# Enhancements to the optimisation process in lens design (I)

Geoff Adams*, Rainer Schuhmann**

*The Optical Software Company, United Kingdom
** LINOS Photonics GmbH, Germany

## ABSTRACT

The full optimisation process includes definition and maintenance of the merit function, selection of optimisation strategy, the actual automatic improvement of the system, designer intervention and review of the final results. Well optimised systems are almost always achieved, not by a single run, but by repeated backtracking and re-optimisation. A number of improvements to the whole process are presented, both to the damped least squares algorithm itself and to the 'handling' & review capabilities. The enhancements include a new method of controlling variable glasses and two simple extensions to the classical damped least squares. These enhancements have been implemented in a widely available software package.

## 1. INTRODUCTION

Practically all reported developments in lens optimisation have consisted of technical details of the algorithms used. However, for the practicing designer, the whole life cycle is important. This starts with the creation/editing of a merit function, involves choice of variables, selection of optimisation strategy, control of variables to practical values, review and re-run of the optimisation and even re-use of the merit function for related systems.

This paper details both life cycle and algorithmic improvements, as implemented in WinLens. First a brief description is given of merit function generation/maintenance tools. This is followed by a discussion of more tools to enable simple and effective review and re-start of an optimisation. These include 'video controls' and 'book marking'.

Two extensions to standard DLS are reported upon, involving solution scaling for improved speed and line search methods which trade speed for depth of search.

Finally a new method for controlling variable glasses is presented. This aims to keep variable glasses near real glasses in n-V space, by the use of a glass 'contribution' function – which is based upon the 'occupancy' of a grid in n-V space.

## 2. MERIT FUNCTION MAINTENANCE

The definition of the merit function and the use of 'relative defects' has been discussed in detail in the sister paper. However little attention was given there to the definition & maintenance process.

One of the major bane's of using old optical design software was the process of creating and maintaining the merit function. Nowadays, of course, most serious programs, including WinLens, use wizards to create a starting merit function.

The merit function is composed of a number of 'defects'. Each defect is the difference between the actual value of some quantity and the target value for that quantity. A defect will usually refer to some ray based aberration, but may include paraxial constraints, system parameter values or even combinations of other defects.

Therefore if some defects refer to specific surfaces or 'earlier' defects, changing or maintaining the structure of the merit function can be non trivial, as for example when components are inserted/reversed or removed or other defects are inserted or

removed. Even less involved tasks, such as changing parameters for a block of defects can be time-consuming and error prone if performed one at a time.

A number of tools have therefore been developed to make this process as painless as possible:

- Cut/copy and paste: one or more defects within the current merit function.
- Drag & drop: from a list of available defects to the current merit function.
- References to parameters: defects referring to specific surfaces remain in proper synchronization even if other components are added/removed, or the parent component is reversed.
- References to defects: defects referring to earlier defects remain in proper synchronization after cut & paste or drag & drop operations in the merit function.
- Automatic variable control: [optional] create defects to control variables within upper and/or lower limits.
- Automatic edge control: [optional] create defects to constrain edge thickness in all spaces, even if components are added/removed.
- Block editing: change defect type or defect parameter for multiple aberrations simultaneously. For defect parameters with a continuous range [such as ray field and aperture co-ordinates], an option to scale the existing values is available.
- Minimal ray set: program automatically deduces smallest ray set necessary for defects in merit function.

These tools, though not perhaps optically interesting, have been found to make the designers overall labours much lighter.

## 3. OPTIMISATION FEEDBACK & REVIEW

It has been said that an unexamined life is not worth living[i]. This may or may not be true, but an unexamined optimisation may well miss a much better solution.

In comparative trials of lens design software, it is always the skill of the designer, which counts. The final polished solutions are the result of repeated trials and modifications, going back over old ground, branching off at some mid point and starting again. To a designer involved in this process, one of the most frustrating things in optimisation is seeing an interesting system flash by, to be lost forever. This frustration is only surpassed in black box software, where no intermediate solutions are displayed at all.

In the light of this, simple tools have been developed to make easy the review/re-run process. First, during optimisation, the program automatically records the data for some or all intermediate solutions to a dedicated folder.

When the optimisation is completed, simple video controls are enabled. These allow the designer to re-run the optimisation in either direction, to jump to the start or end or to single step in either direction. Optimisation can be restarted from any of these solutions.

During optimisation the program also maintains a log of the merit function. This can also be displayed, and the user may go straight to any of the recorded solutions, by clicking on the log spreadsheet.

Once optimisation is restarted, the log is cleared, and the old intermediate solutions are overwritten. Therefore to preserve interesting solutions for later inspection, a book-marking capability has been added. When required by the designer a copy of the current design is saved automatically to a dedicated folder. A list of all book-marked systems, showing time of creation, merit function value and any annotation is available. Then, also when required by the designer, any system in the list will be displayed. This of course is available at any time.

## 4. OPTIMISATION EXTENSION I

The first extension of the damped least squares method was initially reported by Robb[ii]. Because this has been found to give good results and because it leads directly to the second enhancement, some space will be devoted to this extension.

This starts with a very brief review of standard damped least squares. The merit function, $\varphi$, is defined as the sum of the squares of a number of defects, $f_i$. The aim is to minimise the merit function.

Initially assume that all defects are purely linear functions of the variables, $x_j$. In this case all higher order terms are identically zero, so:

$$\boldsymbol{f} = \boldsymbol{f_o} + \boldsymbol{A}.\boldsymbol{x} \tag{1}$$

and the merit function therefore becomes:

$$\varphi = |\boldsymbol{f_o} + \boldsymbol{A}.\boldsymbol{x}|^2 \tag{2}$$

In such a case the merit function would be plotted as a set of concentric 'ellipsoids' in variable space. After expansion and some matrix manipulation, it can be shown that:

$$\varphi = \boldsymbol{f_o}\,\boldsymbol{f_o} + 2\,\boldsymbol{A}^T\boldsymbol{f_o} + \mathbf{x}.\,\boldsymbol{A}^T\mathbf{A}\mathbf{x} \tag{3}$$

and the gradient vector is:

$$\boldsymbol{G} = 2\,\boldsymbol{A}^T\boldsymbol{f_o} + \boldsymbol{A}^T\mathbf{A}\mathbf{x} \quad [\text{or } \boldsymbol{G} = \boldsymbol{G_o} + \boldsymbol{A}^T\mathbf{A}\mathbf{x}] \tag{4}$$

Now, at the minimum of the merit function, $\boldsymbol{G}$ is by definition zero, therefore:

$$2\,\boldsymbol{A}^T\boldsymbol{f_o} = -\,\boldsymbol{A}^T\mathbf{A}\mathbf{x} \tag{5}$$

It is then 'simple' to obtain the solution vector x. This is the least squares solution, and if there are no non-linearity's this would yield the true merit function minimum in a single step. Of course there are almost always non-linearity's. Small non-linearity's means that the minimum would be reached after a few iterations, but large non-linearity's lead to diverging solutions, i.e. the real merit function value gets worse! What is required is some method of constraining the solution vector [i.e. size of the changes in the variables]

This may be achieved by adding another term to the merit function which depends upon the size of the solution.

$$\varphi = |\boldsymbol{f_o} + \boldsymbol{A}.\boldsymbol{x}|^2 + |p\boldsymbol{x}|^2 \tag{6}$$

Where p is know as the damping factor. When (6) is solved, the new solution is given by:

$$2\,\boldsymbol{A}^T\boldsymbol{f_o} = -\,(\boldsymbol{A}^T\mathbf{A} + \mathbf{p}^2\mathbf{I})\mathbf{x}' \tag{7}$$

Note, damping does not take non-linearity's into account, it merely tries to restrict the solution size so that non-linearities are not significant. Therefore the best damping factor has to be established empirically.

If p is zero, there is no damping and eqn 7 reduces to the least squares solution of eqn 5. If p tends toward infinity, then the solution vector becomes parallel to the gradient, $\mathbf{G_0}$, at the starting point, i.e. same direction as the steepest descent method. The actual solution magnitude is, of course, tiny.

To summarize, damping not only alters the magnitude of the solution, it also alters the direction. When a new damping factor is selected, a new direction in solution space is defined.

However, because the damped equations do not take non-linearities into account, there is no guarantee that the damped solutions are minimal. The solution, **x,** will not lead to the true minimum, it also may not give the smallest value for the merit function along the direction $\mathbf{x}/|\mathbf{x}|$. From this it follows that scaling the solution, k.**x**, may well yield improvements in the merit function.

This is now standard in WinLens, as it has repeatedly proved useful. In each cycle, the best damping factor is found. The solution is then scaled repeatedly until the merit function ceases to fall. This solution is then taken as the start point for the next cycle. A clear example is shown below.

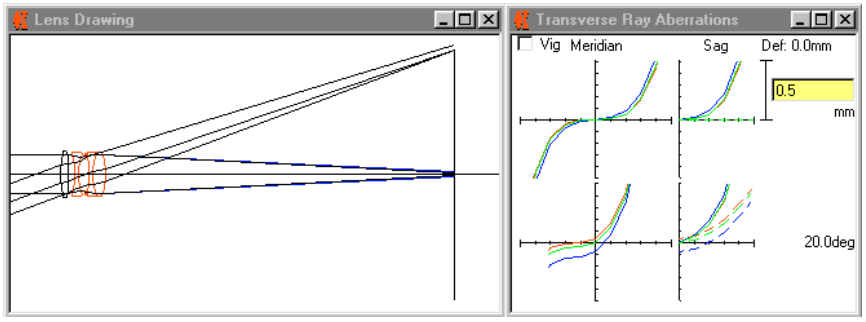



Fig. 1: System at start of optimisation

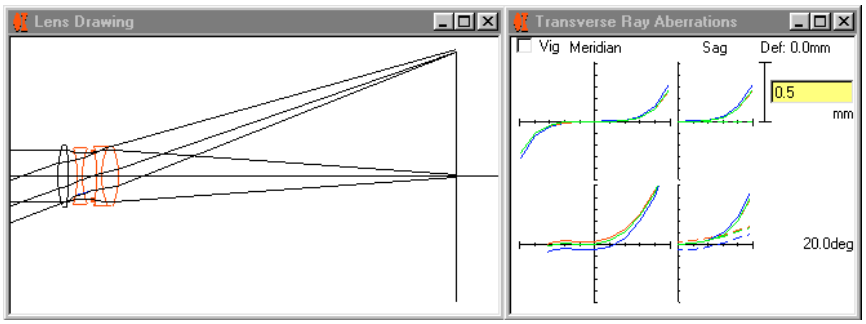



Fig. 2: System generated by best damping factor.

[application of solution vector **x**, after one standard cycle]

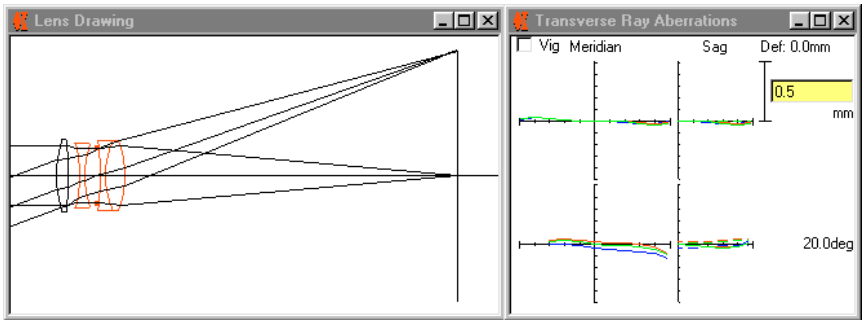



Fig 3: System generated by subsequent scaling.

[application of scaled solution vector k.**x**]

## 5. OPTIMISATION EXTENSION II

The second extension is a logical development of the first. As before several damping factors are evaluated. However for each damping factor, the program will scale the solution until the best result is obtained along that direction. The results of all directions are compared and the best taken.

Clearly each cycle can take much longer to complete. However it is possible that by evaluating directions with less initial promise we may find different minima than revealed by the standard damped least squares. Simple speed is traded for breadth of exploration.

Fig 4: Standard DLS optimization.

Solutions are obtained for different damping factors.

The solution which yields the smallest value of the merit function is taken as the starting point for the next cycle.

By taking the most opportunistic solution, a better minimum is missed. Further optimization will only reach the lesser minimum.

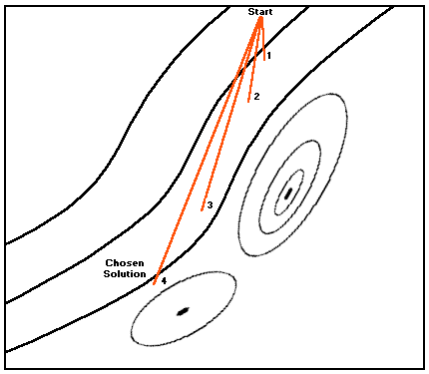


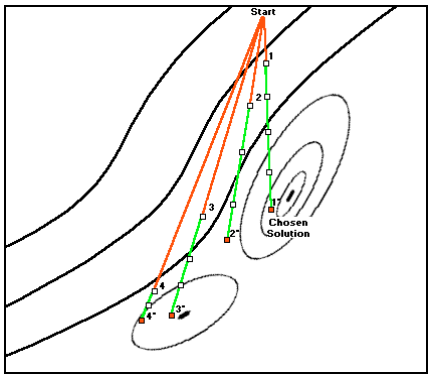

Fig 5: DLS optimisation with line search.

Solutions are obtained for different damping factors.

Each solution is scaled until the merit function ceases to drop.

The scaled solution which yields the smallest value of the merit function is taken as the starting point for the next cycle.

By taking more time a better minimum is reached.




This has been implemented as an option in WinLens, since it only sometimes yields a better minimum.

## 6. GLASS CONTROL

There are a moderate number of real glasses and therefore a discrete set values of index & dispersion values. Since DLS optimisation requires continuous variables, optimisation of glasses implies the use of a theoretical model to predict indices at analysis wavelengths as n & V are varied.

At the end of optimisation, these variable glasses must be substituted with real alternatives, so it is critical that these glasses stay in the region of the real glasses and not stay into uninhabited areas.

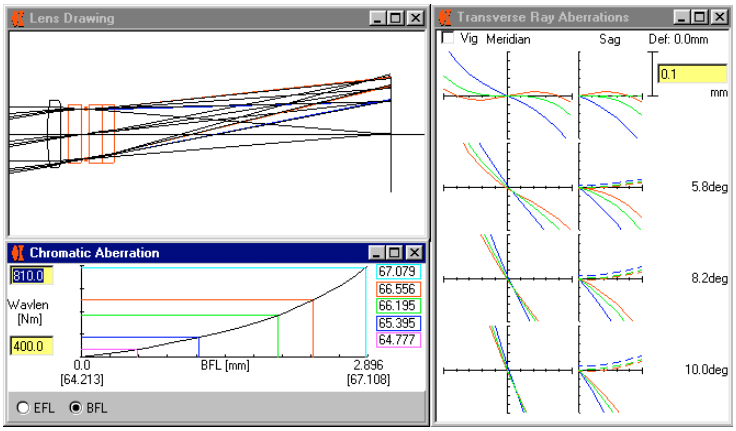
In the past, glass triangles/polygons were used to map the boundary of the real glass region, with glasses moving outside being controlled by boundary violations. This has obvious disadvantages.

A new approach has been developed. The glass map is transformed into a number density or ‘contribution’ function This function is small in regions where there are many glasses, higher in regions with a few glasses, and increases very rapidly as the n-V value moves outside the real glass area.

The glass contribution function is used by the ‘automatic’ glass defects [one for each variable glass]

The results of optimisation on a ‘proto’ tessar may be seen in figs 6-9.

Fig 6: starting system



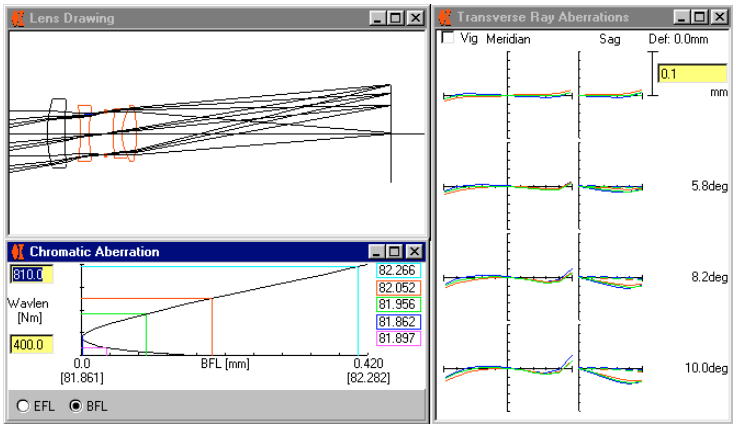

The ‘tessar’ only has power on the first component, all other are plane plates.

**All** glasses are initially BK7, even in the doublet.

All glasses are variable [both index and dispersion]

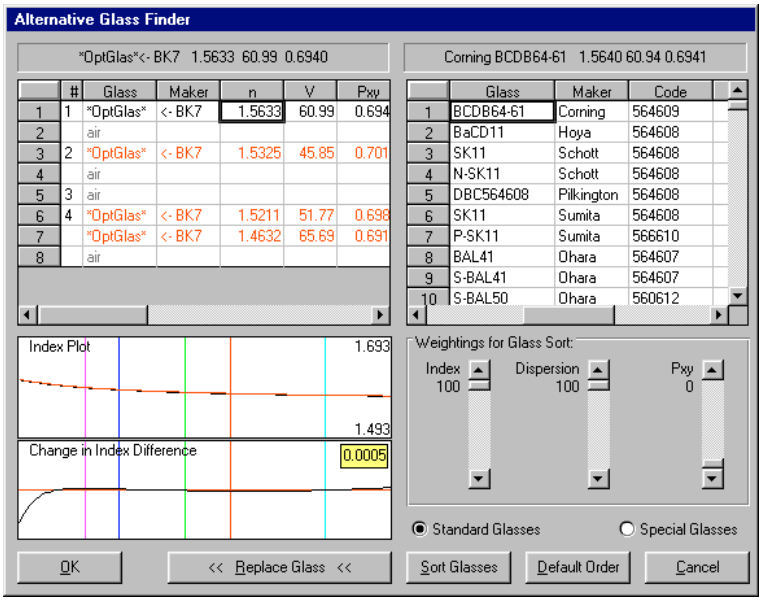
All curvatures & separations are also variable.

Fig 7: optimised system but with variable glasses.




The glasses have only been restrained by the automatic glass defects. There were no manual limits placed on the range of n or V.

Fig 8: substitution of real glasses with the ‘alternate glass finder’ in WinLens.




Note:

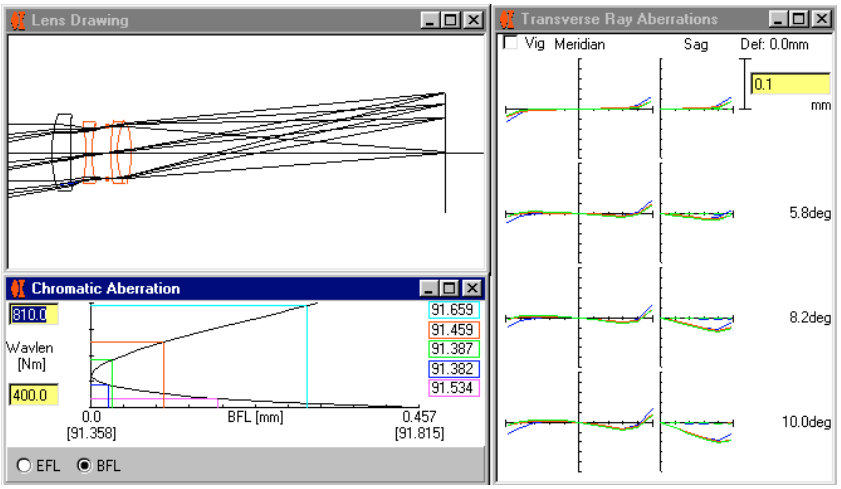
1) All variable glasses are clearly within the real glass region, and close to actual glasses.

2) The doublet glasses have split apart in property during optimisation to provide chromatic correction.

3) the theoretical model of a glass nicely replicates that of a real glass over the spectrum, even though it is only defined by two points

Fig 9: Final system.

System with all replacement real glasses, re-optimized with separations and curvatures only




The glass 'contribution' function may be determined by the user. The user may specify:

- Coarseness of grid on which the initial analysis is made
- Glass manufacturer whose glasses are to be included in the generation of the function
- Glass types [recommended, eco friendly, available and/or obsolete] to be included in the generation of the function

## 7. SUMMARY

[i] Socrates

[ii] Paul N Robb. 'Accelerating convergence in automatic lens design'. Applied Optics, 1979, vol 18, p 4191