SADDLE-POINT CONSTRUCTION METHOD FOR OPTICAL DESIGN

by

Yu Yan

A Thesis Submitted to the Faculty of the
COLLEGE OF OPTICAL SCIENCES
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
In the Graduate College
THE UNIVERSITY OF ARIZONA
2017
STATEMENT BY AUTHOR

The report titled Development of Saddle-Point Construction Method for Optical Design prepared by Yu Yan has been submitted in partial fulfillment of requirements for a master’s degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Yu Yan

APPROVAL BY THESIS DIRECTOR
This thesis has been approved on the date shown below:

______________________________________________
Jim Schwiegerling
Professor of Optical Sciences

Defense Date
6/21/2017
Acknowledgements

I would like to take the time to first thank my advisor Dr. Jim Schwiegerling for his excellent guidance and patience, helping me understand the theory and his instruction in developing the implementation.

My appreciation also goes to Rubing Xu, Xin Chen, Yiyao Liu, who as my good friends, were always willing to support me and giving their best suggestions. And I also want to thank Brian Redman, Chun Xia, Dulce Gonzalez Utrera, Jianbo Zhao, Xuan Wang, Xiangyu Guo, Ziyi Wu for making my time in Tucson so memorable.

Finally and most importantly, I would like to thank my parents, Yongyi Yan (颜永益) and Liangmei Cheng (程良梅) who I love dearly, for always supporting me under any experience I was facing in life.
Dedication

I would like to dedicate my dissertation to the best friend of my life, Yihui Zhou.
## Contents

Acknowledgements ................................................................................................. i

List of Figures........................................................................................................ iv

List of Tables ........................................................................................................... v

Abstract..................................................................................................................... 1

Introduction.............................................................................................................. 2

1.1 Definition ......................................................................................................... 4

1.1.1 Merit function ............................................................................................. 4

1.1.2 Variables ..................................................................................................... 5

1.2 Damped Least Squares Method ....................................................................... 6

1.3 Global Optimization ......................................................................................... 9

Implementation ....................................................................................................... 13

3.1 Light Apply SPC to lens design ................................................................. 13

3.2 Implementation ............................................................................................. 16

Result....................................................................................................................... 20

4.1 Example 1: Lens Growth ............................................................................. 20

4.2 Example 2: Optimization ............................................................................. 24

Summary............................................................................................................... 32

Appendix A – Zemex Macro - Calculate the partial derivative of Merit function ........................................................................................................ 34

Appendix B – Zemex Macro - Searching local minimum from saddle point 37

Appendix C - Python Script (Plot the result, Data Analysis and main SPC body) ........................................................................................................ 39
List of Figures

Figure 1. 1 The structure of Global Optimization ........................................... 9
Figure 2. 1 The properties of Saddle Point ................................................. 11
Figure 2. 2 Cayley's cubic Surface ......................................................... 12
Figure 3. 1 Inserting a null element in a local minimum ......................... 13
Figure 3. 2 Partial derivative of Merit Function profile ......................... 17
Figure 3. 3 The flow chart of SPC process ........................................... 19
Figure 4. 1 Starting Configuration of Lens growth process .................. 21
Figure 4. 2 The Result of local minimum searching from c1= 0.000031 .... 22
Figure 4. 3 The Result of local minimum searching from c2= 0.026099 .... 23
Figure 4. 4 The Result of local minimum connecting both saddle points ... 23
Figure 4. 5 The layout and the performance of the patent .............. 25
Figure 4. 6 Results of afocal part of the wide-angle lens ..................... 26
Figure 4. 7 Results of second part of the wide-angle lens ................. 27
Figure 4. 8 The results of optimization I ............................................ 28
Figure 4. 9 The results of optimization II ............................................ 29
Figure 4. 10 The results of optimization III ......................................... 30
Figure 4. 11 The result of optimization IV .......................................... 31
## List of Tables

3.1 Merit Function Operands used in the algorithm ........................................ 16
4.1 The properties of the original patent ....................................................... 24
Abstract

In this report, Saddle Point Construction method is studied. Compared with several common optimization methods, Saddle Point Construction method shows some advantages in lens design optimization and lens growth. One lens growth case and a wide-angle lens optimization case applying Saddle Point construction method are also given in this report.
Chapter 1

Introduction

In most cases, our goal for optical design is to find a best solution with particular specifications of an optical system, which typically is to find the minimum of the merit function. With the development of computerized ray tracing, this optimization process became easier than ever before. Many optimization algorithms have been broadly used in commercial lens design software (Zemax, CodeV, OSLO etc.). The dominant algorithms used in these software packages are the Damped Least Squares method (DLS) or other modifications of the Damped Least Squares method and Global Optimization (GO). The basic concept of these two methods will be described later in this section. Another technique commercial software uses is the Hammer Algorithm. The Hammer Algorithm is a technique that give a small change in some parameters to shift the starting point and rerun the algorithms, DLS or GO, to escape the local trap. There are also other less popular but calculation consuming methods, such as Genetic algorithm [1] that encode a mass of potential solutions into string structures as chromosomes in biology to constitute a generation and by taking reproduction and crossover steps to find the solution and
simulated annealing algorithm that simulate the process of controlling the
temperature of excited atoms and molecules to a state which has a nonzero
probability of reaching a higher energy state [2].

The Saddle-Point Construction Method (SPC) [3] was first developed by Optics
Research group at Delft University of Technology. Different from DLS and GO,
Saddle-Point Construction Method effectively overcomes some drawbacks of
DLS and GO, such as becoming trapped in a local minimum and giving subpar
results. Instead of only one best solution, SPC usually gives several solutions
(local minimum), provide the designer with more choices. SPC is not only for
optimization, but can also be used in lens growth in a way similar to the Genetic
algorithm.

The mathematical theory of SPC is given in the section II. In section III, The
Implementation of SPC using Zemax macro and Python is described. In section
IV, Lens growth and optimization of a wide-angle lens by using SPC is covered.
1.1 Definition

1.1.1 Merit function

Before stepping into the explanation of algorithms, the definition of merit functions and its variables need to be stated carefully. The goal for the optical design is to find the systems achieving the best performance within some boundaries. The limitations may include price, material selections, tolerance, the total length of the system, etc. The “performance” can be mathematically expressed by small aberration magnitudes. Since the aberrations are always related in a nonlinear manner to the variables, it’s complex to find the direction leading to the minimum. To implement the program easily, the merit function is usually defined by taking the RMS of all aberrations.

\[ \phi = \sqrt{\sum \omega_j^2 (\alpha_j - \alpha_j t)^2} \]  

(1.1)

where \( \omega_j \) is the weight coefficient of the correspond aberration, \( \alpha_j - \alpha_j t \) represent the distance of the current aberration value to the target aberration value. It should be noted that the form of merit function can be slightly changed according to different applications.
1.1.2 Variables

Typically, there are several parameters usually considered as primary variables. 1) anticipated number of surface 2) spacing 3) glass choice 4) surface curvature. There could be other parameters. For example, when the optical system to be optimized contains an aspherical surface, the conic coefficient need to be taken into account. The optimization is the process to seek the location of an acceptable configuration within the given boundaries. Usually, once the merit function is set along with the requirements, derivatives of the merit function are calculated, a new set of variables that reduce merit function can be determined.

Among all the variables, the curvature of the surfaces is the most important parameters used in the optimization process. The curvature determines the first-order properties of the optical system and is usually sensitive to the aberration. When setting the curvature as a variable in optimization, the range of the curvature should be carefully defined that some conditions have to be met. The shape should be possible to manufacture, the ratio of the diameter of the surface to the radius should be much less than unity, the total internal reflection of rays should be avoided, etc.

The spacing in system refers to two parameters. One is the thickness of the less, and the other is the gap between lenses. Usually, the aberration is not sensitive to lens thickness while it’s much more sensitive to the gap because the gap between the lenses affects the relative location to the aperture stop. Similar to curvature, the spacing also has the limits. The spacing value should be buildable, and it can’t be extremely large.
The glass choice itself is always considered as the boundary, because unlike curvature or spacing, the refractive index and Abbe number of the materials are not a continuous function. The selection of the material is always limited by the price and accessibility of this material.

1.2 Damped Least Squares Method

DLS is the most frequently used algorithm among all the optimization methods and also it’s the basis of other optimization methods. Many methods are directly developed by modifying or combining DLS with other algorithms. The DLS method itself is the modification of Least Squares method by adding a damping factor discussed further below.

Usually, we use aberrations of an optical system to constitute the optimization merit function and the aberrations can be described by $f_1, f_2, \ldots, f_i$. Each $f_i$ represents one aberration. Suppose we take $m$ aberrations and $n$ parameters into account, by using taylor series, each aberration can be described as:

$$ f_i = \sum_{j=1}^{n} \alpha_{ij} x_j + f_{i0} $$  \hspace{1cm} (1.2)

Where
\begin{equation}
\alpha_{ij} = \frac{\partial f_i}{\partial x_i} \tag{1.3}
\end{equation}

\(f_{i0}\) is the initial value of the starting point. To get the better performance, the condition for the elimination of the aberrations can be expressed by:

\begin{equation}
\sum_{j=1}^{n} \alpha_{ij} x_j + f_{i0} = 0 \tag{1.4}
\end{equation}

One easy way to define the merit function and applying least-square method is shown as

\begin{equation}
\phi = \sum_{i=1}^{m} f_i^2 = \sum_{i=1}^{m} \left( \sum_{j=1}^{n} \alpha_{ij} x_j + f_{i0} = 0 \right)^2 \tag{1.5}
\end{equation}

Where \(\phi\) is the merit function value.

If we set the partial derivatives \(\frac{\partial \phi}{\partial x_j}\) equals to zero, we can obtain a \(m\) equations of \(n\) unknowns normal equations

\begin{equation}
\bar{A}A\bar{x} = -\bar{A}f_0 \tag{1.6}
\end{equation}

where \(A\) is a \(m\times n\) matrix consist of \(\alpha_{ij}\) in \(i\) columns and \(j\) lines, respectively. The matrix \(\bar{A}\) is the transpose of \(A\). \(x\) and \(f_0\) are the column vectors of \(x_i\) and \(f_{i0}\). By solving the equations above, we can get a new set of parameters and correspond its merit function value. However, the parameters \(x\) changes given by equation (1.5) usually rapidly, since the high order aberrations change slowly. Additionally, the nonlinearities limit the validity of equation (1.4) [4][5].
Therefore, Levenberg, Wynne and Girard proposed a modified "damped" version of this equation which is shown as following:

\[ \phi' = (f_{i0} + A \cdot x)^2 + p^2x^2 \]  

(1.7)

With the similar manner as mentioned in previous, if we set the partial derivatives \( \frac{\partial \phi'}{\partial x_j} \) equals to zero, we can obtain the damped normal equations

\[ (\tilde{A}A + p^2I)x = -\tilde{A}f_0 \]  

(1.8)

Where \( I \) is the identity matrix, \( p \) is the damping factor and tries to keep the values of \( x \) small.

By introducing this damping factor, the validation of the equation (4) can be maintained and by solving this equation, we can get the optimized solution to the optical design.
1.3 Global Optimization

One main problem of the DLS is that the searching process can be trapped in a local region depending on the distribution of the merit function. Thus, only a local minimum (lower values of merit function) of the design is found and there could be other possible better solutions outside of this region. To find these hidden minima in the larger area of the merit function and achieve the global minimum, a mechanism to get out of this trap is needed. One possible way is introducing the escape function. [6] The figure of the merit function and the flow chart are given as:

Figure 1. 1 The structure of Global Optimization
Chapter 2

Saddle Point Optimization Theory

After the merit function is defined, the next step for lens design is to search for critical points at which the derivative of the merit function equals to zero. These critical points could be the local minimum we’re looking for. But when more variables are taken into account, the number of dimensions will increase, and thus the critical points may become saddle points. Mathematically, we can use Hessian matrix to identify saddle points. Hessian matrix is a square matrix where each element is the second-order partial derivative of a function. The number of the negative eigenvalues of this matrix is defined as the Morse Index (MI). That’s to say, if we have N variables constructing an N-dimensional space, critical points with $MI = 0$ are the minima, the critical points with $MI = N$ are the maxima, and the rest with $MI$ values between 1 to $N - 1$ are considered as saddle points.
Take a two-dimensional space as an example, for all saddle points in a two-dimensional function, their $MI = 1$. The saddle point landscape is shown as in figure 2.1a). In the green line direction, the saddle point in the figure is the minimum, while in the red line direction, the saddle point is the maximum. One useful property for these saddle points with $MI=1$ is that in the direction where these saddle points are the maximum, two other local minima are connected to each saddle point along this direction [3] as shown in Figure 2.1b). This property is not unique in 2D space; it also applies to high order dimensional space. Based on this characteristic, constructing saddle points is very useful for minimum searching.
The other *a priori* knowledge for SPC is that most merit function landscapes in lens design follow the similar manner of Cayley's Surface, for either two separated hyperbolic surfaces there are only one single point connecting them [3]. The landscape of a Cayley's cubic surface is shown in figure 2.2. The conjunction point connecting the double conic surfaces is the saddle point we are looking for. Under this *a priori* knowledge, detecting a saddle point with $MI=1$ can be considerably simplified by searching a straight line crossing two surfaces, the point on the line with the derivative equal to zero is the saddle point.

![Cayley's cubic Surface](image)

**Figure 2.2** Cayley's cubic Surface
Chapter 3

Implementation

3.1 Light Apply SPC to lens design

As long as a proper merit function is set up, SPC can be applied to various applications, and any parameter in the merit function can be taken into account. However, for simplicity, in this report, the curvature of surfaces is taken as the main parameter. In the last step of the algorithm, the thickness of the lenses in the configuration is optimized.

![Figure 3.1 Inserting a null element in a local minimum](image)
Gives an optical system with K variables that has already reached its local minimum, the Merit function value for this system is denoted as $MF_0$. To construct a saddle point in this configuration, first, a null element is inserted in this system as shown in figure 4. A null element is typically a lens with zero thickness and two identical surface curvatures. It can also be an air "lens". With this null-element, the optical properties of the system don't change, since there is no effect on the path of the rays in the system. Thus, for any starting values of the inserted null element, the merit function value remains the same.

$$MF(\vec{c}_0, c_{k+1}, c_{k+2}) = MF_0$$

(3.1)

where $c_{k+1} = c_{k+2}$ since two curvatures of the lens equal. $c_k$ is a variable and the constant vector $\vec{c}_0$ defines a line on the equimagnitude hyper-surfaces with $MF = MF_0$. Equation (3.1) represents a linear transformation, a straight corresponding line in the variable space.

In the SPC method, the position of the inserted null element can be arbitrary, However, the curvature of the saddle points leading to the local minimum need to be computed mathematically. A detailed description of the general SPC was given in [3].

Even though the null element is optically transparent, small changes in curvature will affect the merit function values. In the real case, a small difference between $c_{k+1}$ and $c_{k+2}$ is introduced to achieve the SPC method. By calculating the derivative of merit function, we can find the curvature of saddle point.
where $\Delta c$ is a small curvature change. The saddle points are located where the partial derivative vanish, that's

$$\frac{\partial MF}{\partial c_{k+1}} = 0$$  \hspace{1cm} (3.3)$$

The saddle point found has a downward direction and a new upward direction shown in the Figure 2.1 a) by the red curve and green curve respectively. Along the downward direction given by Equation (3.1), two new local minima can be found on opposite sides of the saddle point, having:

$$c_{k+1} = c_{k+2} = c_s \pm \epsilon$$  \hspace{1cm} (3.4)$$

where $c_s$ is the saddle curvature and $\epsilon$ indicates a small curvature change. Then, these two points are optimized and two different local minima will be found.
3.2 Implementation

In this report, the SPC algorithm is implemented mainly in Zemax macro language. Due to the data analysis limitation of Zemax macro, Python is used as an auxiliary to the process. The merit function used in the SPC is the default Zemax merit function and the Operands taken in this algorithm are:

<table>
<thead>
<tr>
<th>Operand Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPHA</td>
<td>Spherical Aberration</td>
</tr>
<tr>
<td>COMA</td>
<td>Coma</td>
</tr>
<tr>
<td>ASTI</td>
<td>Astigmatism</td>
</tr>
<tr>
<td>FCUR</td>
<td>Field Curvature</td>
</tr>
<tr>
<td>LACL</td>
<td>Lateral Color</td>
</tr>
<tr>
<td>AXCL</td>
<td>Axial Color</td>
</tr>
<tr>
<td>EFFL</td>
<td>Effective Focal Length</td>
</tr>
</tbody>
</table>

Table 3.1 Merit Function Operands used in the algorithm

Thanks to the PyDDE package [8], it's possible to interface Zemax and Python. Thus, there are two options to implement the algorithm. One is that the entire algorithm is written in Python, once the variables can also be adjusted via PyDDE. However, the reality is that the algorithm itself is space consuming and the connection established by PyDDE is not strong enough. The efficiency for this option is extremely low. Therefore, the other option is taken instead. In option 2, the entire algorithm is implemented in Zemax macro. Python is just responsible for the Data Analysis part. The results of the Zemax macro are output to a text
file, and by analyzing the output of Zemax macro, Python returns to Zemax with the new starting point information and other relative coefficients. The detailed script of the SPC algorithm is given in the Appendix.

To initiate the SPC, there is a starting configuration, like a doublet or a Cooke triplet. Typically, one null element, zero thickness, is firstly inserted in the middle of two elements. Secondly, the curvatures of two surfaces are changed in a range. The boundary of the range depends on the different system that is needed to be given by the designer, Equation (3.2) is then used to calculate the partial derivative of the merit function. Usually, profiles as in Figure 3.2 with several zero points result. These zero points are the saddle points. After finding these saddle points, additional local minimum along the direction given by the equation (3.1) can be found.

![Partial derivative of Merit Function profile](image)

**Figure 3.2** Partial derivative of Merit Function profile

Once a local minimum is found, this local minimum can be used as the starting point for the next round. Iterating the process given in the paragraph above enables lens growth or optimization. The variable of the process is not limited to
curvature; the distance or other parameters can also be taken as variables. When searching the local minimum process, situations where two surfaces cross over or other unrealistic conditions may arise. Typically, the situation is ignored and the SPC process continued. After the whole process, thickening the element can be used to eliminate unrealistic situations.

Besides adding a non-thickness null element, splitting the lens by a non-thickness airspace is also usually applied in SPC. For splitting the lens, firstly, an air gap is inserted into the lens. The following steps are then similar to adding null element stated above.

For lens growth case, the designer needs to carefully set the boundary conditions for the configuration, including but not limited to the size and the length of the system. The new elements can be added in between or after the starting point setting. For optimization, typically one element need to be removed first and a null element inserted where the original element was, prior to running the SPC process. If the splitting way is taken, the split-element doesn’t need to move. The distance change can be made after the curvature minima are found. The whole process is shown as in Figure 3.3
Figure 3.3 The flow chart of SPC process
Chapter 4

Result

In this section, two SPC examples are going to be described. The first one is utilizing SPC in lens growth, and the second example is a wide-angle lens optimization process by using SPC algorithm.

4.1 Example 1: Lens Growth

In this project, lens growth is not fully explored, since the lens growth process requires more specifications of the configuration and experience of the designer. The role lens growth plays in this project is more like a warm-up and a trial for the script.

In our lens growth example, a doublet grows into a triplet. The flow chart in the Figure 3.3 describe this procedure. Some basic specifications are listed as:

- 100mm focal length
- F/10
- 10 FOV
- The length of the lens group is no longer than 10mm
- Only BK7 and F2 glass used

The starting configuration is shown in Figure 4.1. This doublet configuration is originally from the setup in [3]. Since the parameters are not given specifically, the doublet was optimized using the Zemax default optimization algorithm (Damped Least Square). During the process, the curvature of the last surface is set to variable to maintain the focal length of the whole system.

![Figure 4.1 Starting Configuration of Lens growth process](image)

Following the process stated in the previous section, the null element is inserted and Equation (3.2) applied. The partial derivative of the merit function as in Figure 5 is calculated. From this plot, the two Null Element Saddle Points (NESP) are calculated:

\[ c_{k+1} = 0.000031, c_{k+2} = -0.026099 \]
After the NESPs are found, Equation (3.4) is applied to find the local minimum.

For $c_1=0.000031$. The minimum merit function value is $1.807580103$ when $\epsilon = 0.009548$ and the radii for each surface are $104.395031$ mm and $-105.07512$ mm, respectively. The result is shown in Figure 4.2.

![Graph a) The Merit Function profile of the first lens growth solution and b) the configuration of the first lens growth solution](image)

**Figure 4.2** The Result of local minimum searching from $c_1=0.000031$

For $c_2 = 0.026099$. The minimum merit function value is $2.201306822$ when $\epsilon = -0.005216$ and the radii for each surface are $-31.933578$ mm and $-47.88584$ mm. The result is shown in Figure 4.3.
Both saddle points lead to one common local minimum. For this situation, the minimum merit function value 2.546003429 achieve when $\epsilon = -0.012668$ and the radius for each surface are -79.132705547 mm and -74.454619909 mm.

Since limited operands are included in the merit function and only one lens is inserted, and the other two lenses are fixed as the original setting, there is only a slight improvement for each new configuration. But, the aberrations included in the operands all have been improved. After these three configurations are obtained, the default optimization in Zemax is used to test if these are the best configuration. It shows that the last two solutions merge into the first configuration.
4.2 Example 2: Optimization

In this section, a wide-angle lens assembly patent [7] is taken as the starting point of optimization. The parameters of the original configuration are given in table 4.1.

<table>
<thead>
<tr>
<th>Surface Num.</th>
<th>Curvature(mm)</th>
<th>Thickness(mm)</th>
<th>Nd</th>
<th>Vd</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10.000</td>
<td>0.700</td>
<td>1.7900</td>
<td>52.320</td>
</tr>
<tr>
<td>S2</td>
<td>2.980</td>
<td>2.040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>7.254</td>
<td>0.600</td>
<td>1.7850</td>
<td>59.320</td>
</tr>
<tr>
<td>S4</td>
<td>2.980</td>
<td>2.591</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>50.993</td>
<td>1.400</td>
<td>2.0156</td>
<td>32.590</td>
</tr>
<tr>
<td>S6</td>
<td>-7.086</td>
<td>0.575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S7</td>
<td>∞</td>
<td>0.705</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S8</td>
<td>∞</td>
<td>0.800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S9</td>
<td>-10.722</td>
<td>1.400</td>
<td>1.7809</td>
<td>61.600</td>
</tr>
<tr>
<td>S10</td>
<td>-5.256</td>
<td>0.100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11</td>
<td>9.258</td>
<td>0.600</td>
<td>1.8594</td>
<td>20.700</td>
</tr>
<tr>
<td>S12</td>
<td>3.455</td>
<td>2.430</td>
<td>1.6100</td>
<td>88.670</td>
</tr>
<tr>
<td>S13</td>
<td>-6.232</td>
<td>2.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S14</td>
<td>∞</td>
<td>0.550</td>
<td>1.5168</td>
<td>64.200</td>
</tr>
<tr>
<td>S15</td>
<td>∞</td>
<td>1.616</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Effective Focal Length = 1.998 mm, Field of View =160° |

Table 4.1 The properties of the original patent

And The layout, rayfan of this wide angle lens the ray fan of the original configurations and the Distortion and the Field Curvature from the patent document are shown as in the figure 4.5.
The layout and the performance of the patent

This wide-angle lens configuration is consist of two groups of elements. The effective focal length is 1.998 mm and the half field of view is 80°. The front group is an afocal system to broaden the bundle size, while the second group focuses the rays. To optimize this configuration, the two groups of lenses are analyzed separately.
In this optimization process, material remains the same. The variables are the curvature and the thickness. After optimizing these two groups, the system is reassembled again to form a new system. The method for optimizing lens of using SPC is quite similar to the steps taken in lens growth. Take the first part of this system as an example. There are three elements in this part. The last lens is used to maintain the property of this group, i.e. afocal system. Therefore, only first two lenses are optimized. Firstly, the lens that needs optimizing is replaced by a null element at the same location. Secondly, the SPC script is run to find the saddle point and after that, the local minima are found. By comparing different solutions, the best solution for the configuration. The solutions for the first afocal system part is given in Figure 4.6. The solutions for the second part is given in Figure 4.7.
The optimization results of the second group of lenses have large coma and field curvature. In the individual optimization, since the stop is set at the very front, these aberrations are difficult to eliminate. The effort will be made when combining these two group of lens together.

After simply combining two parts of the system together, some small optimization steps such as adjusting the position of the stop, adding an aspheric surface, and slightly changing the curvature of the surface, etc. are taken to achieved a better performance of the system. It turns out that four configurations go to the same destination and there are total seven configurations found. The
performance of the configurations remains either the same quality or some even achieved a better performance than the original patent. Four best solutions among the seven configurations and their performance are shown in Figures 4.8-4.11.

![Diagram showing ray fans and performance metrics](image)

- a) Layout of the first solution
- b) Performance of the first solution
- c) Field curvature and distortion of the first solution

**Figure 4.8** The results of optimization I

From the Ray fan, it can be observed that, overall, the ray fan scale is smaller than the original configuration. The first optimized result reduces Spherical aberration and Coma, but the performance is limited by Astigmatism especially in the large field. From the Distortion plot, the distortion at the edge is about half of the original configuration.
The second solution provides even smaller Ray Fan scale than the first solution. Same as the first solution, Spherical Aberration and Coma has been significantly reduced. But the second solution still has large astigmatism. The field curvature is smaller than the original patent, and the distortion is at the similar level.
The third solution has the same level of Ray Fan scale and improves spherical aberration as the first solution. Astigmatism remains as the most dominant aberration of all. The Third solution has better Field curvature and the same level of distortion as the original configuration.
The fourth solution has the same level of Ray Fan scale and improves spherical aberration as the first solution. Astigmatism remains as the most dominant aberration, but the Ray Fan of the fourth solution is more smooth than the other solution. The fourth solution has better Field curvature and the same level of distortion as the original configuration.
In this report, several common optimization methods are stated first in the Section I. After briefly comparing those methods; the Saddle Point Construction algorithm is introduced, followed by a theoretical explanation. One important \textit{apriori} result of the Saddle Point Construction is that when a local minimum is found, it’s possible to transform this by increasing the dimension of the landscape. In practice, inserting a non-thickness meniscus or simply splitting the lens with a non-thickness air gap increases the dimension, since more variables are introduced in the merit function landscape. One another important \textit{apriori} knowledge is that the dimensions increased landscape has the property of Cayley cubic. This \textit{apriori} knowledge makes the local searching minimum from saddle point possible that is the minimum of the previous landscape.

Saddle Point Construction have two main implementations, one is lens growth, and the other is optimization. Two examples are given in the report. The lens growth process requires more experience of designers, because more limits and
specifications are needed to be set properly throughout the whole process. SPC used as an optimization method is explored deeper in this report, and one wide angle lens configuration is optimized. Several equal quality and slightly better quality results are found by using SPC. It should be noted that SPC is not the only method used in the whole process. Some conventional techniques of lens design are also taken to help with the optimization process. In most cases, besides giving a better solution by searching local minimum from saddle points, SPC also plays a role in escaping the local trap efficiently. Using SPC tends to find more than one solution. Depending on the limits and specifications, more choices for the application are provided. Additionally, the result of SPC can be used as the starting point of the more advanced optimization.
Appendix A – Zemex Macro - Calculate the partial derivative of Merit function

! Merit-Function Lanscape.zpl
!

OPEN "C:\program\systemsetting.txt"
READ steps
READ dr
READ dc
READ firstsurf
CLOSE

!DECLARE MF_V,double,2,steps
steps = 1000

!the difference between two surfaces
dr = 0

!step length
!pay attention to avoid zero point.
dc = 0.000593

firstsurf = 5
secondsurf = firstsurf

FORMAT 11.9
filename$ = "C:\program\output.txt"

OUTPUT filename$
PRINT "dm","","r"

FOR i = 1,steps,1
! Adjust the radius of the null element
ck = 0.4 - dc*i
ck_p = 0.4 - dc*(i+1)
ck_n = 0.4 - dc*(i-1)

!Calculate the MF of k+1
SURP firstsurf,CURV,ck_p
SURP secondsurf,CURV,1/(RADI(firstsurf)+dr)
   SETOPERAND 1,9,0
   SETOPERAND 2,9,0
   SETOPERAND 3,9,0
   SETOPERAND 4,9,0
   SETOPERAND 5,9,0
   SETOPERAND 6,9,0
   SETOPERAND 7,9,0
   SETOPERAND 8,9,4
OPTIMIZE
   update all

   SETOPERAND 1,9,1
   SETOPERAND 2,9,1
   SETOPERAND 3,9,1
   SETOPERAND 4,9,1
   SETOPERAND 5,9,1
   SETOPERAND 6,9,1
   SETOPERAND 7,9,1
   SETOPERAND 8,9,0
   update all
MC_p = MFCN()

! Calculate the MF of k-1
SURP firstsurf,CURV,ck_n
SURP secondsurf,CURV,1/(RADI(firstsurf)+dr)
   SETOPERAND 1,9,0
   SETOPERAND 2,9,0
   SETOPERAND 3,9,0
   SETOPERAND 4,9,0
   SETOPERAND 5,9,0
   SETOPERAND 6,9,0
   SETOPERAND 7,9,0
   SETOPERAND 8,9,4
OPTIMIZE
   update all

   SETOPERAND 1,9,1
   SETOPERAND 2,9,1
SETOPERAND 3,9,1  
SETOPERAND 4,9,1  
SETOPERAND 5,9,1  
SETOPERAND 6,9,1  
SETOPERAND 7,9,1  
SETOPERAND 8,9,0  
update all  
MC_n = MFCN()  

!Calculate the dM  
dM = (MC_p - MC_n)/(2*dc)  

!R4 = RADI(4)  
!update all  
  !print "the radius is ",1/(0.025-0.0000001*i)  
  !print "the current merit function value is:",MFCN()  
  !MF_V(i) = MFCN()  

PRINT dM," ",ck  
! ,R3," ",R4  

NEXT  

OUTPUT SCREEN  
! Convert the txt file to ANSI  
CONVERTFILEFORMAT filename$, 1
Appendix B – Zemex Macro - Searching local minimum from saddle point

OPEN "C:\program\systemsetting.txt"
READ steps
READ c
READ start
READ e
READ firstsurf
CLOSE

secondsurf = firstsurf + 1
e = e/1000000!
e = 0.000004

FORMAT 11.9
filename$ = "C:\output.txt"
OUTPUT filename$
PRINT "merit_func","","error","","s1_curv","","s2_curv","","s6_curv"

FOR i = 1,steps,1

SURP firstsurf,CURV,c+((start+i)*e)
SURP secondsurf,CURV,c-((start+i)*e)

! Recover the original merit function
SETOPERAND 1,9,0
SETOPERAND 2,9,0
SETOPERAND 3,9,0
SETOPERAND 4,9,0
SETOPERAND 5,9,0
SETOPERAND 6,9,0
SETOPERAND 7,9,0
SETOPERAND 8,9,1
OPTIMIZE
update all

SETOPERAND 1,9,1
SETOPERAND 2,9,1
SETOPERAND 3,9,1
SETOPERAND 4,9,1
SETOPERAND 5,9,1
SETOPERAND 6,9,1
SETOPERAND 7,9,1
SETOPERAND 8,9,0
update all

PRINT MFCN(),",",(start+i)*e,""
PRINT RADI(firstsurf),",",RADI(secondsurf),",",RADI(6)

! Calculate the radius of the last surface to maintain the
! EFFL.

NEXT

OUTPUT SCREEN
! Convert the txt file to ANSI
CONVERTFILEFORMAT filename$, 1
CLOSEWINDOW
Appendix C - Python Script (Plot the result, Data Analysis and main SPC body)

```python
# RESET SYSTEM
from __future__ import division
import math
# import matplotlib as mpl              ### May need to uncomment
# mpl.use('TkAgg')                      ###
from matplotlib import pyplot as plt
plt.style.use('ggplot')
%matplotlib inline
import pandas as pd
import numpy as np
from pandas import DataFrame, Series
from mpl_toolkits.mplot3d import Axes3D

# RESET ZEMAX
import sys
import os
PyZDDEPath = 'C:\Users\colin\Desktop\academy\Research\optimization\data\PyZDDE'
if PyZDDEPath not in sys.path:
    sys.path.append(PyZDDEPath)
#**********************************************************
import pyzdde.zdde as pyz
ln = pyz.createLink()
ln.apr = True
ln.zSetMacroPath('C:\\Users\\colin\\OneDrive\\Documents\\Zemax\\Macros\\')

# Set Input for MFL_NE
#************************************************************

file = open("MF_NE_V.txt", "w")
# steps
file.write("1000\n")
# the difference between two surfaces
file.write("0.01\n")
# step length
# pay attention to avoid zero point
file.write("0.0001\n")
# the first surface number of lens
file.write("1")
```
file.close()

# Pyzdde can only execute the default macro in zemax, so I replaced some macro with my script but keep its original name
ln.zExecuteZPLMacro('CEN', timeout=18000)

# Get the plot,
# remember to modify the figure name
data = pd.read_csv('abc.txt', sep='t')
ax = data.plot.line(x='r', y='dm')
fig = ax.get_figure()
fig.savefig('NullElement_Zpoint.jpg')

# Find the zero point
# In this algorithm, I define the point ahead the crossing is the zero point.
cal = np.zeros(data.dm.count())
for x in range(data.dm.count() - 1):
    cal[x] = data.dm[x] * data.dm[x + 1]
mask = cal < 0
sLength = len(data['r'])
data['MF_min'] = 50 * np.ones(sLength)
data['FirSuR'] = np.zeros(sLength)
data['SecSuR'] = np.zeros(sLength)
data['LaSuR'] = np.zeros(sLength)
data1 = data.loc[mask]
data1.reset_index()

# The number of zero points
ZP_num = data1.dm.count()

Step_ER = 50
StepLen_ER = 0.00008
x = 0

def SetMFER_Input():
    file = open("MF_ER_V.txt", "w")
    # steps
    Step_ER_str = str(Step_ER)
    file.write(Step_ER_str)
    file.write("\n")
    # the curvature
curv_str = str(data1.loc[data1.index[x], 'r'])
    file.write(curv_str)
    file.write("\n")
    # starting point
    start_str = str(startingpoint)
file.write(start_str)
file.write("\n")

# step length
StepLen_ER_str = str(StepLen_ER)
file.write(StepLen_ER_str)
file.write("\n")

# the first surface number of lens
file.write("1")
file.close()

def MFER_Output():
    data1.loc[data1.index[x],'MF_min'] = data_ER.merit_func.min()
    r = data_ER.s1_curv[data_ER.merit_func ==
    data_ER.merit_func.min()].sum()
    data1.loc[data1.index[x],'FirSuR'] = r
    r = data_ER.s2_curv[data_ER.merit_func ==
    data_ER.merit_func.min()].sum()
    data1.loc[data1.index[x],'SecSuR'] = r
    r = data_ER.s6_curv[data_ER.merit_func ==
    data_ER.merit_func.min()].sum()
    data1.loc[data1.index[x],'LaSuR'] = r

    for x in range(0, ZP_num):
        # Reset starting point
        startingpoint = -25
        Step_ER = 50
        StepLen_ER = 80
        SetMFER_Input()

        ln.zExecuteZPLMacro('CHI', timeout=18000)
        data_ER = pd.read_csv('abc.txt', sep=',')
        # ax = data_ER.plot.line(x='error', y='merit_func')
        # fig = ax.get_figure()
        # fig.savefig('c1c2_min.jpg')

        if data_ER.merit_func.loc[0] == data_ER.merit_func.min():
            while data_ER.merit_func.loc[0] ==
            data_ER.merit_func.min():
                startingpoint = startingpoint - 25
                SetMFER_Input()
                ln.zExecuteZPLMacro('CHI', timeout=18000)
                data_ER = pd.read_csv('abc.txt', sep=',')
            else:
                Step_ER = 1000
                startingpoint = startingpoint * 20
                StepLen_ER = 4
                SetMFER_Input()
                ln.zExecuteZPLMacro('CHI', timeout=18000)
                data_ER = pd.read_csv('abc.txt', sep=',')
                MFER_Output()
        elif data_ER.merit_func.loc[data_ER.merit_func.count()-1] ==
        data_ER.merit_func.min():
while data_ER.merit_func.loc[data_ER.merit_func.count() - 1] == data_ER.merit_func.min():
    startingpoint = startingpoint + 25
    SetMFER_Input()
    ln.zExecuteZPLMacro('CHI', timeout=18000)
    data_ER = pd.read_csv('abc.txt',sep=',',)
else:
    Step_ER = 1000
    startingpoint = startingpoint * 20
    StepLen_ER = 4
    SetMFER_Input()
    ln.zExecuteZPLMacro('CHI', timeout=18000)
    data_ER = pd.read_csv('abc.txt',sep=',',)
    MFER_Output()
else:
    Step_ER = 1000
    startingpoint = startingpoint * 20
    StepLen_ER = 4
    SetMFER_Input()
    ln.zExecuteZPLMacro('CHI', timeout=18000)
    data_ER = pd.read_csv('abc.txt',sep=',',)
    MFER_Output()

Sol = data1.loc[data1.MF_min == data1.MF_min.min()]
Sol.reset_index()
curv1 = 1/Sol.FirSuR.sum()
curv2 = 1/Sol.SecSuR.sum()
curvL = 1/Sol.LaSuR.sum()

ln.zSetSurfaceData(surfNum=1,code=ln.SDAT_CURV,value=curv1)
ln.zSetSurfaceData(surfNum=2,code=ln.SDAT_CURV,value=curv2)
ln.zSetSurfaceData(surfNum=11,code=ln.SDAT_CURV,value=curvL)

# Set Input for MFL_THICKNESS&DISTANCE
file = open("MF_THIC_V.txt", "w")

# steps thickness
steps_thic = 80
Steps_thic_str = str(steps_thic)
file.write(Steps_thic_str)
file.write(" \n")

# steps thickness
steps_dis = 150
Steps_dis_str = str(steps_dis)
file.write(Steps_dis_str)
file.write(" \n")

# the first surface number of lens
# make sure the second surface setting
# before stop: second surface = first surf + 1
# after stop : second surface = first surf - 1
# modify in macro
file.write("1 \n")
file.write("2")
file.close()

ln.zExecuteZPLMacro('EDG', timeout=18000)
#Get the plot,
#remember to modify the figure name
data = pd.read_csv('abc.txt', sep=',')

X = np.arange(1, 8 ,0.1)
Y = np.arange(3, 15.1, 0.1)
X, Y = np.meshgrid(Y, X)
Z = data.as_matrix()

fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(X, Y, Z, rstride=5, cstride=5,

ax.set_xlabel('distance(after)')
ax.set_ylabel('thickness')
ax.set_zlabel('merit function')
ax.view_init(azim=-70)

plt.show()
fig = ax.get_figure()
fig.savefig('thic vs distance.jpg')

plt.contourf(X, Y, Z, 20, alpha=.75, cmap=plt.cm.Blues_r)
C = plt.contour(X, Y, Z, 20, linestyles = 'dotted',
colors='black', linewidth=.1)
plt.clabel(C, colors = 'k', fmt = '%3.2f', fontsize=8)
plt.xlabel('distance(after)')
plt.ylabel('thickness')
References


