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.

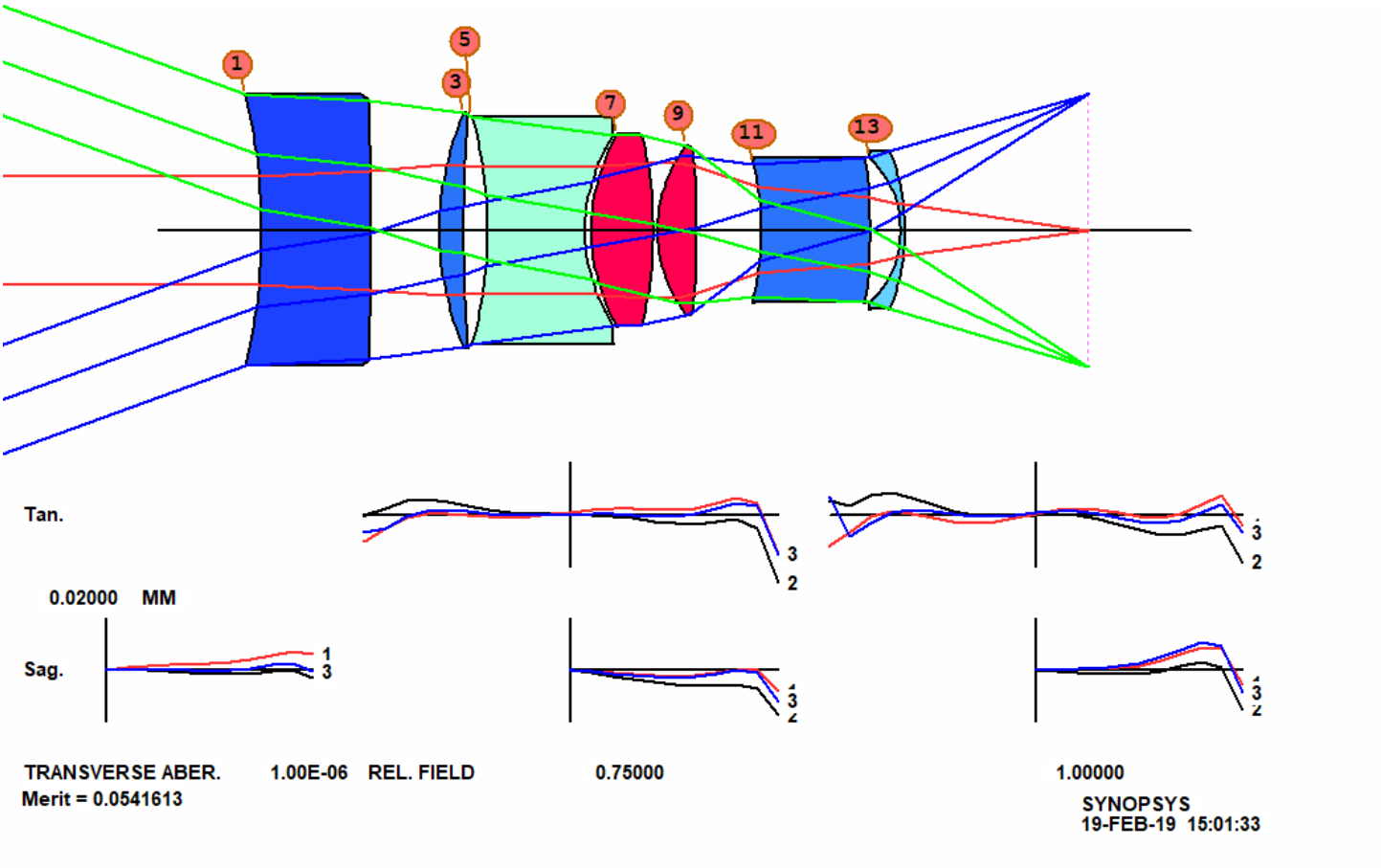
We will first use DSEARCH to find a good starting point. Here is the input:

```
CORE 14
 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, except for thin elements, which we will fix up later. We optimize and anneal, using the file DSEARCH_OPT, which is in a new editor window.

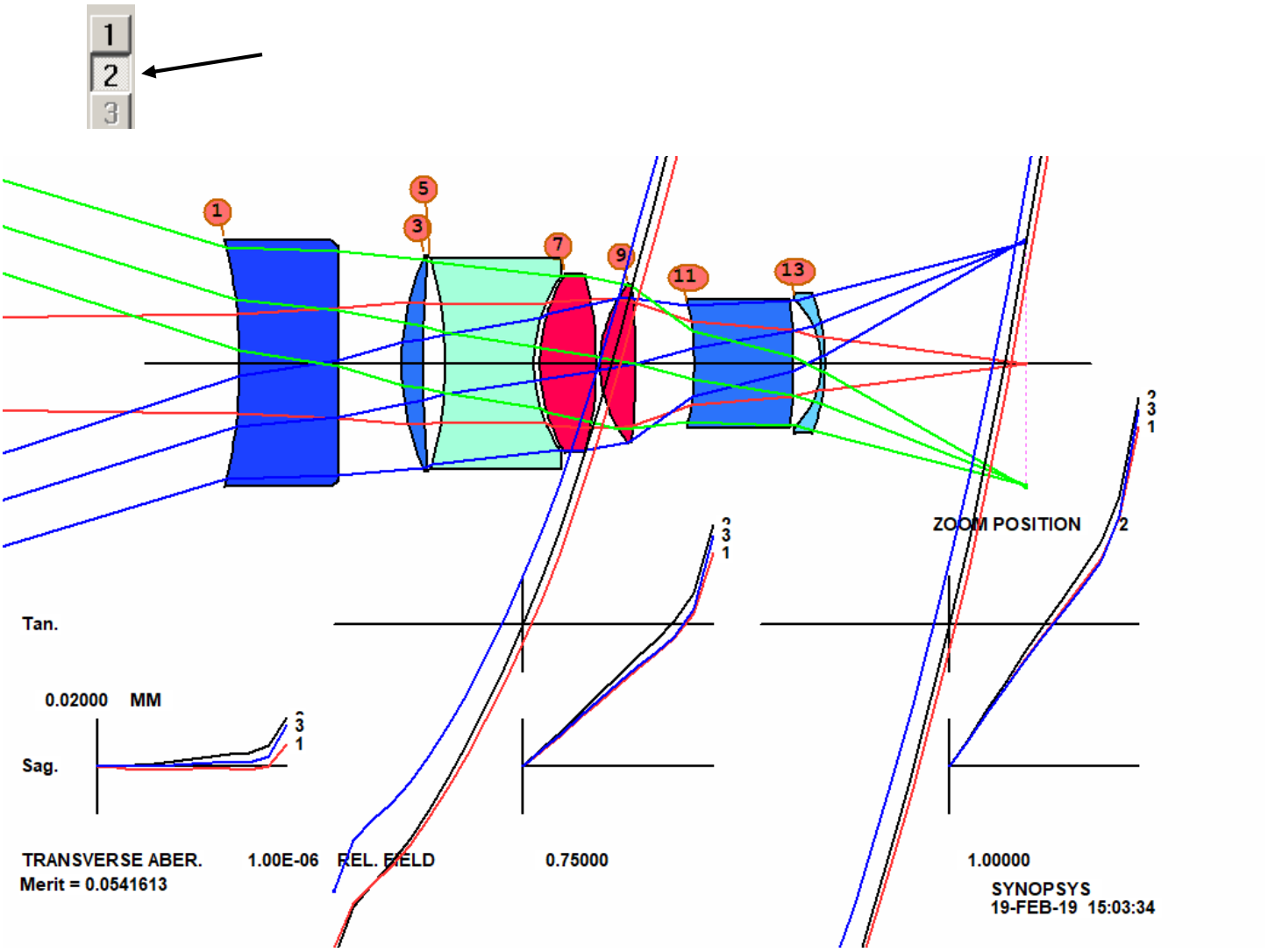


Suppose we want the lens to work over a range of object distances from one meter to infinity. There are two ways to implement that requirement: with multiconfigurations, which is very flexible but complicated, or by declaring this a zoom lens in which the object distance zooms. The second approach is 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 -10              ! declare surface 10 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 10 a real 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:

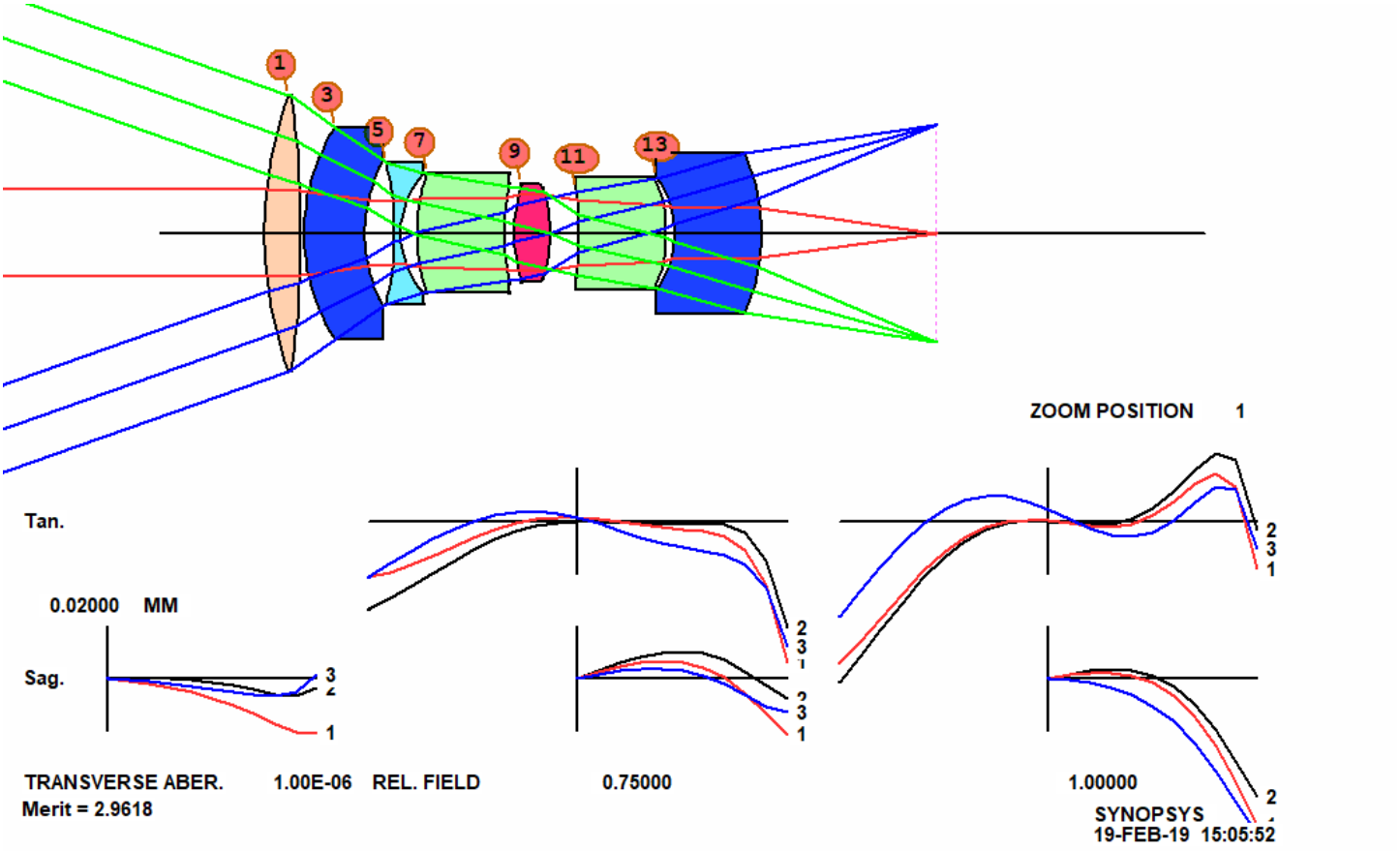
```
AWT: 0.5
PANT          ! Define variables.
!VY 1 YP1     ! Don't vary YP1; it is not compatible with the real pupil declaration
VLIST RAD ALL      ! Varies all radii that are not flat.
VLIST TH ALL       ! varies all thicknesses and airspaces except for the
! back focus, thickness 14, which has a 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 errors at both ends of the zoom range.

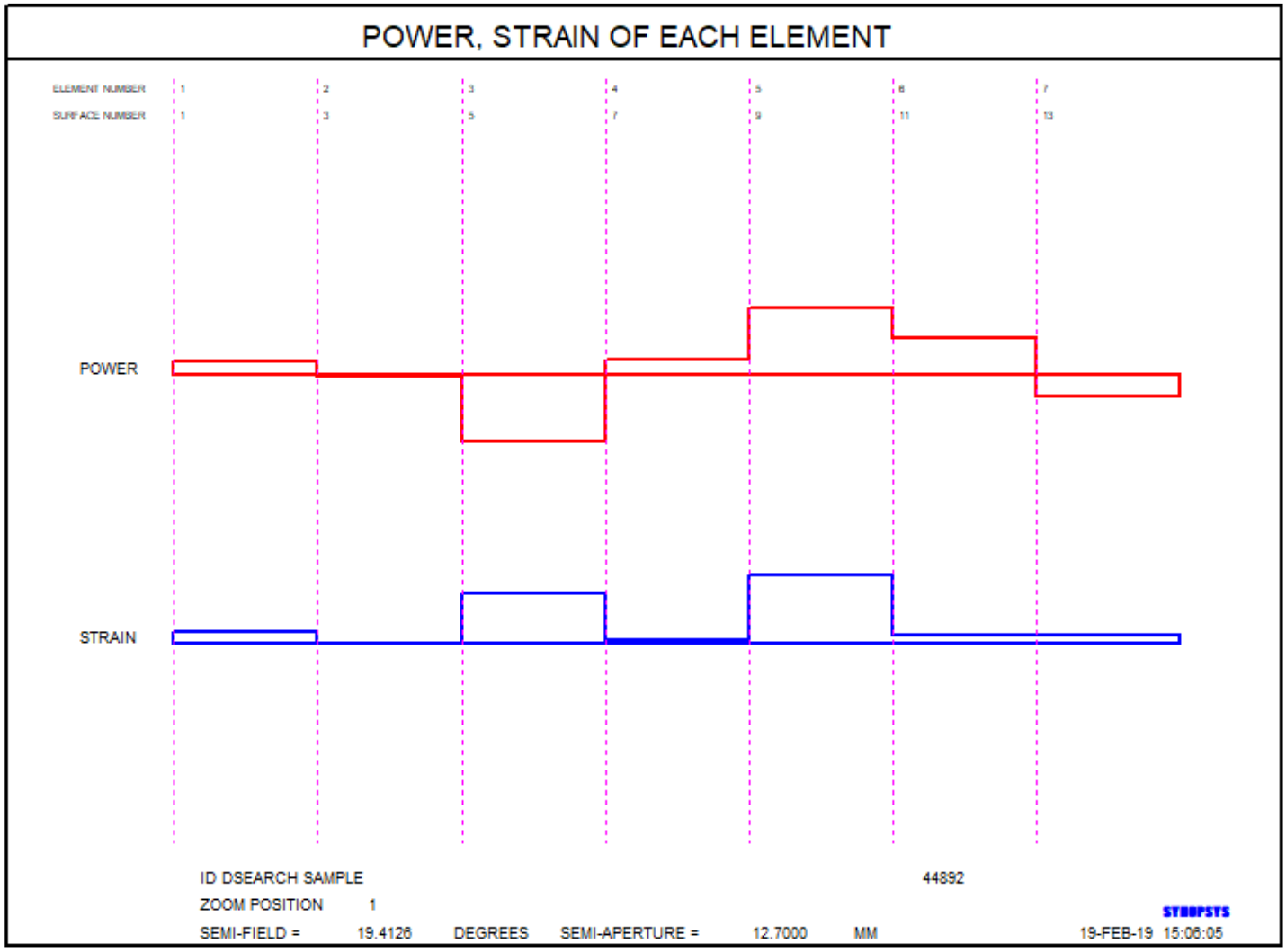


Some subtleties deserve mention: the **`GLM ALL`** variable will vary all glass models currently in the lens, which means all elements, since DSEARCH uses the glass model unless told otherwise. We have to control the focal length since the object height will be continuously adjusted so the image CAO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

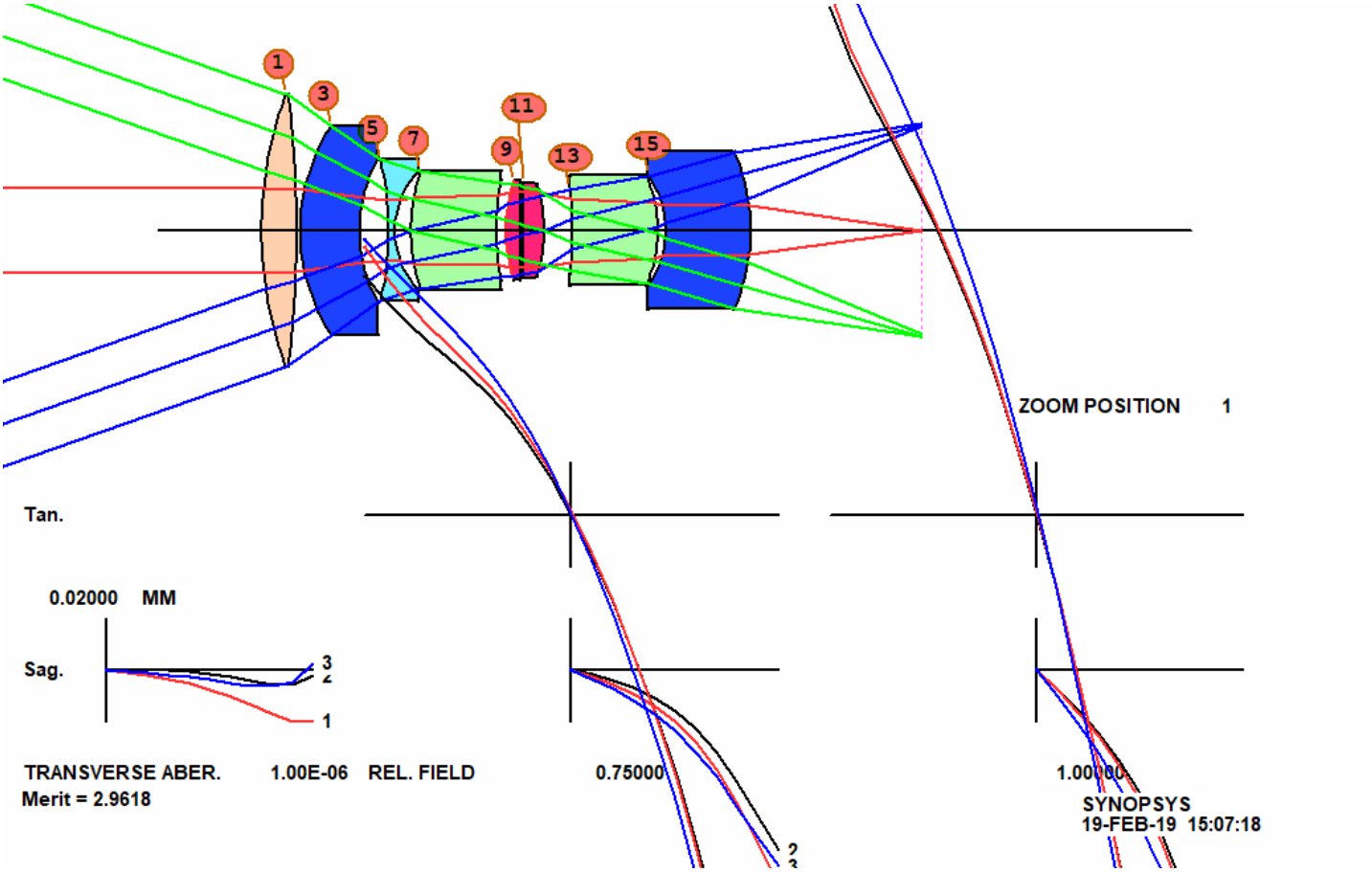
Type **`STRAIN P`** in the CW.



Indeed, element five 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 9 and 10, 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 11 to a glass model by typing

```
11 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 1.59. 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
```

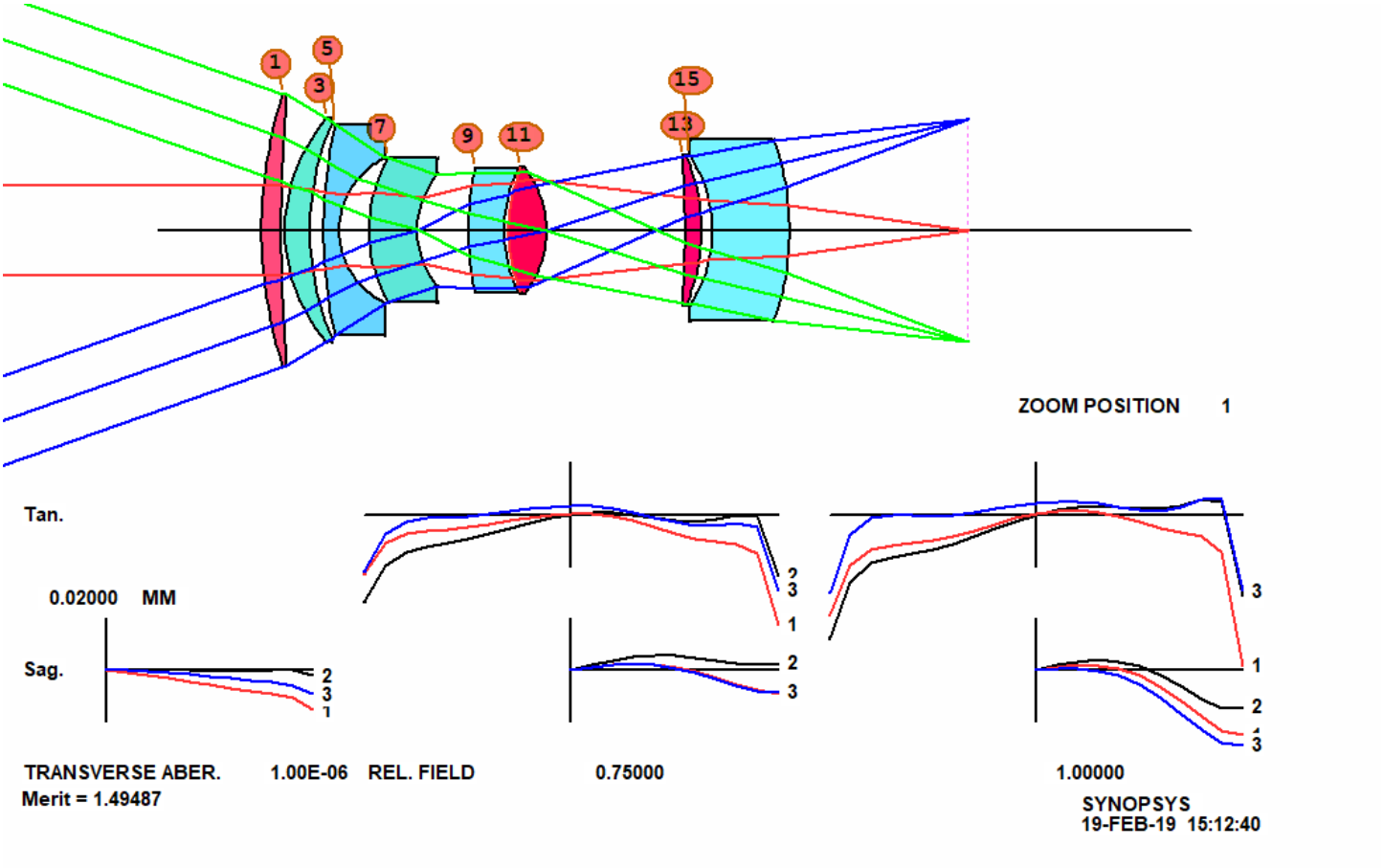
and then add a line before the PANT file:

```
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                          44892
ID1 DSEARCH CASE WAS 00000000000000000000100011     35
 FNAME 'AEI08.RLE                                          '
 MERIT    1.49487
 LOG    44892
```

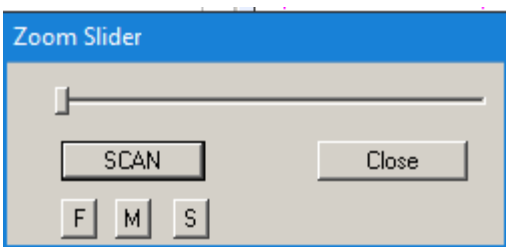
```
WAVL .6563000 .5876000 .4861000
APS            -12
FFIELD
UNITS MM
OBB  0.000000     19.41264     12.70000    -27.20408      0.00000      0.00000     12.70000
  0 AIR
  1 RAD    126.2610314937226   TH      5.65004368
  1 GLM      1.71002696             57.03505884
  2 RAD    548.9829634420437   TH      1.00000000 AIR
  3 RAD     48.8001314259707   TH      7.15931650
  3 GLM      1.90000000             30.45165541
  4 RAD     78.8914205169275   TH      3.92580954 AIR
  5 RAD    144.0284949074012   TH      4.36091211
  5 GLM      1.58044211             40.30348968
  6 RAD     22.8152407802378   TH      8.52017600 AIR
  7 RAD     41.6567444491098   TH     13.58034863
  7 GLM      1.90000000             29.97777667
  8 RAD     24.7675453690692   TH     14.60211862 AIR
  9 RAD     80.8352977692681   TH     10.27627981
  9 GLM      1.57399779             41.33289870
 10 RAD     43.3646405210182   TH      1.29508471 AIR
 11 RAD     49.7804194802432   TH     10.66304480
 11 GLM      1.50100000             77.45538462
 12 RAD    -28.4841065971860   TH     39.36991857 AIR
 13 RAD   -337.5623606237147   TH      4.69577023
 13 GLM      1.64849716             63.04604701
 14 RAD    -53.4480004366635   TH      3.02049855 AIR
 15 RAD    -38.5862175963575   TH     22.01446071
 15 GLM      1.56813982             42.33714495
 16 RAD    -74.9651540676835   TH     50.65153648 AIR
 16 CV      -0.01333953
 16 UMC     -0.13986014
 16 TH      50.65153648
 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 9! And the merit function came down from about 2.96 to 1.49. 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.

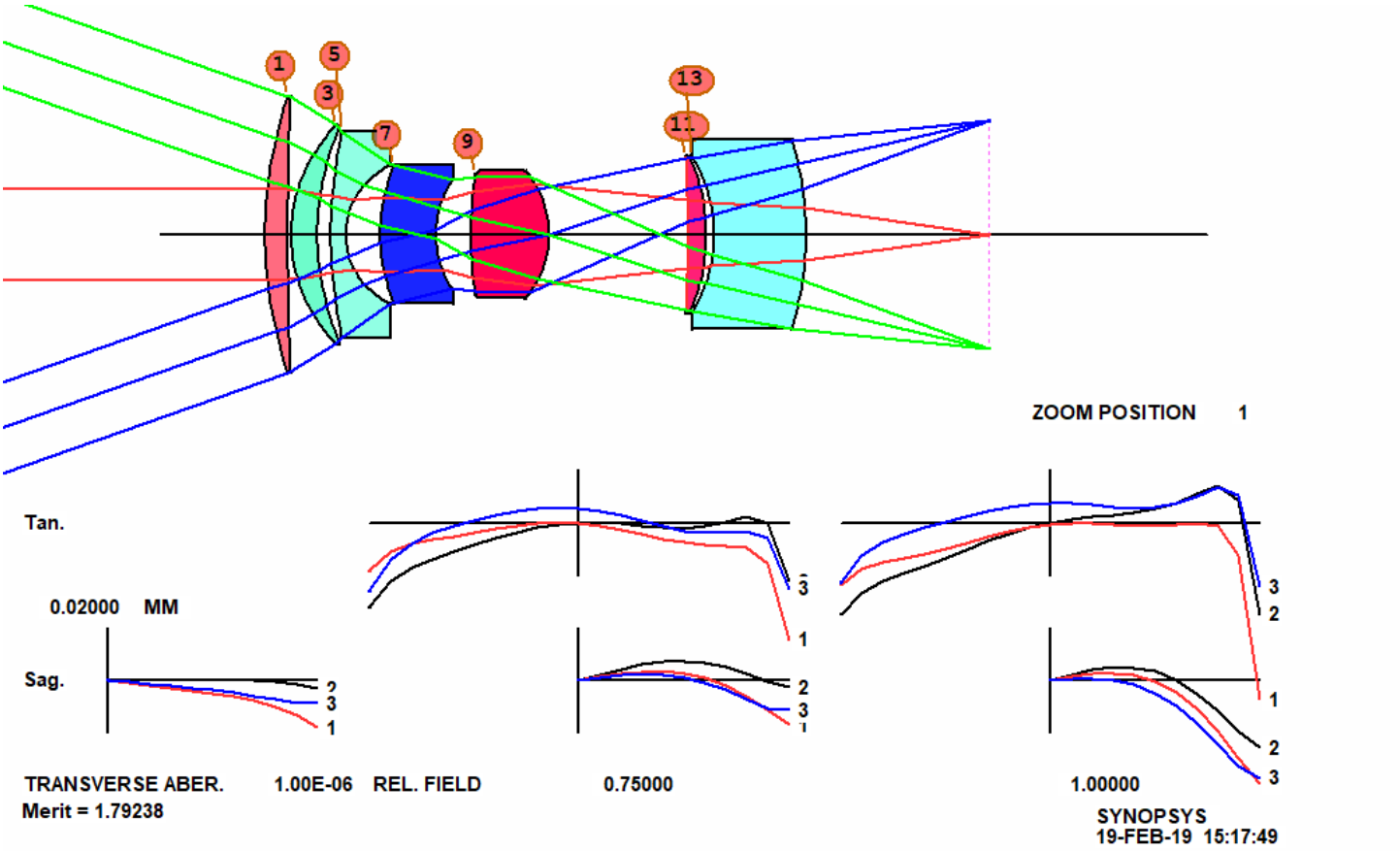
So we have roughed out a lens that works quite well over the entire focus range. Of course we are not yet done. Now we need to assign real glasses again, and it would be a good idea to increase the thickness of some of the elements, delete those thickness variables, and reoptimze. But … wait a minute. The fourth 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 that element at surface 9 can be removed. Allow it to do this, then comment out the AED directive and optimize some more. The merit function goes to 1.79 – not as good as before, but perhaps good enough. And we have eliminated an element. See how AED can make excellent decisions?

```
RLE
ID DSEARCH SAMPLE                          44892
ID1 DSEARCH CASE WAS 000000000000000000100011     35
 FNAME 'AEI08.RLE                                          '
 MERIT    1.79238
 LOG    44892
 WAVL .6563000 .5876000 .4861000
 APS             -10
 FFIELD
 UNITS MM
 OBB  0.000000     19.41264     12.70000    -26.64355      0.00000      0.00000     12.70000
   0 AIR
   1 RAD    120.8868431760548   TH      6.29742193
   1 GLM      1.73018312             55.06595682
   2 RAD    874.1452451507120   TH      1.00000000 AIR
   3 RAD     43.3723238775944   TH      7.22734397
   3 GLM      1.90000000             31.85177396
   4 RAD     75.5982585496480   TH      3.72513003 AIR
   5 RAD    136.0674307510774   TH      4.22284549
   5 GLM      1.67482513             42.21779395
   6 RAD     21.1183776902342   TH      9.49462964 AIR
   7 RAD     50.9917098403988   TH     15.50072204
   7 GLM      1.85120210             23.62968294
   8 RAD     27.2208332948113   TH      9.61128996 AIR
   9 RAD    105.6629879994650   TH     21.75575719
   9 GLM      1.50100000             77.45538462
  10 RAD    -27.0616542071280   TH     38.57499099 AIR
  11 RAD   -348.1756599714766   TH      5.06911986
  11 GLM      1.68456543             59.52245393
  12 RAD    -51.1885698039190   TH      2.16493310 AIR
  13 RAD    -40.9131894386343   TH     25.47601346
  13 GLM      1.55595432             44.66796234
  14 RAD    -87.3681162352767   TH     50.80907300 AIR
  14 CV      -0.01144582
  14 UMC     -0.13986014
  14 TH      50.80907300
  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.