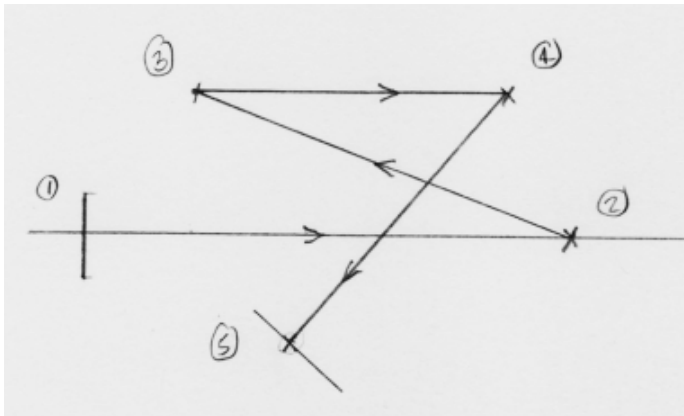


## Lesson 39: Designing a Free-Form Mirror System

Optics fabrication is becoming more and more sophisticated as better methods of shaping and measuring optical surfaces become available. One result is the emergence of “free-form” optics, which consists of lenses or mirrors whose shape is not axially symmetric about the vertex of the part. A simple example is the off-axis paraboloid, where the parent is polished to the required aspheric shape and then the desired part is cut from that parent. More elaborate shapes may involve high-order aspheric terms described with either power series, Zernike, or Forbes polynomials. As interest shifts to such systems, it becomes important for you to be able to design them.

SYNOPTIS provides features that can simplify that process. Read about **FFBUILD** in the help file. In this lesson, we assume you have read that chapter, and we will show how to design a system with free-form mirrors.

The first step is to lay out the rough geometry we are after. Here is an example with three mirrors:



Light will come in from the left at surface 1, hit mirrors located at 2, 3, and 4, and then go to the image plane at 5. Here is the input for FFBUILD:

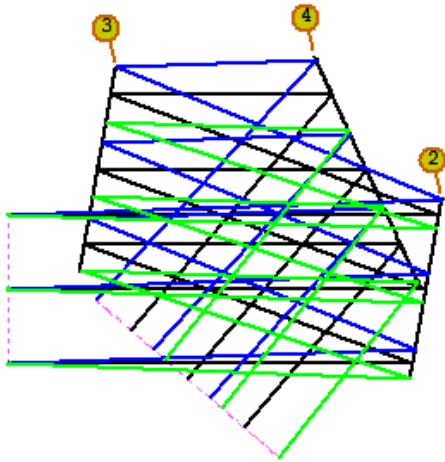
```
FFBUILD 3
SYSTEM
ID EXAMPLE FFBUILD
OBB 0 2 25 0
WAVL CDF
UNI MM
CFOV
END

GEOM
2 MIRROR 0 0 140
3 MIRROR 0 40 30
4 MIRROR 0 40 120
5 IMAGE 0 -30 60 -10 10
END

SHAPES
2 ZERN
3 ZERN
4 ZERN
END
```

In this example the mirrors will be assigned Zernike polynomials, which accept up to 36 coefficients that are functions of the polar coordinates on the surface. Since FFBUILD only supports designs with bilateral symmetry, terms asymmetric in X will not be used.

The above input contains part of our requirements: the semi-field angle is two degrees, is circular, and the semi-aperture is 25 mm; we will add other requirements as we go along. But first we run the above input file, which produces two results: a mirror system (with flat surfaces at the moment) and an optimization MACro that contains most of the input needed to refine this design. Here is that system:



The optimization MACro is quite long and contains variables for the angles and global positions of the mirrors and image plane in Y and Z, as well as for the Zernike coefficients on the mirrors. Most of the variables are commented out, however, since we have found that the process works much better if you first rough out the design with only the radii and angles varying, and then add the other variables gradually as they are needed. Here is part of that MACro; the lines in green are commented out.

```

PANT
SKIP
VY  2  YG
VY  2  ZG
VY  3  YG
VY  3  ZG
VY  4  YG
VY  4  ZG
VY  5  YG
VY  5  ZG
EOS
VY  2  AG
VY  3  AG
VY  4  AG
VY  5  AG
VY  2  RAD
! VY  2  CC  10 -10
! VY  2  G   3
! VY  2  G   4
! VY  2  G   7
! VY  2  G   8
! VY  2  G  10
! VY  2  G  11
! VY  2  G  14
! VY  2  G  15
! VY  2  G  16
! VY  2  G  19
! VY  2  G  20
! VY  2  G  23
! VY  2  G  24
! VY  2  G  26

```

```

! VY  2 G  27
! VY  2 G  30
! VY  2 G  31
! VY  2 G  34
! VY  2 G  35
! VY  2 G  36...

```

Most of the file consists of operands that will control the clearance of the beam as it bounces between mirrors. Here is a portion of that section:

```

LLL 1.0000 1 1.0000
A P CCLEAR 1 0 1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR 1 0 -1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR -1 0 1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR -1 0 -1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR 0 0 1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR 0 0 -1 0 1 3
S CAO 3
LLL 1.0000 1 1.0000
A P CCLEAR 1 0 1 0 1 4
S CAO 4
LLL 1.0000 1 1.0000
A P CCLEAR 1 0 -1 0 1 4
S CAO 4

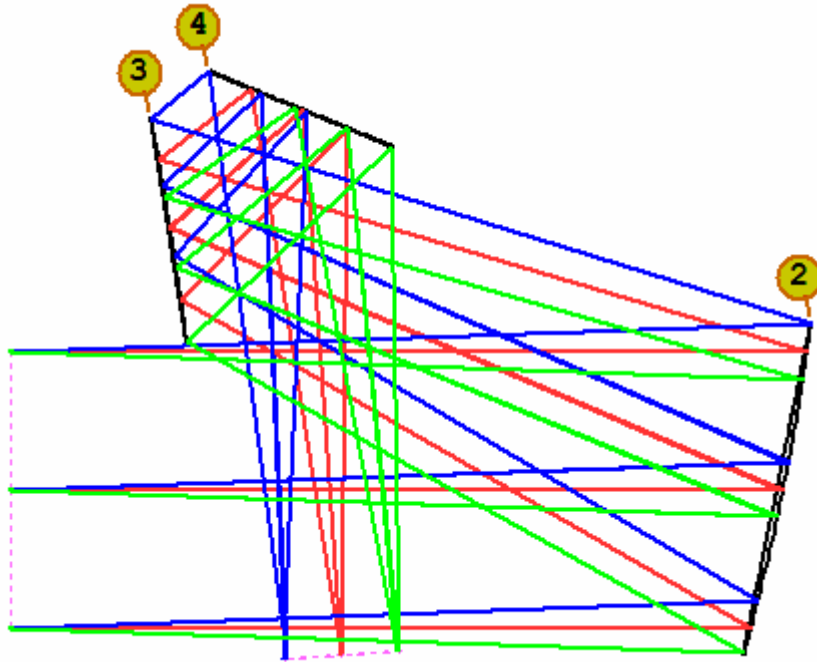
```

In this input, the program has specified a lower limit of 1 mm on the difference between the point where the upper rim ray at full field between surfaces 1 and 2 intersects surface 3, and the clear aperture radius (CAO) of that surface. As a result, if that intersection falls inside that clear aperture, the merit function gets a penalty, but if the ray clears that aperture, the penalty is zero. The program has also assigned the DCCR surface property to the mirrors, so the default clear aperture is centered between the extreme points on the surface required by the rays in the meridional field instead of at the vertex, which is the default. The remaining CCLEAR entries control the clearance between upper and lower rim rays at the top and bottom of the field between each mirror pair and the other mirrors. There are many combinations, and they all must be controlled.

The merit function contains GNR requests for seven points in the Y-Z plane, one in the skew field direction (because we put the CFOV directive in the SYSTEM file), and two in the upper and lower corners, and controls distortion in both X and Y with the GDR request (because we put a desired image size in words 6 and 7 of the IMAGE line in the GEOM section).

```
GDR 0 0.5 4 P 0.100000E+02 -0.100000E+02
```

It is a good idea to run this MACro just as it is, the first time around. That will get the design roughed out well enough so we can gradually improve things. After running it, the system starts to look more reasonable.



The image is formed in the right place and the beams clear all of the mirrors rather well.

We have specified the focal length we are after indirectly by controlling the YA at full field. It would not be a good idea to control FOCL itself, since FOCL is a paraxial property that does not have much meaning with a folded system such as this one. What we want is for the distance between the upper and lower field points at the image to be 20 mm. The program controls this with the **GDR** request, as noted above.

Our system is roughed out, but the image quality is not at all good. It is time to free up some of the other variables. It is wise to sneak up on the system little by little, so we will go slowly. (If you try to go too fast, the system will sometimes jump to a bizarre configuration that is nowhere near the nice region you are in now.) So we remove the comment character (“!”) before the variables for G 3 through G 8 on each surface by deleting the “!” character on those lines.

```

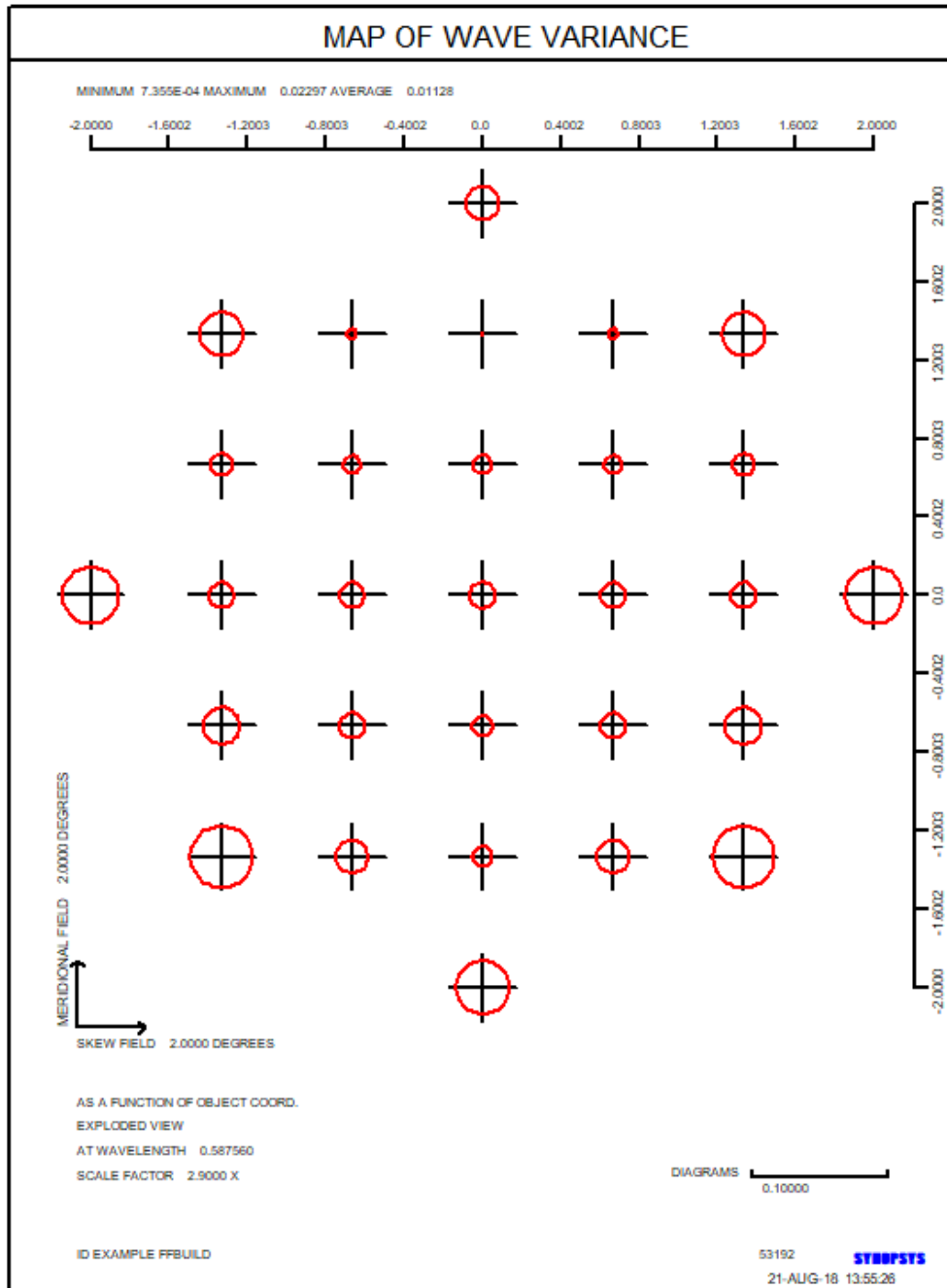
VY  2 RAD
! VY  2 CC 10 -10
VY  2 G  3
VY  2 G  4
VY  2 G  7
VY  2 G  8
! VY  2 G 10
...

```

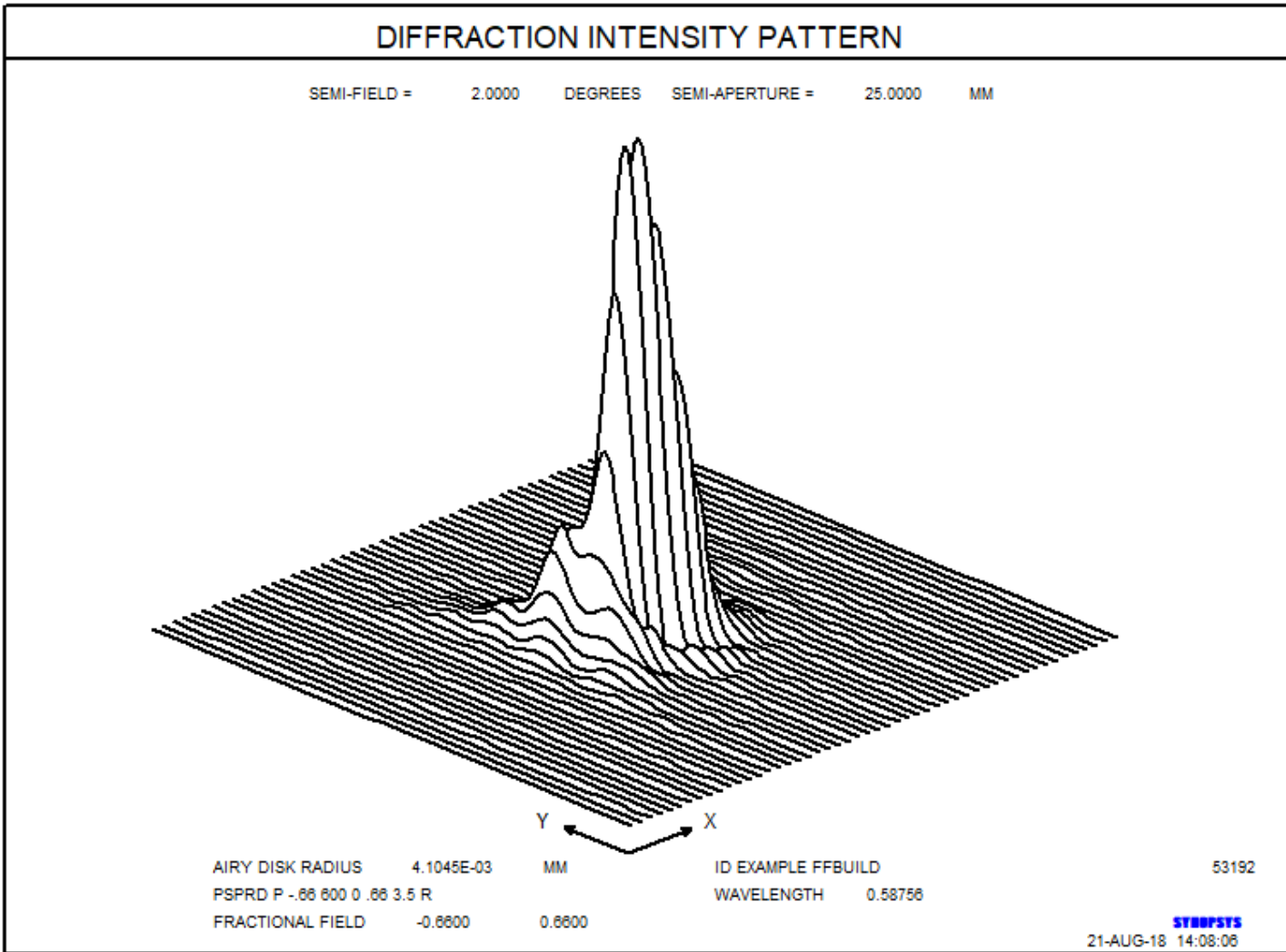
Do the same for surfaces 3 and 4, and then run the MACro and anneal (33, 2, 50). The MF comes down. Free up variables G 10 through G 20 and optimize and anneal again. The MF goes down to 0.0118. This is solid progress. Now free up the remaining G variables as well as the conic constant CC, and comment out the SKIP directive on the YG and ZG variables. (If you run this job yourself, the results will generally be different, since the annealing phase incorporates randomness into the process. But the overall quality will usually be similar.) Optimize and anneal, and the MF has come down to 0.011. Pretty good.

You are probably wondering why we did not let the CC vary right at the start. Well, the surfaces start out flat, and you cannot assign a conic constant to a flat surface. So we wait until we have nicely curved mirrors before letting the CC vary.

It's time to assess where we are. Go to the MAP dialog (MMA) and ask for a map of the wavefront variance over a grid of object points, object points CREC, ray pattern CREC 9, Show circles, EANALOG inch equivalent 0.1, and Execute. This is not bad.

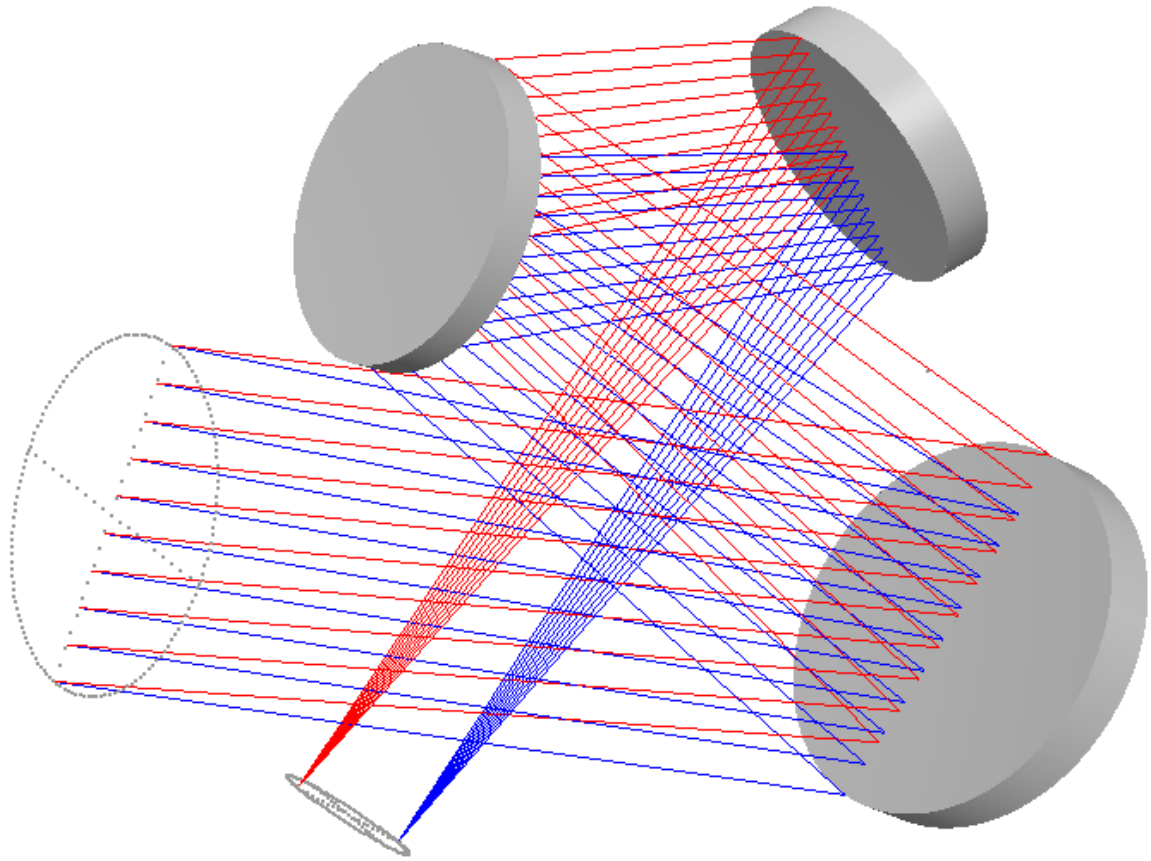


Let's examine the results. The worst field point is at HBAR -66 and GBAR .66. Here is that image, created by the **MDI** dialog:



All other points are better. This is not a bad design. Let's assume for this application we will use a CCD array sensor with pixels of 10 microns on a side, so this looks good.

You can get a better view with RSOLID, which shows only the portion of the surfaces within the decentered CAO. But first we go to the Edge Wizard (**MEW**) and select Create All and adjust the mirror thicknesses as we want them. Now the mirrors are assigned realistic edges and thicknesses. Then we create an RSOLID picture:

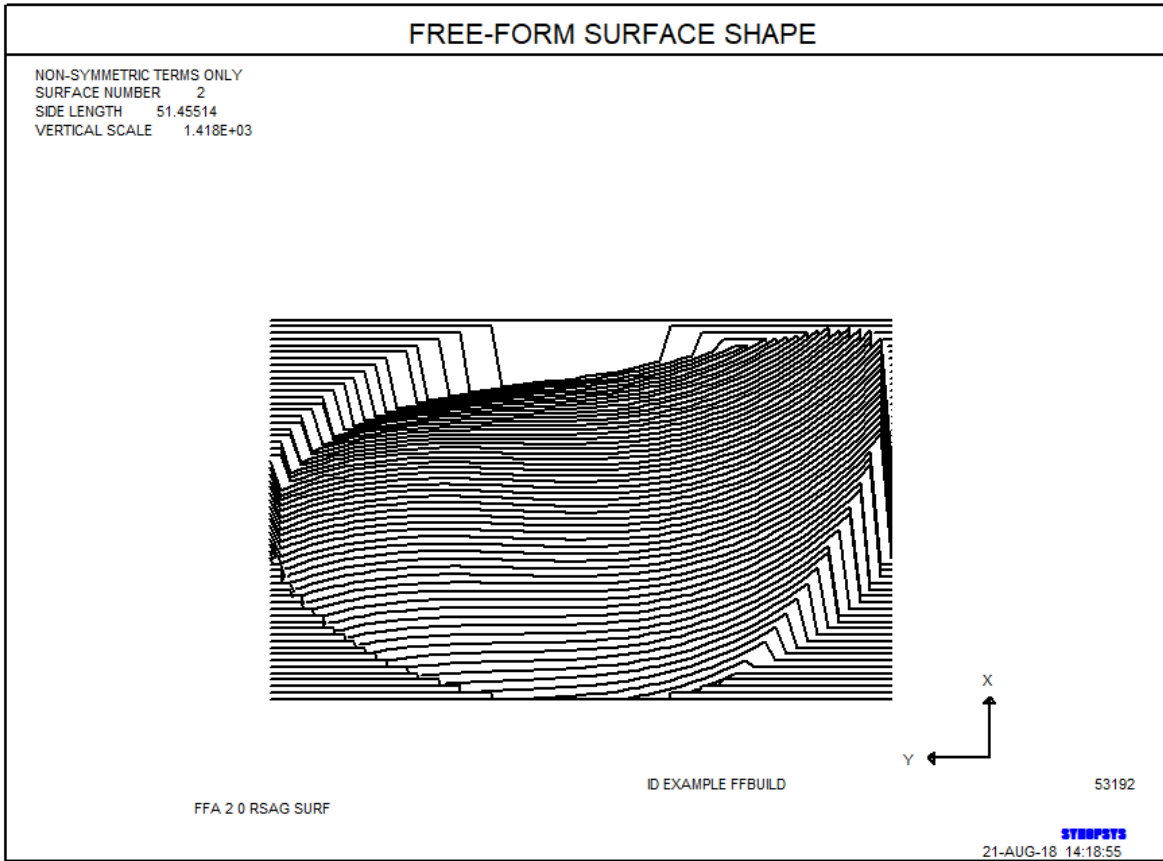


Our free-form system is designed.

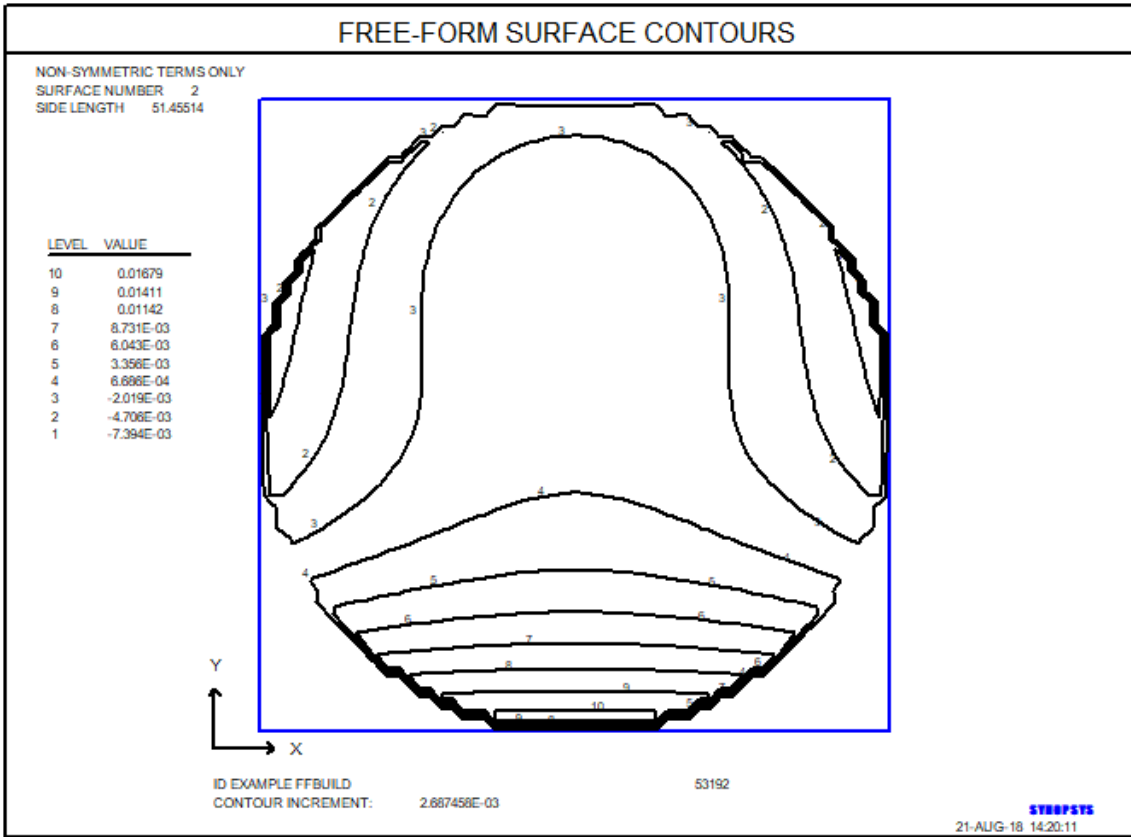
Now we can look at the shapes that resulted. The command

```
FFA 2 0 RSAG SURF
```

produces the picture below, which shows the *difference* between the actual shape and the base symmetric shape:

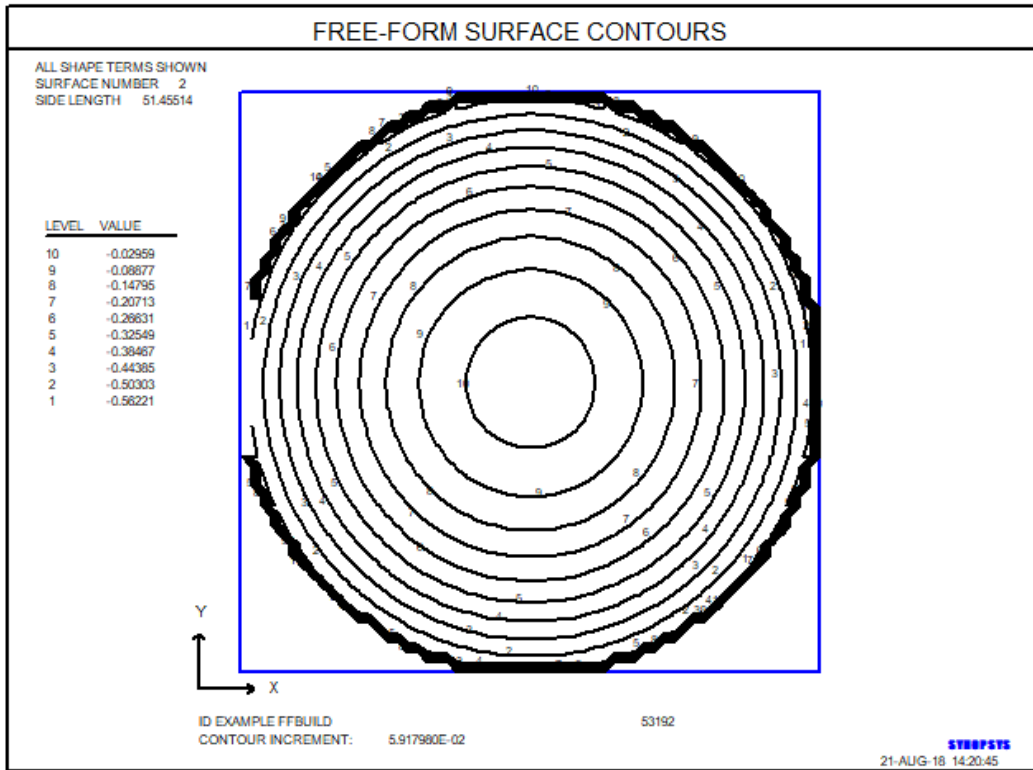


To see contours, we use FFA 2 0 RSAG CONTOUR



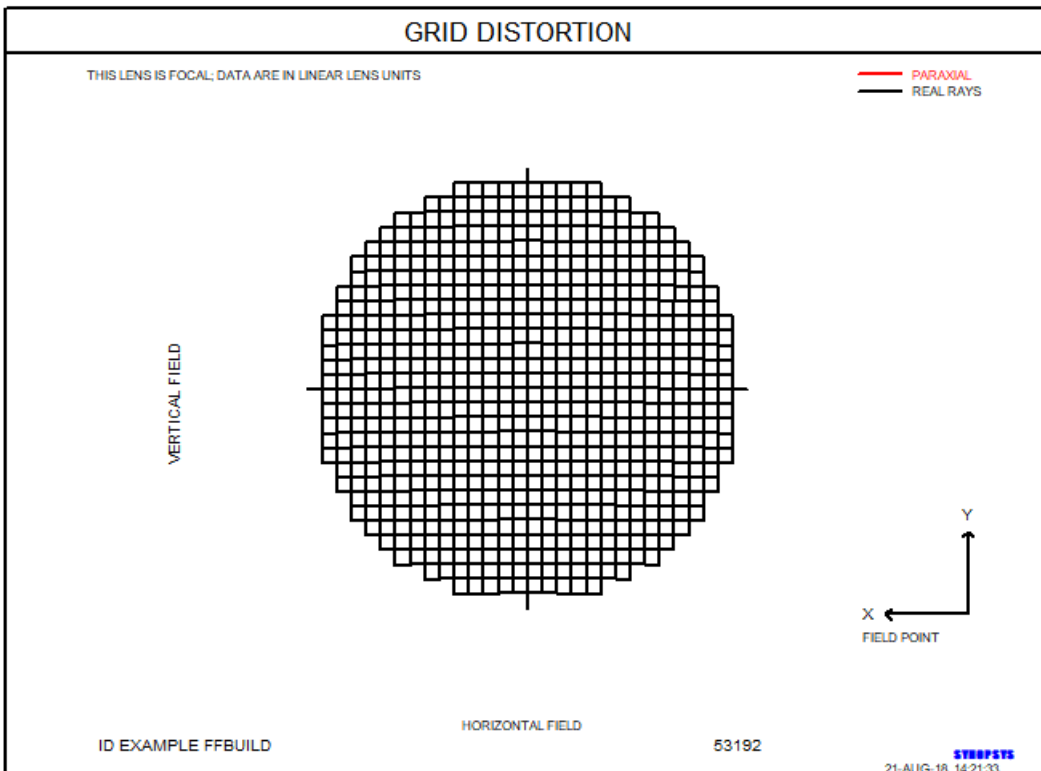


The shape of the actual surface is given by **FFA 2 0 SAG CONTOUR**:

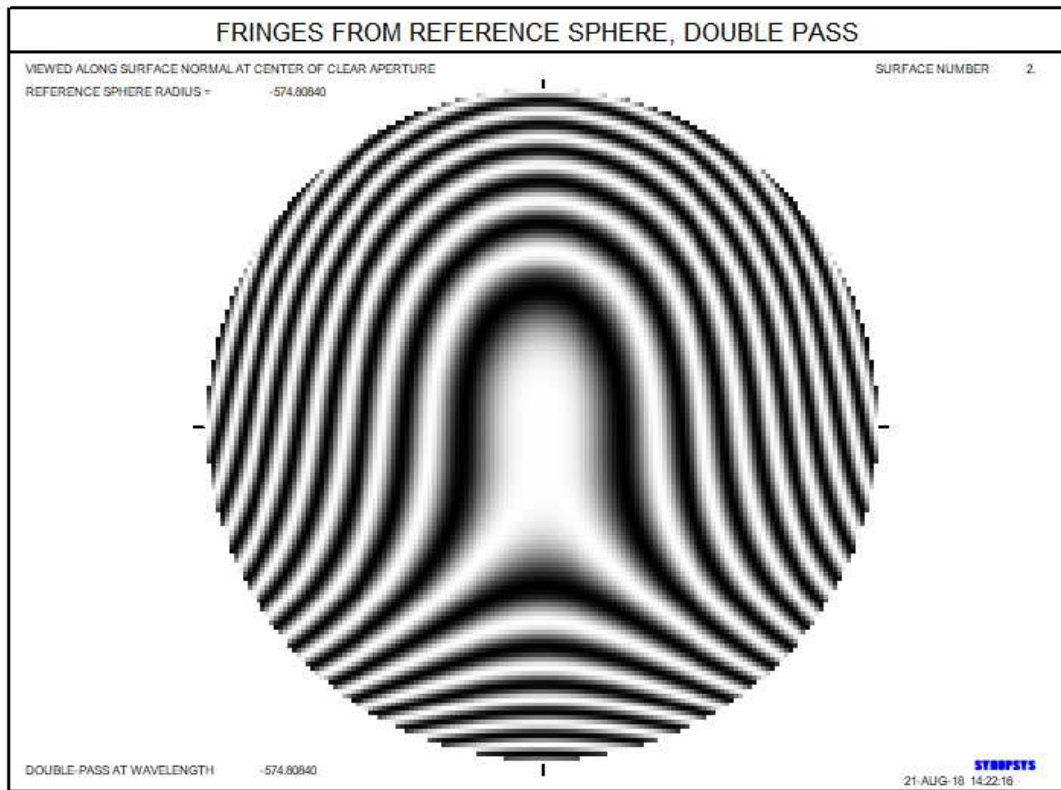


Proceeding in this manner, we can look at the shapes of all of our mirrors.

What about distortion? The GDR request has nicely handled that too. Here is the picture from the command **GDIS 31**. Not bad at all.



One more question remains: How does one test these mirrors? The easiest way is to look at the fringes when tested in an interferometer against a reference wavefront of a known radius. FFA can show that too. Here is the output from the command **FFA 2 0 RFRINGES**:



If you see that fringe pattern, the mirror is perfect.

So that's how one can design a free-form mirror system with advanced tools. The computer does most of the work for you.

Now it is up to you and the shop to communicate well enough so they understand the results and can make the parts correctly. Here are some pointers:

1. In this example, surface 4 is defined by Zernike terms, as we requested. Sometimes one gets even better results with variable G 39, which varies the center point of the expansion – which is therefore not at the vertex. And the latter is not at the center of the clear aperture either. So be careful here. There might be three center points to consider.
2. When presenting these data to the shop, be sure they understand the coordinate systems and locations of the relevant parameters.

Check out the other features of the FFA program. You can create a table of sags across the surface that will be of great interest to the technician running the precision milling equipment.