A zoom lens from scratch: the case for number crunching

D.C. Dilworth

Optical Systems Design, Inc.

ABSTRACT

The art of lens design has long been divided into two camps: theorists and number crunchers. Both camps have vocal supporters, both manage to get the job done, and both occasionally voice disdain for the other. This paper presents the case for number crunching—while recognizing the important contributions that theorists have made and continue to make. We illustrate this case by designing a zoom lens with no starting design and minimum direction from the designer. To this end we use a feature called ZSEARCH™ found in the program SYNOPSYS™[1].

Keywords: lens design, global optimization, zoom lens, Damped Least Squares (DLS), Pseudo-Second Derivative (PSD), SYNOPSYS™.

1. INTRODUCTION

The long and uneven development of our art has of necessity favored whatever tools were available at each period. Galileo polished lenses by trial and error. When Seidel developed his theory of the 3rd-order aberrations in 1857, it became possible to calculate approximately the nature of the optical image. But it was quickly realized that an approximate answer was not good enough: the actual image did not resemble the approximation except in very simple cases. To get a more realistic picture one had to trace rays. By hand.

The effort was staggering. So tedious that some practitioners found it quicker to actually grind and polish a 3rd-order design, measure the image, and then iterate, than it was to trace enough rays to predict what the image would look like. Thus developed two camps. In England, such notables as Charles Hastings and H. Dennis Taylor were firmly in the theorist camp—while recognizing that 3rd-order theory would never suffice on its own without shop measurements of physical lenses. On the continent, Joseph Max Petzval and Ernst Abbe insisted that the computation—meaning ray tracing—must be completed before any fabrication was started. To develop his eponymous lens, Petzval employed a corporal and eight artillery gunners (who were good at arithmetic) for six months of ray tracing.

It is therefore understandable that each camp found fault with the other. The development of logarithms by Napier in the early 17th century showed a way forward—but to use that technology one needed tables of logarithms, which also required immense labor to calculate, to say nothing of the pencil-and-paper calculations afterwards. There was no easy way.

Efforts to develop a higher-order theory produced results that were unwieldy and complex, reaching a climax in the classic text of Arthur Cox[2]. One is there obliged to wade through over 600 pages of dense algebra. I suspect few have read the book. Meanwhile, mechanical calculators made ray-tracing more efficient. Rudolph Kingslake once remarked, “When someone applied for a position in our department at Kodak, I would ask him if he could contemplate pressing the buttons of a desk calculator for the next forty years, and if he said ‘yes’, I would hire him.” There was still no easy way.

2. CONCILIATION

Today there is an easier way: the modern desktop PC and optimization software. Nobody disputes the fact that these tools can accomplish in seconds a job that previously took days or weeks. But some lens designers still cling to old practices, and some in academia still teach those practices. I know an experienced professor who still works up a 3rd-order design before optimizing the lens, and is proud of being able to develop a five-element lens in only five days. (A design that can be done in seconds by a committed number cruncher.) Yes, there is an easier way, but it is not widely acknowledged or utilized. Recently a customer sent me 10 pages of hand calculations for a zoom lens, evidently feeling that such labor was essential before he went to the computer. His effort pointed to the need for a feature like ZSEARCH™.

In a previous paper I showed how a directed search algorithm called DSEARCH™[3] [4] can find excellent designs in short order. I present here a case study of how this newer tool can dramatically shorten the development time for a zoom lens. This is a much harder problem than that addressed by the DSEARCH paper, and a great illustration of the case for number crunching.

3. THE ZOOM LENS PROBLEM

The principle behind both DSEARCH and ZSEARCH is much the same: start at the top of a mountain (where you can see all the valleys in the area), slide downhill in a selected set of directions, and find the lowest valley in each direction. Then sort the results. The top of a hill, in optical terms, corresponds to a lens with all flat surfaces, and the lower the valley, the better the lens.

Since searching for a zoom lens is a more difficult problem than that addressed by DSEARCH, we have to apply some prior knowledge right at the start. To that end we will specify the basic lens configuration—but let the program fill in the details. For this sample problem we require the following:

1. Zoom ratio 8X.
2. Speed F/3.5.
3. Semi-field angle 14 degrees at the wide-angle setting
4. Gaussian image height 5 mm at all zooms.
5. Back focus distance approximately 20 mm.

To describe this problem to ZSEARCH, we first declare these system requirements and design goals:

1. 10 zoom positions
2. Four groups of lenses with three elements in each group
3. The middle two groups will zoom.
4. The stop is at the first surface of group 4.
5. Lens apertures not to exceed 30 mm in radius.
6. No requirement on total length. (We can address this if the need arises.)

You will note that there is some optics theory embodied in the above (even number crunchers do not hesitate to use theory when appropriate). The first group is typically used for range focusing, the central two make up the variator and compensator, and the last group focuses the light on the sensor, the F/number automatically staying constant over the zoom range since the stop is at that group. Theory leads us to these goals; number crunching will do the rest.

3.1 Tradeoffs

How do we know how many elements to include in each group? We don't. This a guess, based on experience. But we have a choice: fewer lenses in each group (which will run faster in ZSEARCH) or more (which will give a better image). Each additional lens doubles the running time of ZSEARCH, so it's best to start with a fairly simple construction and then enhance it if necessary with some of our other tools: the features we call AEI™ (Automatic Element Insertion) and AED™ (Automatic Element Deletion), which can alter the construction of the lens by changing the number of elements in each group after ZSEARCH has finished. Those features are described in more detail below.

The plan is to quickly develop a rather simple design, then run AEI, perhaps more than once, to develop a more complex lens from that simple starting point.

We configure ZSEARCH to run in two phases; the first is a very quick appraisal of the lens in each search direction, where the merit function consists of 3rd and 5th-order aberrations plus a very few real rays. The best of those are then subject to another optimization cycle that corrects a larger set of rays. Monitors are available to keep the maximum lens apertures below a target value, avoid rays near the critical angle, and so on. Those help the program to find practical solutions. Lenses are optimized and annealed at each stage.

The quick mode is important. Since the program has to evaluate a large number of potential designs (4096 in this case), we want to proceed as quickly as possible—and this depends on the speed of the optimization method. For this purpose, we use the PSD III algorithm[5]. The impact is significant. We recently created a test problem to be run on as many lens design programs as possible so we could compare the speed. That problem called for the design of a seven-element lens, starting with all flat surfaces, all thicknesses and airspaces equal, and all glass types in the middle of the glass chart. Average image quality over the field was to be near six microns RMS. A very simple problem, to be sure. When run on two other programs, one of them required almost two hours, the other 40 minutes. With the PSD III algorithm, the goal was met in 0.84 seconds. With that kind of speed, the search tools become very practical.

3.2 Initial Results

This job runs on ZSEARCH in about 16 minutes on our 3 GHz PC, and the best lens it found is shown in Fig. 1. ZSEARCH also shows a picture of the 10 best that were found, from which we can select the one we want; part of this is in Fig. 2.

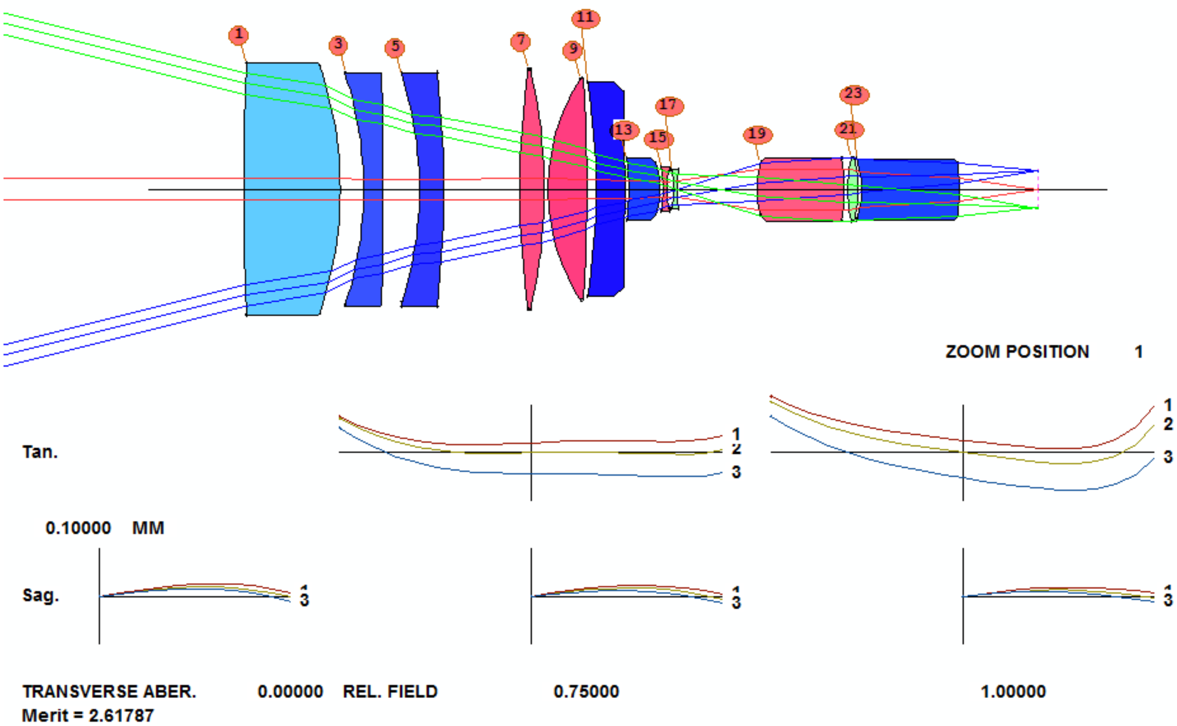


Fig. 1. Lens form returned by ZSEARCH

This lens is not perfect, but it is a good starting point, considering that we gave it nothing but a list of goals and constraints. The program also creates an optimization file, and of course we run it to see if the lens can be improved in its present form. That brings the merit function down, and we anneal it until it gets as low as it can.

We notice that the aperture monitor is the largest single aberration, which means that the lens would like it to be larger, so we increase it to 35 mm. After more optimization and annealing, the lens has the form in Fig. 3. The scale of the rayfans curves has changed to reflect the improved image.

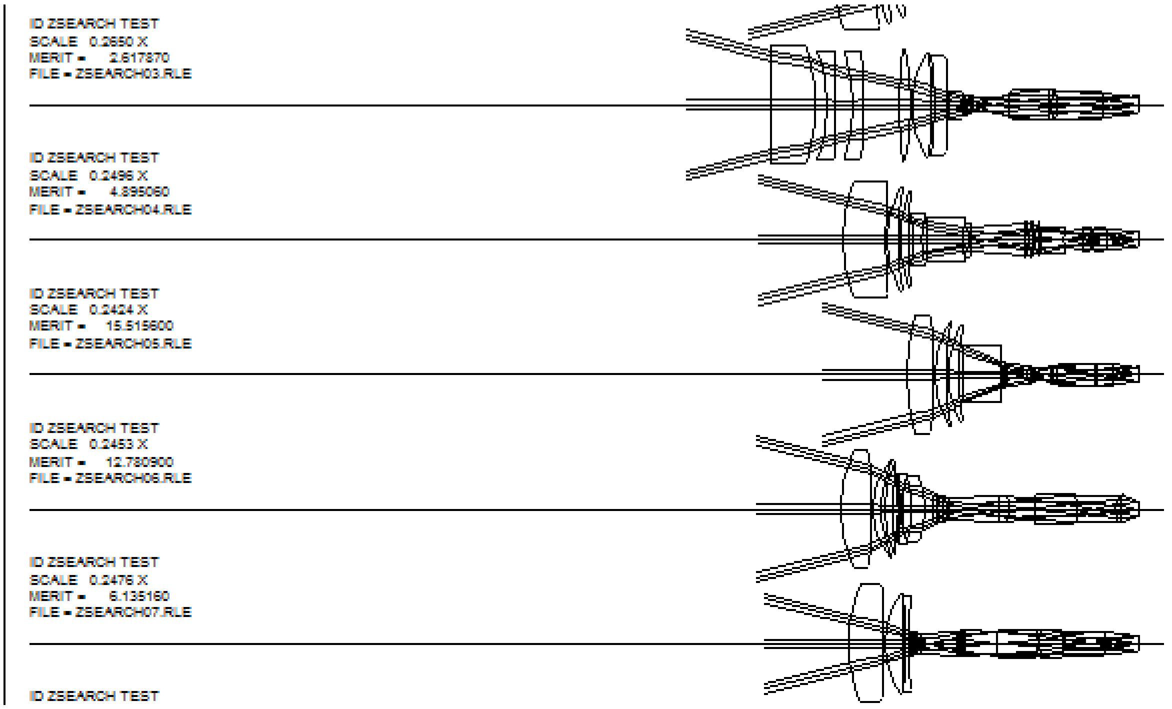


Fig. 2. A portion of the 10 design forms found

Since this lens is still not good enough, it's time to exercise some of our other number-crunching tools. The Automatic Element Insertion (AEI) routine is based on Florian Bociort's[6] saddle-point theory: if one inserts a thin shell with zero power and thickness adjacent to a lens surface, the ray paths do not change—but six additional degrees of freedom are then available. The shell usually changes into a lens element when optimized, and usually the merit function is reduced as well. AEI tries every surface within a specified range and selects the location that yields the lowest value. We also note that the beam does not fill the paraxial stop at 19 very well, so we change this to a real stop, and we increase the minimum edge thickness monitor target since some edges seemed rather thin.

Then we run AEI twice, and it returns the lens in Fig. 4.

The inverse of AEI is the Automatic Element Deletion program AED, which implements the opposite of the saddle-point method: it tries reducing each element to a thin shell of zero power, and if it succeeds, that shell can be removed.

So we run AED to find if any element can be removed with little loss of quality, and then run AEI to find the best place to insert a new one. The lens is improved, and we repeat both these steps, producing the lens in Fig. 5.

This exercise shows how number crunching tools can establish, and modify, the basic lens construction more quickly and rigorously than can a theorist using older technology. Even if the starting lens is poorly constructed, with these tools the design can evolve into a much better one.

Since this is a perfectly acceptable lens, it's time to change the glass models to real glasses. It's number crunching time again. The Automatic Real Glass (ARGLASS™) feature will substitute eligible glass types from a selected catalog for the model on each element and reoptimize, varying the remaining models. When this process finishes, it produces the lens shown in Fig. 6.

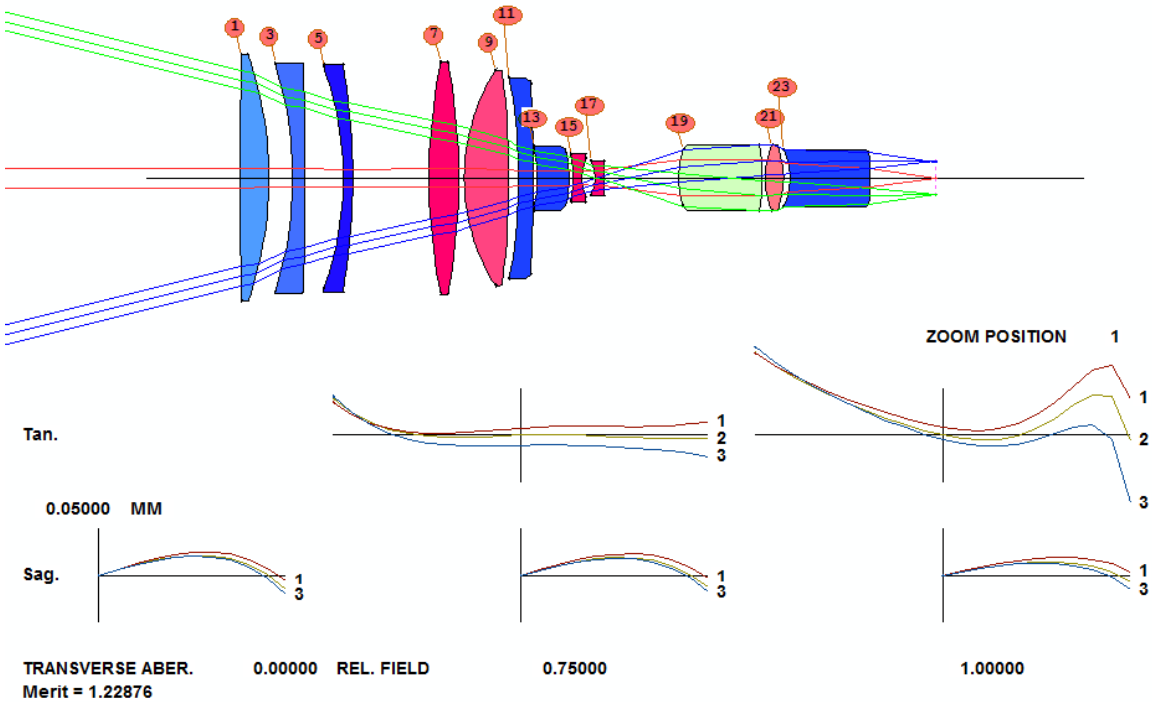


Fig. 3. Zoom lens optimized as far as possible with the starting configuration

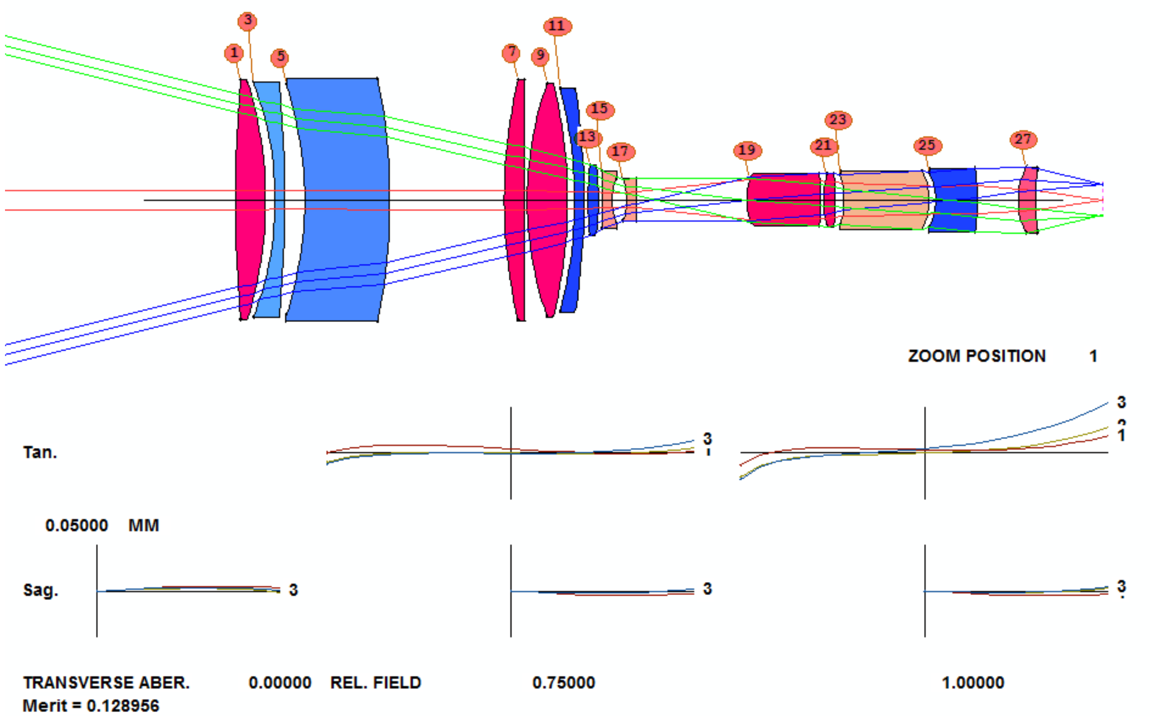


Fig. 4. Lens form after modification by AEI. Two new elements have been added to group 4.

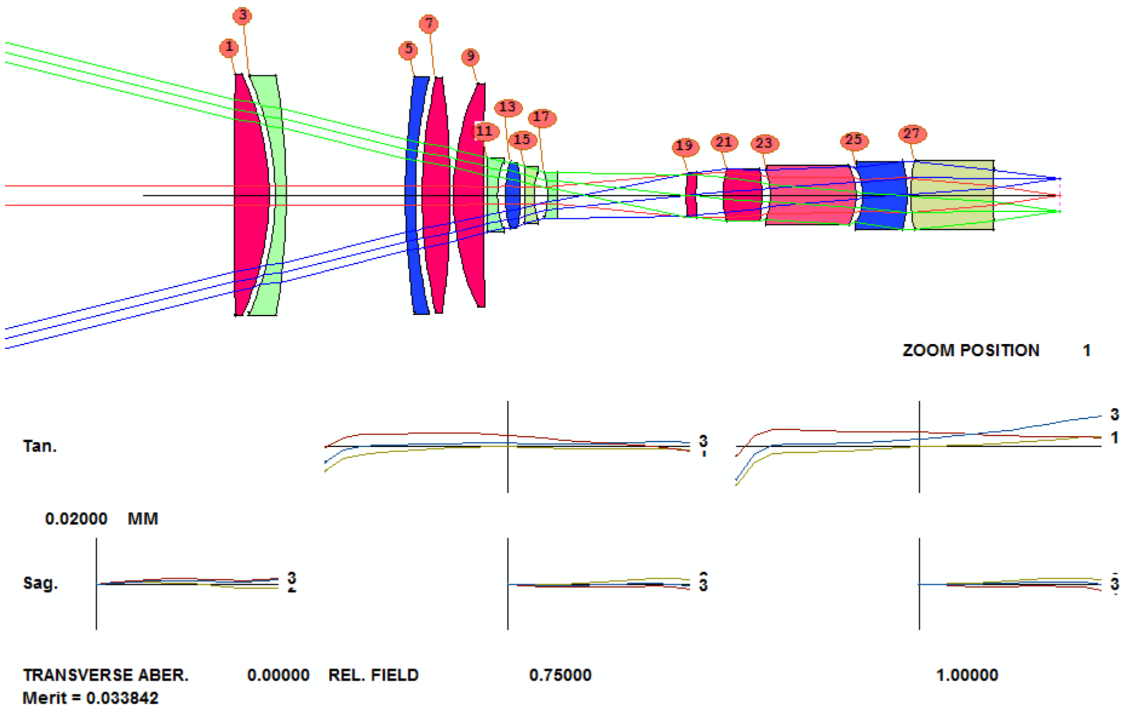


Fig. 5. Lens after twice deleting and adding elements via AED and AEI

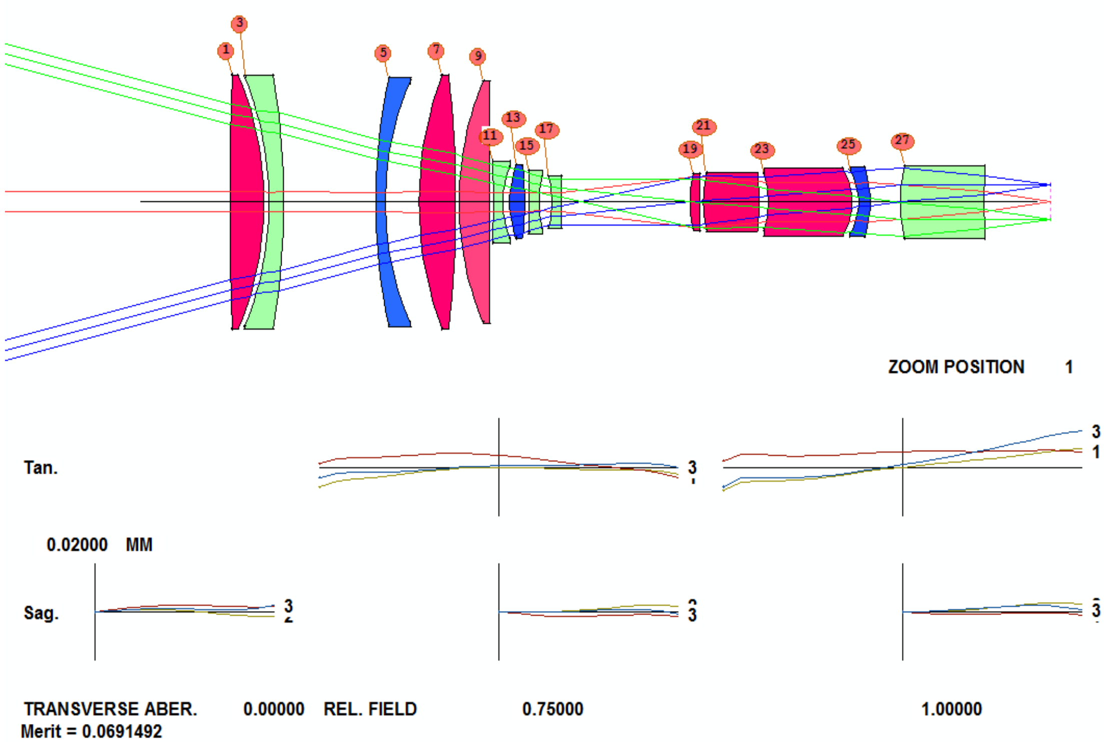


Fig. 6. Lens after replacing model glass with real glass via ARGLASS

We are almost done. To make a lens like this, one needs to calculate a cam curve. That is tricky; one must interpolate between the 10 defined zoom positions and hope the residuals at in-between points will be under control. Another number crunching tool now proves useful: the cam calculation. The software expresses the change in every zooming variable with a power series of a given number of terms, the exponent of each term given by a multiple of some starting number. It can also employ a piecewise-cubic fit. In this example we will use the latter.

In either case, it helps to correct over as many zooms as possible, so we increase to 20 zooms and reoptimize.

Our cam calculation is complete. We can now run the Zoom Slider™, which zooms continuously over the range, displaying the image quality in any of several formats, and find that the image is nicely controlled at every point. Our preliminary lens is done. Fig. 7 shows the lens at all 10 zooms, and Fig. 8 shows the cam curves corresponding to the final polynomial.

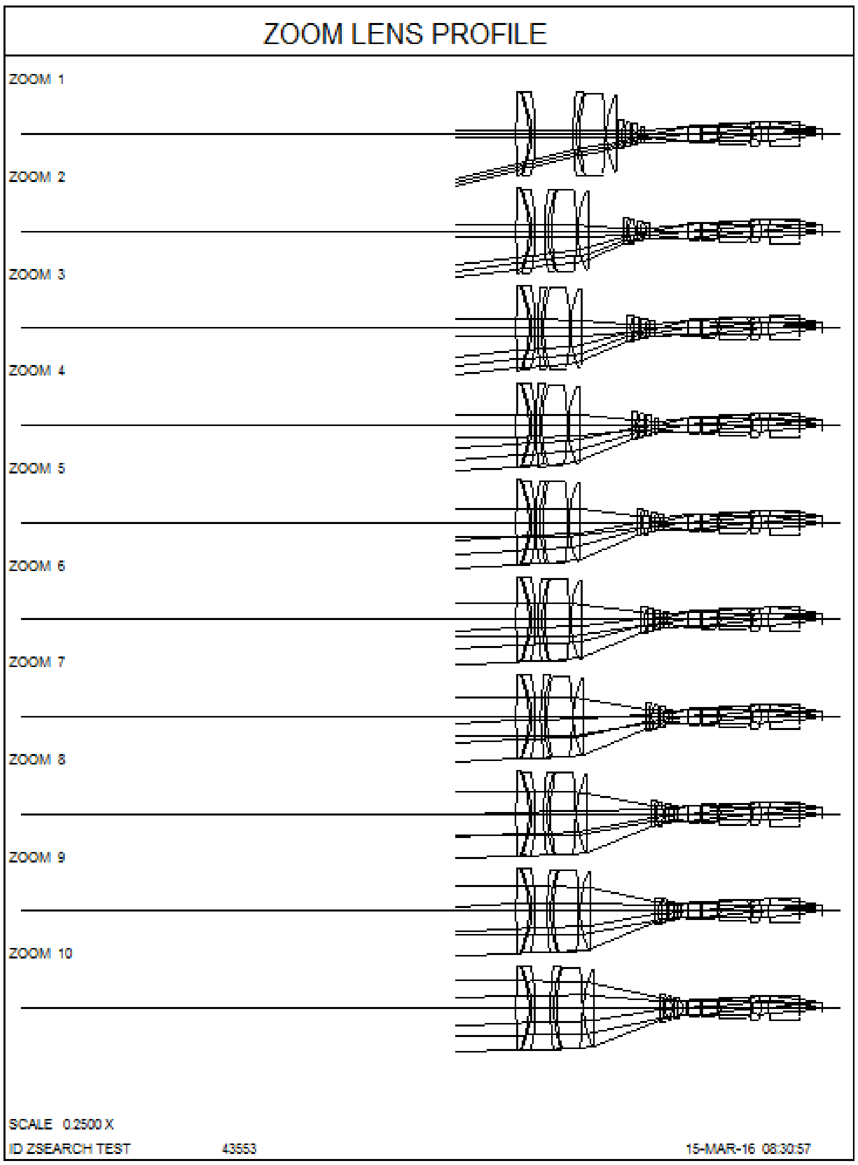


Fig. 7. The final lens at 10 zoom positions

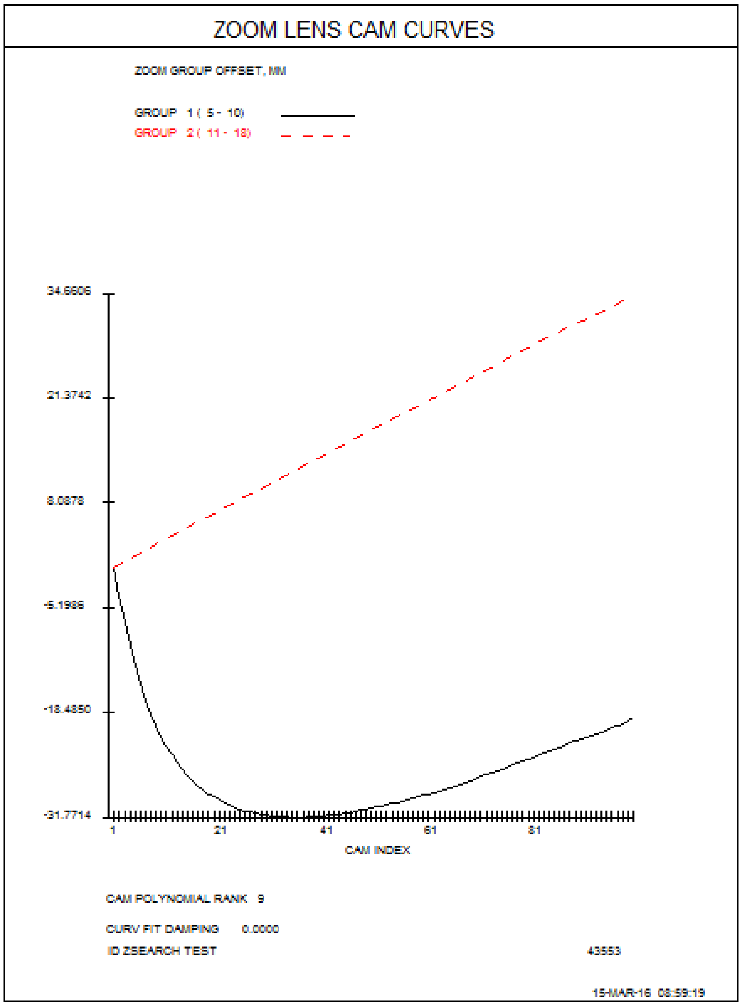
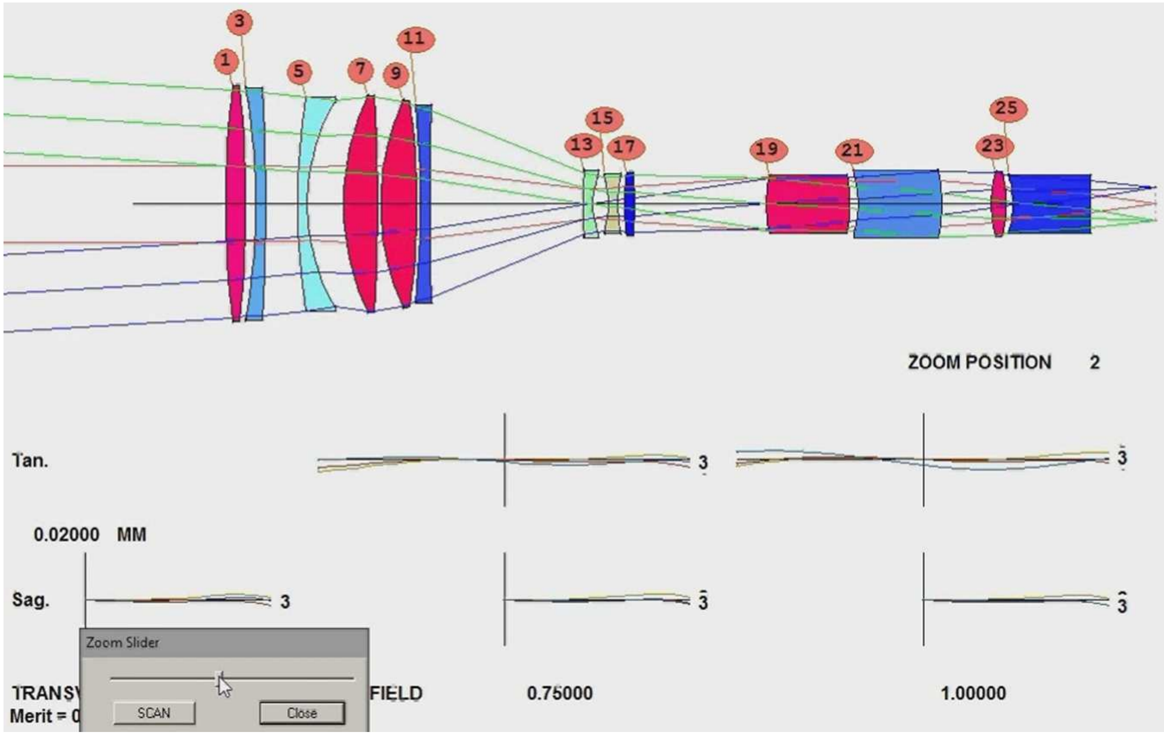


Fig. 8. Cam curves for the final lens

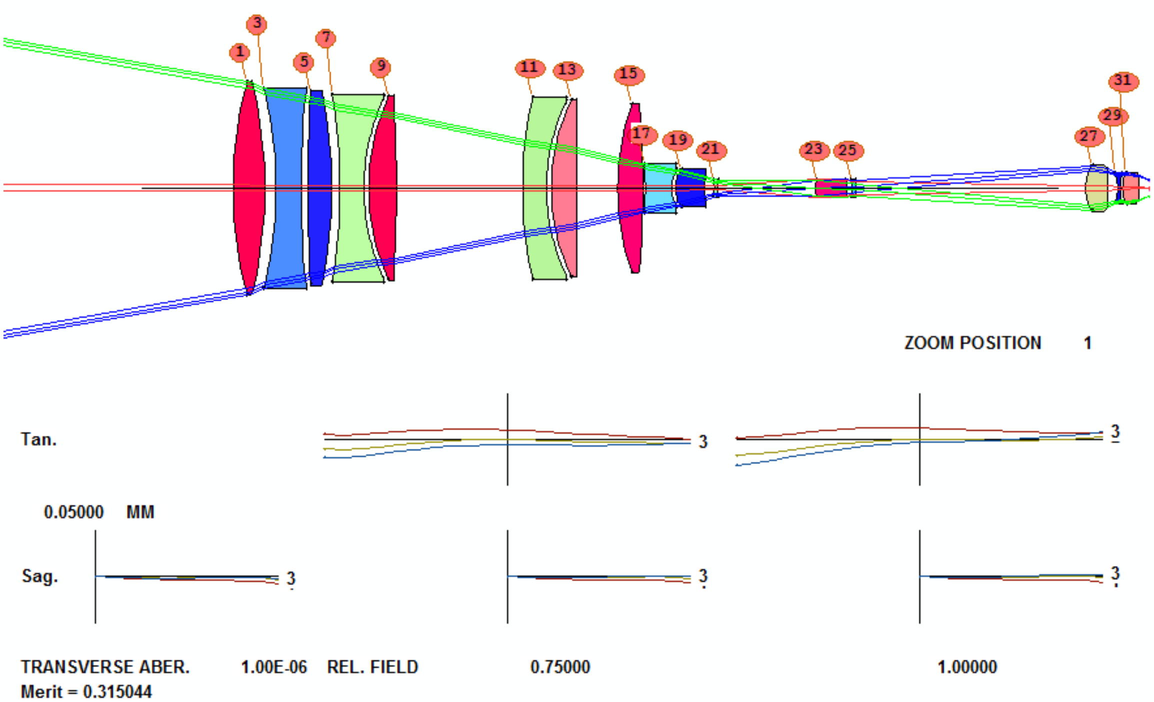
This exercise has shown the power of number crunching in solving what would be a very lengthy task using the tools and methods that predate these powerful features. Of course we would next investigate adapting the first group for range focusing and adjusting lens thicknesses with an eye to manufacturability—but the lesson is clear: theorists will never produce a lens of this complexity so quickly. The whole job required about two hours with these tools, and that is enough for this paper.

If one changes the input for ZSEARCH, the program often finds different valleys. We ran the job described above again, this time asking for more cycles of optimization in the quick phase. The run took longer but produced a lens that required fewer alterations and yielded performance as good as shown above with 13 elements instead of 14. Video 1 shows this lens as the Zoom Slider™ zooms it smoothly. With these tools one can rapidly execute tradeoff studies.

Out of curiosity, we asked ZSEARCH for a 25x zoom lens operating at F/8, using the same procedures described above. The results were encouraging, as shown in Fig. 9 – although the ends of the zoom range are not as good as the center. It appears that this is just beyond the limit of what can be done with a single lens with two zooming groups. To go further, one would likely use a front end that zooms over, say 8X, followed by a zooming relay with perhaps 5X, yielding a 40X lens. We also tried a 90X zoom with 15 elements and three zooming groups, with similar results; the image did not hold up at the extreme end of the zoom, although it was quite good elsewhere. These experiments show what kind of performance one can expect with this new tool.



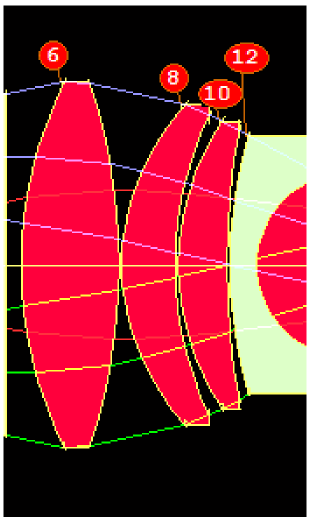
Video 1. A simpler 8X zoom lens designed by ZSEARCH™ zooming via the Zoom Slider™. http://dx.doi.org/doi1



Fig, 9. A 25x zoom lens produced by ZSEARCH. This lens required 16 elements and is a good example of the power of pure number crunching.

4. THE POWER OF NUMBER CRUNCHING

To further illustrate the power of number crunching, I show a lens portion in Fig. 10. This is part of a 13-element lens with poor color correction that a colleague sent me. The first three elements are large positive lenses of crown glass, as shown in panel a. After some number crunching using the SYNOPSYS glass-model variables, the system changed as shown in panel b. Remarkably, the second element had become a large positive *flint* lens, and both primary and secondary color were corrected within the diffraction limit. My colleague was astonished. I suspect that no theorist would have come up with that solution. His deep knowledge of aberration theory had actually *prevented* him from finding it. This is an excellent example of the case for number crunching: it can find solutions that a theorist cannot.

a. 

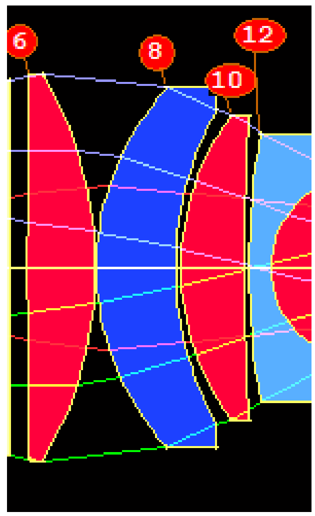
b. 

Fig. 10. Panel a: Portion of a lens designed by a theorist with poor color correction. The first three elements are positive crown lenses, shown in red.

Panel b: The same lens after number crunching with the SYNOPSYS glass model variables. The second element has become a positive *flint* element, shown in blue, and all colors are now in the diffraction limit.

Another exercise is worth reporting. A colleague who is firmly in the Kingslake camp works up a third-order design by hand before resorting to optimization, and I suggested a friendly competition: the design of a 90-degree eyepiece. I set the bar very high—to make the test interesting—and required diffraction-limited imaging over the entire field, small pupil aberrations (to avoid the kidney-bean effect), a sharp tangential image at the edge of the field stop (so it will be sharp to the eye), and minimal distortion, all done with no more than 10 elements. I then asked DSEARCH to come up with a lens to those specifications, requesting only eight elements at first, the result of which is shown in Fig. 11.

This design is already close to perfect, but needed some work. First I added a dummy surface at the intermediate image location to provide a place at which to correct the tangential blur. Then I ran AEI, which inserted an element, and the lens was better. After some more optimizing and then running ARGLASS, the lens changed as shown in Fig. 12. The image consists of a nearly perfect Airy diffraction disk all across the field, and all other goals were met.

The whole process, from start to finish, can be done in only 15 minutes. What did my 3rd-order friend do? After about 100 hours of work, the eyepiece still did not meet specifications and the effort was abandoned. One cannot imagine a better illustration of the case for number crunching. I believe these results point to the future of lens design, a future the old masters would scarcely recognize.

Where does all this leave us? While aberration theory is still worth understanding, today we have powerful tools that generate excellent results without requiring the user to apply much of that theory—which is discomforting to some who have devoted years to mastering and teaching it. Indeed, Kingslake once remarked, "We are losing the ability to design a lens through an effort of the intellect".

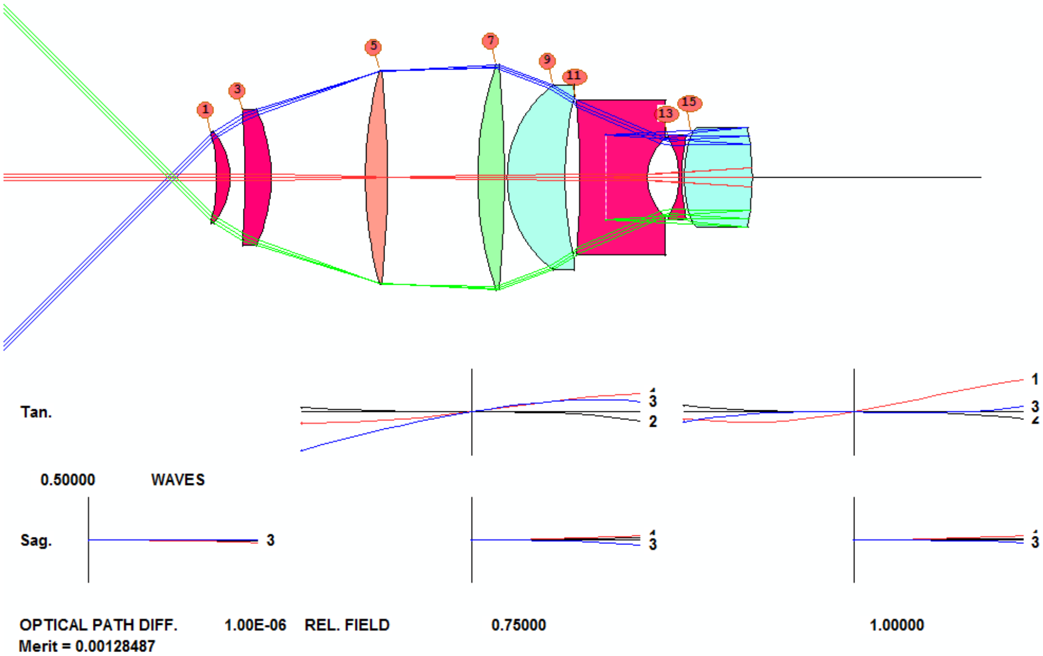


Fig. 11. The best 8-element eyepiece lens returned by DSEARCH.

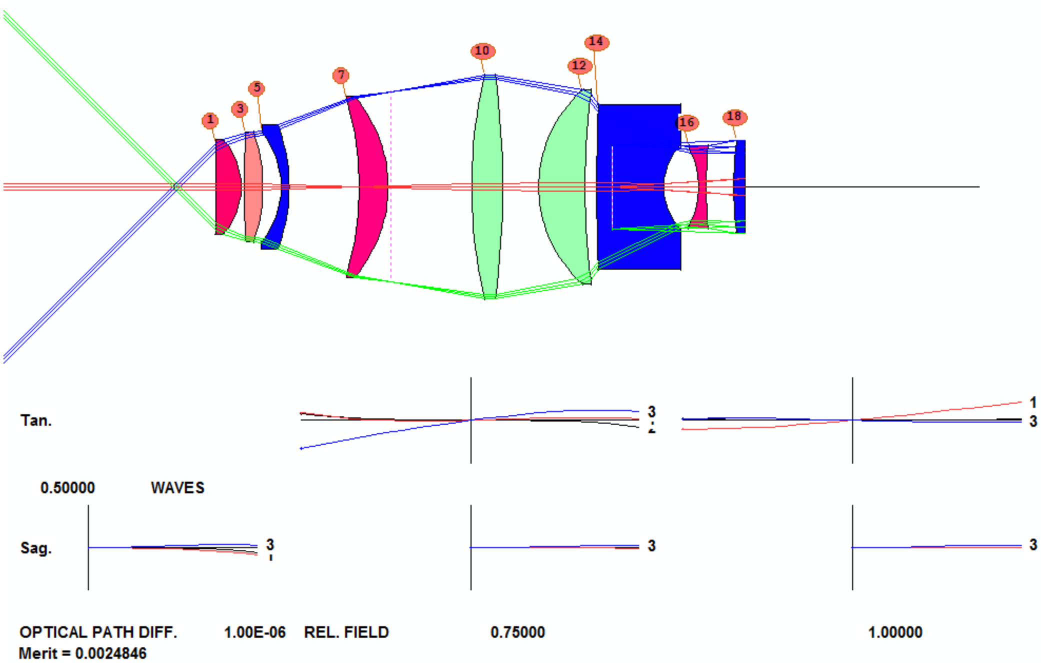


Fig. 12. The eyepiece returned by several number-crunching tools. This is essentially a perfect design, created with no starting point.

That is very true. But we have also lost the ability to make flint arrowheads and ride horses—but who cares? We don't have to do those things anymore.

We are left with the problem of what to teach the newer generations of lens designers. Do they need to master Kingslake's method? To work out a 3rd-order solution by hand? As one teacher remarked to me, "We cannot just teach the students to push a button!" True—but nobody is suggesting we do that. Indeed, we must ask what a beginner needs to know in order to understand what pushing that button does; to understand what lens design is all about and how to use those tools. A great deal of optics, to be sure. That is what should be taught.

We need to deeply consider what knowledge is relevant today, what is a relic of the past, and decide on a curriculum for the future.

---

[1] SYNOPSYS™ is a trademark of Optical Systems Design, Inc.

[2] Arthur Cox, *A System of Optical Design*, Focal Press, London (1964).

[3] Dilworth, D. C., "Novel global optimization algorithms: binary construction and the saddle-point method", Proc. SPIE 8486, 84860A (2012).

[4] Dilworth, D. C. and Shafer, D., “Man versus Machine; a Lens Design Challenge”, SPIE 8841, 88410G-1, (2013).

[5] Dilworth, D. C., “Automatic Lens Optimization: Recent Improvements”, SPIE 554, International Lens Design Conference (1986).

[6] F. Bociort and M. van Turnhout, “Saddle points reveal essential properties of the merit-function landscape”, SPIE Newsroom (24 November 2008)