

New tools for the lens designer

Donald C. Dilworth

Optical Systems Design, Inc.

ABSTRACT

Several common impediments to successful lens design can be removed by application of new techniques. These involve using information that has long been calculated during the design process, but then discarded. Problems so addressed include the nuisance of discovering ray failures in the starting configuration, and that of tolerance desensitization, among others.

Keywords: Lens optimization, ray failures, tolerance desensitization, starting point, stop position, avoiding strain.

1. INTRODUCTION

Optical design codes today have capabilities that were not imagined at the outset, roughly 40 years ago, when we were first trying to figure out how to make them work. Since then, although algorithms have matured and a vast amount of information is calculated, designers still face some practical difficulties. These can be remedied with clever use of information that is already available but usually discarded. Several areas respond well to the new tools that this idea leads to. The first is the common situation when rays will not trace through the intended starting configuration, causing the designer to waste time just trying to get started. Others involve use of 3rd and 5th-order aberrations in a new way.

2. RAY FAILURES

It is always wise to begin a lens design job with a starting point that closely resembles the desired end point, but often nothing suitable is available. We sometimes begin by sketching a configuration on a napkin, for want of anything better. Frequently, though, when we try to optimize the lens derived from such a sketch, some of the rays specified in the merit function do not reach the image, and the merit function cannot be evaluated. Every commercial code, as far as we know, would stop at that point, and the designer then has to edit the starting lens, either manually or with a GUI interface with slider bars, if his program provides that functionality, until all of the rays trace. Only then can the optimization start in earnest.

There are two reasons why rays sometimes fail to trace: missed conic section (MCS), or total internal reflection errors (TIR). In both of these cases, the failure has to do with taking square roots. No matter how the code is written, since one is solving a second-order equation, a radical is involved. If the argument of the radical for the intersection calculation is negative, an MCS error has occurred. And if the radical in the refraction equation is negative, it is a TIR error.

Every code calculates those arguments – and then throws the result away, if it is negative, and aborts the process. But the values of the arguments contain useful information. If the value were exactly zero, for example, the intersection equations put the ray exactly at the tangent point of a spherical surface. And if the refraction argument is zero, the ray is refracted exactly at the critical angle.

This suggests a simple tool, which we have used with great success: When the program finds a ray in the merit function that fails, it remembers which ray was involved, and then creates another merit function that contains only a single quantity, namely the argument that was negative, and gives a small positive target value as the goal. Then, utilizing the same design variables as already specified, it runs a quick optimization behind the scenes. The offending argument quickly becomes less negative, and as soon as it reaches a positive value, that optimization ends and the original merit function is restored. The entire process then starts again, and the offending ray then traces properly. Of course there may be other rays that still fail, but they are easily corrected by means of a second click on a new toolbar button we have

implemented to exercise this tool. One or two clicks on that button usually start the optimization going as originally intended, with a considerable saving of time.

2.1 Example of ray failure correction

An idea one would often hear, at the birth of this technology, was that a lens design program, if it were really good, could start with a collection of plane-parallel plates and come up with an excellent lens. This idea was short lived, as it turned out, once we got a better idea of what we could accomplish in those days. To illustrate the convenience offered by this new tool, however, we will resurrect that notion and show that, today, it is much more practical than it was then. This will also serve as a very bad starting point that will illustrate the new feature. This is shown in Figure 1.

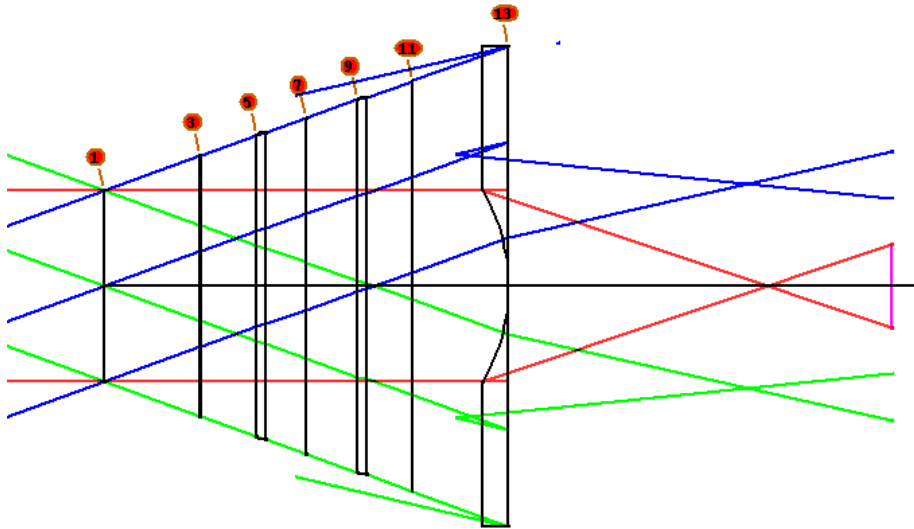


Figure 1. A very poor starting point.

We have created a seven-element lens where all surfaces are flat but one, all the glass types are the same (in the middle of the glass map), and the last surface has been given a paraxial curvature solve that yields an F/number of 2.0. The final airspace is also controlled by a paraxial solve, finding the paraxial focal point. All of the optical power is therefore on the last element, and rays will not trace over the full pupil at the edge of the field, which is at 20 degrees to the axis. The stop is arbitrarily put on the first element.

Let us try to optimize this lens. To do so, we will specify a target of 85 mm for the total length, 25 mm for the back focal distance, and correct the ray intercept errors at three field points. All design parameters are variables, as defined in the following optimization MACro. (GLM refers to glass model variables, and we'll discuss the variable YP1 below.)

```
PANT
VY 0 YP1
VLIST RAD 1 2 3 5 6 7 8 9 10 11 12 13
VLIST TH 1 2 3 4 5 6 7 8 9 11 12 13
VLIST GLM 1 3 5 7 9 11 13
END
```

```
AANT
AEC
ACC
M 85 .1 A TOTL
M 25 .1 A BACK
GSR 0 5 4 1 0 0
GSR 0 5 4 2 0 0
```

```
GSR 0 5 4 3 0 0
GNR .75 1 3 1 .75 0
GNR .75 1 3 1 1.0 0
GNR .75 1 3 2 .75 0
GNR .75 1 3 2 1.0 0
GNR .75 1 3 3 .75 0
GNR .75 1 3 3 1.0 0
END
```

```
SNAP
SYNO 30
```

As expected, we immediately get the error message

```
The optimization process is halted
ERROR: Fatal ray error occurred on the first iteration
```

Instead of trying to alter the design by changing radii, as we hitherto used to do, now we just click on the new tool (twice, in this case), and the optimization takes off. About one minute of simulated annealing then yields the lens shown in Figure 2.

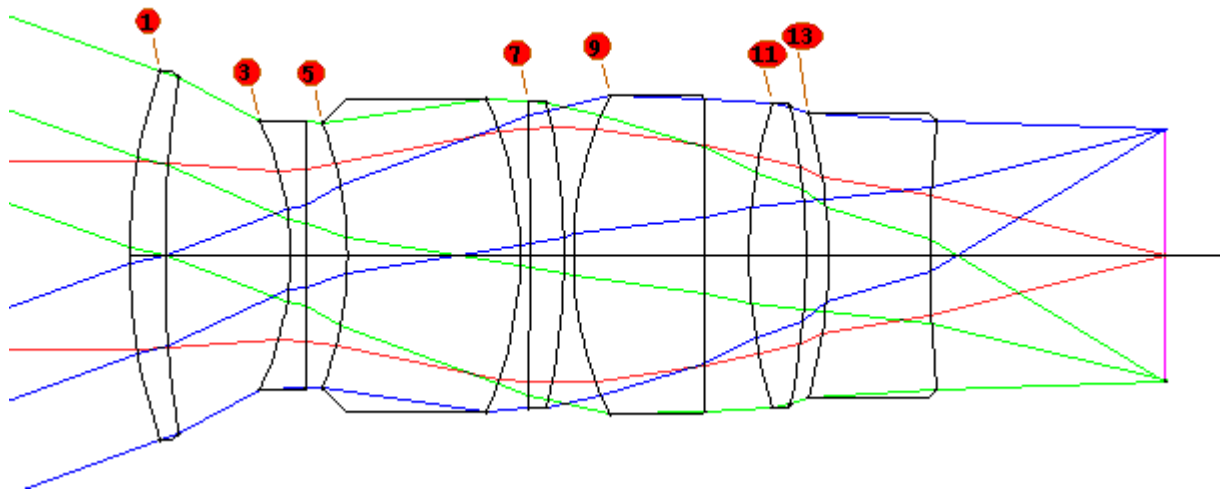


Figure 2. The result of exercising the ray-failure correction tool and optimizing the lens.

A practical design job would require some additional attention, primarily to reduce the thickness of element number three – but to get this far required almost no manual intervention, which is the main point of the new tool.

3. TOLERANCE DESENSITIZATION

Many lens designers have discovered to their dismay that the lens they have worked on so hard will be very expensive to manufacture, due to excessively tight tolerances. Some efforts have been made to implement tolerance desensitization algorithms to help avoid this difficulty, but none have proved popular – and some designers do not concern themselves with the tolerance budget at all, leaving that chore to the customer who will actually build the lens. This is a mistake: if tolerances can be relaxed, it is the designer's job to find out.

Here is another area where calculations that are readily available but not utilized can play an important role. Every program can compute the 3rd-order aberrations, in the process finding the contributions to the sum due to the individual surfaces. Let us look at a portion of a common derivation of those aberrations. (Notation is after H.A. Buchdahl¹.)

At surface j:

$$IA = YA_j c_j + UA_{j-1}$$

$$IB = YB_j c_j + UB_{j-1}$$

$$SA = YA (N_{j-1} / N_j) (N_{j-1} - N_j) (UA + IA)$$

$$SB = YB (N_{j-1} / N_j) (N_{j-1} - N_j) (UB + IB)$$

$$T = (N_{j-1} - N_j) (CC_j c_j + 8DD_j) YA^4$$

and the third-order contributions are:

$$B = SA IA^2 + T \quad (SA3)$$

$$F = SA IA IB + (YB_j / YA_j) T. \quad (CO3)$$

The total aberration is just the sum of the surface contributions, normalized in some way. Thus, the total spherical aberration is just the sum of the values of the quantity B, while the total coma involves three times the sum of the quantity F. (μ is a constant.)

$$SA3 = \mu \sum B$$

$$CO3 = 3 \mu \sum F$$

Consider the lens in Figure 3 – which has the kind of tolerances that are likely to provoke a labor dispute in the shop. Of particular concern are the centration tolerances, in some cases only three microns. This is a very expensive lens indeed. Now, consider an element with a large individual contribution to third-order spherical aberration (SA3) or coma (CO3). Usually there is another element that compensates with aberrations of the opposite sign, resulting in small net aberrations – but when either of those elements is misaligned, the compensation is imperfect. This seems to be the case with elements 3 and 4 in this lens. In such a case, the residual is large because the individual contributions were large. If one could reduce the contributions of the individual elements (or surfaces) one would therefore expect tolerances to come out looser, since the cancellation then need not be so perfect. (In these examples, we calculated the tolerance budget such that the wavefront variance would not exceed 0.05, to a three-sigma confidence.)

Some designers, aware of this effect, have adopted the practice of including a target in the merit function to control the aberrations of one or more individual elements. But there is an easier way – and because it is easier, it is more likely to be frequently utilized. Let us define eight new aberrations that can be included in the merit function definition:

SAT	The sum of the squares of the surface contributions to spherical aberration, SA3.
COT	The sum of the squares of the surface contributions to coma, CO3.
ACD	The sum of the squares of the amount by which CO3 varies as each surface is decentered.
ACT	The sum of the squares of the amount by which CO3 varies as each surface is tilted.
ECD	The sum of the squares of the amount by which CO3 varies as each element is decentered.
ECT	The sum of the squares of the amount by which CO3 varies as each element is tilted.
ESA	The sum of the squares of the element contributions to spherical aberration, SA3.
ECO	The sum of the squares of the element contributions to coma, CO3.

The first two of these new aberrations are easily calculated, once the contributions are known. Aberrations ACD and ACT can be obtained by differentiating the Buchdahl formula with respect to decenter or tilt. The last four keep track of the element contributions rather than surface contributions.

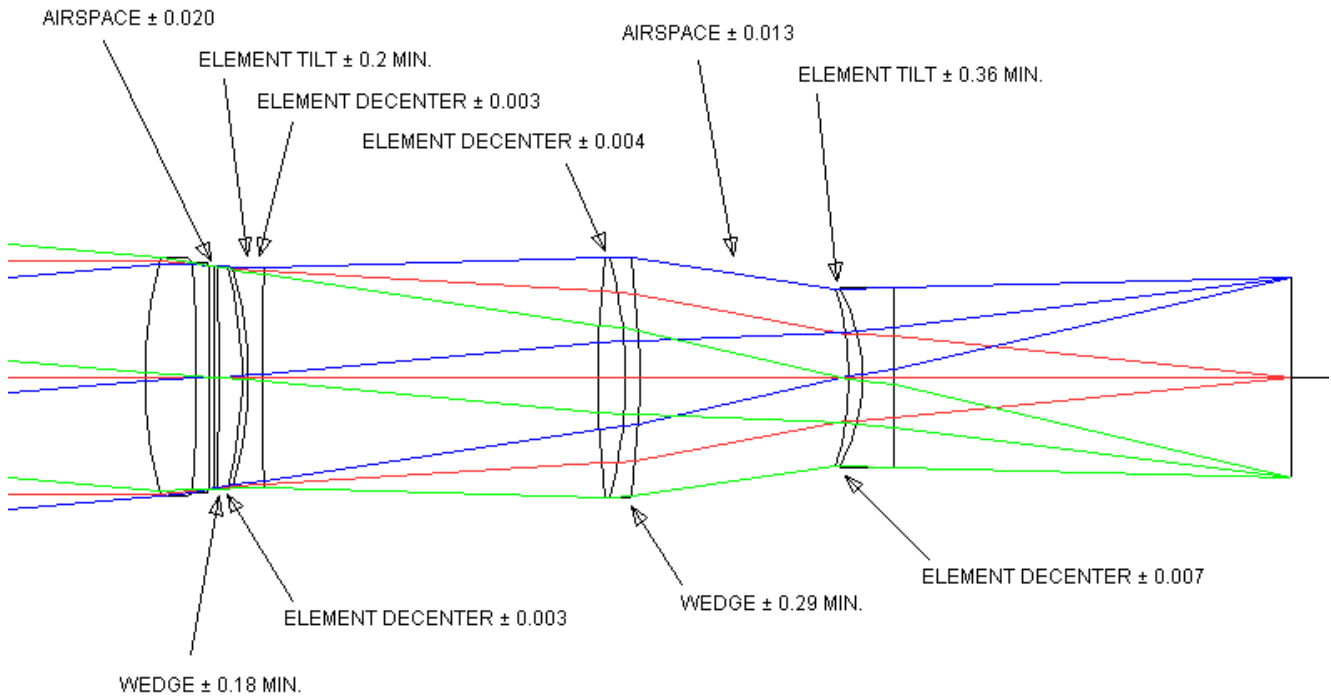


Figure 3. Example lens with very tight tolerances. (The nominal design.)

Now let us show the effect of including some of these new aberrations in the merit function. First, we look at the present values of these quantities, produced by the command THIRD SENS:

NORMALIZED 3RD-ORDER ANALYSIS OF TOLERANCE SENSITIVITY

```

SS OF SA3 BY SURFACE (SAT) =      85.107903
SS OF CO3 BY SURFACE (COT) =      21.404938
SS OF CO3/YDC BY SURFACE (ACD) =    0.007657

SS OF CO3/TILT BY SURFACE (ACT) =   73.889722
SS OF CO3/YDC BY ELEMENT (ECD) =    0.003941
SS OF CO3/TILT BY ELEMENT (ECT) =   31.259708
SS OF SA3 BY ELEMENT (ESA) =        1.944190
SS OF CO3 BY ELEMENT (ECO) =        0.492351

```

Since decenters seem to be the big problem here, we will first try the aberration ECD, with a target value of 0.001. This will reduce the change in coma as elements are decentered, and we hope it will reduce the centration sensitivity. After reoptimizing, the resulting lens has changed somewhat, as shown in Figure 4.

Although we controlled only one of the new aberrations, they have all changed, since things tend to be coupled, and aberration ECD is approximately at the value we requested.

NORMALIZED 3RD-ORDER ANALYSIS OF TOLERANCE SENSITIVITY

```

SS OF SA3 BY SURFACE (SAT) =      5.617179
SS OF CO3 BY SURFACE (COT) =      4.163698

```

```

SS OF CO3/YDC BY SURFACE (ACD) = 0.001758
SS OF CO3/TILT BY SURFACE (ACT) = 18.629842
SS OF CO3/YDC BY ELEMENT (ECD) = 0.001155
SS OF CO3/TILT BY ELEMENT (ECT) = 8.206368
SS OF SA3 BY ELEMENT (ESA) = 0.149362
SS OF CO3 BY ELEMENT (ECO) = 0.112059

```

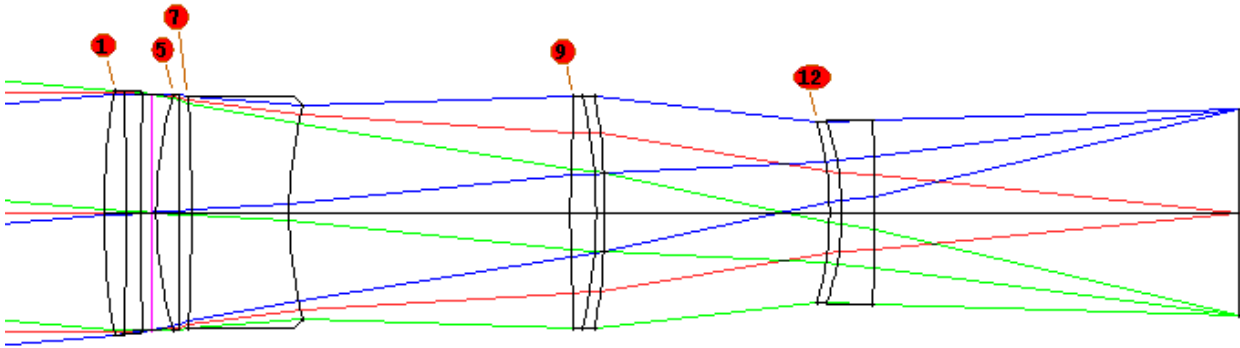


Figure 4. Case A: Lens reoptimized with added control over the new aberration ECD.

There are other possibilities. What happens if we instead control the value of the quantity COT, the sum of the squares of surface coma? This aberration had a value of 21.4 in the nominal design, and came out reduced to 4.16 in Case A. We will request a target value of 0.2, reoptimize, and examine the results, shown in Figure 5.

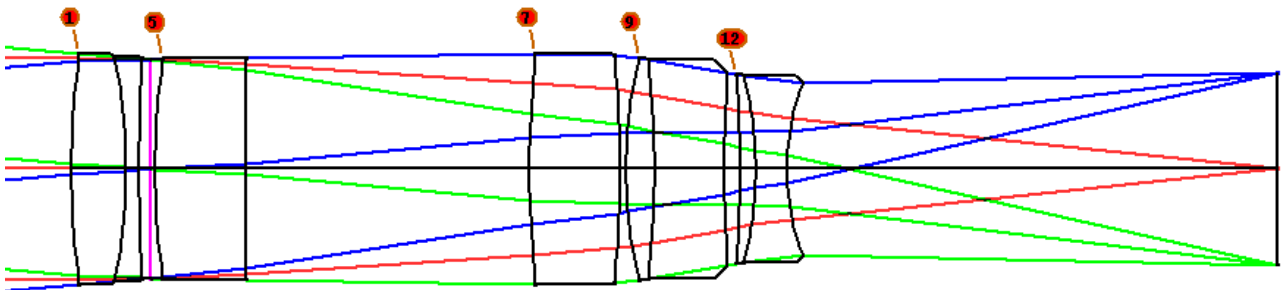


Figure 5. Case B: Lens reoptimized with added control over the new aberration COT.

Now we see a major change to the configuration. The value of aberration COT is close to its target, as shown below.

NORMALIZED 3RD-ORDER ANALYSIS OF TOLERANCE SENSITIVITY

```

SS OF SA3 BY SURFACE (SAT) = 1.328837
SS OF CO3 BY SURFACE (COT) = 0.205343
SS OF CO3/YDC BY SURFACE (ACD) = 0.000337
SS OF CO3/TILT BY SURFACE (ACT) = 12.232184
SS OF CO3/YDC BY ELEMENT (ECD) = 0.000159
SS OF CO3/TILT BY ELEMENT (ECT) = 7.560905

```

SS OF SA3 BY ELEMENT (ESA) = 0.035610
 SS OF CO3 BY ELEMENT (ECO) = 0.007259

Comparing the tolerances for the three cases, some of which are shown below, shows the benefit of this additional optimization. (YDC means Y-decenter.)

	<u>3 TH</u>	<u>6 wedge</u>	<u>7 tilt</u>	<u>5 YDC</u>	<u>7 YDC</u>	<u>9 YDC</u>	<u>12 YDC</u>
Nominal	0.012	0.18 min	0.20 min	0.0036	0.0027	0.0042	0.0071
Case A	0.119	0.81 min	0.52 min	0.012	0.010	0.013	0.013
Case B	0.088	1.53 min	2.77 min	0.036	0.034	0.026	0.030

It seems that the very tight tolerances of the nominal design are relaxed by a factor of from three to about 10 in these examples. One would expect that wedge tolerances would be most affected by the aberration ACT, while airspace tolerances might respond best with a target for SAT or COT. Like all new tools, these features invite exploration, as their utility, power, and limitations are demonstrated by experience.

4. CONFIGURATION HELP – FINDING THE STOP POSITION

When laying out a potential starting lens configuration, placing the stop in an advantageous position has a major influence on the success of the resulting lens. But if no suitable configuration is available to use as a model, one must simply guess where the stop should go. This is another example where using old information in a new way creates a useful design tool.

Every program contains a ray-aiming algorithm to find where to aim the incoming rays so that the chief ray will go through the center of the stop. This is usually done in steps, beginning with a paraxial calculation, and then again with real rays if the situation warrants. The paraxial calculation yields the intercept height of the full-field chief ray on surface 1, and other things are usually scaled from that result.

Let's turn it around. Instead of declaring a stop surface and then solving for the paraxial chief-ray height, start with a given height and see where the stop wants to be. This height, which we call YP1, can then be used as a design variable. After the lens is optimized, one just looks to see where the stop came out, and then places it close to that point and tweaks up the design.

Some designers resort to the old trick of putting the stop on a dummy surface 1 and varying the position of the first element relative to that surface. This is also a good way to find the best stop position, but since people tend to avoid things that are inconvenient, this possibility is overlooked more often than not.

In the example lens in Figure 1, we arbitrarily put the stop on surface 1, and the position was then allowed to vary by means of the design variable YP1. The result in Figure 2 shows that the stop wants to be in the middle of the third element. What happens if we start with the stop on the *last* surface? This new design tool (and the ray failure correction feature) makes it very easy to explore questions of this kind. After a few optimization cycles, the lens comes out as shown in Figure 6. Again, the stop has moved toward the center of the lens, in this case inside the fourth element. Yet another configuration results if we start with the stop near the center of the lens, as shown in Figure 7.

In these examples we started with a very poor lens and, by a slight change in initial conditions, derived three quite different configurations. Why? The answer suggests some profound implications.

When you design a lens, you are exploring *geography*. The lens design space contains many dimensions, with hills and valleys, and the designer's job is to find the lowest valley. Figure 8 shows such a situation, in two dimensions. Suppose your design is in the valley labeled A. Running the simulated annealing feature can in some cases move the design to a lower valley, as at B. But if you are in either of those valleys, you cannot see the deeper valley at C. In fact, the only

place from which you can see all of the valleys is from the top of the hill. And what is on top of that hill? *Plane parallel plates!* Here, the idea we resurrected from the earliest attempts at computer optimization comes back ready to lead us to a new understanding of our craft.

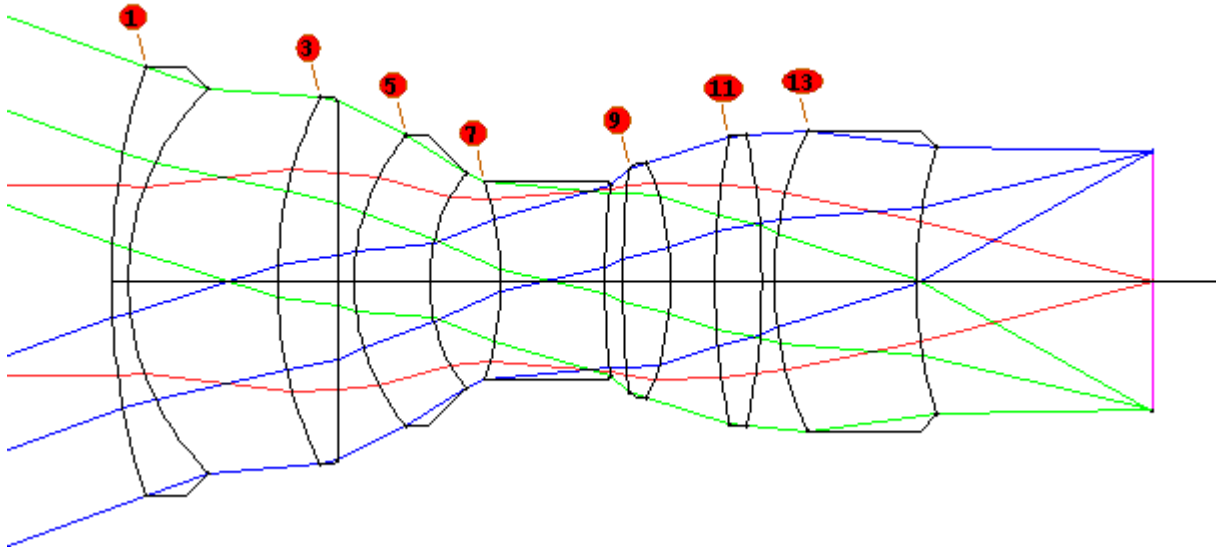


Figure 6. Result of optimizing the plane-parallel starting point, when the stop is initially on the last surface.

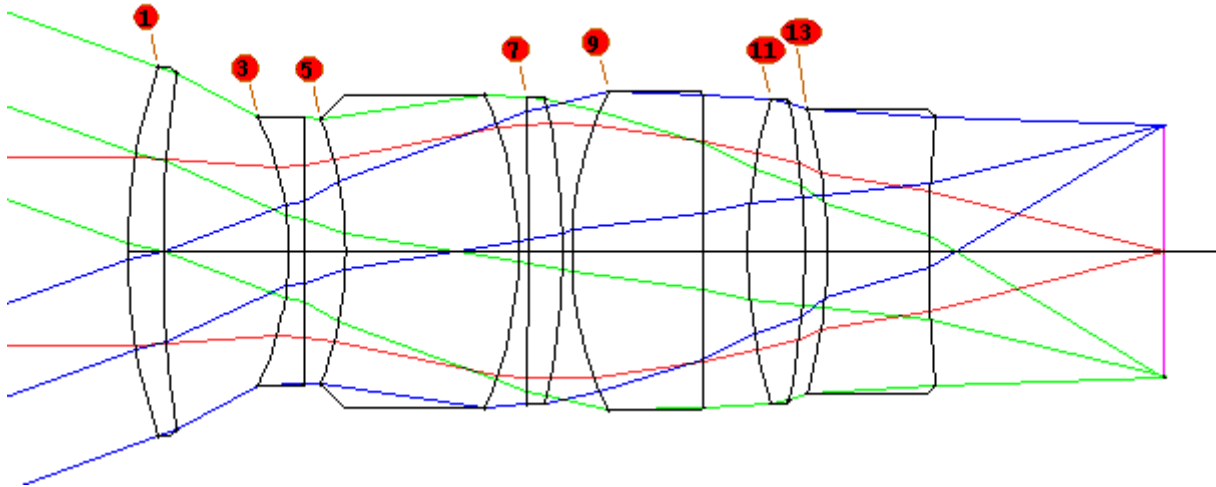


Figure 7. The lens that results from starting with the stop near the center.

Think about what happens, when you start on top of the hill and run a simulated annealing program. It makes small random changes to the design variables and reoptimizes the lens. A given set of changes sends the design off in some direction until it finds a valley somewhere. A different set of changes will probably go off in a different direction, toward a different valley, and if one exercises this procedure a number of times, one will likely discover all of the valleys that surround the starting hill. Changing the problem slightly, as when the stop is initially moved from one point to another, has the effect of moving all of the valleys around, so the optimization process will head toward one or another, taking a different path and ending up at a different optimum in each case.

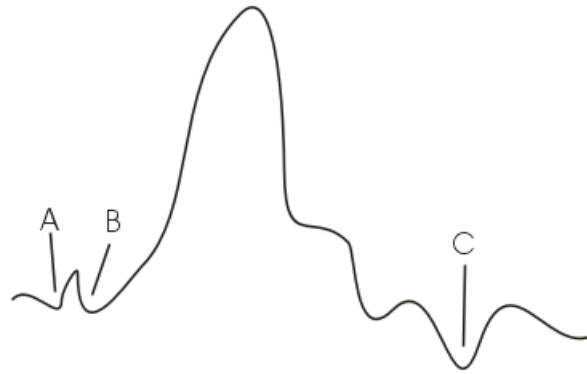


Figure 8. Optimization space geometry, in two dimensions

To illustrate the potential of this process, we ran many cases and discovered a variety of configurations, shown in Figure 9. This probably represents all of the valleys that can be seen from the hilltop. One need only compare the quality of the designs and select a favorite.

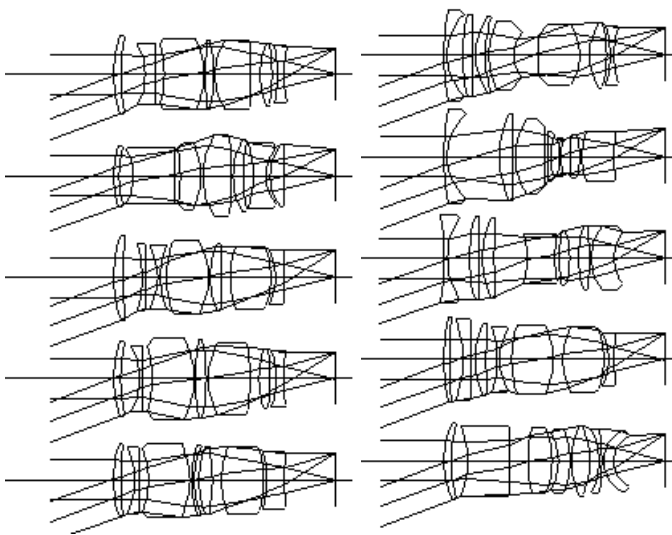


Figure 9. Example of the variety of lenses found in the valleys surrounding a plane-parallel-plate design

5. ELIMINATING STRAIN

When one or more elements contribute large aberrations to the sum, and compensating for those aberrations is difficult or imperfect, one says that the lens is *strained*. This situation arises more often than not when one simply corrects a set of ray intercepts, since most optimization programs have no requirement to avoid it. But another new tool makes this possible.

We define the strain of an element as the sum of the squares of all of the 3rd and 5th-order aberrations of that element, except for secondary color. Here is an example of a lens that is poorly corrected, along with an evaluation of the strain:

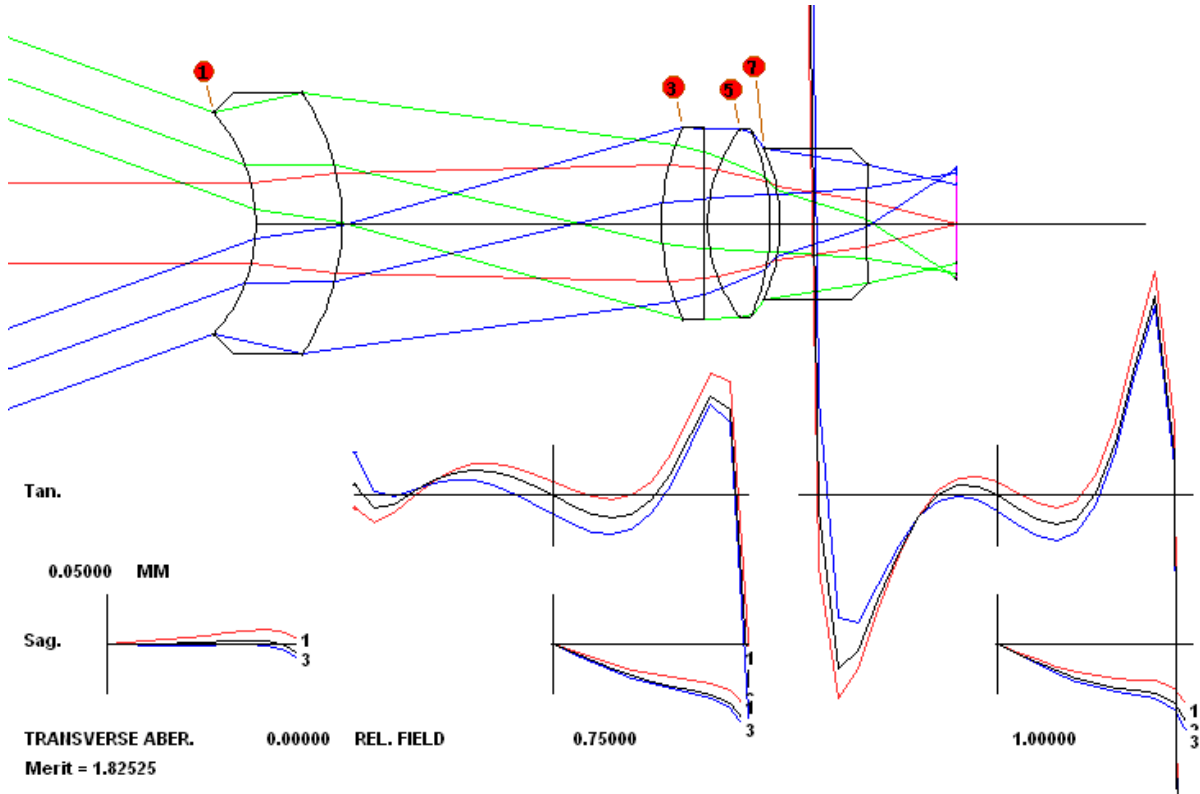


Figure 10. A strained lens

OPTICAL STRAIN ANALYSIS

FORMAT:

SA3	CO3	TI3	SI3	PET	DIS	PAC	PLC
SA5	CO5	TOBSA	SOBSA	ECOMA	TI5	SI5	DI5

ELEMENT NO. 1: SURFACES 1 TO 2; POWER = -0.216347E-02 STRAIN = 0.189175E+01
 0.03521 -0.33569 1.12395 0.42741 0.07914 -0.09339 -0.00305 0.06343
 0.00137 -0.02026 0.11701 0.02668 -0.32655 0.43092 0.07437 -0.01394

ELEMENT NO. 2: SURFACES 3 TO 4; POWER = 0.112445E-01 STRAIN = 0.359384E+01
 -0.20131 -0.61540 -1.59659 -0.65551 -0.18497 -0.05895 0.13721 0.09551
 -0.01768 -0.01500 -0.15428 -0.08199 -0.27335 -0.05073 0.14314 0.03651

ELEMENT NO. 3: SURFACES 5 TO 6; POWER = 0.229770E-01 STRAIN = 0.261783E+02
 -1.90199 -2.57218 -1.58389 -0.80464 -0.41502 -0.05458 0.17760 0.17678
 -0.59989 -1.18520 -2.92694 -0.87751 -1.05790 0.45093 0.35367 0.03666

ELEMENT NO. 4: SURFACES 7 TO 8; POWER = -0.218990E-01 STRAIN = 0.386763E+02
 2.08456 3.54522 1.77769 0.79973 0.31074 0.02899 -0.28987 -0.26753
 0.61104 1.31688 3.32415 0.99267 1.72357 -0.64364 -0.42549 -0.02762

SYNOPTSYS AI>

The poor performance shown by the rayfans plot suggests why optimizing on a set of rays has trouble correcting it: the large error at the left end of the tangential fan tells the program to bring the ray down. But that tends to bring the neighboring rays down as well, including those close to the end that are already *below* the axis. Those, of course, should go up, not down. The problem is analogous to that of subtracting two large numbers that are almost equal. This is never a good idea, especially when, as in lens optimization with finite-difference derivatives, the precision is not perfect. One can put a target in the merit function to reduce the strain, but there is another use for this information.

In the above listing, we see that the last element has the largest strain, so this is the element most likely to benefit if we split it into two elements. Carrying out this step does indeed improve the image, and then repeating the procedure a second time yields the lens in Figure 11.

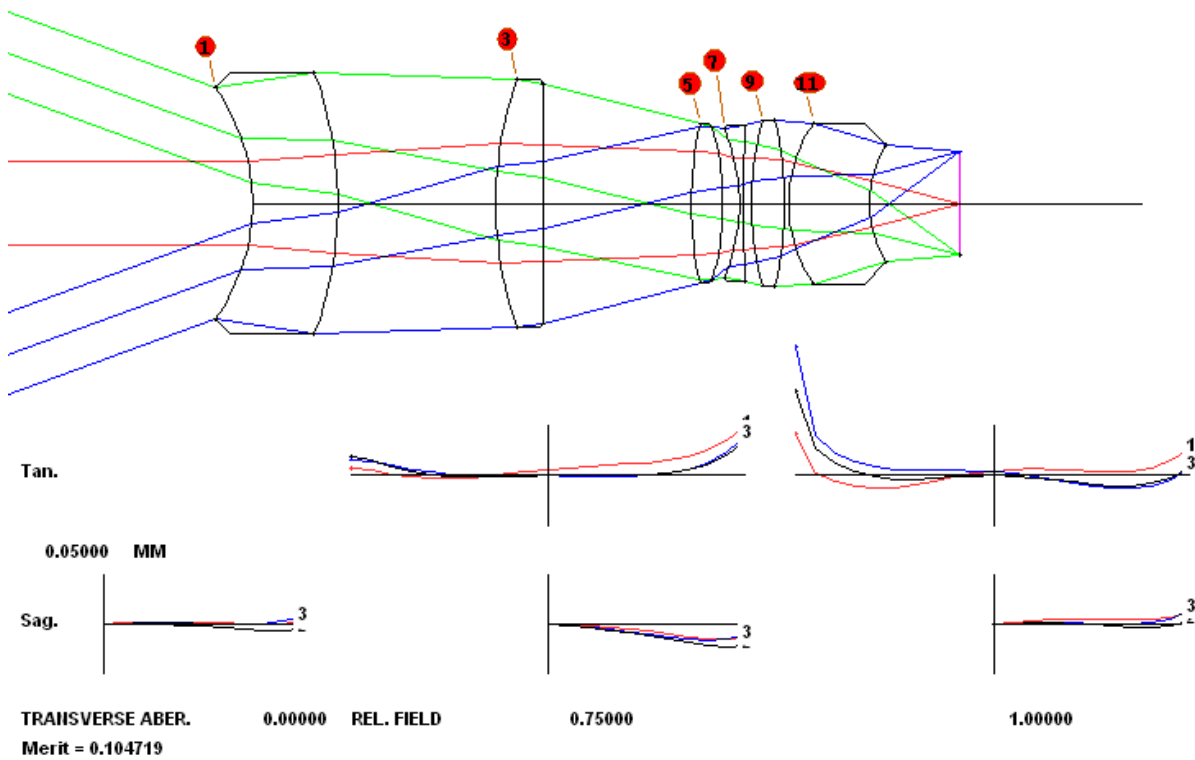


Figure 11. Configuration resulting from the method of reducing strain given in the text.

This is a substantial improvement, and finding this configuration involved little more than evaluating – and reducing – the strain, which went from a value of 38.67 before, to 3.37 (now largest on the first element). This tool can go the other way as well: if an element shows very low strain – and low power – one can very likely remove that element from the lens entirely. This tool, therefore, is a guide that helps you decide how best to alter the construction of your lens.

6. CONCLUSIONS

We have shown in these examples how longstanding difficulties encountered by lens designers can be approached by finding new uses for information that has long been available, but then discarded. The new tools made available with this information can significantly reduce the effort required to design excellent lenses.

¹ H.A. Buchdahl, *Optical Aberration Coefficients*, Dover, 1968.