Lesson 15: Real-World Development of a Lens

In Lesson 14 we designed a 7-element lens starting with nothing but plane-parallel surfaces and had the program fit the design to catalog glass types automatically with the ARGLASS feature. But suppose you have a real application and want to develop it further. This lesson covers some additional procedures that would then be appropriate. To make it a real "real world" lesson, we will show how a designer will follow various clues in order to arrive at a solution, and how not all clues lead to success. That is important too: it is instructive to see how sometimes one wanders into blind alleys. As you develop your skills as a lens designer, you will encounter many of them, and should not be discouraged since it happens to us all. With perseverance, a successful design can usually be found.

We will do this lesson in two ways; first with DSEARCH with the help of a number of other tools. Then, in Lesson 17 we show another approach that is actually quicker and easier. You should know about all of the tools used in both approaches.

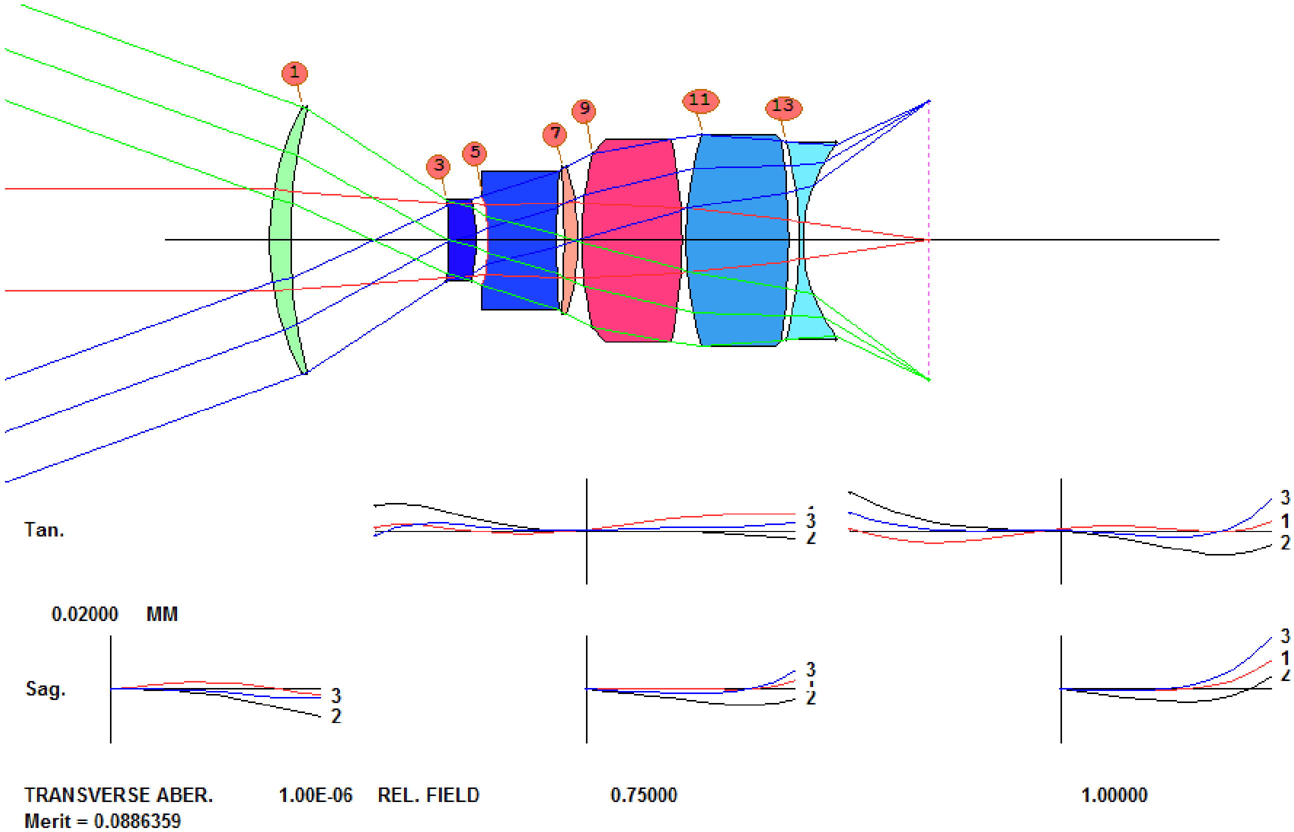
We will first use DSEARCH to find a good starting point. Here is the input:

```
CORE 16
 DSEARCH 6  QUIET
 SYSTEM
 ID DSEARCH SAMPLE
 OBB 0 20 12.7
 WAVL 0.6563 0.5876 0.4861

 UNITS MM
 END
 GOALS
 ELEMENTS 7
 FNUM 3.575
 BACK 50 .01
 STOP MIDDLE
 STOP FREE
 RT 0.5
 FOV 0.0 0.75 1.0 0.0 0.0
 FWT 5.0 3.0 3
 DELAY 999
 RSTART 900
 THSTART 7
 ASTART 15
 NPASS 66
 ANNEAL 200 20 Q
 COLORS 3
 SNAPSHOT 10
 QUICK 44 66
 END
 SPECIAL PANT

 END
 SPECIAL AANT
 LUL 150 1 1 A TOTL
 END
 GO
```

We run this, and the best lens that is returned is quite good. We optimize and anneal, using the file DSEARCH_OPT, which is in a new editor window.

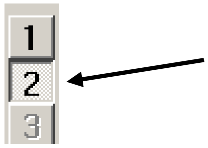


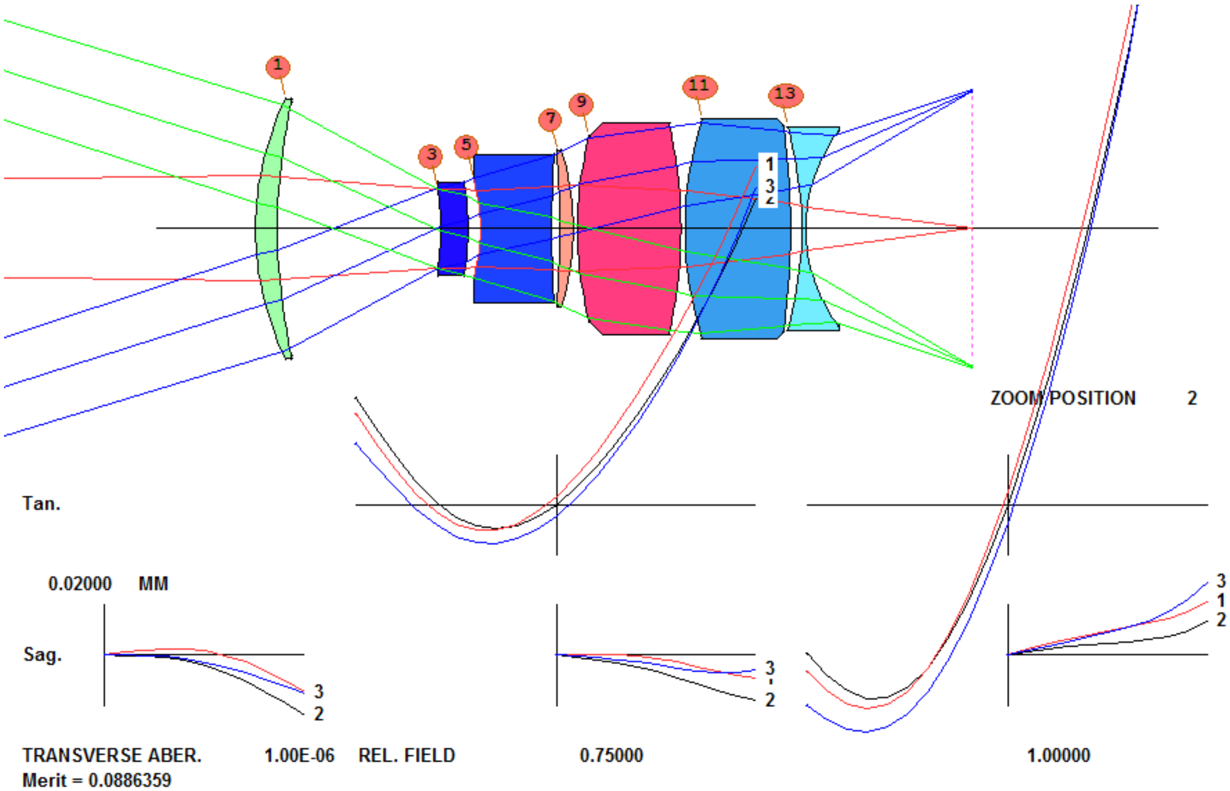
Suppose we want the lens to work over a range of object distances from one meter to infinity. There are two ways to implement that requirement: with multiconfigurations, which is very flexible but complicated, or by declaring this a zoom lens in which the object distance zooms. The second approach is better here, since it is simpler, does what we want, and we can examine intermediate object distances very easily. We have to set up this lens as a ZFILE zoom lens.

```
CHG
APS 3                ! declare surface 3 the stop
15 CAO 32            ! fix the CAO on the image (so FFIELD works)
FFIELD               ! adjust the object height so the image fills the CAO there
14 YMT               ! assign a paraxial focus solve to surface 14
ZFILE 1              ! start of the ZFILE section
14 14                ! there is one zooming group, the last thickness
ZOOM 2               ! ZOOM 1 is default; ZOOM 2 gets OBA object on the next line
OBA 1000 -366.554 12.7  ! the object description at this zoom
END                  ! end of changes
```

Here we declare surface 3 the stop, so all zooms use the same location, set a hard aperture at the image so the FFIELD directive has a target, put a thickness solve on 14 so all zooms refocus automatically, and declare a single zooming group, surface 14. Then we define the object distance for ZOOM 2 at 1000 mm distance, with a negative YPP0 because the value in ZOOM 1 is also negative, and they have to have the same sign.

Run this MACro, and the lens changes to a zoom lens, with only a single airspace zooming in this case. Now you see a new toolbar on the right side of the monitor. What does the image look like in ZOOM 2? If you click on buttons 1 and 2 you see the lens at that zoom setting. Here is zoom 2:





Pretty awful! We have to correct the image at both conjugates. Here is our MACro:

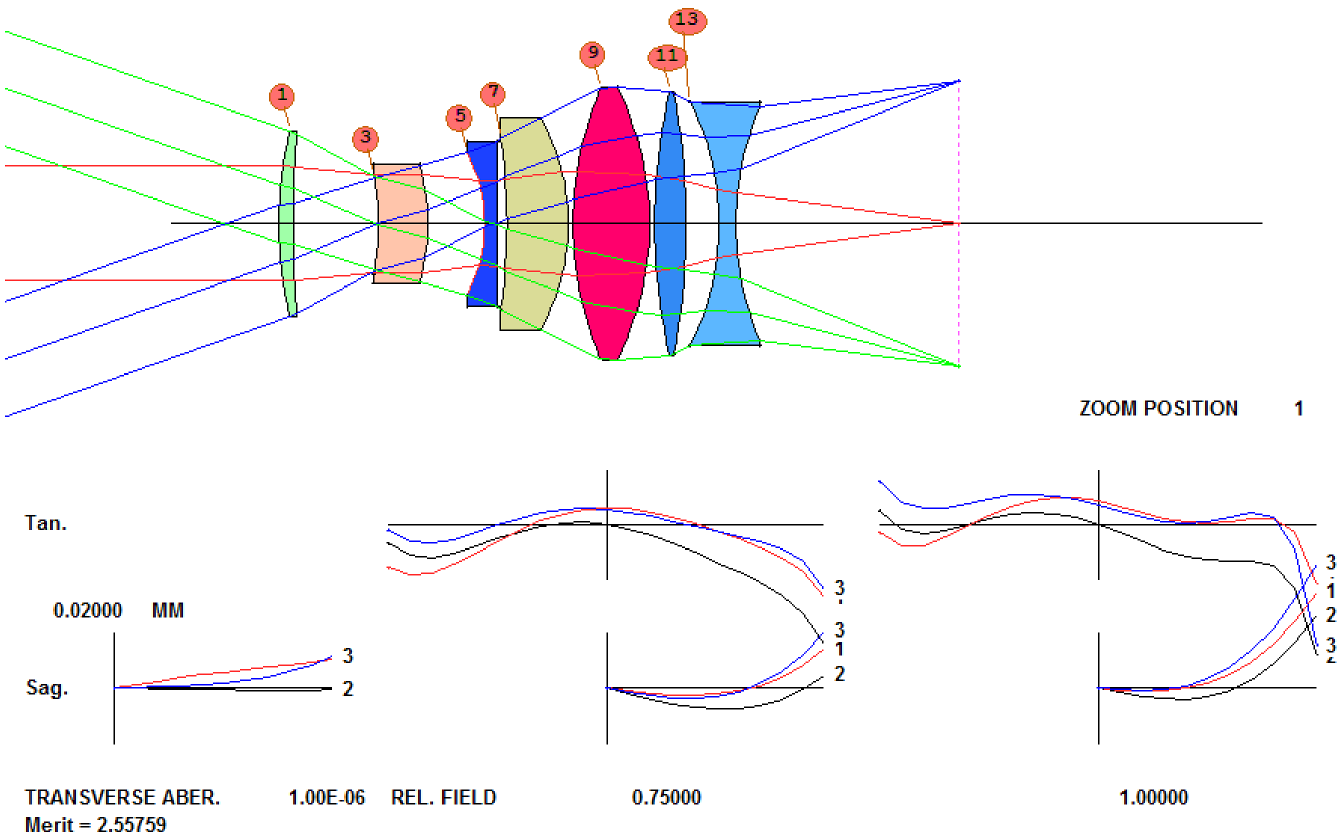
```
AWT: 0.5
PANT          ! Define variables.
CUL 1.9       ! Set upper limit of 1.9 on index variables.
FUL 1.9
!VY 1 YP1     ! Don’t vary YP1; it is not compatible with the real pupil declaration
VLIST RAD ALL       ! Varies all radii that are not flat.
VLIST TH ALL        ! varies all thicknesses and airspaces except for the
! back focus, thickness 14, which has a solve in effect
VLIST GLM ALL
END

AANT                ! Start of merit function definition.
AEC                 ! Activate automatic edge-feathering monitor
ACC                 ! and maximum center thickness monitor.
ADT 6 .1 10         ! Keep diameter/thickness ratio 6 or more
!M 33 2 A GIHT      ! Comment this out, since the FFIELD will control scale
LUL 150 1 1 A TOTL
M 50 .1 A BACK      ! Since the back focus will vary, keep it reasonable
M 90.61 1 A FOCL    ! Add this requirement so the focal length doesn’t change
GSR AWT 10 5 M 0    ! Note how weights are assigned to the several field points,
! and the symbol AWT controls the aperture weighting.
GNR AWT 5.5 4 M .5        ! This creates a ray grid at the ½ field point
GNR AWT 5.5 4 M .7        ! These for the 0.7 field point
GNR AWT 3 4 M 1     ! Full field gets the lowest weight.
```

```
ZOOM 2              ! Targets for zoom 2 (with the object at one meter)
GSR AWT 10 5 M 0    ! Note how weights are assigned to field points.
GNR AWT 5.5 4 M .5        ! This creates a ray grid at the ½ field point
GNR AWT 5.5 4 M .7        ! These for the 0.7 field point
GNR AWT 3 4 M 1           ! Full field gets the lowest weight.
END

SNAP
SYNO 50
```

Run this and anneal, and the lens is better but still not very good, with about equal and opposite errors at both ends of the zoom range.

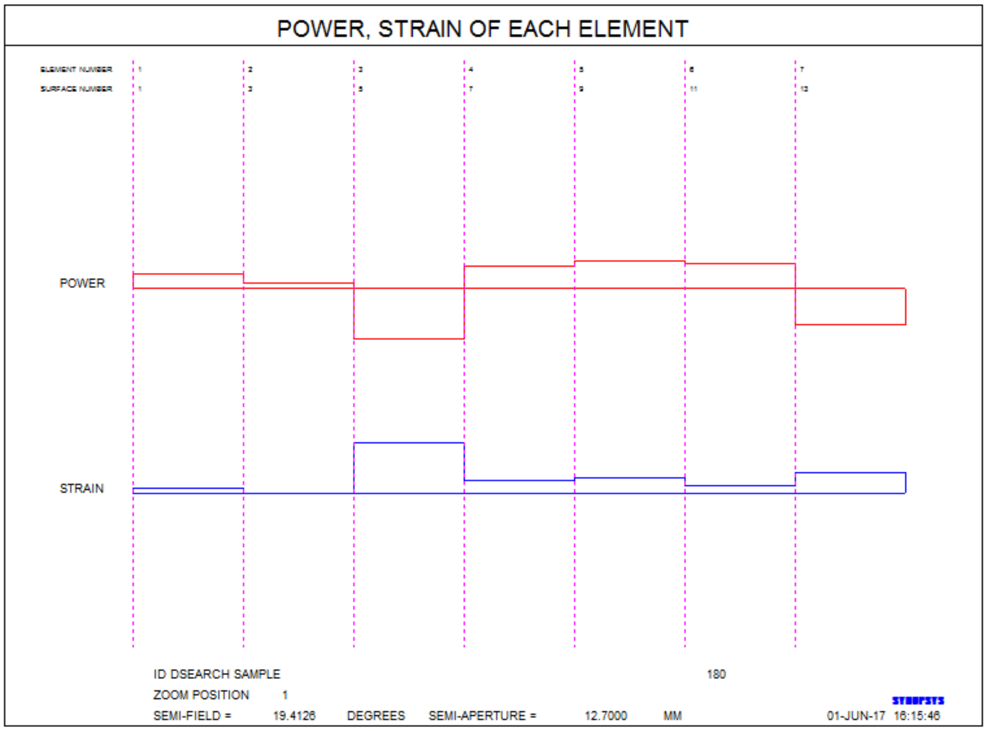


Some subtleties deserve mention: the **GLM ALL** variable will vary all glass models currently in the lens, which means all elements, since DSEARCH uses the glass model unless told otherwise. We have to control the focal length since the object height will be continuously adjusted so the image CAO is filled at full field.

This is better than zoom 2 was before, but there is still a loss of resolution. What to do? We need more variables. What should we add?

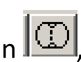
A classic tool for cases like this is the STRAIN calculation. The idea is that the surfaces with the largest strain are contributing most of the low-order aberrations, and splitting an element there might relived that strain.

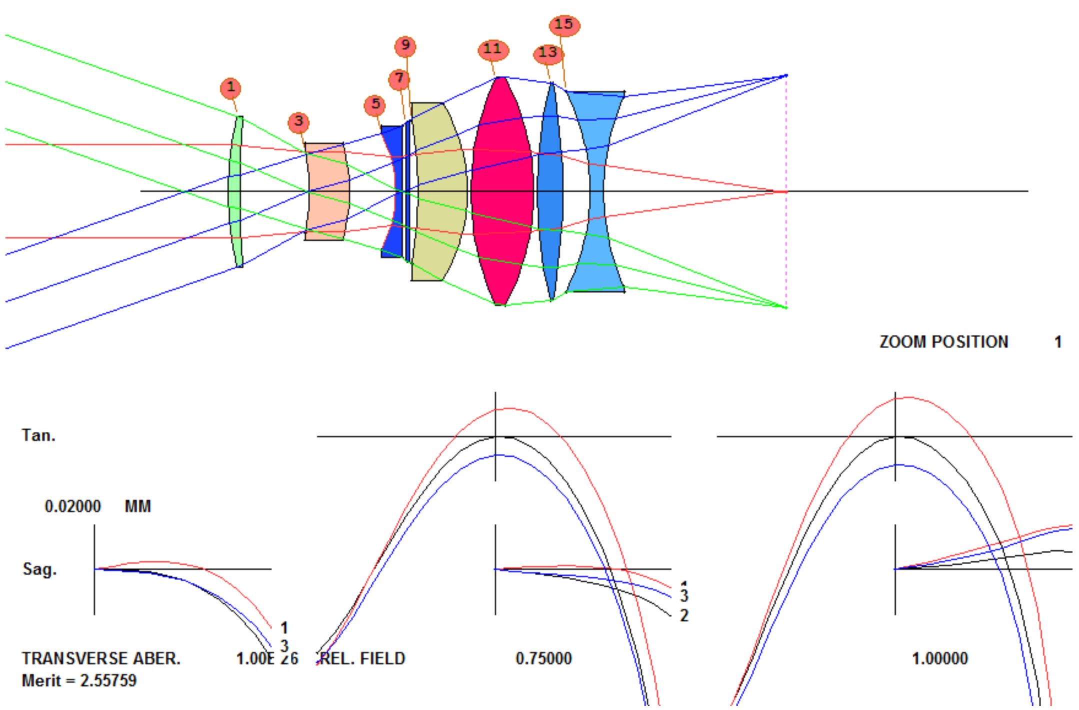
Type **STRAIN P** in the CW.



Indeed, element three has the largest strain. Now we can do one of two things: We can split that element and reoptimize, or we can use a different tool that can figure out the best place to add an element. We will try it both ways. First, let's save this version, so we can go back if things don't work out. Type

STORE 1.

Then go to the WorkSheet (type WS, or click on the button . Then click the button , which lets you split an element by clicking in the PAD display on the axis inside that element. Click between surfaces 5 and 6, splitting the element. Your lens now looks like this:



When the program splits (or adds) an element, it assigns an index pickup, because at that moment it has no other index data. In WS, change the index pickup on surface 7 to a glass model by typing

```
7 GLM
```

in the edit pane, and click Update. That changes to a model glass with properties similar to what were there before.

Make a new checkpoint, close WS, run the optimization again, and we see that the lens has improved slightly. The MF is now 2.53. This is the way lens design has long been done, using classic tools, and it was a slow and arduous process. But today we have better tools. Go back to the version before you split the element:

```
GET 1
```

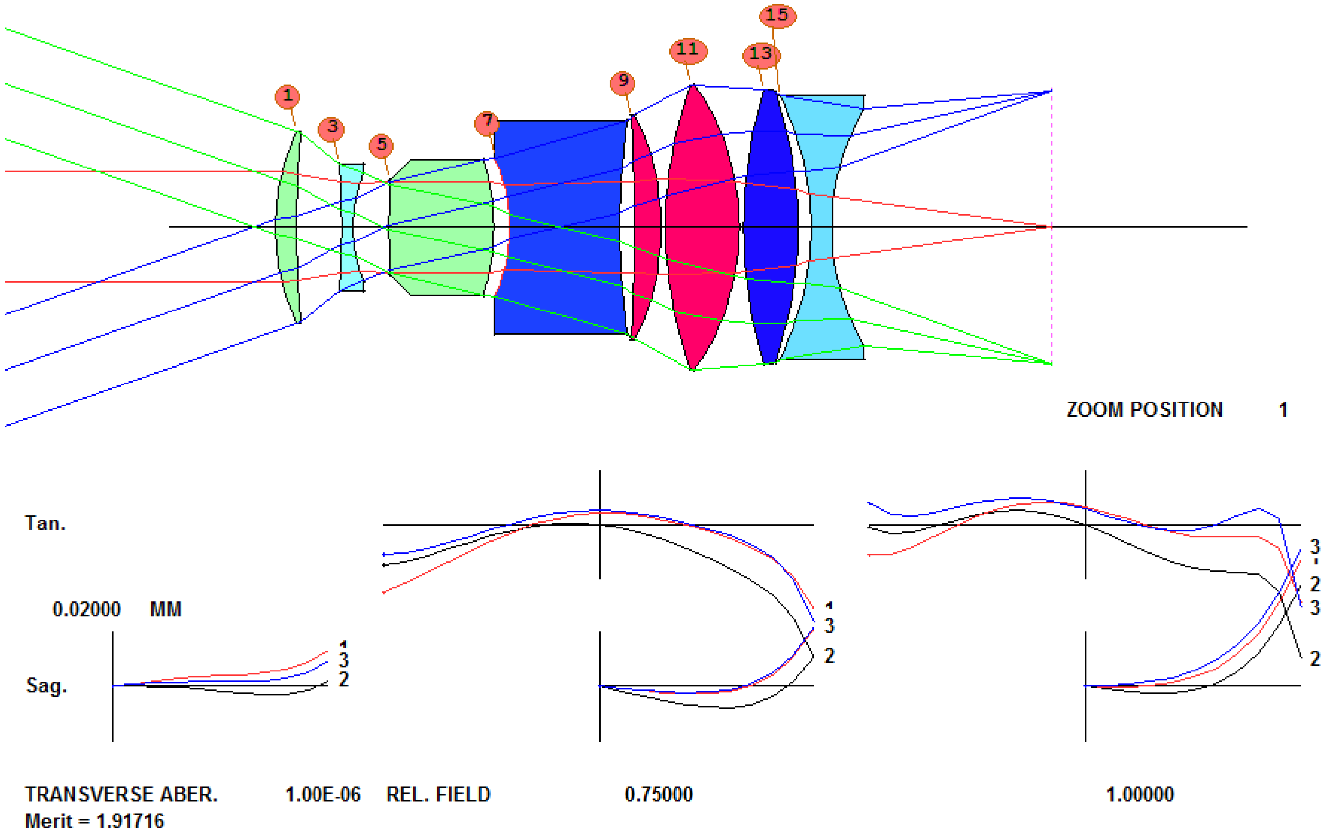
and then add a line before the PANT file:

```
AEI 2 1 14 0 0 0 10 2
```

This will run the Automatic Element Insertion tool (**AEI**). Now the program will search for the best place to insert a new element. Run this, and the lens is better. Comment out the AEI line and run your MACro again, then anneal. Here is the result:

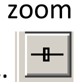
```
RLE
ID DSEARCH SAMPLE                          180
ID1 DSEARCH CASE WAS 00000000000000000001001111     79
 WAVL .6563000 .5876000 .4861000
 APS              5
 FFIELD
 UNITS MM
 OBB  0.000000     19.41264     12.70000    -11.00540      0.00000      0.00000     12.70000
   0 AIR
   1 RAD     53.9413943790523   TH      4.77883929
   1 GLM      1.90000000            37.62897436
   2 RAD    256.2741391536815   TH     10.43791469 AIR
   3 RAD   -240.8321927995665   TH      2.68192838
   3 GLM      1.55017293            45.90619514
   4 RAD     33.0833886630087   TH      8.23819322 AIR
   5 RAD    348.1550734974948   TH     24.04523087
   5 GLM      1.90000000            37.62897436
   6 RAD    -53.2450361188082   TH      3.59481775 AIR
   7 RAD    -41.0817136624587   TH     25.48983049
   7 GLM      1.90000000            22.54554176
   8 RAD    186.3645272710029   TH      3.44409527 AIR
   9 RAD   -336.9999206364553   TH      6.07694173
   9 GLM      1.50000000            73.64948718
  10 RAD    -57.1787045766177   TH      1.00000000 AIR
  11 RAD     95.1542848378137   TH     16.98321961
  11 GLM      1.50000000            73.64948718
  12 RAD    -57.2632094152352   TH      1.00000000 AIR
  13 RAD    108.6802069087533   TH     12.49861869
  13 GLM      1.77103153            26.03009105
  14 RAD    -94.5597002836689   TH      3.05982907 AIR
  15 RAD    -66.0716087885051   TH      4.69827793
  15 GLM      1.57603254            40.99972364
  16 RAD     53.2894699282010   TH     50.43814444 AIR
  16 CV       0.01876543
  16 UMC     -0.13986014
  16 TH      50.43814444
  16 YMT      0.00000000
  17 CAO     32.00000000       0.00000000       0.00000000
  17 CV       0.0000000000000   TH      0.00000000 AIR
 ZFILE   1
 CAM RANK   2
```

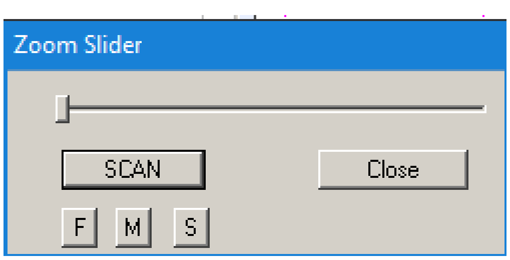
```
CAM EXPONENT    1.00000
16  16
ZOOM    2
OBA   1000.00000      -366.554000       12.7000    0.0000    0.00000000       0.0000  12.7000
ZDATA   0.0000000E+00
END
```



Wow! The program has inserted a new element at surface 3! And the merit function came down from about 2.55 to 1.92. There's a lesson here: The program can figure out how to improve a lens better than you can (unless you are *very* gifted). So it's better to let AEI do it than to try things that seem to make sense. Those things sometimes work, but AEI is better.

Here you see a larger improvement, and the MTF is also better, as you can check yourself. Now we have a lens that is fairly well corrected for both infinity conjugate and at one meter. But what about in-between distances? It would be a rude surprise if we built the lens and found that at an intermediate distance things got really bad. We have to check.

That is one of the reasons we chose to use the ZFILE zoom feature for this job. We can easily scan over the zoom range and spot any points that perhaps need attention. Click the button at the bottom of the zoom-selection bar:  This opens a zoom slider that is fun to watch.



Slide the thumb slowly to the right end, watching the PAD display (or click the SCAN button). The image plane slowly moves back, from the infinity focus to the one-meter focus position. The good news is, the image quality shows little change over the entire range, and in fact gets better near the middle. (If it had changed, we could use the CAM command to create an intermediate focus position, making a total of three zooms, and then add some more targets for the ZOOM 3 position in the AANT file.) You can create and target up to 20 zooms, as you will learn if you type HELP CAM to read about that feature.

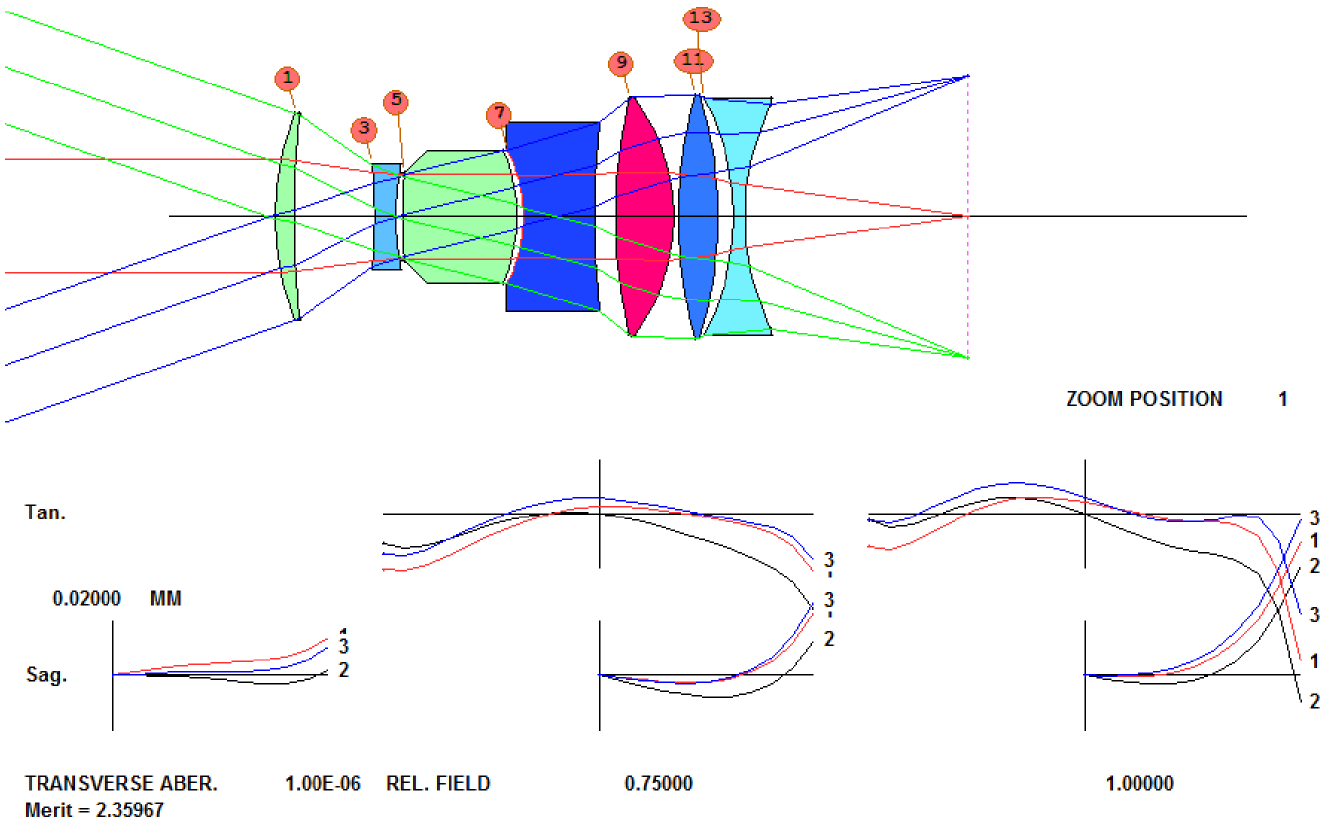
So we have roughed out a lens that works quite well over the entire focus range. Of course we are not yet done. Now we need to assign real glasses again, and it would be a good idea to increase the thickness of some of the elements, delete those thickness variables, and reoptimze. But … wait a minute. The fifth element shown in the picture above bothers us. What is it doing? Use the STRAIN command again, and you see that there is very little power or strain on that element. That's a sign that we just might be able to remove it entirely. We have to try! Remove the AEI directive and replace it with

```
AED 5 QUIET 1 15
```

And run it again, and – wow! The program says the *ninth* element can be removed! Allow it to do this, then comment out the AED directive and optimize some more. The merit function goes to 2.36 – not quite as good as before, but perhaps good enough. And we have eliminated an element. See how AED can make better decisions than you can?

```
RLE
ID DSEARCH SAMPLE                          180
ID1 DSEARCH CASE WAS 0000000000000000001001111     79
 WAVL .6563000 .5876000 .4861000
 APS             5
 FFIELD
 UNITS MM
 OBB  0.000000     19.41264     12.70000    -12.09057      0.00000      0.00000     12.70000
   0 AIR
   1 RAD     62.8507824648534   TH      4.25802685
   1 GLM      1.90000000            37.62897436
   2 RAD    242.2383021934368   TH     17.94509182 AIR
   3 RAD   -155.4943420012135   TH      4.72649410
   3 GLM      1.58912358            39.02768391
   4 RAD     40.3386502191948   TH      1.70305774 AIR
   5 RAD    150.7944944757465   TH     25.55442186
   5 GLM      1.90000000            37.62897436
   6 RAD    -38.9019256687224   TH      1.52918359 AIR
   7 RAD    -31.8151154746487   TH     16.13215543
   7 GLM      1.90000000            22.54554176
   8 RAD    266.4763779948293   TH      4.58032011 AIR
   9 RAD    115.8259371432369   TH     13.04257100
   9 GLM      1.60192516            64.47099564
  10 RAD    -44.3260121545059   TH      1.00000000 AIR
  11 RAD     98.4143150696891   TH      8.84868435
  11 GLM      1.85436291            26.23363793
  12 RAD    -92.1050493948654   TH      3.59579710 AIR
  13 RAD    -56.7923447824885   TH      2.56577649
  13 GLM      1.56906517            42.17387992
  14 RAD     56.3037237015490   TH     50.14291804 AIR
  14 CV       0.01776081
  14 UMC     -0.13986014
  14 TH      50.14291804
  14 YMT      0.00000000
  15 CAO     32.00000000      0.00000000       0.00000000
  15 CV      0.0000000000000   TH      0.00000000 AIR
 ZFILE   1
 CAM RANK   2
 CAM EXPONENT    1.00000
```

```
14  14
ZOOM     2
OBA    1000.00000      -366.554000       12.7000    0.0000    0.00000000        0.0000   12.7000
ZDATA    0.0000000E+00
END
```



So that's how it's done: Figure out what's wrong and use the tools in SYNOPSYS to fix it. Sometimes it's quick and sometimes not. That's what lens design is all about, blind alleys and all.

But that is probably enough for this lesson.

Oh, we almost forgot: Why did we enter the surface number (14) for the zooming group, since the YMT solve will override it anyway? Well, the program requires a group definition, and it won't work otherwise. That's to save you from a serious mistake if you ever leave those data out for a real zoom lens.

We will revisit this problem in Lesson 17 and show how yet other tools can be effectively applied and will save some time.