follows:

$$\sigma^2 = \frac{\text{filterRadius}^2}{32 \cdot \ln(2)}$$

Since the filter is used in every iteration, it is a fixed part of the calculation and should therefore also be included in the gradient. The filter operation is performed before the calculation and thus the previous gradient discussed in Section 2.3 is the derivative of the power conversion efficiency η with respect to the filtered density vector. Hence, the gradient with respect to the density vector can be written as,

$$\frac{\mathrm{d}\eta}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{d}\eta}{\mathrm{d}\tilde{\mathbf{x}}} \cdot \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{x}},$$

where the components are calculated according to [23]:

$$\frac{\mathrm{d}\eta}{\mathrm{d}\mathbf{x}_i} = \sum_{j \in \mathrm{N}_i} \frac{1}{\sum\limits_{k \in N_j} w_{jk}} w_{ij} \frac{\mathrm{d}\eta}{\mathrm{d}\tilde{\mathbf{x}}_j}$$

In the context of this thesis, the Gaussian distribution function is used as weighting function and the filterRadius is equal to the average distance from mesh point to mesh point and thus only the direct neighbors are included in the filtering. In the case of 20389 elements the average distance from mesh point to mesh point is approximately

$$d(\text{point}_i, \text{point}_j) \approx 0.04\,\text{mm}.$$

For the optimization the Adam method from Section 2.3.2.2 and the contact pad introduced in Section 3.1.1 are used. The resulting efficiency after optimization is 21.53 % and the grid pattern can be seen in Figure 3.3.

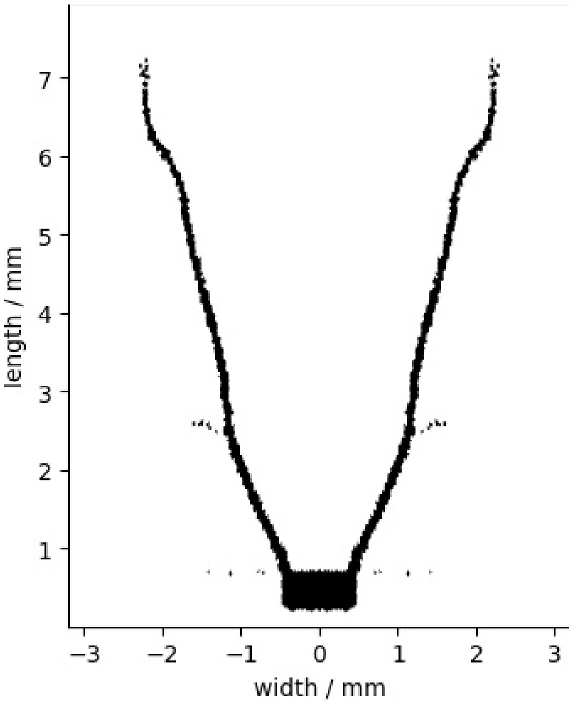


Figure 3.3: Optimization with density filter and Gaussian weighting function.

Compared to Figure 3.2, it is noticeable that the gap at the contact pad could be closed. The effect of filtering is, on the one hand, that no element can have a grid in its neighborhood alone, and on the other hand that the width of a grid arm is equal to the length of at least three elements. One on each side of the middle element. Therefore, the grid arms have a minimum width of 0.12 mm and are much thicker than before. In order to reduce the width of the grid arms during optimization with the density filter, the number of elements must be increased to reduce the length of an element. Since this is associated with a more costly calculation, another filter method is tested.

3.1.2.2 Gaussian Blur

The way Gaussian blur from image processing is applied is quite different from the filters developed in topology optimization, see Section 3.1.2.1. Unlike the presented filter, here all elements are used for filtering the density of an element. Within this thesis, the filter is not used in every iteration, but only after selected iterations. The advantage is that if the optimizer gets stuck, it can be reset and find another grid. By not including the filter in the calculation of the gradient, it is an interference with the monotonically increasing evolution of the power conversion efficiency η during the optimization. As a result, in the next iteration, many gray areas appears in the surroundings of the grid pattern created up to that iteration as illustrated in Figure 3.4. This has the effect that, for example, two neighboring branches are pushed together by the blur.

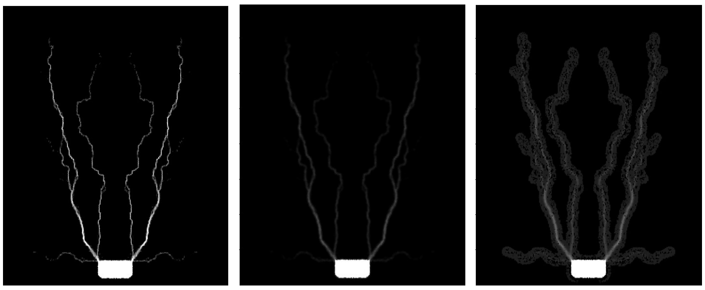


Figure 3.4: Iteration before and after Gaussian blur. The color code is reversed for better visibility.

If the filter is applied, the components of the density vector are reweighted according to this formula,

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{n} \mathbf{x}_j \cdot \exp\left[-\frac{1}{2}\left(\frac{\left\|\mathbf{x}_{pos_j} - \mathbf{x}_{pos_i}\right\|_2^2}{\sigma^2}\right)\right],$$

where the variance σ^2 is calculated as before:

$$\sigma^2 = \frac{\text{filterRadius}^2}{32 \cdot \ln(2)}$$

To normalize the values, the filtered values of the density vector $\hat{\mathbf{x}}$ are divided by the maximum of it:

$$\tilde{\mathbf{x}_i} = \frac{\hat{\mathbf{x}}_i}{\max(\hat{\mathbf{x}})}$$

With the parameters for the Adam algorithm from Section 2.3.2, the contact pad added in Section 3.1.1 and applied Gaussian blur in iterations 18 and 50 with a filterRadius of 0.08 mm the resulting grid pattern is shown in Figure 3.5. The power conversion efficiency η achieved with this grid pattern is similar to the efficiency obtained with the Adam method in Section 2.3.2.2. In Section 3.1.1 it has been shown that the addition of the invariant contact pad leads to a much worse power conversion efficiency η. With the help of Gaussian blur in iterations 18 and 50 a similar power conversion efficiency η as before can be obtained, although the contact pad has been added as initial guess. This indicates that this method is an effective tool for optimization and should be used within further optimizations. The obtained power conversion efficiency η is 22.08 %.

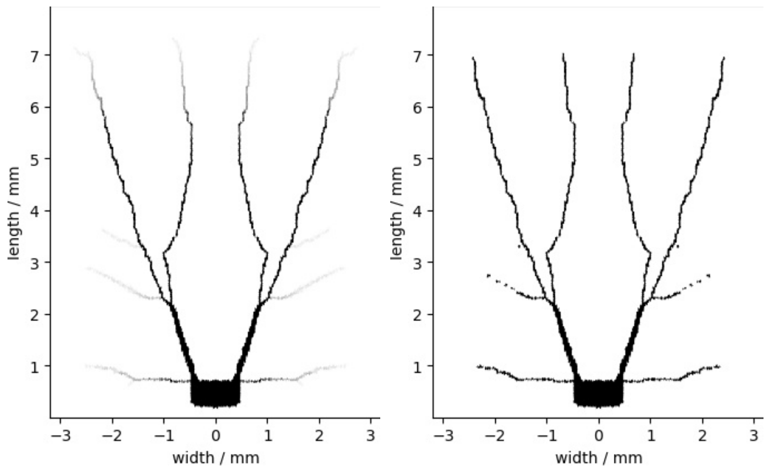
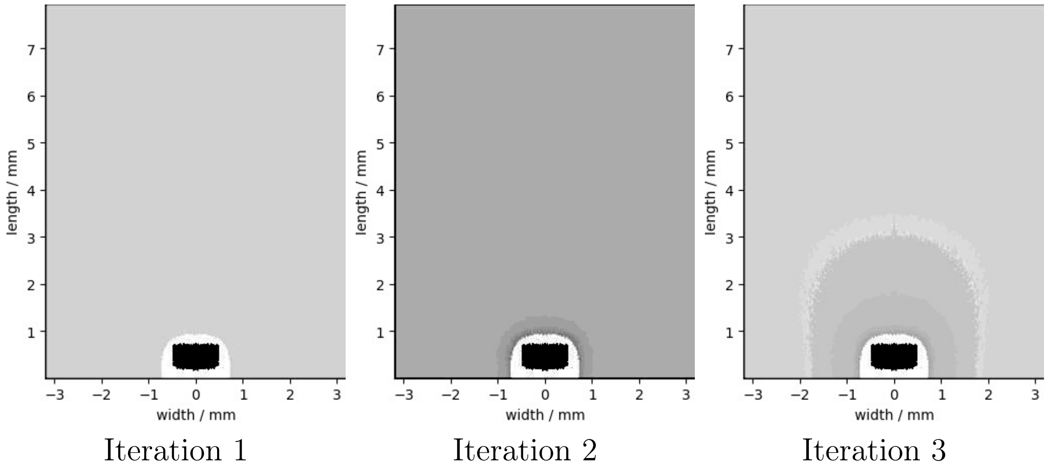
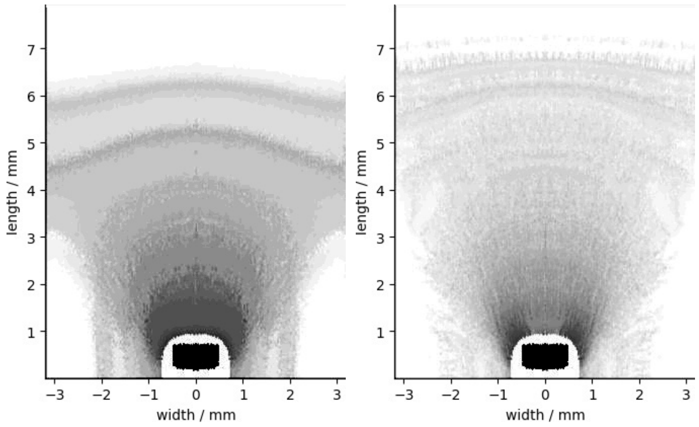


Figure 3.5: Grid patter achieved from optimization with applied Gaussian blur in iteration 18 and in iteration 50 before binarization on the left and after binarization on the right.

3.2 Batch Optimization

Looking at the first iterations of the optimization with the framework presented at the end of Section 3.1.2.2 in Figure 3.6, the question arises whether more clever optimization can be done at the beginning.





Iteration 8 Iteration 15

Figure 3.6: First iterations of the optimization presented in Section 3.1.2.2. The figures have been edited for better recognition. The coloring does not correspond to the colorbar in Figure 2.6 and the different figures do not share the same coloring.

In the first iteration, the density of all elements except the density of the elements in the surroundings of the contact area is increased because the entire solar cell lacks grid. The change is not enough in the first iteration, so more grid is added in the second. In the third iteration, more grid is added near the contact pad, while grid is removed further away from the contact pad. The trend visible in the first 3 iterations continues, as can be seen in iteration 8, before grid arms slowly begin to develop from iteration 15 onwards.

One way to prevent many elements from changing in the same direction in an iteration is to not consider all elements at once. To do this, the elements are divided into different batches B_i. A batch is a subset of the set of all elements E. It is determined how many batches there should be, and then the elements are randomly assigned to the batches until every element is assigned to a batch as shown below.

The number of elements per batch m can be determined from the number of elements $\mathrm{n} \in \mathbb{N}$ and the number of batches $\mathrm{b} \in \mathbb{N}$:

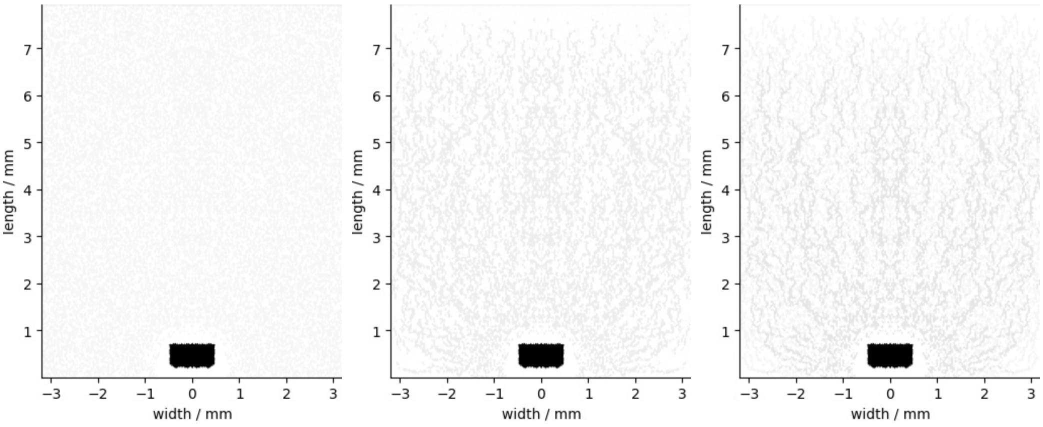
$$\mathrm{m} = \left\lceil \frac{\mathrm{n}}{\mathrm{b}} \right\rceil$$

Let $B_i \subset E$ and $B_i \cap B_j = \emptyset$ for $i \neq j$, then

$$|B_i| = \begin{cases} \mathrm{m} & \text{for}\, i = 1, ..., b-1 \\ \mathrm{n} - (\mathrm{b} - 1)\mathrm{m} & \text{for}\, i = \mathrm{b}. \end{cases}$$

The optimization is done in several steps, depending on how many batches $\{B_i\}$ there are. First the elements of B_1 are optimized, then from B_2 and so on, until each batch has been optimized once, i.e., every element has been optimized. Only then an iteration is over.

Although an iteration becomes much more expensive, since the power conversion efficiency η and the gradient have to be recalculated after each batch, the more individual adjustments save some iterations in the end. A look at Figure 3.7 shows that for the Adam method with 4 batches (AdamBatch), i.e, b = 4, grid arms have already formed after iteration 3. The subtractive random number generator algorithm is a pseudo-random number generator and is used to shuffle all elements before assigning them to batches [49, Chap. 3.5].



Iteration 1 Iteration 2 Iteration 3

Figure 3.7: First iterations of the AdamBatch method.

The fact that this grid pattern has strong distortions plays only a small role, since these can be smoothed with the help of a filter.

To perform an optimization with batches, the number of batches must be defined. To keep the computational effort as low as possible, not too many batches should be chosen, but on the other hand the batches should also have an influence on the optimization. For this reason, four batches (b = 4) are used, each with a quarter of the elements. To calculate the gradient for the batches, all elements that are not in the current batch are set constant, so that their partial derivative is equal to 0 and they are not changed within the batch optimization. Since a large part of the Jacobian matrix consists of zero elements, the matrix operations can be performed much more cost-effectively than with a full batch.

The framework of optimization remains the same as before in Section 3.1.2.2. Without batches, the Gaussian Blur is applied in iteration 18 and 50, with batches in iteration 6 and 16. The filter radius is 0.08 mm both times. The grid pattern obtained in this way can be seen in Figure 3.8. Remarkably, one grid arm goes directly from the external front contact straight up and the other first goes to the side, but then also goes straight up again. In addition, there are the little arms that tap the current from the lower corners. The special thing about this grid pattern is that this structure is similar to the currently used grid pattern, called "Labgrid". The Labgrid has a horizontal grid arm, called bus bar, starting from the contact pad. At the outer ends of this horizontal grid arm, one arm goes vertically upwards. At each of the upper corners of the contact pad, one arm also extends upwards. Thus, the Labgrid has four grid arms. The Labgrid can be seen in Figure 4.1 (left).

Since the objective of this thesis is to produce a better grid than is currently being used, it is promising that optimization will produce similar structures. On the one hand, this means that the result of the optimization is known and therefore could more easily get the approval to test the grid pattern in the lab. And on the other hand, the optimized grid pattern differs from the Labgrid after all, which does not make the optimization effort unnecessary.

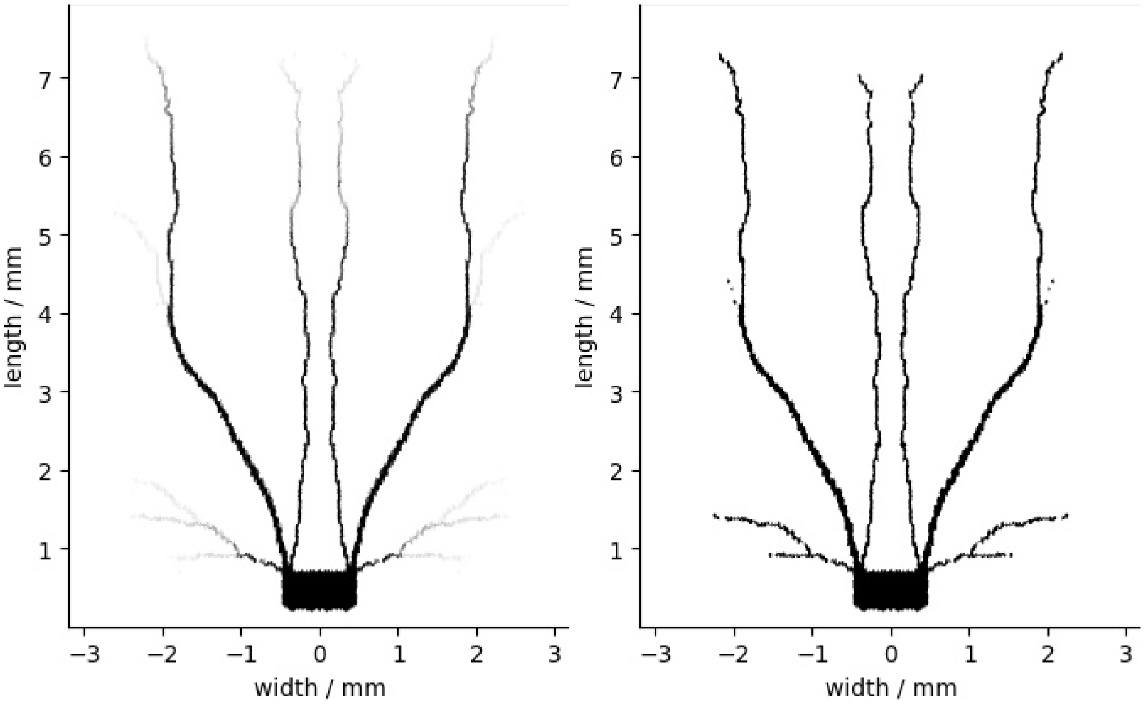


Figure 3.8: Optimized grid pattern before binarization (left) and after binarization (right) with the AdamBatch method.

Again the question arises which optimization method is the better one. Table 3.1 shows some details about the Adam method and the AdamBatch method. This should help in deciding whether the batches are a useful extension.

Optimizer	PCE η	∅ Runtime/It.	No. of It.	Total Runtime
Adam	22.08 %	39.5 sec	193	7623.5 sec
AdamBatch	22.09 %	146.3 sec	49	7168.7 sec
Labgrid	22.40 %			

Table 3.1: Comparison of the Adam method and the AdamBatch method with Gaussian blur.

The AdamBatch method is used for the following optimizations. Not only the slightly better result in shorter time speaks for this, but especially the achieved grid structure. It is true that even with the Adam method, the entire solar cell is well tapped. But with the AdamBatch method, the shortest paths are taken to reach the individual areas of the solar cell. This advantage leads to a further increase of the power conversion efficiency η.

3.3 Solid Isotropic Material with Penalization

Looking at the grid patterns created so far, it is noticeable that the grid pattern without the binarization has many gray areas. Although a threshold is used to remove these gray areas, it does not give the best result. In the best case, the optimization method converges in density vector $\mathbf{x}$ values of 0 and 1. This can not be done through the optimizer. To achieve the goal of a binary density vector $\mathbf{x}$ the calculation must be modified. More precisely the SIMP functions introduced in Section 2.2 must be adapted.

These functions are intended to interpolate the objective function at $\mathbf{x} \in [0,1]^n$. To ensure that at the end of the optimization the density of nearly all elements is in $\{0,1\}$, the intermediate values should be penalized. This can be achieved by the fact that elements with a density between 0 and 1 neither generate much current nor conduct the current well.

The derivation of how to get the functions from Section 2.2 is performed using the SIMP function for conductivity. The density function of the generated current is basically only a vertical reflection. In principle, the interpolating function must meet only two conditions:

$$SIMP_{\text{con}}(0) = 0$$
$$SIMP_{\text{con}}(1) = 1$$

This leads to a monotonically increasing function $SIMP_{\text{con}}$. If an exponential function is to be taken as the basis for interpolation, the following equations result:

$$SIMP_{\text{con}}(0) = a \cdot e^{b \cdot 0 + c} + d = a \cdot e^{c} + d = 0$$
$$SIMP_{\text{con}}(1) = a \cdot e^{b \cdot 1 + c} + d = a \cdot e^{b+c} + d = 1$$

A displacement along the x-axis is not desired, therefore c = 0 and this leads to:

$$SIMP_{\text{con}}(0) = a \cdot 1 + d = 0 \Rightarrow a = -d$$
$$SIMP_{\text{con}}(1) = a \cdot e^{b} + d = 1$$

With $a = -d$ follows:

$$a \cdot e^{b} - a = 1 \Leftrightarrow e^{b} = \frac{1+a}{a} \Leftrightarrow b = \ln\left(\frac{1+a}{a}\right)$$

Hence, for interpolation these general formulas and their derivatives with respect to the components of the density vector $\mathbf{x}_i$ are used:

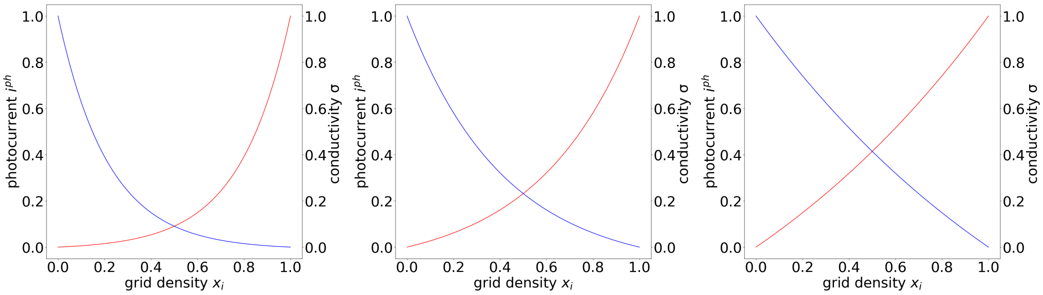
$$SIMP_{\text{con}}(\mathbf{x}_i) = q \cdot e^{\ln(\frac{q+1}{q}) \cdot \mathbf{x}_i} - q \tag{3.2}$$

$$\frac{\mathbf{d}SIMP_{\text{con}}}{\mathbf{dx}_i} = q \cdot \ln(\frac{q+1}{q}) \cdot e^{\ln(\frac{q+1}{q}) \cdot \mathbf{x}_i}$$

$$SIMP_{\text{gen}}(\mathbf{x}_i) = p \cdot e^{\ln(\frac{p+1}{p}) \cdot (1-\mathbf{x}_i)} - p \tag{3.3}$$

$$\frac{\mathbf{d}SIMP_{\text{gen}}}{\mathbf{dx}_i} = -p \cdot \ln(\frac{p+1}{p}) \cdot e^{\ln(\frac{p+1}{p}) \cdot (1-\mathbf{x}_i)}$$

With the exponential function it is possible to interpolate an almost linear function, but also a function with strong curvature through the boundary conditions. Figure 3.9 shows several possibilities of the parameters p and q. The center figure represents the SIMP functions used so far, compare Figure 2.1. In the leftmost image, the penalty for intermediate values is very high. This can be seen from the fact that in the i^{th} element with density $\mathbf{x}_i$= 0.5 not much current is generated but also the conductivity in this element is not very good. In contrast, on the right image, the penalty for intermediate values is low. For the intermediate values one does obtain good results.



$p = q = 0.01$ $p = q = 0.1$ $p = q = 1$

Figure 3.9: Influence of SIMP parameters p and q.

A value of 0.1 for p and q leads to a grid pattern, which has many gray areas, as shown in Figure 3.8 (left). To get a mostly binary density vector $\mathbf{x}$ at the end of the optimization some parameters are tested. The optimization that is performed uses the Adam algorithm with the parameters from Section 2.3.2.2 as the optimizer. In addition to this, the contact pad is added as initial guess according to Section 3.1.1. For a better optimization, four batches are also used and the Gaussian blur is applied in iteration 10 and 20 with a filter radius of 0.08 mm.

In the case of low parameters $p = q = 0.01$, the optimizer tends to avoid intermediate values. Although this is the desired result, since the final values should be 0 and 1, low parameters are not always suitable. At the beginning of the optimization, much more grid is needed than is given by the contact pad. By avoiding intermediate values, the density of the elements quickly goes to 1, where the gradient is positive. However, there the objective function has local minima because the lower and the upper limit of the density vector $\mathbf{x}$ are more efficient than the intermediate values to either conduct current or generate it. This leads to the fact that the grid is no longer removed, resulting in thicker grid arms. The result can be seen in Figure 3.10. The power conversion efficiency η is 21.46 %.

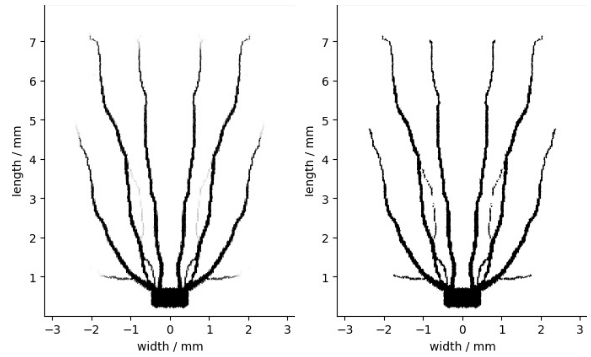


Figure 3.10: Resulting grid pattern with low parameters $p = q = 0.01$ before binarization (left) and after binarization (right).

If the parameters $p = q = 1$, then for an element which has a density of approximately 0.5 already relatively much current is generated and also the element can conduct the current well. This leads to the fact that many thin grid arms are formed and these are mainly gray as shown in Figure 3.11. Many arms are formed because the loss of generated current is not high, even if there is a great amount of metal on the solar cell and the grid arms are thin, because not much grid is needed to conduct the current efficiently.

The problem here is that in the end the values of the density vector $\mathbf{x}$ are rarely at 0 or at 1. Thus the opposite of the desired effect is achieved. The binary final grid pattern is obtained only at great loss of efficiency. Before setting the values to 0 and 1, one has an efficiency of 22.47 % and afterwards only of 20.9 %. However, it must be taken into account that the 22.47 % can never be achieved after binarization, since elements cannot generate much current and at the same time conduct it efficiently. Nevertheless, it is clear that thin grid arms are better than thick ones.

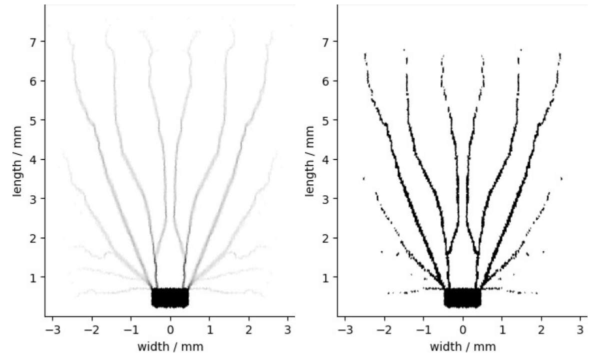


Figure 3.11: Resulting grid pattern with high parameters $p = q = 1$ before binarization (left) and after binarization (right).

In summary, lower values for the SIMP parameters p and q lead to a binary grid pattern that uses too much grid, while higher values cause thin grid arms, which are non-binary. Based on this result, the following approach will be followed:

Within an optimization, a combination of low and high parameter values shall be used to exploit the respective strengths.

Moreover, a variation of the SIMP parameter p has shown that it does not have a very large influence on the efficiency. For this reason, only the parameter q is further developed and $p = 0.1$ is fixed. In particular, two scenarios are interesting:

1. Creating many thin grid arms with a low value of q and then use a high value of q to bring the relevant values of the density vector $\mathbf{x}$ to 0 and 1.

2. Using a high value of q first creates thick grid arms, where grid makes sense. To reduce the amount of grid, then a low value of q is used.

To generate the results, different SIMP parameters are used. For the second approach, the parameter q for the $SIMP_{\text{con}}$ function is set from low to high after a certain number of iterations. This sequence does not work very well. In the end, there are few thin grid arms, but the core problem of obtaining the smallest possible efficiency difference when binarizing is not solved.

The results of the first approach obtained with different values for parameter q for the $SIMP_{\text{con}}$ function are listed in Table 3.2. Until the stated iteration, the parameter q_1 is used. From then on the parameter q_2. The difference Δ of the power conversion efficiency η before binarization and after binarization is given in the last column.

	q_1	q_1	Iteration q is changed	PCE η	$\Delta\eta$ last two iterations
1	1	0.01	20	21.51 %	-0.04 %
2	1	0.01	50	21.74 %	-0.03 %
3	1	0.01	70	21.81 %	-0.01 %
4	0.1	0.01	20	21.61 %	-0.03 %
5	0.1	0.01	50	21.97 %	-0.02 %
6	0.1	0.01	70	22.11 %	-0.01 %

Table 3.2: Comparison of SIMP parameter q changes within the optimization.

In each case, runs 1-3 as well as runs 4-6 have the same parameter values q_1 and q_2. The only thing that changes within these groupings is after how many iterations the parameter q is changed. A look at the column 'power conversion efficiency η' provides that the grid patterns get better the later the parameter q is changed. The reason for this is that the earlier the low value for q is chosen, the more likely one ends up with the result of Figure 3.10. That is, the grid pattern consists of a few thick grid arms. If the parameter q is not changed until iteration 70, the grid pattern will have converged by then for parameter q_1 and there will be minimal changes in power conversion efficiency η. By subsequently using the lower parameter q_2, the structure of the grid pattern is preserved and the only thing that actually changes is the shift of intermediate values to 0 and 1. Since this way the density vector $\mathbf{x}$ converges to a binary vector, a smaller loss of efficiency is also achieved by finally setting the values to 0 and 1 using a threshold of 0.1.

From the experiment with a high and a low SIMP parameter q it is known that a high value leads to several thin arms as shown in Figure 3.11. As the results of the third run in the Table 3.2 show, with an initial high parameter value a less high efficiency is achieved than with a medium parameter value as in the sixth run. The reason for this is the number of grid arms formed for $q = 1$ and $q = 0.1$. Since the resulting grid arms have approximately the same width, the low power conversion efficiency cannot be explained by this.

In summary, with the SIMP parameter $q = 0.1$, the same number of grid arms as the Labgrid has, are obtained. Moreover, if the parameter q is set to a low value like 0.01 after converging for 0.1, the grid pattern is preserved, but the intermediate values can be pushed to the boundaries 0 and 1. The result of an optimization built according to this scheme can be seen in Figure 3.12. Except for very few places, the grid patterns before and after binarization are identical.

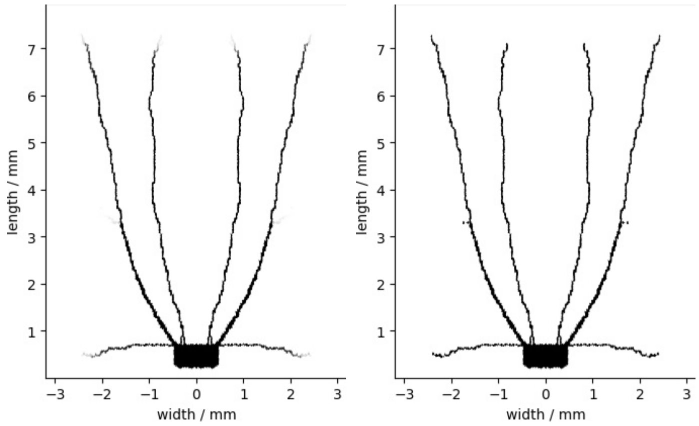


Figure 3.12: Resulting grid pattern with a change of the SIMP parameter q as in the sixth row of Table 3.2 before binarization and after binarization.

3.4 Initial Guess

At the beginning of the optimization, an initial guess of the density vector $\mathbf{x}$ is needed. Basically, this can be any vector whose values lie between 0 and 1. However, it is often the case that the choice of an initial guess can influence the entire optimization. Therefore, it makes sense to take a closer look at the initial guess.

So far, the contact pad, which must be part of the solar cell, has been used as the initial guess, i.e., the components of the density vector $\mathbf{x}$ representing this area have the value 1. All other components have the value 0. With the contact pad, the problem is that no contact have been made without a filter, compare Figure 3.2. This actually already indicates a bad initial guess. An initial guess should lead the optimization in the right direction to a better power conversion efficiency η. For this reason, the currently used grid for solar cells in the laboratory, the so-called Labgrid, will be tested first as initial guess. The idea is to keep the basic structure, and only optimize what is necessary to get an even better grid.

The other approach considered in the following is a strategy for nonlinear localized optimization [50]. This involves optimizing only a fraction of all elements and then using this result as an initial guess for a global optimization. Within this thesis, the direct surroundings of the external front contact is optimized and will be used as initial guess.

3.4.1 Labgrid

The Labgrid can be composed of 4 straight pieces if only half of the cell is considered. The set of positions creating the contour of the Labgrid can be represented as a union of sets $O = O_1 \cup O_2 \cup O_4 \cup O_4$ for $O_i \subset \Omega$. Each subset O_i is the set of positions (x,y) that form a straight grid arm. In Figure 3.13 (top left), the structure of the Labgrid can be seen. If the mesh point of an element is within the set, this element gets grid and thus the corresponding entry in the density vector $\mathbf{x}$ is 1. Because the simplices are not adapted to the shape of the Labgrid and the intervals are partly smaller than the simplices with the used number of points of 20389, the Labgrid is not reproduced exactly. Due to this, the calculated power conversion efficiency η of the Labgrid is only 21.52 % which is significantly less than the 22.40 % of the original Labgrid. The original Labgrid has been created using image processing software and is shown in Figure 4.1 (left).

As mentioned in the previous Section 3.3, if there is already much grid on the solar cell and the density vector $\mathbf{x}$ consists only of 0 and 1, one ends up almost in a local maximum. To give the optimization a new push, the values are set down to 0.08. The value 0.08 is nearly the same value which is also used as the threshold at the end of the optimization, see Figure 2.3. This is the value at which the element tends to have either grid coating or no grid coating. After the density of an element increases or decreases through optimization, the preferences of an element rarely change. Therefore, the value 0.08 is suitable to heuristically obtain any grid pattern with the optimization, but the value is still high enough that the basic pattern of the Labgrid can be recovered. This can be seen in the top right grid pattern in Figure 3.13 by the fact that many grid arms are formed after eight iterations and nearly the original grid arms are restored at the end of the optimization in the bottom grid patterns. The framework of optimization used to generate the results is the same as the one at the end of Section 3.3.

With a filter radius of 0.02 mm of the Gaussian Blur the bottom left grid pattern in Figure 3.13 is created. Here only a rounding of the corners occurs and the grid arms next to the contact pad become a bit thicker. The resultig power conversion efficiency η is 22.23 %. If, on the other hand, a filter radius of 0.08 mm is chosen, a shift of the outer grid arm occurs as can be seen in the bottom right grid pattern. The outer grid arm connects with the inner grid arm, while the lower crossbar remains. This shortens the path for the outer arm and the current passes through fewer elements. This results in a power conversion efficiency of 22.21 %. Another change are the tips of the grid arms. For both filter radii, squiggles occur. This indicates that it is not clear how long these arms should basically be.

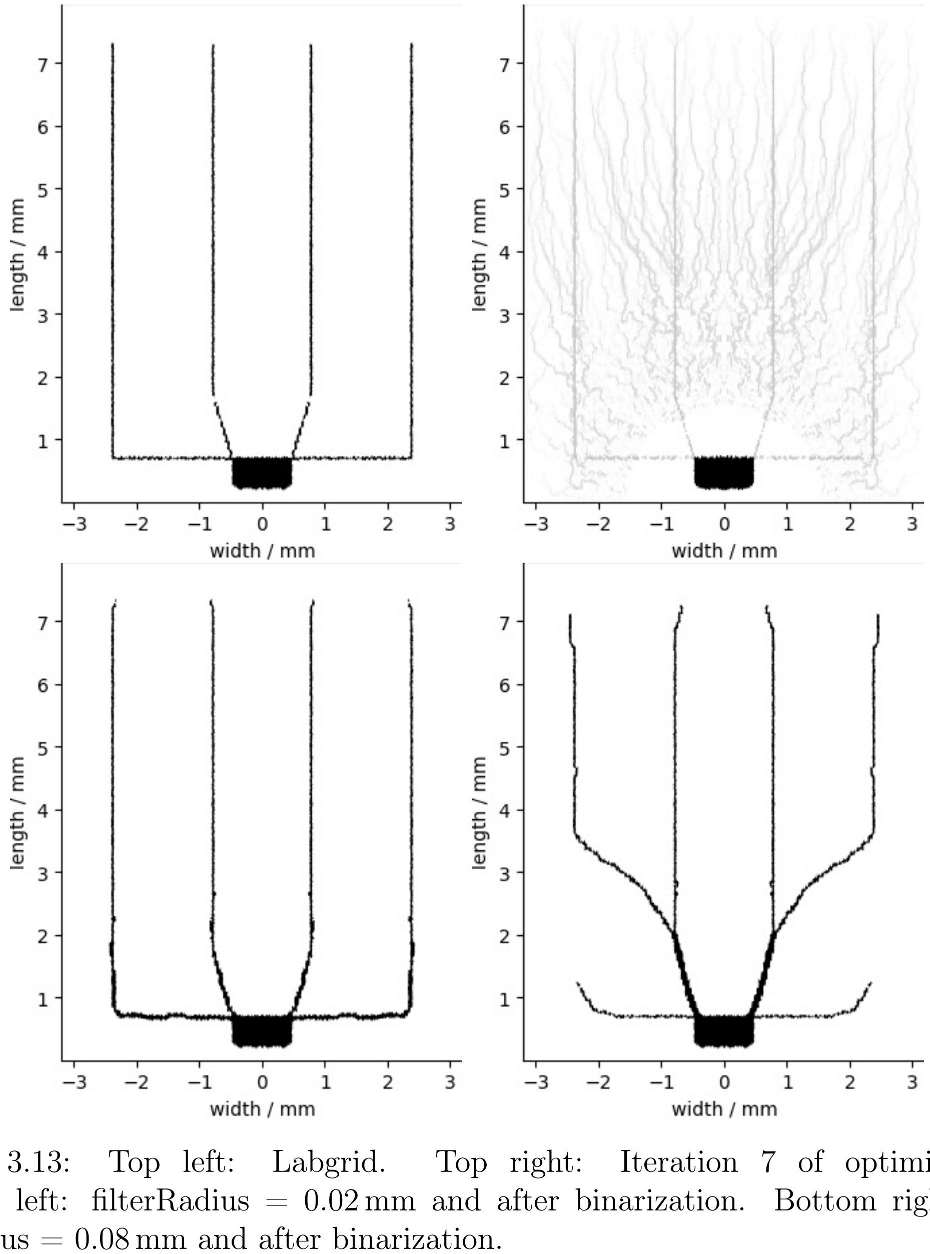


Figure 3.13: Top left: Labgrid. Top right: Iteration 7 of optimization. Bottom left: filterRadius = 0.02 mm and after binarization. Bottom right: filterRadius = 0.08 mm and after binarization.

3.4.2 Nonlinear Localization Strategy

Unlike with the Labgrid as initial guess, the following method involves more effort. A local optimization is to be carried out, which is more costly and which already exhibits the basic structure of the global optimization. This should lead to the fact that the actual global optimization can be accomplished in few iterations and thus even the time up to the convergence can be reduced.

As before, only half of the solar cell is optimized and then mirrored. However, before the optimization is performed, the grid is optimized only on a quarter of the half solar cell. This would be the rectangle defined by $[0,1.6]\times[0,4.475]$. The domain is chosen for two reasons. Firstly, the area around the contact pad is indicative for the whole optimization and secondly, the dimension of the problem can be reduced significantly. The reason is that for each selected element the neighboring elements are needed additionally. However, if one chooses a compact domain, one can disregard all elements outside this domain. The result of this optimization should serve as an initial guess for the global optimization. An example of this is the middle grid pattern in Figure 3.14. In order not to lose the global effect in the local optimization, the power conversion efficiency η of the half solar cell is calculated in each iteration, so that the gradient can be calculated for the half solar cell. As before for the contact pad or for the batches, the elements not considered in the local optimization are set constant. Thus, their derivative is zero. In this way one achieves that the many zero entries and their storage in the CRS format lead to an improvement of the runtime. This is important, otherwise it would not be worthwhile to perform two optimizations.

Both local and global optimization are carried out with the latest framework of optimization from Section 3.3. The local optimization is performed until convergence with subsequent binarization. Therefore, as for the Labgrid, a new impulse is required that the global optimization does not start in a local maximum. This is achieved by setting the values of the density vector $\mathbf{x}$ to a lower value. In the first iteration of the global optimization, the density of each element is set to 0.08 if it was previously 1.

If one looks in Figure 3.14 at the initial guess in the middle and the final result on the right, it is noticeable that the approaches for the arms are further developed, but remain in their basic form. This at least achieves that the required iterations until a basic shape is formed in the global optimization are significantly reduced. Nevertheless, it has not yet been possible to reduce the runtime using the nonlinear localization strategy.

Unfortunately, too much grid is applied in the local optimization due to the lack of grid on the entire cell. This results in a tighter grid pattern design and thus in a worse power conversion efficiency η of 22.04 %.

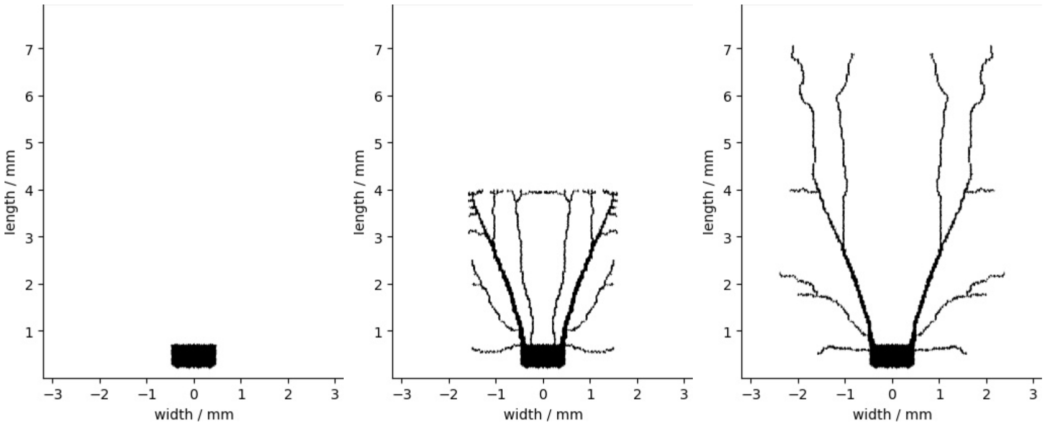


Figure 3.14: Initial guess for local (left) and global (center) optimization and achieved grid pattern.

Comparison Initial Guesses

The initial guess with the nonlinear localization strategy is significantly worse than the initial guess with the Labgrid according to previous results. To make the nonlinear localization strategy work better, one would probably have to readjust the timing of the Gaussian blur and also change the SIMP paramter q for conductivity differently within the local optimization than within the global optimization. Furthermore, the local area which will be optimized first can be chosen independently from the geometry. For example, in a first iteration a loss analysis can be performed to find out in which elements of the solar cell the most efficiency tends to be achieved. The local optimization could then be performed on these elements.

In contrast, the Labgrid is simple to implement and provides a very good power conversion efficiency η immediately. Since there is almost no difference in terms of efficiency between the two options with different filter radii, the larger filter radius with 0.08 mm is adopted as the default settings in addition to the Labgrid as initial guess. The reason for this is that there is a greater change of the initial grid pattern with the larger filter radius and these changes can have an additional effect as the number of mesh points increases. Seen from another perspective, it is already foreseeable how the Labgrid will change with an increasing number of mesh points if the filter radius remains small and is therefore not be investigated further.

3.5 Implementation of Topology Optimization

After all methods used for the optimizations have been presented in Chapter 3, the implementation from Section 2.3.4 will be extended with them. The contact pad, which is not changed during optimization, has been added. This is initialized in lines 9-16 in Listing 3.1. Moreover, the Labgrid is used as the initial guess, which leads to a very high power conversion efficiency η (lines 22-27), see Section 3.4. In lines 36 and 40-43, the SIMP approach presented in Section 3.3 is implemented. The Gaussian blur which is used to get out of a local maximum and thus leads to straighter grid arms is implemented in line 46. Finally in line 48-57, the approach presented in Section 3.2 is used, where the Adam method is no longer applied to the whole density vector **x**, but only to subsets.

Listing 3.1: Optimization Loop

```
 1 //initialize mesh points
 2 var numberPoints = 20000;
 3 var points = EquidistantDistribution(numberPoints);
 4 
 5 //create discretization
 6 var mesh = VoronoiDiagram(points);
 7 
 8 //set fixed elements (contact pad C)
 9 var densityVector = new Vector.Create<double>(numberPoints);
10 var fixedElements = new Vector.Create<bool>(numberPoints);
11 for (var meshPoint.position in mesh)
12     if (meshPoint in C)
13     {
14         densityVector[meshPoint.index] = 1;
15         fixedElements[meshPoint.index] = true;
16     }
17 
18 //set initial guess (Labgrid O)
19 for (var meshPoint.position in mesh)
20     if (meshPoint in O)
21         densityVector[meshPoint.index] = 1;
22 
23 //set the values of the density vector to 0.08
24 SetNonZeroValuesToLowerValue(densityVector);
25 
26 //initialize previousGradient
27 var gradient = new Vector<double>(numberPoints)
28 
29 //set max iteration
30 var maxIteration = 100;
31 
32 //set SIMP parameter p
33 SIMP_GeneratedCurrent = 0.1;
34 
35 //Optimization loop, ends after max iteration
36 for (int iteration = 0; iteration < maxIteration; iteration++)
37     if (iteration < 70)
38         SIMP_Conductivity = 0.1;
39     else
40         SIMP_Conductivity =  0.01;
41 
42     if (iteration == 10 || iteration == 20)
43         densityVector = GaussianBlur(densityVector, filterRadius
               = 8e-5)
44 
45     var Batches = CreateBatches(densityVector, b = 4);
46     fixedElements_Contactpad = fixedElements.Clone();
47     for (var batch in Batches)
48         fixedElements[new Range.All] = true;
```

```
        fixedElements[batch] = false;
        var efficiency = FunctionValue(densityVector);
        var gradient = CalculateGradient(densityVector ,
           fixedElements);
        (densityVector , previousGradient) =
                Optimizer_Adam(densityVector , gradient ,
                   previousGradient , iteration)
        fixedElements = fixedElements_Contactpad.Clone();
```

Implementation of Optimization in C# with source code of simulation platform twinPV [3].

4 Proposal of a New Optimized Grid Pattern

The main objective of the thesis is not only to build an optimization, but also to create a new grid pattern proposal with the help of this optimization.

For this purpose, this chapter first presents the Labgrid, which is the grid pattern to be surpassed. In contrast to the Labgrid as initial guess, compare Section 3.4.1, where the Labgrid is represented as best as possible on a given mesh, this time the contour of the Labgrid is created exactly by pairs of mesh points as in Section 2.2 the boundary of the solar cell. This results in an accurate representation of the Labgrid.

Afterwards, the methods which are to be finally used for the optimization are summarized and applied in the correct order. These methods are then used to ultimately optimize a grid pattern to be more effective than the Labgrid. Therefore only the number of mesh points is varied.

4.1 Realistic Labgrid

The currently used Labgrid was created with the help of a spatially resolved simulation and a finite element method [51, 52, 53, 54]. Additionally, various assumptions were made. For example, no matter where current is generated on the solar cell, the path to a good conductor should never be more than 1 mm away. The decision to set the length of the path to a maximum of 1 mm is based on empirical values, according to which the efficiency becomes worse if the electrons have to flow more than 1 mm in the poorly conducting TCO. By the geometry of a solar cell, the number and the length of the individual grid arms were thus determined. Hence, the distance between the outer grid arms and the edge of the solar cell is half the distance between the individual grid arms. Furthermore, this is also the distance to the lower and upper edge. This way, each grid arm has its own area where the generated current flows onto the arm. After construction, all areas are the same size. In addition, the grid arms were designed to be very thin so that as much sunlight as possible reaches the absorber.

Although many assumptions were made, the result of this grid pattern speaks for itself. The power conversion efficiency η calculated under the same conditions as for optimization is 22.40 %. In comparison, the maximum power conversion efficiency η achieved so far with optimization was only 22.21 %. The efficiency of the Labgrid must be surpassed, and in doing so, the assumptions in particular are to be optimized.

4.2 Framework of Optimization

In the context of this thesis, several methods have been developed and tested, which are now summarized to ensure that the best possible optimization can be performed. The configuration of the solar cell is given in Table 2.2.

The first step of the optimization is to determine the number of mesh points to use in order to discretize the solar cell with an area of 6.4 mm × 7.95 mm, see Section 2.3. A higher number is reflected in a better representation of a solar cell. Due to lack of computer capacity it is not possible to use more and more points. For this reason, the number of points is to be increased bit by bit in order to be able to draw a conclusion as to how much potential would theoretically still be there by increasing the number of mesh points.

After decomposing the solar cell into different simplices based on the mesh points, compare Section 2.2, the elements on the solar cell that are invariable during the optimization are set. In the case of this thesis, these are only the elements that form the contact pad and their corresponding density is 1 as described in Section 3.1.1. The optimizer that can best maximize the power conversion efficiency η before adding the contact pad is the Adam method, see Section 2.3.3, with the same parameters as in Section 2.3.2.2 and is therefore used.

Next is the initial guess: The Labgrid, which was developed from experience, is suitable as a good initial guess. Due to the fact that many assumptions were made for it, which were certainly not the best, there is still much to optimize here. For this reason and the already high efficiency, the Labgrid is used as an initial guess. In order not to start the optimization in a local maximum, the densities, where grid is present, are set to 0.08 as described in Section 3.4.1.

The framework conditions that determine the procedure of the optimization have to be defined. To obtain a good grid pattern, there is no convergence criterion, but a maximum number of iterations is set. The exact number of iterations will be specified later. Basically, the optimization can be divided into four sections. Gaussian blurs are applied at the end of the first and second sections, and the SIMP parameter q for conductivity between the third and fourth sections. By choosing the maximum number of iterations for each section large enough, one can be sure that changes are not made until the previous optimization section has converged.

In other words, when the grid pattern does not change in retrospect, the Gaussian blur with the filter radius 0.08 mm is applied. Experience shows that this happens after 10 iterations. But since there are still some impurities such as squiggles or close grid arms left, the Gaussian blur is used for a second time. This time, however, no attention is paid to convergence. The reason for this is that if the Gaussian blur is applied too late, some 'bad' grid arms will have developed, while others will have regressed. Also, the filter works better when as many elements as possible still have grid. The Gaussian blur is used the second time in iteration 20. In Section 3.1.2.2 the Gaussian blur has been introduced.

As the series of experiments in the 3.3 section showed, the best result is obtained by setting the SIMP parameter p for the generated current and, at the beginning, also the SIMP parameter q for the conductivity to 0.1. This gives some thin grid arms whose density is an intermediate value between 0 and 1. This is the basis of the later final grid pattern, which is why there are no modifications until there are no more significant changes between the iterations. This situation occurs after 70 iterations. Therefore, only then the SIMP parameter q for the conductivity is set to 0.01. This ensures that the thin grid arms are preserved, but their density becomes nearly binary in later iterations. The optimization process is allowed to converge again before the density values are binarized using a threshold of 0.1. For this purpose 30 iterations are provided. This results in a maximum number of 100 iterations. The termination error can be minimized by increasing the maximum number of iterations.

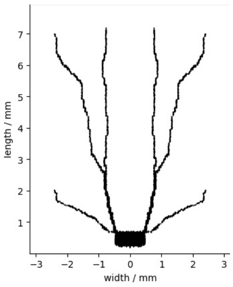
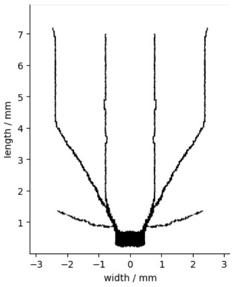
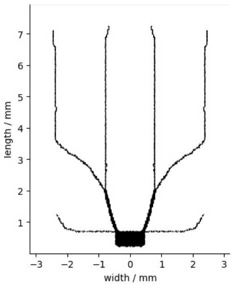
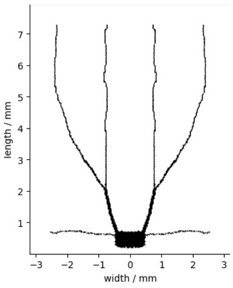
The Adam method in combination with batches has turned out to be a more efficient optimizer than the Adam method in terms of maximizing the power conversion efficiency η and runtime as shown in Table 3.1. Consequently, this optimization method is used to find the optimal grid pattern.

4.3 Final Grid Pattern

Based on the framework of optimization presented in the last section, a design for a new grid pattern will be created. Different numbers of mesh points will be considered to continue possible trends with an increasing number of mesh points. This may allow a prediction of how the grid pattern would evolve if it would be possible to choose an arbitrarily high number of mesh points.

In contrast to the previous optimization results, the grid pattern, which will be the final result of this thesis, will be modified after optimization manually with a image processing software. This means, for example, that the grid arms are straightened where it seems that the distance on the solar cell should be covered as fast as possible or the grid arms are made thinner where the arms are alternately thinner and thicker as shown in Figure 3.13. After editing, the important points of the edited grid pattern are reproduced by added mesh points similar to the boundary of the design domain in Section 2.2 or the Labgrid in Section 3.4.1. Afterwards, a mesh with the additionally mesh points is created by Voronoi diagram and is used to recalculate the power conversion efficiency η.

The final grid pattern is constructed using the designs of the optimized grid patterns for 5000, 10000, 15000, 20000, 25000, 30000 and 40000 mesh points. The results for each number of mesh points can be seen in Table 4.1. As the number of mesh points increases, so do the system requirements. With the system presented in Table 2.2, optimization with more than 40000 points is not possible.

Number of points	Power Conversion Efficiency η	Grid Pattern
10000	21.91 %	
15000	22.12 %	
20000	22.21 %	
25000	22.21 %	

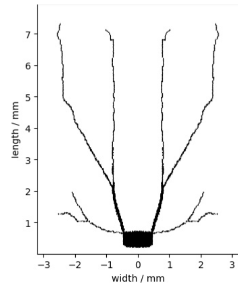
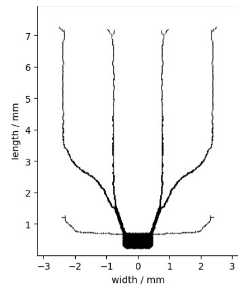
Number of points	**Power Conversion Efficiency η**	**Grid Pattern**
30000	22.23 %	
40000	22.29 %	

Table 4.1: Optimized grid patterns with different numbers of mesh points.

The basis for the new grid pattern is the result of the optimization with 40000 mesh points. Starting from the contact pad, the new grid is constructed. A noticeable feature of the optimization results is that a horizontal grid arm is always located at almost the same height as the upper edge of the contact pad. This grid arm is strongly developed in the optimization with 10000, 15000 and 30000 points with the outer ends upwards. For this reason, such a grid arm is reproduced with the image processing software, where the outer ends are higher than the contact pad.

In each optimization, it is noticeable that a thicker grid arm is formed, starting on the inside of the top corner of the contact area. This varies in length until it splits into two more grid arms. The distance from the contact pad to this bifurcation is taken from the grid pattern that emerged at 40000 mesh points. One of these created arms merges directly into the middle grid arm, which is already predefined by the Labgrid. The other of the created grid arms continues as an extension of the thicker grid arm starting at the contact pad to the outer grid arm, which is also already given by the Labgrid. Within the construction the thicker grid arm has a width of 0.045 mm.

Another common feature of the optimized grid patterns is that at the top of the vertical grid arms, tangles are created. This is most noticeable in the result of the optimization with 40000 mesh points. At the end of these grid arms is another bifurcation. One of the grid arms goes to the upper left while the other goes to the upper right. The same behavior can be seen with the lower lateral grid arms of the result of the optimization with 30000 points. A bifurcation may also occur there.

This feature is also included in the construction of a new grid pattern.

The result of this construction can be seen in Figure 4.1 (right). Under the same conditions as the original Labgrid, this grid pattern achieves a power conversion efficiency η of 22.40 %. The reason why the customized grid pattern design is 0.11 percentage points better than the optimized version with 40000 mesh points is mainly because the grid arms are uniform in width and the straight grid arms save metal, minimizing the area of shading on the solar cell. The grid arms have a width of 0.030 mm.

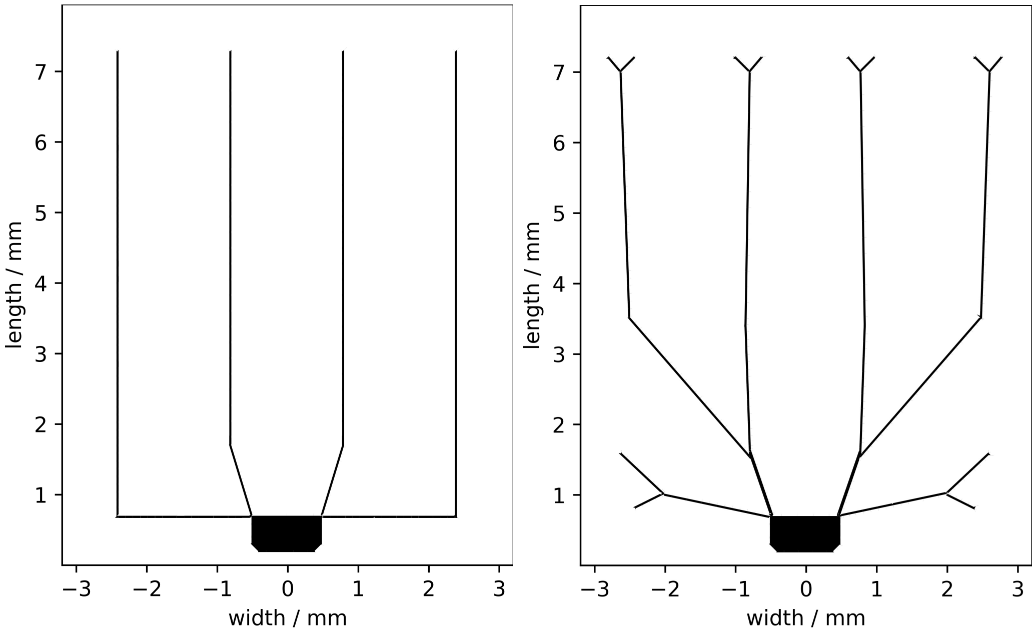


Figure 4.1: Labgrid on the left and the proposal of a new grid pattern on the right.

5 Summary and Outlook

In Chapter 2, the calculation function has been set up and different optimizers have been tested to see how they would handle this task. The first step, to calculate the power conversion efficiency η, has been to divide the geometry of the solar cell, i.e., the design domain Ω, into finitely many simplices. To form the simplices around the mesh points, a Voronoi diagram has been used. In order to perform a FEM, the stiffness matrix has been assembled. For this, Maxwell's equations have been used to obtain an equation to describe the charge flows for each simplex depending on its neighbors. From this, the nonlinear system of equations has been formed. The system of equations has been reformulated into a root-finding problem, which has been solved with the Newton-Raphson method. With the right-hand side of the system of equations $\mathbf{i}^{\text{gen}}$ dependent on the solution vector $\mathbf{u}$, the resulting power p_{cell} of the solar cell has been calculated. The ratio of the incident power of the sunlight to the power from the solar cell is the power conversion efficiency η of a solar cell. The accuracy of this calculation was investigated and verified by Mario Zinßer [4]. Thus, the model can be taken for granted.

Based on this calculation, the grid pattern should be optimized so that the power conversion efficiency η is maximized. In order to do this, an optimizer has been needed. Since both the gradient-free and the second order optimizers require much effort, only the gradient has been included in the optimization of the grid pattern. The gradient of the power conversion efficiency η with respect to the density vector $\mathbf{x}$ has been derived in Section 2.3. For this, however, a simplification has been made to avoid having to solve a recursive problem. Nevertheless, the gradient could be used for optimization.

In the next Section 2.3.2, several gradient-based optimizers have been investigated and tested to see how well they apply to the grid optimization problem. Unfortunately, the Quasi-Newton optimizer BFGS has not been successful. This does not have to be a general statement, since only a library had been tried and thus it had been not possible to adapt the optimization method individually to the problem.

Several variations and modifications of the gradient descent method have been tried. These have converged in almost the same grid pattern with the exception of the standard method with enormous step size. Since the Adam method has turned out to be the most efficient optimizer, the optimization has been designed around this algorithm.

In the beginning of Chapter 3, constraints have been added to the grid pattern and thus to the density vector $\mathbf{x}$. This has ensured that the resulting optimized grid pattern can actually be used in reality. This has included the addition of the contact pad on the one hand, and on the other hand the removal of impractical deficiencies of the grid pattern. This has been implemented in Section 3.1, completing the model of a solar cell. After adding a contact pad, the result of optimization has been a grid pattern that has led to a power conversion efficiency η of 19.09 %. A gap has been created around the contact pad during optimization, which is why the grid arms have not been connected to the external front contact. In order to close this gap and remove impurities such as squiggles, several filtering methods have been presented. With the Gaussian blur a power conversion efficiency η of 22.08 % has been achieved.

As a result of the fact that there has been a significant change in optimization behavior due to the addition of the contact pad, a method for piecewise optimization of the density vector $\mathbf{x}$ within an iteration step has been introduced in Section 3.2. Instead of optimizing all elements in one step, they have been randomly divided into batches. This has led to a better optimization behavior, because within one iteration the density changes of the neighboring elements have been already considered. This has reduced the runtime until convergence and also has increased the power conversion efficiency η to 22.09 %.

Already in Chapter 2.2 an intermediate of the two options 'an element has a density of 0 and no grid' and 'an element has a density of 1 and therefore a grid' has been allowed to guarantee a continuous optimization, which has been further explained in Chapter 3.3. This has been realized using the Solid Isotropic Material with Penalization approach. Depending on how high the penalty has been chosen for a density that is not allowed in practice, a different optimization behavior can be enforced. In conclusion, the best result has been achieved by initially tolerating an intermediate of no grid and grid and penalizing less, and towards the end trying to achieve a binarization of the density vector $\mathbf{x}$ with a high penalization of the intermediate values. After implementing this method, a power conversion efficiency η of 22.11 % has been achieved.

In the previous sections, only the adjustment of the density vector $\mathbf{x}$ has been discussed and optimized. The question of where to start with optimization has not yet been addressed. In principle, three options have been presented. The first possibility has been to optimize the grid pattern starting from the contact pad, which must be present in any case. The second possibility has been obvious: the optimization will take place starting from the currently used Labgrid. The last possibility has been the most elaborate method and, in contrast to the second method, allows a grid pattern without specifications. Based on a local optimization with the contact pad as initial guess, a global optimization has been performed afterwards. This method is called nonlinear localization strategy. With the Labgrid as the initial guess, a power conversion efficiency η of 22.21 % has been achieved.

All these presented methods have been combined in the Chapter 4 to create a grid pattern which exceeds the achieved power conversion efficiency η of the Labgrid. As a basis for constructing such a grid pattern, optimization has been performed with different numbers of mesh points. Based on the resulting grid patterns with increasing mesh points, the grid pattern has been modified using image processing software and then has been re-imported as a density vector $\mathbf{x}$ to calculate the power conversion efficiency η. This has once again significantly increased the power conversion efficiency η to 22.40 %. This is a similar high efficiency which has been achieved with the Labgrid.

Considering that any improvement of power conversion efficency η is a success, it can be said that the methods presented in Chapter 3 have worked and led to a better result. Finally, with all methods together grid patterns have developed that are competitive with the grid patterns currently in use.

In order to achieve even better grid patterns in the future, the following approaches can be taken:

Basically, the batches introduced in Section 3.2 are a kind of nonlinear localization strategy. Thus, only the two extremes have been used so far. First, optimization has been performed on a few randomly distributed elements in each iteration, and later optimization has been performed on a local domain of the solar cell. In the first case, for example, the batches can be chosen to first optimize the elements that actually have a large impact on the power conversion efficiency η. The division could be designed based on the gradient. In the second case, instead of optimizing to convergence, only the batches that are relevant can be optimized several iterations at a time. With this approach, optimization can be focused on the maximum change in power conversion efficiency η.

Often, with a higher resolution of the model, a better result is achieved. In the case of this thesis, this would mainly correspond to increasing the number of mesh points. As the number of points increases, the power conversion efficiency η rises, as can be seen in Table 4.1. This in turn could produce an even more detailed grid pattern with better efficiency. However, there will be an upper limit, since all the current can never be dissipated.

Another possibility to get a better grid pattern is to construct a grid pattern based on the results of the optimization without any special initial guess. This could be, for example, the optimizations that start only with the contact pad. The problem here has been that the grid arms cannot be straight to the same extent as when using the Labgrid as initial guess. However, since this can be modified manually with an image processing software, as described in Chapter 4, it is possible to improve the result of these optimizations significantly.

Although it can be assumed that the constructed grid patterns will achieve similar efficiency in the laboratory, it is still necessary to test these grid patterns in the laboratory. This will be done in the laboratory of the ZSW in the future.

Bibliography

[1] IPCC. Der Weltklimarat - Zwischenstaatlicher Ausschuss für Klimaänderungen. `https://www.de-ipcc.de/119.php`. [08.10.2021].

[2] Felix Creutzig, Peter Agoston, Jan Christoph Goldschmidt, Gunnar Luderer, Gregory Nemet, and Robert C. Pietzcker. The underestimated potential of solar energy to mitigate climate change. *Nature Energy*, 2(9), 2017.

[3] Mario Zinßer, Tim Helder, Andreas Bauer, and Benedikt Braun. Source code of simulation platform twinPV. `https://github.com/ZSW-MAT`, 2021. [08.10.2021].

[4] Mario Zinßer, Moritz Loy, Tim Helder, Andreas Bauer, Theresa Magorian Friedlmeier, and Michael Powalla. Finite element simulation of electrical intra-device physics of thin-film solar cells and its implications on the efficiency. *manuscript submitted to Journal of Photovoltaics*, 2021.

[5] Michael Powalla, Stefan Paetel, Erik Ahlswede, Roland Wuerz, Cordula D. Wessendorf, and Theresa Magorian Friedlmeier. Thin-film solar cells exceeding 22% solar cell efficiency: An overview on CdTe-, cu(in,ga)se2-, and perovskite-based materials. *Applied Physics Reviews*, 5(4):041602, 2018.

[6] Shogo Ishizuka, Jiro Nishinaga, Masayuki Iioka, Hirofumi Higuchi, Yukiko Kamikawa, Takashi Koida, Hajime Shibata, and Paul Fons. Si-doped cu(in,ga)se2 photovoltaic devices with energy conversion efficiencies exceeding 16.5% without a buffer layer. *Advanced Energy Materials*, 8(11):1702391, 2018.

[7] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. `https://www.pveducation.org/pvcdrom/design-of-silicon-cells/sheet-resistivity`, 2016. [17.10.2021].

[8] Mario Zinßer, Tim Helder, Andreas Bauer, Theresa Magorian Friedlmeier, Julia Zillner, Jan-Philipp Becker, and Michael Powalla. Optical and electrical loss analysis of thin-film solar cells combining the methods of transfer-matrix and finite elements. *submitted to Solar Energy Materials & Solar Cells*, 2021.

[9] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. `https://www.pveducation.org/pvcdrom/solar-cell-operation/solar-cell-efficiency`, 2016. [08.10.2021].

[10] Steven J. Byrnes. Multilayer optical calculations. *arXiv: Computational Physics*, 2016.

[11] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. `https://www.pveducation.org/pvcdrom/pn-junctions/band-gap`, 2016. [08.10.2021].

[12] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. `https://www.pveducation.org/pvcdrom/tandem-cells`, 2016. [08.10.2021].

[13] Martin P. Bendsøe and Ole Sigmund. *Topology Optimization.* Springer Berlin Heidelberg, 2004.

[14] G. L. Dirichlet. J. Reine Angew. Math. 40:209–227, 1850.

[15] G. Voronoï. J. Reine Angew. Math. 124:198 – 287, 1908.

[16] B. Barroqueiro, A. Andrade-Campos, and R. A. F. Valente. Designing self supported SLM structures via topology optimization. *Journal of Manufacturing and Materials Processing*, 3(3):68, 2019.

[17] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, 2001.

[18] Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[19] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989.

[20] M. Zhou and G.I.N. Rozvany. The COC algorithm, part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3):309–336, 1991.

[21] H. P. Mlejnek. Some aspects of the genesis of structures. *Structural Optimization*, 5(1-2):64–69, 1992.

[22] Martin Denk, Klemens Rother, Mario Zinßer, Christoph Petroll, and Kristin Paetzold. NODAL COSINE SINE MATERIAL INTERPOLATION IN MULTI OBJECTIVE TOPOLOGY OPTIMIZATION WITH THE GLOBAL CRITERIA METHOD FOR LINEAR ELASTO STATIC, HEAT TRANSFER, POTENTIAL FLOW AND BINARY CROSS ENTROPY SHARPENING. 1:2247–2256, jul 2021.

[23] D. K. Gupta, M. Langelaar, M. Barink, and F. van Keulen. Topology optimization of front metallization patterns for solar cells. *Structural and Multidisciplinary Optimization*, 51(4):941–955, 2014.

[24] James Clerk Maxwell. VIII. a dynamical theory of the electromagnetic field. 155:459–512, 1865.

[25] Karl Hollaus. Einführung in partielle Differentialgleichungen und Methode der Fniten Elemente. `https://www.asc.tuwien.ac.at/~khollaus/Teaching/PDE_FEM_16.pdf`, 2016. [13.10.2021].

[26] Peter Arbenz. The poisson equation. `http://people.inf.ethz.ch/arbenz/FEM17/pdfs/0-19-852868-X.pdf`, 2017. [29.11.2021].

[27] Eric Heinemeyer. Direkte und inverse Randwertprobleme aus der Modellierung elektrostatischer Materialprufungsverfahren. `https://num.math.uni-goettingen.de/picap/pdf/E564.pdf`, 2003. [27.11.2021].

[28] Elena Roibis-Millan, Juan L. Cubero-Estalrrich, Alejandro Gonzalez-Estrada, Rocio Jado-Puente, Marlon Sanabria-Pinzon, Daniel Alfonso-Corcuera, Jose Miguel Alvarez, Javier Cubas, and Santiago Pindado. Lambert w-function simplified expressions for photovoltaic current-voltage modelling. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. IEEE, 2020.

[29] Extrem Optimization. Package extreme.numerics. `https://www.nuget.org/packages/Extreme.Numerics/`, 2021. [15.10.2021].

[30] John A. Nelder and Roger Mead. A simplex method for function minimization. *Comput. J.*, 7:308–313, 1965.

[31] Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. *Machine Learning Refined.* Cambridge University Press, 2020.

[32] J.H. Lambert. Observationes variae in mathesin puram. *Acta Helvetica Physico-Mathematico-Anatomico-Botanico-Medica*, 3:128–168, 1758.

[33] Anders Klarbring and Bo Torstenfelt. Dynamical systems and topology optimization. *Structural and Multidisciplinary Optimization*, 42(2):179–192, 2010.

[34] Sebastian Ruder. An overview of gradient descent optimization algorithms. `https://ruder.io/optimizing-gradient-descent/index.html#gradientdescentvariants`. [08.10.2021].

[35] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014.

[36] C. G. BROYDEN. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[37] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.

[38] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

[39] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24(111):647–647, 1970.

[40] Claas Abert, Christian Huber, Florian Bruckner, Christoph Vogler, Gregor Wautischer, and Dieter Suess. A fast finite-difference algorithm for topology optimization of permanent magnets. *Journal of Applied Physics*, 122(11):113904, 2017.

[41] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[42] Geoffrey Hinton Tijmen Tieleman. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4:26–31, 2012.

[43] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization.* Mit Press, 2019.

[44] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.

[45] W. Wettling and E. Schaffer. Silicium-solarzellen mit höchsten wirkungsgraden. 1996.

[46] Tyler E. Bruns and Daniel A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26-27):3443–3459, 2001.

[47] T. E. Bruns and D. A. Tortorelli. An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 57(10):1413–1430, 2003.

[48] Eric W. Weisstein. Gaussian function. `https://mathworld.wolfram.com/GaussianFunction.html`. [08.10.2021].

[49] Donald Knuth. *The art of computer programming*, volume 2. Addison-Wesley Pub. Co, Reading, Mass, 3 edition, 1973.

[50] Philippe Cresta, Olivier Allix, Christian Rey, and Stéphane Guinard. Nonlinear localization strategies for domain decomposition methods: Application to post-buckling analyses. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1436–1446, 2007.

[51] Joop van Deelen, Lennaert Klerk, and Marco Barink. Optimized grid design for thin film solar panels. *Solar Energy*, 107:135–144, 2014.

[52] G.J.M. Janssen, L.H. Slooff, and E.E. Bende. 2d - finite element model of a cigs module. In *2012 38th IEEE Photovoltaic Specialists Conference*, pages 001481–001485, 2012.

[53] Stefan Eidelloth, Felix Haase, and Rolf Brendel. Simulation tool for equivalent circuit modeling of photovoltaic devices. *IEEE Journal of Photovoltaics*, 2(4):572–579, 2012.

[54] Yulia Galagan, Erica W.C. Coenen, Sami Sabik, Harrie H. Gorter, Marco Barink, Sjoerd C. Veenstra, Jan M. Kroon, Ronn Andriessen, and Paul W.M. Blom. Evaluation of ink-jet printed current collecting grids and busbars for ITO-free organic solar cells. *Solar Energy Materials and Solar Cells*, 104:32–38, 2012.