

Electrostatic Lens System Design and Optimization Using Evolutionary Algorithms

Hesam Mahmoudi Nezhad, N.

DOI

[10.4233/uuid:b2e0711a-e0c2-41b3-ae0a-c8e407efb2ac](https://doi.org/10.4233/uuid:b2e0711a-e0c2-41b3-ae0a-c8e407efb2ac)

Publication date

2024

Document Version

Final published version

Citation (APA)

Hesam Mahmoudi Nezhad, N. (2024). *Electrostatic Lens System Design and Optimization Using Evolutionary Algorithms*. [Dissertation (TU Delft), Delft University of Technology].
<https://doi.org/10.4233/uuid:b2e0711a-e0c2-41b3-ae0a-c8e407efb2ac>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

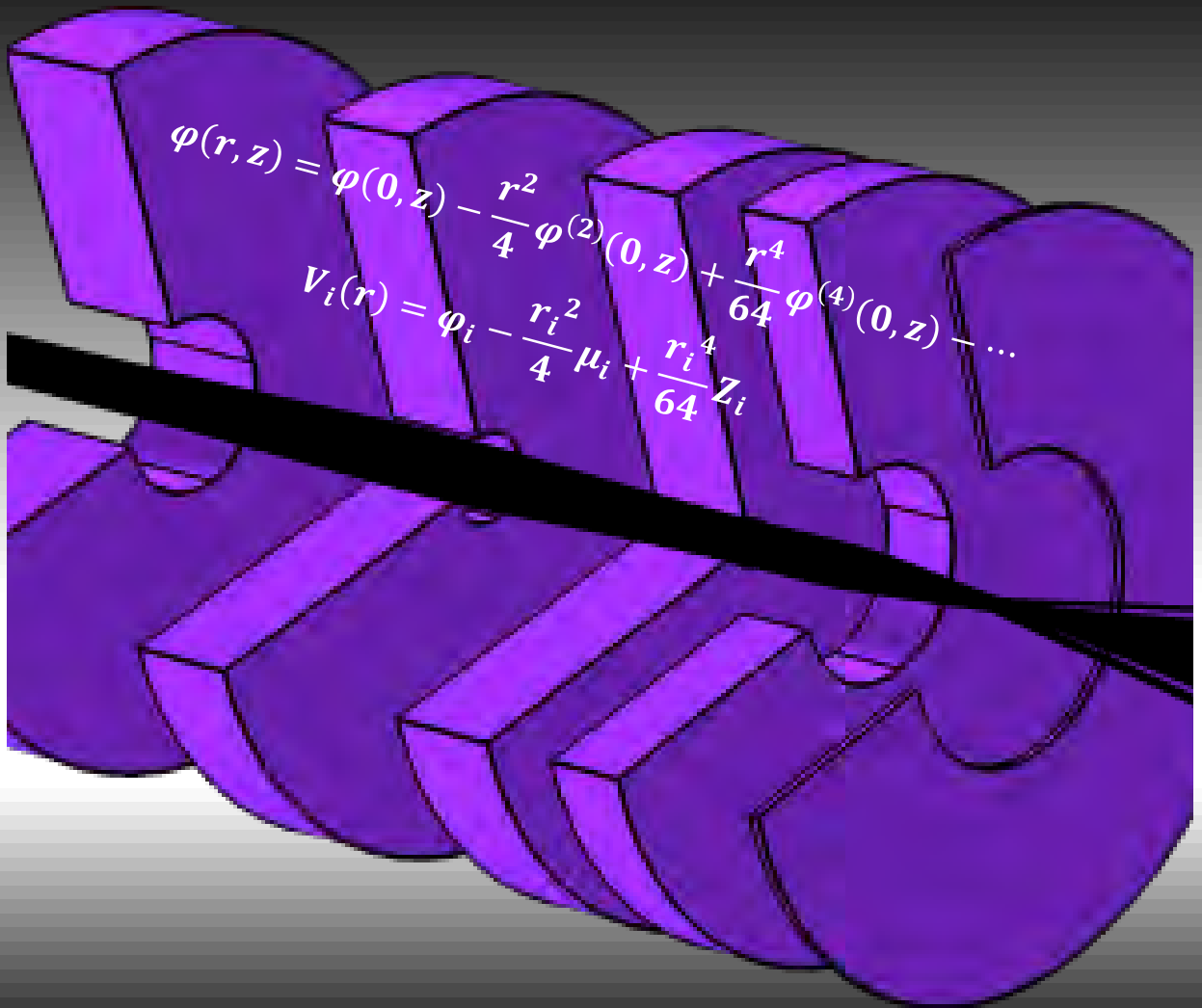
Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Electrostatic Lens System Design and Optimization Using Evolutionary Algorithms

Neda Hesam Mahmoudi Nezhad

Delft University of Technology



Electrostatic Lens System Design and Optimization Using Evolutionary Algorithms

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof.dr.ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
24th April 2024 at 17:30

By

Neda HESAM MAHMOUDI NEZHAD

Master of Science in Applied Physics, Delft University of Technology,
The Netherlands
born in Shiraz, Iran

This dissertation has been approved by the promotor.

Composition of the doctoral committee:

Rector Magnificus	Chairperson
Em. Prof. dr. ir. P. Kruit	Delft University of Technology, promotor
Dr. C. W. Hagen	Delft University of Technology, promotor
Dr. M. Ghaffarian Niasar	Delft University of Technology, copromotor

Independent members:

Prof. dr. ir. P. Bauer	Delft University of Technology
Prof. ir. P.T.M. Vaessen	Delft University of Technology
Prof. dr. A. Khursheed	Politecnico di Milano
Dr. T. Wirtz	Luxembourg Institute of Science and Technology

Reserve member:

Dr.ir. J.P. Hoogenboom	Delft University of Technology
------------------------	--------------------------------



Printed by Gildeprint | www.gildeprint.nl

Cover design by Neda Hesam Mahmoudi Nezhad

Cover illustration: A multi-electrostatic lens system with an electron beam passing through it, visualized using COMSOL.

ISBN 978-94-6496-106-5

An electronic version of this dissertation is available at <http://repository.tudelft.nl/>.

Copyright © 2024 by Neda Hesam Mahmoud Nezhad

To my husband Mohammad, my daughters Dorsa & Elysa and my parents

Contents

1.	Introduction	1
	References	7
2.	Field Calculation	9
2.1	Introduction	9
2.2	Field Calculation by Second Order Electrode Method (SOEM)	9
2.3	Field Calculation by FEM using COMSOL Multi-physics	16
2.4	Field Calculation by SOEM Versus FEM	23
	References	28
3.	Optimization	29
3.1	Optimization in General	29
3.2	Genetic Algorithm (GA)	34
3.2.1	Framework of Genetic Algorithm	34
3.2.2	GA operators	36
3.2.2.1	Encoding	36
3.2.2.2	Selection	37
3.2.2.3	Crossover	38
3.2.2.4	Mutation	40
3.2.2.5	Reproduction (Crossover Factor , Elitism)	40
3.2.2.6	Stopping Criteria	41
3.3	Particle Swarm Optimization (PSO)	41
3.3.1	Introduction	41
3.3.2	Modeling and Algorithm	41
3.4	Simulated Annealing (SA)	44
3.4.1	Introduction	44
3.4.2	Modeling and Algorithm	45
3.5	Calculus-Based Optimization (CBO)	47

3.6	Defining our case-study optimization problem	49
	References	51
4.	Multi-Electrode Lens Optimization Using Genetic Algorithms	57
	Abstract.....	57
	Keywords.....	57
4.1	Introduction	58
4.2	SOEM.....	59
4.3	Lens geometry.....	60
4.4	The objective function	61
4.5	Accuracy of SOEM	63
4.6	Genetic Algorithm optimization using SOEM	65
4.6.1	GA Parameters	67
4.7	Results of GA optimization with SOEM.....	68
4.8	GA optimization with SOEM and COMSOL	71
4.9	Conclusion.....	74
	References	75
5.	Multiple Criteria Optimization of Electrostatic Electron Lenses Using Multi-Objective Genetic Algorithms	77
	Abstract.....	77
5.1	Introduction	78
5.2	Lens system geometry and optimization parameters	80
5.3	The Objective functions	81
5.4	Ray-tracing: SOEM versus COMSOL.....	82
5.5	Optimization using MOGA	84
5.5.1	MOGA using potential calculation with SOEM	85
5.5.2	MOGA using potential calculation with COMSOL.....	86
5.5.3	MOGA using potential calculation with COMSOL and MOGA+SOEM-optimized input systems	88

5.6	MOGA+SOEM+COMSOL versus MOGA+COMSOL.....	90
5.7	Examples of systems optimized by MOGA+SOEM+COMSOL.....	91
5.8	Conclusion	93
	References	94
6.	Local versus Global Optimization of Electron Lens System Design	97
	Abstract	97
	Keywords	98
6.1	Introduction.....	99
6.2	Optimization Parameters	100
6.2.1	Objective Function.....	100
6.2.2	Free Variables and Bounds	101
6.2.3	Constraints.....	102
6.3	Implementation in MATLAB	102
6.3.1	Applying Global and Local Optimizers	102
6.3.2	Constraint Implementation	103
6.3.3	Comparison of two approaches on constraint implementation	105
6.4	local versus global optimization	107
6.5	Conclusion	110
	References	111
7.	Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses.....	113
	Abstract	113
	Keywords	113
7.1	Introduction.....	114
7.2	Optimization Problem	114
7.3	GA Tuning Parameters.....	115
7.3.1	Population size and number of Generation	115
7.3.2	Crossover and Mutation	118

7.3.3	Crossover Fraction	120
7.4	Conclusion.....	121
	References	122
8.	Comparison of Different Optimization Techniques in Electron Lens Design	123
	Abstract.....	123
	Keywords.....	124
8.1	Introduction	125
8.2	Defined Optimization Problem	126
8.3	Optimization Techniques Applied in Our Case-Study	128
8.3.1	Calculus-Based local optimization	128
8.3.2	SA	128
8.3.3	GA.....	129
8.3.4	PSO	129
8.4	Results and Analysis.....	130
8.5	Conclusion.....	133
	References	134
9.	A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization	137
	Abstract.....	137
	Keywords.....	138
9.1	Introduction	139
9.2.1	Fundamental equations of the quintic spline	144
9.2.2	Solving the Laplace equations up to the fourth order derivatives	148
9.2.3	Calculation of Δf_1 and Δf_{N-1}	149
9.2.4	Matrix formulation of the equations	152
9.3	Accuracy of FOEM	155
9.4	Optimization of electrostatic electron lens systems based on FOEM and a comparison with SOEM- and FEM-based optimization	161

9.5	Conclusion	171
	References	173
10.	Fast Electrostatic Lens System Design (FELSD) Program	175
10.1	“Single Lens System Calculation” (SLC)	175
10.2	Lens System Optimization (“LSO”)	181
11.	Summary and Conclusion	185
12.	Samenvatting en Conclusies	193
	Appendix A: The Theory of Cubic Spline	201
	Appendix B: Insights into COMSOL Meshing.....	205
	Appendix C: Matrix Representation of FOEM equations	213
	Acknowledgements	217
	About the author	221
	List of Publications.....	222

1. Introduction

Optimization, in general, is the procedure of finding a system (defined by its parameters) within the specified ranges, which is best at satisfying the defined optimization goal based on its particular criteria. In this thesis, the aim is to perform optimization for electrostatic lens system design using evolutionary algorithms. To explain the work, the main concepts from the title of the thesis should first be outlined: “What are the electrostatic lenses?”, “What does the design/optimization of an electrostatic lens system mean?” ,” How does the charged particle optical lens designer perform the design optimization problems?”

The lens systems are categorized into two main groups, namely optical lenses and charged particle optical lenses. The optical lenses are mainly made from glasses ,that we are all familiar with, are used to bend or focus the light beam according to light optics principles [1]. The other group is the charged particle optical lenses, which are categorized into two groups: electrostatic lenses and magnetic lenses. These lenses are used to control the charged particle passing through them using electric fields (electrostatic lenses) or magnetic fields (magnetic lenses) based on the charged particle optics principles [2]. Here, in our case-study we only consider the electrostatic charged particle optical lenses. Hence, hereafter by charged particle optical lenses or electron lenses we mean the electrostatic lenses.

Figure 1.a shows optical lenses where light rays are passing through and Figure 1.b illustrates this schematically. This illustration is similarly presented for the electrostatic lenses while charged particle rays pass through in Figure 2 (a) and (b).

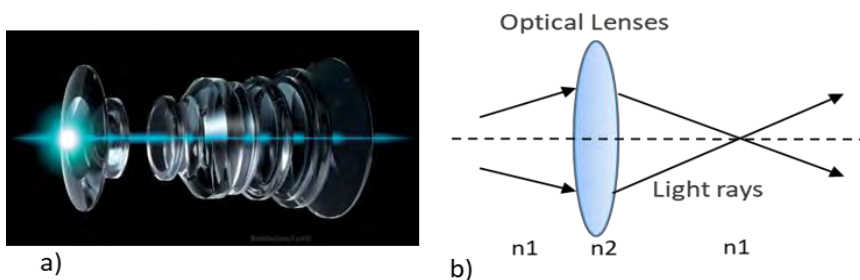


Figure1. Optical lenses (a), with the schematic illustration of the light rays passing through the lens (b).

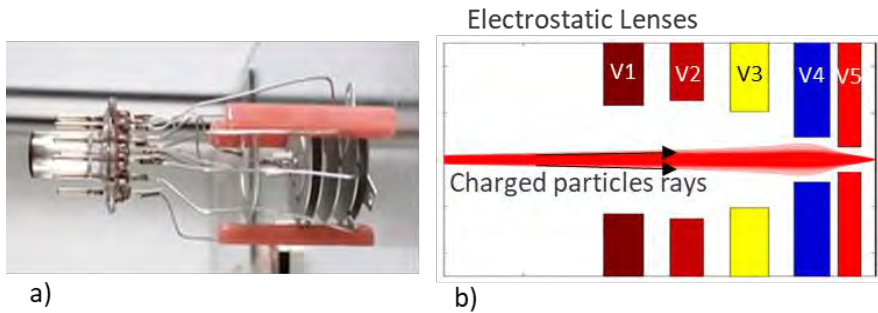


Figure 2. A multi-electrode electrostatic lens (a), with the schematic illustration of charged particle rays passing through the lens (b).

The main application of the lenses are in imaging systems. Rays coming from an object point intersect the image plane at the points which can be derived using ray tracing methods. In reality, the image of this point is not a single point, but a spread spot. The size formed from the collection of the intersection points of the traced rays from an object point with the image plane is called 'spot size' [3]. A better qualified lens system is the one where its image suffers less from the aberrations to have a smaller spot size.

The aim of the design of electrostatic lens systems is generally to focus the Primary Beam (PB) [4] at a specific point while having the least aberration induced by the lens. In some situations more functions are aimed to be optimized such as, to have the highest detection efficiency of Secondary Electrons (SEs) [5] at the image place. There are also typical constraints involved in the design problem such as the maximum allowable electric field between sequential electrodes to prevent discharges. It should be mentioned that in our case-studies, the lens systems are considered to be influenced by only the spherical and chromatic aberrations. To find a lens system which is optimized regarding the defined performance, while simultaneously taking constraints into account, the lens parameters (geometries and voltages) can all be varied.

To optimize a lens system, the objective function (the function which the designer has defined to evaluate the performance of the system) should be calculated and evaluated for each system. The objective function depends on the electric potential distribution in the space along the axis created by the lenses. To obtain the electric potential, the existing methods are generally based on meshing either the entire space of the lens system or the electrode surfaces using Finite Element Method (FEM) [6], Finite Difference Method (FDM) [7] or charge density method [8]. Hence, calculation of the objective function for such systems are computationally expensive.

Designing electrostatic lens systems manually is therefore quite a challenge and a laborious work for the optical designers when a large number of lens parameters is involved, such as the electrode thicknesses, the aperture sizes of the electrodes, the distances between the electrodes, and the electrode voltages. This becomes even more challenging once the number of electrodes increases.

Currently it is not difficult to find electron-optical software to conduct accurate field calculations such as EOD [9], GPT [10], CPO [11], etc., however, they only change one lens geometry parameter or the voltages to influence the aberrations or to auto-focus, but are not capable of changing the whole shapes of the lenses. Assuming these software tools can be also used in an optimization loop by changing all lens geometries and voltages as the free parameters, it can take months to get the results [12, 13]. For instance, the design of even a simple system using COMSOL [14] takes such a long computational time that the designer might not have the patience to wait for the result (assuming the computational memory allows them to do so).

Therefore, although optimization routines are very powerful techniques for finding optima of complex functions, they have not been used extensively to optimize charged particle optical lens designs, and applying a fully-automated optimization routine has not yet been feasible. So far, such optimizations are mostly carried out manually in a trial and error fashion.

To reduce the computation time of field calculations and to make an automated optimization process feasible, we were searching for a method which enabled us to perform the field calculation more quickly. Adriaanse et al. (1989) presented a fast method to calculate the electric field by means of approximation of the axial potential with a cubic spline and solving a set of linear equations [15-17]. In their method, the terms of the Laplace equation with higher order than the second derivative are truncated and the method is named the Second-Order Electrode Method (SOEM). At that time (1989) when SOEM was presented, it worked with local optimization, for simple systems with only one minimum in the parameter space. However, nowadays after the technological revolution resulting in high speed computers, we wanted to once again consider SOEM but in a more broad category as an aid in the process of electron lens optimisation. Since the field calculated by this method is based on approximation of Laplace equation, an inaccuracy in the results was expected. We hence first performed a study on calculation of the accuracy of the field calculated using SOEM compared to the existing accurate field calculation methods such as FEM. In addition, the computational time of these two methods are evaluated and

compared. The results are presented in section 2.4, after a brief explanation on the theory of SOEM (section 2.2) and FEM (section 2.3) with their numerical implementations in our case studies in MATLAB.

The comparisons of SOEM with FEM showed promising results for SOEM, having some inaccuracy in the calculation but compromising with its very fast field calculation. The method of SOEM hence is considered to be implemented in addition of FEM in a surrogate model [18] to aid our optimization problem. Surrogate models, also known as surrogates, are computationally inexpensive approximation models, employed to assist in evaluation of computationally expensive functions to reduce the computational time spent on the problem. Studies on surrogate-assisted evolutionary algorithms such as Genetic Algorithm (GA) [19, 20] began a decade ago. In subsequent years this method has been implemented in many different applications to optimize single and multi-objective optimization problems and found to be very efficient [21-23]. The surrogates (called the “Low Fidelity model”) can be implemented through different strategies combined with the accurate function calculation (known as the “High Fidelity model”) to assist the optimization process.

A recent work [24] has been published on the design of an ion optical device with many possible free parameters, involving multi-objective functions (up to two). There, an adjoint variable method is implemented. However, the optimization based on the adjoint method is suitable only for a few objective functions, while evolutionary optimization techniques such as GA are capable of optimizing multiple objective functions. Another difference is that the adjoint based optimization is a local optimization technique, whereas the evolutionary algorithms are considered as global optimization techniques due to their metaheuristic characteristics that can automatically search more spaces through the objective function landscape and prohibit trapping in local optima. In the design of electrostatic lens systems, such as in our case study, the objective function landscape (even in the case of a single objective function) seems to have many local minima. In addition, multiple objective functions are usually aimed to be optimized, while the number of free parameters is not huge (in the order of tens). In such circumstances, EA/GA appears to be a better fit to the problem.

In our work, we therefore propose a method in which a surrogate-assisted EA (mainly GA) is used. The procedure is such that a surrogate-assisted GA is used by first running a GA optimization using field calculation by SOEM (Low Fidelity), resulting in a number

of approximately good systems that are then inserted into a subsequent GA optimization using field calculation by COMSOL (High Fidelity).

The reason GA has been selected as the optimization technique was its good performance in optical lens system optimization and similar complex optimization problems. The implementation of surrogate models for a single objective function problem is proposed and presented in chapter 4 (published as [12]). Since in this model, both SOEM and COMSOL are implemented for the field calculation in an optimization aided by GA, it is called “SOEM-COMSOL-GA”. The “SOEM-COMSOL-GA” is applied on typical lens systems of 6 electrode lenses. the results of this optimization are presented together with the comparison with “SOEM-GA” and “COMSOL-GA”.

In chapter 4, the optimization with a single objective function is proposed which is hence called SOGA (Single Objective optimization using GA). As mentioned above, there are situations in which we have to use a multi-objective optimization problem, that we have called MOGA (multi-objective function optimization using GA). Study on MOGA while using SOEM and COMSOL as the field calculation for optimization of multi-electrode lens systems is successfully performed (published as [13]) and represented in chapter 5. A 5 electrode lens system is taken as another typical example of this study.

The ray-tracing for the situation where we only deal with a primary beam can be performed within paraxial approximation. However, for the situations where optimization of SEs is also aimed, real ray-tracing is needed. The details of how the ray-tracings are performed in different situations are explained in chapters 4 and 5. From chapters 4 and 5, we will see that GA could perform efficiently in our optimization problems, both for SOGA and MOGA situations.

However, although GA shows a high performance in our case-study (presented in chapters 4 and 5), we were curious to know first whether a local optimization technique could also handle the problem and replace a global optimization such as GA to result in a short computational run time for the optimization process. A study to compare local versus global optimization for multi-electrostatic lens systems is performed with a typical 6 electrode lens system. The results are presented in chapter 6 (published as [25]).

So far the results presented were performed by a GA with its tuning parameters set as the default of MATLAB. A further analysis used different tuning parameters of GA to

find those with which the GA performs with the highest efficiency. This analysis is presented in chapter 7 (published as [26]).

Another study is done to discover if other well-known optimization techniques can outperform GA. This study is presented in chapter 8 (published as [27]).

Since calculation of the electric field plays a major role in all computations and simulations, another important study performed in this thesis aims to find another fast method of field calculation which has a higher accuracy. We succeed in this goal by proposing a novel fast method of field calculation, with reasonably high accuracy.

Chapter 9 presents this method (submitted for publication as [28]). The method is performed by solving the Laplace equation near the optical axis by keeping the terms in the off-axis potential expansion up to the fourth order derivatives of the axial potential with respect to the optical axis coordinate. This method is hence named the Fourth-Order Electrode Method (FOEM) by the authors. To derive the equations, a quintic spline approximation [29] of the axial potential is calculated by solving three sets of linear equations simultaneously. The sets of equations are extracted from the Laplace equation and the fundamental equations that describe a quintic spline. The accuracy and speed of this method is compared with other field calculation methods, such as the Finite Element Method (FEM) and Second Order Electrode Method (SOEM). Using the previously developed Genetic Algorithm (GA) based optimisation program for electrostatic lens systems the effectiveness of this new potential calculation method in calculating and optimising optical parameters of electrostatic lens systems is compared with FEM and SOEM based optimisations and the results are presented.

A user-friendly software is developed using all presented methods in MATLAB [30]. The software and related flowcharts are presented in chapter 10.

References

- [1] Hecht, Eugene, Optics (4th edition), Addison Wesley, (2002).
- [2] V.K. Zworykin, J. Hillier, G.A. Morton, E.G. Ramberg, and A.W. Vance, Electron Optics and the Electron Microscope, John Wiley and Sons, Inc. (1945).
- [3] Robin Smith, Geometrical Optics and Lens Design Lecture Notes, Imperial College, London, (2012).
- [4] Miklos Szilagyi, Electron and Ion Optics, Plenum Press, New York (1988).
- [5] L. Reimer, "Scanning Electron Microscopy, Physics of Image Formation and Microanalysis", Springer Series in Optical Sciences, Vol. 45 (1998).
- [6] The Finite Element Method in Electromagnetics (IEEE Press) 3rd Edition, Jianming Jin, John Wiley & Sons. IEEE Press, (2014).
- [7] Instructive Review of Computation of Electric Fields using Different Numerical Techniques, J. Faiz, M. Ojaghi, Int. J. Engng Ed. Vol. 18, No. 3, pp. 344, (2002).
- [8] Frank H. Read, Nicholas J. Bowring, The CPO programs and the BEM for charged particle optics, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 645, Issue 1, Pages 273-277 (2011).
- [9] B. Lencova, and J. Zlamal, Electron Optical Design program package EOD 5.001.
- [10] S.B. van der Geer, M.J. de Loos, The General Particle Tracer code: Design, Implementation and Application, Eindhoven: Technische Universiteit Eindhoven, (2001).
- [11] <https://www.integratedsoft.com/applications/charged-particle-optics>, (2021).
- [12] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C.W. Hagen and P. Kruit, Int. J. Mod. Phys. A, Vol 34, 1942020 (2019).
- [13] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, P. Kruit and C. W. Hagen, J. Vacuum Science & Technology B 39, 062605 (2021).
- [14] COMSOL Multi-physics, version 5.3a (2017).
- [15] J. Adriaanse, H. van der Steen and J. Barth, J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena 7, 651 (1989).
- [16] H. van der Steen, J. Barth and J. Adriaanse, Nucl. Instrument. Methods Phys. Res. Sec. A: Accelerators, Spectrometers, Detectors and Associated Equipment 298, 377 (1990).
- [17] H. van der Steen and J. Barth, J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena 7, 1886 (1989).

- [18] B. Liu, Q. Zhang, and G. Gielen, *Simulation-Driven Modeling and Optimization*, edited by S. Koziel, L. Leifsson, and X. Yang, Springer, Cham (2014).
- [19] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, London (1998).
- [20] N. Hesam Mahmoudi Nezhad, *Optical system optimization using genetic algorithms*, Master's thesis, Delft University of Technology (2014).
- [21] X. Cai, L. Gao, X. Li, and H. Qiu, *Swarm Evol. Comput.* 48, 288 (2019).
- [22] A. Diaz-Manriquez, G. Toscano-Pulido, and W. Gomez-Flores, *Proceedings of 2011 IEEE Congress on Evolutionary Computation (CEC)*, p. 2155. New Orleans, LA, 5–8 June, (2011).
- [23] Y. Jin, *Swarm Evol. Comput.* 1, 61 (2011).
- [24] L. T. Neustock, P. C. Hansen, Z. E. Russell, and L. Hesselink, *Sci. Rep.* 9, 11031(2019).
- [25] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, C. W. Hagen, and P. Kruit, *2020 IEEE 6th International Conference on Optimization and Applications, (ICOA)* (2020).
- [26] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, C. W. Hagen, and P. Kruit, *2023 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modelling and Optimization (NEMO' 2023)*.
- [27] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, C. W. Hagen, and P. Kruit, *2023 IEEE 9th International Conference on Optimization and Applications, (ICOA)* (2023).
- [28] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, C. W. Hagen, and P. Kruit, is submitted to Springer Nature SNPP (2023).
- [29] T. Yuan Ying & A. Saaban , *Quintic Spline Method for Solving Linear and Nonlinear Boundary Value Problems*, O. Alayed, *Sains Malaysiana* 45(6), p. 1007, (2016).
- [30] MATLAB, version r2016 b, <https://www.mathworks.com/products/matlab.html>

2. Field Calculation

2.1 Introduction

As already stated in chapter 1, the common techniques to accurately calculate the lens field, are computationally expensive. The so-called SOEM method, is mentioned as a method for fast, but rough, field calculation. However, the theory behind SOEM, is not explained there. Moreover, questions as “how much faster can SOEM make the calculation?” and “How much is the in-accuracy introduced in the results by SOEM, compared to the accurate method of FEM?” are yet left un-answered. This chapter is devoted to resolving these questions.

In section 2.2., the theory of field calculation by SOEM is explained. In the next section, 2.3., field calculation by COMSOL using FEM is briefly introduced. Then, in section 2.4., the two methods are implemented for a number of case-studies. The comparison of the results taken from both methods, regarding their computational time and their accuracy, are presented at the end of the section.

2.2 Field Calculation by Second Order Electrode Method (SOEM)

Adriaanse et al. [1, 2], in 1989, proposed a technique for a fast axial field calculation which only needs to mesh the optical axis, rather than the whole space. This method is explained in detail below.

To derive the field along the optical axis, the only information at hand are the voltages on each electrode and the Laplace equations which correlate those voltages with the axial potentials on their corresponding points along the axis. This relation for the potential, in a rotationally symmetrical system, can be written as:

$$\varphi(r, z) = \varphi(0, z) - r^2\varphi^{(2)}(0, z)/4 + r^4\varphi^{(4)}(0, z)/64 - \dots \quad (2.1)$$

In which, $\varphi(0, z)$ is the potential along the axis. Denotations z and r stand for the axial and radial coordinate, respectively. $\varphi^{(n)}(0, z)$ expresses the n th derivative of φ with respect to z . To make the equations more easily solvable, Adriaanse et al. omitted third and higher order terms in the Laplace equation:

$$\varphi(r, z) = \varphi(0, z) - r^2 \varphi^{(2)}(0, z)/4 \tag{2.2}$$

For that reason, this method is called Second Order Electrode Method, abbreviated to SOEM.

To numerically solve this problem, the surface of each electrode is divided into a few finite number of points. To each of these points a point on the optical axis is assigned. The meshing of the optical axis is therefore determined by the pre-defined divisions at the surface of each electrode. The voltages at each meshed point along the optical axis, are correlated to the voltages of their corresponding points at each electrode, using Eq. 2.2. This is schematically shown in Figure 2.1, where a system of multi-electrode lenses (here four), are sketched with their voltages defined as V1, V2, V3 and V4, respectively.

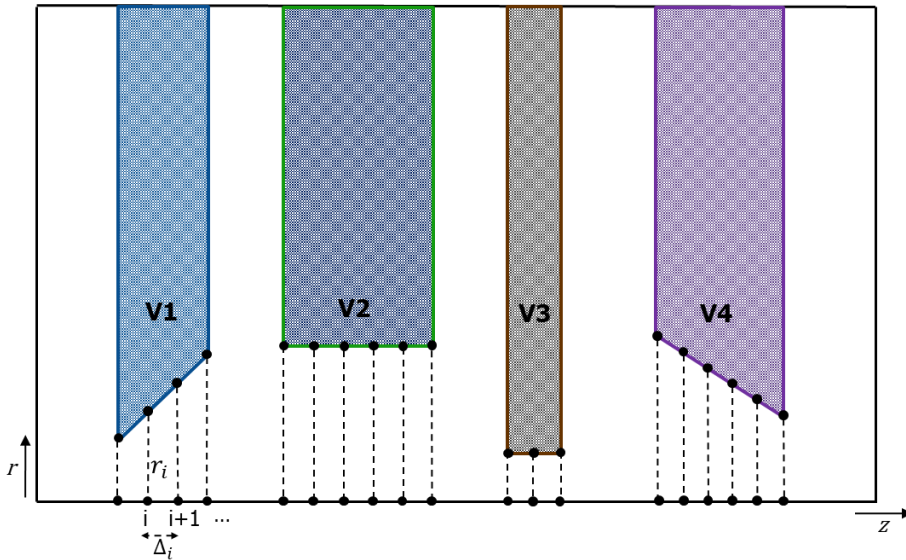


Figure 2.1. A schematic representation of a four-electrode lens system, with the meshed points on the electrodes and their corresponding points along the optical axis.

Considering the meshed points along the axis defined as z_i , for $i = 1, \dots, N$, equation 2.2 can be numerically written as:

$$\varphi_i(r_i, z_i) = \varphi_i(0, z_i) - \frac{r_i^2 \varphi_i(0, z_i)^{(2)}}{4}, i = 1, \dots, N \tag{2.3}$$

Where, $\varphi_i(r_i, z_i)$ is equal to the voltages V_i of the electrodes at their corresponding point i .

For simplicity, the axial potential $\varphi_i(0, z_i)$ can be represented by φ_i . The equation 2.3 can be re-written as:

$$V_i = \varphi_i - \frac{r_i^2 \varphi_i^{(2)}}{4}, i = 1, \dots, N \quad (2.4)$$

Equation 2.4 involves a set of N equations, with $2N$ un-known variables, φ_i and $\varphi_i^{(2)}$. To solve these equations and find axial potential φ_i , another set of N equations is needed.

However, the potential of these points in a lens system are not independent from each other, which means, N equations of Eq. 2.4 have extra relations with each other. The simplest way to establish these relations is to use polynomial functions.

Since in the simplified Laplace equations (Eq. 2.2-2.4), the second derivative exists, the minimum polynomial order that can properly show those relations is the third order. Therefore, Adriaanse et al. have used cubic splines (i.e. piece wise built-function from polynomials of the third-order) to achieve these extra relations [1], from which another set of equations in addition to Eq.2.4 are found, that makes the problem uniquely solvable.

To apply cubic splines in our field calculation problem, $\varphi(x)$ should be represented as a cubic spline. The theory of the cubic splines and its formulation are presented in Appendix A. There, a cubic spline function is denoted by $S(x)$. The formulation of cubic splines which correlates the spline function with its second derivatives is expressed in equation A.9. To represent the axial potential in a format of cubic splines, $S_i(x)$ and $S_i^{(2)}(x)$ in equation A.9 should be replaced by $\varphi_i(x)$ and $\varphi_i''(x)$, to represent the axial potential in a format of discrete cubic splines. This results in:

$$\varphi_{i+1} \left(\frac{1}{\Delta_i} + \frac{1}{\Delta_{i+1}} \right) = \frac{\varphi_{i+2}}{\Delta_{i+1}} + \frac{\varphi_i}{\Delta_i} - \frac{1}{6} (\varphi_i'' \Delta_i + 2\varphi_{i+1}'' (\Delta_i + \Delta_{i+1}) + \varphi_{i+2}'' \Delta_{i+1})$$

$$i = 1, \dots, N - 2 \quad (2.5)$$

This equation correlates φ_i with its two neighbouring points and their second derivatives. To obtain a similar form of notation as used in Adriaanse et al. [1], we change variable i to $i - 1$ in Eq. 2.5:

$$\varphi_i \left(\frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} \right) = \frac{\varphi_{i+1}}{\Delta_i} + \frac{\varphi_{i-1}}{\Delta_{i-1}} - \frac{1}{6} (\varphi''_{i-1} \Delta_{i-1} + 2\varphi''_i (\Delta_{i-1} + \Delta_i) + \varphi''_{i+1} \Delta_i)$$

$$i = 2, \dots, N - 1 \quad (2.6)$$

It should be noted that, this change does not influence the equation, but is applied only for the sake of consistency with Adriaanse et al.'s notations.

To solve the cubic spline in the normal way, N data points (x_i, y_i) are needed to fit $N - 1$ cubic splines to those data. Here, instead of data points, we have Eq. 2.6 which produces $N - 2$ equations. We therefore need to have two extra equations to create N equations in total.

To add two more required equations, it is assumed that the boundary conditions are to have the spline and its second derivative be zero at the borders (A.10 in Appendix A). This physically means to have zero field and derivative at the first and last electrodes:

$$\varphi_i = \varphi'_i = 0, \quad i = 1, N \quad (2.7)$$

Equations 2.6 and 2.7, together with equation 2.4, produce $2N$ equations. By solving them together, φ_i can be derived as:

$$\varphi_{i-1} \left(\frac{2\Delta_i}{3r_{i-1}^2} - \frac{1}{\Delta_{i-1}} \right) + \varphi_i \left(\frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} + \frac{4}{3} \frac{\Delta_{i-1} + \Delta_i}{r_i^2} \right) + \varphi_{i+1} \left(\frac{2\Delta_i}{3r_{i+1}^2} - \frac{1}{\Delta_i} \right)$$

$$= \frac{2V_{i-1}}{3r_{i-1}^2} \Delta_{i-1} + \frac{4V_i}{3r_i^2} (\Delta_{i-1} + \Delta_i) + \frac{2V_{i+1}}{3r_{i+1}^2} \Delta_{i+1}$$

$$i = 2, \dots, N - 1 \quad (2.8)$$

Denoting the coefficients of φ_{i-1} , φ_i and φ_{i+1} by a_i , b_i and c_i , and the right-side of the equation by d_i , Eq. 2.8 is written as:

$$a_i = \frac{2\Delta_i}{3r_{i-1}^2} - \frac{1}{\Delta_{i-1}}, \quad b_i = \frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} + \frac{4\Delta_{i-1} + \Delta_i}{3r_i^2}$$

$$c_i = \frac{2\Delta_i}{3r_{i+1}^2} - \frac{1}{\Delta_i}, \quad d_i = \frac{2V_{i-1}}{3r_{i-1}^2}\Delta_{i-1} + \frac{4V_i}{3r_i^2}(\Delta_{i-1} + \Delta_i) + \frac{2V_{i+1}}{3r_{i+1}^2}\Delta_{i+1}$$

$$i = 2, \dots, N - 1 \tag{2.9}$$

$$a_i\varphi_{i-1} + b_i\varphi_i + c_i\varphi_{i+1} = d_i \tag{2.10}$$

Equation 2.10, correlates the axial potential at each point to the axial potential at its two neighbouring points. This equation can be written as a matrix representation, as in the following:

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & a_N & b_N \end{pmatrix} \begin{pmatrix} \varphi_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \varphi_N \end{pmatrix} = \begin{pmatrix} d_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{pmatrix} \tag{2.11}$$

Or, in a close format:

$$[E]_{N \times N} \cdot [\Phi]_{N \times 1} = [D]_{1 \times N}$$

$$[E] = \begin{pmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & a_N & b_N \end{pmatrix}, \quad [\Phi] = \begin{pmatrix} \varphi_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \varphi_N \end{pmatrix}, \tag{2.12}$$

$$\text{and } [D] = \begin{pmatrix} d_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{pmatrix}$$

The multiplier matrix of the axial potentials' vector, $[E]$, forms a tridiagonal matrix. This matrix-vector equation, can be quickly solved by numerical methods, and matrix-solvers software such as MATLAB.

However, to solve this equation, all elements of matrix $[E]$ should be known. Elements a_i, b_i, c_i and d_i , for $i = 2, \dots, N - 1$ are known parameters that can be calculated from the geometry of the lenses and electrode voltages using equation 2.9, while the first and last row of this matrix, the elements b_1, c_1 and a_N, b_N , are un-known and cannot be derived from equation 2.9. To calculate these values, the boundary conditions should be implemented, which matches the two degrees of freedom of equation 2.12.

These boundary conditions can be derived from the fact that, for a lens defined in a region of a non-zero field, the axial electric field has to vanish at the two end points of the lens. As the electric field equals zero in the region outside the lens, its derivative is also zero in that region. Due to the continuity of the field and its derivative, it is concluded that φ_1'' and φ_N'' are equal to zero.

$$\varphi_1'' = \varphi_N'' = 0 \quad (2.13)$$

Another constraint is that the axial potentials at the two end points, should be equal to the potentials of the outer electrodes. Therefore, the boundary conditions at the two end points $i = 1$ and $i = N$ are:

$$\varphi_1 = V_1 \text{ and } \varphi_N = V_N \quad (2.14)$$

Where, V_1 and V_N , are the voltages at the first and last electrodes, respectively.

These constraints are schematically shown in Figure 2.2 for the points $i = 1$ and $i = N$.

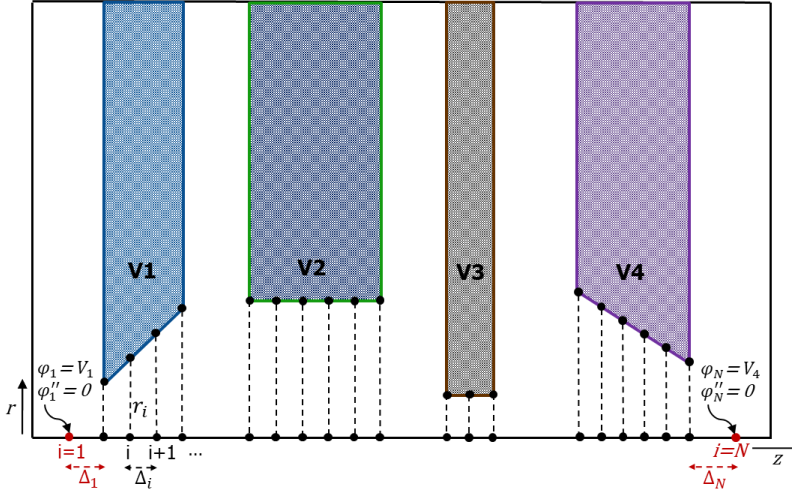


Figure 2.2. A schematic representation of a four-electrode lens system, with the meshed points on the electrodes and their correspondent points along the optical axis. The two end points ($i = 1, N$), at which the field becomes zero and the voltage is equal to the voltage of the last correspondent electrode, marked in red.

The distances from the outer surfaces of the first and last electrodes, up to where the electric field vanishes to zero, are defined as the first and last mesh of the lens (Δ_1 and Δ_N). These parameters are shown in figure 2.2 (marked in red). The values of Δ_1 and Δ_N are unknown and should be also calculated. The Taylor expansion is used to achieve this aim. The Taylor expansion for the axial potential at each point, with respect to its neighbouring point and its derivatives, can be expressed as:

$$\varphi_2 = \varphi_1 + \varphi_1' \Delta_1 + \frac{1}{2} \varphi_1'' \Delta_1^2 + \frac{1}{6} \varphi_1^{(3)} \Delta_1^3 \quad (2.15)$$

In which, $\varphi_1^{(3)}$ can be written in terms of φ_2'' , φ_1'' and the first mesh Δ_1 as:

$$\varphi_1^{(3)} = (\varphi_2'' - \varphi_1'') / \Delta_1 \quad (2.16)$$

As mentioned, there is no field at the points $i = 1$ and $i = N$ which means, $\varphi_1' = \varphi_N' = 0$. Due to the continuity of the first and second derivatives, this results in $\varphi_1'' = \varphi_N'' = 0$. Substituting this into equation 2.16, we obtain:

$$\varphi_1^{(3)} = \varphi_2'' / \Delta_1 \quad (2.17)$$

Replacement of equation 2.17 into equation 2.15, and considering φ_1' and φ_1'' equals to zero, leads to:

$$\varphi_2 = \varphi_1 + \frac{1}{6} \varphi_2'' \Delta_1^2 \quad (2.18)$$

Substitution of $i = 1$ in equation 2.4, and replacement of φ_1'' in terms of V_1 , r_1 and φ_1 , results in:

$$\varphi_1 = \varphi_2(1 - 2\Delta_1/3r_2^2) + V_1(2\Delta_1/3r_2^2) \quad (2.19)$$

To satisfy the constraint that φ_1 equals V_1 :

$$r_2 = (2/3)^{1/2} \Delta_1 \quad (2.20)$$

The above expressions are written for the first electrode. Similar expressions can be written for the last electrode to derive Δ_N in terms of r_{N-1} :

$$r_{N-1} = (2/3)^{1/2} \Delta_N \quad (2.21)$$

From equations 2.20 and 2.21, the first and last mesh can be derived. Knowing Δ_1 and Δ_N , all elements of matrix E are known and the matrix can be solved.

2.3 Field Calculation by FEM using COMSOL Multi-physics

To calculate electric fields inside an electrostatic lens system, the Laplace equation should be solved. Combining equations 2.22, 2.23, and 2.24, in which E is electric field (in unit of $\left[\frac{V}{m}\right]$), V is electric potential (in unit of $[V]$), D is electric displacement field (in unit of $\left[\frac{V}{m^2}\right]$), ρ is free electric charge density (in unit of $\left[\frac{C}{m^3}\right]$) and ε is permittivity of material (in unit of $\left[\frac{C}{Vm}\right]$), the Poisson equation (represented as equation 2.25) can be derived. If space charge is considered to be zero, the Poisson equation is reduced to the Laplace equation.

$$E = -\nabla V \quad (2.22)$$

$$\nabla \cdot D = \rho \quad (2.23)$$

$$D = \varepsilon E \quad (2.24)$$

$$\nabla^2 V = \frac{\rho}{\varepsilon} \quad (2.25)$$

In COMSOL Multi-physics [3] the module that includes Poisson's equation is known as 'Electrostatic physics' which can be chosen from AC/DC module. In order to solve

equation 2.25, boundary conditions and permittivity values of different regions which are present in the simulation should be defined by the user. To create an electrostatic model, first the dimension of the study has to be selected. Different possibilities are available such as 0D, 1D, 1D-Axisymmetric, 2D, 2D-Axisymmetric, and 3D (see Figure 2.3). If the geometry under study can be realized with lower dimensions, then it is recommended to carry out such simplification because the simulation will then be lighter, faster and more accurate. Since throughout this thesis the main focus is on axisymmetric electrostatic lenses, a 2D axisymmetric dimension is selected for all models.

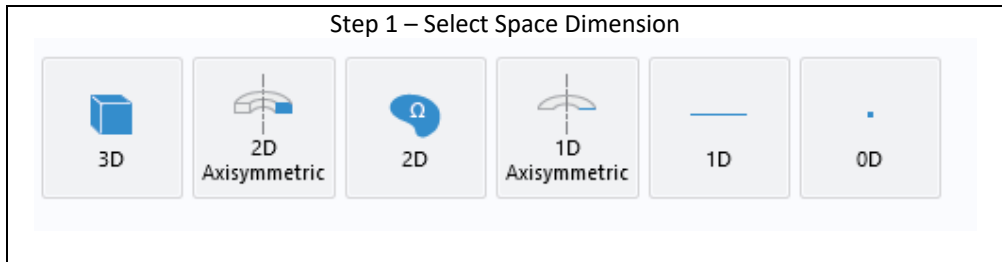


Figure 2.3. A schematic representation of the step for selecting spatial dimensions in COMSOL.

Next is to select the physics under study (Figure 2.4), which as explained above, is ‘electrostatic physics’. The next step is to select the study type; ‘stationary’, ‘time dependent’, or ‘frequency domain’ options are available. For electrostatic field calculation ‘stationary study’ should be selected.

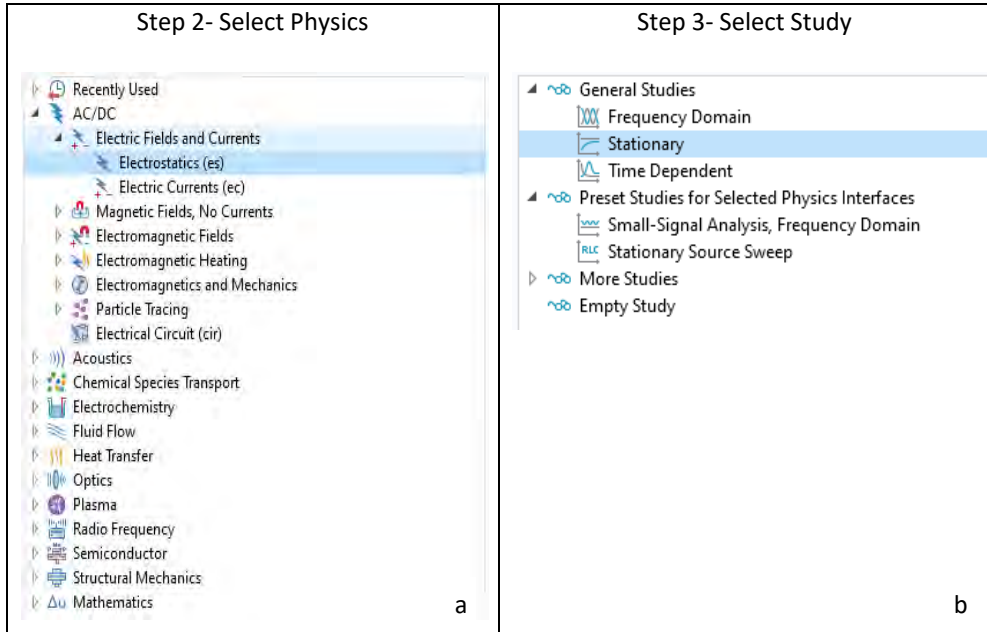


Figure 2.4 Schematic representation of the process of selecting the physics (a) and study options (b) in COMSOL.

Next step is to create the geometry of the lens and to assign the material properties of each domain (Figure 2.5). Since for electrostatic study the only material property that matters is permittivity, the permittivity of the material is the main parameter here. The lens is inside a vacuum and voltages of all electrodes will be defined, therefore the vacuum permittivity for all domains can be used.

Thereafter, boundary conditions including voltages of all electrodes are assigned and the whole geometry is meshed (Figure 2.6). Finally the simulation is executed and the results can be extracted. The results can be extracted at any position (point, line, etc.) in the geometry. Electric field calculation in COMSOL should be fully-automated and be implemented as a function inside the optimization program, therefore it is necessary to connect COMSOL to MATLAB and perform all the above mentioned steps via MATLAB.

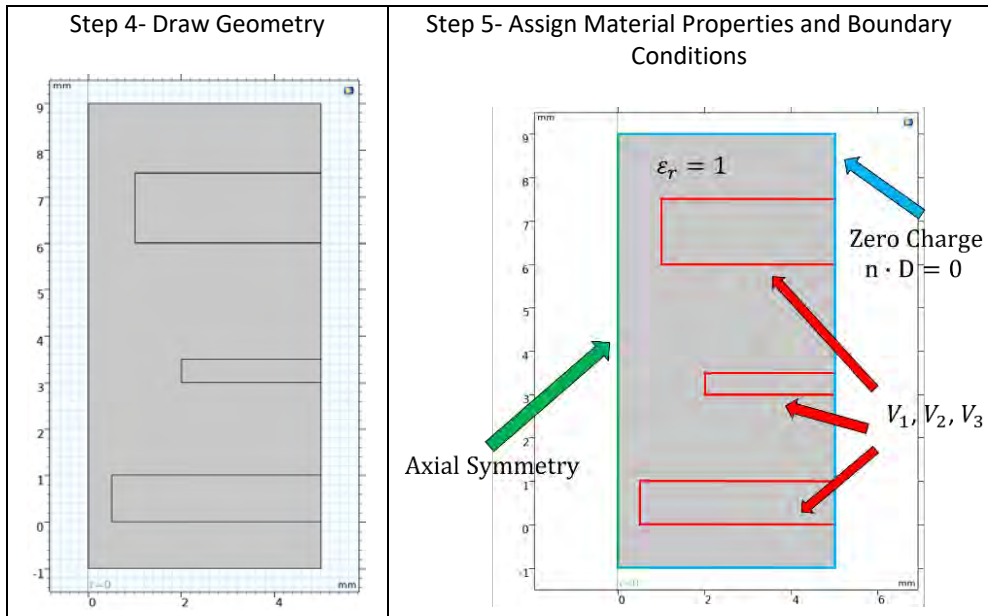


Figure 2.5 A schematic representation of the step of drawing the geometry (a) and assigning material properties and boundary conditions (b) in COMSOL.

Every step mentioned above can be performed directly in the graphical user interface of COMSOL or it can be implemented using equivalent codes in MATLAB. Finally, the code created in MATLAB is connected to the COMSOL server and uses the COMSOL engine to perform the calculation.

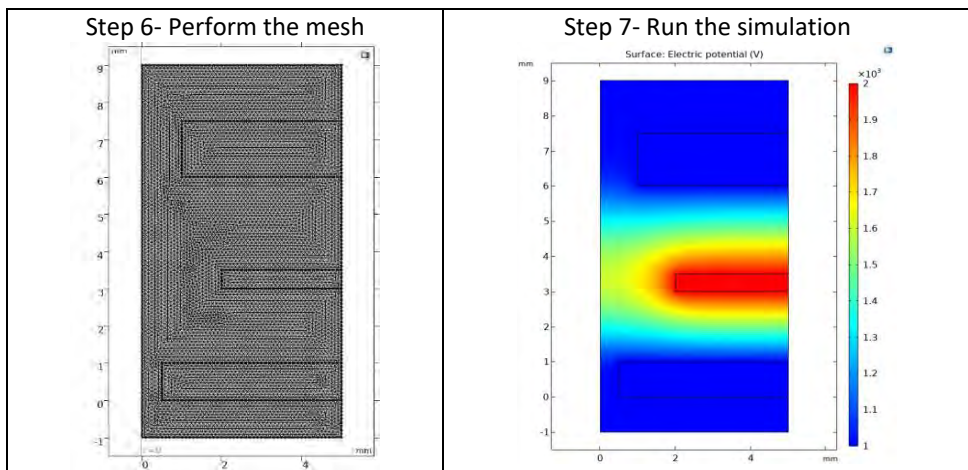


Figure 2.6 Schematic representation of the step of the meshing (a) and running the simulation (b) in COMSOL.

Since all steps are now in the format of MATLAB codes, it is straightforward to define a function which receives a list of input parameters needed to create the COMSOL model, to perform the calculation and to extract any parameter (e.g. voltage, electric field, etc. along the axis of symmetry) that is needed to be calculated by the simulation. In this case, the optimization program can send the input of each model to that function and the function calculates and outputs the desired parameters.

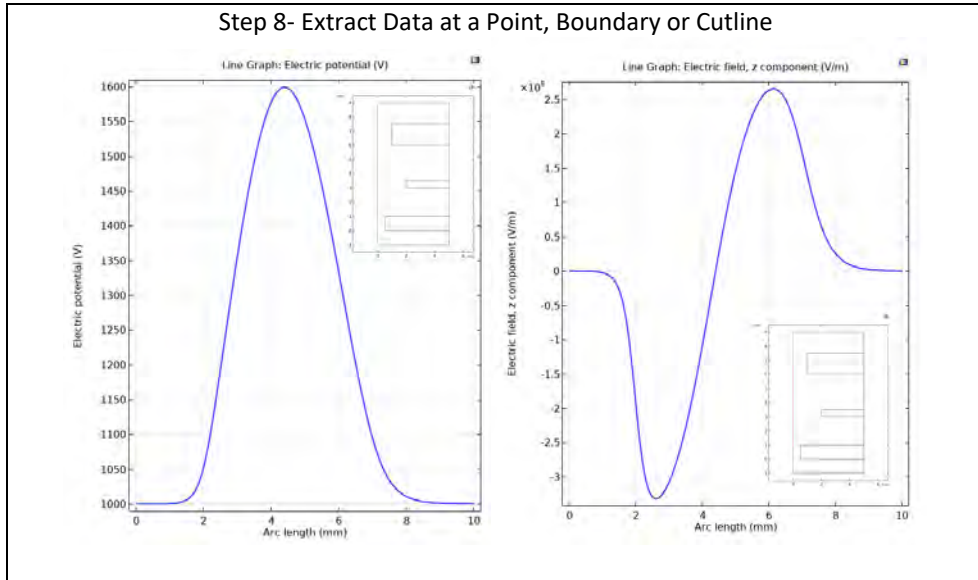


Figure 2.7 Schematic representation of the step of extracting data: electric potential versus arc length (a) and electric field versus arc length (b).

An axial symmetrical lens system with disk shaped lenses can be produced using rectangles in 2D axisymmetric geometry in COMSOL. In the program that has been created, to generate an N electrodes lens system, $4N-1$ parameters are needed. Voltages (V_1, V_2, \dots, V_N), inner radii (r_1, r_2, \dots, r_N), and thicknesses of the electrodes (h_1, h_2, \dots, h_N), are $3N$ parameters. The other $N-1$ parameters are the gaps between electrodes, (g_1, g_2, \dots, g_{N-1}). Using the input geometrical parameters it is possible to draw the electrode lens systems. In COMSOL a rectangle can be made with the following commands:

```
model.component('comp1').geom('geom1').create('r1', 'Rectangle');
model.component('comp1').geom('geom1').feature('r1').set('size', [5 1]);
model.component('comp1').geom('geom1').feature('r1').set('pos', [1 2]);
model.component('comp1').geom('geom1').run('r1');
```

Through the second line it is possible to set the length and width of a rectangle. The width is set equal to an electrode thickness. The length is the subtraction of the inner

electrode radius from a fixed number w . The w is the radius of the lens column (from the axis to the outer radius) which should be defined by the user, as shown in Figure 2. 8.

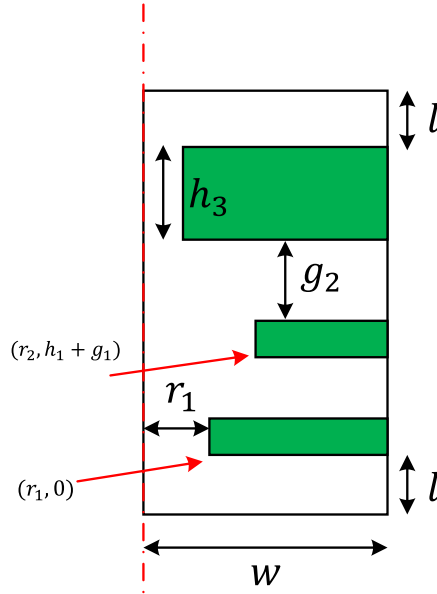


Figure 2.8. The geometry of an electrode as it is defined in COMSOL.

The position of the rectangle is identified based on the position of the lower left corner of the rectangle. The first electrode is placed at position $(r, z) = (r_1, 0)$. The position of the next electrode is defined based on the inner radius of that electrode, and also the gap between that electrode and the previous electrode. Using a loop, all electrodes are first created and finally the whole lens system is placed inside a rectangle with a width of w and height of:

$$2l + \sum_{i=1}^{N-1} (h_i + g_i) + h_N \quad (2.26)$$

In which l is a length that is defined by the user before the first and after the last electrode. After this stage each electrode is assigned with its voltage. After running the simulation desired parameters have to be extracted.

For paraxial ray tracing, the voltage distribution and its first and second derivatives along the axis of symmetry are needed. To achieve a good resolution of the desired parameters the axis of symmetry is discretized using uniformly distributed points

separated with a distance of dz to be set by the user (our default value is $10 \mu m$). The vector of the data points on the axis of symmetry are then defined by:

$$z = \sum_{j=0}^{2l/dz} -l + (j * dz) + \sum_{i=1}^{N-1} (h_i + g_i) + h_N \quad (2.27)$$

The following commands can be used to extract the abovementioned parameters along the axis of symmetry. In the following, zz is a data set on the axis of symmetry over which the parameters are to be evaluated. It consists of a matrix of 2 rows and a number of columns equal to the size of vector z . The first row contains values of r (all equal to 0 because it is on axis of symmetry) and on the second row values of vector z .

```
data00 = mphinterp(model,'V','coord',zz);
data10 = mphinterp(model,'es.Ez','coord',zz);
data20 = mphinterp(model,'d(es.Ez,z)','coord',zz);
```

For real ray tracing, it is necessary to extract electric field in the whole space. The following commands are used to extract r and z components of electric field on all mesh points in the space.

```
Ez = mpheval(model, 'es.Ez');
EZ=[transpose(Ez.p(1,:)) transpose(Ez.p(2,:)) transpose(Ez.d1(1,:))];
Er = mpheval(model, 'es.Er');
ER=[transpose(Er.p(1,:)) transpose(Er.p(2,:)) transpose(Er.d1(1,:))];
```

Since the mesh points are irregularly distributed in space, it is necessary to map the value of the electric field from the mesh points obtained by COMSOL, on a grid like mesh to make the calculation of real ray tracing easier. In the following, the first line creates the grid data, lines 2-4 are the data obtained from COMSOL (r coordinate of the mesh point, z coordinate of the mesh point, and the value of z component of the electric field at that mesh point). Line 5 uses a MATLAB function to create an interpolation function and line 6 evaluates that function at the regular grid mesh point that is created on line 1. In this way, the z component of the electric field can be evaluated at the regular grid points. A similar approach is used to find the r component of electric field at the regular grid points.

```
[xqz,yqz] = meshgrid(Min_x:mesh_dr:Max_x,Min_y:mesh_dz:Max_y);
xz=EZ(:,1);
yz=EZ(:,2);
vz=EZ(:,3);
F = scatteredInterpolant(xz,yz,vz);
vqz=F(xqz,yqz);
```

2.4 Field Calculation by SOEM Versus FEM

To compare the accuracy and computational time of field calculations using the SOEM [1, 2] and FEM methods, the tests on six different typical electrostatic lens systems have been conducted, consisting of 3, 4, and 5 electrodes. These systems are represented in 2D in Figure 2.9. The axial potential calculation is performed using the methods of SOEM and FEM methods (implemented with COMSOL[3]). MATLAB [4] is used for the SOEM calculations. Initially, the axial potential and its derivatives are compared graphically. For the purpose of graphical comparison, since all lens systems exhibit similar trends, the graphs related to only one system (system 1 from Figure 2.9) have been depicted and presented. The overlaid graphs can be found in Figures 2.10 and 2.11.

In electron optical lens systems, the aberration coefficients are the factors which determine the quality of the lens systems. The lower the values of these parameters, the less the aberrations exist and therefore the better the lens system [5-7]. In our case-study, the lens systems are only suffering from the spherical and chromatic aberrations. Hence, the chromatic and spherical aberration coefficients (denoted by C_c and C_s) are the deterministic factors in the lens design and optimization here. These parameters are functions of the axial potential and its derivatives. The presentation of the formulation is given in equations 2.28 and 2.29. More details on the formulations and their derivations are skipped here and can be found in the references [7-9].

$$C_s = \frac{1}{16\phi_0^{\frac{1}{2}}} \int_{z_0}^{z_i} \phi^{\frac{1}{2}} \left(\frac{5}{4} \left(\frac{\phi''}{\phi} \right)^2 + \frac{5}{24} \left(\frac{\phi'}{\phi} \right)^4 r_\alpha^4 + \frac{14}{3} \left(\frac{\phi'}{\phi} \right)^3 r_\alpha^3 r_\alpha' - \frac{3}{2} \left(\frac{\phi'}{\phi} \right)^2 r_\alpha^2 r_\alpha'^2 \right) dz \quad (2.28)$$

$$C_c = \phi_0^{\frac{1}{2}} \int_{z_0}^{z_i} \left(\frac{3}{8} \right) \frac{\phi'^2}{\phi^2} r_\alpha^2 dz \quad (2.29)$$

Here, ϕ_0 represents the potential at the object side, and $r_\alpha(z)$ denotes the principal imaging ray [5], which travels from the object side along the optical axis at an angle 1. The principal imaging ray $r_\alpha(z)$ is determined through ray tracing.

Figure 2.9 presents six different typical electrostatic lens systems (in 2D), which were chosen as test systems to compare the methods of SOEM and FEM. The colours in the table indicate the voltages applied to each electrode. To enhance visualization of the lens system, a schematic representation of the primary beam passing through the lens system (highlighted in red) is included. The units along the axes of the graphs are in millimetres.

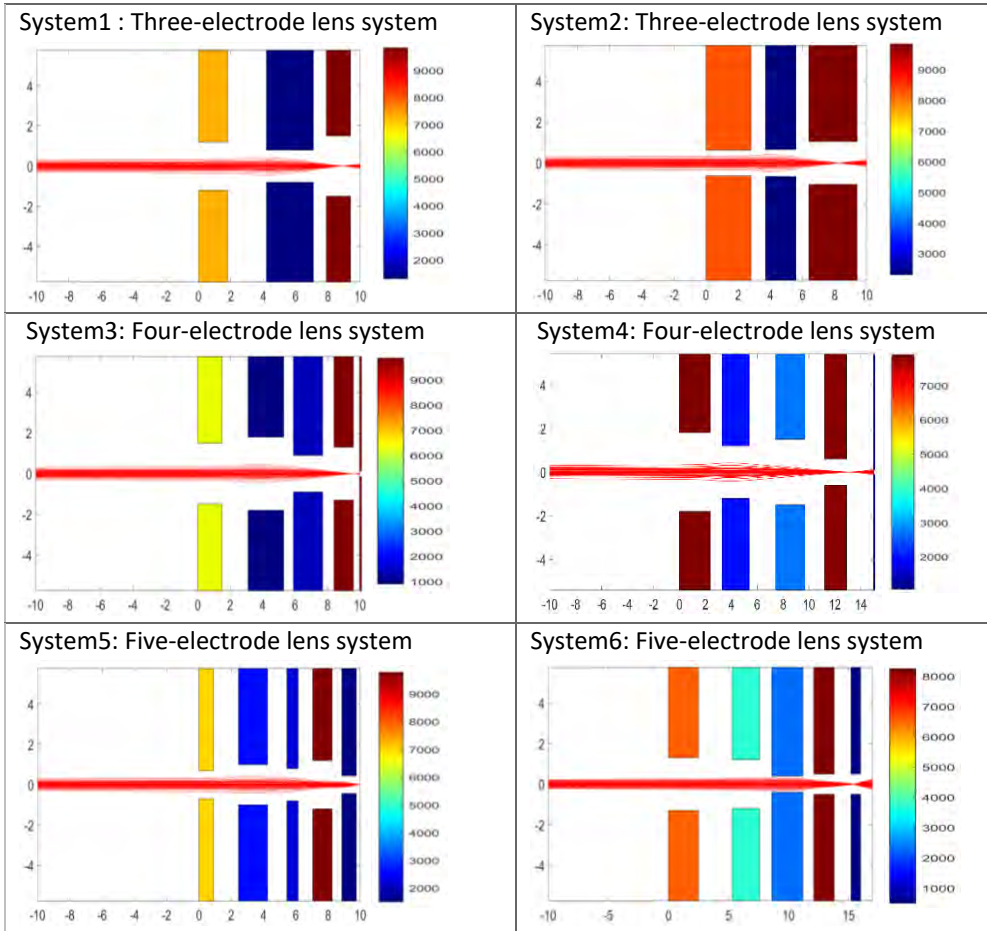


Figure 2.9. The six typical electrostatic lens systems are depicted in 2D. The colors in the table represent the voltages applied to each electrode. To facilitate visualization of the lens system, a schematic diagram showing the primary beam passing through the lens system (highlighted in red) is provided. The units along the axes of the graphs are in millimeters.

Precision in computing C_s and C_c plays a critical role in electron lens design. As these parameters are functions of the potential and its derivatives, they offer a valuable means to qualitatively and quantitatively assess the accuracy of axial potential calculations. Hence, these optical parameters are also extracted from the computed axial potential and its derivatives using SOEM and FEM methods, followed by a comparative analysis. The related graphs are presented in Figure 2.12. The quantitative data associated to these optical parameters are showcased and subjected to comparison within Tables 2.1 and 2.2.

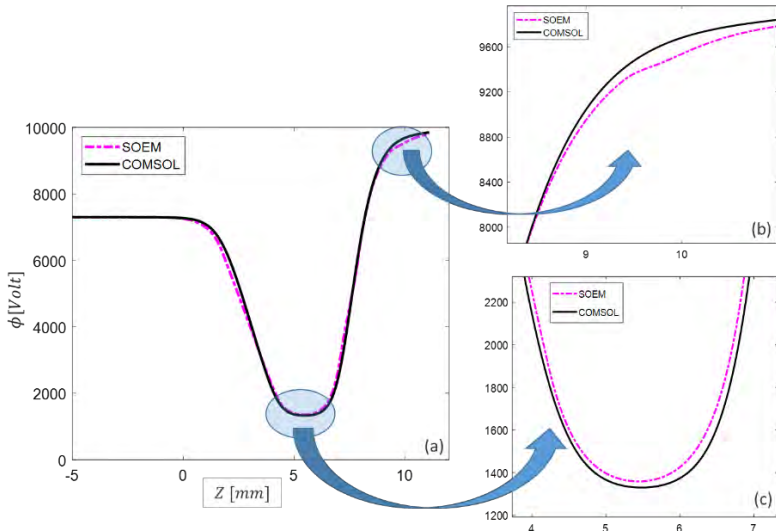


Figure 2.10. a). The overlapped graphs of axial potential for System 1 from Figure 2.9, calculated with different methods of SOEM and FEM. b) and c) the enlarged sections of the graphs.

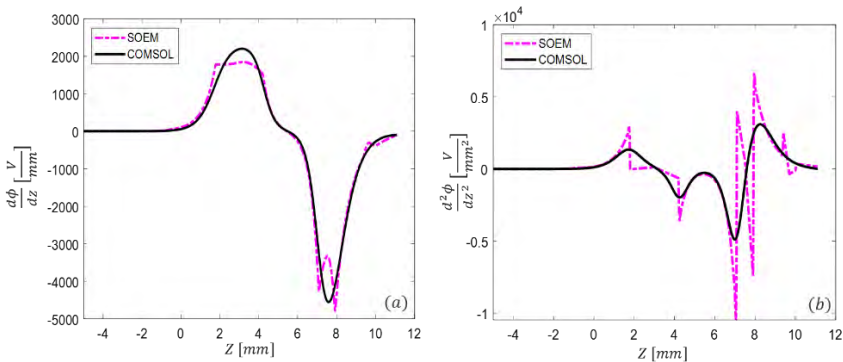


Figure 2.11. The overlapped graphs of first a) and second b) derivative of the axial potential for System1 from Figure 2.9, calculated with two different methods of SOEM and FEM.

In Figure 2.10a, a graphical comparison is made between the axial potential computed using two different methods of SOEM and FEM. This comparison reveals a generally close alignment between the axial potential graphs. Nevertheless, looking at the zoomed plots in Figures 2.10b and 2.10c, subtle deviations become evident. These deviations become more recognizable when observing the graphs of ϕ' (Figure 2.11a) and ϕ'' (Figure 2.11b). In the Cs and Cc graphs, as depicted in Figure 2.12, a pattern similar to that of the axial potential derivatives emerges, with a notable degree of alignment, yet minor deviations arise, particularly at the peaks of these graphs. This

deviation is more recognizable in C_s graphs. In our case study, this situation occurs because C_s exhibits a dependence on both the first and second derivatives of the axial potential, whereas C_c relies solely on the first derivative. The larger fluctuations in the second derivatives (as evident in Figure 2.11b compared to 2.11a) contribute to this observation.

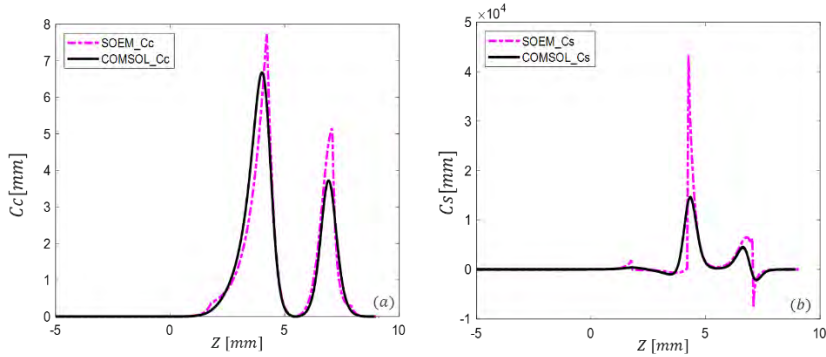


Figure 2.12. The overlapped graphs of chromatic (a) and spherical (b) aberration coefficients for System1 from Figure 2.9, calculated with different methods of SOEM and FEM.

In addition to the visual comparisons, for a comprehensive quantitative analysis, we have computed values for optical parameters including chromatic and spherical aberrations (C_c and C_s), as well as the image position (X_c), which is a crucial factor in lens design [7]. These calculations encompass all six lens systems outlined in Figure 2.9, and the results are presented in Tables 2.1 and 2.2. It's worth noting that the C_s and C_c data presented are related to the image side.

Table 2.1. The data related to the optical parameters such as chromatic and spherical aberrations (C_c and C_s) and the image position (X_c), calculated by FEM and SOEM, for six different typical electrostatic lens systems (presented in Figure 2.9).

Systems		System1	System2	System3	System4	System5	System6
X_c (mm)	FEM	8.92	8.26	9.65	13.39	10.03	15.43
	SOEM	9.12	8.62	9.87	13.78	10.35	15.65
C_c (mm)	FEM	18.71	15.26	23.73	26.13	1.82	0.98
	SOEM	19.45	16.80	24.10	18.78	2.03	0.77
C_s (mm)	FEM	21.13	54.70	22.88	217.65	7.14	12.48
	SOEM	30.77	104.58	37.07	392.57	13.19	11.77

A quick glance at Table 2.1 reveals that the optical parameters, when computed using the SOEM method, have values reasonably close to the accurate values obtained through FEM calculations. To assess the deviation of SOEM-derived data for each

system from the accurate data, we determine the error by considering COMSOL data (FEM) as the reference for accuracy. These errors are documented in Table 2.2 for each individual system. In Table 2.2, the final column provides the average error value across all six lens systems.

Table 2.2. The percentage of error in calculation of the optical parameters derived by SOEM, for six different typical electrostatic lens systems (presented in Figure 2.9) compared with the data calculated with the accurate method of FEM.

Systems	System1	System2	System3	System4	System5	System6	Error avg.
Error in Xc by SOEM	2.24 %	4.36 %	2.28 %	2.91 %	3.19 %	1.42 %	2.73 %
Error in Cc by SOEM	3.95%	10.09 %	1.56 %	28.13 %	11.53 %	21.43 %	12.79 %
Error in Cs by SOEM	45.62 %	91.18%	62.01 %	80.37 %	84.73 %	5.69 %	61.60 %

It is demonstrated in Table 2.2 that SOEM yields reasonably accurate values for Xc , exhibiting an average value of error of 2.73%, when compared to the accurate FEM values. The error percentage, on average, for Cc calculation by SOEM is 12.79% and 61.6% in average for the Cs calculation. Notably, the absolute error percentages for Cs are considerably higher compared to those for Cc and Xc calculations, owing to their different dependence on the potential and its derivatives. The computation times needed to evaluate the optical parameters for each lens system, using FEM and SOEM are ~ 60 sec, and 0.4 sec, respectively.

In summary, the quantitative data comparison shows, similar as the graphical comparison, that with SOEM, optical parameters which depend on the first derivative of the potential, such as Xc and Cc, can be calculated with reasonable accuracy but has large deviations for parameters related to the second derivative of the axial potential, such as Cs. However, due to the very fast calculation it is worthwhile to use it in an optimization routine as a first estimate to be complemented with more accurate FEM calculations to achieve shorter optimization times. This is the subject of the study in the coming sections.

Note that the accuracy and computational time of a field calculation method are also influenced by the respective meshing technique employed. For detailed insight into the meshing parameters utilized in COMSOL, please refer to Appendix B.

References

- [1] J. Adriaanse, H. van der Steen and J. Barth, J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena 7, 651 (1989).
- [2] H. van der Steen, J. Barth and J. Adriaanse, Nucl. Instrument. Methods Phys. Res. Sec. A: Accelerators, Spectrometers, Detectors and Associated Equipment 298, 377 (1990).
- [3] COMSOL Multi-physics, version 5.3a (2017), <https://www.comsol.com/>.
- [4] MATLAB, version r2016 b, <https://www.mathworks.com/products/matlab.html>.
- [5] V. K. Zworykin, G. A. Morton, E. G. Ramberg, J. Hillier, and A. W. Vance, Electron Optics and the Electron Microscope (John Wiley and Sons, New York, 1945).
- [6] D. W. O. Heddle, Electrostatic Lens Systems, Second Edition (IOP Publishing Ltd, Institute of Physics Publishing Bristol and Philadelphia, 2000).
- [7] M. Szilagy, Electron and Ion Optics (Plenum, New York, 1988).
- [8] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C. W. Hagen, and P. Kruit, Int. J. Mod. Phys. A 34, 1942020 (2019).
- [9] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, P. Kruit and C. W. Hagen, J. Vacuum Science & Technology B 39, 062605 (2021).

3. Optimization

3.1 Optimization in General

In science and engineering, an optimization problem refers to the problem of finding the best element among all existing elements, with regard to certain criteria. There might be constraints involved in the problem. To optimize function $f(X)$, subject to equality constraints $g(X)$ and inequality constraints $h(X)$, the optimization process can be mathematically formulated as:

$$\text{minimize } f(X) \quad S \rightarrow R, \text{ from set } S \text{ into set of real numbers of } R \quad (3.1)$$

In which, S can be a set of finite or infinite real numbers, X_i^m , $i = 1, 2, \dots, n$: for an n element X , where X is a vector in m -variable space

$$g_j(X) = 0, \quad j = 1, 2, \dots, p \quad (3.2)$$

$$h_k(X) \leq 0, \quad k = 1, 2, \dots, q \quad (3.3)$$

$$\text{if } p > 0 \text{ and } q > 0: \text{ constrained optimization problem} \quad (3.4)$$

$$\text{if } p = 0 \text{ and } q = 0: \text{ unconstrained optimization problem} \quad (3.5)$$

Minimization and maximization can be formulated as equations 3.6 and 3.7, respectively:

$$\begin{aligned} X_{opt} \in S, \text{ if } f(X_{opt}) \leq f(X) \text{ for all } X \in S \\ \rightarrow X_{opt}: \text{ minimum point of } f(X) \end{aligned} \quad (3.6)$$

$$\begin{aligned} X_{opt} \in S, \text{ if } t(X_{opt}) \geq t(X) \text{ for all } X \in S \\ \rightarrow X_{opt}: \text{ maximum point of } f(X) \end{aligned} \quad (3.7)$$

In the standard format, by convention, the optimization problem is often defined as the “minimization” problem as it is formulated in equation 3.6. However, the problem can be converted to a “maximization” problem in a straightforward way by negating $f(X)$. Equations 3.8 and 3.9 show converting maximization problem for a function of $t(X)$ into minimization problem of $f(X)$ by negating $t(X)$:

$$f(X) = -t(X), \quad t(X): S \rightarrow R \quad (3.8)$$

$$f(X_{opt}) \leq f(X) \leftrightarrow t(X_{opt}) \geq t(X) \text{ for all } X \in S \tag{3.9}$$

$\rightarrow X_{opt}$: minimum point for $f(X)$

The function $f(X)$ is called the objective function, merit function or cost function. The diagram of an optimization process (here afterwards minimization), can be schematically represented as Figure 3.1.

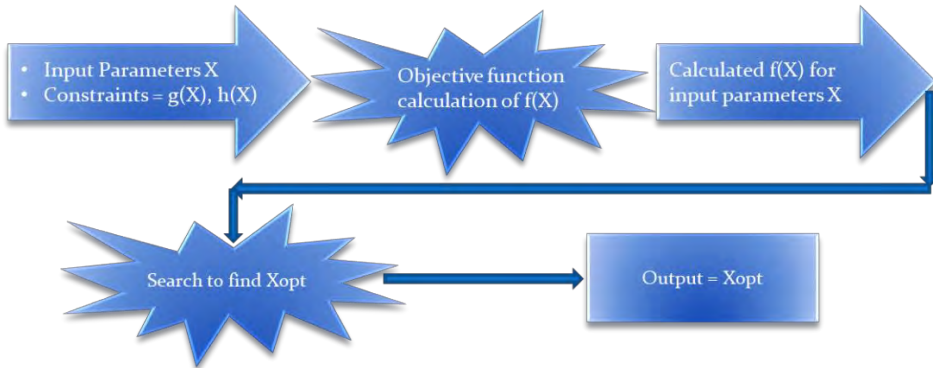


Figure 3.1. Diagram of optimization problem in general, where the process is to optimize a function $f(X)$ for input parameters of the elements X . The constraints are the equality and inequality constraints of $g(X)$ and $h(X)$, respectively. X_{opt} is the minimum point found through the optimization process as its output.

Optimization problems can be divided into different categories depending on different views and specifications of the problems. Figure 3.2 presents this categorization in general, schematically. Note that different categorized views and branches presented are not necessarily mutually exclusive. For instance, a static optimization problem can be either a constrained or an unconstrained problem. At the same time, the problem can also be a discrete or continuous optimization problem. These categorizations are explicitly explained in the following.

1. The “Static” optimization refers to the problem where the output is independent of time, while in “dynamic” optimization, the output varies with time. For instance, consider an optimization problem to find the best route to drive from a starting point to a destination. If the problem is considered from a distance point of view, it would be static and the solution can be found using a map. While, if the problem is meant to find the fastest route, then it is a dynamic one and the output depends on the time of day, accidents, weather, routes’ conditions and so on.

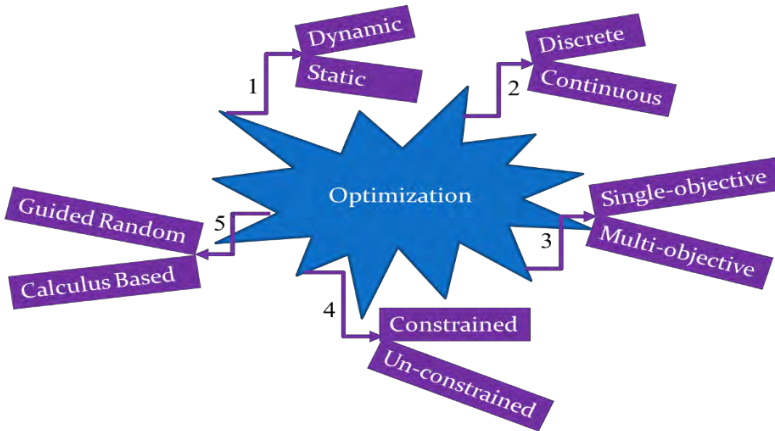


Figure 3.2. Schematic categorization of the optimization problems in general.

2. In “Discrete” optimization, some or all of element X belong to a discrete set and can only include finite numbers within possible range of values. While, in “continuous” optimization, the elements X can consist of an infinite number within a range of values. Expressing this mathematically, S in equation 3.1 would be a set of finite numbers for discrete optimization and a set of infinite real numbers for continuous optimization. Discrete optimization is sometimes also called combinatorial optimization, since the solutions of this problem involve different combinations of parameters selected from finite (or sometimes countable) sets of all possible parameters.

3. Where there is only one objective function defined to be optimized, the optimization is called a “single-objective optimization” problem (hereafter “SOP”). This optimization is also known as single-parameter optimization. However, in many optimization problems, there is more than one objective function defined to be optimized. These types of optimization are referred to as “multi-objective optimization” problems (hereafter “MOP”). Looking at equation 3.1, these optimizations can be mathematically expressed as:

$$\text{minimize } f_n(X) \ S \rightarrow R, \quad n = 1, 2, \dots, N, \quad N > 1 : \text{ MOP} \quad (3.10)$$

$$\text{minimize } f_n(X) \ S \rightarrow R, \quad n = 1 : \text{ SOP} \quad (3.11)$$

where N refers to total number of objective functions.

4. In “Constrained” optimization, there are some equality and/or inequality constraints (in equations 3.2-3.3 mentioned as $g(X)$ and $h(X)$) involved in the

optimization problem. In the problems where there exist no equality/inequality constraints, the optimization problem is an “Unconstrained” optimization problem.

5. Optimization techniques are divided into two main categories, regarding their search strategies. In the first group, namely “Calculus Based”, the search is started from an initial point, while approaching towards a better point is performed by calculation of the best direction. Finding the best direction in these search methods, is based on calculus and generally uses the derivatives of the objective function. The result is particularly dependent on the selected initial point. These techniques tend to be fast, however, easily become stuck in the local minima [1]. In the second group, “Guided Random”, the search can start either from defined or randomly-created initial point/points. The direction of the movement towards the better points is found based on probabilistic calculations. These techniques are shown to be slower compared to the first group, but have greater success in reaching the global minimum [1].

The last abovementioned category (i.e. 5th category in Figure 3.2), is assumed as the main categorization. In principle, the optimizations are named according to the subdivisions of this category. A schematic of the search-techniques classification is represented in Figure 3.3. As shown, the search techniques are divided into two main groups of calculus-based and guided-random search techniques.

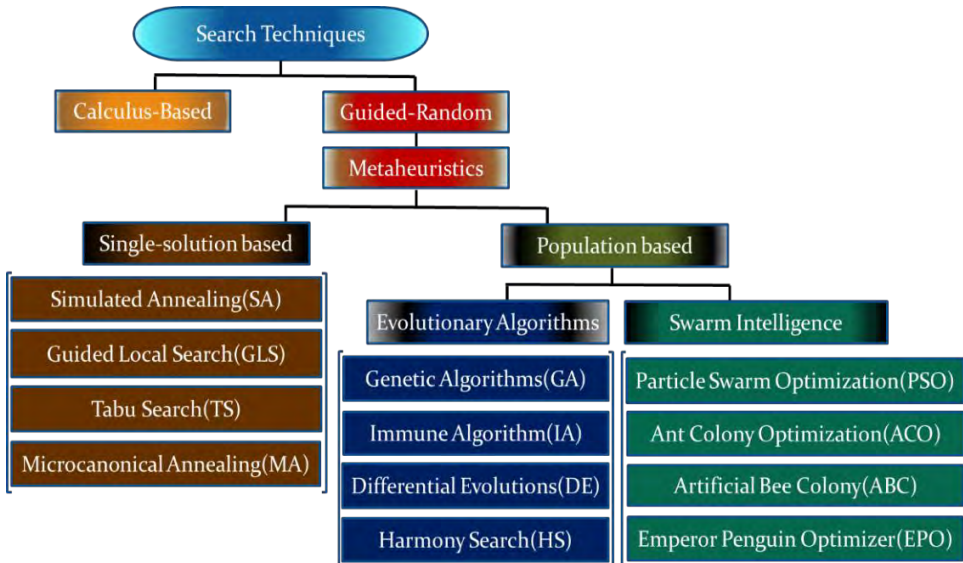


Figure 3.3. Schematic of “Search Techniques” classification.

A metaheuristic is an iterative search approach which efficiently explores and exploits the search space, using intelligent concepts inspired by nature, to reach near-optimal solutions. The metaheuristic is a guided-random search (Figure. 3.3), which is proven to work efficiently in solving real-life complex problems from different fields of politics, economics, science, engineering and management [2, 3]. The metaheuristics have the following characteristics [4]:

- Easy to integrate with existing problems.
- Do not require the gradient information of the objective function.
- Possible to implement in a broad range of problems with different subjects.
- The algorithms are approximate and usually non-deterministic.

Moreover, as previously mentioned, for problems where there are multiple local minima, metaheuristics generally outperform the calculus-based techniques, since they tend to escape better from the local minima. In our problem, it is expected that there are multiple local minima, because as we will see in Chapter 5, the objective function is a highly non-linear and complex function with respect to its parameters. Therefore, our main focus is on metaheuristic search techniques.

The majority of metaheuristic algorithms are mimicking physics law, swarm behavior of organisms to search food, and evolution processes in nature [5]. These algorithms can be divided into two main groups of “single-solution based” and “population-based” algorithms (Figure 3.3) regarding their technique in finding better solutions [6]. A single-solution based algorithm utilizes only one solution and tries to improve this solution by a local search. Due to this search characteristic, it may become stuck in a local minima [2, 7]. The most well-known single-solution based metaheuristic algorithms are Simulated Annealing (SA) [8], Guided Local Search (GLS) [9], Tabu Search (TS) [10], and Micro-canonical Annealing (MA) [11].

The population-based algorithms consider multiple solutions and try to improve all of them simultaneously by their search procedures. Therefore, these algorithms have a higher chance of exploring a larger area of objective function landscape, hence, they have higher diversity and less chance of becoming trapped in a local minima. The population-based algorithms can themselves be subdivided into two different groups, namely “Evolutionary algorithms” and “Swarm Intelligence”, according to their specific search technique, either taken from the evolutionary process or imitating a swarm behavior, respectively (Figure 3.3).

The most commonly known algorithms of the first group (i.e. evolutionary algorithms) are Genetic Algorithm (GA) [12], Immune Algorithm (IA) [13], Differential Evolutions (DE) [14], and Harmony Search (HS) [15]. For the second group (i.e. swarm intelligence) several well-known algorithms are Particle Swarm Optimization (PSO) [16], Ant Colony Optimization (ACO) [17], Artificial Bee Colony (ABC) [18], and Emperor Penguin Optimizer (EPO) [19].

In the following section, some of the most known optimization algorithms, such as GA, SA, PSO and ABC as a representative of each group, are introduced in detail and thereafter implemented in our case-study. A “Calculus Based” optimization, is also then presented and implemented in our case-study, as a local optimization technique. Implementations of these algorithms for our case-studies and their related flowcharts, followed by the defined case-study problems, are presented at the end of this section.

3.2 Genetic Algorithm (GA)

In the early 1950s, the idea of evolution-based and genetics computation procedures came to the attention of scientists. There were many contemporary scientists around the world, such as Friedman, Tooms, Baricelli, Bledsoe, Box, Bremermann, Reed and Holland who were working independently on this subject [20]. However, John Holland was the first to really popularize this idea and achieved credit as the inventor of genetic algorithms. Holland’s idea became publicly known through one of his students, David Goldberg, who had solved a complex problem of controlling gas-pipeline transmission in his dissertation [21]. Holland was also the first who developed ways to apply evolutionary phenomena, such as adaptation, to computer applications [22, 23]. His first genetic algorithms were direct application taken from his observation in natural evolution [23]. In this section, the framework of the genetic algorithm and its operators involved, are explained in detail.

3.2.1 Framework of Genetic Algorithm

The GA mimics natural evolution, based on Darwin’s theory [24]. The algorithm starts with randomly created elements, called initial population, represented by $P_1(X_i^m)$ ($i = 1, 2, \dots, Npop$). Similar to the general cases of optimization mentioned in section 3.1, X^m represents an element with a vector in m -variable space. A so-called parameter of “population”, defines the number of elements in each generation, denoted here by $Npop$. In nature, the elements are the chromosomes of different organisms. In electron lens design, the set of electron lens systems from which the lens design variables are determined, represents the elements.

The GA uses a variety of genetic operators such as Crossover, Mutation and Elitism, to gradually improve a set of elements $P_j(X_1^m, \dots, X_{Npop}^m)$, in a so-called “generation” of j , towards the next generation of $j + 1$, having a better set of elements $P_{j+1}(X_1^m, \dots, X_{Npop}^m)$, regarding their objective function values. The algorithm proceeds until it satisfies the stopping criteria. Different conditions can be set as the stopping criteria, such as a maximum computational run time, a specified value of objective function, or a maximum number of generations. In our case-study, reaching a maximum number of generations, denoted here by G_{max} , is appointed to end the process. A simplified algorithm for a classical GA is presented below:

Table 3.1 : **Classical Genetic Algorithm**

Input
<i>Population size, Npop Maximum number of generations, G_{max}</i>
Output
<i>The global optimum solution, $X_{globopt}^m$</i>

start
<i>Generate Npop number of elements $X_i^m (i = 1, 2, \dots, Npop)$, as the initial population, for the first generation.</i>
<i>Set counter of generations $j = 1$</i>
While ($j < G_{max}$)
<i>Select the Npop better elements for the reproduction</i>
<i>Apply the genetic operators such as Crossover, Mutation, and Elitism on the selected elements to produce the new Npop elements</i>
<i>Replace the old population with the new population</i>
<i>Increment generation counter by 1</i>
end while
<i>represent the global optimum, $X_{globopt}$</i>
end

The flow diagram of this algorithm is presented in Figure 3.4.

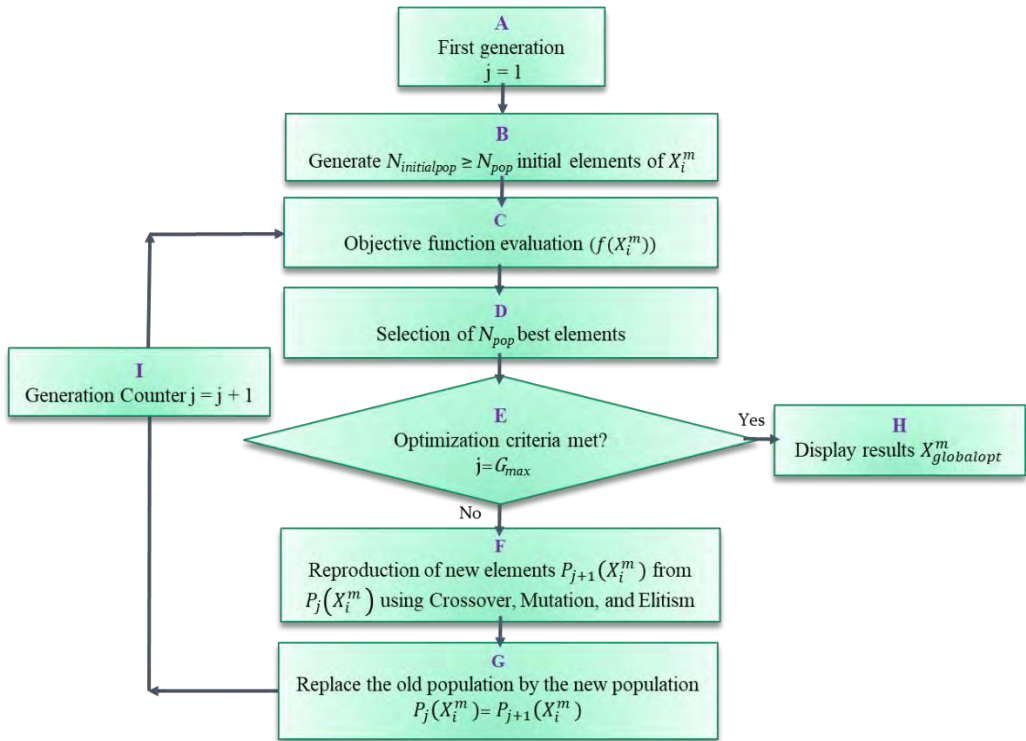


Figure 3.4. The flow diagram of a classical Genetic Algorithm.

3.2.2 GA operators

There are a variety of genetic operators involved in the GA process, such as encoding, selection, crossover, mutation and elitism. These operators can be performed via different schemas. In Figure 3.5, the most well-known schemas for each operator are represented. A brief explanation on how these operators work is given in this subsection. Taking the proper schema for each operator depends on the problem, and can influence the performance of the GA. The study of the tuning of these parameters, with more detailed explanation on the most proper schemas in our case-study, is fully discussed in section 3.4.

3.2.2.1 Encoding

To enable the GA to initialize processing the information, the given data of each element has to be encoded in a bit-string format [24, 25]. The encoding can be performed differently according to the type of problem.

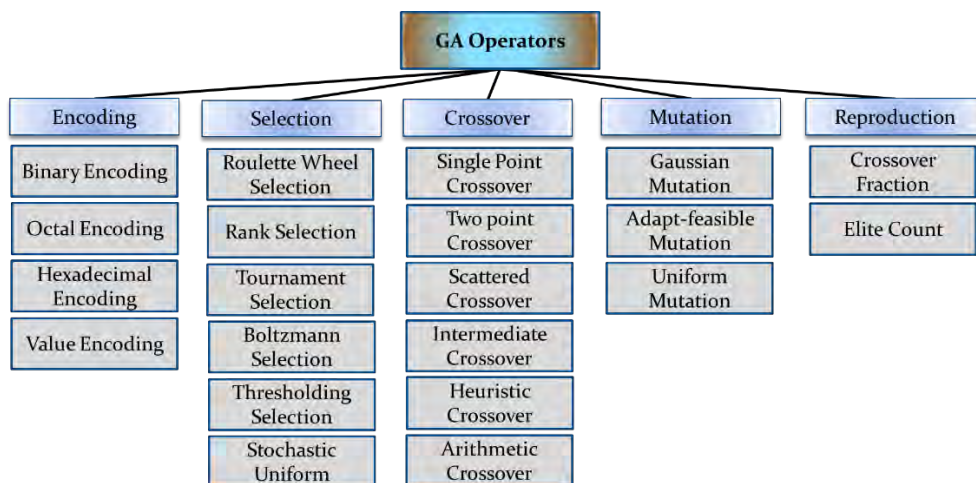


Figure 3.5. The Genetic Algorithm operators and their related schemas.

The most common encoding options, as depicted in figure 3.5, are “Binary”, “Octal”, “Hexadecimal” and “Value-based” encoding. In “Binary” encoding, each element is represented as bit strings including 1 or 0 [26]. This encoding method is the one most generally used, since the implementation of mutation and crossover on the elements, having a format of strings of 0 and 1, are fast and easy.

Another encoding option is “Octal” encoding. In this method, similar to “Binary” encoding, the elements are converted to strings, but include the numbers between 0 and 7. In “Hexadecimal” encoding, the method is still the same, while the strings can include the numbers between 0 and 9 and A to F.[27, 28].

The other option is value encoding. In this schema, the element is also converted to strings. However, the strings can involve the values from the real integer numbers, or characters [29]. This encoding technique is mainly useful when the elements have complicated values. Where the value includes real integer numbers, the encoding technique is called “Real-coded GA” (RGA). RGA is used in a variety of real-life optimization problems. Its main advantage over the Binary-coded GA (BGA) is its robustness, efficiency and accuracy. However, it suffers from premature convergence [31]. BGA may achieve better results regarding the convergence, however, it has difficulty on reaching the specific desired precision [30].

3.2.2.2 Selection

Selection is the step used to determine which elements will participate in the next generation of the reproduction process [31]. The most commonly used schemas for

selection are “Roulette Wheel”, “Rank”, “Tournament”, “Boltzmann”, “Thresholding” and “Stochastic Uniform”.

“Roulette wheel” selection is based on the working principle of a roulette wheel. The sections of the wheel are allocated to different elements, proportional to their objective function value. The wheel is then rotated randomly to select specific elements. The wheel stops at a point where its value has been assigned randomly by the algorithm. This method highly relies on the stochastics. It can hence cause a premature convergence towards a local minimum [31].

To modify this selection schema, De Jong and Brindle, introduced the “Rank” selection, based on the concept of determinism in selection. “Rank” selection, similar to the “Roulette wheel”, works based on a rotating wheel, but now the rankings of the elements are used instead of their objective functions in allocating the sections in the wheel to the elements, so that ranking of the elements are assigned to them according to their objective function values. Therefore, the elements each have a chance according to their rankings. This selection schema has a lower chance of facing premature convergence compared to the “Roulette Wheel” selection method [31].

3.2.2.3 Crossover

Crossover is an operator of GA, which is used to create new elements (in GA context, called offspring), by combining the old elements (in GA context, called parents). The crossover performs through different schemas. The most well-known among them are, “single-point crossover”, “two-point crossover”, “intermediate crossover”, “scattered crossover”, “heuristic crossover” and “arithmetic crossover”.

In a “single-point Crossover” [32], first, a random integer number A , between 1 and the number of variables (m), is generated by GA. The vector entries from the first parent numbered below or equal to N_c , will be swapped with the same vector entries from the second parent. The tails from both parents will also be swapped similarly. This is schematically shown in figure 3.6 for two elements, assuming $A=2$. Note: parent 1 and parent 2 are represented as P_1 and P_2 . Offspring 1 and offspring 2 are illustrated by O_1 and O_2 .

In “two-point crossover” [32], two random integer numbers A and B , between 1 and the number of variables (m), are created by GA. The vector entries from the two parents which are numbered between A and B , will be exchanged with each other to create the new elements. This process is schematically illustrated for two elements, assuming $A=2$ and $B=5$, in figure 3.6.

In “scattered crossover” [33], a binary vector, with the vector entries’ number equal to the vector entries’ number of parents, is randomly created by the GA. The vector entries from the first parent with values of 1 will be replaced by the corresponding vector entries from the second parent. An example is presented in Figure 3.6, assuming the random vector is [0 1 1 0 0 0 0].

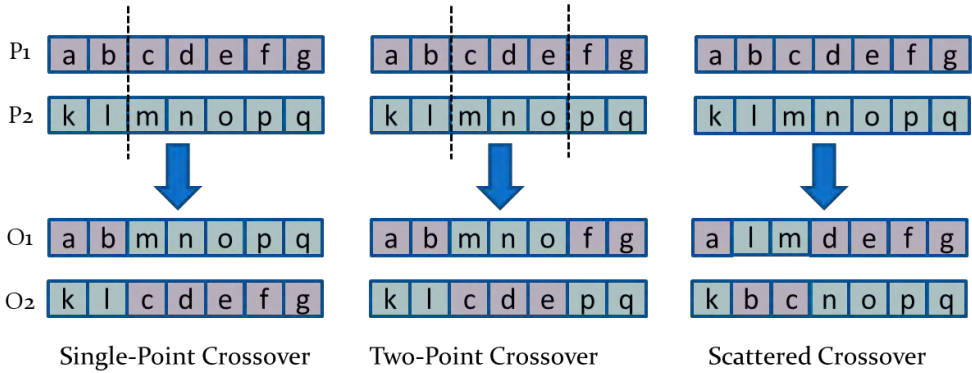


Figure 3.6. A schematic of different crossover schemas.

In the “Intermediate Crossover” [33], a single parameter (called “Ratio”), is initially created by GA. This “Ratio” can be either a scalar or a row vector of length equal to the number of variables (m). The GA then creates offspring using a weighted average of the parents, based on this “Ratio”, and the following equation:

$$O_1 = P_1 + rand \cdot Ratio(P_2 - P_1) \tag{3.12}$$

Where, as noted before, parent 1 and parent 2 are represented as P_1 and P_2 . Offspring 1 and offspring 2 are illustrated by O_1 and O_2 .

In the “Heuristic” crossover [34], GA first indicates the better and worse parents by their objective function values. The value for the parameter of “Ratio” is specified by the user. The crossover operator then creates the offspring using the “Ratio” and based on the following formula (assuming P_1 is better than P_2):

$$O_1 = P_2 + Ratio(P_1 - P_2) \tag{3.13}$$

The generated offspring therefore will achieve a value in the interval between the two parents’ values. The distance of the offspring from the best and the worst parent is determined by the parameter “Ratio”.

In the “Arithmetic” crossover [35], the operator creates the offspring by taking the average of the two parents. The created offspring therefore have always a value in the interval between the upper and lower pre-determined bounds.

3.2.2.4 Mutation

The mutation operator is used to enhance the diversity on the elements space by changing the vector entries of the elements in a random manner. This way, the GA has a higher chance to search the full space of possible solutions and to achieve the global optimum. There are different schemas through which the mutation can be performed. Some prevalent schemas, “Uniform”, “Gaussian” and “Adaptive-feasible” mutation are explained below.

The “Uniform” mutation[36], performs via a two-step process. In the first step, part of the vector entries of each element is selected to be mutated. In the second step, the selected vector entries of the elements, would be replaced by randomly created numbers with the probability determined by a so-called “Rate” factor. The “Rate” factor is a number in the range of [0 1], which is pre-defined by the user.

In the Gaussian mutation[33], the GA adds a randomly created number, which is selected from a Gaussian distribution, into the vector entries of each element. The Gaussian distribution is defined by two factors of “scale” and “shrink”. The Gaussian distribution is centered at zero, while its width is determined by the “scale” parameter, which diminishes to the smaller values by the factor of “shrink” in the course of generations. In other words, the “scale” is used to specify the amount of deviation at the first generation and the “shrink” determines how the initially introduced deviation will be diminished as the generations proceed.

In the adaptive-feasible mutation[37], the GA applies the mutation on each element with a probability proportional to its objective function value. A better element will therefore be less prone to mutation and have a lower probability of being discarded. hence, the direction of mutation is called to be in the direction of the worst elements.

3.2.2.5 Reproduction (Crossover Factor , Elitism)

The “Elitism” is the operator in the reproduction process, used to prevent the loss of the best elements as the generations proceed. The “Elite number” is a number assigned by the user, which determines how many of the best elements are to be kept unchanged. In each generation, after the objective function evaluations, the “Elitism” does not permit the vector entries of the best specified elements to encounter any

changes. In this way, the best elements are always transferred to the next generations and are not lost.

In the reproduction process, there is another main factor, “Crossover Factor”, which controls the reproduction implementation. This parameter is assigned by the user and determines how many of the elements in each generation are encountered by the “Crossover” and how many will go through the mutation process.

3.2.2.6 Stopping Criteria

The algorithm stops when the defined stopping criteria is reached. The stopping criteria can be defined by the user, in different ways:

“Function Tolerance”: The algorithm stops when the difference between two sequential objective function values becomes smaller than a default value.

“Maximum Iterations” The algorithm continues until the number of iterations reaches the defined maximum number of iterations.

“Maximum Function Evaluations”: The algorithm stops when the number of function evaluations reaches a specified number.

“Maximum Time”: The algorithm continues until the execution time reaches a defined value for the maximum execution time.

“Objective Limit”: The algorithm runs until the value of the objective function reaches a value less than a defined value for the objective function.

3.3 Particle Swarm Optimization (PSO)

3.3.1 Introduction

Particle swarm optimization (PSO) is a population-solution based, meta-heuristic global optimization method [38]. This technique is based on the algorithm initially proposed by Kennedy and Eberhart in 1995 [39], with modifications presented later by Mezura-Montes and Pedersen [40, 41].

3.3.2 Modeling and Algorithm

The algorithm starts by generating the initial particles. To each particle an initial position and velocity are assigned. The objective function will be evaluated at each

particle position. The best objective function value and the best position will be determined accordingly.

To find the optimum point, each particle selects a new velocity and position, based on its current velocity, its best found positions, and the best found positions of its neighboring particles. The velocity and position of the particles will be updated in each iteration. A simplified algorithm of this approach is presented in Table 3.2 [42].

Where, p_i represents particles position in PSO, which is similar to X^m_i in the general context of optimization. In which i represents the particle index, $iter$ shows the current iteration number, $N_{totparticle}$ is the total number of particles, p the position of the particle, V is velocity of the particle and f stands for the objective function. In each iteration, p_{best} (the best element in a specific iteration), g_{best} (the best element among all iterations counted up to a specific iteration), V and p will be updated using the equations 3.14-3.17.

$$p_{best}(i, iter) = \arg \min [f(p_i(j))] \quad (3.14)$$

$$i = 1, 2, \dots, N_{totparticle}, j = 1, \dots, iter$$

$$g_{best}(iter) = \arg \min [f(p_i(j))] \quad (3.15)$$

$$i = 1, \dots, N_{totparticle}, j = 1, \dots, iter$$

$$V_i(iter + 1) = V_i \cdot V_i(iter) + a_1 \cdot r_1 (p_{best}(i, iter) - p_i(iter)) \quad (3.16)$$

$$+ a_2 \cdot r_2 (g_{best}(iter) - p_i(iter))$$

$$p_i(iter + 1) = p_i(iter) + V_i(iter + 1) \quad (3.17)$$

The algorithm will be ended when the defined stopping criteria is reached. The stopping criteria can be determined by the user using different schemas. These schemas are defined similarly to other optimization algorithms, and have been explained previously in section 3.2.2.6.

Table 3.2: **Standard Particle Swarm Optimization**

Input

Total number of particles, $N_{totparticle}$

Maximum number of iterations, I_{max}

Output

The global optimum solution, $p_{globopt}$

start

Generate randomly $N_{totparticle}$ number of particles $p_i(0)$, ($i = 1, \dots, N_{totparticle}$), as the initial particles, for the first iteration

Set the initial values of $p_{best}(i, 0)$ equal to the value of their corresponding initial positions: $p_{best}(i, 0) = p_i(0)$

Set $g_{best}(0)$ equal to the minimum value of the whole particles' (swarm) initial positions: $g_{best}(1) = \arg \min [f(p_i(0))], i = \{1, 2, \dots, N_{totparticle}\}$

Set counter of iteration: $iter = 1$

Generate randomly $N_{totparticle}$ number of velocities as initial velocities $V_i(0)$ corresponding to each particle.

While ($iter < iter_{max}$)

Calculate the objective function for all p_i

Find the p_{best} related to each p_i and g_{best} regarding all p_i , using equations 3.14 and 3.15

Pick random numbers a_1 and a_2 in the interval of $[0, 1]$

Update particles velocity and position using equation 3.16 and 3.17

If $f(p_i(iter)) < f(p_{best}(i, iter))$

Update the best position of particle i : $p_i(iter) = p_{best}(i, iter)$

If $f(p_i(iter)) < f(g_{best}(i, iter))$

Update the swarm's best position : $p_i(iter) = g_{best}(i, iter)$

$iter = iter + 1$

end while

represent the global optimum $p_{globopt}$

end

The flowchart of this algorithm is shown schematically in Figure 3.7.

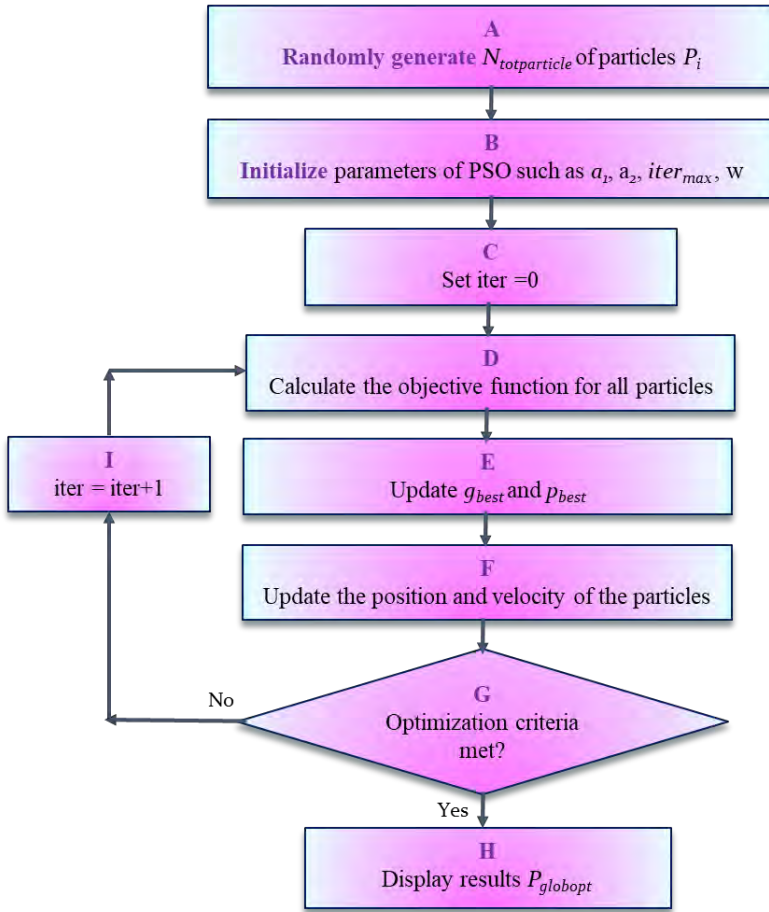


Figure 3.7. The flow diagram of a standard Particle Swarm algorithm.

3.4 Simulated Annealing (SA)

3.4.1 Introduction

Simulated annealing (SA) is a single-solution based, meta-heuristic algorithm implemented for unconstrained/bound-constrained global optimization problems [43]. It was initially presented, independently, by the two contemporary researchers Kirkpatrick et al. [44] in 1983, and by Cerny [45] in 1985.

The algorithm simulates the physical process of annealing in the material. Annealing means to heat up a substance to a specified level above its melting temperature, and then to lower the temperature down gradually, in a specific manner, to reach the

minimum energy level of the system and make the material crystalize. This crystalized substance, having all its lattice atoms perfectly aligned, is an example of nature finding a beautiful optimum structure of a substance. However, if the cooling process is performed too quickly, the crystalized state will never be reached and the substance becomes an amorphous solid not a crystal. The key to achieving the minimum energy level and the crystalized state, is to carefully control the rate of temperature decrement. The annealing process should be performed under specific conditions, otherwise the material will not reach its minimum energy level and will not crystalize. The format of the annealing function, for the first time derived by Geman and Geman in 1984, is shown to be inversely proportional to a logarithmic function of time [46].

The method has an easy implementation into algorithm. Therefore, it is one of the most common global optimization techniques used to solve a variety of practical optimization problems such as, traveling salesman [45, 47], communications systems [48], timetabling [49, 50], job shop scheduling [51], and continuous optimization [52].

The probability that SA reaches the global optimum point becomes higher when the annealing process is extended to a longer time [52]. Hence, it might end up in a local minimum if the annealing time is not enough to reach a global minimum, or SA takes too long to reach the global point.

To avoid being trapped in local minima and/or to speed up the convergence speed, various techniques are presented by researchers, including adaptive simulated annealing (ASA) [53], faster annealing schedules [54], or hybridized SA with other meta-heuristic optimization techniques, such as genetic algorithm [55], particle swarm optimization [56], or artificial bee colony [57], and so on. In this section, the theory and modeling for a classical SA is presented. A hybridized SA with GA (called hereafter HSAGA), with PSO (called HSAPSO) and with ABC (called HSAABC), are also implemented in our work. The results taken from HSAGA, HSAGA and HSAABC, together with the comparison of results solely from GA, SA, PSO and ABC, are represented in chapter 5.

3.4.2 Modeling and Algorithm

The SA algorithm, mimicking physical annealing in nature, can be formulated as the following steps:

The algorithm initializes a first guess (randomly created) state X^m which, similar to the general optimization case, stated in section 3.1, X^m represents an element with a

vector in m -variable space. In addition, a first initial temperature T_0^m should also be defined, in correspondence with elements X^m as the initial temperature of component m with which to start annealing.

At each iteration, a new element is randomly generated by SA. Generating this new element is based on a probability distribution which is related to the temperature. Assuming the element as a point in the space, the step length and direction from the current point to the next point can be determined using different schemas. Some more common annealing techniques are mentioned below:

- “Annealing fast”: In this annealing schema, the step length is equal to the current temperature. The direction is selected in a uniformly random format.
- “Annealing Boltzmann”: The step length is equal to the square root of the temperature. The direction is selected in a uniformly random format.

In the next step, the algorithm accepts the new elements with lower objective function and replaces the old one with the new element. However, with a certain probability, it also accepts the element having a higher objective function value. The reason why it also accepts the worst elements is to provide a wider solution space to be explored by the SA, and hence, to lower the chance of being trapped in a local minima in the early iterations.

The probability at which the acceptance of the worst elements occur can also be defined differently. One common schema to define this probability is:

$$P = \frac{1}{1 + \exp\left(\frac{f(X_i) - f(X_{i-1})}{T_i}\right)^2} \quad (3.18)$$

Where T_i is the current temperature and $f(X_i)$ is the objective function value for the element X_i . The value of the objective function in SA, is analogous to the energy level of the substance in nature, in which the “heating” of the substance occurs by modifying the free variables of the problems X_i .

Since in equation 3.18, $f(X_i) > f(X_{i-1})$ and $T_i > 0$, the probability P is a value between 0 and 1/2, where smaller T_i , leads to smaller acceptance probability. Also, having a larger difference of $f(X_i)$ and $f(X_{i-1})$, leads to a smaller chance of acceptance.

The SA decreases the temperature, while storing the best element found. The temperature T_i , hence also needs to be updated. The update functionality can form via different schemas:

- Temperature exponential :

$$T_i^m = T_0^m \cdot (0.95)^{k_i^m} \quad (3.19)$$

- Temperature fast:

$$T_i^m = T_0^m / k_i^m \quad (3.20)$$

- Temperature Boltzmann:

$$T_i^m = T_0^m / \log(k_i^m) \quad (3.21)$$

Where, k_i^m is the annealing parameter, equals the iteration number for component m , until reannealing.

The algorithm will stop when the defined stopping criteria is reached. The stopping criteria are explained in section 3.2.2.6. and can be applied similarly to this algorithm. The SA algorithm is illustrated schematically in Figure 3.8.

3.5 Calculus-Based Optimization (CBO)

Calculus-based optimization (CBO) is an optimization technique which uses the gradient (derivatives) of the objective function. A common type of this optimization is the method of gradient descent (also called steepest descent). This method is implemented on the objective function $f(X)$, starting from an initial point of X_1 , to progressively search in the neighborhood of this point to reach the local optimum point. Note that in this method, the function $f(X)$ should be differentiable in all neighboring points which are progressively taken under search. The algorithm starts from an initial point of X_0 , taking the steps of δ_N , moving towards the direction of the negative gradient of the objective function to progressively reach to the local minimum of the function. If it is a maximization problem, the direction will be in the direction of the positive gradient of the function to reach the maximum point. This process is formulated mathematically below, for a minimization problem:

$$X_{N+1} = X_N - \delta_N \cdot f'(X_N), N \geq 0 \quad (3.22)$$

$$f(X_0) \geq f(X_1) \geq f(X_2) \geq \dots f(X_N) \quad (3.23)$$

In which, X_N is a local minimum of the objective function f , which is reached after N sequential iterations, when the search started from X_0 .

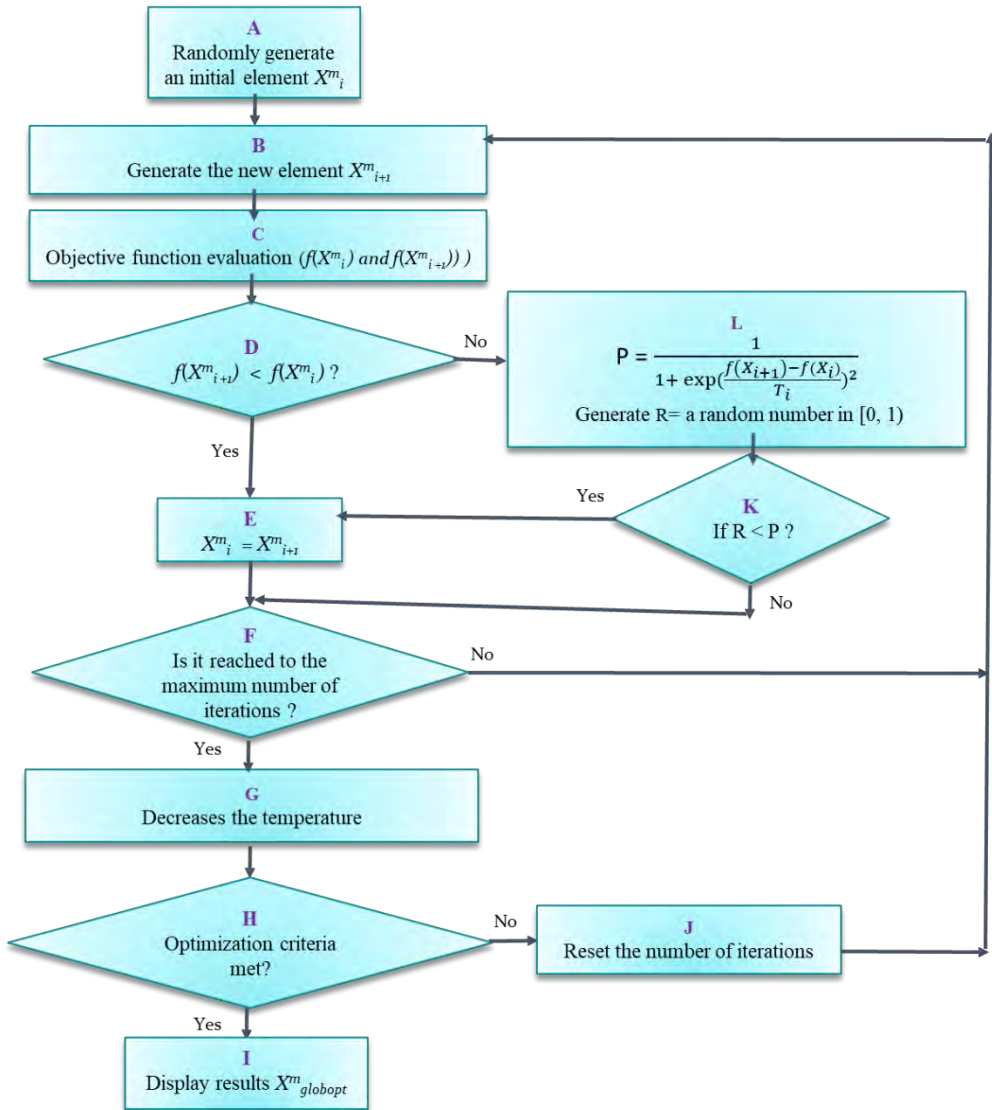


Figure 3.8. The flow diagram of a Simulated Annealing algorithm.

This process is schematically shown in Figure 3.9. As it is illustrated, by applying this technique, starting from two different initial points ($X1_0$ and $X2_0$), it will end at two different local minimum points ($X1_N$ and $X2_N$).

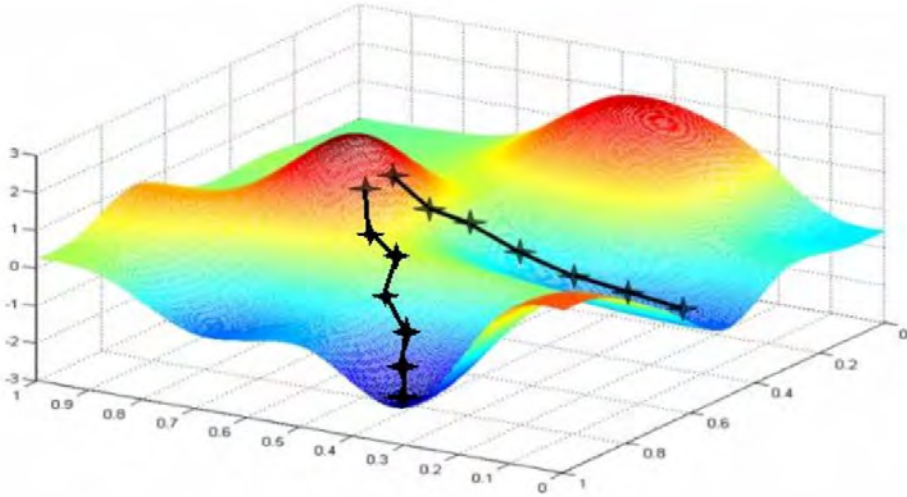


Figure 3.9. Schematic illustration of a steepest descent (gradient descent) method. It is applied on a 3-D objective function, starting from two different initial points and ends at two different minima [58].

3.6 Defining our case-study optimization problem

The optimization problem, as explained in section 3.1, is defined mainly by two factors; objective functions ($f(X)$ in Eq. 3.1), and the free parameters of optimization in m variable space (X^m in Eq. 3.2). In electron-lens-system design, there might be different aims for the optimization depending on the application. Here, two main optimization problems which exist in multi-electrode lens systems design, are considered.

In the first application (hereafter called “Application1”), the aim is to acquire the smallest spot size at the image side (D_{spot} in Eq. 5.1). The spot size is a function influenced by the geometry of the lens and the voltages of the electrodes. The optimization for this case-study, is therefore to find the minimum value of the spot size by tuning the voltages of the electrodes and geometry of the lens.

The other application (hereafter called “Application2”), aims to acquire the least spot size, while getting the highest detection efficiency of the secondary electrons (explained in detail in section 5.3).

The optimization problem can then be formulated as:

$$f_N(X) = D_{spot}, \quad n = 1 \quad : \textit{Application1} \quad (3.24)$$

$$f_N(X) = [D_{spot}, SE_{DE}], \quad n = 1, 2 \quad : \textit{Application2} \quad (3.25)$$

$$X^m = [T_i \ G_i \ R_i \ V_i], \quad i = 1, \dots, N_{el} \quad (3.26)$$

Where, N_{el} stands for the number of electrodes. T_i , G_i , R_i and V_i are denoted for thicknesses, gaps, radii and voltages correspondent to each electrode, respectively.

Some constraints/bounds may also exist in these problems. The constraints can be brought into the objective function part. This issue will be discussed in greater detail later for each case-study regarding the situation and analysis-case of the problems. The discreteness and continuity of the parameter of the optimization also differ depending on the case-study and situation of the defined problem. This subject will also be explained more specifically when each case-study is defined.

References

- [1] Practical Genetic Algorithms, R. L. Haupt, S.E. Haupt. John Wiley & Sons, Inc. (1998).
- [2] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, *Multimedia Tools and Applications* 80, pg. 8091, (2021).
- [3] V. Kumar, J.K. Chhabra, D. Kumar, Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *J. Comput. Sci.* 5(2), pg. 144, (2014).
- [4] I. El-Henawy, N.A. Abdelmegeed, *Meta-Heuristics Algorithms: A Survey*, *International Journal of Computer Applications* (0975 – 8887), Volume 179, No.22, February (2018).
- [5] E. Bonabeau, M. Dorigo, G. Theraulaz, (1999) *Swarm intelligence: from natural to artificial systems*, (Oxford University Press, Inc. 1999).
- [6] M. Abd Elaziz, et al., *Advanced metaheuristic optimization techniques in applications of deep neural networks: a review*, *Neural Computing and Applications*, Springer, (2021).
- [7] V. Kumar, D. Kumar, An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems. *Adv. Eng. Softw.* 112, pg. 231, (2017).
- [8] P. J. M. van. Laarhoven, *Simulated annealing : theory and applications*, E. H. L. Aarts, Dordrecht: D. Reidel. 1987. (ISBN 90-277-2513-6. OCLC 15548651)
- [9] P. H. Mills, *Extensions To Guided Local Search*, A thesis submitted for the degree of Ph.D. Department of Computer Science, University of Essex, (2002).
- [10] F. Glover, "Tabu Search – Part 1". *ORSA Journal on Computing.* 1 (2),pg. 190, (1989).
- [11] A. Linhares, *Microcanonical Optimization Applied to the Traveling Salesman Problem*, *International Journal of Modern Physics C* 9(1), pg. 133, (1998).
- [12] D.E. Goldberg, J.H. Holland, *Genetic algorithms and machine learning*. *Mach Learn* 3(2), pg. 95, (1988).
- [13] Chien-Wei Chu, Min-Der Lin, Gee-Fon Liu, Yung-Hsing Sung, *Application of immune algorithms on solving minimum-cost problem of water distribution network*, *Mathematical and Computer Modelling*, Volume 48, Issues 11–12, Pages 1888-1900, ISSN 0895-7177,(2008).
- [14] S. Rainer, P. Kenneth, *Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces*. *J Global Optim* 11(4), pg. 341, (1997).

- [15] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, K. Zenger, "Harmony Search Method: Theory and Applications", *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 258491, (2015).
- [16] R. Eberhart, J. Kennedy. A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS'95.*, pg. 39. IEEE, (1995).
- [17] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization - artificial ants as a computational intelligence technique. *IEEE Comput Intell Mag* 1, pg. 28, (2006).
- [18] D. Karabođa. "An Idea Based on Honey Bee Swarm For Numerical Optimization", *Computer Science*, (2005).
- [19] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, *Knowledge-Based Systems*, Volume 159, Pg. 20, ISSN 0950-7051, (2018).
- [20] Melanie Mitchell. *An Introduction to Genetic Algorithms*, (1996).
- [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and machine Learning*, (New York: Addison-Wesley Longman Publishing Co., 1989).
- [22] J. H. Holland, "Adaptation" in Rosen, R. and Snell, F.M. (Eds), *Progress in Theoretical Biology*, Vol. 4 pg. 263, (1976).
- [23] J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, 1992, (University of Michigan Press, Ann Arbor, MI, 1975).
- [24] Mitchell Melanie, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, London, England, Fifth printing (1999).
- [25] J-Y. Lee, M-S. Kim, C-T. Kim and J-J. Lee, Study on encoding schemes in compact genetic algorithm for the continuous numerical problems, *SICE Annual Conference 2007, Takamatsu*, pg. 2694, (2007).
- [26] G. Shi, H. lima and N. Sannomiya, A new encoding scheme for solving job shop problems by genetic algorithm, *Proceedings of 35th IEEE Conference on Decision and Control, Kobe, Japan*, pg. 4395, vol.4, (1996).
- [27] Sivanandam SN, Deepa SN, *Introduction to genetic algorithm*, 1st edn. Springer-Verlag, Berlin Heidelberg, (2008).
- [28] Kumar A, Encoding schemes in genetic algorithm. *Int J Adv Res IT Eng* 2(3), pg. 1, (2013).
- [29] Lima S.J.A., de Araújo S.A., A new binary encoding scheme in genetic algorithm for solving the capacitated vehicle routing problem. In: Korošec P., Melab N., Talbi EG.

- (eds) *Bioinspired Optimization Methods and Their Applications (BIOMA)*, Lecture notes in computer science, vol 10835. Springer, Cham, (2018).
- [30] Lee CKH, A review of applications of genetic algorithms in operations management. *Eng Appl Artif Intell* 76:1–12, (2018).
- [31] Jebari K, Selection methods for genetic algorithms. Abdelmalek Essaâdi University. *International Journal of Emerging Sciences* 3(4):333–344, (2013).
- [32] G. K. Soon, T. T. Guan, C. K. On, R. Alfred and P. Anthony, "A comparison on the performance of crossover techniques in video game," 2013 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, pg. 493, (2013).
- [33] N. Hesam Mahmoudi Nezhad, *Optical System Optimization Using Genetic Algorithms*, Master of Science Thesis (2014).
- [34] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of Genetic Algorithms I, FOGA*, G. J. E. Rawlins. Ed. pp. 205-218, 1990.
- [35] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer, (1996).
- [36] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," New York: Addison-Wesley, (1989).
- [37] M. Chen and F. Liao, "Adaptive mutation operators and its applications," *Journal of Dayeh University*, vol. 7, no. 1, pg. 91, (1998).
- [38] Yudong Zhang, Shuihua Wang, and Genlin Ji, A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications, *Mathematical Problems in Engineering*, Vol.2015, Article ID 931256, (2015).
- [39] Kennedy, J., and R. Eberhart. "Particle Swarm Optimization." *Proceedings of the IEEE International Conference on Neural Networks*. Perth, Australia, pg. 1942, (1995).
- [40] Mezura-Montes, E., and C. A. Coello Coello. "Constraint-handling in nature-inspired numerical optimization: Past, present and future." *Swarm and Evolutionary Computation*, pg. 173, (2011).
- [41] Pedersen, M. E. "Good Parameters for Particle Swarm Optimization." Luxembourg: Hvass Laboratories, (2010).
- [42] Y. Zhang, S. Balochian, P. Agarwal, V. Bhatnagar, and O. J. Housheya, "Artificial intelligence and its applications," *Mathematical Problems in Engineering*, vol. 2014, Article ID 840491, (2014).
- [43] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by simulated annealing*, Science, Volume 220, Number 4598, pg. 671, (1983).

- [44] Zhao Xinchao, Simulated annealing algorithm with adaptive neighborhood, *Applied Soft Computing*, Volume 11, Issue 2, Pg.1827, (2011).
- [45] V. Černý, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pg. 41, (1985).
- [46] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pg. 721-741, (1984).
- [47] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, *Operations Research*, Vol. 21, No. 2, pg. 498, (1973).
- [48] S. Salcedo-Sanz, R. Santiago-Mozos, C. Bousono-Calzon, A hybrid hopfield network-simulated annealing approach for frequency assignment in satellite communications systems, *IEEE Trans. Syst. Man Cybernet. Part B: Cybernet.*, 34 (2), pg. 1108, (2004).
- [49] J.M. Framinana, C. Schusterb, An enhanced timetabling procedure for the no-wait job shop problem: a complete local search approach, *Comput. Operations Res.*, 331, pg. 1200, (2006).
- [50] E.K. Burke, A. Eckersley, B. McCollum, Using simulated annealing to study behaviour of various exam timetabling data sets, *Proceedings of the Fifth Metaheuristics International Conference (MIC 2003)*, Kyoto, Japan, August (2003).
- [51] P.J.M. Laarhoven, van, E.H.L. Aarts, J.K. Lenstra, Job shop scheduling by simulated annealing. *Operations Research*, 40(1), pg.113, (1992).
- [52] M. Locatelli, Convergence of a simulated annealing algorithm for continuous global optimization, *J. Global Optimization*, 18 , pg. 219, (2000).
- [53] L. Ingber, Adaptive simulated annealing (ASA): lessons learned, *Pol. J. Control Cybernet.*, 25 (1), pg. 33, (1996).
- [54] H.H. Szu, R.L. Hartley, Fast simulated annealing, *Phys. Lett. A*, 122, pg. 157, (1987).
- [55] Y. Elhaddad, O. Sallabi , A New Hybrid Genetic and Simulated Annealing Algorithm to Solve the Traveling Salesman Problem, *Proceedings of the World Congress on Engineering 2010 Vol I WCE 2010*, London, U.K., (2010).
- [56] Y. Zhang, S. Wang, Y. Sun, G. Ji, P. Phillips, and Z. Dong, "Binary structuring elements decomposition based on an improved recursive dilation-union model and RSAPSO method," *Mathematical Problems in Engineering*, vol. 2014, Article ID 272496, (2014).

[57] Shi-Ming Chen, Ali Sarosh, Yun-Feng Dong, Simulated annealing based artificial bee colony algorithm for global numerical optimization, Applied Mathematics and Computation, Volume 219, Issue 8, Pg. 3575, (2012).

[58] <http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent>

4. Multi-Electrode Lens Optimization Using Genetic Algorithms

Abstract

In electrostatic charged particle lens design, optimization of a multi-electrode lens with many free optimization parameters is still quite a challenge. A fully automated optimization routine is not yet available, mainly because the lens potential calculations are often done with very time-consuming methods that require meshing of the lens space. A new method is proposed that improves on the low speed of the potential calculation while keeping the high accuracy of the mesh-based calculation methods. This is done by first using a fast potential calculation based on the so-called Second-Order Electrode Method (SOEM), at the cost of losing some accuracy, and then using a Genetic Algorithm (GA) for the optimization. Then, by using the parameters of the approximate systems found from this optimization based on SOEM, an accurate GA optimization routine is performed based on potential calculation with the commercial finite element package COMSOL. A six-electrode electrostatic lens was optimized accurately within a few hours, using all lens dimensions and electrode voltages as free parameters and the focus position and maximum allowable electric fields between electrodes as constraints.

Keywords

Multi-electrode lens design; optimization; second-order electrode method (SOEM); genetic algorithms (GAs).

This chapter is based on the following publication:

N Hesam Mahmoudi Nezhad, M Ghaffarian Niasar, Ali Mohammadi Gheidari, CW Hagen, P Kruit," Multi-electrode lens optimization using genetic algorithms", International Journal of Modern Physics A, Volume 34, 2019.

4.1 Introduction

Although optimization routines are very powerful techniques for finding optima of complex functions, they have not been used extensively in the field of charged particle optics. To find the optimum design of electrostatic lenses has always been quite challenging, especially when multiple electrodes are involved. Mostly, this is done manually in a trial and error fashion. Given a certain lens geometry and voltages on the electrodes, the electrostatic field is usually calculated numerically, meshing either the entire volume of the lens (finite element method) or the electrode surfaces (charge density method). From the axial potential and its derivatives (at least up to second-order) the first-order optical properties and the aberration coefficients of the lens can then be calculated [1]. To optimize the design, an objective function is then defined which needs to be minimized by changing the lens geometry and/or electrode potentials. For each trial-design this sequence of steps, of which the field calculation is most time consuming, needs to be done. Some attempts toward automated optimization were made in the past by Szilagy et al., [2,3] Adriaanse et al. [4] and Barth et al. [5-8] They avoided the time-consuming field calculation and calculated the axial potential using the Second-Order Electrode Method (SOEM) [4,5]. However, at that time, computers were still slow and the results were not very accurate. Now that more powerful computers are available it is worthwhile to revisit the problem. A new approach is proposed in which a Genetic Algorithm (GA) is used as the optimization technique. The lens systems considered are rotationally symmetric electrostatic lens systems with multiple electrodes. To achieve a quick evaluation of the objective function, SOEM is used. Although this comes at the cost of losing some accuracy, it allows for rapid optimization using the GA. Using the parameters from this optimization, a more accurate GA optimization is subsequently performed, in which the objective function is calculated using the electrostatic field simulated with COMSOL [9] (based on the finite element method). As a demonstration, a six-electrode lens system was optimized for the smallest focused probe size within a few hours, with all geometries and electrode voltages as free parameters, and the focus position and maximum breakdown fields between electrodes as constraints. To extend the method to more complex systems and designs is rather straightforward. This chapter is organized as follows. In section 4.2 a brief summary of SOEM will be presented. The geometry of the example system used in this study is described in section 4.3. The fitness function, or objective function, needed for the optimization is discussed in section 4.4. Section 4.5 addresses the accuracy of SOEM compared to more accurate

methods such as finite element methods. The GA is introduced in section 4.6 and the parameters relevant in the optimization process are discussed. Results of the GA optimization using various methods to calculate the objective function are presented in section 4.7 and 4.8, and conclusions are drawn in section 4.9.

4.2 SOEM

In a rotationally symmetric system, the potential in space can be expressed in terms of the axial potential and its derivatives with respect to the axial coordinate z :

$$\varphi(r, z) = \varphi(0, z) - r^2/4\varphi^{(2)}(0, z) + r^4/64\varphi^{(4)}(0, z) - \dots \quad (4.1)$$

Here, $\varphi(0, z)$ is the axial potential, $\varphi^{(n)}(0, z)$ is the n th derivative with respect to z and r is the radial coordinate. To allow for a fast calculation of the axial potential, this equation is solved by fitting the axial potential with a cubic spline [10] (i.e. a twice differentiable function built piece wise from polynomials of second-order). This approximation requires to omit terms higher than second-order in Eq. 4.1, which is then rewritten for each spline interval as:

$$\varphi''(r, z) = 4 [\varphi_i(0, z) - \varphi_i(r, z)] / r^2 \quad (4.2)$$

where i is the index of a discrete axial point. A cubic spline on an arbitrary mesh of N can be written as [4]:

$$\varphi\left(\frac{1}{\Delta_i} + \frac{1}{\Delta_{i-1}}\right) = \frac{\varphi_{i-1}}{\Delta_{i-1}} + \frac{\varphi_{i+1}}{\Delta_i} - \frac{\varphi_{i-1}''\Delta_i + 2\varphi_i''(\Delta_i + \Delta_{i-1}) + \varphi_{i+1}''\Delta_i}{6}, \quad (4.3)$$

$$i = 2, \dots, N - 1$$

Here, Δ_i is the interval between point i and $i + 1$. In our lens model, the points placed inside each electrode are assumed to be equidistant and the gaps between electrodes are taken as one single spline interval. Equation 4.2 can be substituted in Eq. 4.3 to get:

$$\varphi_{i-1} \left(\frac{2\Delta_i}{3r_{i-1}^2} - \frac{1}{\Delta_{i-1}} \right) + \varphi_i \left(\frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} + \frac{4\Delta_{i-1} + \Delta_i}{3r_i^2} \right) + \varphi_{i+1} \left(\frac{2\Delta_i}{3r_{i+1}^2} - \frac{1}{\Delta_i} \right) = \quad (4.4)$$

$$\frac{2V_{i-1}}{3r_{i-1}^2} \Delta_{i-1} + \frac{4V_i}{3r_i^2} (\Delta_{i-1} + \Delta_i) + \frac{2V_{i+1}}{3r_{i+1}^2} \Delta_{i+1}, \quad i = 2, \dots, N - 1$$

where r_i is the radius of the electrode at position i and V_i is the electrode potential at point i . In Eq. 4.4, renaming the coefficients of φ_{i-1} , φ_i and φ_{i+1} by a_i , b_i and c_i , respectively, and the right-hand side of the equation by d_i , it follows that:

$$\begin{aligned}
 a_i &= \frac{2\Delta_i}{3r_{i-1}^2} - \frac{1}{\Delta_{i-1}}, \quad b_i = \frac{1}{\Delta_{i-1}} + \frac{1}{\Delta_i} + \frac{4\Delta_{i-1} + \Delta_i}{3r_i^2} \\
 c_i &= \frac{2\Delta_i}{3r_{i+1}^2} - \frac{1}{\Delta_i}, \quad d_i = \frac{2V_{i-1}}{3r_{i-1}^2}\Delta_{i-1} + \frac{4V_i}{3r_i^2}(\Delta_{i-1} + \Delta_i) + \frac{2V_{i+1}}{3r_{i+1}^2}\Delta_{i+1} \\
 a_i\varphi_{i-1} + b_i\varphi_i + c_i\varphi_{i+1} &= d_i
 \end{aligned} \tag{4.5}$$

Equation 4.5 correlates the axial potential at each point to its value at the neighboring points, and can be represented as a tri-diagonal matrix:

$$\begin{bmatrix}
 b_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\
 a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 \\
 0 & a_3 & b_3 & c_3 & 0 & 0 & 0 \\
 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\
 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\
 0 & 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\
 0 & 0 & 0 & 0 & 0 & a_N & b_N
 \end{bmatrix} \times \begin{bmatrix} \varphi_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \varphi_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{bmatrix} \tag{4.6}$$

where the matrix elements (a_i , b_i , c_i and d_i , $i = 2 : N - 1$) are known parameters related to the geometry of the lenses and electrode voltages. Applying boundary conditions (zero field and derivative at the first and last electrode), the tridiagonal matrix can be solved very fast by numerical methods (for which MATLAB [11] is used in this work) to derive the axial potential at all points i . Accordingly, first and second derivatives can also be numerically calculated.

4.3 Lens geometry

The example lens system considered here is a six-electrode rotationally symmetric electrostatic lens as schematically shown in Figure 4.1. The object is assumed to be far left from the lens system (10 times the length of the system) and the image plane is fixed at $z = 3.5$ mm from the entrance of the lens. The initial electron energy is eV_1 , where V_1 is the voltage of the first electrode and the landing energy of the electrons in the image plane is fixed at 1 keV. The lens parameters such as thicknesses of the electrodes, radii of the holes, gaps between electrodes and the voltages of each electrode, are the variable parameters of the optimization problem. The ranges of these free parameter are from 0.05 to 0.3 mm, 0.01 to 0.3 mm and 0.1 to 0.6 mm, for

thicknesses, radii and gaps, respectively. The voltages at each electrode (V_i) are free to change in the range of 500 V to 5 kV. However, there are some constraints. First of all, the maximum field between two sequential electrodes has to be less than 10 kV/mm to avoid voltage breakdown, i.e.

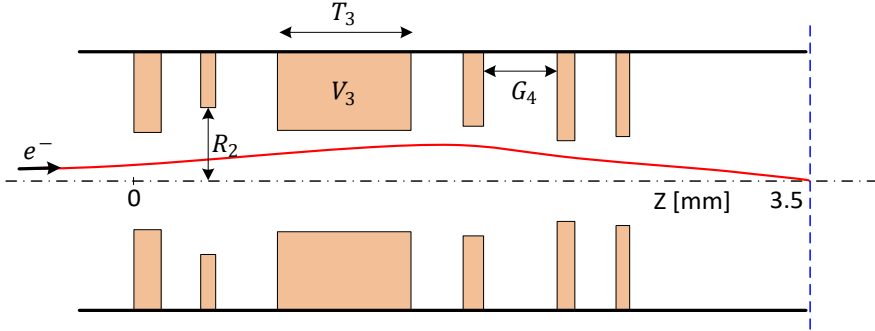


Figure 4.1. Schematic of a six-electrode lens (T_i : thicknesses, R_i : radii, G_i : gaps between electrodes, V_i : voltages of the electrodes).

$$(V_{i+1} - V_i) / (T_{i+1} - T_i) < 10 \text{ kV/mm (no tolerance)} \quad (4.7)$$

where V_i and T_i are the voltage and the thickness of the i th electrode, respectively. The second constraint is that the image plane is fixed at a position X_c .

$$X_c = \text{const. (with 2\% tolerance)} \quad (4.8)$$

4.4 The objective function

In a rotationally symmetric probe-forming electron-optical system, the image of a point source is a blurred spot due to geometrical and chromatic aberrations of the imaging system. Geometrical aberrations strongly depend on the geometry of the lens as well as the voltages on the electrodes. The optimization routine can be used to find the optimum lens design by tuning the geometry of the lens and/or the voltages of the electrode to minimize the aberration contributions, thereby obtaining the smallest spot size. Since the aim is to minimize the spot size at the image side, it is taken as the objective function (or the so-called “fitness function” in the context of GA). Assuming that the lens only suffers from axial spherical and chromatic aberrations, the spot size can be expressed as [12]:

$$D_s^2 = (0.18 C_s \alpha^3)^2 + (0.6 C_c \alpha \frac{\Delta U}{U})^2 \quad (4.9)$$

where, if D_s is calculated at the image side (as is the case here), all parameters in the equation are taken at the image side. The first term is the contribution from spherical aberration, with C_s the spherical aberration coefficient of the lens, the second term is the contribution from chromatic aberration, with C_c the chromatic aberration coefficient of the lens, α is the half opening angle of the beam (chosen to be 10 mrad in this case-study), ΔU is the energy spread of the electron source (taken as 1 eV) and U is the acceleration energy (equal to the potential at the image plane = 1 kV). Note that the source image contribution is neglected in Eq. 4.9. The aberration coefficients are first calculated at the object side. Then using the magnification, they are converted to the image side. The spherical and chromatic aberration coefficients can be calculated from the axial potential, its derivatives and a principle imaging ray $r_\alpha(z)$, starting on the optical axis in the object with angle 1, using the following integrals [13]:

$$C_s = \frac{1}{16\varphi_0^{1/2}} \int_{Z_o}^{Z_i} \varphi^{1/2} \left\{ \frac{5}{4} \left(\frac{\varphi''}{\varphi} \right)^2 + \frac{5}{24} \left(\frac{\varphi'}{\varphi} \right)^4 \right\} r_\alpha^4 + \frac{14}{3} \left(\frac{\varphi'}{\varphi} \right)^3 r_\alpha^3 r_\alpha' - \frac{3}{2} \left(\frac{\varphi'}{\varphi} \right)^2 r_\alpha^2 r_\alpha'^2 \right\} dz \quad (4.10)$$

$$C_c = -\varphi_0^{1/2} \int_{Z_o}^{Z_i} \left(\frac{3\varphi'^2}{8\varphi^2} \right) r_\alpha^2 dz \quad (4.11)$$

whereas Z_o and Z_i denote the z-axis positions of the object and image, respectively. C_s and C_c are calculated at the object side, and φ_0 refers to the potential at the object side. The principle imaging ray $r_\alpha(z)$ is obtained by ray tracing. The relativistic equation of motion, in the paraxial approximation, takes the following form [13]:

$$\frac{1 + \varepsilon}{1 + 2\varepsilon} \varphi(z) \cdot r_\alpha''(z) + \frac{1}{2} \varphi'(z) \cdot r_\alpha'(z) + \frac{1}{4} \varphi''(z) \cdot r_\alpha(z) = 0 \quad (4.12)$$

where $\varepsilon(z) = e/(2m_0c^2)$ in which e , m_0 and c are the elementary charge, electron rest mass and velocity of light, respectively. The equation is solved numerically. The magnification M is obtained by using another principle ray r_b , starting in the object plane at a height h and with angle 0, and calculating the height in the image plane, as determined by the axis crossing of r_α at the image side, and dividing by h . C_s and C_c at the image side are then obtained by:

$$C_{s_{im}} = M^4 \cdot \left(\frac{V7}{V1}\right)^{\left(\frac{3}{2}\right)} \cdot C_{s_{obj}} \quad (4.13)$$

$$C_{c_{im}} = M^2 \cdot \left(\frac{V7}{V1}\right)^{\left(\frac{3}{2}\right)} \cdot C_{c_{obj}} \quad (4.14)$$

Inserting $C_{s_{im}}$ and $C_{c_{im}}$ into Eq. 4.9 the spot size D_s at the image side is calculated. When, in the remainder of the paper, the spot size is mentioned, the spot size in the image plane is meant.

4.5 Accuracy of SOEM

EOD [14] is a professional software package commonly used by electron optical lens designers, while COMSOL is a multi-physics finite element-based solver software which is not very commonly used so far by electron optical designers. We wanted to use a well-known and reliable electron optical software package as the main reference (benchmark) for comparison. But we also wanted to make use of a program which can be integrated in MATLAB to accurately calculate the potential in an automated way, and could be used together with our other codes (ray-tracing and optimization). As EOD could not easily be automated and integrated into MATLAB, we have done that part using COMSOL. Although it is expected that COMSOL also produces accurate results, for more certainty, EOD is also taken as a benchmark initially to compare to results obtained from COMSOL. It has been shown that COMSOL gives results as accurate as EOD (Figure 4.2), COMSOL then has been used as an accurate software program to calculate the potential, and it has been used as a reference for the rest of the study. To evaluate the accuracy of the potential and its derivatives as calculated using SOEM, a comparison was made to results obtained with EOD, as well as with the finite element package COMSOL. The calculation was done for the lens system of Figure 4.1. The axial potentials and its first and second derivatives, as calculated using SOEM, COMSOL and EOD for two different designs of the system of six-electrodes (Figure 4.1), are shown in Figures 4.2 and 4.3. Another example of a three-electrode lens system can be found in [15].

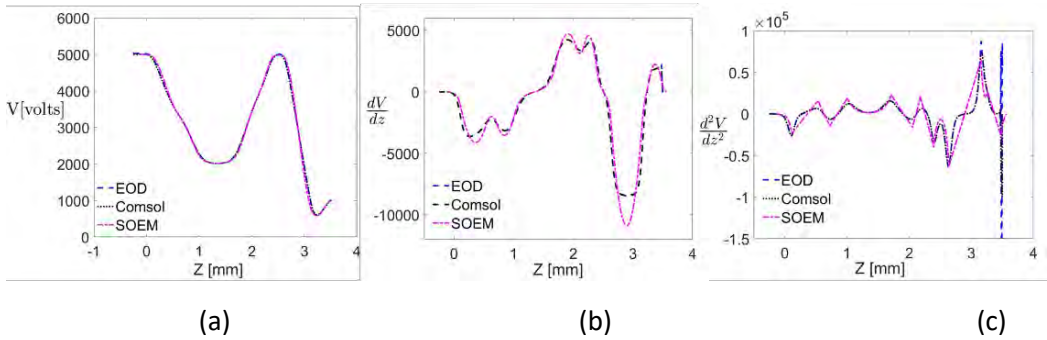


Figure 4.2. Comparison of the axial potential and its first and second derivatives, calculated by SOEM, COMSOL and EOD, for a six-electrode lens of Figure 4.1, $T_i = [0.11, 0.18, 0.70, 0.12, 0.24, 0.10] \text{ mm}$, $R_i = [0.10, 0.23, 0.21, 0.21, 0.09, 0.08] \text{ mm}$, $G_i = [0.43, 0.29, 0.35, 0.22, 0.52] \text{ mm}$, $V_i = [5000, 3300, 2000, 3850, 5000, 550] \text{ V}$. (Results: C_s COMSOL = 1.6 mm, C_c COMSOL= 0.59 mm, C_s SOEM = 1.3 mm, C_c SOEM= 0.52 mm; all values taken at the image side)

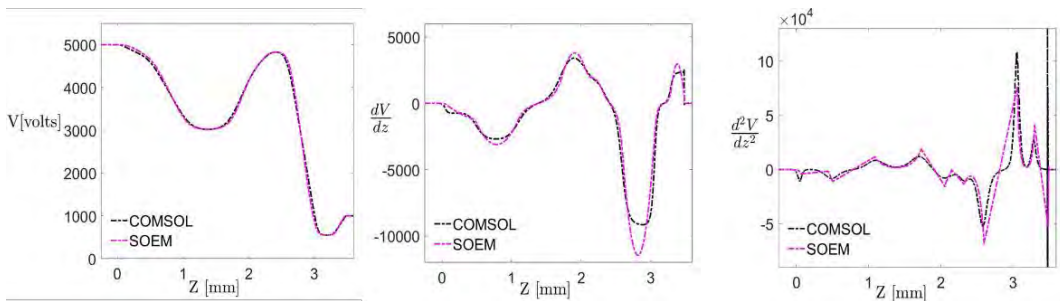


Figure 4.3. Comparison of the axial potential and its first and second derivatives, calculated by SOEM and COMSOL, for a six-electrode lens of Figure 4.1, $T_i = [0.06, 0.06, 0.64, 0.09, 0.27, 0.24] \text{ mm}$, $R_i = [0.05, 0.19, 0.23, 0.24, 0.12, 0.06] \text{ mm}$, $G_i = [0.38, 0.59, 0.33, 0.18, 0.46] \text{ mm}$, $V_i = [5000, 4750, 3000, 4550, 4850, 550] \text{ V}$ (Results: C_s COMSOL = 3.13 mm, C_c COMSOL = 0.69, C_s SOEM = 9.88 mm, C_c SOEM = 1.41 mm; all values taken at the image side).

The results calculated with COMSOL and EOD agree very well with each other, even for the second derivatives. But the potential calculated by SOEM is slightly different, which becomes more visible in the derivatives. At first sight, the difference between the derivatives calculated by SOEM and COMSOL in both figures looks pretty much the same. But when calculating the aberration coefficients with both methods, almost the same C_s and C_c are found for the system of Figure 4.2 but for the system of Figure 4.3 COMSOL results in a three times lower C_s and a two times lower C_c . This means that the objective function is a quite sensitive function of the potential, and the

approximate potential as determined by SOEM may lead to deviations of the objective function. Irrespective of that, the higher computation speed of SOEM, compared to the commercial programs, still makes it attractive to use in an optimization routine. As many parameters are involved in optimizing the lens system considered here, it is expected that the objective function landscape may have many local minima. Therefore, a global optimizer is needed. Due to its random nature, the GA is a good candidate to find such a global solution [16,17]. In section 4.6 GA will be briefly introduced.

4.6 Genetic Algorithm optimization using SOEM

The GA is a heuristic search technique with a logic which mimics the natural evolution process [18]. The GA optimization procedure starts with a set of systems (each system representing a “chromosome” in the context of GA) called the initial population. The initial population can be a randomly generated state denoted by $P_1(x_1, \dots, x_n)$, where the x_i are vectors representing the chromosomes of the individual systems i . For the lens system of Figure 4.1, the initial population is the set of lens systems with each system characterized by a vector containing its design parameters, i.e. a vector of 23 parameters including six electrode thicknesses (T_i) (the third thickness, as a choice, is kept dependent to the other thicknesses and gaps, to keep the length of system constant), 6 radii (R_i), 5 gaps (G_i) and 6 electrode voltages (V_i). This initial population gradually evolves toward a set of chromosomes (set of lens parameters) with better properties, to be judged by evaluating the objective function, a function of the chromosomes (lens design parameters). The evolution leads to a new generation, a new population of systems $P_{i+1}(x_1, \dots, x_n)$ closer to the optimum of the objective function. The creation of a new population, in a so-called Simple Genetic Algorithm (SGA), is mainly done using operators such as selection, crossover and mutation. Systems are selected based on the value of the objective function. The crossover and mutation operators serve as a mechanism to produce the population for the next generation. Crossover is the operator which combines two individuals (parents) to create a new individual (offspring). The idea behind using this operator is to create new individuals with better characteristics than their parents by inheriting the best characteristics from each parent. Crossover is applied on the population during evolution based on a user definable crossover probability. Mutation is a procedure applied in the algorithm, analogous to biological mutation, to give genetic diversity by altering genes in a chromosome of a population from one generation to the next generation. This procedure is applied in the algorithm to give each individual a better

opportunity to reach the full space of possible solutions by enhancing the diversity of the population. Elitism is an additional operator, which could be added to the process to enhance the result of the GA. This operator copies a certain number (i.e. the elite count) of best parents, unchanged, into the next generation. This prevents the GA from wasting time in rediscovering these best parents which could otherwise have been discarded while breeding new populations. A schematic of a simple GA is shown in Figure 4.4. SOEM is used in the objective function evaluation part of the optimization. Generation is the number of iterations the GA algorithm executes before it terminates. This parameter can be set as the stopping criterion of the GA, as was done in this study. The population size is the number of individuals present in each generation. The MATLAB GA module with constraints was used for the optimization. The potential calculation using SOEM and the ray-tracing codes to evaluate the objective function were coded in MATLAB as well. In section 4.6.1 the choice of GA optimization parameters is discussed.

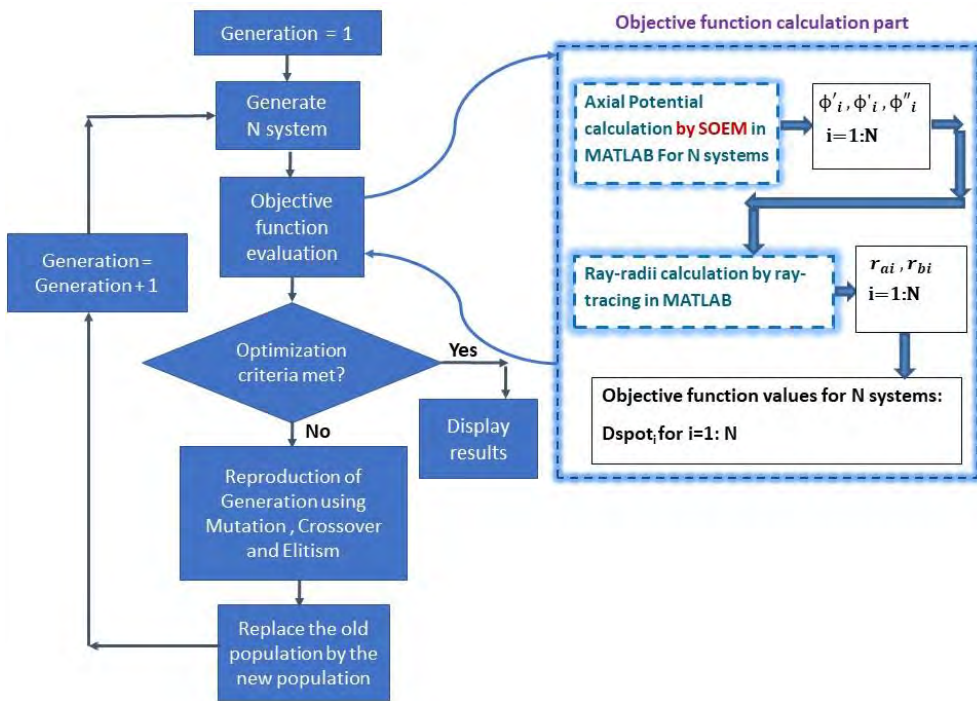


Figure 4.4. A schematic of the GA, with SOEM used for the objective function evaluation.

4.6.1 GA Parameters

The GA has a variety of parameters such as initial population, population size, generation, crossover, mutation and elitism (elite count). These parameters should be tuned well based on the exact application, to get the best performance.

Increasing the population size, the GA will search the space more extensively. This increases the probability of reaching the global solution and reduces the chance of getting trapped in local optima. However, increasing the population size to large values will also significantly increase the execution time of the GA. Thus, it is a tradeoff between run time and the best result. As a good compromise a population size of 50 was chosen in this study. The MATLAB GA routine contains different crossover and mutation types. For optimization problems with constraints, as is the case here, most of the available crossover and mutation types are not applicable. Among the ones applicable, the types “Intermediate” and “Adapt-feasible” for crossover and mutation, respectively, were found to be the most efficient ones.

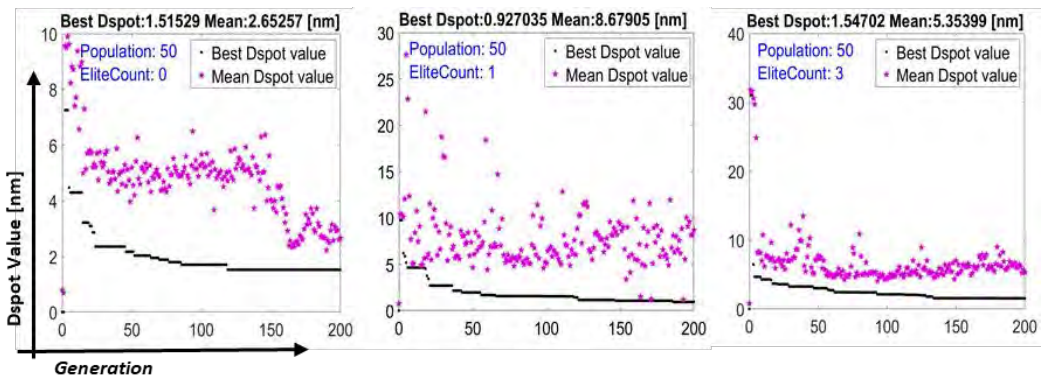


Figure 4.5. Three different GA runs to assess the generation number and elite count. The elite count is indicated in the graphs, and the lowest Dspot value achieved in each graph is indicated above the graphs.

The effects of increasing the generation number and elite count on the performance of the GA was also assessed. As the generation number has a direct effect on the execution time of the algorithm, a tradeoff between result and execution time has to be made. Figure 4.5 shows a few examples of these assessments. The plots show the value of the objective function, or the spot size D_{spot} , as a function of the generation number, for three different values of the elite count: 0, 1 and 3. For each generation number the mean value of D_{spot} is plotted (purple pentagram, light pentagram in gray scale) as well as the smallest, or best, value of D_{spot} (black dots, dark dots in gray scale).

It is seen, in all plots, that increasing the number of generations to more than 200 did not significantly lower the best value of D_{spot} , while the execution time would continue to increase. Therefore, the number of generations was limited to 200 in this study. Analyzing the effect of the elite count parameter on the result of the GA, it is observed that a value of one (the middle graph in Fig. 4.5) led to the smallest value of D_{spot} ($D_{\text{spot}} = 0.927$). So, a value of one was chosen in this work.

Summarizing, the MATLAB GA parameters chosen for this study were: generation number 200, population number 50, elite count 1, selection type stochastic uniform, crossover type Intermediate, and mutation type Adapt-feasible.

4.7 Results of GA optimization with SOEM

A GA run starts with the first generation of 50 systems, randomly created within the parameter ranges defined in section 4.3, and then evolves for 200 generations. In a run 10000 systems in total are evaluated, of which around 1500 systems (15%) fulfilled the constraints and went through the selection, based on the value of the objective function as determined by SOEM. The spot sizes D_{spot} obtained for these 1500 systems are sorted in descending order and plotted in that order in Fig. 4.6 (the black * symbols). Note that the horizontal axis represents the position of the systems in the sorted array according to their spot sizes and not the systems in which the plotted spot sizes were found. The plot also contains information on the generation in which each spot size value was obtained. Corresponding to each point on the black curve, there is a purple pentagram (or pentagrams) in the graph. The pentagrams show the generation in which the system index has occurred (the pentagram lies on a line that passes through that black dot and is perpendicular to x-axis). It is seen that larger spot sizes can be found in all generations, basically but the smallest spot sizes (on the right-hand side of the graph) occur predominantly in higher generations. The smallest spot size of 0.93 nm was found first in the 187th generation.

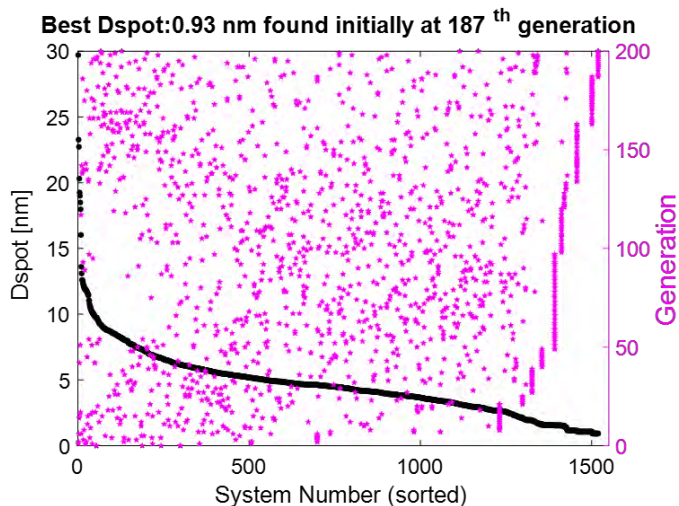


Figure 4.6. The spot size (Dspot, left-hand axis), taken from one optimization run using SOEM, sorted in decreasing order (black (dark in gray scale) dots), and the generation in which the spot size occurred (purple (light in gray scale) pentagrams, right-hand axis).

In section 4.5, it was seen that SOEM only provides an approximate potential, compared to the potential obtained from COMSOL, and it was seen for two different systems that the values of C_s and C_c are quite sensitive to the actual potential. Therefore, it was investigated how far off the SOEM optimization results are from COMSOL results.

The values of C_s , C_c , X_c (the image plane position), and Dspot (spot size) were obtained by an optimization with SOEM, and were then compared to the values as calculated using COMSOL for the systems found. As, by nature of the GA, the resulting data of a run, are jumping up and down to reach to a final good result (as is seen in Figure 4.6), the data are again sorted by their objective function value in descending order, to more clearly see the trends. Because it is time consuming to do the field calculation with COMSOL, not all 1500 systems were calculated. Of 980 systems only one out of 20 systems were taken, resulting in still a representative number of 49 systems to compare. In Figure 4.7, the values of C_s , C_c , X_c and Dspot, based on the SOEM optimization, are plotted in descending order (the black star symbols). For each system indicated with a black star the same parameters were then inserted in the field calculation with COMSOL. The resulting values of C_s , C_c , X_c and Dspot are shown as red stars in the plots.

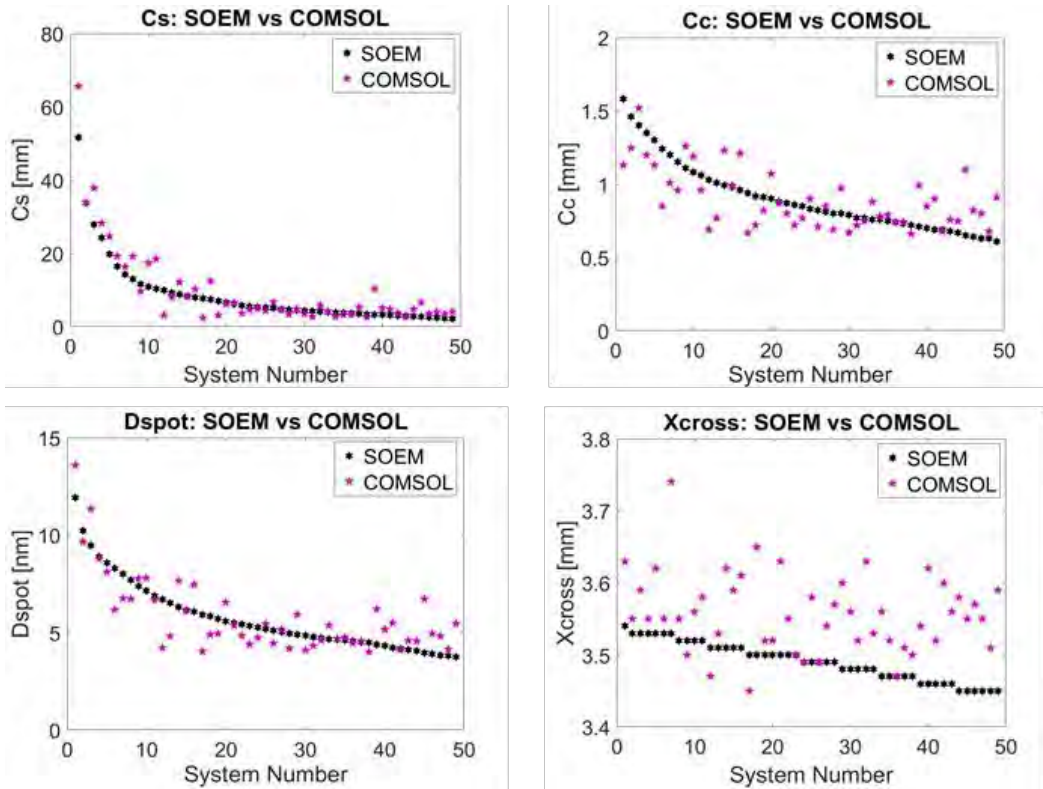


Figure 4.7. The sorted data of spherical aberration (Cs), chromatic aberration (Cc), spot size (Dspot) and Xc (image plane position) for 49 systems, taken from an optimization run based on SOEM and the comparison with accurate values calculated using COMSOL.

It is seen in all plots that the SOEM-based data and the COMSOL-based data show the same trend. The SOEM-based values are mostly close to the more realistic COMSOL-based ones, only in some cases deviations up factors of 2–3 can be seen, as was already seen for the two systems shown in Sec. 4.5. Please note that the smallest spot size in Figure 4.7 is 3.8 nm, and not 0.927 as in Figure 4.6, because only 980 systems are considered instead of 1500. A very critical parameter is the value of Xc, which is one of the constraints. In Figure 4.7, it is seen that the SOEM-based values of Xc are often lower than the COMSOL-based ones, by up to 5%. Such deviations, however, are still considered to be acceptably small. As such, the SOEM-based optimization offers a very good approximate parameter estimation. But because of the observed deviations, SOEM-based GA is not advisable when accurate results are desired. Therefore, a more accurate method is needed, which is proposed in Sec. 4.8.

4.8 GA optimization with SOEM and COMSOL

To improve the optimization accuracy, the possibility to integrate COMSOL in the optimization routine was investigated. This provides a more accurate calculation of the potential and the objective function but the optimization will become more time consuming. As a compromise, a first rough optimization with SOEM is performed (GA-SOEM), resulting in a few approximately optimized systems. Then, using the parameters of the systems found, a new set of systems is defined and fed to the GA among the population of the first generation. Then a more accurate but somewhat slower, COMSOL-based optimization is done, starting from these almost good systems (GA-SOEM+COMSOL). A schematic of this optimization flowchart is shown in Figure 4.8. This method prevents the GA from running for many more generations before reaching a good system. This method has been tested and succeeded in very accurately optimizing the example system of Sec. 4.3, within a few hours.

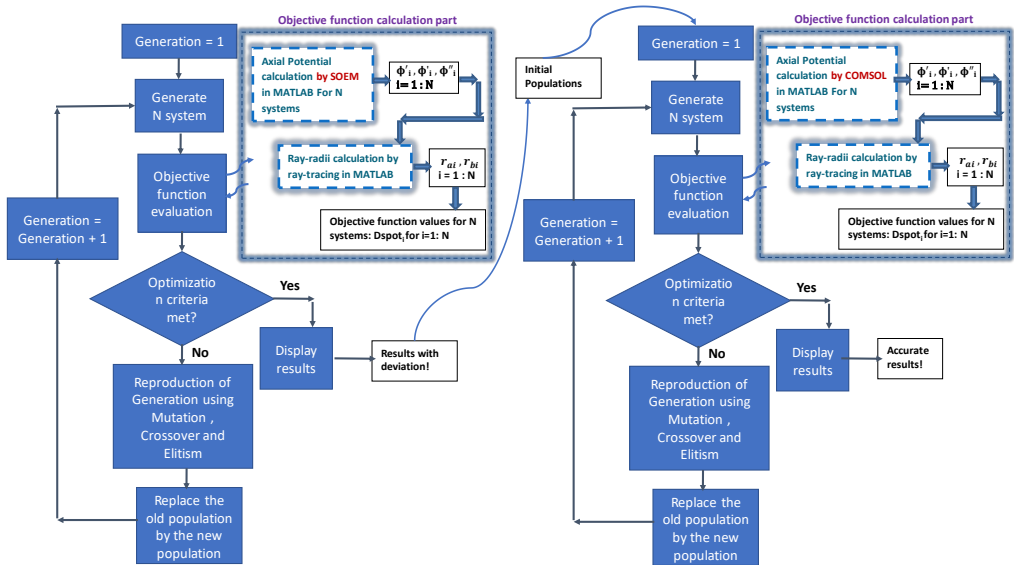


Figure 4.8. Schematic of the flowchart of the GA based on a combination of SOEM and COMSOL.

In Figure 4.9(a), the resulting spot size as obtained from the GA-SOEM, with 50 populations and 200 generations, is shown as a function of the generation number. Again, as in Figure 4.5, the minimum spot size and the mean spot size are plotted. This optimization run took about 30 min and the best spot size was 0.927 nm. For this best system the potential and its derivatives were subsequently calculated with COMSOL,

resulting in a more accurate spot size for this system of 4 nm and a slightly different location of the focal plane.

In the next step, the 20 best systems found by GA-SOEM are fed as initial population into GA-SOEM+COMSOL. MATLAB GA automatically creates the additional 10 random systems to create a total of 30 initial populations. Afterwards, GA-SOEM+COMSOL is run for only 20 generations to reach the best solution of 3.317 nm (see Figure 4.9(b)), i.e. slightly smaller than the 4 nm. This took about 1 h. To continue for more generations did not lead to further improvement. Thus, running GA-SOEM followed by GA-SOEM+COMSOL resulted in a good and accurate system in about 1 h. To assess the speed of GA-SOEM+COMSOL, this method is compared with the situation where the GA is only based on COMSOL (GA-COMSOL). In Figure 4.9(c), the result of GA-COMSOL is presented, where the GA started from randomly created systems and ran for 70 generations, with a population of 50. The execution time was about 5.5 h. The smallest spot size at the first generation was about 12 nm. As is shown, after 70 generations, it still has not arrived at the best solution obtained from GA-SOEM+COMSOL. It is thus concluded that it is worthwhile to first use GA-SOEM, although its results sometimes deviate a bit from accurate results obtained from GA-COMSOL, as a rough optimization and as preoptimized input for a short COMSOL-based GA optimization. A considerable time saving is achieved this way.

As a final example, to show the usefulness of an optimization routine, a manual design of a five-electrode lens system is shown in Figure 4.10(a). The parameters to vary were the electrode diameters, their thicknesses and voltages. The values of C_s , C_c and D_{spot} obtained after many hours of trial and error simulations, are shown in Figure 4.10(a). For comparison, using the GA-SOEM+COMSOL method, a six-electrode system could be found with a smaller spot size, C_s and C_c (see Figure 4.10(b)) in a reasonably fast time (~few hours). A precise time is not given for the traditional methods (manual optimization by the designers), as it is very dependent on the expertise and knowledge of the designer to manually reach a satisfactory result.

However, the consumed time would be far longer than the presented fully-automated optimization technique presented in this paper, which basically tests thousands of systems and reaches an optimized system in a few hours. It should also be mentioned that it is not only a matter of time comparison. To manually reach a satisfactory result for complex systems with all parameters as variables, with many objective functions to be minimized simultaneously, is a very cumbersome task and is not guaranteed in

a reasonable time. This is indeed the advantage of having an automated optimization technique to ensure getting a satisfactory result within a short amount of time.

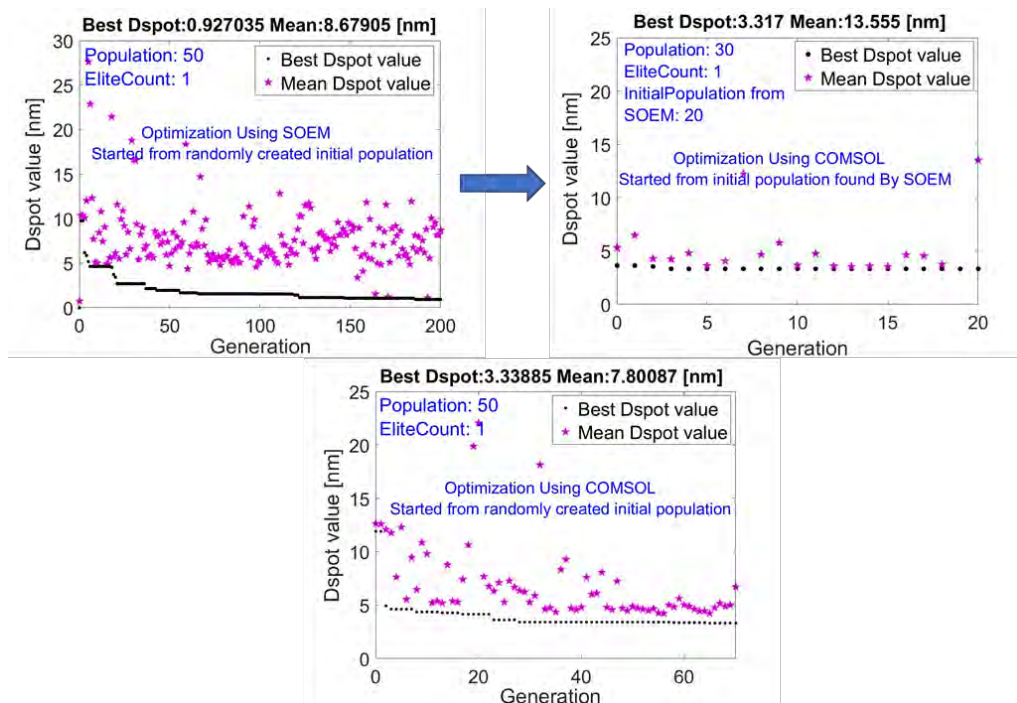


Figure 4.9. (a) optimization using GA-SOEM, (b) optimization using GA-SOEM+COMSOL, starting from 20 initial populations resulting from GA-SOEM (plot a) and (c) optimization only using GA-COMSOL.

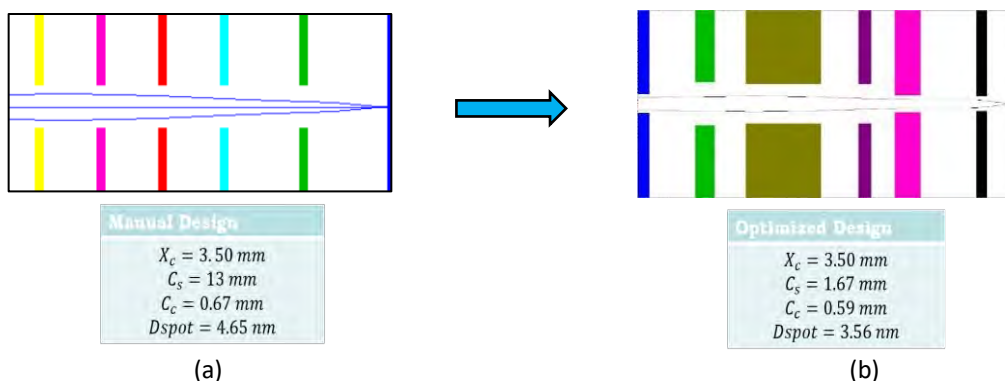


Figure 4.10. An example of a manually optimized five-electrode system (a), compared to a six electrode system optimized using the GA-SOEM+COMSOL method (b).

4.9 Conclusion

Fully automated optimization routines can relieve the laborious manual design of charged particle optics components. However, such routines suffer from low computational speed. The main reason for the low computation speed is the time-consuming process of the lens potential calculation, which is needed to calculate the objective function. This is often done with very time-consuming methods based on a meshing of the lens space. A new method is proposed that improves on the low speed of the potential calculation while keeping the high accuracy of the mesh-based calculation methods. In this method, the potential of the system is calculated using a fast potential calculation based on the so-called SOEM. A set of coarse optima for the design is then selected using a GA. Then by using the parameters of the approximate systems found from this optimization based on SOEM, an accurate GA optimization routine is performed based on a potential calculation with the commercial finite element package COMSOL. A six-electrode electrostatic lens was optimized accurately within a few hours, using all lens dimensions and electrode voltages as free parameters and the focus position and maximum allowable electric fields between electrodes as constraints.

References

- [1] V. Zworykin, *Electron Optics and the Electron Microscope* (J. Wiley & Sons, 1945).
- [2] M. Szilagyı and J. Szep, *J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena* 6, 953 (1988).
- [3] M. N. Szilagyı, *Proc. IEEE* 73, 412 (1985).
- [4] J. Adriaanse, H. van der Steen and J. Barth, *J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena* 7, 651 (1989).
- [5] H. van der Steen, J. Barth and J. Adriaanse, *Nucl. Instrument. Methods Phys. Res. Sec. A: Accelerators, Spectrometers, Detectors and Associated Equipment* 298, 377 (1990).
- [6] H. van der Steen and J. Barth, *J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena* 7, 1886 (1989).
- [7] M. A. J. van der Stam, J. E. Barth and P. Kruit, *Design of a multimode transport lens with optimization program SOEM*, in *Charged-Particle Optics* (1993), pp. 45–57.
- [8] J. E. Barth, H. van der Steen and J. Chmelik, *Proc. SPIE* 2522, 128 (1995).
- [9] Comsol Multiphysics, version 5.3a (2017), <https://www.comsol.com/>.
- [10] G. D. Knott, *Interpolating Cubic Splines* (Birkh user, Boston, 1999).
- [11] Matlab, version r2018 b, <https://www.mathworks.com/products/matlab.html>.
- [12] J. Barth and P. Kruit, *Optik* 3, 101 (1996).
- [13] M. Szilagyı, *Electron and Ion Optics* (Springer Science & Business Media, 1988).
- [14] Eod, version 4.008 (1999-2015 spoc), <http://www.lencova.com/index.php/about-eod>.
- [15] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, T. Janssen, C. Hagen and P. Kruit, *Optimization of electrostatic lens systems using genetic algorithms*, in *Recent Trends in Charged Particle Optics and Surface Physics Instrumentation* (2018), pp. 24–26.
- [16] K. H oschel and V. Lakshminarayanan, *J. Opt.* 1 (2018).
- [17] N. Hesam Mahmoudi Nezhad, *Optical system optimization using genetic algorithms*, Master’s thesis, Delft University of Technology (2014).
- [18] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, London, 1998).

5. Multiple Criteria Optimization of Electrostatic Electron Lenses Using Multi-Objective Genetic Algorithms

Abstract

The design of an electrostatic electron optical system with five electrodes and two objective functions is optimized using multi-objective genetic algorithms (MOGA) optimization. The two objective functions considered are minimum probe size of the primary electron beam in a fixed image plane, and maximum secondary electron detection efficiency at an in-lens detector plane. The time-consuming step is the calculation of the system potential. There are two methods to do this. The first is using COMSOL (finite element method) and the second is using the second order electrode method (SOEM). The former makes the optimization process very slow but accurate, and the latter makes it fast but less accurate. A fully automated optimization strategy is presented, where a SOEM-based MOGA provides input systems for a COMSOL-based MOGA. This boosts the optimization process and reduces the optimization times by at least ~10 times, from several days to a few hours. A typical optimized system has a probe size of 11.9 nm and a secondary electron detection efficiency of 80 %. This new method can be implemented in electrostatic lens design with one or more objective functions and multiple free variables, as a very efficient, fully automated optimization technique.

This chapter is based on the following publication:

Neda Hesam Mahmoudi Nezhad, Mohamad Ghaffarian Niasar, Ali Mohammadi Gheidari, Pieter Kruit, Cornelis Wouter Hagen, "Multiple criteria optimization of electrostatic electron lenses using multiobjective genetic algorithms", *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, Volume 39, 2021.

5.1 Introduction

Designing electrostatic electron optical systems manually can become quite challenging when the number of electrodes increases. A large number of parameters is involved, such as the electrode thicknesses, the spacings between the electrodes, the aperture sizes of the electrodes and the electrode voltages. These can all be varied to optimize the performance of the optical system, while simultaneously taking typical constraints into account, such as the maximum allowable field between electrodes to prevent discharges. It was demonstrated [1] that the design of such systems with a single objective function, i.e. a single requirement that needs to be fulfilled for optimized performance, can be automated using optimization techniques based on genetic algorithms [2]. A six-electrode electrostatic lens system was optimized to focus a primary electron (PE) beam to the smallest possible spot in a fixed image plane. The objective function in this case was the spot size, consisting of contributions from the spherical aberration and the chromatic aberration of the lens. To determine the lens properties and the objective function, requires calculation of the lens field. This can be done most accurately using a finite element method (FEM), such as that offered by the commercial simulation package COMSOL. However, calculating the objective function using COMSOL is relatively slow (~1 min for each system evaluation). This makes the optimization extremely time consuming. A faster but less accurate method is to calculate the axial potential of the lens using the second order electrode method (SOEM) [3-6] (~1 sec for each system evaluation). In evolutionary algorithms (EA) such as GA, the objective function usually needs to be called numerous times in the optimization process, as many as hundreds or thousands of times. Using COMSOL then makes the optimization extremely time consuming.

A promising technique to reduce the computation time in such problems, is the use of surrogate models [7], also known as surrogates. These are computationally inexpensive approximation models, employed to assist in evaluation of computationally expensive functions to reduce the computational time of the problem. Studies on surrogate-assisted evolutionary algorithms such as GA began a decade ago. In the subsequent years, this method has been implemented in many different applications to optimize single and multi-objective optimization problems and found to be very efficient [8-10]. The surrogates (called the “Low Fidelity model”) can be implemented through different strategies combined with the accurate function

calculation (known as the “High Fidelity model”) to assist the optimization process [11]. In the previous study [1], a surrogate-assisted GA was used by first running a GA optimization using field calculation by SOEM (Low Fidelity), resulting in a number of approximately good systems that were then inserted into a subsequent GA optimization using field calculation by COMSOL (High Fidelity). With this approach an optimized design was obtained in a reasonable time.

In this work a similar optical system is considered, but now with two objective functions. Challenging and labour-intensive examples of such electron optical designs can be found in [12, 13], but the example to be used in this work is a five-electrode lens that focuses a PE beam in a fixed sample plane and simultaneously projects the secondary electrons (SE) emitted from the sample back through the lens onto a detector. This situation is typically encountered in electrostatic scanning electron microscopes with in-lens detection. To optimize this system requires a multi-objective optimization.

A recent work [14] has been published on the design of an ion optical device with many possible free parameters, involving multi-objective functions (up to two). There, an adjoint variable method is implemented. In the optimization based on the adjoint method, the computational time would stay nearly constant with increment of the free parameters, in contrast with the optimization using the evolutionary algorithms where it would increase dramatically. This gives an advantage over the evolutionary optimization technique for electron/ion design problems, where many free parameters are involved. However, the optimization based on the adjoint method is suitable for a few objective functions, while evolutionary optimization techniques such as GA are capable of optimizing multiple objective functions. Another difference is that the adjoint-based optimization is a local optimization technique (which can be used for global optimization problems by sampling different initial points), whereas the evolutionary algorithms are considered as global optimization techniques due to their meta-heuristic characteristics which can automatically search more spaces through the objective function landscape and prohibit trapping in local optima.

In the design of electrostatic lens systems, such as in our case-study, the objective function landscape (even in the case of a single objective function) appears to have many local minima [15]. In addition, multiple objective functions are usually aimed to be optimized, while the number of free parameters is not huge (in the order of tens). In such circumstances, EA/GA appears to be a better fit to the problem. Hence, a MOGA [16] is proposed to be used here. A surrogate model, similar to that in the

previous work [1], is then applied in the problem to assist MOGA. It is by no means obvious that the same two-step optimization strategy, combining SOEM and COMSOL, as was successfully used in the single objective function case, can be applied here too. The different energy and angular distributions of the PE and SE may cause the latter to follow trajectories considerably further away from the optical axis than the former. This means that the potential as approximated by the use of SOEM may not be sufficiently accurate to trace the SE towards the in-lens detector. Although a more accurate potential can be obtained using COMSOL, this will make the optimization really slow (one to two orders of magnitude slower, compared to optimization using SOEM), especially considering that a MOGA optimization scheme with two non-linear objective functions needs thousands of runs [15].

The objective of this work is to investigate whether the combination of an initial optimization using the less accurate SOEM and a subsequent optimization using COMSOL can also optimize systems with two objective functions in a reasonable time.

5.2 Lens system geometry and optimization parameters

A schematic drawing of the electrostatic electron probe-forming objective lens used in this study is presented in Figure 5.1. The lens, consisting of five planar electrodes, in combination with the sample positioned in a fixed image plane, is used to focus the PE beam, coming from the left, onto the sample plane. This geometry resembles an electrostatic scanning electron microscope design presented elsewhere [17-19], and some of the dimensions were taken from there.

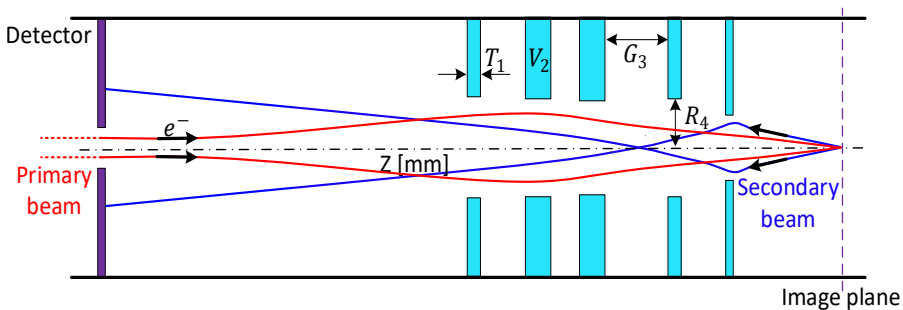


Figure 5.1. Cross-sectional schematic of a rotationally symmetrical 5-electrode lens system: The primary and secondary electrons pass through the same lens. T_i , R_i , and V_i , correspond to the thickness, radius and voltage at each electrode i , G_i indicates the gap between two sequential electrodes. There are 19 free variables in total.

The SE emitted from the sample by exposure to the PE beam are accelerated into the lens and directed towards the detector. The detector is a disk-shaped electrode (radius of 2.5 mm) with a central hole (radius of 0.5 mm) around the optical axis. The origin of the coordinate along the optical axis is taken at the surface of the first electrode, i.e. the surface closest to, and facing the detector. The detector plane is positioned at -23 mm and the sample plane is at +17 mm.

During the optimization process, the parameters that define the geometry of the lens, such as the thickness of the electrodes (T_i), the aperture radius of the electrodes (R_i) and the gap between electrodes (G_i) are allowed to vary within certain boundaries. The allowed intervals are $1 \text{ mm} < T_i \text{ (} i=1..5 \text{)} < 3 \text{ mm}$, $0.1 \text{ mm} < R_i \text{ (} i=1..5 \text{)} < 2 \text{ mm}$, and $1 \text{ mm} < G_i \text{ (} i=1..4 \text{)} < 3 \text{ mm}$. The voltage of the first electrode V_1 is allowed to vary from 6-8 kV whereas the other voltages V_i , ($i=2..5$) can range from 700 V to 10 kV. There are 19 free variables in total. The detector is kept at the same voltage as the first electrode V_1 . The voltage at the image plane is fixed to 600 V. Two constraints are set: a fixed image plane position ($X\text{-crossover}=17 \text{ mm}$) and a voltage breakdown condition (electric field $< 15 \text{ kV/mm}$).

5.3 The Objective functions

The first objective function is the probe size, and its calculation is described in detail in [1]. In brief, the probe size contains contributions from two axial aberrations only (spherical and chromatic aberration), and is given by [20]:

$$D_s^2 = (0.18 C_s \alpha^3)^2 + \left(0.6 C_c \alpha \frac{\Delta U}{U}\right)^2 \quad (5.1)$$

Where C_s and C_c are the spherical and chromatic aberration coefficient in the image plane, respectively, α is the half opening angle at the probe (here taken to be 10 mrad), U is the landing energy at the sample and ΔU is the energy spread in the PE beam (here taken as 1 eV). In eq. (1) the contributions from the geometric source image and diffraction are neglected. This is because, for very small probe currents, the contribution from the source image is negligible compared to those from the axial aberrations and the contribution from diffraction does not change during the optimization, since the landing energy and the half opening angle of the PE beam at the probe are fixed. The spherical and chromatic aberration coefficients can be calculated from aberration integrals, which contain the axial potential, its derivatives and a principle ray $r_a(z)$, starting in the object on the optical axis with unit angle (45

degrees) [21]. Furthermore, the magnification has to be taken into account, which is obtained by tracing a second principle ray $r_b(z)$ from the object plane, at unit height and zero angle, to the image plane. The position of the image plane is found where $r_a(z)$ crosses the optical axis.

The second objective function is the detector collection efficiency of the SE emitted from the sample. This requires tracing SE from the sample plane at $z=+17$ mm to the detector plane at $z=-23$ mm. The starting angles of the SE emerging from the sample are chosen from a uniform distribution of polar angles θ (the emission angle with respect to the sample surface normal) between 0.01 and 1.5 rad. The azimuthal angle is kept constant because of the rotational symmetry of the lens. However, in reality the angular distribution of the SE-yield is proportional to $\cos\theta$ [22, 23] and the SE are emitted within a solid angle $d\Omega = 2\pi \sin\theta d\theta$. Therefore, the detector signal is weighed by a factor of $\sin\theta \cos\theta \sim \sin 2\theta$. For simplicity, the energy of the emitted SE is taken from a uniform distribution between 1 and 10 eV.

5.4 Ray-tracing: SOEM versus COMSOL

The ray tracing and the determination of the lens properties requires calculation of the lens field. The SOEM approach [1,3] makes use of the fact that the solution to the Laplace equation can be expressed in terms of the axial potential and its derivatives with respect to z . By ignoring terms higher than second order and using a cubic spline approximation to the axial potential, the latter can be obtained by solving a set of linear equations. The results are accurate within the paraxial approximation and prone to deviations further from the optical axis [1,6]. COMSOL is an accurate method to calculate the potential in the space of the lens system. However, as this method meshes the entire lens space to calculate the potential using the finite element method, it is associated with long computation times [1]. Both approaches will be used to calculate the objective functions of the example system.

The ray tracing is done using a MATLAB code. For the PE, which follow small angle trajectories, the equation of motion in the paraxial approximation is used, as in [1]. However, for the SE which follow larger angle trajectories, a more accurate real ray tracing is performed. The equations of motion are [24]:

$$\frac{E_r q}{m} \frac{t^2}{2} + v_{0r}t + r_0 = r \quad (5.2)$$

$$\frac{E_z q}{m} \frac{t^2}{2} + v_{0z}t + z_0 = z \quad (5.3)$$

Here E_r and E_z are the radial and axial components of the field, respectively. The equations are solved numerically, by taking very small time-steps of Δt , starting from (r_0, z_0) , with initial velocity of v_{0r} and v_{0z} (calculated based on the initial energy of the SE).

In this section the effect is studied, on both objective functions, of calculating the potential by SOEM, resp. by COMSOL. Its effect on the probe size objective function can be judged from the two principle rays $r_a(z)$ (i.e. the ray starting in the object on the optical axis with unit angle (45 degrees)) and $r_b(z)$ (i.e. the ray starting from the object plane at unit height and zero angle). Figure 5.2 shows, for a typical system within the range mentioned in section 5.2, both rays traced through the potential as calculated by SOEM, resp. COMSOL. It is seen that the principle rays calculated by SOEM have a slight deviation but reasonably good overlap, with respect to the ones calculated by COMSOL.

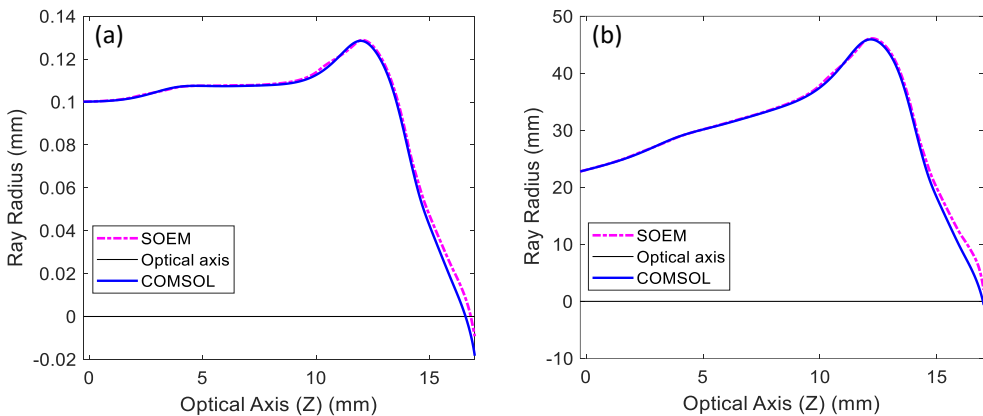


Figure 5.2. The ray-radius along the optical axis z resulting from the potential calculation by SOEM (pink dashed line) and COMSOL (blue solid line) for a) the principle ray $r_a(z)$, i.e. the ray starting in the object on the optical axis with unit angle (45 degrees), and b) the principle ray $r_b(z)$, i.e. the ray starting from the object plane at unit height and zero angle.

It is to be expected that the method of calculating the potential has a greater effect on the second objective function. For SE starting at the sample plane ($z=17$ mm) with angles ranging from 0.01 to 0.5 rad, and with an energy of 1 eV, the trajectories are calculated through fields calculated by SOEM and COMSOL to the detector at $z=-23$ mm. The results are shown in Figure 5.3, where in Figure 5.3a the SOEM-calculated

trajectories are shown as dashed lines and the COMSOL-calculated trajectories as solid lines.

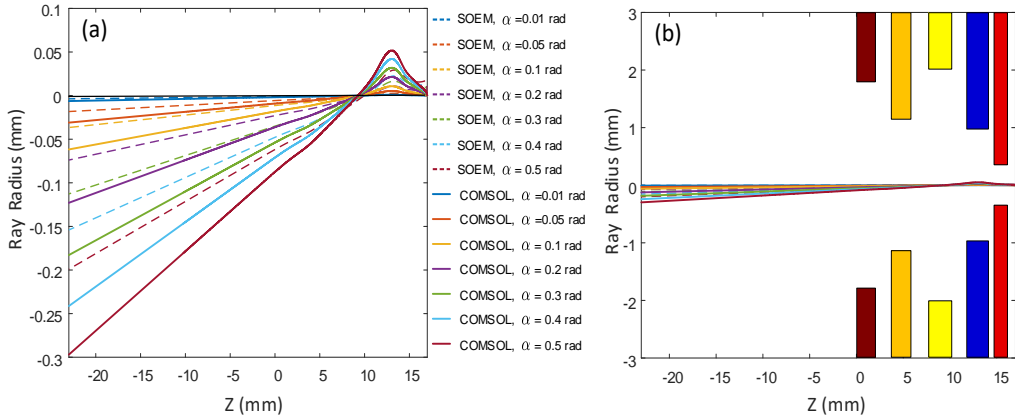


Figure 5.3. a) Trajectories of SE, emitted at various angles (colour scale) from the sample at $z=17\text{mm}$ to the detector at $z=-23\text{ mm}$, obtained using potential calculation by SOEM (dashed lines) and COMSOL (solid lines). b) The same trajectories as in a) on a larger vertical scale, including also the lens geometry. The colours indicate the potentials applied to the electrodes in this system (red: 6000 V, blue: 1094 V, yellow: 4370 V, orange: 4700 V, brown: 6600V).

Figure 5.3b shows the same rays on a larger vertical scale including the lens geometry. Although the SOEM-rays clearly deviate from the COMSOL-rays, they are not dramatically far off. Considering that the required computation time for SOEM-rays is around 1 second and for COMSOL-rays around 60-70 seconds, it may be worthwhile in the optimization of lens systems to initially use SOEM for the potential calculation, thereby creating some approximately good systems, and feed those to a more accurate COMSOL-based multi-objective function genetic algorithm (MOGA) optimization. This is the subject of the next section.

5.5 Optimization using MOGA

The Genetic Algorithm (GA), categorized as one of the evolutionary algorithms, mimics natural evolution (inspired by Darwin's theory). It starts with a randomly generated initial population which includes a set of systems, denoted by $P_1(x_1, \dots, x_n)$. In natural evolution, ' x_i ' represent the chromosomes of different members. In electron lens system optimization, ' x_i ' represents the electron lens systems. Hence, P_1 includes a set of initially randomly generated electron lens systems, defined based on the lens

system variables. In nature, across time, the initial population is gradually improved towards members which are better matched with their environmental conditions. In GA, analogously, the initial population evolves toward a new set of systems ($P_{i+1}(x_1, \dots, x_n)$) which better satisfy the conditions of the problem in each so-called "Generation". The new population is mainly created by the operators "Selection", "Crossover" and "Mutation". Conditions which are determined to be optimized are formulated by a so-called objective function.

The multi-objective GA (MOGA) [16] differs from a classical GA in how the objective function value is assigned and ranked to each member in the population. The initial population is created randomly, similar to that for the GA. The next generations are derived using the non-dominated ranking [25]. Each member gets a non-dominated rank corresponding to its relative objective function. The members of the population are then evaluated, ranked, sorted and selected by the MOGA based on the non-dominated classification. A set of Pareto fronts [25] (the non-dominated solutions for which none of the objective functions can be improved in value without degrading the values of other objective functions) are found and presented to the user at the end of the optimization process. The user has then the choice to select the system which best suits their problem among them. The remaining features of the algorithm, such as "Crossover" and "Mutation" are the same as those in a classical GA.

The proposed algorithms here is run on a PC with an Intel (R) Xeon (R) W-2123 CPU @3.60 GHz and 32 GB of RAM.

5.5.1 MOGA using potential calculation with SOEM

It will first be analyzed how MOGA optimization of the example lens system performs when SOEM is used for the potential calculation (hereafter called MOGA+SOEM). The parameters used for the MOGA in this case-study are population size = 100, maximum generations = 1000 and crossover fraction = 0.5. The optimization starts with an initial population of 100 systems randomly created by MOGA. Multiple runs are performed from which two example runs are presented in the Figures 5.4a and b. The figures show for a typical run the SE detection efficiency versus the probe size. In a relatively short time of 20 hours, MOGA evaluated 100,000 systems from which 30,000-40,000 systems passed the constraints. Hereafter these systems are referred to as good-systems.

In Figures 5.4a and 5.4b, which do not include all systems but only the good-systems, a trend towards decreasing probe size and increasing detection efficiency is seen. The

Pareto front [16] points for the two runs are illustrated as the dark blue and cyan points. By MOGA optimization of the voltages and the geometry of the lens the initial probe size of 60 nm is reduced to only 9 nm with a detection efficiency ranging from 0 % to 49 %. The system with a probe size of 18 nm and a detection efficiency of 49 % is indicated by 'L' in Figure 5.4a, the system with 9 nm probe size and 14 % detection efficiency by 'K' in Figure 4b. It takes on average ~200 min to reach the Pareto front systems in this optimization.

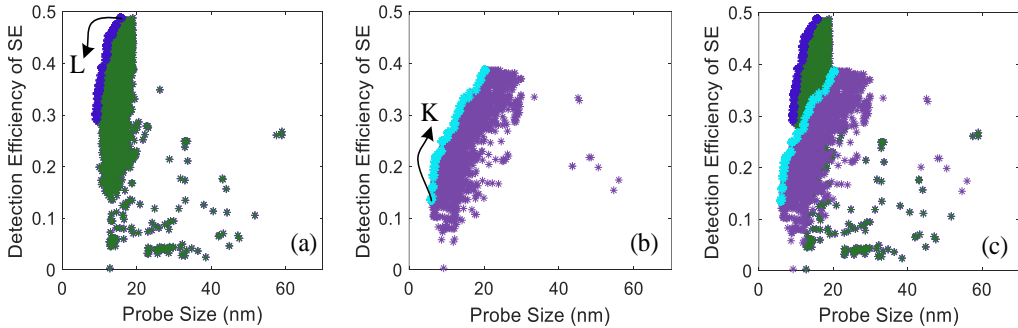


Figure 5.4. SE detection efficiency versus probe size for good-systems obtained by MOGA optimization based on potential calculation by SOEM. The optimizations tends towards smaller probes and larger detection efficiencies. Two different runs are shown in green (a) and purple (b). The Pareto fronts are shown as dark blue and cyan points in (a) and (b), respectively. The points L and K are explained in the text. (c) The data from a) and b) overlapped, including both Pareto fronts, illustrating how different runs may lead to different optimization results.

Figure 5.4c shows Figures 5.4a and 5.4b in one graph. It is clearly seen that for different runs the Pareto fronts vary considerably. This reveals the complexity of the (highly non-linear) objective functions and the existence of multiple local minima. The large number of free parameters in the design problem can easily cause the optimization to reach a different local optimum (Pareto front) in different runs.

5.5.2 MOGA using potential calculation with COMSOL

Next, a MOGA optimization is performed where for the calculation of the objective functions the necessary lens potential is calculated using COMSOL. Hereafter, we will call this method MOGA+COMSOL. All systems in the initial population are created randomly by MOGA itself. Since the computation time for the objective function calculation by COMSOL is dramatically longer, the population size is taken here to be 50 instead of 100, and the maximum generations as 200 instead of 1000. The crossover fraction is kept the same as before, i.e. 0.5.

In total, 10,000 systems have been evaluated and the entire optimization took about 7000 minutes (~5 days). Of these 10,000 systems, 3,000 - 4,000 systems passed the constraints and are thus labeled as good-systems. Figures 5.5a and 5.5b show the detection efficiency versus probe size for the good-systems of two separate runs. The Pareto front points of these two runs are shown as the black and orange points. Figure 5.5c shows the overlap of the plots in Figure 5.5a and 5.5b.

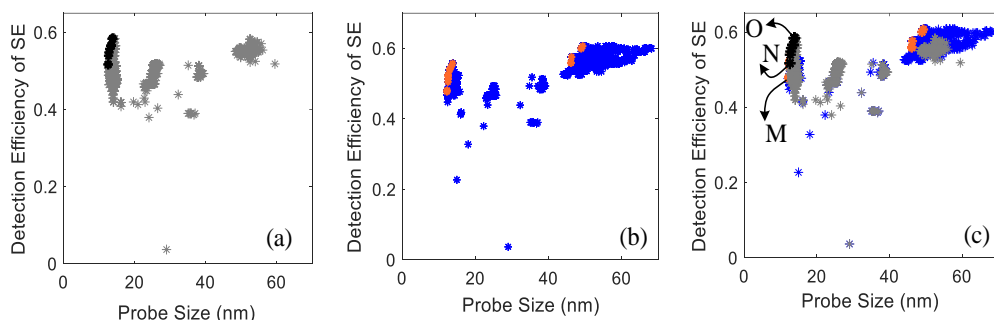


Figure 5.5. SE detection efficiency versus probe size for good-systems obtained by MOGA optimization based on potential calculation by COMSOL. The optimization tends towards smaller probes and larger detection efficiencies. Two different runs are shown in grey (a) and dark blue (b). The Pareto fronts are shown as black and orange points in (a) and (b), respectively. (c) The data from a) and b) overlapped, including both Pareto fronts, illustrating how different runs may lead to different optimization results. The points O, N and M are explained in the text.

Table 5.1 summarizes the optical parameters for three different systems in the Pareto-front of these MOGA+COMSOL optimizations. As can be seen from Table 5.1, it takes at least ~900 minutes of optimization time to reach the Pareto-front system with a probe size of 17.5 nm with an SE detection efficiency $DE_{SE} = 57\%$ (system O in Figure 5c) or ~2000 minutes to reach a system with spot size $D_{spot} = 13.0$ nm and $DE_{SE} = 52\%$ (system N in Figure 5.5c). The smallest spot size reached in this optimization, was $D_{spot} = 12.3$ nm with $DE_{SE} = 48\%$ nm (system M in Figure 5.5c), which took ~4000 min. It can be concluded that the computational time of MOGA+COMSOL to reach a satisfactory result is extremely long. In the next sub-section the use of good-systems from MOGA+SOEM as an input for MOGA+COMSOL is considered as a means to reduce the optimization time.

Multiple Criteria Optimization of Electrostatic Electron Lenses Using Multi-Objective Genetic Algorithms

Table 5.1. The optical parameters of three examples of non-dominated optimized systems (taken from the Pareto-front in Figure 5c) found by MOGA+COMSOL.

Example systems from the Pareto-front found by MOGA+COMSOL	System M	System N	System O
$X_{\text{crossover}}$	17.0±0.1 mm	17.0±0.1 mm	17.0±0.1 mm
Probe size (D_{spot}) (nm)	12.3 nm	13.0 nm	17.5 nm
Detection Efficiency (DE_{SE})	48 %	52 %	57 %
Computational time on average to find such system	~ 4000 min	~ 2000 min	~ 900 min

5.5.3 MOGA using potential calculation with COMSOL and MOGA+SOEM-optimized input systems

In an attempt to reduce the computational times involved in MOGA+COMSOL optimization, as initial input systems with pre-optimized geometries and voltages obtained by MOGA+SOEM were fed into MOGA+COMSOL. Hereafter, this will be called MOGA+SOEM+COMSOL. 10,000 systems with the same optimization parameters as used in MOGA+COMSOL (population size= 50, maximum generations= 200 and crossover fraction= 0.5) were evaluated. 20 systems are taken from Pareto front systems found by MOGA+SOEM and fed into the initial population, together with 30 systems created randomly by MOGA. The MOGA starts with these 50 initial systems while calculating the objective functions based on potential calculation by COMSOL. This is schematically shown as a flow-chart in Figure 5.6.

The optimization results for two runs are plotted as SE detection efficiency versus probe size in the Figures 5.7a and b. The Pareto front systems are indicated as well, by the red and green stars in Figure 5.7a and 5.7b, respectively. The results from both runs are overlapped in Figure 5.7c. Table 5.2 presents the optical parameters of three nondominated optimized systems taken as representatives from the Pareto front systems. It is seen that after only ~100 minutes, the optimization found an optimized system having a probe size of 17.5 nm and a very high detection efficiency of 85 % (system R in Figure 5.7c), or a system with a probe size of 12.2 nm and a detection efficiency of 81 % (system Q in Figure 5.7c), found after ~300 minutes. After ~2500 minutes, this optimization reached a system with the very small spot size of 8.6 nm and detection efficiency of 73 % (system P in Figure 5.7c).

It is noted that the time consumed in MOGA+SOEM to generate the initial data (~200 min) should be added to the computational time of this optimization.

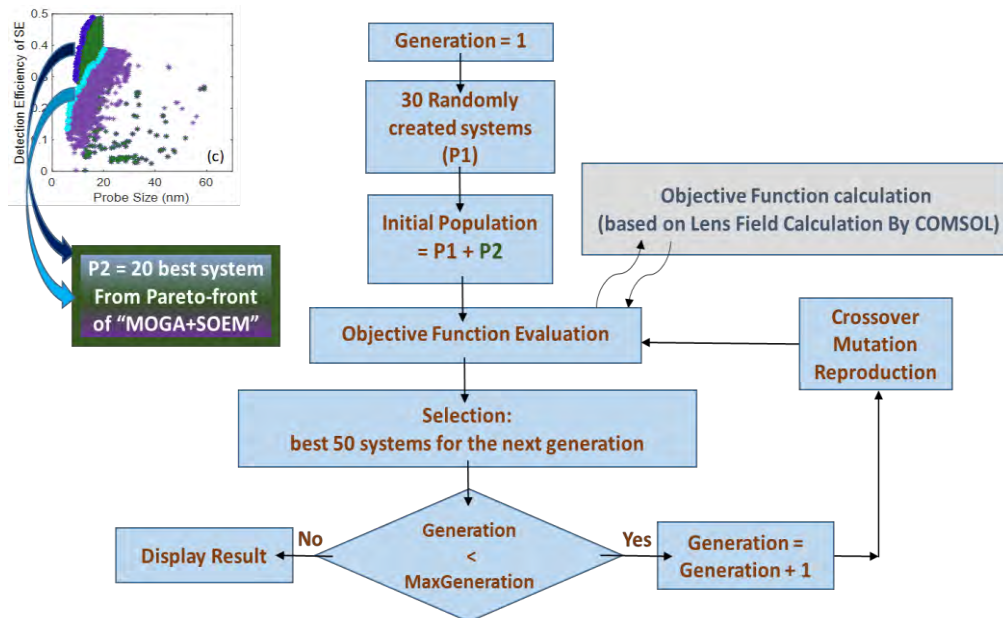


Figure 5.6. Schematic of the flowchart of MOGA+SOEM+COMSOL.

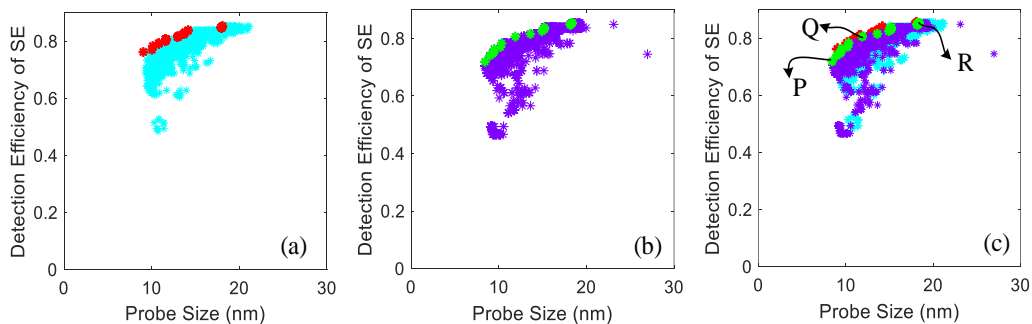


Figure 5.7. SE detection efficiency versus probe size for good-systems obtained by MOGA optimization based on potential calculation by COMSOL, but with feeding in 20 initial systems obtained from MOGA+SOEM optimization. Two different runs are shown in cyan (a) and purple (b). The Pareto fronts are shown as red and green points in (a) and (b), respectively. (c) The data from a) and b) overlapped, including both Pareto fronts. The points P, Q and R are explained in the text.

Multiple Criteria Optimization of Electrostatic Electron Lenses Using Multi-Objective Genetic Algorithms

Table 5.2. The optical parameters of three examples of non-dominated optimized systems (taken from the Pareto front in Figure 7c) found by MOGA+SOEM+COMSOL.

Examples of systems from the Pareto front found by MOGA+SOEM+COMSOL	System P	System Q	System R
$\chi_{\text{crossover}}$	17.0±0.1 mm	17.0±0.1 mm	17.0±0.1 mm
Probe Size (D_{spot})	8.6 nm	12.2 nm	17.5 nm
Detection Efficiency (DE_{SE})	73 %	81 %	85 %
Computational time on average to find such system	~ 2500 min	~ 300 min	~ 100 min

5.6 MOGA+SOEM+COMSOL versus MOGA+COMSOL

Comparing the data in table 5.2 and table 5.1, the MOGA+SOEM+COMSOL optimization achieved much better systems in a considerably shorter time than the MOGA+COMSOL optimization. For instance, reaching a system with a spot size of 17.5 nm only takes 100 min in the former case and 900 min in the latter, even with a dramatically higher SE detection efficiency. It should be noted that the minimum spot size of 8.6 nm with a detection efficiency of 85 %, reached in the time frame of ~300-2500 min, could not be reached at all by the MOGA+COMSOL optimization, even after evaluation of 10,000 system (~7000 minutes).

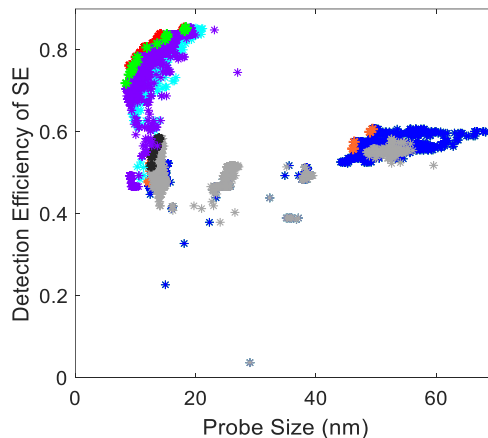


Figure 5.8. Comparison of the MOGA+COMSOL results of Figure 5.5c (2 runs in gray and dark blue symbols; the Pareto front points are shown in black and orange) and MOGA+SOEM+COMSOL results of Figure 5.7c (2 runs in cyan and purple symbols; the Pareto front points are shown in red and green).

In Figure 5.8 the results from Figure 5.5c and Figure 5.7c are compared in one graph, i.e. a comparison between optimized systems obtained by MOGA+COMSOL and MOGA+SOEM+COMSOL, resp., including the Pareto front systems. It is noted that the runs in the latter case show much more overlap than what was seen in the former case. Furthermore, The systems optimized by MOGA+SOEM+COMSOL clearly outperform the ones optimized by MOGA+COMSOL. Hence, it is concluded that, despite the deviating field calculation by SOEM, it is still very effective to use this technique in the optimization, to find preliminary, approximately good-systems, to be fed into another optimization, which is performed based on accurate field calculation by COMSOL, to boost the optimization.

5.7 Examples of systems optimized by MOGA+SOEM+COMSOL

The evolution of some lens systems during the optimization by MOGA+SOEM+COMSOL are summarized in Figure 5.9. A clear change is observed in the geometries and voltages of the systems while the optimization is progressing until it finally reaches a set of optimized systems. It should be noted that those SE hitting the electrodes, do not contribute to the detection efficiency. The PE are started at $z = -23$ mm, with angles ranging between 0.1 and 0.8 rad, and traced by paraxial ray-tracing. The horizontal axis runs from -10 to +17 mm, for a better visualization. SE trajectories are produced by real ray-tracing. In the Figure 5.9, these are drawn for visualization purposes only. To validate the final optimization results, an optimized system found by MOGA +SOEM+COMSOL (system A4 in Figure 5.9) is simulated using the EOD software package from SPOC [26]. Figure 5.10 shows the EOD simulation result for the geometry and the trajectories of both PE and SE in this system. The optical parameters of system A4, as obtained from MOGA +SOEM+COMSOL and EOD, are compared in table 5.3. The presented results show that the optimization result agrees very well with the EOD simulation.

Table 5.3. A comparison of the optical parameters of system A4 from Figure 9, as obtained by MOGA+SOEM+COMSOL and EOD.

	MOGA+SOEM+COMSOL	EOD
System A4	$X_{\text{crossover}} = 16.99$ mm Probe size = 11.91 nm Detection Efficiency = 80%	$X_{\text{crossover}} = 16.99$ mm Probe size = 11.90 nm Detection Efficiency = 81%

Multiple Criteria Optimization of Electrostatic Electron Lenses Using Multi-Objective Genetic Algorithms

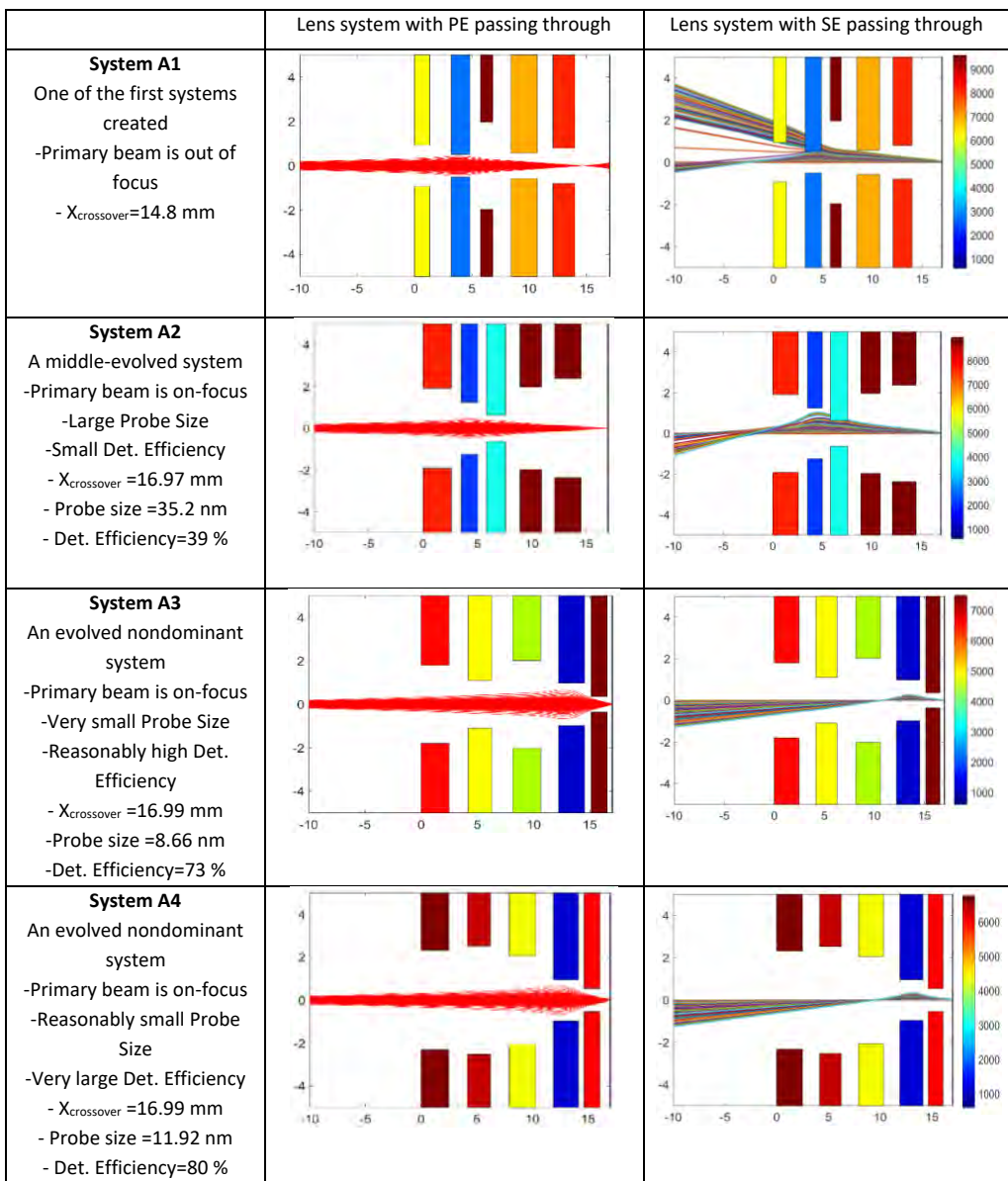


Figure 5.9. System improvement by MOGA+SOEM+COMSOL optimization, while trying to find optimum values for the two objective functions and satisfying the constraints. System A1: constraint not satisfied. System A2: constraint satisfied, very low SE detection efficiency (DE_{SE}) and relatively high spot size (D_{spot}). System A3: constraint satisfied, very small D_{spot} and relatively low DE_{SE} . System A4: constraint satisfied, very high DE_{SE} and reasonably small D_{spot} . The color scale in the middle column indicates the electrode potential values in V. The units along the axes of the graphs are mms.

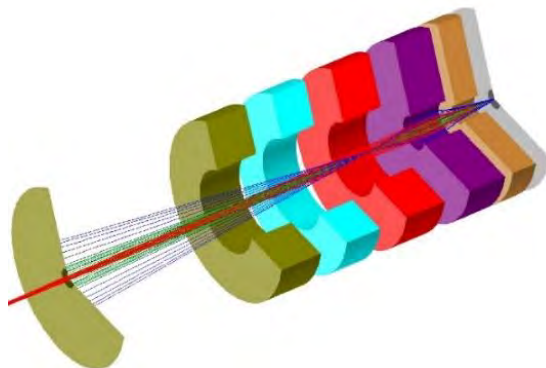


Figure 5.10. An optimized system found by MOGA+SOEM+COMSOL (system A4, in Figure 5.9), simulated using EOD. PE trajectories are shown in red. Only SE trajectories with initial energy of 1 eV and 10 eV are shown, in blue and green, respectively.

5.8 Conclusion

The design of a pure electrostatic lens system with 5 electrodes and two objective functions was optimized using a surrogate-assisted multi-objective genetic algorithm (MOGA) approach. The system is required to focus a PE beam into a fixed image plane with minimum probe size and attract SE emitted from the image plane back into the lens on to an SE detector with the highest possible detection efficiency. The main challenge of such optimization is the long computation time necessary for an accurate COMSOL-based field calculation. The much faster, but less-accurate, field calculation technique SOEM combined with MOGA does not provide the best optimized systems, but reasonably good ones that can be used as input for a COMSOL-based MOGA optimization. This strategy has been demonstrated to lead to optimized systems in a reasonably short time of a few hours, compared to many days when using COMSOL-based MOGA only. The optimization process is fully automated and may help electron optics designers to quicker find solutions for complex electrostatic systems.

References

- [1] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C. Hagen and P. Kruit, *Int. J. Mod. Phys. A*, Vol 34, 1942020 (2019).
- [2] M. Mitchell, *An Introduction to Genetic Algorithms*, (MIT Press, Cambridge, London, 1998).
- [3] J.P. Adriaanse, H.W.G. van der Steen, and J.E. Barth, *J. Vac. Sci. Technol.*, B 7, pp. 651-666 (1989).
- [4] M.A.J. Van der Stam, J.E. Barth, and P. Kruit., *Proc. SPIE Int. Soc. Opt. Eng.*, Vol. 2014, pp.45-56 (1993).
- [5] J.E. Barth, H.W.G. van der Steen and J. Chmelik, *Proc. SPIE Int. Soc. Opt. Eng.*, Vol. 2522, pp. 128-137 (1995).
- [6] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, T. Janssen, C. Hagen and P. Kruit, *Recent Trends in Charged Particle Optics and Surface Physics Instrumentation*, pp. 24–26 (2018).
- [7] B. Liu, Q. Zhang and G. Gielen, *Simulation-Driven Modeling and Optimization*, Springer Proceedings in Mathematics & Statistics Vol. 153, edited by S. Koziel, L. Leifsson, X. Yang, (Springer, Cham, August 2014).
- [8] C. Xiwen, G. Liang, and L. Xinyu, *IEEE Trans. Evol. Comput.*, Vol. 24, No. 2, April (2020).
- [9] A. D'íaz-Manr'iquez, G. Toscano-Pulido and W. Gomez-Flores, 2011 IEEE C. Evol. Computat. (CEC), pp. 2155-2162 (2011).
- [10] Y. Jin, *Swarm Evol. Comput.*, Vol. 1, Issue 2, pp. 61-70 (2011).
- [11] B. Liu, S. Koziel, and Q. Zhang, *Int. J. Comput. Sci. Eng.*, 12, pp. 28-37 (2016).
- [12] M.A.R. Krielaart and P. Kruit, *Ultramicroscopy*, 220 (2021).
- [13] A. Mohammadi-Gheidari and P. Kruit, *Nucl. Instrum. Methods Phys. Res., Sect. A*, Vol. 645, pp. 60-67 (2011).
- [14] L.T. Neustock, P.C. Hansen, Z.E. Russell, and L. Hesselink, *Sci. Rep.* 9(1), pp.1-12 (2019).
- [15] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, C. W. Hagen, and P. Kruit, 2020 IEEE 6th International Conference on Optimization and Applications, (ICOA) (2020).
- [16] T. Murata, and H. Ishibuchi, *Proc. of 1995 IEEE International Conference on Evolutionary Computation (ICEC)*, pp. 289-294, Perth, Australia (1995).
- [17] M.P.C.M. Krijn, and J.M. Krans, United States Patent Application Publication, Patent No. US 2002/0079449 A1, June. 27 (2002).

- [18] J.M. Krans, S.G. Den Hartog, and M.P.C.M. Krijn, United States Patent Application Publication, Patent No. US 2002/0109089 A1, Aug. 15 (2002).
- [19] J.M. Krans, United States Patent, Patent No. US 6,646,261 B2, Nov. 11 (2003).
- [20] J.E. Barth, and P. Kruit, "Addition of different contributions to the charged particle probe size", *Optik101*, pp. 101-109 (1996).
- [21] Miklos Szilagy, *Electron and Ion Optics*, Plenum Press, New York (1988).
- [22] W. Oppel, and H. Jahrreiss: "Messungen der Winkelverteilung von Sekundar-elektronen an dunnen freitragenden Al- und Au-Schichten", *Z. Physik* 252, 107 (1972).
- [23] L. Reimer, "Scanning Electron Microscopy, Physics of Image Formation and Microanalysis", *Springer Series in Optical Sciences*, Vol. 45 (1998).
- [24] V.K. Zworykin, J. Hillier, G.A. Morton, E.G. Ramberg, and A.W.Vance, *Electron Optics and the Electron Microscope*, John Wiley and Sons, Inc. (1945).
- [25] W. Pinheiro dos Santos, "Evolutionary Computation", Chapter 26: Adaptive Formation of Pareto Front in Evolutionary Multi-objective Optimization, (I-Tech, Vienna, Austria 2009).
- [26] B. Lencova, and J. Zlamal, *Electron Optical Design program package EOD 5.001*

6. Local versus Global Optimization of Electron Lens System Design

Abstract

In electron optics, the design of electron lens systems is still a challenge. To optimize such systems, the objective function which should be calculated, depends on the electric potential distribution in the space created by the lenses. To obtain the electric potential, the existing methods are generally based on some mathematical techniques which need to mesh the space of the lens system and derive the electric potential at all mesh points. Hence, calculation of the objective function for such systems is computationally expensive. Therefore, applying a fully automatic optimization routine has not yet been feasible, especially for lens systems with many free variables. Hence, the study of the objective-function landscape of such problems has not yet been performed.

One of the questions of interest for optical designers, that has not been studied in the literature, is whether this problem can be solved by a local optimizer or is it necessary to apply a global optimizer. Recently we succeeded in implementing a method (based on a so-called SOEM (Second Order Electrode Method) technique) which calculates the electric potential in a fast and reasonably accurate way. In this paper, that method, is implemented to perform the study of local versus global optimization for electron lens design. The global optimization method here is performed by GA (Genetic Algorithm). The objective function is taken to be the probe size of the electron beams at the image plane. The results of our study show that the objective function of this problem has many local minima and the optimization of such problems cannot be handled by a local optimizer. GA is shown to perform well by overcoming these multiple-local minima to arrive at a global minimum.

This chapter is based on the following publication:

Neda Hesam Mahmoudi Nezhad, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "Local versus global optimization of electron lens system design", IEEE 6th International Conference on Optimization and Applications (ICOA), 2020.

Keywords

Electron Lens Design, Global Optimization, Local Optimization, Genetic Algorithms, SOEM (Second Order Electrode Method)

6.1 Introduction

Optimization of multi electrode lens systems, is still a challenging task for electron-optical designers. The challenge is due to the fact that the objective function for such problems (i.e. spot size) cannot be analytically formulated. The existing methods that numerically calculate the objective function, are based on methods such as finite element method, boundary element method, or finite difference method that all need to mesh the space of electrode lens systems. Calculation of optical parameters on all these mesh points is then needed to derive the objective function for each system. Using such calculation intensive methods in an optimization routine significantly increases the optimization time. It causes running a fully-automatic optimization routine to be not easily feasible, especially when many free parameters are involved.

Due to the problem mentioned above, to our knowledge, there is not yet a fully automated optimization routine which can handle the optimization of a multi electrode lens system, having all its geometry and voltages as free parameters. Therefore, studying the objective function landscape of such problems, even the question of whether the optimization of multi-electrode lens systems is a global or local optimization problem, has still not been performed and answered in the literature. This topic is of interest to electron optical designers and can be very useful in speeding up the optimization process in electron lens design.

Recently we have introduced and implemented successfully a fast method to calculate the electric potential (hereafter we call it potential), and therefore the objective function, in electron-lens design problems [1] (based on a so-called Second Order Electrode Method (SOEM) method, first proposed by Adriaanse and Barth in 1989 [2-4]). Once we had developed such an automated and fast routine, we decided to use that to perform the above mentioned study on the electron lens design. The main intention of this work is to investigate the objective function landscape of a multi-electrode lens design system, and whether or not it has many local minima, and if this problem can be handled by a local optimizer or not.

The objective function (spot size) calculated by SOEM has some deviation from its precise value [5], which could be derived using Finite Element Method (FEM), however, it is shown in our previous work [1], that the trend of decrement of spot size that resulted from optimization based on SOEM is almost the same as the one calculated accurately based on FEM (using COMSOL Multi-physics software, version 5.3a). The formulation of objective function calculation (derivation of spot size from the potential)

in SOEM, mathematically, is also similar to the one calculated by FEM. The main intention of this work was not to run the optimization to accurately ascertain the best optimized system, but simply to study the local versus global optimization, so this can be performed by the fast, approximate SOEM method. Therefore, all optimization for this study is performed based on SOEM. Genetic Algorithm (GA) [6] is used as a global optimizer to make this study.

This optimization problem also has a constraint. In order to apply the constraint two approaches are taken. The outcome of these approaches are also analyzed and illustrated in this paper. The chapter is structured as follows. In section 6.2, Optimization parameters, namely objective function, free variables, bounds and constraints for multi-electrode lens optimization problems, are described. Implementation of local and global optimization in MATLAB (Matrix Laboratory, version r2016 b), as well as constraint implementation are addressed in section 6.3. The study of local versus global optimization is described and analyzed in section 6.4. Section 6.5 contains the conclusion of this study.

6.2 Optimization Parameters

6.2.1 Objective Function

In electron-optical imaging systems, the main application of electrode lenses is bending and focusing a bunch of electron beams by electric fields for the purpose of image formation. The better the electrode-lens system, the higher the resolution of the image. To achieve an image with higher resolution, a so-called optical parameter of “spot size” (the cross section area of a bunch of electron beams passing through the system, at the image plane), should be minimized. Thereby, this optimization problem is to obtain the system with the smallest spot size at the image side. So, the spot size (at the image side) is taken as the objective function.

The spot size, assuming the lens only suffers from axial aberrations such as chromatic and spherical aberrations, can be estimated as [7]:

$$D_s^2 = (0.18 C_s \alpha^3)^2 + (0.6 C_c \alpha \frac{\Delta U}{U})^2 \quad (6.1)$$

where, D_s stands for spot size, C_s and C_c are spherical and chromatic aberration coefficients, respectively. This estimation is presumed to be valid for our case-study. α is the half opening angle of the beam (chosen to be 10 milliradian in this case-study), ΔU is the energy spread of the electron source (taken as 1eV) and U is the acceleration energy (equal to the potential at the image plane = 1kV), these values are typical for

low voltage electron beam systems. To get the spot size (D_s) at the image side, all parameters mentioned in equation 6.1 should be calculated at the image side. Here, first C_s and C_c are calculated at the object side, then using the magnification (M), they are converted to their correspondent values at the image side.

C_c and C_s at the object side, can be calculated as a function of axial potential (φ), its first and second derivatives (φ' and φ''), and the imaging principle ray of $r_\alpha(z)$ starting on-axis, from the object side, with a slope of 1 as [8]:

$$C_s = \frac{1}{16\varphi_0^{1/2}} \int_{z_0}^{z_i} \varphi^{1/2} \left\{ \frac{5}{4} \left(\frac{\varphi''}{\varphi} \right)^2 + \frac{5}{24} \left(\frac{\varphi'}{\varphi} \right)^4 \right\} r_\alpha^4 + \frac{14}{3} \left(\frac{\varphi'}{\varphi} \right)^3 r_\alpha^3 r_\alpha' - \frac{3}{2} \left(\frac{\varphi'}{\varphi} \right)^2 r_\alpha^2 r_\alpha'^2 \Bigg\} dz \quad (6.2)$$

$$C_c = -\varphi_0^{1/2} \int_{z_0}^{z_i} \left(\frac{3\varphi'^2}{8\varphi^2} \right) r_\alpha^2 dz \quad (6.3)$$

where φ_0 refers to the potential at the object side.

The above mentioned principle ray ($r_\alpha(z)$) is derived by ray tracing numerically, using the equation of motion of electrons, in the paraxial approximation [8]:

$$\frac{1+\varepsilon}{1+2\varepsilon} \varphi(z) \cdot r_\alpha''(z) + \frac{1}{2} \varphi'(z) \cdot r_\alpha'(z) + \frac{1}{4} \varphi''(z) \cdot r_\alpha(z) = 0 \quad (6.4)$$

The spot size at the image side is calculated by inserting C_s and C_c at the image side into equation 6.1. From now on, where the spot size is mentioned, it means the spot size at the image side.

Spot size, as shown above, can be derived from the axial potential and its derivatives. For multi electrode lenses, with many free parameters, to reach a satisfactory result, this function should be evaluated for thousands of systems within an optimization routine [1]. As the potential calculation part is the most time-consuming part of the objective function evaluation, a fast method of potential calculation for this study is needed. Therefore, SOEM has been applied for the potential calculation part of this research.

6.2.2 Free Variables and Bounds

To explain the optimization parameters such as free variables, bounds and constraints for multi-electrode lens systems more specifically, an example of a system of multi-electrode lenses with 6 lenses is taken as the case-study. This choice was made

arbitrarily, simply as an example which has enough complexity as a type of multi-electrode lens system. Changing it to other numbers of electrodes is straightforward. A schematic of this system in 3D is shown in Figure 6.1. As the system is rotationally symmetrical, it can be converted and solved as a 2D problem (Figure 6.2).

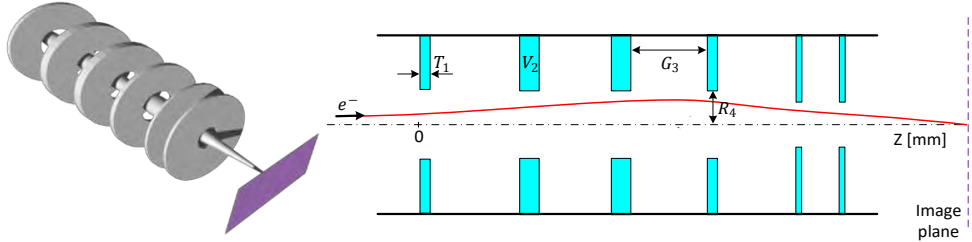


Figure 6.1. a. Schematic of the rotationally symmetrical electrostatic lens system including 6 electrodes in 3D. b. Schematic of the rotationally symmetrical electrostatic lens system including 6 electrodes in 2D. T_i , R_i , and V_i , correspond to thicknesses, radii and voltages at each electrode, G_i , are gaps between two sequential electrodes.

6.2.3 Constraints

The constraint is related to the image position. It is aimed to get the image at a fixed position X_c . For our case-study, this distance is taken to be 3.5 mm from the entrance plane of the first electrode (shown in Figure 6.3). A deviation up to a few micrometer ($\sim 3 \mu\text{m}$) is acceptable in our case-study. Therefore the constraint is determined as:

$$3.48 \text{ mm} < X_c < 3.52 \text{ mm} \tag{6.5}$$

6.3 Implementation in MATLAB

6.3.1 Applying Global and Local Optimizers

Having defined the optimization parameters, a global and local optimizer can now be implemented. Genetic Algorithm (GA) and a so-called “fmincon” local optimizer, in MATLAB, are used as the global and local optimizer, respectively. The choice of using GA as a global optimizer was based on our experience of its successful performance on optical lens system design [9] and recently in electron lens system design [1]. There are different optimization solvers which can be implemented as the local optimizer in MATLAB. Our optimization problem is a nonlinear problem. In MATLAB, “fmincon” is offered as a suitable optimizer for such a problem. This optimizer does not guarantee to find the global minima, but does guarantee to arrive at the local minima of the objective function-landscape’s valleys, dependent on the point where it is initiated. Therefore, this optimizer is used as a local optimizer for this case-study.

6.3.2 Constraint Implementation

Generally, an optimization program can be divided into two major parts. The function which generates systems (here we call it “Optimization function” and in this work this is performed by GA or “fmincon”), and the “objective function calculation” part. An optimization program with constraints includes the third additional part; the “constraint evaluation” function.

The “Optimization function” part generates systems starting with those randomly created in GA, whereas in “fmincon” it starts from an initially given system. Afterwards the process continues in GA by generating the new systems using GA parameters such as crossover, mutation, population, generation, etc. and ends based on its stopping criteria (that can be determined as the maximum number of generation, execution time, value of the objective function). In “fmincon” this process is performed based on the objective functions` gradient, controlled by several limiting parameters such as iterations, tolerance, etc. The boundaries of the problem (maximum and minimum ranges of the geometry and voltages of the systems) should also be determined in this part.

Based on our previous experience of using GA in optimization of such systems having constraints [1], also considering the fact that the intention of this investigation was not to get the best optimized system, but to achieve some results out of GA which can be used as comparison between a global and a local optimizer, the GA parameters are chosen to be: population = 50 and maximum generation = 100. Crossover and Mutation are selected to be 'crossoverarithmetic' and 'mutationadaptfeasible', respectively. Elitism is taken to be 1.

In a standard way, for the optimization routine with constraints, the generated systems are then imported to the second part, i.e. “constraint evaluation”, to check whether or not the constraints are satisfied. The third part, the “objective function calculation”, calculates and evaluates the objective function (spot size) for the systems which had already satisfied the constraints.

However, due to the nature of gradient-based “fmincon” and the possibility of having a non-smooth objective function, in order to ensure the constraints be well-satisfied, the constraints are applied within the objective evaluation part and not by implementing the constraint function as a separate function in MATLAB. To have both GA and “fmincon” run in a similar fashion, for the sake of having fair comparison, this strategy is applied to both. This approach is presented below, and schematically shown in Figure 6.3.

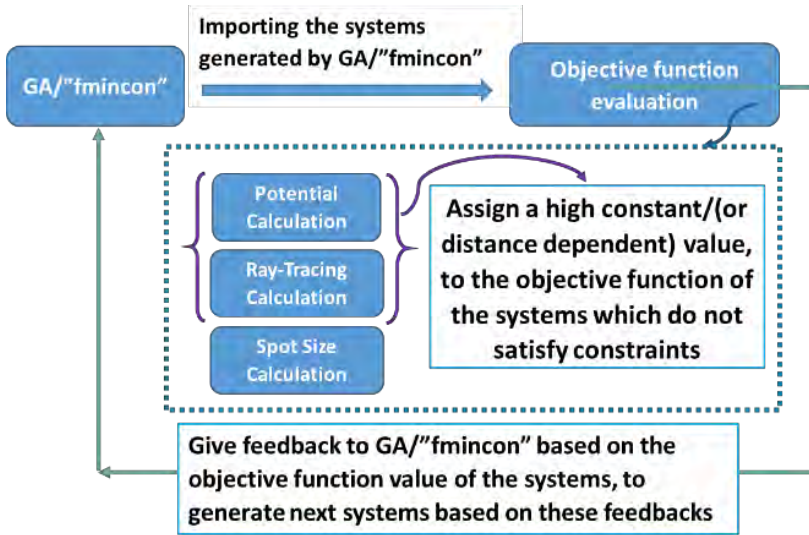


Figure 6.3. Schematic of the optimization routine, when the constraints are brought inside the body of objective function evaluation code.

After the first part (“optimization function”) generates the system/systems, they are evaluated for the constraint. If the constraint is not satisfied, a high constant value is assigned to the objective function, and the function ends, otherwise, it continues to calculate the real image plane position. Therefore, the systems which do not satisfy the constraint automatically would have a very small chance of being chosen as the parents to bring to the next generations for breeding the offspring in GA, or to be among the next selected systems in “fmincon”.

To choose the high constant value, it should be noted that this value should be high enough that it exceeds the real feasible values for the systems which are within the constraints. Otherwise, these systems could enter the selection process by their non-real low objective function values. As the spot size for the systems in which their image plane is within the ranges of $3.48 \text{ mm} \leq X_c \leq 3.52 \text{ mm}$, mostly is below 40 nm, taking a value of 100 can be a proper value. Therefore, this approach (here called approach A) is applied by assigning a constant value of 100 as the objective function value for systems which are not within the constraints:

Approach A:

If $3.48 \text{ mm} \leq X_c \leq 3.52 \text{ mm}$: Spot size = its real value

If $X_c > 3.52 \text{ mm}$: Spot size = 100

If $X_c < 3.48 \text{ mm}$: Spot size = 100

However, by this approach, all systems which are not within the ranges of the constraint, no matter how far they are out of the range, get a similar value of 100. It sounds more efficient to give a high value that depends on the distance of them from the border of the constraint. The spot size based on this approach, considering the value of maximum feasible spot size (i.e. = 40), is formulated as:

Approach B:

$$3.48 \text{ mm} \leq X_c \leq 3.52 \text{ mm} \quad : \text{ Spot size} = \text{real value}$$

$$\text{If } X_c > 3.52 \text{ mm} : \text{ Spot size} = 50 * (X_c - 3.52 + 1)$$

$$\text{If } X_c < 3.48 \text{ mm} : \text{ Spot size} = 50 * (3.48 - X_c + 1)$$

Choosing this format of formulation and taking value of 1 to the offset, and 50 for multiplication factor was arbitrary, simply to ensure that it takes a value above 40. Taking other values may speed up the optimization further and can be played with. However, for now this format and value, was working properly and efficient enough for this study.

To take the more efficient constraint implementation approach, these two methods (i.e. approach A and B) are tested and compared. As this comparison is performed only to select the better constraint implementation approach for the rest of the study, therefore, it was sufficient to apply only one of the optimization methods. GA is implemented for this search. Since GA, as a global optimizer, is assumed to cover systems having more variety with larger differences in their objective function values, it is better for such an evaluation and data analysis. The results of this comparison are discussed in the following section.

6.3.3 Comparison of two approaches on constraint implementation

As GA is a semi-heuristic search optimization technique, the optimizations have been run for each approach (approach A and B) 10 times to provide statistically more reliable data. Each time GA is run for 5000 systems (Population=50 and Generation=100). As representative of the 10 runs, the plots for three runs are presented in Figure 6.4. The first row (a, b and c) is related to approach A, the second row (d, e and f) corresponds to approach B.

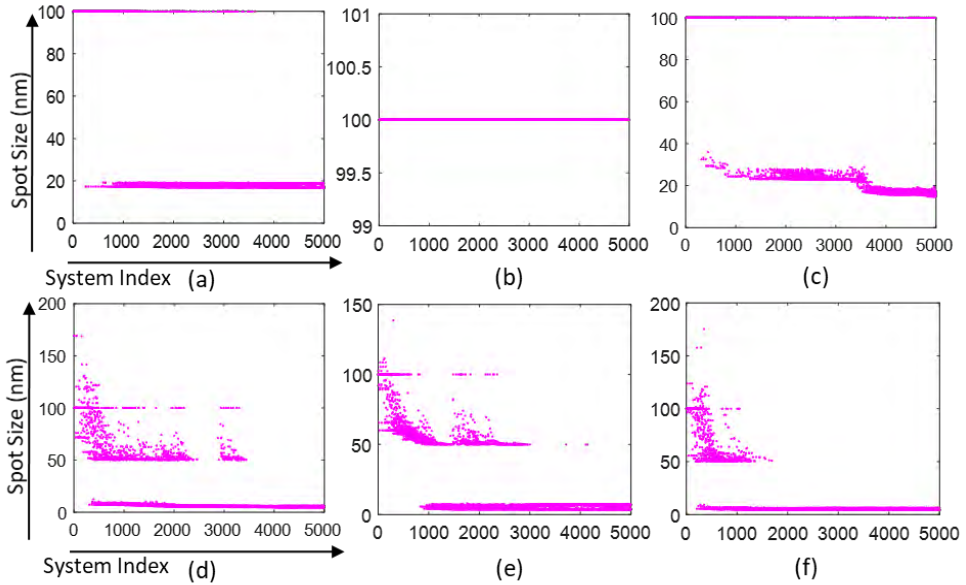


Figure 6.4. Three runs related to implementation of two different approaches A and B (the first row (i.e. (a), (b) and (c)) are results of approach A, the second row (i.e. (d), (e) and (f)) are related to approach B).

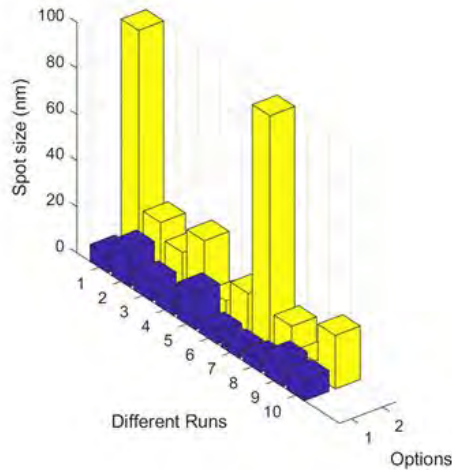


Figure 6.5. Result comparison from implementation of approach A and B (shown by options 2 and 1 in the figure, respectively). The plot represents the data related to the minimum value of the objective function which is found after 5000 system iterations for 10 different runs.

As can be seen at a glance from both Figure 6.4 and Figure 6.5, approach B could get better results. This has also been logically anticipated, as approach B endows the

optimization routine with greater intelligence compared to approach A. This is achieved by providing the optimization process with insights into the varying degrees of deterioration among discarded systems, unlike approach A, where all discarded systems are considered equally weighted. The minimum objective functions reached by approach B after 5000 iterations (shown by the blue columns in Figure 6.5) has in general, lower values compared to the ones from approach A (yellow columns). For some runs, GA could not even find a system which satisfies the constraint, within evaluation of 5000 systems (Figure 6.4(b) or first and 7th column in Figure 6.5).

Hence, in total, it is clear from the figure, that approach B could, on average, find a much better system. It can be concluded that, using a value depending on the distance from the border of the constraints (approach B) helps the optimization to get better results. Therefore approach B is taken as the constraint implementation approach for our problem.

6.4 local versus global optimization

By selecting the constraint implementation method, now our study of global versus local optimization can be performed. To conduct this study, the following steps are taken. First, the optimization routine using the GA for the selected specific number of generations and populations is executed. The constraint on image plane, is applied using approach B. Some points which were intermediate systems found by GA are taken and fed to “fmincon” as the initial points to start with. “fmincon” is executed from each point until it converges to a solution. This is shown schematically in Figure 6.6.

To analyze, the data from one of the GA runs (Figure 6.4(d)) out of 10 previous runs is taken as a sample. Figure 6.7 presents this sample. The top plot, includes the objective function data for the systems which are within the constraints, together with those which have not satisfied the constraints (the one in which their values are above 40 nm). The bottom plot shows the data only for those which satisfied the constraint, by omitting the systems which were not within constraint. The total number of systems to be evaluated with GA, is taken to be 5000 (population=50, generation=100).

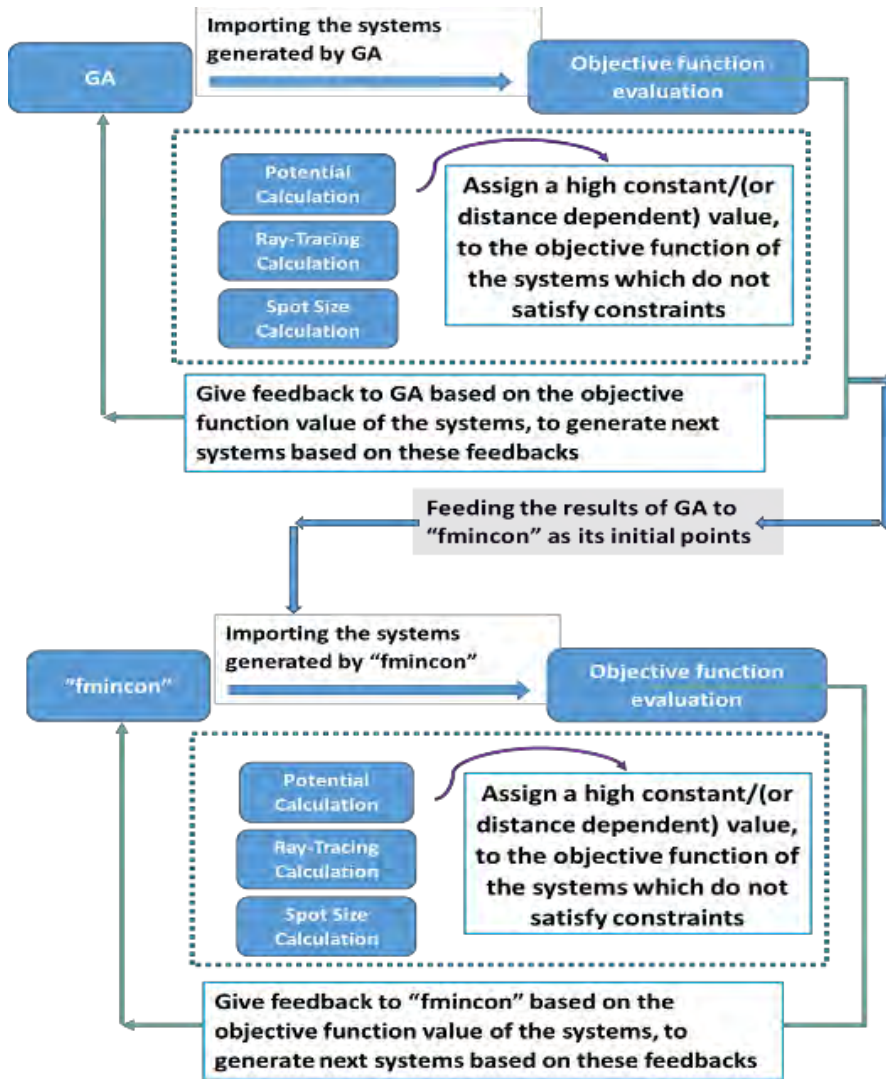


Figure 6.6. Schematic of optimization implementation for combined "fmincon" and GA.

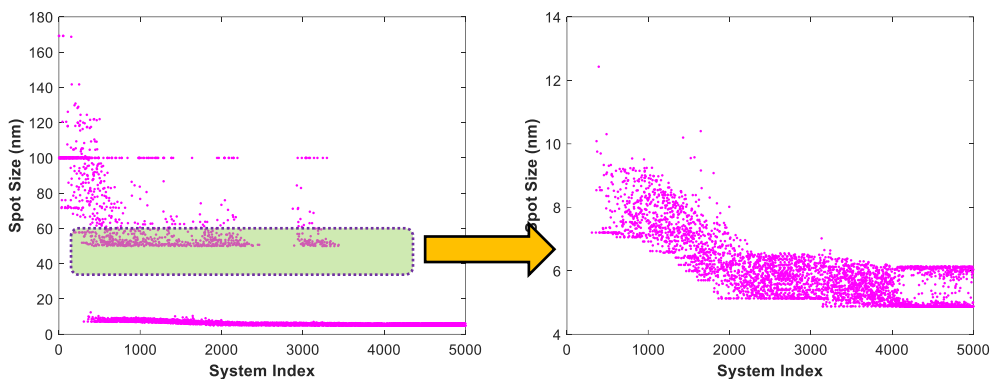


Figure 6.7. A run using GA with 5000 evaluated systems (left): spot size versus iterations, for all systems including the ones which have not satisfied the constraints, (right): spot size versus iterations only for the systems which had satisfied the constraints.

Four different intermediate points are taken from this run (points A, B, C and D, shown in Figure 6.8), and fed to “fmincon” as its initial system to start with.

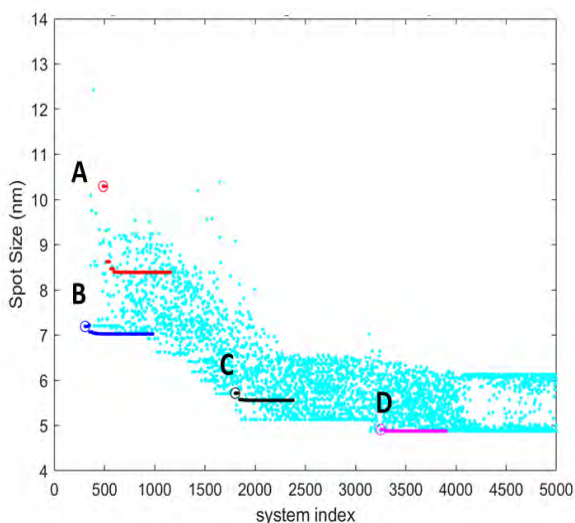


Figure 6.8. Spot size versus iterations, for “fmincon” and GA.

If the optimization problem is a global one, it is expected that “fmincon” can arrive at the same solution as GA. However, as can be seen, each system could be improved only slightly by “fmincon” and could not arrive at the last point which had been found by GA (Figure 6.8) (point A: reached from 10.35 nm to 8.39 nm, point B from 7.19 nm to 7.02 nm, point C from 5.72 nm to 5.55 nm, point D from 4.91 nm to 4.87 nm). It

shows that this problem is a global optimization problem and not a local one, and there are multiple local minima in the objective function landscape of this problem.

It also illustrates that GA can properly perform this optimization problem, by starting from randomly generated systems having high spot size, overcoming many local minima, reaching to a satisfactory result of a system with very small spot size. Moreover, looking at the last point tested by “fmincon” (point D in Figure 6.8), it is seen that this point could also not be improved further by the “fmincon” than the point where GA converged. This shows that GA could act by itself as a good local optimizer and bring the point from different valleys of objective functions to their local point.

6.5 Conclusion

In this work, an investigation on local versus global optimization has been performed to find out whether the objective function landscape of electron lens system optimization is either a global or a local problem. The results show that the search space of this problem has multiple local minima. For example, starting from a probe size of 10.35 nm, the local optimizer reaches a probe size of 8.39 nm. The global optimizer, starting from a probe size of 10.35 nm, reaches a probe size of 4.87 nm. A local optimizer, therefore, is not sufficient to find a satisfactory result of such problems and a global optimizer should be used instead.

It is also shown that a Genetic Algorithm acts as a powerful global optimizer, which could handle this complex multi-dimensional objective function problem having many local minima. However, a local optimizer might be used in addition to GA, to speed up the process of finding the global point.

References

- [1] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C.W. Hagen and P. Kruit , “Multi-electrode lens optimization using genetic algorithms”, *Int. Journal of Modern Physics A*, Vol. 34 , 1942020 (16 p.), 2019.
- [2] J.P. Adriaanse, H.W.G. van der Steen, and J.E. Barth, “Practical optimization of electrostatic lenses”, *J.Vac. Sci. Technol. B7*, pp. 651-666, 1989.
- [3] M.A.J. Van der Stam, J.E. Barth, P. Kruit., “Design of a Multi-Mode Transport Lens with Optimization Program SOEM”, *SPIE Vol. 2014 Charged Particle Optics*, pp.45-56, 1993.
- [4] J.E. Barth, H.W.G. van der Steen and J. Chmelik, “Improvement to the electrostatic lens optimization method SOEM”, *proceeding of SPIE vol. 2522*, pp. 128-137, 1995.
- [5] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, T. Janssen, C.W. Hagen and P. Kruit, *Optimization of electrostatic lens systems using genetic algorithms*, in *Recent Trends in Charged Particle Optics and Surface Physics Instrumentation* , pp. 24–26, 2018.
- [6] Mitchell Melanie, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, London, 1999.
- [7] J.E. Barth, P. Kruit, “Addition of different contributions to the charged particle probe size”, *Optik101*, pp. 101-109, 1996 .
- [8] Miklos Szilagy, *Electron and Ion Optics*, Plenum Press, New York, 1988.
- [9] N. Hesam Mahmoudi Nezhad, *Optical system optimization using genetic algorithms*, Master’s thesis, Delft University of Technology, 2014.

7. Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses

Abstract

The design of electrostatic electron lenses involves the choice of many geometrical parameters for the lens electrodes as well as the choice of voltages applied to the electrodes. The purpose of the design is to focus the electrons at a specific point and to minimize the aberrations of the lens. In a previous study, genetic algorithm optimization was introduced to aid the designer. For speeding up the electrostatic field calculations, new methods for analytical approximations of the field near the optical axis were introduced. In this paper, the influence of the main tuning parameters of the Genetic Algorithms are analyzed. The analysis is performed on a typical electrostatic lens system including 6 electrodes. Different combinations of population sizes and number of generations are taken and the quality of the optimized lens system is compared. It is seen that within a constant computational effort (time or total number of system evaluations), a lower population size with a larger number of generations can achieve better results compared to having larger population size and fewer generations. The combination of Crossover Heuristic with Mutation Gaussian showed significantly better results compared to all other combinations of Mutations and Crossovers. Crossover Fraction is also evaluated to find the most suited values of this parameter.

Keywords

Genetic Algorithms, Tuning Parameters, Electrostatic Lens, Lens Design Optimization.

This paper is based on the following publication:

Neda Hesam Mahmoudi Nezhad, Mohamad Ghaffarian Niasar, Cornelis W. Hagen and Pieter Kruit: "Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses", Submitted to IEEE international conference on numerical electromagnetic and multiphysics modeling and optimization (NEMO), 2023.

7.1 Introduction

Optimization routines such as Genetic Algorithms (GAs), though very effective for finding the optima in complex functions, have not been extensively used for the optimization of electrostatic electron lenses. In such lenses, the objective function is a combination of obtaining the correct focus position and the minimization of lens aberrations. For calculating these, one needs to find the electric field of the lens which is generally calculated by accurate methods such as the Finite Element Method (FEM). To perform the optimization while all geometries and voltages of the lens electrodes may vary, thousands of systems need to be evaluated. Using the accurate field calculation methods such as FEM (60 seconds per system evaluation on a modern PC), the optimization takes a very long time, up to several days [1,2]. In 2018, we presented an optimization technique based on a fast but approximate model to calculate the fields around the optical axis [1, 3, 4] (0.4 second per system evaluation on the same PC). We combined this with an accurate method to calculate the field in a second optimization step. In a further search to reduce the computation time, we now analyze the influence of the tuning parameters of the Genetic Algorithm. We shall vary the population size, the number of generations, mutation method and cross-over type. In this study there is no need of fine tuning the electron lens by an accurate field calculation, so only our approximate method (the second order electrode method, SOEM) is implemented to calculate the electric field. The study is performed on a lens system having 6 electrodes.

7.2 Optimization Problem

An example of an electrostatic lens with 6 lens electrodes is selected as the case-study to perform the optimization. A cross-section of the round lens is shown in Fig 7.1. The free variables for the optimization are the thicknesses (T_i), Radii (R_i) and voltages (V_i) of each electrode, and the gaps between the electrodes (G_i). In total, there are 23 free variables. Any imaging system such as the electrostatic lens system suffers from aberrations. The smaller the aberrations, the higher the resolution of the image and therefore the higher the quality of the lens system. The aberrations can be calculated by aberration integrals, using the electric field on axis and a first order (aberration-free) trajectory. These aberrations can be combined into a contribution to the spot size when the lens is used to image an electron source on a sample as is done

in a scanning electron microscope. The objective function for the optimization problem is hence the spot size at the image side.

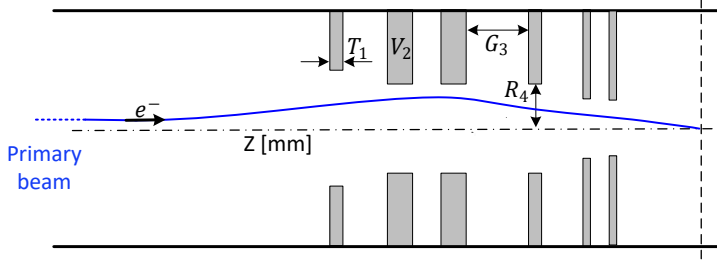


Figure 7.1. A 2D illustration of a typical multi-electrode lens systems with 6 electrodes.

In our case-study it is presumed that the lens only suffers from spherical and chromatic aberrations. The spot size (presented by D_s in equation 7.1) can be calculated using the equation below [5].

$$D_s^2 = (0.18 C_s \alpha^3)^2 + (0.6 C_c \alpha \frac{\Delta U}{U})^2 \quad (7.1)$$

Where C_c and C_s stand for chromatic and spherical aberration coefficients, respectively. α (the half opening angle of the beam) is taken as 10 milliradian. U and ΔU (the acceleration energy and the energy spread of the electron source) are chosen here to be 1 kV and 1 eV, respectively. The constraint of this optimization problem is to have the image at a fixed position X_c (at 15mm). X_c is also a function of the electric field and can be calculated using ray-tracing. In our case-study MATLAB is used as the programming language. To calculate the objective function and image position (i.e. D_s and X_c), the field calculation is performed by SOEM and our ray-tracing codes use the paraxial approximation. The computational work related to this study is performed on a PC with an Intel (R) Xeon (R) W-2123 CPU @3.60 GHz and 32 GB of RAM.

7.3 GA Tuning Parameters

7.3.1 Population size and number of Generation

The population size (P) is determined by the number of members (here electrostatic lens systems) in each generation. The number of generations in an optimization execution is called G. To perform the assessment, different combinations of population sizes and number of generations are taken, both with the values 20,

50,100, 200 and 500, so 25 combinations in total. Since in GA each run with identical parameters can yield a different optimization result, the optimization is run 10 times to give statistically reliable results. The results are represented in Figure 7.2. The average, maximum and minimum of the Objective Function (OF) values are given in blue, gray and green bars, respectively. The units of the objective function are in nanometers. The black thin bars inside the blue bars illustrate the standard deviation in the averaged OF values in 10 runs. The corresponding times are given in Figure 7.2b. Note that in our optimization problem the constraint function is added to the objective function. By giving the lens systems which did not satisfy the constraint a very high contribution to the objective function (approach A mentioned in chapter 6), the lens systems which do not satisfy the constraints are automatically thrown out of the solution pools. To evaluate the GA performance, OF values and the execution time are the two factors which should be evaluated together. It can be seen in Figure 7.2a. that, as expected, increasing the population size and the number of generations, the average value of the objective function decreases (visualized by the dashed yellow lines). From Figure 7.2b it can be seen that at very high population sizes and number of generations, the execution time increases dramatically while the OF values shows no marked improvement. Hence, it can be concluded that if GA is run for a shorter time-frame, a larger improvement can be recognized in that short period rather than longer. This conclusion is in line with the investigations performed on GA in other optimization problems [6].

Another study on population and generation is to evaluate the GA performance within a fixed time, that is with a specified number of system evaluations $N_{PG} = P \cdot G$; the important question is then “for a fixed value of N_{PG} , which combination of P and G achieves the better results”. To study this, the cases with the same N_{PG} having different combinations of P and G are picked from Figure 7.2a and shown together in Figure 7.3. The options with smaller G and larger P are shown in dark blue bars, and the options with larger G and smaller P are shown by light blue bars. The black thin bars inside the blue bars illustrate the standard deviation in the averaged OF values in 10 runs.

In all cases except the first one (i.e. $N_{PG} = 1000$), the light blue bars are lower than the dark blue bars. Note however that the first case shows an unstable result as can be concluded from the large standard deviation. Clearly $N_{PG} = 1000$ is too small for the optimization with so many free parameters. It is hence concluded that for the same amount of evaluations N_{PG} , a GA optimization with a higher number of generations and a smaller population size achieve a better result than a larger population size and a lower number of generations. This conclusion is in contrast with what has been reported in [7]. However, there were also other studies [8] in line with what we conclude here.

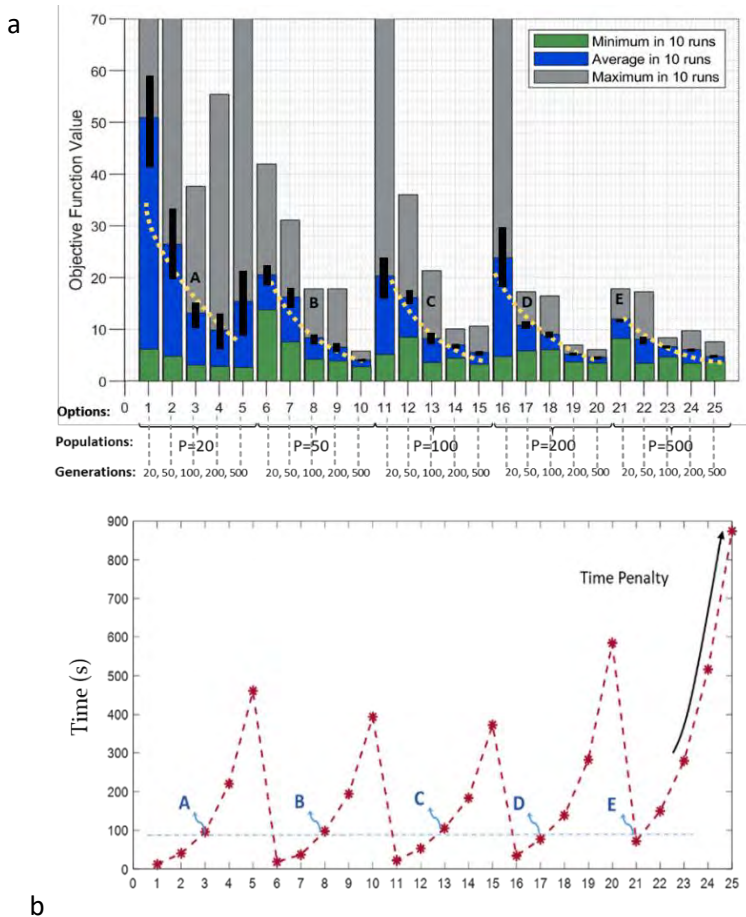


Figure 7.2. a. GA performance for 25 different combinations of P and G (shown in the x-axis) b. The corresponding time for each of the 25 cases.

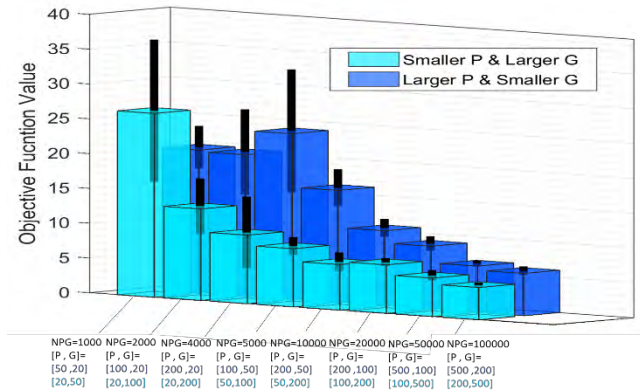


Figure 7.3. Graphs of GA runs to illustrate comparison of similar values of N_{PG} , having different combinations of P and G.

In [8] it is shown that a large number of generations is better when the optimization problem has many basins of attraction, with multiple local minima in the objective function landscape. In such situations, having a large population size would not help GA to search the area more extensively, but it will degrade the GA performance since it will cause GA to be trapped in the wrong basin of attractions and stay in a local minimum. So probably our situation is like that. To continue the rest of the analysis on other GA tuning parameters, a fixed value of P and G is taken at which the computation time was not very long, while the objective function was reasonably small. For this aim, option 8 (pointed out by 'B' in Fig. 7.2.b.) with $N_{PG}=5000$ ($P=50$, $G=100$) and run time ~ 100 Sec, is selected.

7.3.2 Crossover and Mutation

This section is devoted to discovering the most suitable options of the two main tuning parameters of GA, Crossover and Mutation. The mutation type concerns the distribution of the random changes in the population within one generation. The crossover type determines how the offspring in a next generation is formed from parents in the earlier generation.

There are 3 different Mutation methods namely Gaussian, Adapt-Feasible and Uniform, and 5 different options for Crossover; Scattered, Heuristic, Single point, Two point and Arithmetic available in MATLAB (in total 15 different combinations). Figure 7.4a gives the results of the comparison.

The Y-axis shows the average value (blue bars), the maximum (gray bars) and minimum value (green bars) of the objective function value for 10 runs per combination. As can be seen from Figure 7.4a, the best performance comes from the combination Crossover Heuristic and Mutation Gaussian. However, to be able to better compare the options, a 2D graph for averaged values of the objective functions, in a categorized manner (grouped by their different Mutation methods) is given in Figure 7.4b. Looking at this figure, by comparing bars with the same colors, it is seen that all cases with different crossover methods have the smallest OF values when combined with Mutation Gaussian.

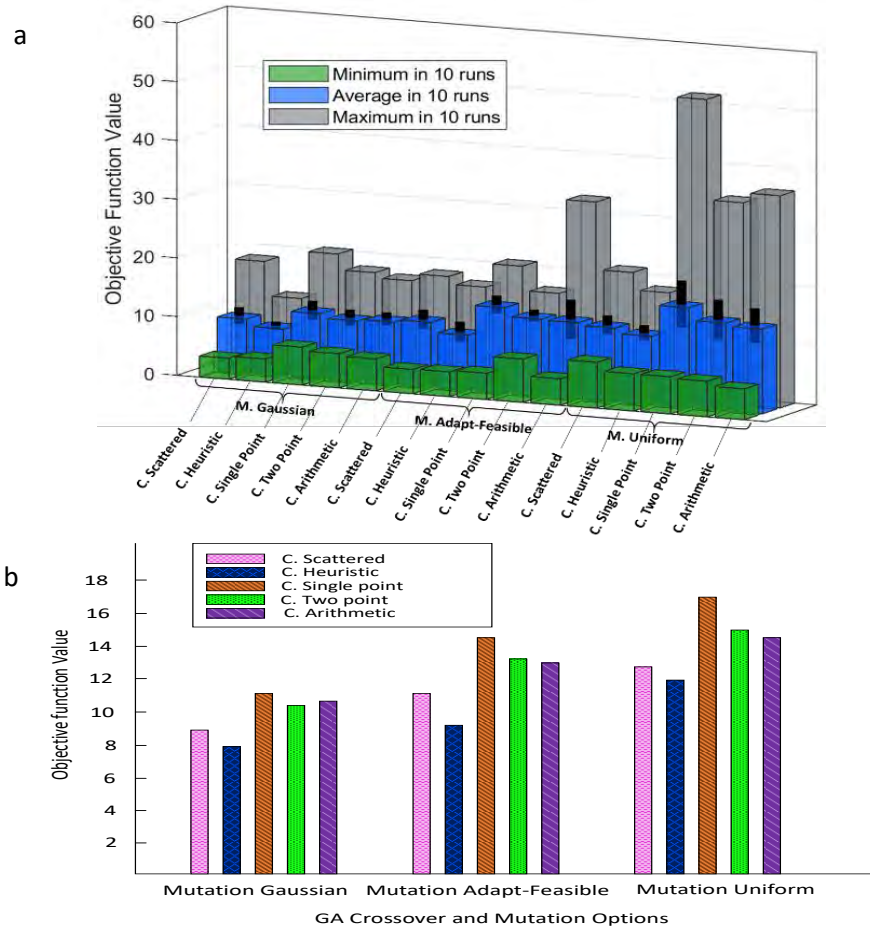


Figure 7.4. a. 3-D Bar graph illustrating the objective function values averaged over 10 runs for 15 different combinations of Crossover and Mutation GA options. b. The data of graph a when categorized according to Mutation and Crossover type.

Among Crossover methods, it is seen that Crossover Heuristic achieves better results than other mutation types. Noticeable is that Mutation Uniform performs the worst. Crossover Single-point also achieves the worst result compared to other Crossover options. It is important to mention that the less favorable tuning parameters in our case study, which have demonstrated poorer results, are not configured as the default settings in MATLAB's Genetic Algorithm (GA). Regarding the other tuning parameters, the variations in results, also by considering the standard deviations, are not substantial. As a result, users can confidently utilize either MATLAB's default GA settings or the recommended tuning parameters provided here, without significant concern about encountering substantial differences in outcomes.

7.3.3 Crossover Fraction

Another tuning parameter of GA which can influence the results is the Crossover Fraction, that is the fraction of individuals that is incorporated in the next generation through the crossover process. The study is performed on 9 different values of the Crossover Fractions varying from 0 to 1 with a step sizes of 0.1. The GA is run 10 times with a Population of 50 and Generation of 100. The Crossover and Mutation methods are taken as Crossover Heuristic and Mutation Gaussian. The results are given in Figure 7.5, with the average values as bars in blue with the error bars as thin black bars. Crossover fractions of 0.5 or 0.6 give the smallest values of the objective function. The result can be understood by realizing that having an intermediate value of the Crossover Fraction allows enough diversity in the population occurs while the diversity is not too high to avoid the convergence of the GA.

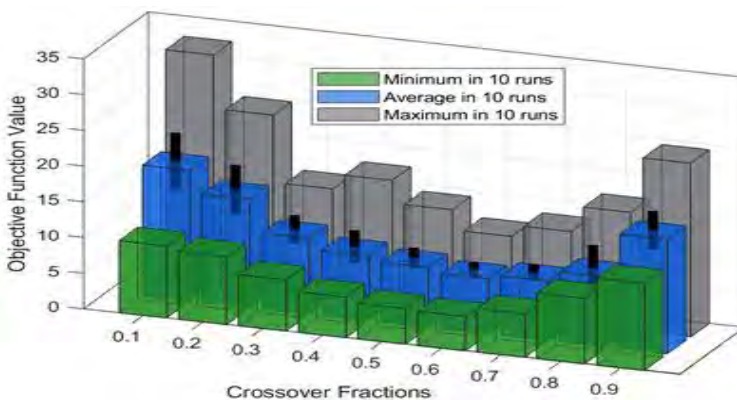


Figure 7.5. 3-D Bar graph illustrating the objective function values averaged over 10 runs for 9 different Crossover Fraction values.

7.4 Conclusion

Having implemented a Genetic Algorithm optimization for electrostatic electron lens design allowed us to perform a study on the influence of GA tuning parameters. The study is performed on a typical lens with six electrodes which has 23 free variables. The extension to more complex designs is straightforward. Our study illustrates that the GA has the robustness to be implemented as a global optimizer for electrostatic lens design. It also shows that there is not one optimized design because the value of the final objective function is different every time the GA is run. This implies that the fine tuning of the GA parameters is important for optimizing the performance of the GA. An analysis is performed of the impact of the values of population size and number of generations. As expected, the results improve by increasing both values. However, a population of 50 with 100 generations can provide reasonably good results. Increasing to higher values of population and generation will not significantly improve the results while the related computational time dramatically increases. It is also seen that within a constant computational effort (time or total number of system evaluations), having a lower population size than the number of generations can achieve better results than having a larger population size than the number generations. The Crossover and Mutation types as the main tuning parameters of GA are analyzed to find the most suitable options. It is found that irrespective of the type of Crossover, the Mutation Gaussian achieves the best result. Moreover, Crossover Heuristic shows the best performance among different crossover types. The combination of Crossover Heuristic with Mutation Gaussian shows significantly better results than all other combinations of Mutations and Crossovers. Crossover Fraction is also evaluated to find the most suitable values of this parameter. It is shown that a Crossover Fraction of 0.5 or 0.6 achieve the best results. The guidelines provided here for tuning the GA parameters may be helpful not only for the optimization of electrostatic lens designs, but also for other GA optimization of functions with similar complexity.

References

- [1] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C. W. Hagen, and P. Kruit, *Int. J. Mod. Phys. A* 34, 1942020 (2019).
- [2] N. H. M. Nezhad, M. G. Niasar, C. W. Hagen, and P. Kruit, *Proceedings of 2020 IEEE 6th International Conference on Optimization and Applications (ICOA)*, (IEEE, Piscataway, NJ, 2020), p. 9094475.
- [3] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, P. Kruit and C. W. Hagen, *J. Vacuum Science & Technology B* 39, 062605 (2021).
- [4] J.P. Adriaanse, H.W.G. van der Steen, and J.E. Barth, "Practical optimization of electrostatic lenses", *J.Vac. Sci. Technol. B*7, pp. 651-666, 1989.
- [5] J.E. Barth, P. Kruit, "Addition of different contributions to the charged particle probe size", *Optik*101, pp. 101-109, 1996.
- [6] M. Sipper et al. Investigating the parameter space of evolutionary algorithms, *Bio Data Mining* 11:2, (2018)
- [7] D. Vrajitoru, *Soft Computing in Information Retrieval*, F. Crestani et al. (eds.), Springer-Verlag Berlin Heidelberg 2000, Large Population or Many Generations for Genetic Algorithms? Implications in Information Retrieval, ISBN : 978-3-7908-2473-5.
- [8] T. Chena, K. Tangb, G. Chenb and X. Yao, *Theoretical Computer Science* 436 (2012) 54–70.

8. Comparison of Different Optimization Techniques in Electron Lens Design

Abstract

To design electron lens systems, applying a fully automated optimization routine has not yet been feasible, especially for the case where the optimization has many free variables of the lens system, such as all parameters that define the geometry of the lens electrodes and the voltage of each electrode. Hence, the study of the implementation of different optimization procedures has not yet been possible either. In one of our previous studies, we have proposed to use the so-called Second Order Electrode Method (SOEM) which performs the electrostatic field calculations in a very short time by the approximations of the field near the optical axis. There, using SOEM in field calculation, a Genetic Algorithm (GA) was successfully implemented to optimize the electron lens systems. One of the questions that has not been studied and answered in the literature yet, is whether the GA is the most suitable option among different optimization techniques for the design/optimization of electron lens systems. In this paper, by implementing the SOEM technique as the field calculation method, different optimization procedures are implemented and their performances are compared. For this study, a typical six electrode lens system is employed. The implemented optimization techniques include calculus-based local optimization ('Fmin') and metaheuristic methods such as GA, Particle Swarm Optimization (PSO), and Simulated Annealing (SA). The results demonstrate that the population-based global optimization techniques like GA and PSO significantly outperform single solution-based local optimization methods such as 'Fmin' and SA. Additionally, PSO shows slightly better performance than GA, although it cannot be concluded that PSO will always outperform GA for every electron lens design problem. Furthermore, in the comparison between the two single-based optimization techniques, the

This chapter is based on the following publication:

Neda Hesam Mahmoudi Nezhad, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "Comparison of Different Optimization Techniques in Electron Lens Design", 2023 IEEE 9th International Conference on Optimization and Applications (ICOA).

metaheuristic approach (SA) outperforms the calculus-based one ('Fmin'). Hence, we recommend implementing metaheuristic, global, population-based optimization techniques like GA and PSO for the optimization electron lens systems.

Keywords

Electron Electrostatic Lens Design, Global Optimization, Local Optimization, Metaheuristic based Optimization Algorithms, Genetic Algorithms, Particle Swarm Optimization, Simulated Annealing, SOEM (Second Order Electrode Method).

8.1 Introduction

Design and optimization of electron lens systems are yet a laborious work for electron-optical designers. In such lens system optimization, the objective function is to obtain the correct focus position while to minimize lens aberrations. To calculate these, the electric field of the lens system, which is generally calculated by accurate methods such as the Finite Element Method (FEM) [1] have to be derived. To perform the electron lens system optimization while voltages and all geometries of the electrodes are free parameters of the optimization, thousands of systems need to be evaluated [2, 3]. Using the existing accurate field calculation methods such as FEM (60 seconds per system evaluation on a modern PC), a fully automated optimization becomes impractical in a feasible time [2, 3]. Due to the problem mentioned above, to our knowledge, there is not yet a fully automated optimization routine which performs the optimization of a multi electrode-lens system, having all its geometric dimensions as free parameters. Therefore, studying the performance of different optimization techniques for electron lens system design has not been studied either. Previously, the authors have presented an optimization technique [2, 3] based on a fast but approximate so-called Second Order Electrode Method (SOEM) (proposed by Adriaanse in 1988) [4, 5] to calculate the electric fields around the optical axis (0.4 second per system evaluation on the same PC) while implementing Genetic Algorithms (GA) [6, 7]. Now, having such an automated and fast routine developed, we decided to use that to perform the above mentioned study on comparison of the performance of different optimization techniques for the electron lens design optimization. In our previous studies on electrostatic lens system design optimization, it was demonstrated that in optimization of electrostatic lens systems, a global optimization is needed [8]. Our case-study is a highly-nonlinear complex optimization problem [3, 8]. In many papers it has been shown that for such optimization problems the meta-heuristic optimization techniques can be the most suitable choice. However, which one of the meta-heuristic technique provides the best result in our case-study? This question will be answered in this paper. In addition, a comparison with a calculus-based local optimization is presented here to show the difference between the performance of a local optimization compared to a global one.

First, a brief recap of different types of the meta-heuristic optimizations and the reason why it is hard to predict in advance which type is the best choice for our case-study, is presented. In section 8.2 the optimization problem is defined. Section 8.3

presents a brief introduction of the most well-known meta-heuristic optimization techniques that are going to be compared in our case-study. In 8.4 the results of the implementation and comparison are provided and the conclusion is presented in 8.5.

More than 50 years have passed since the time when the mathematical foundations of Metaheuristic-based optimization algorithms (hereafter called MA) [9-11] were introduced by the pioneers of this work who include Holland [12], Schwefel [13], Fogel et al. [14] and Rechenberg [15]. MA includes many different algorithms, the most common of which are Genetic Algorithm (GA) [16], Particle Swarm Optimization (PSO) [17] and Simulated Annealing (SA) [18]. From the beginning, the practitioners of this field had questions on how to select the optimization algorithm type. “Is there any specific type of optimization algorithm which outperforms the others?”, and “Can we predict a specific optimization algorithm which can achieve the best performance out of the choice of algorithms for a defined optimization problem?”

In all these years, despite considerable efforts performed by the researchers in this field, it has been discovered that finding the type of optimization algorithm which is best in a specific optimization problem, is neither a problem that can be generalized, nor one easy to predict before running the optimization [19]. It is represented as “No Free Lunch” (NFL) theory in optimization [20].

In other words, a specific type of MA which outperforms the other types in one optimization problem may not do so in another problem with a different objective function landscape. Having the information from the objective function landscape will help to select the most appropriate type of MA. However, knowing the objective function landscape in advance, before running the optimization, is a challenge as it requires the optimization to be run first. Hence, it became a dilemma how to select the most appropriate algorithm among MAs for a specific optimization problem. One way to find this is to run the optimization for different problems, with a variety of MAs, to ascertain the situations at which the optimization achieves the best result. Here, our work is to perform such study in electron optics, for optimization of electrostatic lens systems, which to the best of our knowledge does not yet exist.

8.2 Defined Optimization Problem

For this case study we selected an example of an electrostatic lens with 6 lens electrodes to perform the comparison analysis of the above-mentioned different optimization techniques. A cross-section of the round lens is shown in Fig 8.1. The free variables for the optimization are the thicknesses (T_i), Radii (R_i) and voltages (V_i) of

each electrode, and the gaps between the electrodes (G_i). There are 23 free variables in total. The electrostatic lens system, as any imaging system, suffers from aberrations. The smaller the aberrations, the higher the resolution of the image and therefore the higher the quality of the lens system. The aberrations can be calculated by aberration integrals, using the electric field on axis and a first order (aberration-free) trajectory. These aberrations can be combined into a contribution to the spot size when the lens is used to image an electron source on a sample as is conducted in a scanning electron microscope. Hence, the objective function for the optimization problem is the spot size at the image side.

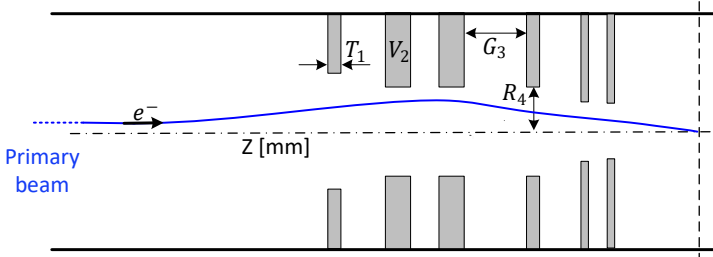


Figure 8.1. A 2D illustration of a typical multi-electrode lens systems with 6 electrodes.

It is presumed in our case study that the lens only suffers from spherical and chromatic aberrations. The spot size (presented by D_s in equation 8.1) can be calculated using the equation below [15].

$$D_s^2 = (0.18 C_s \alpha^3)^2 + (0.6 C_c \alpha \frac{\Delta U}{U})^2 \tag{8.1}$$

Where C_c and C_s represent chromatic and spherical aberration coefficients, respectively. α (the half opening angle of the beam) is taken as 10 milliradian. U and ΔU (the acceleration energy and the energy spread of the electron source) are chosen here to be 1 kV and 1 eV, respectively. This optimization problem's constraint is to have the image at a fixed position X_C (at 15mm from the surface of the first electrode). X_C is also a function of the electric field and can be calculated using ray-tracing. Our case-study used MATLAB as the programming language. To calculate the objective function and image position (i.e. D_s and X_C), the field calculation is performed by SOEM and our ray-tracing codes use the paraxial approximation. A PC with an Intel (R) Xeon (R) W-2123 CPU @3.60 GHz and 32 GB of RAM was used to perform the computational work related to this study.

8.3 Optimization Techniques Applied in Our Case-Study

The optimization procedures which are implemented here include four optimization techniques. The first one is taken from the category “Calculus-Based” which is a local optimization. In MATLAB, this function is referred to as "Fmin" or "fmincon," but for the sake of simplicity, we will subsequently refer to it as "Fmin". The others are from the category of “Metaheuristics”. One is taken from the “Single-solution Based”, i.e. Simulated Annealing (SA). The two others are the most well-known optimization techniques from the “Population-Based” category namely, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).

8.3.1 Calculus-Based local optimization

The so-called “Fmin” in MATLAB, is a “Calculus-based optimization (CBO) technique. CBO uses the gradient (derivatives) of the objective function. This method is implemented on the objective function $f(X)$, starting from an initial point of X_0 , taking the steps of δ_N , moving towards the direction of the negative gradient of the objective function to ultimately reach the local minimum of the function. If it is a maximization problem, the direction will be that of the positive gradient of the function to reach the maximum point. Note that in this method, the function $f(X)$ should be differentiable in all neighboring points which are progressively taken under search. More details on the algorithm can be found in section 3.5 of this thesis.

8.3.2 SA

Simulated annealing (SA) is a single-solution based, meta-heuristic algorithm. This algorithm simulates the physical process of annealing in a material. Annealing means to heat up a substance to a specified level above a phase transition temperature, and then to lower the temperature gradually, in a specific way, to reach the minimum energy level of the system until the material crystalizes. This crystalized substance, with all its lattice atoms perfectly aligned, is an example of nature finding a beautiful optimum structure of a substance. If, however, the cooling process is performed too fast, the crystalized state will not be reached and the substance becomes an amorphous solid rather than a crystal. The way to achieving the minimum energy level and the crystalized state, is to carefully control the rate of temperature decrement.

In the mathematical optimization context, mimicking from the nature process, a trial point (electrostatic lens system parameters in our case-study optimization problem) is randomly generated based on a defined normal probability distribution within the

specified ranges of the parameters. The distance between this trial point and the previous point is based on a probability distribution with a scale parameter depending on the current temperature. The temperature is a pre-defined value by the user. The algorithm then determines whether or not the new point is better than the previous point. If it is better, the previous point is replaced with the new point. If the new point is worse than the previous point, the algorithm may still select the new point based on a probability function. The algorithm lowers the temperature in every iteration. After a certain number of iterations the reannealing process is activated in which the temperature is increased again. The algorithm keeps exploring the landscape until a stopping criteria is reached. More details on the algorithm, formulation and related flowcharts can be found in section 3.4.

8.3.3 GA

Randomly created values of the parameters (elements,) called initial population, are the starting point for the GA , a population-solution based, meta-heuristic global optimization method, mimicking natural evolution. The parameter “population” defines the number of elements in each generation, denoted here by N_{pop} . In nature, the elements are the chromosomes of the organism. In electron lens design, the set of electron lens systems from which the lens design variables are determined, represents the elements.

The GA uses a variety of genetic operators namely Crossover, Mutation and Elitism, to gradually improve a set of elements in a so-called “generation” towards the next generation, having a better set of elements, regarding their objective function values. The algorithm proceeds until it satisfies the stopping criteria, which could be set as, for instance, a maximum computational run time, a specified value of objective function, or a maximum number of generations. In our case-study, reaching a specified computational time is chosen to end the process. More details on the algorithm, formulation and related flowcharts can be found in section 3.2.

8.3.4 PSO

Particle swarm optimization (PSO) presented in 1995 [23], is another type of population-solution based, meta-heuristic global optimization method which mimics the intelligent collective behaviour of some animals in nature such as birds or flocks.

The algorithm starts by generating a set of initial particles (electrostatic lens systems parameters), and assigns a velocity (randomly selected from a range given by the user) to each particle. It evaluates the objective function for each particle and defines the

best location and lowest objective function. It chooses new velocities based on particle current velocity, the particle's individual best location and best location of all particles. It then updates the particle position (current position plus the velocity, while making sure the boundaries and constraints are fulfilled). The iteration continues until a stopping criteria is reached. When the defined stopping criteria is reached the algorithm will be ended. The stopping criteria can be determined by the user using different schemas. In this case-study, reaching a certain computational time is set as the stopping criteria. More details on the algorithm, formulation and related flowcharts can be found in section 3.3.

8.4 Results and Analysis

To compare the performance of the above-mentioned different optimization techniques, the minimum value of the Objective Function (OF) reached by the optimization procedures should be evaluated. However, to achieve a fair comparison, the run time is set to a constant value as the stopping criteria of the optimizations. As stated in chapter 7, a reasonable computation time could be around $T = 1500$ sec. This value therefore is set as the stopping criteria of the optimizations.

It should be noted that for the local optimizations, such as "Fmin", the optimization routine may stop before this fixed time, due to the fact that a calculus-based optimization would stop when it reaches the minimum point at each basin of attractions.

To perform the comparisons, as the first step, an analysis is performed among the four optimization routines running 10 times to achieve statistically-reliable results. The GA and PSO are run without initial data. However, SA and Fmin need to be assigned an initial data set to start. A random initial system is taken and given as the starting point of SA and Fmin. For some of the initial data, these two optimizations could not find any system which satisfied the constraints. As the first step for these algorithms, an initial system is taken which allows the SA and Fmin to reach an optimum system which satisfied the constraints (this initial system is found by trial and error among different randomly created systems).

Figure 8.2 shows the minimum objective function values, averaged over 10 runs, achieved by the four different optimization techniques. The units of the objective function are in nanometers. The green bars represent the averaged minimum objective function values. The black thin bars inside the green bars represent the standard deviation in the averaged values for 10 runs.

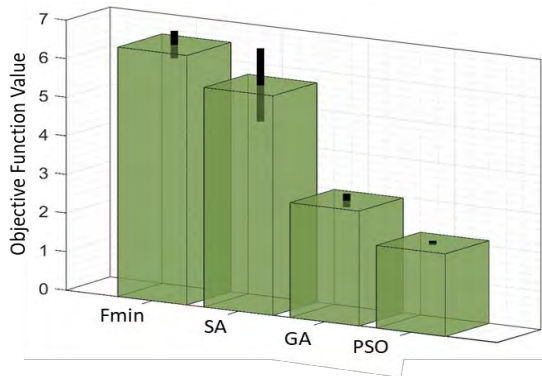


Figure 8.2. The comparison of the performance for four different Optimization techniques namely Fmin, SA, GA and PSO. The presented values are averaged over 10 runs.

Among the 10 runs, one example run (with the objective function value in the middle range) is depicted as the representative of all runs and illustrated in Figure 8.3. The y-axis shows the objective function value in the course of the runs. The x-axis presents the system evaluations. The blue, pink, red and green stars are representatives of runs related to the PSO, GA, Fmin and SA, respectively.

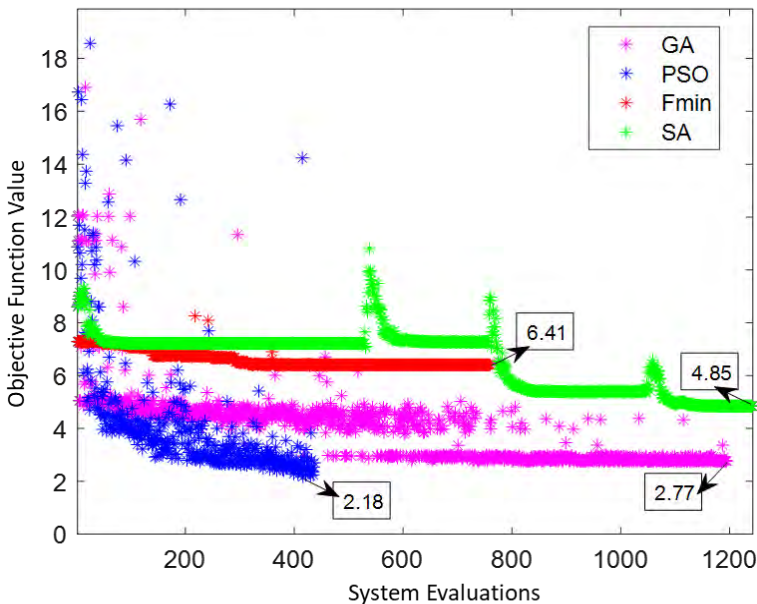


Figure 8.3. Comparison of the progress in the four different Optimizations. The presented data is taken from one run among 10 runs as an example representative of different runs.

From Figure 8.2 and 8.3, it can be seen that the two optimizations GA and PSO clearly outperform the two others Fmin and SA. Moreover, the standard deviations for the former optimizations (GA and PSO) are much lower compared to the two latter ones (Fmin and SA). This is due to the nature of being meta-heuristic and also to starting and proceeding with populations and not only a single system. This provides a much higher chance of finding the area in the parameter landscape with lower values of the objective function. Due to this, in some literature, SA is assumed as a local optimizer and not a global one.

Since for Fmin and SA, the results depend on the initial data (X_0), with which the optimization started, these two optimizations are run with different initial data to achieve a statistically more reliable analysis. Six different initial data sets are given as the starting points. The corresponding runs are called Run1 to Run6, respectively. Each “Run” is executed 10 times to provide statistically reliable results. For the Fmin, because it is based on the derivative of the function, the root finding is always the same and there is no standard deviation. It should also be noted that there are some situations that these two optimizations could not find any systems which have satisfied the constraints.

The data is visualized in figure 8.4. The blue (purple) bars represent the OF for 6 “Run”s related to the SA (Fmin). The black thin bars inside the blue bars are representative of the standard deviations for different runs.

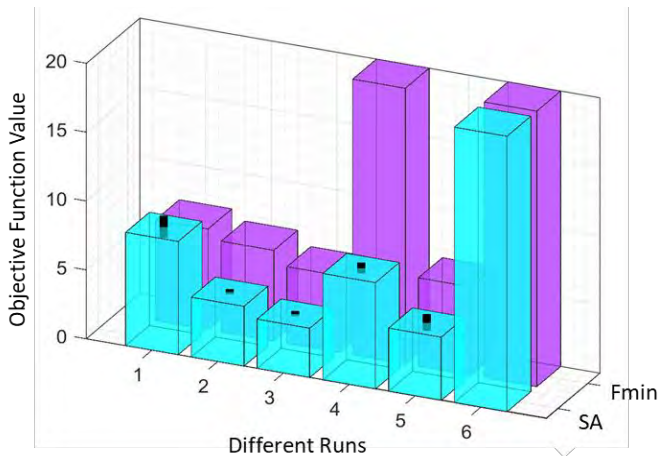


Figure 8.4. Comparison of the performance of Fmin and SA. The presented values are averaged over 10 runs.

As can be seen, the OF related to the SA are in most cases lower than the ones of Fmin. There is only one case (Try1) in which Fmin could perform better than SA. In all other tries SA could outperform the Fmin. This can be due to the metaheuristic nature of the SA which does not exist in the calculus-based optimization procedures such as Fmin. The inherent metaheuristic nature of the optimization process enables a more comprehensive exploration of the search space, preventing it from being confined to a single local minimum associated with the specific basin of attraction of the initial starting point.

8.5 Conclusion

Different optimization techniques such as Fmin, SA, PSO and GA are implemented for electrostatic electron lens design on a typical lens system with six electrodes (23 free variables) to perform a study on the comparison of their performance. The extension of the electrodes to more complex designs is straightforward. In this study it is recognized that the GA and PSO outperform Fmin and SA. The reason can be that the former ones are population-based optimization techniques while the latter ones are single-based optimization ones. It is also seen that the PSO achieves better results than the GA. However, the difference between PSO and GA is not as much as the difference between population-based optimizations and single solution-based optimizations. It is hence advised to implement the meta-heuristic, global, population-based optimizations such as GA and PSO. It is not possible to conclude that PSO will outperform GA for every problem in electron lens design.

Moreover, from the comparison between two single solution-based optimization techniques, it is illustrated that the metaheuristic one (SA) outperforms the calculus-based one (Fmin). These two optimization methods work as a sort of local optimization which might not even find any system within the constraints and thus have limited use in electron lens design.

References

- [1] The Finite Element Method in Electromagnetics (IEEE Press) 3rd Edition, Jianming Jin, John Wiley & Sons. IEEE Press, 2014.
- [2] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, C. W. Hagen, and P. Kruit, *Int. J. Mod. Phys. A* 34, 1942020 (2019).
- [3] N. Hesam Mahmoudi Nezhad, M. Ghaffarian Niasar, A. Mohammadi Gheidari, P. Kruit and C. W. Hagen, *J. Vacuum Science & Technology B* 39, 062605 (2021).
- [4] M. A. J. van der Stam, J. E. Barth, and P. Kruit, *Proc. SPIE* 2014, 45 (1993).
- [5] J. Adriaanse, H. van der Steen and J. Barth, *J. Vacuum Sci. Technol. B: Microelectron. Process. Phenomena* 7, 651 (1989).
- [6] Mitchell Melanie, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, London, England, Fifth printing (1999)
- [7] N. H. M. Nezhad, M. G. Niasar, C. W. Hagen, and P. Kruit, *GA Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses*, proceedings of 2023 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO'2023).
- [8] N. H. M. Nezhad, M. G. Niasar, C. W. Hagen, and P. Kruit, *Proceedings of 2020 IEEE 6th International Conference on Optimization and Applications (ICOA)*, Beni Mellal, Morocco, 20-21 April 2020 (IEEE, Piscataway, NJ, 2020), p. 9094475.
- [9] T. Jones. *Evolutionary algorithms, fitness landscapes, and search*. PhD thesis, University of New Mexico, 1995.
- [10] Fernando G. Lobo, Cláudio F. Lima and Zbigniew Michalewicz (Eds.), *Parameter Setting in Evolutionary Algorithms*, *Studies in computational studies* Vol. 54. Editor-in-chief Prof. Janusz Kacprzyk , Springer-Verlag Berlin Heidelberg (2007)
- [11] K. De Jong: *Parameter Setting in EAs: a 30 Year Perspective*, *Studies in Computational Intelligence (SCI)* 54, 1–18, Springer-Verlag Berlin Heidelberg (2007).
- [12] J.H. Holland. *Outline for a logical theory of adaptive systems*. *JACM*, 9:297– 314, 1962.
- [13] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technical University of Berlin, Berlin, Germany, 1975.
- [14] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, 1966.
- [15] I. Rechenberg. *Cybernetic solution path of an experimental problem*. In *Library Translation 1122*. Royal Aircraft Establishment, Farnborough, (1965).
- [16] D.E. Goldberg, J.H. Holland, *Genetic algorithms and machine learning*. *Mach Learn* 3(2):95–99, (1988).

- [17] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, K. Zenger, "Harmony Search Method: Theory and Applications", *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 258491, 10 pages, 2015. (<https://doi.org/10.1155/2015/258491>).
- [18] Chien-Wei Chu, Min-Der Lin, Gee-Fon Liu, Yung-Hsing Sung, Application of immune algorithms on solving minimum-cost problem of water distribution network, *Mathematical and Computer Modelling*, Volume 48, Issues 11–12, Pages 1888-1900, ISSN 0895-7177,2008. (<https://doi.org/10.1016/j.mcm.2008.02.008>).
- [19] S. Rainer, P. Kenneth, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359, (1997).
- [20] Wolpert, D.H., Macready, W.G., "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation* 1, 67, (1997).
- [21] J.E. Barth, P. Kruit, "Addition of different contributions to the charged particle probe size", *Optik101*, pp. 101-109, (1996).
- [22] P. J. M. van. Laarhoven, *Simulated annealing : theory and applications*, E. H. L. Aarts, Dordrecht: D. Reidel. (ISBN 90-277-2513-6. OCLC 15548651), (1987).
- [23] D. Wang, D. Tan and L. Liu, Particle swarm optimization algorithm: an overview. *Soft Comput.*, 22, 387–408 (2018).

9. A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

Abstract

In electron optics, calculation of the electric field plays a major role in all computations and simulations. Accurate field calculation methods such as the Finite Element Method (FEM), Boundary Element Method (BEM) and Finite Difference Method (FDM), have been used for years. However, such methods are computationally very expensive and make the computer simulation challenging or even infeasible when trying to apply automated design of electrostatic lens systems with many free parameters. Hence, for years, electron optics scientists have been searching for a fast and accurate method of field calculation to tackle the aforementioned problem in the design and optimization of electrostatic electron lens systems.

This paper presents a novel method for fast electric field calculation in electrostatic electron lens systems with reasonably high accuracy to enable the electron-optical designers to design and optimize an electrostatic lens system with many free parameters in a reasonably short time. The essence of the method is to express the off-axis potential in an axially symmetrical coordinate system in terms of derivatives of the axial potential up to the fourth order, and equate this to the potential of the electrode at that axial position. Doing this for a limited number of axial positions, we get a set of equations that can be solved to obtain the axial potential, necessary for calculating the lens properties. We name this method the Fourth-Order Electrode Method (FOEM) because we take the axial derivatives up to the fourth order. To solve the equations, a quintic spline approximation of the axial potential is calculated by solving three sets of linear equations simultaneously. The sets of equations are extracted from the Laplace equation and the fundamental equations that describe a quintic spline. The accuracy and speed of this method is compared with other field

This chapter is based on the following publication:

Neda Hesam Mahmoudi Nezhad, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "A New Approach for Fast Field Calculation in Electrostatic Electron Lens System Design and Optimization", *Journal of Nature (Scientific Reports)* Article no. 4859, pg. 1-21, 2024.

A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

calculation methods, such as the Finite Element Method (FEM) and Second Order Electrode Method (SOEM). The new field calculation method is implemented in design/optimization of electrostatic lens systems by using a Genetic Algorithm (GA) based optimization program for electrostatic lens systems developed by the authors. The effectiveness of this new field calculation method in optimizing optical parameters of electrostatic lens systems is compared with FEM and SOEM and the results are presented.

Keywords

Field Calculation, Electrostatic Lens Systems, Lens Design Optimization, Second Order Electrode Method, Fourth Order Electrode Method

9.1 Introduction

In electron optics, the quality of an electrostatic lens system is specified by its electron-optical properties, which are determined from its electric field [1- 3]. Calculation of the electric field for electrostatic lens systems is hence the most significant step needed in evaluation of the lens system. Therefore, the speed and accuracy of calculating the electric field (or electric potential) plays a major role in electron-optics computation, design and optimization. The electric field can be calculated using numerical methods such as FEM [4], Boundary Element Method (BEM) [5] and Finite Difference Method (FDM) [6]. Although these methods result in very accurate values for the electric field (which consequently results in accurate determination of optical parameters), all these methods are time consuming. For instance, calculation of the potential distribution using FEM and performing ray tracing takes ~1 min for each system [7]. Note: All computer simulations in this study were conducted on our system, a PC equipped with an Intel® Xeon® W-2123 CPU @3.60 GHz and 32 GB of RAM, using the MATLAB programming language. The reported computation times are estimated based on the results obtained on the mentioned PC and programming language. However, as demonstrated in the following sections, the study primarily focuses on the comparative analysis of computational times across different methods, making the specific PC and programming language used less significant; what matters are the relative numbers.

Currently it is not difficult to find electron-optical software such as SIMION [8], EOD [9], GPT [10], CPO [11], etc. to design/optimize electron lens systems which conducts the field calculations by the aforementioned accurate field calculation methods. However, in the design process they only change one lens geometric parameter or the voltages to influence the aberrations or to auto-focus, they are not capable of changing the whole shape of the lens. Assuming these programs can be used in an optimization loop by changing all lens geometries and voltages as free parameters, it can take many days or even weeks to get the results due to the computational time required for the accurate field calculation [7, 12]. For instance, the design of even a simple system using FEM by COMSOL (a multi-physics design modelling software [13]) takes such a long computational time that the designer might not have the patience to wait for the result [7, 12].

In recent years, a method is presented by Lars et al. [14] for the optimization of ion lens design based on an accurate field calculation (FEM), by implementing an adjoint

optimization. This works in a feasible short time and is suitable for multi-objective functions of up to two. However, in electrostatic lens design, we sometimes want more than two objective functions in the design process. Moreover, we have found that electrostatic lens design optimization often has multiple local minima [15]. This hence requires to implement a global optimization technique such as an evolutionary algorithm instead of a local optimization technique, such as the adjoint method (further insights into the evaluation of popular evolutionary optimization algorithms, including Genetic Algorithms (GA) [16, 17], Particle Swarm Optimization (PSO) [18], and Simulated Annealing (SA) [19], alongside a gradient-based local optimization [17], for the optimization of electrostatic lens systems, along with their comparisons, can be found in our other recent study [20, 21]). As a consequence, implementation of any of the above mentioned previously-existed accurate field calculation methods in electron lens system optimization programs (to find the optimum shape of electrodes and potentials), in which usually thousands of systems must be evaluated to obtain a reasonable result, makes the total optimization slow and impractical [7,12]. Therefore, a fast field calculation method is highly demanded in the field of charged particle optical lens design.

Adriaanse et al. (1989) proposed a fast method to calculate the electric field by means of approximation of the axial potential with cubic splines and solving a set of linear equations [22-24]. In their method, the terms of the Laplace equation with higher order than the second derivative are truncated and the method is named the Second-Order Electrode Method (SOEM). Hesam et al. (2019) compared the accuracy of SOEM and FEM methods as well as their computational time for optical parameter calculations in electrostatic lens systems [7, 25]. It was seen that due to the approximation applied some, although very fast (~0.4 sec for each system evaluation), suffers from inaccuracy, specifically for the calculation of optical parameters, such as aberration coefficients. Hesam et al. (2019) [7, 12] proposed a fully-automated technique, utilising a combination of SOEM and FEM, by implementing the Genetic Algorithm (GA) for optimization of multi-electrostatic lens systems which made their design possible with all geometries and voltages of the lens system as free parameters in a reasonable time (around several hours for combination of SOEM and FEM, compared to many days if only FEM was used) [7, 11]. The time-consuming aspect of the described method is associated with the 'fine-tuning' phase of FEM. During this stage, data is input into another segment of the program responsible for recalculating optical parameters using FEM. This iterative process refines and corrects results obtained from SOEM, thereby significantly enhancing the accuracy of optical

parameter calculations. This fine-tuning is essential and necessary to ensure the reliability of the overall optimization.

In the previously presented approach based on SOEM, first running a GA-aided optimization based on field calculation by SOEM is executed. Subsequently, the optimized system's data is used as the initial input for another GA-aided optimization based on field calculation by FEM and to run this part as another full-optimization. Due to the inaccuracies in the field calculation by SOEM, this fine-tuning process for SOEM involves a full-GA-aided optimization run with multiple iterations to correct the systems and identify optimized configurations, where their optical parameters are calculated with the high accuracy of FEM (referred to as SOEM+FEM+GA).

Hence, though it is major progress for electron optics designers to tackle the challenge of designing electrostatic lens systems in a fully-automated way in a reasonable time, the computational time is still not considered to be very short. This is due to the degree of optical parameter calculation inaccuracy in SOEM, which needs more time to be fine-tuned by FEM at the end of the optimization. To increase the accuracy of the field calculation part by SOEM, the Laplace equations therefore need to be solved by considering the higher order terms as well.

In this paper, a new method for fast and reasonably accurate field calculation is presented. This method is based on solving the Laplace equation by retaining terms up to the fourth order derivatives. Hence the method is named the Fourth-Order Electrode Method (FOEM). The accuracy and computation time for field calculation of this method is evaluated and compared with the existing methods SOEM and FEM.

Next, the recently introduced FOEM method is incorporated into an optimization routine utilizing a Genetic Algorithm to assess its efficiency in optimizing electrostatic lens systems, referred to as FOEM+GA and its comparison with a GA-aided optimization based on field calculation by FEM (FEM+GA). In this context, efficiency refers to the time and effort needed to discover an optimized lens system with accurately calculated optical parameters, considering the quality of the optimized lens system found by that method within the given time.

After evaluating the efficiency of FOEM+GA and FEM+GA methods, further analysis is crucial. The results of the FOEM+GA optimization are then employed in another optimization based on FEM calculations, referred to as FOEM+FEM+GA. This analysis aims to determine whether, for FOEM+GA to achieve optical parameter calculation accuracy equivalent to FEM, a comprehensive optimization run based on FEM (as the

fine-tuning step) is necessary after GA+FOEM, or alternatively, the FOEM+GA alone is sufficiently effective, requiring only a brief fine-tuning step by FEM (potentially by just a few iterations in FOEM+FEM+GA) rather than a complete FOEM+FEM+GA optimization with an extensive number of iterations.

Additionally, a comparison with the previously introduced SOEM based optimization method (SOEM+FEM+GA [7, 12]) is essential to investigate the superiority of the newly introduced method based on FOEM in optimizing electrostatic lenses.

This paper is organised as follows. In section 9.2, the mathematical derivation for the axial potential calculation using FOEM is explained and presented. To derive the axial potential formulation, first the fundamental equations of the quintic spline [26] for a dataset with unequal gaps between the data points are derived in sub-section 9.2.1. Next, in sub-section 9.2.2, using these equations in combination with the Laplace equation and proper boundary conditions, a system of linear equations is formed, from which the axial potential and its first and second derivatives are calculated. The accuracy and speed of the field and optical parameters calculation of this new method in comparison with FEM and SOEM is presented in section 9.3 for different typical electrostatic lens systems including 3, 4 and 5 lens electrodes. In section 9.4, the newly introduced field calculation method (FOEM) is implemented in a Genetic Algorithm-based optimization program (FOEM+GA).

Subsequently, in Section 9.4.2, the improvement trends and accuracy calculations of optical parameters for the optimized systems in a single run of GA-aided optimization based on FOEM are investigated and compared with situations where the optimized systems are calculated using SOEM and FEM. This is done to obtain more statistically reliable results for the accuracy comparisons of optical parameters calculated using different methods of FOEM, FEM, and SOEM. Additionally, we illustrate and compare the trends of improvements in finding better lens systems in the mentioned single run of GA optimization based on FOEM with situations where the optimized systems are calculated using SOEM and FEM.

In Section 9.4.3, a comparative study on the efficiency of various GA-aided optimization approaches based on FOEM, FEM, and SOEM is presented. Building on this study, to assess the efficiency of FOEM+GA, the results of FOEM+GA are fed into another optimization based on FEM (FOEM+FEM+GA), and the outcomes are presented at the end of this section. Conclusions are drawn in Section 9.5.

9.2 Mathematical derivation of the axial potential for electrostatic electron lens systems using FOEM

In this section, the aim is to derive the axial potential formulation for rotationally electrostatic lens systems with multi-electrodes. An illustrative example of such lens systems is provided schematically in both 3D (Figure 9.1.a) and 2D (Figure 9.1.b) below. Where, T_j , R_j , and V_{EL_j} refer to the thicknesses, radii, and voltages at each electrode, respectively ($j = 1, 2, \dots, tot$, where tot is the total number of electrodes). While G_j represents the gaps between two consecutive electrodes ($(j = 1, 2, \dots, tot - 1)$).

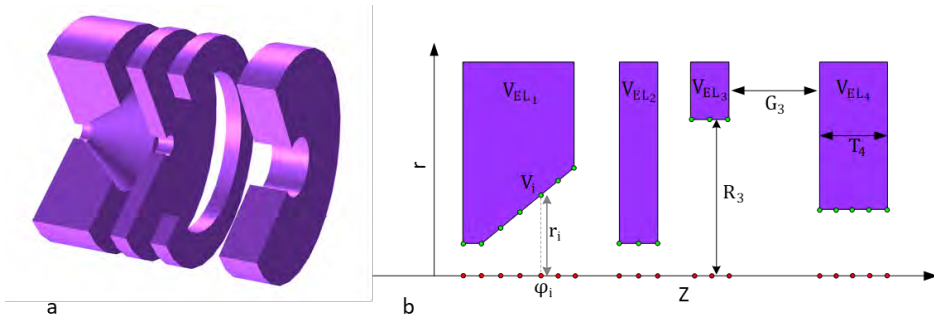


Figure 9.1. a. Illustration of the rotationally symmetrical electrostatic lens system with 4 electrodes in a 3D representation. b. Schematic depicting the rotationally symmetrical electrostatic lens system with 4 electrodes in a 2D representation. Here, T_j , R_j , and V_{EL_j} denote the thicknesses, radii, and voltages at each electrode. G_j represents the gaps between two consecutive electrodes.

It can be demonstrated that for such geometries with axial symmetry, the spatial potential distribution can be obtained from the axial potential and its derivatives. The solution of the Laplace equation for a rotationally symmetrical geometry in terms of the axial potential and its derivatives is given by [27].

$$\varphi(r, z) = \varphi(0, z) - \frac{r^2}{4} \varphi^{(2)}(0, z) + \frac{r^4}{64} \varphi^{(4)}(0, z) + \dots + \frac{(-1)^m r^{2m}}{(m!)^2 2^{2m}} \varphi^{(2m)}(0, z) + \dots \quad (9.1)$$

in which r is the radial distance to the axis, z is the horizontal distance from the entrance of the lens system, $\varphi(r, z)$ is the potential in space at coordinate (r, z) , $\varphi(0, z)$ is the axial potential at coordinate $(0, z)$, and $\varphi^{(2i)}(0, z)$ is the $(2i)^{th}$ derivative of the axial potential at coordinate $(0, z)$. When the symmetry axis is

discretized into N points, and the terms are truncated after the fourth order derivative of the axial potential, equation (1) can be expressed as:

$$V_i(r_i) = \varphi_i - \frac{r_i^2}{4} \varphi_i^{(2)} + \frac{r_i^4}{64} \varphi_i^{(4)} \quad i = 2, 3, \dots, N - 1 \quad (9.2)$$

Considering equation (2), it is evident that the potential distribution along the axis (φ_i) is linked to voltages at the surface of the electrodes ($V_i(r_i)$), where the values of $V_i(r_i)$ are known at each electrode and correspond to the electrode voltages assigned to that specific electrode ($V_{ELj}, j = 1, 2, \dots, tot$).

However, as it is seen, directly deriving the voltage distribution along the axis (φ_i) from this equation (i.e. equation (2)) is not feasible due to the greater number of unknowns compared to the available equations. Therefore, introducing additional conditions becomes necessary for solving the equations.

In this paper it is proposed that the axial potential can be effectively modelled by fitting it to a piecewise fifth-order spline equation. When combined with Laplace's equation, these equations can be solved together to determine the values of voltages along the axis.

9.2.1 Fundamental equations of the quintic spline

While a method to construct the quintic spline for a dataset with equal intervals between data points is presented in [26], it is important to note that, in general, a lens system may have unequal gaps between the electrodes and different thicknesses for the lenses. Consequently, the set of data points created on the electrodes may have unequal intervals. As there are no references found that derive the aforementioned equations for general unequal intervals, this section addresses that gap. Therefore, in the following section, the fundamental equations of the quintic spline are obtained and formulated for the scenario in which intervals between the data points are not equal.

To initiate the formulation, let's consider a dataset represented as (x_i, φ_i) , where $i = 1, 2, \dots, N$, comprising N data points, each with two components. The first component is denoted as x_i , and its corresponding value is φ_i . Our goal is to fit splines to this set of N data points. Specifically, a spline is fitted between data points (x_i, φ_i) and (x_{i+1}, φ_{i+1}) .

For a data set $\{(x_i, \varphi_i)\}, i = 1, 2, \dots, N$, the quintic spline function $S(x)$ is defined by (3):

$$S(x) = S_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 1, 2, \dots, N - 1 \quad (9.3)$$

in which $S_i(x)$ is a fifth order polynomial defined on the interval $[x_i, x_{i+1}]$ and $S_i(x_i) = \varphi_i$.

To ensure a smooth transition from one spline to the next, additional conditions are imposed:

$$S_i^{(r)}(x_{i+1}) = S_{i+1}^{(r)}(x_{i+1}), \quad r = 0, 1, 2, 3, 4 \quad (9.4)$$

where $S_i^{(r)}(x)$ is the r^{th} derivative of $S_i(x)$.

When all these splines are combined, a generalized spline named $S(x)$ is formed. At the data points on this generalized spline, certain conditions are imposed during spline construction, as mentioned above.

Since the polynomial is fifth order, the fourth derivative of the polynomial is a linear polynomial. Therefore the 4th derivative of $S_i(x)$ can be written as (9.5).

$$S_i^{(4)}(x) = Z_{i+1} \frac{(x - x_i)}{\Delta_i} + Z_i \frac{(x_{i+1} - x)}{\Delta_i} \quad (9.5)$$

in which $Z_i = S_i^{(4)}(x_i)$, $x \in [x_i, x_{i+1}]$, and $\Delta_i = x_{i+1} - x_i$. Integrating (9.5) four times, and after some rearrangements, equation (9.6) can be obtained.

$$S_i(x) = Z_{i+1} \frac{(x - x_i)^5}{120\Delta_i} + Z_i \frac{(x_{i+1} - x)^5}{120\Delta_i} + A_i(x - x_i)^3 + B_i(x_{i+1} - x)^2 + C_i(x - x_i) + D_i(x_{i+1} - x) \quad (9.6)$$

In Eq. 9.6, A_i, B_i, C_i , and D_i , $i = 1, 2, \dots, N - 1$ are coefficients which can be determined in terms of φ_i, μ_i, η_i , and Z_i , which are defined in Eq. 9.7.

$$\varphi_i = S_i(x_i), \quad \mu_i = S_i^{(2)}(x_i), \quad \eta_i = S_i^{(3)}(x_i), \quad Z_i = S_i^{(4)}(x_i) \quad (9.7)$$

Taking the derivative of Eq. 9.6 three times results in Eq. 9.8 and using the identities given by Eq. 9.7, results in Eq. 9.9, after simplification.

$$S_i^{(3)}(x) = Z_{i+1} \frac{(x - x_i)^2}{2\Delta_i} - Z_i \frac{(x_{i+1} - x)^2}{2\Delta_i} + 6A_i \quad (9.8)$$

$$A_i = \frac{\Delta_i}{12} Z_i + \frac{\eta_i}{6} \quad (9.9)$$

Eq. 9.10 is then obtained by taking the derivative of Eq. 9.6 twice, which produces Eq. 9.11, using the identities given by Eq. 9.6 and after simplification.

$$S_i^{(2)}(x) = Z_{i+1} \frac{(x - x_i)^3}{6\Delta_i} + Z_i \frac{(x_{i+1} - x)^3}{6\Delta_i} + 6A_i(x - x_i) + 2B_i \quad (9.10)$$

$$B_i = \frac{\mu_i}{2} - \frac{\Delta_i^2}{12} Z_i \quad (9.11)$$

Replacing $x = x_{i+1}$ in Eq. 9.5 and using the identities given by Eq. 9.7 yields Eq. 9.12 which produces Eq. 9.13, after simplification.

$$\varphi_{i+1} = Z_{i+1} \frac{\Delta_i^4}{120} + A_i \Delta_i^3 + C_i \Delta_i \quad (9.12)$$

$$C_i = \frac{\varphi_{i+1}}{\Delta_i} - \frac{\Delta_i^3}{12} Z_i - \frac{\Delta_i^3}{120} Z_{i+1} - \frac{\Delta_i^2}{6} \eta_i \quad (9.13)$$

(9.14) is obtained by replacing $x = x_i$ in (9.6) and using the identities given by (9.7). Following simplification, using (9.11) in (9.14) then produces (9.15).

$$D_i = \frac{\varphi_i}{\Delta_i} - Z_i \frac{\Delta_i^3}{120} - B_i \Delta_i \quad (9.14)$$

$$D_i = \frac{\varphi_i}{\Delta_i} + \frac{9\Delta_i^3}{120} Z_i - \frac{\mu_i}{2} \Delta_i \quad (9.15)$$

In the equations below, the conditions of continuity of derivatives given in (9.4) for $r = 1, 2, 3$ are applied. For the case when $r = 3$, after simple evaluation, (9.16) and (9.17) are obtained. By equating (9.16) and (9.17) and using (9.9), Eq. (9.18) is derived.

$$S_i^{(3)}(x_{i+1}) = Z_{i+1} \frac{\Delta_i}{2} + 6A_i \quad (9.16)$$

$$S_{i+1}^{(3)}(x_{i+1}) = -Z_{i+1} \frac{\Delta_{i+1}}{2} + 6A_{i+1} \quad (9.17)$$

$$\frac{\Delta_i}{2} Z_{i+1} + \frac{\Delta_i}{2} Z_i + \eta_i - \eta_{i+1} = 0 \quad (9.18)$$

For the case when $r = 2$, through a straightforward assessment, (9.19) and (9.20) are obtained. By equating (9.19) and (9.20) and using (9.9) and (9.11), Eq. (9.21) is produced.

$$S_i^{(2)}(x_{i+1}) = Z_{i+1} \frac{\Delta_i^2}{6} + 6A_i \Delta_i + 2B_i \quad (9.19)$$

$$S_{i+1}^{(2)}(x_{i+1}) = Z_{i+1} \frac{\Delta_{i+1}^2}{6} + 2B_{i+1} \quad (9.20)$$

$$\eta_i = -\frac{2\Delta_i}{6} Z_i - \frac{\Delta_i}{6} Z_{i+1} - \frac{\mu_i}{\Delta_i} + \frac{\mu_{i+1}}{\Delta_i} \quad (9.21)$$

For the case when $r = 1$ and after simple evaluation, (9.22) and (9.23) are obtained. By equating (9.22) and (9.23) and using (9.9), (9.11), (9.13), and (9.15), equation (9.24) then results.

$$S_i^{(1)}(x_{i+1}) = Z_{i+1} \frac{\Delta_i^3}{24} + 3A_i \Delta_i^2 + C_i - D_i \quad (9.22)$$

$$S_{i+1}^{(1)}(x) = -Z_{i+1} \frac{\Delta_{i+1}^3}{24} - 2B_{i+1} \Delta_{i+1} + C_{i+1} - D_{i+1} \quad (9.23)$$

$$\begin{aligned} \frac{11\Delta_i^3}{120} Z_i + \frac{4\Delta_i^3 + 4\Delta_{i+1}^3}{120} Z_{i+1} + \frac{\Delta_{i+1}^3}{120} Z_{i+2} + \frac{2\Delta_i^2}{6} \eta_i + \frac{\Delta_{i+1}^2}{6} \eta_{i+1} + \frac{\Delta_i}{2} \mu_i \\ + \frac{\Delta_{i+1}}{2} \mu_{i+1} = \frac{\varphi_{i+2}}{\Delta_{i+1}} - \frac{\varphi_{i+1}}{\Delta_{i+1}} - \frac{\varphi_{i+1}}{\Delta_i} + \frac{\varphi_i}{\Delta_i} \end{aligned} \quad (9.24)$$

On substituting (9.21) in (9.18) and (9.24), equations (9.25) and (9.26) are derived, which are referred to in this paper as the fundamental equations of the quintic spline.

$$\begin{aligned} \frac{\Delta_i}{6} Z_i + \frac{2(\Delta_i + \Delta_{i+1})}{6} Z_{i+1} + \frac{\Delta_{i+1}}{6} Z_{i+2} - \frac{\mu_i}{\Delta_i} + \frac{\mu_{i+1}}{\Delta_i} + \frac{\mu_{i+1}}{\Delta_{i+1}} - \frac{\mu_{i+2}}{\Delta_{i+1}} = 0 \quad i \\ = 1, 2, \dots, N-2 \end{aligned} \quad (9.25)$$

$$\begin{aligned} -\frac{7\Delta_i^3}{360} Z_i - \frac{8(\Delta_i^3 + \Delta_{i+1}^3)}{360} Z_{i+1} - \frac{7\Delta_{i+1}^3}{360} Z_{i+2} + \frac{\Delta_i}{6} \mu_i + \frac{2(\Delta_i + \Delta_{i+1})}{6} \mu_{i+1} \\ + \frac{\Delta_{i+1}}{6} \mu_{i+2} = \frac{\varphi_{i+2}}{\Delta_{i+1}} - \frac{\varphi_{i+1}}{\Delta_{i+1}} - \frac{\varphi_{i+1}}{\Delta_i} + \frac{\varphi_i}{\Delta_i} \quad i \\ = 1, 2, \dots, N-2 \end{aligned} \quad (9.26)$$

9.2.2 Solving the Laplace equations up to the fourth order derivatives

Once the quintic spline is formulated, the axial potential has to be numerically written up to the fourth order, to be simultaneously solved together with the spline equations.

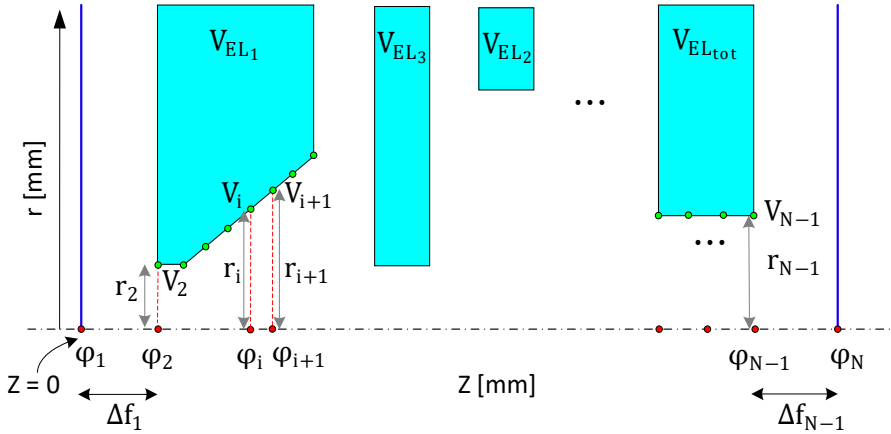


Figure 9.2. A typical rotationally symmetrical electrostatic lens system in 2D. The z-axis is the axis of rotational symmetry.

A typical rotationally symmetrical electrostatic lens system in 2D is shown in figure 9.2. For a clearer visualization of the lens geometry in 2D and 3D, along with detailed explanations of its parameters, please refer to Figure 9.1. The geometry of the electrodes, including the gaps between them and the voltages at each electrode, determine the lens properties.

As aforementioned, by omitting the terms above the fourth derivatives and using the identities in (9.27), Eq. (9.1) can be written numerically as (9.28).

$$\varphi(r, z) = \varphi(0, z) - \frac{r^2}{4} \varphi^{(2)}(0, z) + \frac{r^4}{64} \varphi^{(4)}(0, z) \quad (9.27)$$

$$V_i(r) = \varphi_i - \frac{r_i^2}{4} \mu_i + \frac{r_i^4}{64} Z_i \quad i = 2, 3, \dots, N - 1 \quad (9.28)$$

In electrostatic lens systems, it can be assumed that the axial potential before the first electrode and after the last electrode approaches the potential of the first and the final electrode, respectively. However, the exact position for which the above condition is valid must be identified. Referring to figure 9.2, this means the distances

Δf_1 and Δf_{N-1} must be properly calculated. Equations (9.25) and (9.26) and (9.28) each form $N - 2$ equations, resulting in a total of $3N - 6$ equations, while there are N unknown φ_i, μ_i, Z_i for $i = 1, 2, \dots, N$ which results in $3N$ unknowns in total. However, as stated above, it is assumed that $\varphi_1 = V_2$ and $\varphi_N = V_{N-1}$, in which V_2 is the voltage of the first electrode (i.e. V_{EL1}) and V_{N-1} is the voltage of the last electrode (i.e. V_{ELtot}), are both known values. Therefore, based on the above assumption, the two unknown parameters are already known, but instead, the distances Δf_1 and Δf_{N-1} are unknown, which again results in a total of $3N$ unknown parameters. Considering that there are only $3N - 6$ equations, this means 6 boundary conditions must be selected first. Here we assume $\mu_1 = \mu_N = \eta_1 = \eta_N = Z_1 = Z_N = 0$. These assumptions are all reasonable, since before the first point and after the last point there are field-free regions and therefore assuming the second, third and fourth derivatives of voltage to be zero is also justifiable. Note that all voltages V_i belonging to the same electrode j are equal to V_{ELj} .

It should also be noted that, In equation 9.28, we discretized the Z axis, and as a result, the Z coordinate is embedded in the indices of the parameters. For instance, $z=0$ corresponds to φ_1 , and with a discretization step of, for example, 0.001 mm, the point $z=0.001$ is associated with φ_2 , and so on. When we refer to the 'first electrode,' it implies that there are no other electrodes before it. Consequently, the voltage preceding the first electrode will be dominated by the voltage of the first electrode. This holds true for the last electrode as well. If any other part of the microscope preceding the first electrode has a voltage different from that of the first electrode, we assume that part to be the first electrode. Preceding this assumed first electrode, the voltage will tend to become equal to that of the first electrode.

9.2.3 Calculation of Δf_1 and Δf_{N-1}

In this section, first the values of Δf_1 and Δf_{N-1} are calculated for the case when there is a hole (opening in the electrode near the axis) in the first and in the last electrodes, as shown in figure 9.3.a. In this case it is assumed that the voltages at the outer radius of the lens before the first electrode and after the last electrode are set to the voltage of the first electrode and the last electrode, respectively. It means that on the axis, starting from the entrance of the lens toward the left (with reference to figure 9.3.a), the voltage gradually approaches the voltage of the first electrode, and at point 1 the voltage is exactly the same as that of the first electrode. Based on these conditions it can be concluded that the electric field at the first and the last point is equal to zero.

A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

The Taylor expansion of axial potential for φ_2 and assuming $\Delta f_1 = \Delta f$ can be written as:

$$\varphi_2 = \varphi_1 + \varphi_1' \frac{\Delta f}{1!} + \varphi_1'' \frac{\Delta f^2}{2!} + \varphi_1^{(3)} \frac{\Delta f^3}{3!} + \varphi_1^{(4)} \frac{\Delta f^4}{4!} + \varphi_1^{(5)} \frac{\Delta f^5}{5!} + \dots \quad (9.29)$$

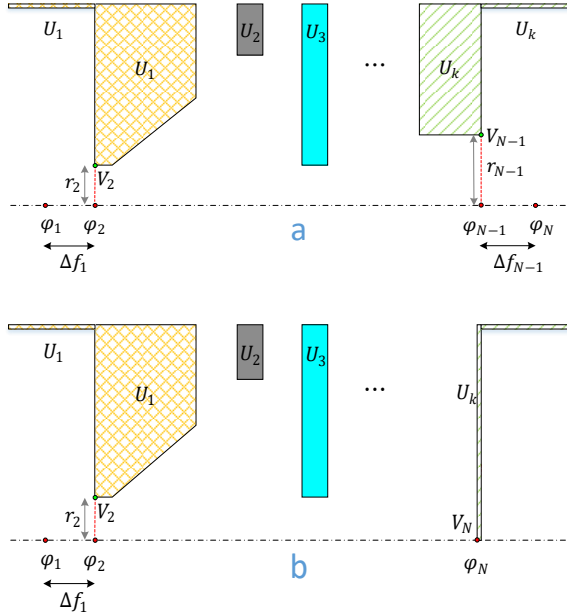


Figure 9.3. Lens system illustration showing the voltages at the outer radius of the lens before the first electrode and after the last electrode are set to the voltage of the first electrode and the last electrode, respectively. a) shows holes in both the first and the last electrodes. b) shows a hole in the first electrode but the last electrode is closed.

Considering the boundary conditions, and also the fact that the electric field at point 1 is zero, equation (9.29) can be simplified to (9.30) by ignoring terms with higher than fifth order derivatives. Alternatively, equation (9.31) can also be written. Combining (9.30) and (9.31) results in (9.32).

$$\varphi_2 = \varphi_1 + \varphi_1^{(5)} \frac{\Delta f^5}{5!} \quad (9.30)$$

$$\varphi_1^{(5)} = \frac{\varphi_2^{(4)} - \varphi_1^{(4)}}{\Delta f} = \frac{\varphi_2^{(4)}}{\Delta f} \quad (9.31)$$

$$\varphi_2 = \varphi_1 + \varphi_2^{(4)} \frac{\Delta f^4}{5!} \quad (9.32)$$

Considering equation (9.28) for $i = 2$, equation (9.33) can be obtained.

$$V_2 = \varphi_2 - \frac{r_2^2}{4} \varphi_2^{(2)} + \frac{r_2^4}{64} \varphi_2^{(4)} \quad (9.33)$$

Equation (9.21) for $i = 1$, by considering that in this case $\Delta_1 = \Delta f$ and using the boundary conditions at point 1, can be written as equation (9.34):

$$\varphi_2^{(2)} = \frac{(\Delta f)^2}{6} \varphi_2^{(4)} \quad (9.34)$$

By combining equations (9.32) and (9.33) and defining $X = \left(\frac{r_2^4}{64} - \frac{r_2^2}{4} \frac{(\Delta f)^2}{6} \right)$, equation (9.35) is derived:

$$\varphi_2^{(4)} = \frac{V_2 - \varphi_2}{X} \quad (9.35)$$

By combining (9.32) and (9.35), equation (9.36) is obtained. Equating $\frac{\Delta f^4}{5!X} = -1$ produces the desired result of $\varphi_1 = V_2$, and results in equation (9.37), after some simplification .

$$\varphi_1 = \varphi_2 \left(1 + \frac{\Delta f^4}{5!X} \right) - V_2 \frac{\Delta f^4}{5!X} \quad (9.36)$$

$$\Delta f^4 - 5r_2^2 \Delta f^2 + \frac{15}{8} r_2^4 = 0 \quad (9.37)$$

Equation 9.37 has four roots, two of which are negative and two positive (Eq. 9.38). The negative ones cannot be acceptable. From the two positive roots, only the smaller one is acceptable. This is because if the larger one is selected as Δf_1 (as shown in Figure 9.4b), considering the larger root as point 1, the smaller root falls between point 2 and point 1. In this way, there is another point on the axis that has a potential equal to V_2 . This means that the voltage before point 2 is not smoothly approaching V_2 , but oscillating, as shown in Figure 9.4b, which is not possible. Therefore, the larger root cannot be an acceptable solution of the equation.

$$\Delta f = r_2 \sqrt{\frac{5 - \sqrt{\frac{35}{2}}}{2}}, \quad \Delta f = r_2 \sqrt{\frac{5 + \sqrt{\frac{35}{2}}}{2}} \quad (9.38)$$

The same concept is valid for the last electrode and Δf_{N-1} can be obtained. In this case, to obtain Eq. 9.34, equations 9.18 and 9.21 must be evaluated for $i = N - 1$, and after combining them Eq. 9.34 is obtained. The rest of the procedure is the same and the smaller root shown in Eq. 9.38 can be used to calculate Δf_{N-1} with replacement of r_2 with r_{N-1} .

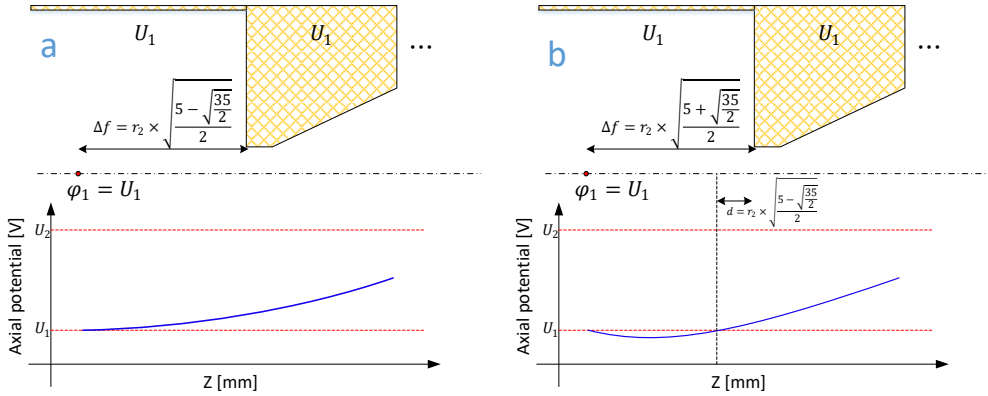


Figure 9.4. Proper selection of Δf_1 and Δf_{N-1} from the roots of Eq. 9.37. a) axial potential smoothly approaches the voltage of the first electrode when Δf is equal to the smaller root of Eq. 9.37, b) axial potential oscillates before it reaches the voltage of the first electrode when Δf is equal to the larger root of Eq. 9.37.

It is worth mentioning that, on the boundary conditions, if the first electrode is open (contains a hole) and the last electrode is completely closed (no hole), as shown in Figure 9.3b, the final data point is φ_N , and its voltage is known and is equal to V_N , the voltage of the last electrode. As before, there are $3N - 6$ equations, while there are $3N$ unknowns in total. However, $\varphi_1 = V_2$ and $\varphi_N = V_N$, which means 2 unknowns are already known, but instead, the distance Δf_1 must be identified, which can be found using (9.38). For this case it is not necessary to add another point after the last electrode and, hence, Δf does not exist after the last electrode. Therefore, only 5 extra boundary conditions are needed. The extra boundary conditions imposed are $\mu_1 = \mu_N = \eta_1 = Z_1 = Z_N = 0$. Note that for this case it is not necessary to set $\eta_N = 0$. In this case, while the electric field at the first point is zero, the electric field at point N is an unknown value which depends on the lens system's geometry and voltages.

9.2.4 Matrix formulation of the equations

First, the matrix formulations of the equations are explained for the scenario in which both the first and the last lenses have a hole in them. Assume $[\varphi]_{N \times 1}$ is the vector of the axial potential, $[\mu]_{N \times 1}$ is the vector of the second derivative of the axial potential, and $[Z]_{N \times 1}$ is the vector of the fourth derivative of the axial potential. Equation 9.25 in the matrix form can be written as Eq. 9.39. Elements of matrices $[A]$ and $[B]$ can be obtained from Eq. 9.25. The close format of matrices are represented in the text. The open-format of matrices are given in the appendix C for more clarity.

$$[A]_{(N-2) \times N} [Z]_{N \times 1} = [B]_{(N-2) \times N} [\mu]_{N \times 1} \quad (9.39)$$

With the boundary conditions $Z_1 = Z_N = \mu_1 = \mu_N = 0$, it is possible to reduce equation Eq. 9.39 and obtain Eq. 9.40. In Eq. 9.40 $[Z^*]$ and $[\mu^*]$ are the vectors of the fourth derivative and second derivative of axial potential, respectively, starting from point 2 until point $N - 1$. Matrices $[A^*]$ and $[B^*]$ are the reduced $[A]$ and $[B]$.

$$[A^*]_{(N-2) \times (N-2)} [Z^*]_{(N-2) \times 1} = [B^*]_{(N-2) \times (N-2)} [\mu^*]_{(N-2) \times 1} \quad (9.40)$$

Equation 9.26 in the matrix form can be written as Eq. 9.41. Elements of the matrices $[C]$, $[D]$ and $[K]$ can be obtained from Eq. 9.26. With the boundary conditions $Z_1 = Z_N = \mu_1 = \mu_N = 0$, it is possible to reduce Eq. 9.40 to equation (42). In Eq. 9.42 matrices $[C^*]$, $[D^*]$ and $[K^*]$ are the reduced $[C]$, $[D]$ and $[K]$. In Eq. 9.42 $[U_0]$ is a vector of size $(N - 2) \times 1$, in which element 1 is φ_1 , which is equal to the voltage of the first electrode, element N is φ_N , which is equal to the voltage of the last electrode, and all the other elements are zero.

$$[C]_{(N-2) \times N} [Z]_{N \times 1} + [D]_{(N-2) \times N} [\mu]_{N \times 1} = [K]_{(N-2) \times N} [\varphi]_{N \times 1} \quad (9.41)$$

$$\begin{aligned} [C^*]_{(N-2) \times (N-2)} [Z^*]_{(N-2) \times 1} + [D^*]_{(N-2) \times (N-2)} [\mu^*]_{(N-2) \times 1} \\ = [K^*]_{(N-2) \times (N-2)} [\varphi^*]_{(N-2) \times 1} + [U_0]_{(N-2) \times 1} \end{aligned} \quad (9.42)$$

Equation 9.28 can be written in matrix form, as in Eq. 9.43. In equation 9.43, vector $[V]$ represents the voltages of all points on the electrodes (therefore it is a known vector) and matrix $[r]$ is a diagonal matrix of size $(N - 2) \times (N - 2)$ and the diagonal elements are the distances between each point on the electrode to its mapped counter point on the axis of symmetry.

$$\begin{aligned} [V]_{(N-2) \times 1} = [\varphi^*]_{(N-2) \times 1} - \frac{[r]_{(N-2) \times (N-2)}^2}{4} [\mu^*]_{(N-2) \times 1} \\ + \frac{[r]_{(N-2) \times (N-2)}^4}{64} [Z^*]_{(N-2) \times 1} \end{aligned} \quad (9.43)$$

Equation 9.40 can be written as Eq. 9.44. Using Eq. 9.44 and Eq. 9.42, equation Eq. 9.45 can be calculated. Combining equation Eq. 9.45 and Eq. 9.43 results in equation Eq. 9.46, after simplification. Rewriting Eq. 9.46, equation Eq. 9.47 can be obtained from which $[\varphi^*]$ can be calculated. Using $[\varphi^*]$ in Eq. 9.45 $[Z^*]$ is derived and using $[Z^*]$ in Eq. 9.44, $[\mu^*]$ can be obtained. Adding the first and last element to the vectors $[\varphi^*]$, $[\mu^*]$ and $[Z^*]$ using the boundary conditions, $Z_1 = Z_N = \mu_1 = \mu_N = 0$

and the fact that φ_1 is equal to the voltage of the first electrode and φ_N is equal to the voltage of the last electrode, the full vectors of $[\varphi]$, $[\mu]$ and $[Z]$ can be determined.

$$[\mu^*] = [B^*]^{-1}[A^*][Z^*] \quad (9.44)$$

$$[Z^*] = \{[C] + [D][B^*]^{-1}[A^*]\}^{-1}([K^*][\varphi^*] + [U_0]) \quad (9.45)$$

$$[V] - \left(\frac{[r]^4}{64} - \frac{[r]^2}{4} [B^*]^{-1}[A^*] \right) \{[C] + [D][B^*]^{-1}[A^*]\}^{-1}[U_0] \quad (9.46)$$

$$= \left(I + \left(\frac{[r]^4}{64} - \frac{[r]^2}{4} [B^*]^{-1}[A^*] \right) \{[C] + [D][B^*]^{-1}[A^*]\}^{-1}[K^*] \right) [\varphi^*]$$

$$[\varphi^*] = [P]^{-1}[Q] \quad (9.47)$$

$$[P] = \left(I + \left(\frac{[r]^4}{64} - \frac{[r]^2}{4} [B^*]^{-1}[A^*] \right) \{[C] + [D][B^*]^{-1}[A^*]\}^{-1}[K^*] \right)$$

$$[Q] = [V] - \left(\frac{[r]^4}{64} - \frac{[r]^2}{4} [B^*]^{-1}[A^*] \right) \{[C] + [D][B^*]^{-1}[A^*]\}^{-1}[U_0]$$

Having established $[\varphi]$, $[\mu]$ and $[Z]$, it is possible to formulate quintic spline equations $S_i(x)$ for $i = 1, 2, \dots, N - 1$. $S_i(x)$ can be calculated using Eq. 9.6. Coefficients A_i , B_i , and D_i can be directly calculated using Eq. 9.9, Eq. 9.11 and Eq. 9.15. To calculate C_i , η_i for $i = 1, 2, \dots, N - 1$ must first be determined from Eq. 9.21, and based on the assumed boundary condition $\eta_N = 0$. Having all η_i and using Eq. 9.13, it is now possible to calculate C_i .

If the last electrode is closed, the same strategy as explained above can be used to calculate $S_i(x)$ for $i = 1, 2, \dots, N - 1$. The only difference is that, after determining η_i for $i = 1, 2, \dots, N - 1$, the value of η_N should be calculated using Eq. 9.18 by setting $i = N - 1$.

Note: To solve the matrix equations, various software and programming languages like MATLAB or Python can be employed for this task. In general, a linear matrix equation in the form of $Ax = B$ can be easily solved, for example, in MATLAB by defining the matrices A and B and using the command $x = A^{-1}B$. Since all matrices presented in Section B-2 can be computed from the geometry data and electrode voltages, the final matrices can be effortlessly calculated through a straightforward implementation in MATLAB. Following this, as explained, it is feasible to compute

cubic splines, and from there, the voltage distribution along the axis of symmetry can be derived.

9.3 Accuracy of FOEM

To analyse the accuracy of calculation of the electric field and optical parameters using FOEM, six different typical electrostatic lens systems including 3, 4 and 5 electrodes, are taken as test systems (represented in 2D in figure 9.5). The axial potential calculation is performed by the two previously existing methods of SOEM and FEM (using COMSOL [13]) and by the newly-presented FOEM method. For the SOEM and FOEM calculation, we use MATLAB [28]. First, the axial potential and its derivatives are graphically compared. In the graphical comparison, only the graphs related to one system (system1 from figure 5) are depicted and presented here since all lens systems show almost similar trends. The overlaid graphs are presented in figures 9.6 and 9.7.

A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

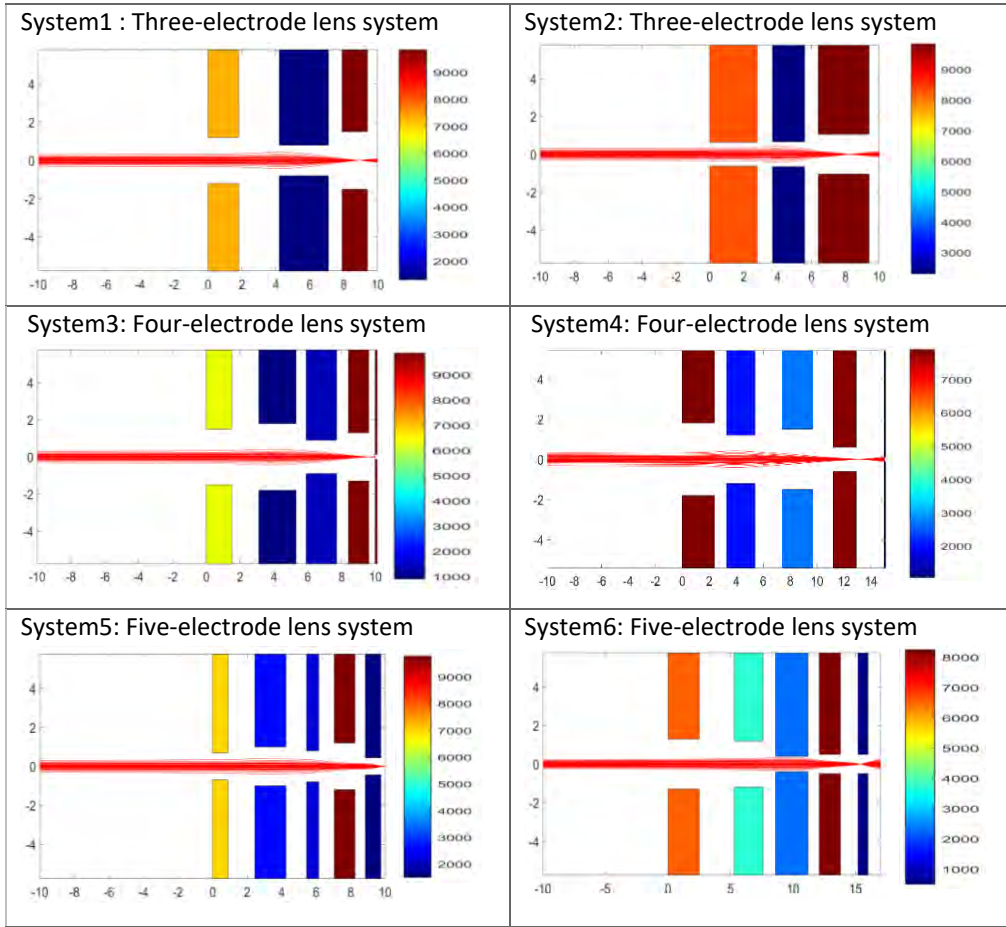


Figure 9.5. Six different typical electrostatic lens systems (in 2D) taken as test systems for making the comparison of the methods of SOEM, FOEM and FEM. The colours indicate the voltages at each electrode. The primary beam passing through the lens system (in red), as taken from our raytracing codes in MATLAB using FEM field calculation, is sketched for a better visualisation of the lens. The units along the axes of the graphs are in millimetres.

In electron optical lens systems, the aberration coefficients are the factors which determine the quality of the lens systems. The lower the values of these parameters, the less the aberrations exist and therefore the better the lens system [1-3]. In our case-study, the lens systems are only suffering from the spherical and chromatic aberrations. Hence, the chromatic and spherical aberration coefficients (denoted by C_c and C_s) are the deterministic factors in the lens design and optimization here. These parameters are functions of the axial potential and its derivatives. The presentation of the formulation is given in equations 9.48 and 9.49. More details on the formulations

and their derivations are skipped in this paper and can be found in the references [7, 12].

$$C_S = \frac{1}{16\phi_0^{\frac{1}{2}}} \int_{z_0}^{z_i} \phi^{\frac{1}{2}} \left(\frac{5}{4} \left(\frac{\phi''}{\phi} \right)^2 + \frac{5}{24} \left(\frac{\phi'}{\phi} \right)^4 r_\alpha^4 + \frac{14}{3} \left(\frac{\phi'}{\phi} \right)^3 r_\alpha^3 r_\alpha' - \frac{3}{2} \left(\frac{\phi'}{\phi} \right)^2 r_\alpha^2 r_\alpha'^2 \right) dz \quad (9.48)$$

$$C_c = \phi_0^{\frac{1}{2}} \int_{z_0}^{z_i} \left(\frac{3}{8} \right) \frac{\phi'^2}{\phi^{\frac{5}{2}}} r_\alpha^2 dz \quad (9.49)$$

Where, ϕ_0 is the potential at the object side, and $r_\alpha(z)$, refers to the principle imaging ray [1], travelling from the object side along the optical axis, with angle 1. The principle imaging ray $r_\alpha(z)$ is calculated by ray tracing.

The accuracy in the calculation of C_S and C_c is important in electron lens design. Since these parameters are functions of the potential and its derivatives, hence they can be used as an important qualitative and quantitative measure to check the accuracy of the axial potential calculations. Therefore, these optical parameters are also derived from the calculated axial potential and its derivatives by FOEM, SOEM and FEM and compared. The related graphs are presented in figure 9.8. The quantitative data related to these optical parameters are presented and compared in Tables 1 and 2.

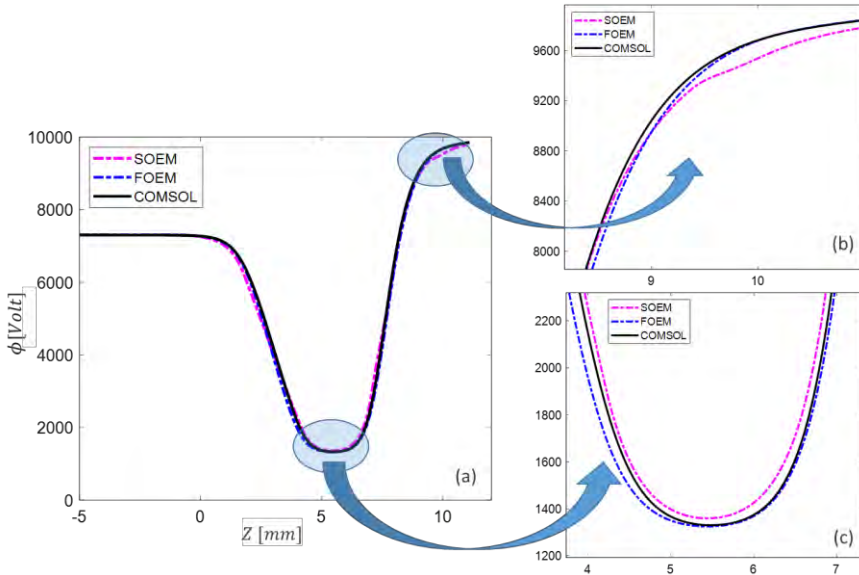


Figure 9.6. a). The overlapped graphs of the axial potential for System1 from figure 9.5, calculated with different methods of SOEM, FOEM and FEM. b) and c) the enlarged sections of the graphs.

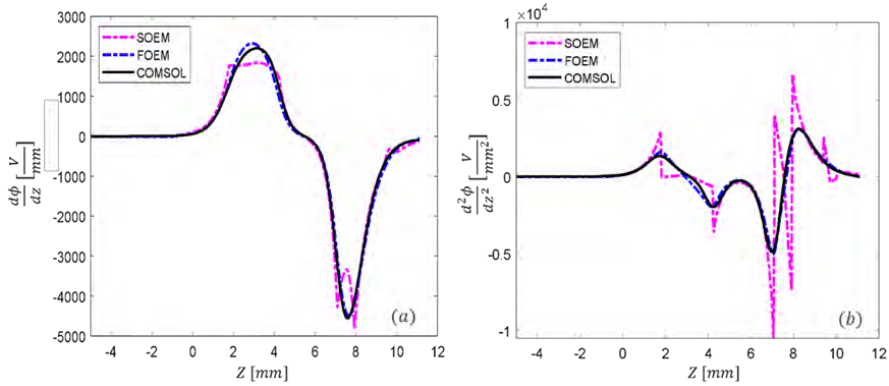


Figure 9.7. The overlapped graphs of the first a) and second b) derivative of the axial potential for System1 from figure 5, calculated with different methods of SOEM, FOEM and FEM.

Figure 9.6.a, compares graphically the axial potential calculated by the three different methods (SOEM, FOEM and FEM), and a reasonably good overlap in the axial potential graphs can be seen. However, when zooming in (figure 9.6.b and c, slight deviations are recognized. These deviations become more recognizable in the graph of ϕ' (figure 9.7.a), and ϕ'' (figure 9.7.b). It is evident that the deviations are mainly between SOEM and FEM, while there is reasonably good overlap between FOEM and FEM.

In the C_s and C_c graphs, shown in figure 9.8, similar to the graphs of the axial potential derivatives, the graphs of FOEM are in much higher agreement with FEM than the ones of SOEM. This behaviour is more recognizable in the C_c graphs. This is because in our case-study, C_s is dependent on both the first and second derivative of the axial potential, while C_c is only a function of the first derivative, and the fluctuations in the second derivatives (as seen in figure 9.7.a and b) are larger.

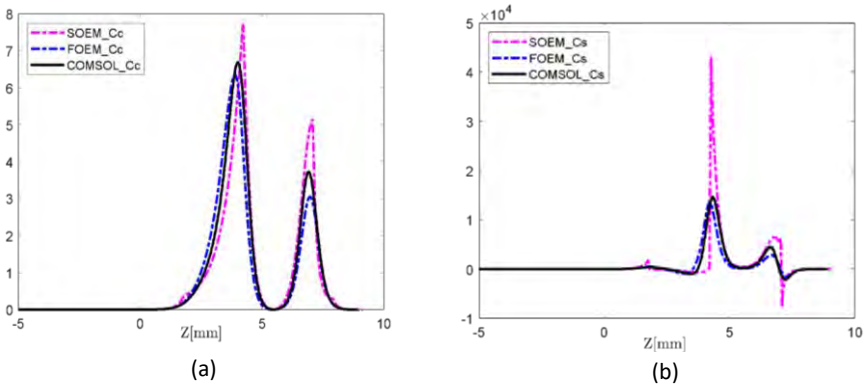


Figure 9.8. The overlapped graphs of chromatic (a) and spherical (b) aberration coefficients for System1 from figure 9.5, calculated with different methods of SOEM, FOEM and FEM. The vertical axis has unit of mm.

In addition to the graphical comparisons, to also achieve quantitative data comparisons, the values of optical parameters such as chromatic and spherical aberrations (C_c and C_s) as well as the image position (X_c) (another important factor in lens design) [1], are calculated for the six lens systems (from figure 9.5) and presented in Tables 9.1 and 9.2. The presented data of C_s and C_c are related to the image side.

Table 9.1. The data related to the optical parameters such as chromatic and spherical aberrations (C_c and C_s) and the image position (X_c), calculated by FEM, FOEM and SOEM, for six different typical electrostatic lens systems (presented in figure 9.5).

Systems		System1	System2	System3	System4	System5	System6
$X_c(mm)$	FEM	8.92	8.26	9.65	13.39	10.03	15.43
	FOEM	8.87	8.29	9.61	13.12	9.97	15.39
	SOEM	9.12	8.62	9.87	13.78	10.35	15.65
$C_c(mm)$	FEM	18.71	15.26	23.73	26.13	1.82	0.98
	FOEM	18.85	15.35	23.45	27.79	1.89	0.94
	SOEM	19.45	16.80	24.10	18.78	2.03	0.77
$X_s(mm)$	FEM	21.13	54.70	22.88	217.65	7.14	12.48
	FOEM	21.98	56.85	21.82	206.65	7.47	11.02
	SOEM	30.77	104.58	37.07	392.57	13.19	11.77

From Table 9.1, it can be seen at a glance that all optical parameters, when calculated by FOEM are much closer to the accurate values calculated by FEM, compared to those calculated by SOEM. The difference between FOEM and SOEM in being more compatible with FEM, can be especially realized in C_s values.

To estimate the deviation of the calculated data with FOEM and SOEM for each system, with respect to the accurate data, the error is determined by taking COMSOL data (FEM) as the reference for accurate values. The errors are presented in Table 9.2 for each system. The last column in Table 9.1 presents the average value of the error for 6 lens systems. This average is computed as a mathematical mean of the errors over the six different lens systems ($(\sum_{i=1}^6(error_i))/6$).

As it is seen at a glance from Table 9.2, the reported errors from calculation with FOEM for all systems are much smaller than the ones calculated with SOEM. It should be noted that it might be possible to occasionally get a result with a slightly larger calculation error with FOEM than with SOEM. This scenario can occur due to the determination of the total values of C_s and C_c through the integration of summations involving terms dependent on ϕ , ϕ' and ϕ'' . Fluctuations are evident in the graphs of these parameters, particularly seen in figures 9.8.a and b for ϕ' and ϕ'' . When examining the overlap of graphs from FOEM and SOEM with those from FEM in these figures, larger deviations are noticeable in SOEM compared to FOEM. However, in the calculations of C_s and C_c , owing to the complexities of the functions, the terms in the integral may occasionally compensate for each other. This compensation results in the total values of C_s and C_c calculated for SOEM occasionally getting closer to those calculated with FEM compared to FOEM with FEM, even though the absolute values at each point in the graphs of ϕ , ϕ' and ϕ'' for FOEM and FEM exhibit closer values. Such instances, however, are infrequent. In general, as shown in Table 9.2, the calculated data by FOEM consistently demonstrates much smaller errors compared to SOEM.

Table 9.2. The percentage of error in calculation of the optical parameter derived by FOEM and SOEM, for six different typical electrostatic lens systems (presented in Table 9.2) compared with the data calculated with the accurate method of FEM.

Systems		System1	System2	System3	System4	System5	System6	Error avg.
Error in XC	FOEM	0.56 %	0.36 %	0.41 %	2.02 %	0.60 %	0.26 %	0.70 %
	SOEM	2.24 %	4.36 %	2.28 %	2.91 %	3.19 %	1.42 %	2.73 %
Error in Cc	FOEM	0.75 %	0.59 %	1.18 %	6.35 %	3.85 %	4.08 %	2.80 %
	SOEM	3.95%	10.09 %	1.56 %	28.13 %	11.53 %	21.43 %	12.79 %
Error in Cs	FOEM	4.02 %	3.93 %	4.63 %	5.05 %	4.62 %	11.70 %	5.66 %
	SOEM	45.62 %	91.18%	62.01 %	80.37 %	84.73 %	5.69 %	61.60 %

As illustrated in Table 9.2, FOEM values provide reasonably accurate results for X_c (with the average value of error 0.7%), with respect to the accurate values of FEM. Also, noticeable is that, FOEM resulted in much smaller error percentages (approx. 4 times smaller) than SOEM with respect to the accurate values. A similar trend of having higher accuracy for FOEM than SOEM is seen in the values of C_c . The error percentage, on average, for C_c calculation by FOEM is around 5 times smaller than by SOEM (i.e. 2.8% for FOEM versus 12.79%). Having much smaller error percentage (~4-5 times) for C_s calculation in FOEM compared to SOEM (5.6% for FOEM versus 61.6%) keeps the same trend of higher accuracy of FOEM. However, the absolute values of the error percentages in C_s are much higher than in the cases of C_c and X_c calculation. which is due to their different functionality to the potential and its derivatives. The computation time needed to evaluate the optical parameters for each lens system, using FEM, FOEM and SOEM are ~ 60 sec, 0.5 sec, and 0.4 sec, respectively.

In summary, the quantitative data comparison, similar to the graphical analysis, demonstrates that FOEM performs the optical parameters calculation in a very short time with reasonably high accuracy, showing deviations ranging from around 1% to a maximum of 6% compared to FEM. Notably, FOEM exhibits significantly higher accuracy in both field calculation and optical parameter calculations compared to SOEM. This is particularly evident for optical parameters dependent on the second derivative of the potential, such as C_s .

9.4 Optimization of electrostatic electron lens systems based on FOEM and a comparison with SOEM- and FEM-based optimization

As discussed earlier, the primary goal of developing a fast field calculation method for electrostatic lens systems was to incorporate it into an automated optimization routine for lens system optimization. With the identification of the proposed fast field calculation method (FOEM), the next step is to implement it in an optimization routine (here Genetic Algorithm) to assess its efficiency in producing optimized lens systems with accurately calculated optical parameters, all within a reasonably short computational runtime. This objective will be addressed in this section.

In this section therefore first a brief introduction on Genetic Algorithm (GA) in the context of electrostatic electron lens system optimization is provided in subsection 9.4.1. Then a section is introduced (section 9.4.2), to conduct an optimization that

calculates the potential using FOEM (referred to as 'FOEM+GA') and verify the accuracy of the calculated optical parameters associated with the resulting lens systems within a single run of GA. The aim of this section is to analyze the accuracy and trends in optical parameter calculation using FOEM, SOEM, and FEM in a single run of GA Optimization.

The problem in comparing the outcomes of the three methods is that they all may yield small values for the optimization parameter, but the calculation of the parameters by SOEM and FOEM may be inaccurate and when the parameters of the resulting optimized designs are calculated by FEM, they may be much higher. The solution to that problem is of course to calculate the optical parameters of the resulting designs by FEM.

There is one more check we need to do in order to have a fair comparison between FOEM and SOEM. In a previous study [7, 12], we followed a SOEM optimization by a FEM optimization that was fed by the results of the SOEM optimization. This turned out to be very efficient as compared to a strict FEM optimization. Now the question is if a strict FOEM optimization, or a FOEM optimization with subsequent FEM optimization can compete with that earlier approach. A detailed exploration of the aforementioned subjects is presented in Section 9.4.3.

9.4.1 Introduction to Electrostatic Electron Lens Optimization by FOEM+GA

The Genetic Algorithm (GA), a member of the evolutionary algorithms, emulates natural evolution, drawing inspiration from Darwin's theory. The process commences with the creation of a randomly generated initial population denoted as $P_1(x_1, \dots, x_n)$ where ' x_i ' signifies the chromosomes in natural evolution and, in the context of electrostatic electron lens system optimization, represents the electrostatic electron lens systems. Consequently, P_1 constitutes an initial set of randomly generated electrostatic electron lens systems based on the variables of the lens system. In the context of Genetic Algorithms (GA), the user has the option to provide some or all of the initial elements of the first population instead of implementing solely the randomly generated ones.

In natural evolution, over time, the initial population gradually evolves towards members that are better adapted to their environmental conditions. Likewise, in the GA, the initial population progresses through successive "generations" towards a new

set of systems $[P_{i+1}(x_1, \dots, x_n)]$ that better satisfy the problem conditions. The conditions identified for optimization are specified by a designated "objective function". The term "generation" in the GA corresponds to the number of iterations the algorithm undergoes before termination. In this study, this parameter serves as the stopping criterion for the GA. The GA is determined by also other tuning operators such as "crossover," and "mutation."

9.4.2 The calculation of optical parameters in a single run of GA Optimization based on FOEM, versus SOEM and FEM

To undertake this analysis, a full optimization is performed utilizing the Genetic Algorithm to optimize a four-electrode electrostatic lens system, with optical parameters calculated using the FOEM technique. Based on insights from our recently published work on tuning GA parameters for electrostatic lens system optimization [21], also considering the fact that the primary goal of this investigation is not to obtain the best-optimized system but rather to derive results for comparing different field calculation approaches implemented within GA, the chosen GA tuning parameters are as follows: population = 50, generation = 100, Crossover : "Heuristic" and Mutation: "Gaussian" techniques, with a crossover fraction of 0.6 (indicating the proportion of the population through which the next generation is created by the crossover function).

All the geometries of the lenses, such as thicknesses (T_j), radii (R_j) and gaps between electrodes (G_j), as well as the voltages at each electrode (V_{EL_j}) are free parameters of the optimization. These parameters can range in the interval of $1\text{ mm} < T_j$ ($j = 1, \dots, 4$) $< 3\text{ mm}$, $0.3\text{ mm} < R_j$ ($j = 1, \dots, 4$) $< 2\text{ mm}$, $1\text{ mm} < G_j$ ($j = 1, 2$) $< 3\text{ mm}$ and $700\text{V} < V_{EL_j} < 10\text{ kV}$ ($j = 2, 3$). The voltage of the first electrode is 7000V. In total, there are 12 free lens parameters in the optimization. The voltage at the image plane is kept fixed at 1000 V. The total length of the lens system is kept constant (here 15mm). A schematic drawing of such a lens system can be seen in figure 9.5 (System4). The origin of the x-coordinate is at the left side of the first electrode's surface. The primary beam comes from far behind the origin (-100 mm), to focus at the image plane. During the optimization the constraints are to have a fixed image position (at $X_c = 15\text{ mm}$, with a tolerance of ± 0.05), and to have a maximum electric field $< 15\text{ kV/mm}$ (to avoid the voltage breakdown between sequential electrodes) . The optimization objective is to obtain systems with the lowest value of the spherical

and chromatic aberrations. The objective function is therefore a function of C_S and C_C . How to define the objective function based on C_S and C_C can be different and depends on the application. Here, a summation of C_C and C_S , with weights of 5 and 1, respectively, is taken. The objective function can be mathematically written as:

$$\text{Objective function} = 5 \cdot C_C + C_S \quad (9.50)$$

The abovementioned choice of the geometry, objective function and constraints, resembles an earlier reported probe-forming electrostatic objective lens system [5]. However, since the aim is to compare the results of “FOEM+FEM+GA” with “SOEM+FEM+GA”, this specific selection does not influence the results of the analysis and can also be altered with other logical choices.

For the optimization, a GA is used with the generation and population of 100 and 50, respectively. Therefore, a total of 5000 systems are being evaluated in this optimization. From the results of this optimization, the 30 best non-identical systems (the system with the lowest objective function values with different designs) are taken for the comparison analysis. The related data are presented in figure 9.9.

Note: we didn't include all 5000 systems from their respective generations in the graph since they were too crowded. Instead, we manually selected the 30 best non-identical systems—those with the lowest objective function values and different designs—for this comparison. The reason for not selecting the real 30 best systems out of the GA run was to avoid having almost 30 identical systems that GA converged to as solutions in the last generations. This would have hindered a proper observation of the trend of decrement. Additionally, not having 30 almost very different lens systems would not provide a suitable basis for testing optical parameter accuracy calculations. Therefore, we manually chose 30 systems to represent the overall trend of decrement in their objective functions. These systems were sorted in the plot and identified by their sorted system numbers (System 1 to 30) in figure 9.8.

The data of X_C , C_C and C_S , for these 30 systems are given as squares (in red) in figures 9.9.a, b, and c, respectively. The optical parameters of X_C , C_C and C_S for these 30 systems are again calculated using SOEM and FEM (by COMSOL). The data calculated by SOEM are presented with stars (in blue), and by FEM are shown with circles (in black) in the same figures (9.9.a – 9.9.d). As it is clear from figure 9a, the data for X_C are reasonably accurate both for FOEM and SOEM compared to FEM, but with higher accuracy for FOEM than for SOEM. It is noticeable from figures 9.9.b,c and d, that the data of C_C and C_S for FEM and SOEM follow the same decreasing trend as FOEM,

however, with much less deviation from the accurate data of FEM for FOEM data compared to those of SOEM.

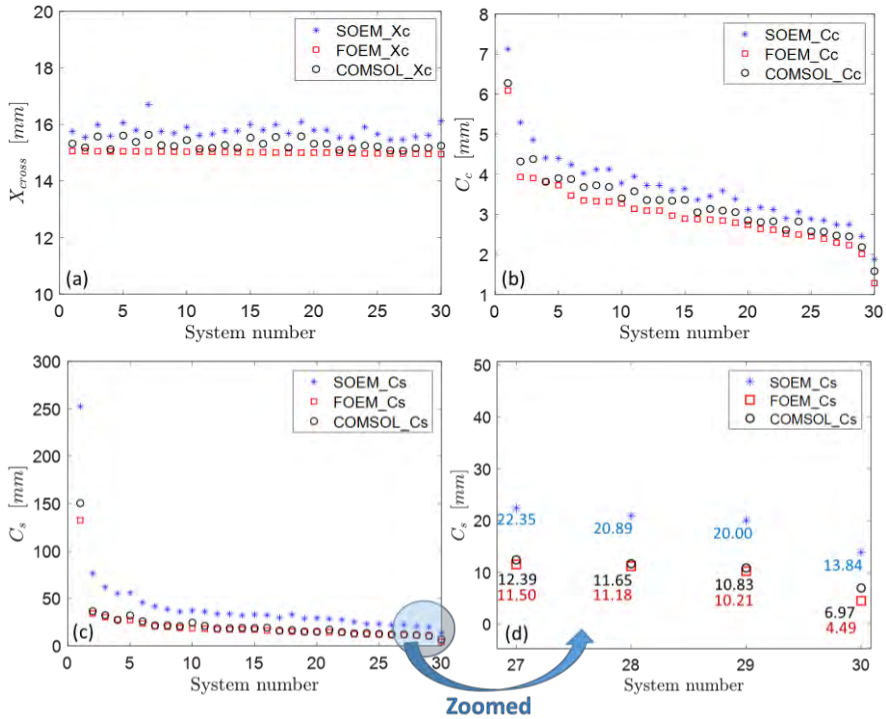


Figure 9.9. (a) Comparison of crossover point (X_C), (b) chromatic aberration coefficient (C_C), (c, d) and spherical aberration coefficient (C_S), for the best 30 systems obtained using SOEM, FOEM and COMSOL. The 30 systems are the best non-identical optimised systems derived from an optimisation using GA with data calculation by FOEM.

The first conclusion is that both SOEM and FOEM can be used in an initial optimization to start from no initial data and reach reasonably optimized systems. However, it is important to note that the inaccuracy in the data obtained with SOEM is large, specifically in C_S (see figure 9.9.c and zoomed-in in 9.9.d).

9.4.3 The efficiency of GA optimization based on FOEM, FEM and SOEM

In this subsection, the objective is to compare the outcomes of FOEM+GA, FEM+GA, and SOEM+GA. To achieve this, different optimizations, as detailed below, will be executed and subsequently compared: FEM+GA optimization starting with random parameters (very time consuming), then FEM+GA starting with a feed from SOEM and

finally FEM+GA starting with a feed from FOEM. The generation (denoted by ‘G’) and population (denoted by ‘P’) in GA are taken as 100 and 50, respectively. The results of these three optimizations are presented in Figure 9.10.a, 9.11.a and 9.12.a. Figures 9.10.b, 9.11.b and 9.12.b depict a zoomed-in view. As previously discussed, the results from FEM are accurate. Therefore, the results taken from “FEM+GA” do not need to be fine-tuned. In order to get accurate and optimized out-put results, the 10 best optimized systems are taken and fed into another optimization by GA as its initial population, but this time the objective function is calculated using FEM. The results of these optimizations which are called “SOEM+FEM+GA” and “FOEM+FEM+GA”, are presented in figures 9.11.c and 9.12.c, respectively. The quantitative data related to these optimizations are represented and summarized in Table 9.3.

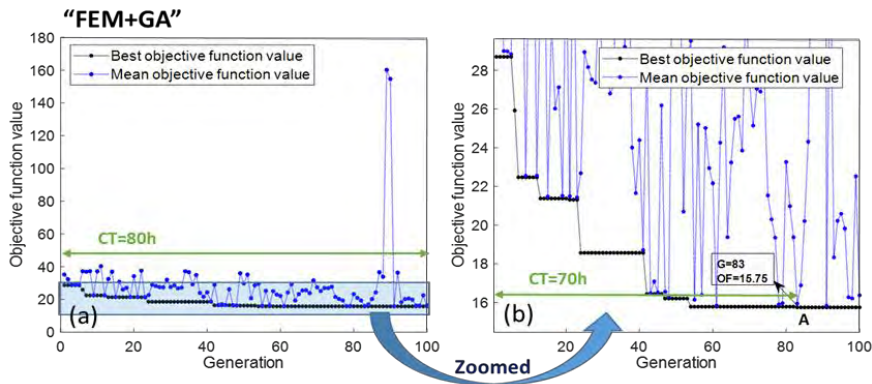


Figure 9.10.(a) Optimisation of a 4-electrode lens system using GA, while the optical parameters are calculated by FEM (“FEM+GA”). The graph represents the objective function (OF) values (Y-axis) versus generation (X-axis). The black/blue dots show the best/mean objective function values at each generation. Figure b shows the zoomed-in data of figure a. System A is the best system found in this optimisation with OF=15.75, occurring at generation 83. ‘CT’ stands for the computation time needed to run the optimisation till the generation indicated by the green arrow.

As is seen from figure 9.10.a, to run the full optimization for “FEM+GA”, the computation time (denoted by CT in the figures 9.10-9.12), is 80 hours. However, the best system, with the objective function of 15.75 (shown by ‘A’ in figure 9.10.b), is reached at generation G=83. Hence, the computation time needed to reach the best system in this case is around 70 hours. After this time, the optimization could not be improved any further within its 100 generations. Therefore the CT needed for “FEM+GA” can be reported as 70 hours and the best system has OF=15.75 (represented in Table 9.3 as well).

Figure 9.11, shows the results of “SOEM+FEM+GA”. The first plot, 9.9.a, presents the improvement of the objective function versus generation during the optimization. The total computation time to run this optimization is CT~33 minutes. The zoomed-in data is shown in figure 9.11.b where it is seen that the optimization has improvements almost until the last generation. The best system (system B in figure 9.11.b) is found at generation of G=99, with an objective function of OF=19.01. The computation time to find this system is ~32 minutes. The related data for this system is given in Table 9.3. As is evident from Table 9.3, the image position of this system when derived from “SOEM+GA” is reported to be $X_C_{SOEM}=14.96$ mm. However, when the optical parameters of this system are calculated accurately by FEM, $X_C_{FEM}=14.54$ mm and $OF_{FEM}=13.56$ is achieved.

Here OF_{FEM} seems to have a small value of 13.56. However, it should be noted that the accurate X_c when calculated by FEM turned out to be $X_C_{FEM}=14.54$ and not at the expected required value of $X_C=15$ mm. It means, this system is not at the required fixed-image position and might have a larger OF value, when it is tuned to have its X_C at 15 mm.

A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

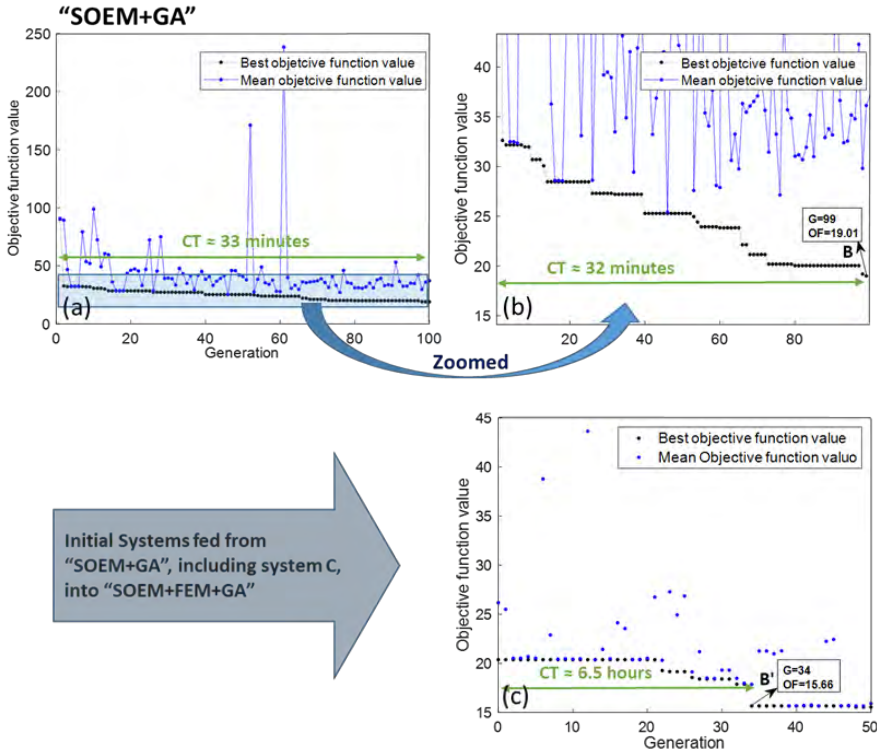


Figure 9.11. (a) The results from an optimisation of a 4-electrostatic lens system using GA, while the optical parameters are calculated by SOEM ("SOEM-GA"). The graph represents the objective function (OF) values (Y-axis) versus generation (X-axis). The black/blue dots show the best/mean objective function values at each generation. Figure (b) shows the zoomed-in data of figure (a). System B is the best system found in this optimisation. 'CT' stands for the computation time needed to run the optimisation until the generation indicated by the green arrow. Figure (c) presents the result of an optimisation using GA, called "SOEM-FEM-GA", with the initial data fed from a previous optimisation of "SOEM-GA". A lens system, depicted by B', is an optimised system resulting from "SOEM-FEM-GA" with OF=15.66, occurring at generation 34.

This system, together with nine other optimized systems, are fed into another optimization of "FEM+GA" (with P=20, G=50), as their initial population. The optimized system (B' in figure 9.11.c), is found at generation of G=34. The optimization did not show recognizable improvement afterwards. Therefore G=34 is taken to be the generation at which the optimization reached the best system. The computation time to reach this system is ~6.5 hours. Figure 9.11.c shows that the best system found by SOEM actually had an OF=21 when properly focused instead of the values OF=19.01

or 13.56 as discussed above. The FEM optimization could improve the design to OF=15.66.

To complete the comparison, an optimization is performed using FOEM . The results of this optimization are presented in figure 9.12.a. It is evident that the computation time for a full optimization is around 41.5 minutes. The best system (system C with OF=10.9) here is found at G=68 after around 30 minutes. No further improvement is recognized after that.

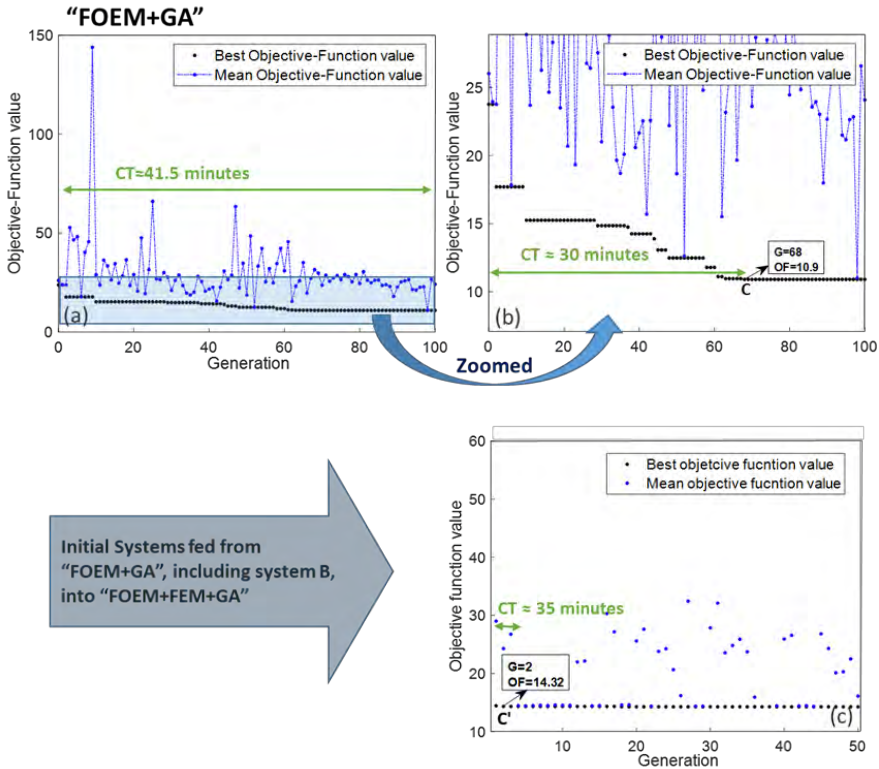


Figure 9.12. (a) The results from an optimisation of a 4-electrostatic lens system using GA, where the optical parameters are calculated by FOEM (“FOEM-GA”). The graph represents the objective function (OF) values (Y-axis) versus generation (X-axis). The black/blue dots show the best/mean objective function values at each generation. Figure (b) shows the zoomed-in data of figure (a). System B is the best system found in this optimisation. ‘CT’ stands for the computation time needed to run the optimisation until the generation indicated by the green arrow. Figure (c) presents the result of an optimisation using GA, called “FOEM-FEM-GA”, with the initial data fed from a previous optimisation of “FOEM-GA”. A lens system depicted by C’, is an optimised system resulting from “FOEM-FEM-GA” with OF=14.32, occurring at generation 2.

A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization

As seen from Table 9.3, the image position and objective function values of system C derived by “FOEM+GA” are $X_{C_FOEM} = 15.00$ mm, and $OF_{FOEM} = 10.9$. These parameters when they are calculated accurately by FEM are reported to be $X_{C_FEM} = 14.93$ mm and $OF_{FEM} = 15.18$ (see Table 9.3). To achieve the accurate and optimized out-put system, this optimized system (system C in figure 9.12.b), together with nine other best systems are taken and fed into another optimization of “FEM+GA”, with $P=20$, $G=50$ as the initial population. The result of this optimization is presented in figure 12c. As is evident from figure 9.12.c, the optimization already reached the optimized system (system C' in figure 9.12.c) after only the first generation (at $G=2$, $OF=14.32$). After the second generation the optimization improvements are almost negligible. This means that the initial systems which had been fed were close to the accurate solutions calculated by FEM.

The reason why the “SOEM+FEM+GA” optimization takes a longer time than “FOEM+FEM+GA” to find the optimized system is that the objective function and image position of the optimized system found by this optimization (system B) deviate more from the corresponding accurately calculated values by FEM (see Table 9.3) than those achieved by “FOEM+GA” (system C).

Table 9.3. The data related to the comparison between three optimisation strategies of “FEM-GA”, “FOEM-FEM-GA” and “SOEM-FEM-GA”.

Different Optimization Strategies	1 st Optimization	Tuning	2 nd optimization	Total Time
“FEM”	“FEM+GA” Time (T1)= 70 hours System A : $OF_{FEM}=15.75$, $X_{C_FEM}=14.99$	Tuning not needed	—	= T1 = 70 hours
“SOEM”	“SOEM+GA” Time (T1)= 32min System B : $OF_{SOEM}=19.01$, $X_{C_SOEM}=14.96$ Accurately calculated by FEM: $OF_{FEM}=13.56$, $X_{C_FEM}=14.54$	Tuning Time (T2)= 10 min Tuned system B $OF_{FEM}=20.37$, $X_{C_FEM}=15.04$	“SOEM+FEM+GA” Time (T3)= 6.5 hours System B', by “FEM+GA” : $OF_{FEM}=15.66$, $X_{C_FEM}=14.99$	= T1 + T2 + T3 = 7.25 hours
“FOEM”	“FOEM+GA” Time (T1)= 30 min System C : $OF_{FOEM}=10.90$, $X_{C_FOEM}=15.00$ Accurately calculated by FEM: $OF_{FEM}=14.93$, $X_{C_FEM}=15.18$	Tuning Time (T2)= 10 min Tuned system C $OF_{FEM}=14.35$, $X_{C_FEM}=15.02$	“FOEM+FEM+GA” Time (T3)= 35 min System C', by “FEM+GA” : $OF_{FEM}=14.32$, $X_{C_FEM}=15.01$	= T1_FOGA + T2_TunFOGA (T3 not needed) = 40 min

In summary, based on what has been discussed above and reported in Table 9.3, it is seen that the total computation time needed to run a fully-automated optimization, for “FEM+GA”, “SOEM+FEM+GA” and “FOEM+FEM+GA”, is 70 hours, 7 hours and 1.25 hour (i.e. 35 min + 40 min), respectively. The optimized systems found by these three

methods, which satisfied the constraints, have objective function values of 15.75, 15.66 and 14.32 ,respectively. Although “SOEM+FEM+GA” could reach an optimized system with low objective function within a reasonably short time (~7 hours), “FOEM+FEM+GA” could reach it ~10 times faster and it is even ~70 times faster than the optimization based on “FEM+GA”. From these results, it is clear that “FOEM+FEM+GA” could reach a lower value of the objective function and in an enormously shorter computation time than the other two optimization strategies.

However, what is noticeable here is that, as it is seen from the last row of the Table 9.3, the optimized system found by only “FOEM+GA” got close enough to the accurate optimized system, which after a very short time of tuning (10 minutes) could achieve an optimized system which could not be further improved by a full optimization of FEM+GA (after 35 min of “FOEM+FEM+GA”, the optimized system has the optical parameters of: OF_FEM=14.32 with XC_FEM=15.01 mm, resulting from improving on a previously optimized system of : OF_FEM=14.35, XC_FEM=15.02 mm).

Hence, it is evident that FOEM results are close enough to the accurate results such that this method has the possibility to be even used separately in an optimization run without the need of another full run of FEM, while SOEM needed that. This method therefore can have high impact on the process of automated design and optimization of electrostatic lens systems.

To have a visualisation of the electrostatic lens system optimized by “FOEM+GA” , system C’ is presented in figure 9.13 (2D in a, 3D in b).

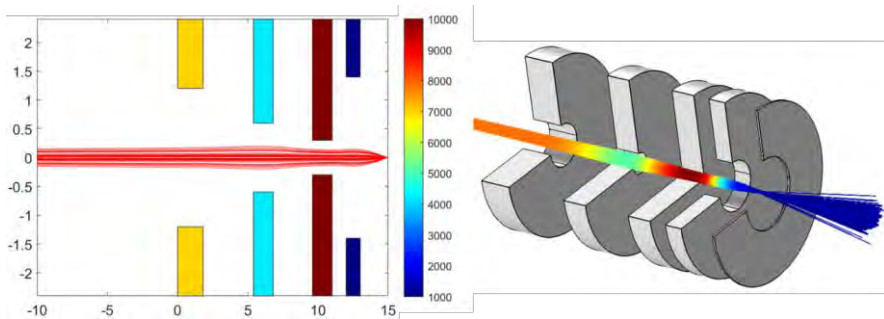


Figure 9.13. The optimised electrostatic lens system (system C’ in figure 9.12.c) including 4 electrodes, optimised by “FOEM-GA”, in 2D (a) and 3D (b).

9.5 Conclusion

In this work, a novel fast field calculation method, with reasonably high accuracy, for cylindrical rotationally-symmetrical electrostatic lens systems is presented. It is

performed by solving the Laplace equation while keeping the terms up to its fourth order derivatives and utilizing quintic splines. This method is therefore called Fourth Order Electrode Method (FOEM) by the authors. First, fundamental equations of the quintic spline are derived for a system with unequal distances between the points, which is new as far as we know. Using the Laplace equation and by means of discretizing the axis-of-symmetry and approximation of the axial potential in each section with a quintic spline, a mathematical equation for calculating the potential along the axis of symmetry is derived. Accuracy and computation time of the proposed method in calculation of the axial potential, its first and second derivatives and related optical parameters, are compared with other field calculation methods, such as the Finite Element Method (FEM) and Second Order Electrode Method (SOEM). As expected, especially the second derivative of the axial potential is calculated much more accurate in FOEM than in SOEM, which is especially important for the calculation of the spherical aberration of a lens system.

The proposed method is then used to optimize an electrostatic lens system by implementing Genetic Algorithm (GA) optimization and the computation time of optimization and effectiveness of evaluated optical parameters are compared with optimization performed by “FEM” as well as “SOEM+FEM”. It is concluded that using “FOEM”, optical parameters calculated by FOEM achieve reasonably high accuracy compared with values obtained by FEM and the optimization based on FOEM can be performed significantly faster (70 times) than by FEM. When the final design found by FOEM optimization is recalculated by FEM, the resulting lens parameters are very close to the results from the much slower FEM optimization.

When the results from the FOEM optimization are fed into the GA optimization with FEM field calculation, it is found that the results only improves marginally. For other systems than our example, this may be different, but based on the experience in the analyzed example it is expected that such a “fine tuning” step can be quite short.

This new method might be of interest to electron optics designers when a fast calculation of the axial field in electrostatic lens systems is needed. The most obvious application is the fully-automated design and optimization of multi-electrode lens systems, with many free parameters and multi-objective functions.

References

- [1] Zworykin, V. K., Morton, G. A., Ramberg, E. G., Hillier, J., and Vance, A. W., *Electron Optics and the Electron Microscope* (John Wiley and Sons, New York, 1945).
- [2] Heddle, D. W. O., *Electrostatic Lens Systems*, Second Edition (IOP Publishing Ltd, Institute of Physics Publishing Bristol and Philadelphia, 2000).
- [3] Szilagy, M., *Electron and Ion Optics* (Plenum, New York, 1988).
- [4] Jin, J., *The Finite Element Method in Electromagnetics* (John Wiley & Sons. IEEE Press, 2014).
- [5] Read, F. H., Bowring, N. J., The CPO programs and the BEM for charged particle optics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. 645, 273-277 (2011).
- [6] Faiz, J., Ojaghi, M., Instructive Review of Computation of Electric Fields using Different Numerical Techniques. *Int. J. Engng Ed.* 18, 344-356, (2002).
- [7] Hesam Mahmoudi Nezhad, N., Ghaffarian Niasar, M., Mohammadi Gheidari, A., Hagen, C. W., and Kruit, P., Multi-electrode lens optimization using genetic algorithms. *Int. J. Mod. Phys. A* 34, 1942020 (2019).
- [8] Dahl, D. A., "SIMION for the personal computer in reflection," *Int. J. Mass Spectrom.* 200 (2000).
- [9] Lencova, B., and Zlamal, J., Electron Optical Design program package EOD 5.001; <https://www.lencova.com/> (2019).
- [10] Van der Geer, S.B., De Loos, M.J., *The General Particle Tracer code : Design, Implementation and Application*, Eindhoven: Technische Universiteit Eindhoven, 2001. – Proefschrift. – ISBN 90-386-1739-9.
- [11] INTEGRATED Engineering Software, <https://www.integratedsoft.com/applications/charged-particle-optics>.
- [12] Hesam Mahmoudi Nezhad, N., Ghaffarian Niasar, M., Mohammadi Gheidari, A., P. Kruit and Hagen, C. W., Multiple criteria optimization of electrostatic electron lenses using multi-objective genetic algorithms. *J. Vacuum Science & Technology B* 39, 062605 (2021).
- [13] COMSOL Multi-physics, version 5.3a, <https://www.comsol.com/release/5.3> (2017).
- [14] Neustock, L. T., Hansen, P. C., Russell, Z. E., and Hesselink, L., Inverse Design Tool for Ion Optical Devices using the Adjoint Variable Method. *Sci. Rep.* 9, 11031 (2019).
- [15] Hesam Mahmoudi Nezhad, N., Ghaffarian Niasar, M., Hagen, C. W., and Kruit, P., Local versus global optimization of electron lens system design, Proceedings of 2020

IEEE 6th International Conference on Optimization and Applications (ICOA) (IEEE, Piscataway, NJ), 9094475 (2020).

[16] Mitchell, M., *An Introduction to Genetic Algorithms* (MIT, Cambridge, 1998).

[17] Hesam Mahmoudi Nezhad, N., *Optical System Optimization using Genetic Algorithms*, Master's thesis, Delft University of Technology, [DOI: 10.13140/RG.2.2.18066.68803](https://doi.org/10.13140/RG.2.2.18066.68803) (2014).

[18] Kennedy, J., and Eberhart, R., Particle Swarm Optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1942-1948 (1995).

[19] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., Optimization by Simulated Annealing, Science, Volume 220, Number 4598, p. 671, (1983).

[20] Nezhad, N. H. M., Niasar, M. G., Hagen, C. W. and Kruit, P., Comparison of Different Optimization Techniques in Electron Lens Design. 2023 9th International Conference on Optimization and Applications (ICOA), Abu Dhabi, United Arab Emirates, 1-5, (2023).

[21] Hesam Mahmoudi Nezhad, N., Ghaffarian Niasar, M., Hagen, C. W., Kruit, P., Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses, 2023 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Winnipeg, MB, Canada, 170-173 (2023).

[22] Adriaanse, J. P., Van der Steen, H. W. G., and Barth, Practical optimization of electrostatic lenses, J. Vac. Sci. Technol. B 7, 651–666 (1989).

[23] Van der Stam, M. A. J., Barth, J. E., and Kruit, P., Design of a multimode transport lens with optimization program SOEM, *Proc. SPIE 2014*, 45, 45-56 (1993).

[24] Barth, J. E., van der Steen, H. W. G., and Chmelik, Improvement to the electrostatic lens optimization method SOEM, J., *Proc. SPIE 2522*, 128, 128-137 (1995).

[25] Hesam Mahmoudi Nezhad, N., Ghaffarian Niasar, M., Mohammadi Gheidari, A., Janssen, T., Hagen, C. W., and Kruit, P., Recent Trends in Charged Particle Optics and Surface Physics Instrumentation (Proceedings of 16th International Seminar), edited by F. Mika et al., (MJ servis spol. s.r.o., Brno, Czech Republic, 2018), 26-28 (2018).

[26] Yuan Ying, T. & Saaban, A., Quintic Spline Method for Solving Linear and Nonlinear Boundary Value Problems, O. Alayed, Sains Malaysiana 45(6), 1007–1012, (2016).

[27] Barth, J. E. and Kruit, P., Addition of different contributions to the charged particle probe size, *Optik 101*, 101-109 (1996).

[28] Matlab, version r2016 b, <https://nl.mathworks.com/products/matlab.html>.

10. Fast Electrostatic Lens System Design (FELSD) Program

In the previous chapters, different methods for field calculation of electrostatic lens systems (fast and reasonably accurate with SOEM or FOEM, or slow and accurate with COMSOL) were presented. Using them, the required optical parameters for an electrostatic lens system to evaluate the quality of the lens system could be calculated in an automated way. Thereafter, some techniques for optimization of lens systems were proposed and successfully implemented which could design/optimize electrostatic lens systems in a fully-automated way. The details of the program designed to achieve the above-mentioned tasks are presented in this chapter. Since our program enables the fast design of an optimized electrostatic lens system, it is named the ‘Fast Electrostatic Lens System Design (FELSD)’ program.

The ‘FELSD’ program can be used for different scenarios. One scenario (we call it S-No.1) is when there is a known lens system and the user only aims to calculate the optical parameters of this system. (S-No.2) is when there is a known system but in addition to wanting to know the optical parameters of that system, the user intends also to improve the lens system quality. A third scenario (S-No.3) is when no initial known lens system is at hand, but there are some criteria (and/or constraints) which affect the user searching for a lens system that has the best quality within the defined boundaries.

The program therefore consists of two parts; “Single Lens System Calculation” and “Lens System Optimization”. For the above-mentioned first scenario (S-No.1), the former option (“Single Lens System Calculation”) can be used. In the situation where optimization of a specified lens system is needed (i.e. S-No.2 and S-No.3), the latter option (“Lens System Optimization”) is implemented. These two options are explained in detail in the following two sub-sections 10.1 and 10.2.

Note: The codes are written in MATLAB 2018b The program is converted into a user friendly app format to make the implementation easier for external users (called FELSD App). The FELSD App can be executed in the MATLAB environment.

10.1 “Single Lens System Calculation” (SLC)

To calculate the optical parameters of a lens system various input parameters are needed. The data related to the lens itself includes the lens geometry and voltages at

each electrode. Moreover, the data related to the optics of the ray-tracing for the lens system, including the position from where the PB is coming, the opening angle, the angles and energies at which SEs are coming from the sample plane (if SE tracing is needed), and the step sizes for the meshing of the lens space, etc. In our program, the data related to the lens itself is called “Lens Parameters”, and the latter category which are related to the ray-tracing is specified by “Initial Data”.

As seen in the flow diagram illustrated in Figure 10.1 for the “Single Lens System Calculation” (SLC) by feeding the input data into the program, the lens can be first sketched for the sake of visualization. Then, the field can be calculated by SOEM, FOEM or COMSOL based on the user’s aim. Thereafter, having the electric field, by implementing the electron-optical formulas (discussed in chapters 4 and 5), the ray tracing can be derived and optical parameter calculations are performed by the program. Finally the output data will be extracted. The output data can include the spherical and chromatic aberrations, the probe size of the PB, the detection deficiency of the lens system, etc. The input and output data are presented in detail in Tables 10.1-10.2.

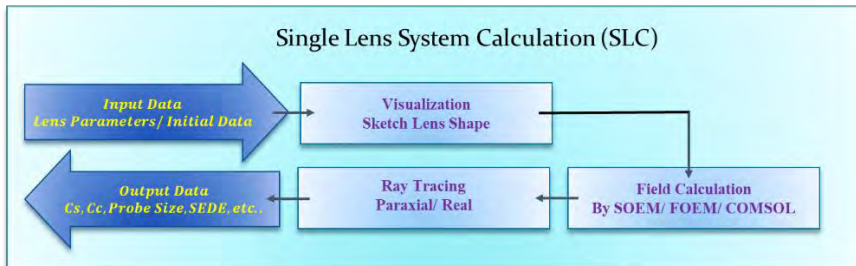


Figure 10.1. Schematic flowchart of “Single Lens System Calculation” of the FELSD program.

The option SLC can be performed utilizing a fast but less accurate method (by SOEM, or FOEM) or a relatively slow but highly accurate method of FEM (using COMSOL). The related flowcharts are different. These flowcharts are given below as Figure 10.2 (for SOEM/FOEM) and figure 3 (for COMSOL).

As illustrated in Figure 10.2, to calculate the SLC by SOEM or FOEM, first the meshing along the optical axis will be performed (box SLCF2 in Figure 10.2). Note that the approach to meshing is application-dependent. Detailed explanations for various applications are provided in Section 3. For additional insights into ‘application 1’, refer to equation 3.24. Then, the field will be calculated using the theory of SOEM or FOEM at the mesh points. For the meshing stage, our MATLAB program meshes the axis along the optical axis (Z-axis) based on the step size Δz which is given as an input

parameter by the user for the PB calculation. The first and second derivatives of the field are then derived numerically at the specified mesh points. Thereafter, the ray-tracing in a paraxial regime is performed (based on the equations in section 2) for the PB. The output data will be extracted at the end.

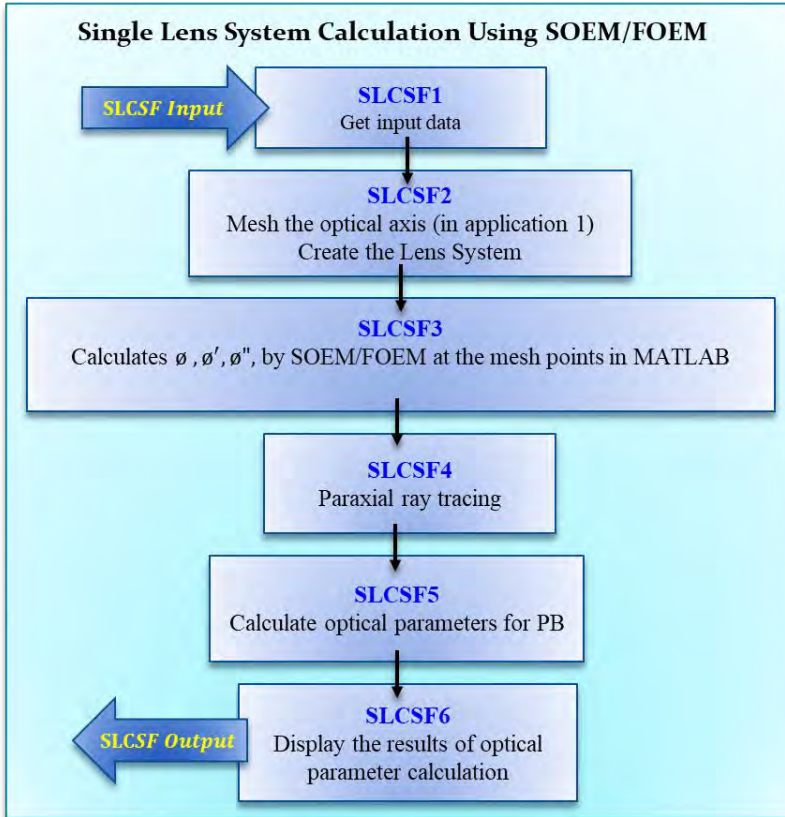


Figure 10.2. Flow diagram of the calculation of the optical parameters for a single lens system, using SOEM/FOEM for the field calculation.

The flow diagram for SLC by COMSOL is illustrated in Figure 10.3. To calculate the SLC using COMSOL, first COMSOL itself automatically creates the meshes in the whole space (box SLCC3 in Figure 10.3). Afterwards, it calculates the field and derivatives in the space as well as along the optical axis. Using our MATLAB program connected to COMSOL, the calculated field and derivatives, both along the optical axis and in the space, are taken as its output data.

However, in the field calculation by COMSOL an intermediate stage is needed for the meshing (box SLCC2 and SLCC4 in Figure 10.3). Note that the meshing strategy is

contingent on the specific application. Section 3 offers explanations on different applications. For more insights into ‘application 1’ and ‘application 2’, refer to equation 3.24 and 3.25 accordingly. The mesh points defined in MATLAB should be connected to the corresponding mesh points created in COMSOL. This step is performed as follows; when doing real ray tracing using COMSOL, first the data obtained by COMSOL in the space of the lens system must be mapped to a uniform rectangular grid. The reason for this is that data directly obtained from COMSOL are only available on the mesh points that are used in COMSOL. These mesh points are not regular and complicated interpolation is needed to calculate the electric field in the whole space to be used for real ray tracing. Therefore first the data on the irregular mesh of COMSOL is mapped to a regular grid. This is done through an interpolation function available in MATLAB. This data is then used to interpolate the electric field in any point of space when doing real ray tracing. When doing ray tracing, first the location of the electron beam in the next time step is determined. Based on this it is known which element of the rectangular grid must be used for the interpolation. The interpolation in the element of interest is simply conducted using a linear weighted interpolation of values of electric field on the four corners of the rectangular section. The weights used in the interpolation are proportional to the distance between the point of interest and the four corners of the rectangular section.

In this way the field and its derivatives along the optical axis, as well as in the whole space, are known at the specified mesh points as defined in MATLAB. The former is used in the PB tracing and the latter is used for SE tracing calculation. The details of this approach can be found in section 5.

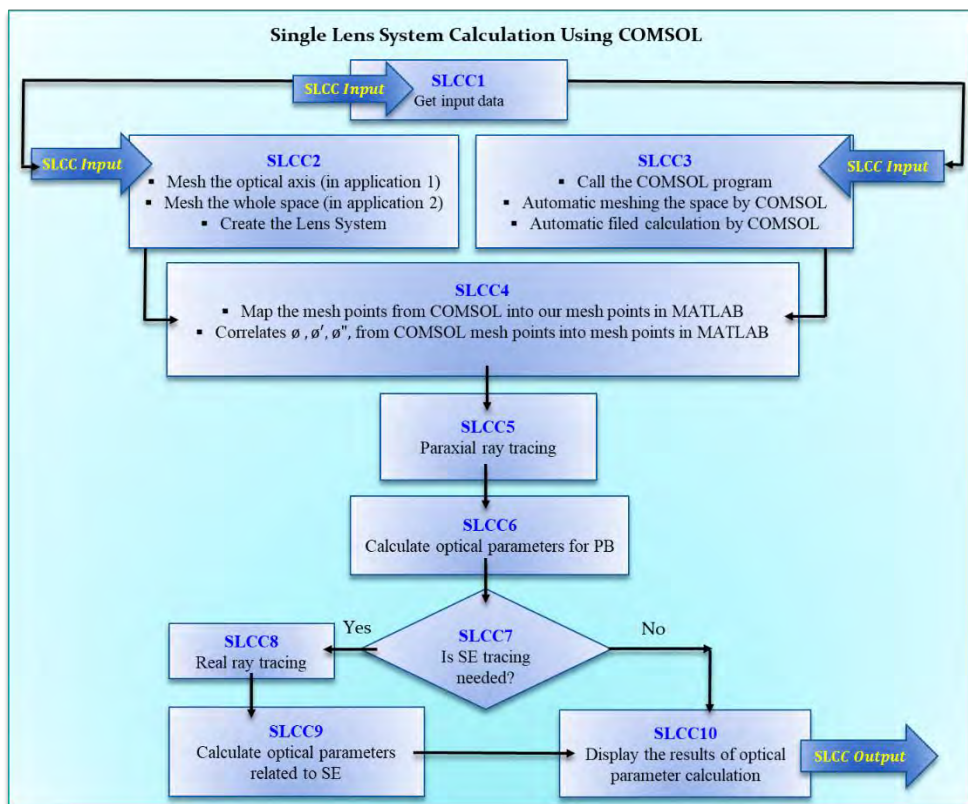


Fig. 10.3. Flow diagram of the calculation of the optical parameters for a single lens system, using COMSOL for the field calculation.

The related user-friendly app developed in MATLAB, is schematically shown below in Figure 10.4. To select the option SLC first the tab “Single” should be selected. The input data, can be inserted manually by the user into the on-screen boxes or it can be automatically read from a pre-provided text/Excel file by choosing “Load input Data”. Then, to choose field calculation by SOEM or FOEM, the button in orange of “SOEM” or “FOEM” can be pressed. If the field is to be calculated by COMSOL, the green button “calculation single system” should be pressed. After pressing these buttons, the output data will be calculated and illustrated in the output data box. Details of the input data related to the lens (“Lens Parameters”), including their notations and dimensions, are given in Table 10.1, and details of additional input data needed for the simulation and ray tracing are given in Table 10.2

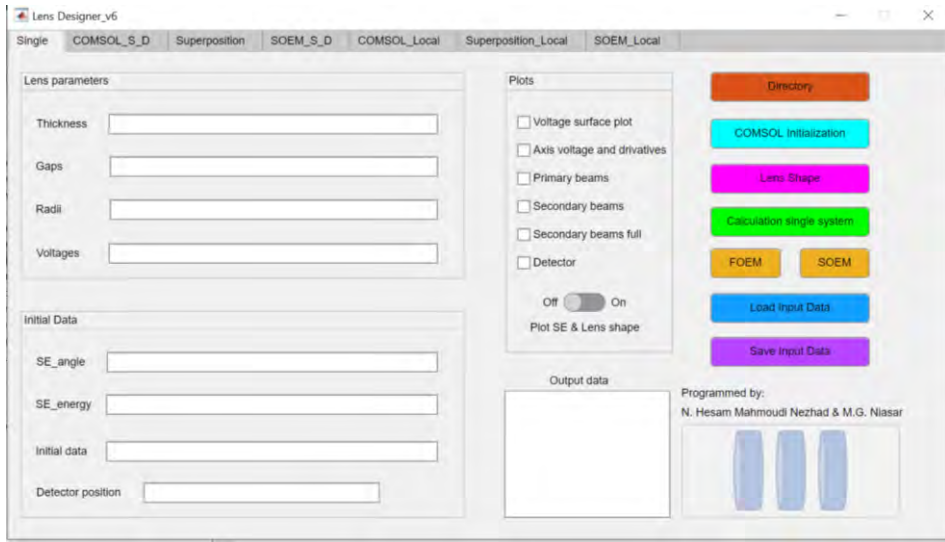


Figure 10.4. Schematic of the FELSD App, illustrating the page related to the “Single Lens System Calculation” option.

Table 10.1. Input data related to the lens geometry and voltages (“Lens Parameters”)

“Lens Parameters”	Notations/ Dimensions
Thickness	$[T_1 T_2 T_3 \dots T_{wafer}] / \text{mm}$
Gaps (spaces between two sequential electrodes)	$[G_2 G_3 \dots \text{Gap}_{\text{Lens\&wafer}}] / \text{mm}$
Radii	$[R_1 R_2 R_3 \dots R_{wafer}] / \text{mm}$
Voltages at each electrode	$[V_1 V_2 V_3 \dots V_{wafer}] / \text{V}$

Note1: The number of electrodes can be changed to any desired number. The shape of the electrode for now is cylindrical, hence each electrode can be defined by a fixed radius (the distance of the electrode from the optical axis). The input data which are needed to calculate the ray-tracing for the primary and secondary electron beam (“Initial Data”)are given in Table 10.2.

Note 2: In the provided data in table 10.1, the term 'wafer' refers to the sample. In table 10.2, 'width of the electrode lens system' (W_{el}) denotes the radial distance from the optical axis to the upper surface of the electrodes. Note3: The SLCC-Input/Output, and the SLCSF-Input/Output are the same data as the SLC-Input/Output-Data which are provided in Tables 10.1 and 10.2.

Table 10. 2. Input data related to the ray tracing (“Initial Data”)

“Initial Data”	Notations /Dimensions
The position from which PBs are coming (reference is the surface of the first electrode, counting from the left)	X_0 , mm
Position until where the PBs are sketched in the MATLAB figures	X_{end} , mm
Step size along the z axis (dz)	dz, mm
Length of the lens system	L_{system} , mm
The detector position (reference is the surface of the first electrode)	$X_{detector}$, mm
The distance before the first electrode where the field is closed in COMSOL	$X_{befclosedCOM}$, mm
The distance after the wafer(sample), where the field is closed in COMSOL	$X_{afclosedCOM}$, mm
Width of the electrode lens system	W_{el} , mm
The time step for the real ray tracing calculation (e.g here 1e-13 sec)	Sec
The step size of the mesh along the z-coordinate for the real ray tracing.	dz, mm
The step size of the mesh along the radial coordinate for the real ray tracing.	dr, mm
The number of iterations to calculate real ray tracing	Dimensionless
Maximum allowed electric field	E_{max} , V/mm
Opening angle at the image side	Alpha, rad
Maximum allowed electric field between two sequential electrodes	E_{maxeli} , V
The angles at which SEs are coming from the sample plane	SE_{angle} , rad
The energies at which SEs are coming from the sample plane	SE_{energy} , eV
The distance behind the wafer where the electric field is closed in COMSOL	X_{close} , mm

10.2 Lens System Optimization (“LSO”)

To optimize a lens system, the user can choose to use local optimization (the so-called ‘Fmin/Fmincon’ function in MATLAB, explained in section 6.3) or a global optimizer (for now, the Genetic Algorithm), with the possibility of varying all lens geometries and voltages of the electrodes. It is also possible to only vary the voltages at the electrodes and use the superposition optimization.

The superposition option works as follows: in the first step before the optimization starts, a set of base calculations is performed. 1 volt is applied to electrode 1 while all the other electrodes are assigned with zero volts. The lens system is meshed and the field is calculated (here by COMSOL) throughout the lens system. The data (electric field) is stored in a matrix where its element index represents the r-z coordinates of the lens system. The same procedure is repeated by assigning 1 volt to the next

electrode while all the other electrodes are at zero volt. A matrix of data corresponding to each of these calculations is stored (for an N-electrode lens system the procedure will be repeated N times and at the end we will have N matrices of field data at each mesh point of the lens system). In the next stage the optimizer produces a set of initial data for which all the geometry data is exactly the same as was used in the first stage, but the voltages of the electrodes are assigned different values for each system under investigation. To obtain the electric field in the whole space for a set of voltages, the base matrix of each electrode is multiplied by its voltage value after which the modified matrices of all electrodes are added together. Since the base matrices are already available, this addition can be performed very quickly. Having the electric field in the space, the program can calculate the optical parameters. This procedure is repeated throughout the entire optimization. This optimization is particularly useful when a roughly optimized lens geometry is available, and the user's goal is to make slight improvements to the lens system by adjusting voltages or to achieve a quick auto-focus.

The standard optimization procedure, as discussed earlier (where all geometry and voltages serve as free parameters in the optimization), is simply referred to as 'Lens System Optimization' (LSO), denoting optimization not based on superposition. Users can choose from SOEM, FOEM, and COMSOL to calculate the field. The related flowchart of how LSO works is given below in Figure 10.5. As shown in the flowchart Figure 10.5, the lens system optimization starts by feeding LSO-input data into the program. In LSO, the input data are included; a single initial lens system (if the optimization is a local optimizer or multiple initial lens systems (if it is a global optimizer) to start the optimization with, or, the parameters of the optimization procedure (e.g. the tuning parameters of GA such as number of generations, population size, etc.), the objective function, the constraints, boundaries, etc.

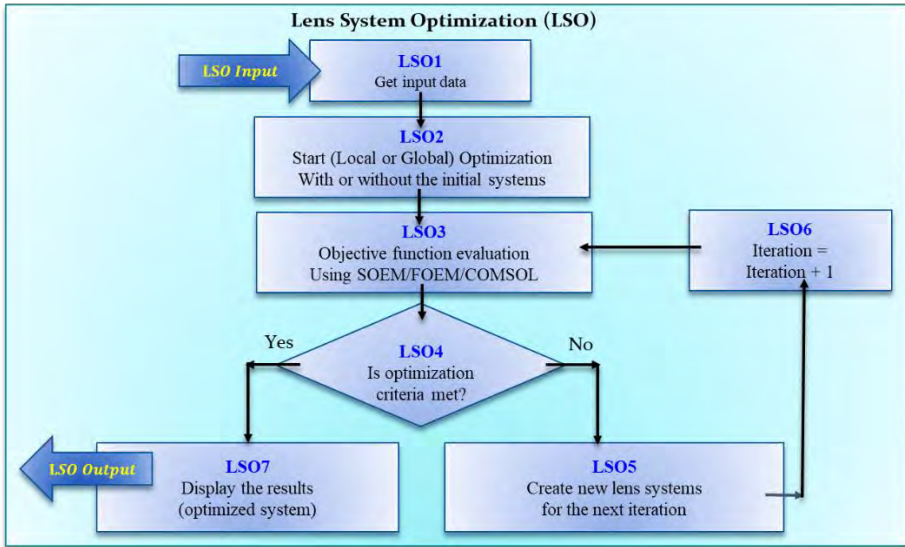


Figure 10.5. The program’s flow diagram of lens system optimization.

To select the optimization method when the field calculation is to be performed by COMSOL, and the optimization by a global optimizer such as GA, the user should press the “COMSOL_S_D” button. For optimization with the field calculation by SOEM, and a global optimizer, select the “SOEM_S_D” button. If a local optimizer is preferred instead of a global optimizer then the user should select the “COMSOL_Local” or “SOEM_Local” buttons. If the user only wants to vary the voltages for a fixed lens geometry, and optimize using superposition, the “Superposition” button should be selected. In the above-mentioned cases, since the appearance of the App page is similar, but only the tabs at the top are different, only the page related to the “COMSOL_S_D” is shown as representative for the global optimization. For the other options the appearance is similar to this (Fig. 10.6 top figure). For local optimization, the page related to “SOEM_Local” is illustrated as representative for local optimization (Figure 10.6, yellow button at the bottom of the figure).

The objective function can be defined using the parameters of A, B, C and D, and the formula given in the relevant box as shown in Figure 10.6. The input data for the optimization, including the lens parameters (if needed) and optimization tuning parameters can be manually inserted. These can also be read from a loaded pre-prepared text file. The optimization is then run by pressing the “Perform GACO_D optimization” button.

Note: The LSO-Input/Output, are the data of the SLC Input/Output-Data provided in Table 10.1 and 10.2, in addition to the Initial data related to the optimization procedure explained in the above.



Figure 10.6. Schematic of the FELSD App, illustrating the page related to the “Lens System Optimization” options.

11. Summary and Conclusion

The design of electrostatic charged particle lenses involves changing many geometrical parameters of the lens electrodes as well as changing the voltages at each lens electrodes. The objective of the design is often to get the electrons passing through the lens system to be focused at a specific point and to minimize the aberrations of the lens. To make such a design is a laborious task for electron optical lens designers. A fully-automated optimization routine to relieve the laborious manual design of charged particle lens systems has been demanded for years, however, to achieve this outcome while many free optimization parameters are involved in the lens system design is quite a challenge. This is mainly due to calculations of the lens electric potential which are in general carried out with very time-consuming techniques that require meshing of the lens space. Currently it is not difficult to find the electron-optical software to conduct accurate field calculations such as EOD, GPT, CPO, Simion, etc., that can be used in an optimization loop. However, it can take months to get the results. For instance, the design of even a simple system using COMSOL takes such a long computational time that the designer might not have the patience to wait for the result (assuming the computational memory allows them to do so). Therefore, although some charged particle optics design programs exist which could change a few geometrical parameters of the lenses or the voltages (EOD, GPT, CPO, Simion, etc.), a fully-automated optimization routine which could make a design where all geometries and voltages of the lenses could be varied in a feasible time did not exist. A first attempt (SOEM) was made more than 30 years ago, but it still had too many limitations to be widely used. The main objective of this thesis therefore was to find a technique which would enable electron optical designers to tackle this problem.

To achieve this aim, many questions had first to be studied and answered. The first question was “What is the main limitation which makes having a fully-automated optimization as yet infeasible and how can it be tackled?”. The main challenge, as mentioned above, is the high computational time of the existing field calculation method for each lens system evaluation, while the total number of system evaluations are very long (in the order of thousands for a typical multi-electrode lens system with few electrodes for all geometries and voltages of the system as free parameters of the optimization). To evaluate so many systems using the existing field calculation methods, an infeasible amount of time and computational memory is required To

overcome this difficulty, we were seeking for another method of field calculation rather than the existing ones including FEM (Finite Element Method), BEM (Boundary Element Method) and FDM (Finite Difference Method).

A new method is proposed in chapter 4 which could improve on the low speed of the field calculation while keeping the high accuracy of the mesh-based calculation methods. The method is to first use a fast field calculation based on the so-called Second-Order Electrode Method (SOEM) (background is presented in chapter 2), at the cost of losing some accuracy, and then to use a Genetic Algorithm (GA) for the optimization (background provided in chapter 3). Afterwards, by using the data related to the approximate lens systems found from this optimization based on SOEM (called SOEM-GA), another GA optimization routine is performed based on accurate field calculation method of FEM using the commercial multi-physics software package of COMSOL (background is provided in chapter 2). Using this method an example of six-electrode electrostatic lens system was optimized within some hours where all lens geometries and electrode voltages were as the free optimization parameters and the focus position and maximum allowable electric fields between electrodes were as constraints.

The next question was “Can a local optimization technique be implemented in the place of a global optimization technique such as GA to design the electrostatic lens systems?”. This question is answered in chapter 5. There, both a local, namely “Fmin” (background in chapter 3), and a global optimization technique, namely GA, is implemented on a group of typical electrostatic lens systems with 6 electrodes, to investigate whether the objective function landscape of such a design optimization is a local problem or a global one. It is illustrated in chapter 5 that the search space of our case-studies has multiple local minima. Implementing a local optimizer is hence not sufficient to reach to the optimum lens system and a global optimization is necessary to be used instead. It is also discovered that the GA performs as a robust global optimizer which could handle this complex optimization problem having many local minima.

The optimization problems discussed until chapter 5, was a single-objective optimization problem, meaning that if there exist more than one objective function which are non-conflict objective functions they could be combined with different weights to form one single objective function. However, there are some situations with two or more conflicting objective functions such as having both PB (Primary Beams) focusing at the image plane and having the highest detection efficiency for the

SE (Secondary Electrons) emerged from the image plane, simultaneously. In such situations it is better to use the multi-objective optimization techniques such as multi-objective genetic algorithm (MOGA) . The question was then whether MOGA can handle such complex design (electrostatic lens systems design while there exist multiple objective functions in addition to having all geometries and voltages as free parameters of the optimization) or not. This subject is studied in chapter 6.

The strategy was similar to the SOGA (Single-Objective Genetic Algorithm) to use first a much faster, but less-accurate field calculation technique using SOEM combined with MOGA. Afterwards, to use approximately good systems as input for a COMSOL-based MOGA optimization. Without implementation of this strategy for MOGA, there was no way to find whether this method would also work efficiently for MOGA as it did for SOGA. A typical electrostatic lens system including 5 electrode lenses and two objective functions, was taken to try this strategy. The aim of the design problem was to focus a PE beam into a fixed image plane with minimum probe size and to attract SE emitted from the image plane back into the lens on to an SE detector with the highest detection efficiency. It is illustrated that by using the above-mentioned strategy with a surrogate-assisted multi-objective genetic algorithm (MOGA), electrostatic lens systems could be successfully optimized in a reasonably short time of a few hours, compared to many days when using COMSOL-based MOGA only. This new method of MOGA can be hence implemented in electrostatic lens design with one or more objective functions and multiple free variables, as a very efficient, fully automated optimization technique.

Note: Since in electrostatic lens system optimization/design problems more general case-studies involve non-conflicting objective functions, our main focus for the rest of the work is yet in using SOGA. However, MOGA is tested and presented in chapter 6 for other situations if needed.

Another important study was on tuning the GA parameters which is discussed in chapter 7. Having implemented a Genetic Algorithm optimization for electrostatic electron lens design allowed us to perform a study on the influence of GA tuning parameters. The study is performed on an example of multi-electrostatic lens system including 6 electrodes which has 23 free variables. The extension to more complex designs is straightforward. First, an investigation is performed of the influence of the values of population size and number of generations. As expected, by increasing both values the optimization achieved better results. However, it has been shown that a population of 50 with 100 generations could provide reasonably good results in our-

case studies. Increasing to higher values of population and generation did not significantly improve the results while the correspondent run time was dramatically increased. Moreover, it is also discovered that having a fixed computational effort (time or total number of system evaluations), a lower population size with larger generation size usually offers better results compared to the reverse case. An analysis also is performed on the Crossover and Mutation types as the main tuning parameters of GA to find the most suitable options. It is illustrated that irrespective of the type of Crossover, the Mutation Gaussian achieves the best result. It is also found that among different crossover types, Crossover Heuristic achieves the best performance. It is also discovered that, the combination of Crossover Heuristic with Mutation Gaussian would provide significantly better results than all other combinations of Mutations and Crossovers. Moreover, an analysis is performed on the Crossover Fraction which led to the result that a Crossover Fraction of 0.5 or 0.6 achieve the best results. The guidelines provided in chapter 7 for tuning the GA parameters which might be helpful not only for the optimization of electrostatic lens designs, but also for other GA optimization of functions with similar complexity.

In chapter 8, different meta-heuristic optimizations algorithms such as Genetic Algorithm, Particle Swarm Optimization and Simulate Annealing also a calculus based local minimization were implemented and compared. The analysis is performed on a typical lens system with six electrodes (23 free variables). It was shown that the population-based meta-heuristic optimization namely, PSO and GA generate significantly better outcomes compared to the single-based metaheuristic optimization (namely SA), and calculus based local minimization (namely "Fmin"), while PSO offers slightly better results than GA. Moreover, from the analysis performed on SA and "Fmin", it is discovered that these two optimizations work as a sort of local optimization which therefore better not to be implemented in our case-study where the constraints and objective functions are complex with multiple local minima. The comparison between these two local optimizations illustrated that the metaheuristic one (SA) outperforms the calculus-based one (Fmin).

In chapter 9 a novel method of fast field calculation is presented. As mentioned previously, in all other areas of computations and simulations in electron optics, calculation of the electric field plays a major role. Accurate field calculation methods such as the Finite Element Method (FEM), Boundary Element Method (BEM) and Finite Difference Method (FDM), have been used for years. However, such methods are computationally very expensive and make the computer simulation challenging or

even infeasible when trying to apply automated design of electrostatic lens systems with many free parameters. Hence, for years, electron optics scientists have been searching for a fast and accurate method of field calculation.

Second Order Electrode Method (SOEM) which was introduced in 1980s by Adriaanse is based on truncation of the Laplace equation from the second term. Satisfactory accuracy and very fast field calculation (0.1 second for each system) could be achieved. However, some deviation in optical parameters was observed compared to the case where electric field is calculated by means of accurate models such as FEM. A novel method of fast field calculation is for the first time developed by the authors which is presented in chapter 9. This method offers much higher accuracy in field calculation than the previously presented method of SOEM, but yet at a very short computation time (0.5 second for each system) similar order as the SOEM. The field calculation formulation is performed by solving the Laplace equation near the optical axis by keeping the terms in the off-axis potential expansion up to the fourth order derivatives of the axial potential with respect to the optical axis coordinate. This method is hence named by the authors Fourth Order Electrode Method (FOEM). Optical parameters obtained by FOEM are close to the values calculated by FEM and considering the reduction of ~ 100 in calculation time, it also provides a good basis for even a faster optimization technique than the previously presented one (namely "SOEM-COMSOL-GA") for electrostatic lens system design.

This recently proposed method is implemented on the optimization of typical electrostatic lens systems including 6 electrode lenses using the GA as the optimization algorithm. The effectiveness of evaluated optical parameters and the computation time of optimization are compared with optimization performed by "FEM" as well as by "SOEM-FEM". It is illustrated that using "FOEM", optimization was performed significantly faster (70 times) and optical parameters calculated by FOEM achieve reasonably high accuracy compared with values obtained by FEM. This new method might be of interest to electron optics scientists where a fast calculation of field for electrostatic lens systems is needed. This method also enables electron optical designers, after searching for a solution for many years, to design/optimize multi-electrode lens systems, with many free parameters of the lens system, and multi-objective functions, in a fully-automated optimization process.

By implementation of all the above-mentioned theories and formulations in MATLAB, a software was developed which has the capability of calculating optical parameters using field calculation based on FEM as well as calculating optical parameters using

fast field calculation methods such as SOEM and FOEM. The software is able to calculate optical parameters of a single lens system, as well as perform full global based optimization such as GA and PSO utilizing SOEM and FOEM and FEM as well as doing local optimization such as SA and “Fmin”. Furthermore, superposition based field calculation method using accurate FEM field calculation is introduced in the software with which a fixed geometry lens system can be optimized over a given voltage range with full accuracy and high speed. The flowcharts and the detailed explanations for the user-friendly software are presented in chapter 10. Since our program can perform the design/optimize of an electrostatic lens systems fast, it is named ‘Fast Electrostatic Lens System Design (FELSD)’ Program. The ‘FELSD’ program can be implemented for different scenarios. One scenario is when there is a known lens system and the user only aims to calculate the optical parameters of this system. The second scenario is when there is a known system and the user intends to improve the lens system quality. A third scenario is when no initial known lens system is in hand, but there are some criteria (and/or constraints) which affect the user searching for a lens system that has the best quality within the defined boundaries. The program is therefore comprised of two main parts; “Single Lens System Calculation” and “Lens System Optimization”. The codes are written in MATLAB 2018b. The FELSD App can be executed in MATLAB environment.

To summarize, the main contribution of our work is the presentation of a fully-automated optimization technique (which was also the main objective of this research work) for the electro optics community based on SOEM, and more significantly FOEM, combined with an evolutionary algorithm such as GA or PSO. The presented technique enables electron optical designers, who have searched for a solution for many years, to design/optimize complex multi-electrode lens systems, with many free parameters of the lens system and multi-objective functions, in only a few hours. Currently it is not difficult to find the electron-optical software to conduct accurate field calculations such as EOD, GPT, CPO, etc., that can be used in an optimization loop however, it can take months to get the results. For instance, the design of even a simple system using COMSOL takes such a long computational time that the designer might not have the patience to wait for the result (assuming the computational memory allows them to do so). However due to its fast field calculation and a fast optimization, the technique presented in this thesis enables designers to easily change the requirements/constraints of the design by seeing the result quickly and then deciding whether to run it again with other conditions. By creating this possibility, our software to offers

charged particle optics designers anywhere in the world an efficient tool to develop a fast and efficient charged particle electrostatic lens design for the first time.

As the main future works, there can be other additional options to be added to the presented software to make it yet a more general charged particle optics design program package. The additional options can be for instance to have any geometry shape of the lenses possible in the design, in place of only having the rectangle lenses. Also, to have the magnetic lenses in the design in addition to presence of electrostatic lenses. Moreover, as one of the very useful application of this software can be to implement it for design and optimization of multi-electron beam lens systems. This application has been so far partly performed and preliminary successful results are achieved. A full design of multi-electron beam lens system using the presented software is hence planned and under progress.



12. Samenvatting en Conclusies

Het ontwerp van elektrostatische geladen-deeltjeslenzen omvat het veranderen van veel geometrische parameters van de lenselektroden en het veranderen van de spanningen op elke lenselektrode. Het doel van het ontwerp is vaak om de elektronen die door het lensstelsel gaan op een specifiek punt te focuseren en de aberraties van de lens te minimaliseren. Een dergelijk ontwerp is lastig voor ontwerpers van elektronenoptische lenzen. Er is al jaren vraag naar een volledig geautomatiseerde optimalisatieroutine om het moeizame handmatige ontwerp van lenssystemen onnodig te maken. Het is echter een behoorlijke uitdaging om dit resultaat te bereiken omdat er veel vrije optimalisatieparameters betrokken zijn bij het ontwerp van een lensstelsel. Dit is voornamelijk het gevolg van de berekeningen van de elektrische potentiaalverdeling in de lens, die over het algemeen worden uitgevoerd met zeer tijdrovende technieken die meshing van de lensruimte vereisen. Hoewel er ontwerpprogramma's voor geladen-deeltjesoptica bestaan die enkele geometrische parameters van de lenzen of de spanningen ervan kunnen variëren (EOD, GPT, CPO, Simion, enz.), bestond er nog geen volledig geautomatiseerde optimalisatieroutine die het ontwerp kon maken binnen een redelijke tijd, inclusief de mogelijkheid om alle geometrieën en spanningen van de lenzen te variëren. Meer dan 30 jaar geleden werd een eerste poging gedaan (SOEM), maar die had nog te veel beperkingen om breed ingezet te worden. Het hoofddoel van dit proefschrift was dan ook het vinden van een techniek die ontwerpers van elektronenoptica in staat zou stellen dit probleem aan te pakken.

Om dit doel te bereiken moesten veel vragen worden beantwoord. De eerste vraag was "Wat is de belangrijkste beperking waardoor een volledig geautomatiseerde optimalisatie nog niet haalbaar is en hoe kan dit worden aangepakt?" De belangrijkste uitdaging, zoals hierboven vermeld, is de lange rekentijd van de bestaande veldberekeningsmethoden voor elke lensstelsel-evaluatie, terwijl het totale aantal systeemevaluaties zeer hoog is (in de orde van duizenden voor een typisch multi-elektrode lensstelsel met enkele elektroden, waarbij alle geometrieën en spanningen van het systeem als vrije parameters van de optimalisatie worden beschouwd). Om zoveel systemen te evalueren met behulp van de bestaande veldberekeningsmethoden, is een onacceptabele hoeveelheid tijd en rekengeheugen vereist. Om dit probleem op te lossen, waren we op zoek naar een andere methode

van veldberekening in plaats van de bestaande, waaronder FEM (Finite Element Method), BEM (Boundary Element Method) en FDM (Finite Difference Method).

In hoofdstuk 4 wordt een nieuwe methode voorgesteld die de snelheid van de veldberekening zou kunnen verhogen, terwijl de hoge nauwkeurigheid van de mesh-gebaseerde berekeningsmethoden behouden blijft. De methode houdt in eerst een snelle veldberekening te gebruiken op basis van de zogenaamde Second-Order Electrode Method (SOEM) (de achtergrond hiervan wordt gepresenteerd in hoofdstuk 2), hoewel dit leidt tot enig verlies van nauwkeurigheid. Hierbij wordt gebruik gemaakt van een genetisch algoritme (GA) voor de optimalisatie (de achtergrond wordt toegelicht in hoofdstuk 3). Daarna wordt, door uit te gaan van de bij benadering bepaalde gegevens van de lenssystemen die zijn gevonden uit de optimalisatie op basis van SOEM (genaamd SOEM-GA), een andere GA-optimalisatieroutine uitgevoerd op basis van de nauwkeurige veldberekeningsmethode FEM, met behulp van het commerciële multi-physics softwarepakket COMSOL (de achtergrond wordt besproken in hoofdstuk 2). Met behulp van deze methode werd binnen enkele uren een elektrostatisch lensstelsel met zes elektroden geoptimaliseerd, waarbij alle lensgeometrieën en elektrodespanningen de vrije optimalisatieparameters vormden en de focuspositie en maximaal toegestane elektrische velden tussen elektroden de beperkingen waren.

De volgende vraag was: “Kan een lokale optimalisatietechniek worden geïmplementeerd, in plaats van een globale optimalisatietechniek zoals GA, om de elektrostatische lenssystemen te ontwerpen?” Deze vraag wordt beantwoord in hoofdstuk 5. Daar wordt zowel een lokale optimalisatietechniek geïmplementeerd, namelijk Fmin (de achtergrond wordt gegeven in hoofdstuk 3), als een globale, namelijk GA, op een groep typische elektrostatische lenssystemen met zes elektroden. Het doel hiervan was om te onderzoeken of het optimaliseringslandschap van zo’n ontwerp een lokaal of een globaal minimum bevat. In hoofdstuk 5 wordt aangetoond dat de zoekruimte van onze casestudies meerdere lokale minima heeft. Het implementeren van een lokale optimalisatie is daarom niet voldoende om het optimale lensstelsel te bereiken. In plaats daarvan is een globale optimalisatie nodig. Ook werd duidelijk dat de GA presteert als een robuuste globale optimizer, die dit complexe optimalisatieprobleem met veel lokale minima aankan.

De optimalisatieproblemen die tot hoofdstuk 5 zijn besproken, waren een optimalisatieprobleem met één doel, wat betekent dat als er meer dan één optimalisatie functie bestaat, en deze niet met elkaar conflicteren, deze kunnen

worden gecombineerd met verschillende gewichten om één enkele optimalisatiefunctie te vormen. Er zijn echter situaties met twee of meer tegenstrijdige optimalisatiefuncties, zoals in het geval van tegelijkertijd primaire bundels die scherpstellen op het beeldvlak, als de hoogste detectie-efficiëntie voor de secundaire elektronen die uit het beeldvlak komen. In dergelijke situaties is het beter om multi-functie optimalisatietechnieken te gebruiken, zoals het multi-functie genetische algoritme (MOGA). De vraag was daarom of MOGA dergelijke complexe ontwerpen aankan (dat wil zeggen het ontwerpen van elektrostatische lenssystemen met meerdere optimalisatiefuncties, waarbij alle geometrieën en spanningen als vrije parameters van de optimalisatie worden beschouwd). Dit onderwerp wordt behandeld in hoofdstuk 6.

De strategie was vergelijkbaar met de SOGA (Single-Objective Genetic Algorithm): eerst werd een veel snellere, maar minder nauwkeurige veldberekeningstechniek gebruikt met SOEM in combinatie met MOGA. Vervolgens werden bij benadering juiste systemen gebruikt als input voor een MOGA-optimalisatie in COMSOL. Zonder de implementatie van deze strategie voor MOGA was het onmogelijk om uit te zoeken of deze methode ook voor MOGA efficiënt zou werken, zoals voor SOGA. Om deze strategie uit te proberen, werd een typisch elektrostatisch lensstelsel gemaakt met vijf elektroden en twee optimalisatiefuncties. Het doel van het ontwerpprobleem was om een primaire bundel te focuseren op een vast beeldvlak met een minimale spotgrootte en om secundaire elektronen die vanuit het beeldvlak werden geëmitteerd terug door de lens, te sturen naar een detector met de hoogste detectie-efficiëntie. Er werd aangetoond dat met deze strategie, elektrostatische lenssystemen met succes konden worden geoptimaliseerd in een redelijk korte tijd (enkele uren), vergeleken met vele dagen bij gebruik van alleen MOGA in COMSOL. Deze nieuwe methode van MOGA kan daarom worden geïmplementeerd in elektrostatisch lensontwerp met een of meer optimalisatiefuncties en meerdere vrije variabelen, als een zeer efficiënte, volledig geautomatiseerde optimalisatietechniek. (Opmerking: Aangezien bij optimalisatie- of ontwerpproblemen van het elektrostatische lensstelsel meer algemene casestudies betrekking hebben op niet-conflicterende optimalisatiefuncties, lag onze belangrijkste focus voor de rest van het werk op het gebruik van SOGA. MOGA wordt echter getest en gepresenteerd in hoofdstuk 6 voor andere relevante situaties.)

Een andere belangrijke studie was gericht op het afstemmen van de GA-parameters, hetgeen wordt besproken in hoofdstuk 7. Het implementeren van een GA-

optimalisatie voor het ontwerp van elektrostatische elektronenlenzen maakte het mogelijk onderzoek te doen naar de invloed van GA-afstemparameters. De studie werd uitgevoerd met behulp van een voorbeeld van een multi-elektrostatisch lensstelsel met zes elektroden, dat 23 vrije variabelen had. De uitbreiding naar complexere ontwerpen is eenvoudig. Eerst werd onderzocht wat de invloed is van de waarden voor populatieomvang en het aantal generaties. Zoals verwacht heeft de optimalisatie betere resultaten opgeleverd door beide waarden te verhogen. Er werd echter aangetoond dat een populatie van 50 met 100 generaties redelijk goede resultaten kan opleveren in onze casestudies. Hogere waarden van populatie en generatie verbeterden de resultaten niet significant, terwijl de doorlooptijd sterk toenam. Bovendien werd ook ontdekt dat met een vooraf bepaalde rekeninspanning (tijd of totaal aantal systeemevaluaties), een kleinere populatieomvang met een grotere generatieomvang meestal betere resultaten oplevert dan in het omgekeerde geval. Er werd ook een analyse uitgevoerd op de Crossover- en Mutation-types als de belangrijkste afstemparameters van GA om de meest geschikte opties te vinden. Er werd aangetoond dat, ongeacht het type Crossover, de Mutation Gaussian het beste resultaat oplevert. Er werd ook vastgesteld dat van de verschillende Crossover-types Crossover Heuristic de beste prestaties levert. Bovendien leverde de combinatie van Crossover Heuristic met Mutation Gaussian significant betere resultaten op dan alle andere combinaties van Mutations en Crossovers. Daarnaast werd een analyse uitgevoerd op de Crossover Fraction, waaruit bleek dat een Crossover Fraction van 0,5 of 0,6 de beste resultaten geeft. Hoofdstuk 7 beschrijft richtlijnen voor het afstemmen van de GA-parameters, die niet alleen nuttig kunnen zijn voor de optimalisatie van elektrostatische lensontwerpen, maar ook voor andere GA-optimalisaties van functies met vergelijkbare complexiteit.

In hoofdstuk 8 worden verschillende metaheuristische optimalisatie-algoritmen geïmplementeerd en vergeleken, zoals Genetic Algorithm (GA), Particle Swarm Optimization (PSO) en Simulated Annealing (SA), evenals een op calculus gebaseerde lokale minimalisatie. De analyse werd uitgevoerd op een typisch lensstelsel met zes elektroden en 23 vrije variabelen. Er werd aangetoond dat de op populatie gebaseerde metaheuristische optimalisaties, namelijk PSO en GA, significant betere resultaten genereren dan de single-solution-based metaheuristische optimalisatie (namelijk SA) en de op calculus gebaseerde lokale minimalisatie (namelijk Fmin), terwijl PSO enigszins betere resultaten gaf dan GA. Bovendien werd uit de analyse uitgevoerd op SA en Fmin vastgesteld dat deze twee optimalisaties werken als een soort lokale optimalisatie. Het is daarom het beste om deze niet te implementeren in

onze casestudie, waar de beperkingen en optimalisatieuncties complex zijn, met meerdere lokale minima. De vergelijking tussen deze twee lokale optimalisaties toonde aan dat de metaheuristische (SA) beter presteert dan de op calculus gebaseerde (Fmin).

In hoofdstuk 9 wordt een nieuwe methode voor snelle veldberekening gepresenteerd. Zoals eerder vermeld, speelt de berekening van het elektrische veld een grote rol in alle andere gebieden van berekeningen en simulaties in elektronenoptica. Nauwkeurige veldberekeningsmethoden, zoals de Finite Element Method (FEM), Boundary Element Method (BEM) en Finite Difference Method (FDM), worden al jaren gebruikt. Dergelijke methoden zijn echter rekenkundig erg duur en maken de computersimulatie lastig of zelfs onmogelijk, wanneer wordt getracht geautomatiseerde ontwerpen te maken van elektrostatische lenssystemen met veel vrije parameters. Daarom zijn wetenschappers op het gebied van elektronenoptica al jaren op zoek naar een snelle en nauwkeurige methode voor veldberekening.

De Second Order Electrode Method (SOEM), die in 1980 door Adriaanse werd geïntroduceerd, is gebaseerd op het afkappen van de Laplace-vergelijking in cilindrische coördinaten rond de optische as bij de term met de tweede orde afgeleide van de axiale potentiaal. Deze methode bereikt een acceptabele nauwkeurigheid en zeer snelle veldberekening (0,1 seconde voor elk systeem). Er wordt echter enige afwijking in optische parameters waargenomen in vergelijking met het geval waarin het elektrische veld wordt berekend door middel van nauwkeurige modellen zoals FEM. Een nieuwe methode voor snelle veldberekening is nu voor het eerst ontwikkeld en wordt gepresenteerd in hoofdstuk 9. Deze methode biedt een veel grotere nauwkeurigheid bij veldberekening dan de methode SOEM, maar heeft nog steeds een zeer korte rekentijd (0,5 seconde voor elk systeem), vergelijkbaar met SOEM. De formulering van de veldberekening wordt uitgevoerd door de Laplace-vergelijking nabij de optische as op te lossen door ervoor te zorgen dat de termen in de potentiaalfunctie buiten de as tot aan de vierde orde afgeleide van de axiale potentiaal meegenomen worden. De auteurs hebben deze methode daarom Fourth Order Electrode Method (FOEM) genoemd. Optische parameters verkregen door FOEM liggen dicht bij de waarden berekend door FEM en, gezien de vermindering van een factor ~ 100 in rekentijd, biedt het ook een goede basis voor een nog snellere optimalisatietechniek dan de eerder gepresenteerde techniek (namelijk SOEM-COMSOL-GA) voor het ontwerp van elektrostatische lenssystemen.

Deze nieuwe methode is geïmplementeerd voor de optimalisatie van typische elektrostatische lenssystemen, waaronder zes-elektrodelenzen, met behulp van GA als optimalisatie-algoritme. De verkregen optische parameters en de rekentijd van de optimalisatie werden vergeleken met optimalisaties uitgevoerd door zowel FEM als SOEM-FEM. Er werd aangetoond dat met behulp van FOEM de optimalisatie aanzienlijk sneller werd uitgevoerd (70 keer) en dat de optische parameters berekend door FOEM een redelijk hoge nauwkeurigheid bereiken in vergelijking met waarden verkregen door FEM. Deze nieuwe methode kan interessant zijn voor wetenschappers op het gebied van elektronenoptica wanneer een snelle veldberekening voor elektrostatische lenssystemen nodig is. Deze methode stelt ontwerpers van elektronenoptica ook in staat om multi-elektrode lenssystemen, met veel vrije parameters van het lensstelsel en multi-functie optimalisaties, te ontwerpen of te optimaliseren in een volledig geautomatiseerd optimalisatieproces.

Door alle bovengenoemde theorieën en formuleringen in MATLAB te implementeren, werd een software programma ontwikkeld dat de mogelijkheid heeft om optische parameters te berekenen met behulp van veldberekening op basis van FEM, en optische parameters te berekenen met behulp van snelle veldberekeningsmethoden zoals SOEM en FOEM. De software is in staat om de optische parameters van een systeem met één lens te berekenen, evenals volledige globale optimalisatie door GA en PSO met behulp van SOEM, FOEM en FEM, en lokale optimalisatie zoals SA en Fmin. Bovendien zijn op superpositie gebaseerde veldberekeningsmethoden met behulp van nauwkeurige FEM-veldeberekening in de software opgenomen, zodat een lensstelsel met vaste geometrie kan worden geoptimaliseerd over een bepaald spanningsbereik met zeer hoge nauwkeurigheid en hoge snelheid. De stroomschema's en gedetailleerde uitleg voor deze gebruiksvriendelijke software worden gepresenteerd in hoofdstuk 10. Aangezien ons programma het ontwerp of de optimalisatie van een elektrostatische lensstelsel snel kan uitvoeren, heet het 'Fast Electrostatic Lens System Design (FELSD)'-programma. Het FELSD-programma kan voor verschillende scenario's worden geïmplementeerd. Een eerste scenario is wanneer er een lensstelsel bekend is en de gebruiker de optische parameters van dit systeem wil berekenen. Een tweede scenario is wanneer het systeem bekend is en de gebruiker de kwaliteit van het lensstelsel wil verbeteren. Een derde scenario is wanneer er geen bekend lensstelsel aanwezig is, maar er enkele criteria en/of beperkingen zijn en de gebruiker op zoek is naar een lensstelsel dat de beste kwaliteit heeft binnen de gedefinieerde grenzen. Het programma bestaat dan ook uit twee hoofdonderdelen: "Single Lens System Calculation" en "Lens System

Optimization". De codes zijn geschreven in MATLAB 2018b. De FELSD-app kan worden uitgevoerd in een MATLAB-omgeving.

Samenvattend kan worden geconcludeerd dat het hoofddoel van deze studie in voldoende mate is bereikt en dat er software is ontwikkeld die in staat is een elektrostatisch lensstelsel met rotatiesymmetrische rechthoekige elektroden te optimaliseren, die kan worden gebruikt door ontwerpers van elektronenoptische en ionenoptische lenzen. Toekomstig onderzoek moet worden gedaan om een meer algemene software te creëren die externe gebruikers in staat stelt om elke gewenste geometrie van elektrostatische lenssystemen te implementeren, niet alleen rechthoekige. Daarnaast is er onderzoek nodig dat alle verschillende combinaties van optimalisatie technieken in de software kan implementeren, waarbij de gebruiker zelf kan kiezen welke hij implementeert. Een andere toekomstige uitbreiding van dit werk kan zich richten op implementatie van magnetische lenssystemen naast elektrostatische lenssystemen.



Appendix A: The Theory of Cubic Spline

Assume that there are n points from x_1 to x_N , and it is aimed to fit cubic splines on those points. To fit a cubic spline on the data set x_1 to x_N , $S(x)$ is written as:

$$S(x) = \begin{cases} S_1(x), & x_1 \leq x < x_2 \\ S_2(x), & x_2 \leq x < x_3 \\ \vdots & \\ S_{N-1}(x), & x_{N-1} \leq x < x_N \end{cases} \quad (\text{A.1})$$

Where, S_i is a third order polynomial, which can be generally written as:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad , i = 1: N - 1 \quad (\text{A.2})$$

In which, a_i , b_i , c_i and d_i are constant parameters. In a close-format, it can be expressed as:

$$S(x) = S_i(x), \quad x \in [x_i, x_{i+1}] \quad , i = 1: N - 1 \quad (\text{A.3})$$

To satisfy the continuity conditions, the derivatives of the function at the borders of each spline should be equal. Since the polynomials are of the third order, this condition can be written up to the second derivatives, as:

$$S_i^{(r)}(x_{i+1}) = S_{i+1}^{(r)}(x_{i+1}) \quad , r = 0, 1, 2 \quad (\text{A.4})$$

Normally, splines are used to fit a function on N given points, where the values at each point are known. However, there are also problems where the values of the function at the points are not known, but instead some relations between the values of the function and its derivatives at the points are given. In such situations, writing the $S_i(x)$ in the format of Eq. A.2, makes solving the problem complicated. Instead, we rewrite $S_i(x)$ differently, to make solving easier. This is explained in the following.

Since the polynomial is of the third order, the second derivative of the polynomial is a linear polynomial. Therefore, $S_i(x)$ can be written as:

$$S_i^{(2)}(x) = \mu_{i+1} \frac{(x-x_i)}{\Delta_i} + \mu_i \frac{(x_{i+1}-x)}{\Delta_i} \quad (\text{A.5})$$

In which, $\mu_i = S_i^{(2)}(x_i)$, and $\Delta_i = x_{i+1} - x_i$

After integrating Eq. A.5 twice, we obtain:

$$S_i(x) = \mu_{i+1} \frac{(x - x_i)^3}{6\Delta_i} + \mu_i \frac{(x_{i+1} - x)^3}{6\Delta_i} + A_i(x - x_i) + B_i(x_{i+1} - x) \quad (\text{A.6})$$

Where A_i, B_i , for $i = 1, 2, \dots, N - 1$ are coefficients which must be determined in terms of u_i , and μ_i , in which:

$$u_i = S_i(x_i), \text{ and } \mu_i = S_i^{(2)}(x_i) \quad (\text{A.7})$$

The coefficients A_i and B_i can be derived by replacing $x = x_i$ and $x = x_{i+1}$ in Eq. A.6, respectively, as:

$$A_i = \frac{u_{i+1}}{\Delta_i} - \mu_{i+1} \frac{\Delta_i}{6}, B_i = \frac{u_i}{\Delta_i} - \mu_i \frac{\Delta_i}{6} \quad (\text{A.8})$$

By imposing the condition of continuity of the derivatives on Eq. A.6 at the spline's borders ($x = x_i$ and $x = x_{i+1}$), and some mathematical works, the following equation can be derived:

$$u_{i+1} \left(\frac{1}{\Delta_i} + \frac{1}{\Delta_{i+1}} \right) = \frac{u_{i+2}}{\Delta_{i+1}} + \frac{u_i}{\Delta_i} - \frac{1}{6} (\mu_i \Delta_i + 2\mu_{i+1} (\Delta_i + \Delta_{i+1}) + \mu_{i+2} \Delta_{i+1}) \quad (\text{A.9})$$

Rewriting the equation in terms of the cubic splines $S_i(x)$ again, results in:

$$\begin{aligned} S_{i+1}(x) \left(\frac{1}{\Delta_i} + \frac{1}{\Delta_{i+1}} \right) &= \frac{S_{i+2}(x)}{\Delta_{i+1}} + \frac{S_i(x)}{\Delta_i} \\ &\quad - \frac{1}{6} (S_i^{(2)}(x) \Delta_i + 2S_{i+1}^{(2)}(x) (\Delta_i + \Delta_{i+1}) + S_{i+2}^{(2)}(x) \Delta_{i+1}) \end{aligned} \quad (\text{A.10})$$

This equation needs two more equations to make it solvable. The two boundary conditions are used to solve the problem. These boundary conditions can be assumed to be:

$$S_1(x) = S_N(x) = 0 \quad (\text{A.11})$$

This equation together with eq. A.10, can be solved to obtain $S_i(x)$ and $S_i^{(2)}(x)$. Having $S_i(x)$ and $S_i^{(2)}(x)$, the coefficients A_i and B_i can be calculated, from which equation A.6 can be fully determined and the cubic spline fit $S(x)$ can be obtained.

Appendix B: Insights into COMSOL Meshing

For the lens systems utilized in this thesis, given their similar geometric scales, a consistent meshing strategy for COMSOL, has been employed. The mesh was generated by applying an edge mesh of $10\ \mu\text{m}$ along the axis-symmetric line and along the boundaries of all electrodes. The remaining space was meshed using a free triangular mesh with a predefined "Normal" setting in COMSOL, featuring a maximum element growth rate of 1.3. Example of code used to create meshes are provided below:

```
model.mesh('mesh1').create('edg1', 'Edge');
model.mesh('mesh1').feature('edg1').selection.set([1:3*(n+1)]);
model.mesh('mesh1').feature('size').set('custom', 'on');
model.mesh('mesh1').feature('size').set('hmax', '0.01');
model.mesh('mesh1').run('edg1');
model.mesh('mesh1').create('ftri1', 'FreeTri');
model.mesh('mesh1').feature('ftri1').create('size1', 'Size');
model.mesh('mesh1').run('ftri1');
model.mesh('mesh1').run;
```

The choice of a $10\ \mu\text{m}$ value for the edge mesh stemmed from testing various edge mesh values to achieve a balance between reasonably accurate optical parameter calculations and manageable computational time. Ultimately, we found that the above-mentioned meshing values offered an optimal balance for our work, and we maintained consistency with these values throughout our study.

To provide insight into the meshing data and its influence on the results and computational time, an example among our tests, the analysis for two typical lens systems (referred to as System P and System Q), conducted with three different values of the edge mesh ($10\ \mu\text{m}$, $100\ \mu\text{m}$, and $200\ \mu\text{m}$), has been presented in Table B.1 (for System P) and Table B.2 (for System Q). The lens geometries related to Systems P and Q are illustrated in Figures B.1 and B.2, respectively.

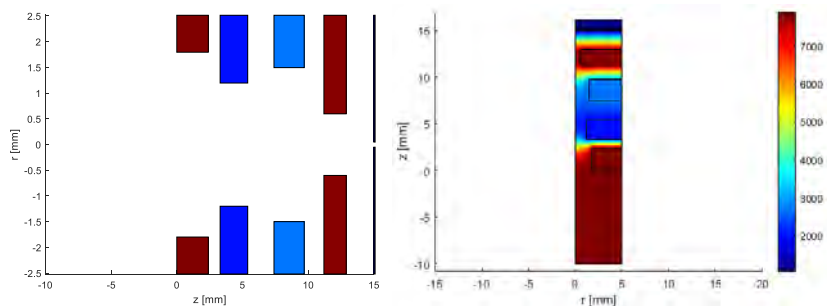


Figure B.1. a. Schematic of the lens system geometry of System P (sketched in MATLAB), b. Visualization with the voltages at each electrode.

The table in below (Table B.1) schematically illustrates the variation in mesh element sizes for System P, plotted in COMSOL. Different mesh element sizes ($10\ \mu\text{m}$, $100\ \mu\text{m}$ and $200\ \mu\text{m}$) are depicted to provide insight into the meshing strategy employed for System P. The right side of the table provides a zoomed plot of the left side, enhancing visualization.

Table B.1: Mesh element sizes variation ($10\ \mu\text{m}$, $100\ \mu\text{m}$ and $200\ \mu\text{m}$) schematics for System P (plotted in COMSOL).

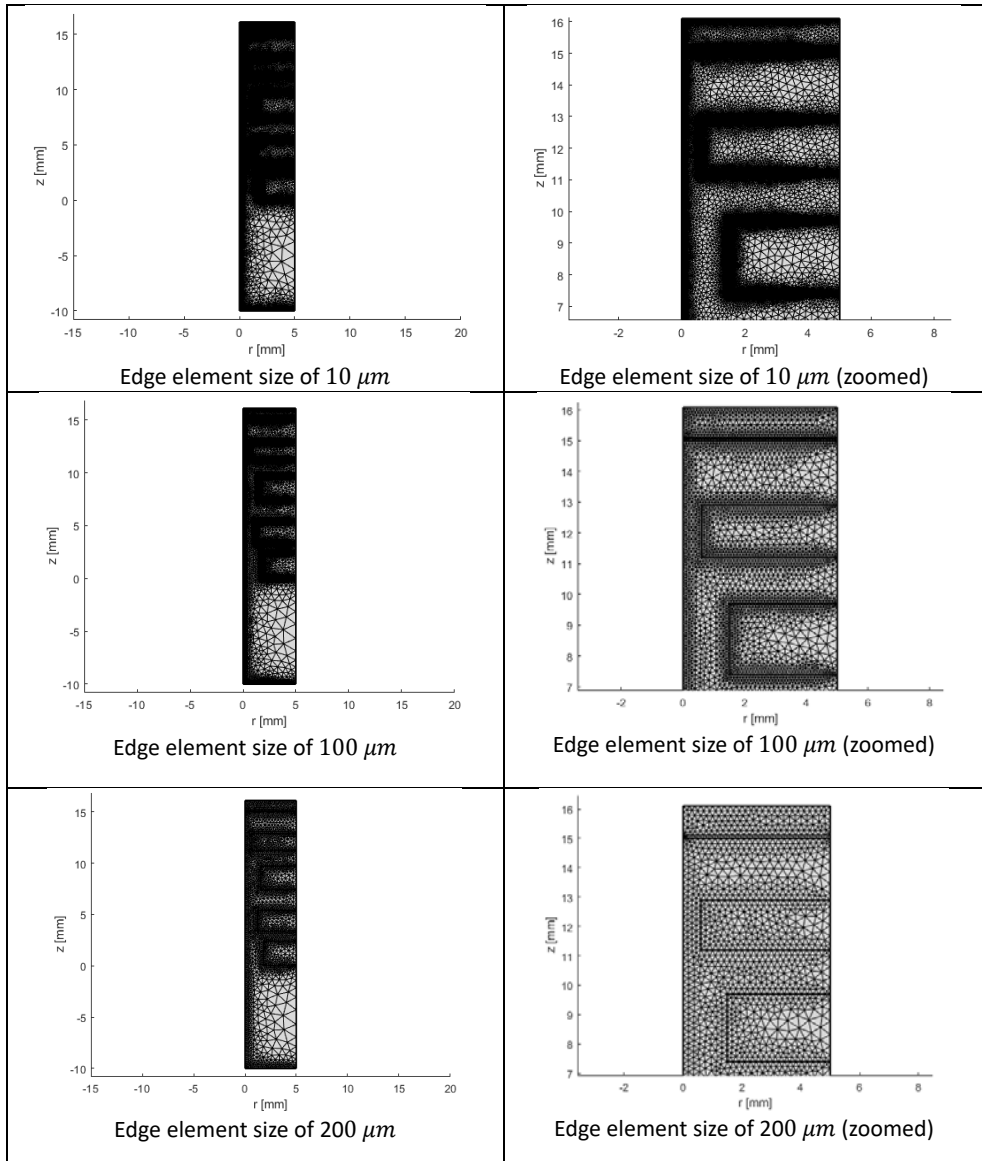


Table B.2: Comparison of voltage graphs and their first and second derivatives for System P with different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$.

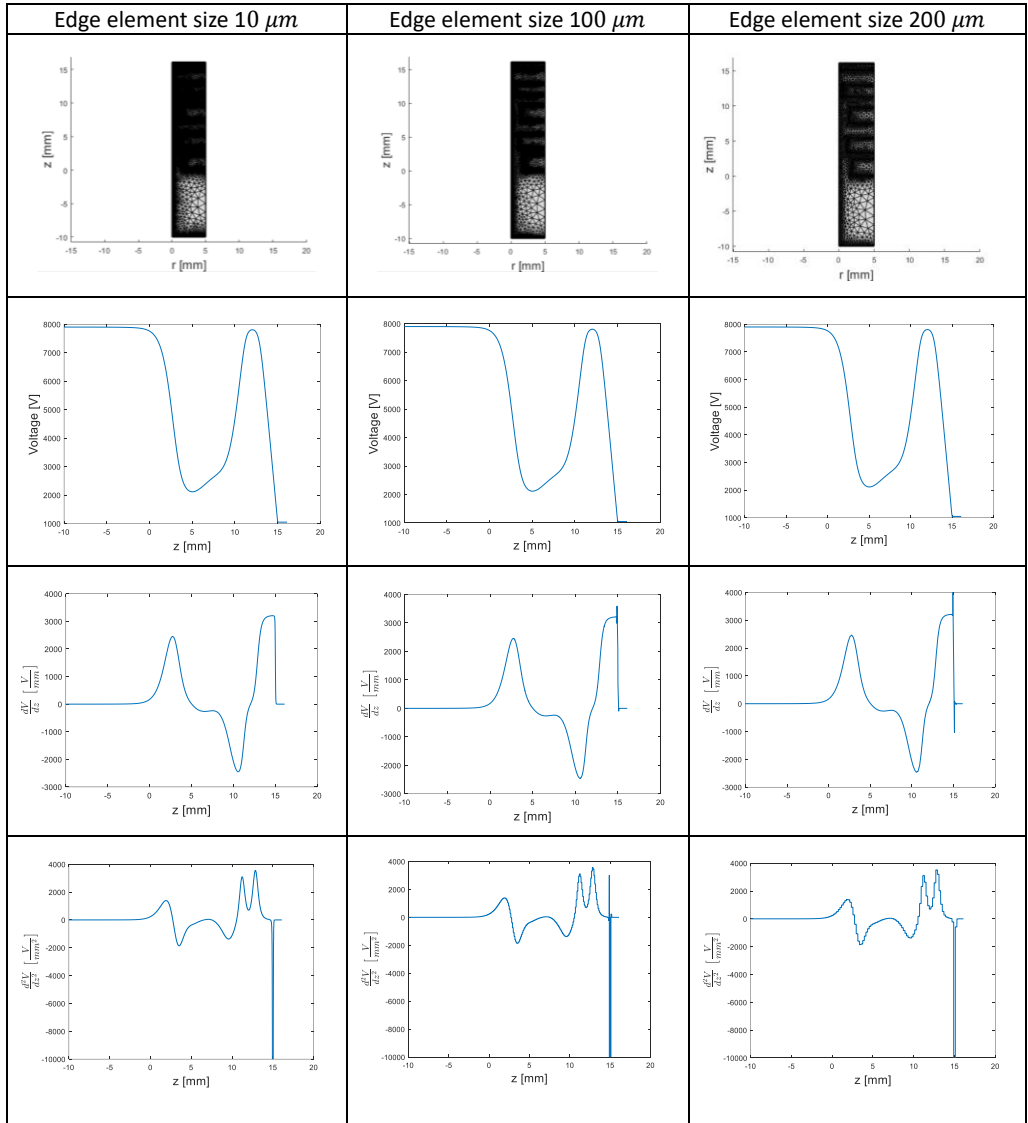


Table B.2 presents a comparison of various parameters, including voltage graphs and their first and second derivatives for System P, utilizing different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$. As observed, deviations are discernible in the first and second derivatives of the potential, particularly at the peaks of the graphs.

Below, in Table B.3, the number of elements, edge elements, and mesh vertices corresponding to each edge mesh sizes of $10\ \mu\text{m}$, $100\ \mu\text{m}$, and $200\ \mu\text{m}$, are presented. For each edge mesh size, the optical parameters of C_s , C_c , and X_c calculated using the potential, first and second derivatives, are provided to offer a quantitative insight into the influence of different mesh element sizes on these parameters. Additionally, the corresponding computational time for each element size option is provided in the table below.

Note: The optical parameters C_s , C_c , and X_c represent the spherical and chromatic aberration coefficients and image position, respectively. For further details on these parameters, please refer to Section 2.4.

Table B.3: Meshing data in COMSOL and comparison of optical parameters of C_s , C_c , and X_c and computational time for System P using different mesh element sizes of $10\ \mu\text{m}$, $100\ \mu\text{m}$, and $200\ \mu\text{m}$.

Edge mesh	Number of Elements	Edge elements	Mesh vertices	Computational time	Optical parameters
$10\ \mu\text{m}$	177605	8679	90728	63 sec	X_c (mm)= 13.38 C_c = 26.13 C_s = 217.50
$100\ \mu\text{m}$	11785	945	6124	42 sec	X_c (mm)= 13.38 C_c = 26.20 C_s = 218.25
$200\ \mu\text{m}$	4688	499	2471	38 sec	X_c (mm)= 13.39 C_c = 26.19 C_s = 218.66

As observed in Table B.3, the optical parameters exhibit slight deviations from the accurate values calculated using an edge mesh size of $10\ \mu\text{m}$. Therefore, it appears that the deviations induced by increasing the edge mesh sizes from 10 to $200\ \mu\text{m}$ for the calculated optical parameters in this scenario can be disregarded. However, it is necessary to also test other lens system geometries that may be more sensitive to meshing, such as situations where the inner radius of an electrode becomes smaller.

For this test example, the lens system used in the previous case (System P) has been modified. For instance, the inner radius of the second electrode has been reduced from $1.2\ \text{mm}$ to $0.1\ \text{mm}$. The geometry of this test lens system (System Q) is

schematically depicted in Figure B.2.a, with visualization of the voltages at each electrode (b).

Table B.4. illustrates the comparison of voltage graphs and their first and second derivatives for System Q, utilizing different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$.

Meshing data in COMSOL alongside a comparison of optical parameters (C_s , C_c , and X_c) and computational time for System Q, employing different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$, are presented in Table B.5.

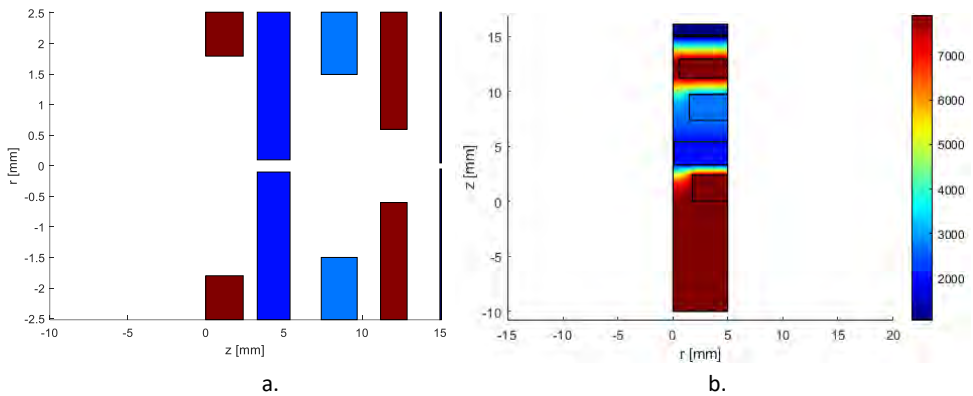


Figure B.2. a. Schematic of the lens system geometry of system Q (sketched in MATLAB), b. Visualization with the voltages at each electrode.

Table B.4: Comparison of voltage graphs and their first and second derivatives for system P with different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$.

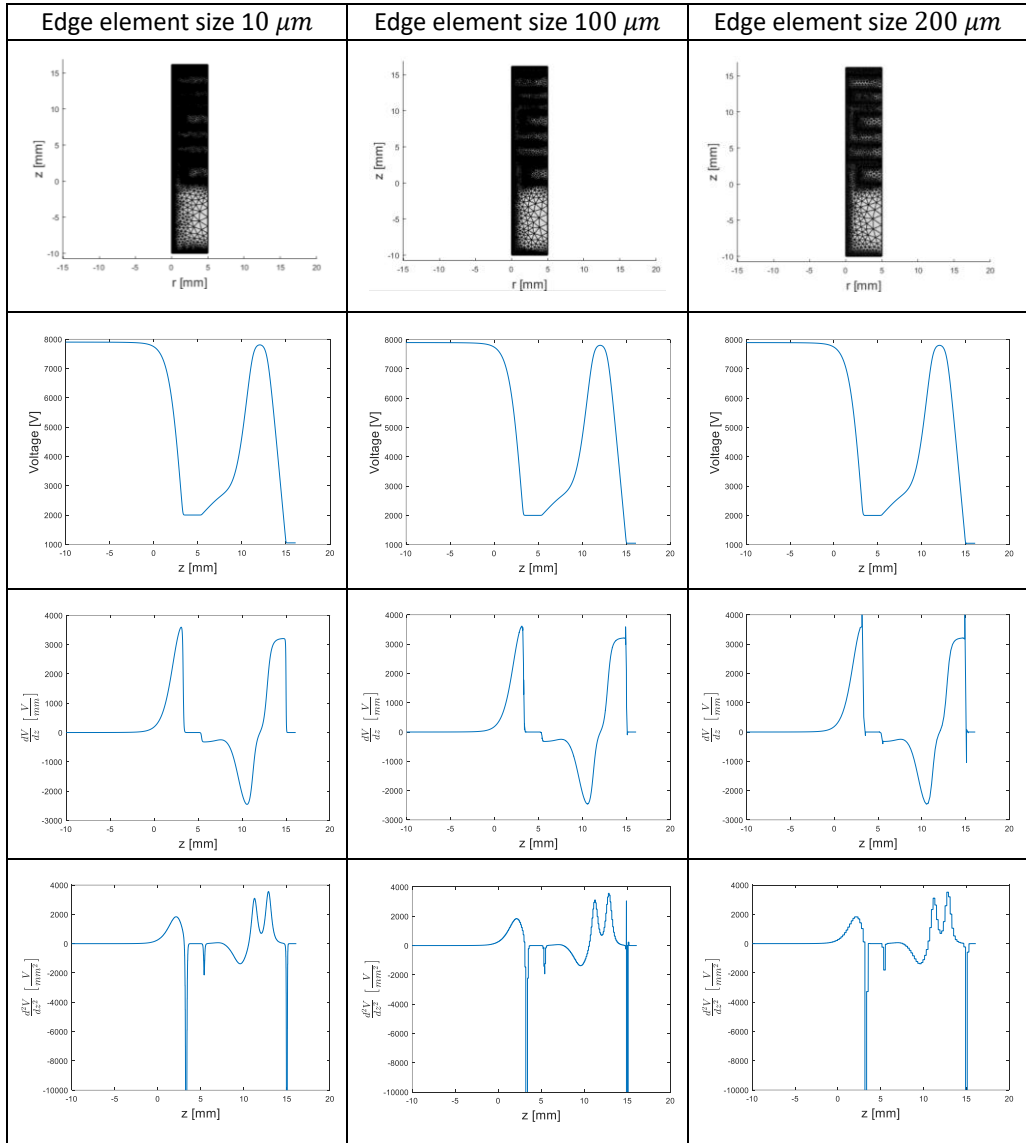


Table B.5: Meshing data in COMSOL and comparison of optical parameters of C_s , C_c , and X_c and computational time for System Q using different mesh element sizes of $10 \mu m$, $100 \mu m$, and $200 \mu m$.

Edge mesh	Number of Elements	Edge elements	Mesh vertices	Time to solve one	
$10 \mu m$	181501	8899	92676	69 sec	$X_c (mm) = 8.31$ $C_c = 9.80$ $C_s = 699.43$
$100 \mu m$	11819	967	6141	48 sec	$X_c (mm) = 8.33$ $C_c = 9.95$ $C_s = 647.65$
$200 \mu m$	4695	512	2475	39 sec	$X_c (mm) = 6.81$ $C_c = 4.18$ $C_s = 177.58$

A noticeable deviation in the calculation of the optical parameters becomes apparent with an increase in the edge mesh size values beyond $10 \mu m$. This underscores the importance of selecting a small edge element size, such as $10 \mu m$, to ensure accurate calculation of optical parameters for certain lens geometries. Therefore, this value has been designated as the edge mesh size in this study.

$$\begin{aligned}
 & \begin{pmatrix} c_{12} & c_{13} & & & & \\ c_{22} & c_{21} & & c_{23} & & \\ & c_{33} & & c_{34} & c_{35} & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & c_{(N-4)(N-4)} & c_{(N-4)(N-3)} & c_{(N-4)(N-2)} & \\ & & & a_{(N-3)(N-3)} & c_{(N-3)(N-2)} & c_{(N-3)(N-1)} \\ & & & & c_{(N-2)(N-2)} & c_{(N-2)(N-1)} \end{pmatrix}_{(N-2) \times (N-2)} \begin{pmatrix} Z_2 \\ Z_3 \\ \vdots \\ \vdots \\ Z_{N-2} \\ Z_{N-1} \end{pmatrix} \\
 + & \begin{pmatrix} d_{12} & d_{13} & & & & \\ d_{22} & d_{21} & & d_{23} & & \\ & d_{33} & & d_{34} & d_{35} & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & d_{(N-4)(N-4)} & d_{(N-4)(N-3)} & d_{(N-4)(N-2)} & \\ & & & d_{(N-3)(N-3)} & d_{(N-3)(N-2)} & d_{(N-3)(N-1)} \\ & & & & d_{(N-2)(N-2)} & d_{(N-2)(N-1)} \end{pmatrix}_{(N-2) \times (N-2)} \begin{pmatrix} \mu_2 \\ \mu_3 \\ \vdots \\ \vdots \\ \mu_{N-2} \\ \mu_{N-1} \end{pmatrix} \\
 = & \begin{pmatrix} k_{12} & k_{13} & & & & \\ k_{22} & k_{21} & & k_{23} & & \\ & k_{33} & & k_{34} & k_{35} & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & k_{(N-4)(N-4)} & k_{(N-4)(N-3)} & k_{(N-4)(N-2)} & \\ & & & k_{(N-3)(N-3)} & k_{(N-3)(N-2)} & k_{(N-3)(N-1)} \\ & & & & k_{(N-2)(N-2)} & k_{(N-2)(N-1)} \end{pmatrix}_{(N-2) \times (N-2)} \begin{pmatrix} \varphi_2 \\ \varphi_3 \\ \vdots \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \end{pmatrix} \\
 & + \begin{pmatrix} k_{11}\varphi_1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ k_{(N-2)(N)}\varphi_N \end{pmatrix}
 \end{aligned}$$

Matrix Representation of Eq. 9.42 :

$$\begin{aligned}
 \begin{pmatrix} V_2 \\ V_3 \\ \vdots \\ \vdots \\ V_{N-2} \\ V_{N-1} \end{pmatrix} &= \begin{pmatrix} \varphi_2 \\ \varphi_3 \\ \vdots \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \end{pmatrix} - \frac{1}{4} \begin{pmatrix} r_2 & & & & & \\ & r_3 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & r_{N-2} & \\ & & & & & r_{N-1} \end{pmatrix}_{(N-2) \times (N-2)}^2 \begin{pmatrix} \mu_2 \\ \mu_3 \\ \vdots \\ \vdots \\ \mu_{N-2} \\ \mu_{N-1} \end{pmatrix} \\
 &+ \frac{1}{64} \begin{pmatrix} r_2 & & & & & \\ & r_3 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & r_{N-2} & \\ & & & & & r_{N-1} \end{pmatrix}_{(N-2) \times (N-2)}^4 \begin{pmatrix} Z_2 \\ Z_3 \\ \vdots \\ \vdots \\ Z_{N-2} \\ Z_{N-1} \end{pmatrix}
 \end{aligned}$$

Acknowledgements

I would like to express my gratitude to my supervisor, em. prof.dr.ir. Pieter Kruit, for his continuous support and patience throughout this journey. Pieter, your optimism and patience have been a constant source of encouragement, lifting my spirits during challenging times. Thank you also for the time you kindly devoted whenever I needed assistance. Despite numerous occasions when I had to send you articles and work for review on short notice, you patiently provided feedback. I am particularly grateful for your assistance during the conference in US-Florida, where you dedicated your time to help me review my slides before my presentation, even by not attending another talk. Your availability, guidance, and patience have been invaluable to me. Additionally, I would like to express my gratitude for the financial support you've provided for this PhD project.

I would also like to extend my heartfelt thanks to my other supervisor, dr. Cornelis W. (Kees) Hagen, for his kind support, time and invaluable advice provided during our scheduled meetings every week throughout the whole early years of my PhD journey. Kees, I am especially thankful for the time you always generously devoted to reviewing my work, often on short notice and under time constraints, sacrificing your personal time to provide valuable feedback and editing. Your professional editing of my articles has greatly enhanced the quality of my work. Thank you for your guidance, mentorship, and dedication. I also remember the many non-scientific discussions we had during our meetings, covering a wide range of topics from cultural exchanges to social issues and even politics, which was always very nice for me to engage in.

I would also like to express my sincere appreciation to dr. ir. Mohamad Ghaffarian Niasar, my daily supervisor. Throughout my PhD journey, you consistently maintained an optimistic outlook on our research outcomes, encouraging me to persevere through challenging scientific problems even when I felt disappointed that our approach might not lead to success. Thank you also for generously dedicating your time whenever I had questions or needed discussions, particularly regarding software development and mathematical aspects. Your kind support and belief in our work were instrumental in overcoming obstacles and in bringing this project to success.

To dr. ir. Jacob Hoogenboom, the head of our Imaging Physics (ImPhys) department, and the MINT group, I would like to thank you, Jacob, for your great leadership as the head of this group and department, and for your support during my PhD time.

Additionally, I am thankful for graciously accepting to be a reserve committee member at my defence despite your busy schedule.

To my respected committee members, prof. dr. Pavol Bauer, prof.ir. Peter Vaessen, prof. dr. Tom Wirtz, and prof. dr. Anjam Khursheed, I want to extend my sincere gratitude for accepting to be part of my committee and for dedicating your time and expertise to review my PhD thesis, especially considering your busy schedules. Thank you for providing insightful feedback and constructive comments that have contributed to refining my research.

To my colleagues at the Imaging Physics (ImPhys) department, formerly known as the Charged Particle Optics (CPO) group and now renamed as Microscopy Instrumentation & Techniques (MINT), including Aya, Xiaoli, Maurice, Ali, Kerim, Mike, Ryan, Luc, Martin, Diederik, Mathijs, Dustin, and many others (whom I apologize if I missed mentioning here), whose warmth and kindness made my time there truly enjoyable, I would like to extend my appreciation. During the final years of my PhD, the constraints imposed by the COVID-19 pandemic, along with subsequent remote work situations, prevented me from being on-site for an extended period in the wonderful environment at ImPhys. Despite this, my colleagues remained approachable and readily available to engage in discussions whenever I had questions or needed assistance.

To Anjella, our great former secretary at ImPhys-CPO-MINT, I would like to express my heartfelt gratitude. Throughout my time at CPO-MINT, you were always there when needed, often going above and beyond by extending her assistance via WhatsApp, even during your personal time at home. Anjella, thank you for always being there for all students at ImPhys.

To our recent secretary, Hannah, thank you for your time, kind assistance and support whenever I needed help with sorting out official tasks.

To our technicians, Han and Johan, I express my gratitude for your time, kindness, positivity, and support whenever I required assistance with technical matters during my PhD tenure.

To my wonderful sisters, Yalda and Salma, your love and encouragement have been a source of inspiration throughout my doctoral journey. Your consistent belief in me have given me the strength to persevere through every challenge. Thank you for being my confidantes, cheerleaders, and friends.

To my best, dearest, parents in the world, my Mum and Dad, Thank you for your unconditional love and support throughout my entire life including my PhD journey. Your boundless sacrifices, endless patience, and love have been my greatest source of strength. Thank you for all of this and for being the best parents a child could ever ask for.

To my beloved husband, Mohammad, I want to express my deepest gratitude for your constant support, love, and encouragement throughout my life, especially during my doctoral journey. Your presence has been a pillar of strength, guiding me through challenges. Your unwavering support, has been invaluable to me. Whether it was lending a listening ear when I faced challenges at work or providing reassurance during moments of doubt, you have been my rock. Thank you for being my partner in every sense of the word. I am grateful for your love and patience, which have made this journey possible.

To my adorable angels, my princesses, my sweet daughters, Dorsa and Elysa, Your presence in my life has been a constant source of joy, and inspiration. Your laughter, hugs, and words of encouragement have brought light to even the most difficult days of my doctoral journey. Thank you for being the brightest stars in my sky. You are everything to me, providing the motivation I needed never to give up through every up and down during my PhD journey. Thank you for your patience for the days and nights when I could not be with you due to my studies. I am endlessly grateful for your presence in my life; you've reminded me of what truly matters in life!

At the end, I would like to express my gratitude to all those who contributed to the realization of this study and the completion of this thesis. This achievement is not solely mine, but ours collectively.

About the author

Neda Hesam Mahmoudi Nezhad was born in 1978 in Shiraz, Iran. She gained admission to the National Organization for the Development of Exceptional Talents (NODET, known as SAMPAD in Persian) by successfully passing a competitive national exam, first for middle school and later for high school. Completing her high school education in mathematics and physics, she earned her diploma in 1996.

Following this, Neda pursued her bachelor's degree in Physics at the University of Shiraz in Iran, graduating in 2000. Subsequently, she was admitted to Shahid Beheshti University (Laser and Plasma Research Institute) in Tehran, Iran, where she specialized in Photonics, ultimately earning her master's degree in 2005. Her master's thesis was titled "Analytical Calculation of Plasma's Polarizability Irradiated by Ultra-Short Laser Pulses."

In 2008, Neda moved to the Netherlands as a knowledge migrant, continuing her academic journey through acceptance into the Erasmus-Mundus Program in Physics. She successfully completed her second master's degree in 2013 in Optics within the Imphys/Optics group at the Technical University of Delft in the Netherlands, under the supervision of Prof. J. Thijssen and Dr. F. Bociort. Her master's thesis focused on "Optical System Optimization using Genetic Algorithms".

Following this accomplishment, she joined the Charged Optics Group in 2016 as a guest researcher, contributing to the project focused on "Electron Lens System Optimization Using Genetic Algorithms," under the guidance of Prof. Pieter Kruit. In 2017, she commenced her Ph.D. journey at the same group, which underwent a name change from CPO to MINT (Microscopy Instrumentation & Techniques), at TU Delft, the Netherlands. Her doctoral research centred around the project "Electrostatic Electron Lens System Design and Optimization Using Evolutionary Algorithms," under the supervision of Dr. C.W. Hagen and Prof. P. Kruit as the promoters, and with Dr. M. Ghaffarian Niasar as the co-promoter. Neda's contributions to this project resulted in the development of a fully automated software for the design and optimization of Electrostatic Electron lens systems, leading to several publications documented in this thesis.

List of Publications

1. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "A New Approach for Fast Field Calculation in Electrostatic Electron Lens Design and Optimization", *Journal of Nature (Scientific Report)*, Article no. 4859, pg. 1-21, 2024.
2. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "Comparison of Different Optimization Techniques in Electron Lens Design", *IEEE proceedings of 9th International Conference on Optimization and Applications (ICOA)*, pg. 1-5, 2024.
3. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar , Cornelis W. Hagen and Pieter Kruit, "Tuning Parameters in the Genetic Algorithm Optimization of Electrostatic Electron Lenses", *IEEE proceedings of international conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pg. 170-173, 2023.
4. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, A. Mohammadi Gheidari, Pieter Kruit, Cornelis Wouter Hagen, "Multiple criteria optimization of electrostatic electron lenses using multi-objective genetic algorithms", *Journal of Vacuum Science & Technology B39*, pg. 062605, 2021.
5. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, "Genetic Algorithm Performance and the Influence of its Control Parameters on the Optimization of Optical Lens Design", *2021 IEEE Proceedings of Congress on Evolutionary Computation (CEC)*, pg. 65-70 (Kraków, Poland, 2021).
6. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, A. Mohammadi Gheidari, Pieter Kruit, Cornelis Wouter Hagen, "Probe Size and Detection Efficiency Optimization in Electrostatic Lens Systems Using Multi-Objective Genetic Algorithms", *65th International Conference on Electron, Ion, and Photon Beam Technology and Nanofabrication (EIPBN 2021)*, Underline Science Inc. DOI: [10.48448/Oqcq-zy39](https://doi.org/10.48448/Oqcq-zy39), New Orleans, Louisiana, USA, 2021.

7. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, Cornelis Wouter Hagen, Pieter Kruit, “*Local versus Global Optimization of Electron Lens System Design*”, *2020 IEEE Proceedings of 6th International Conference on Optimization and Applications (ICOA)*, pg. 9094475, 2020.
8. **Neda Hesam Mahmoudi Nezhad**, Mohamad Ghaffarian Niasar, A. Mohammadi Gheidari, Cornelis Wouter Hagen, Pieter Kruit, “*Multi-electrode Lens Optimization Using Genetic Algorithms*”, *International Journal of Modern Physics A*, Vol. 34, 1942020, 2019.
9. **N. Hesam Mahmoudi Nezhad**, M. Ghaffarian Niasar, A. Mohammadi Gheidari, T. Janssen, C. W. Hagen, and P. Kruit, “*Optimization of Electrostatic Lens Systems Using Genetic Algorithms*”, *Recent Trends in Charged Particle Optics and Surface Physics Instrumentation (Proceedings of 16th International Seminar)*, edited by F. Mika et al., (MJ servis spol. s.r.o., Brno, Czech Republic), pg. 24-26, 2018.



TU Delft

Delft
University of
Technology