



AN036

Investigation of the Generation of Phase Branch Points with Wave-Optics Simulation

Author: Justin Mansell, Ph.D.
Revision: 1/10/12

Table of Contents

1	Abstract	2
2	Introduction	2
3	Branch Detection	2
4	Numerical Model – Generation of Phase Tears & Branches.....	3
4.1	Phase Aberrations.....	3
4.2	Propagation Parameters.....	4
4.3	$N_f=10$ Propagation Results	5
4.3.1	Half-Pyramid	5
4.3.2	Gaussian Bump.....	6
4.3.3	One-Dimensional Sinusoid.....	7
4.3.4	Full Pyramid	8
4.3.5	Two-Dimensional Sinusoidal Grating.....	9
4.3.6	Full Ramp	10
4.3.7	Half Ramp – Other Axis	12
4.3.8	Gauss Normalized Random Sum of 9 Zernikes.....	12
4.3.9	Gauss Normalized Random Sum of 14 Zernikes	14
4.4	Discussion of Modeling Results.....	16
5	Conclusions	16
6	Acknowledgements.....	16
7	References	16

1 Abstract

Propagation of light through a strong turbulent atmosphere is known to produce phase ambiguities that correlate with deep fades in the irradiance. These ambiguities, called branch points and branch cuts in the literature have been an active area of research for many years. This application note documents some simple numerical modeling experiments that were conducted to attempt to illicit a simplified cause of this phenomenon that has been studied primarily mathematically.

2 Introduction

Branch points and branch cuts in the complex optical field are a commonly observed phenomenon both in numerical simulations and measurements of propagation through fairly strong turbulence.¹ Fried described branch cuts concisely when he said, "It is shown that, when the scalar field associated with the propagation of a distorted wave function has nulls in its intensity pattern, the phase function that goes with that scalar field has branch points at the location of these nulls and that there are unavoidable 2π discontinuities across the associated branch cuts in the phase function." The most common wavefront sensor for adaptive optics, the Shack-Hartmann wavefront sensor, has great difficulty reconstructing the wavefront at branch points using common numerical reconstruction methods.² There has been considerable work in handling these branch points for adaptive optics.³ Most of this work has been done using highly sophisticated numerical methods on complex laboratory or computer generated phase profiles often using very sophisticated mathematical methods. This application note attempts to approach branch points in a simple phenomenological way to try to determine things like:

- What is the simplest way of generating branch points? Do you need multiple phase screens or can it be accomplished with a single simple continuous phase screen and propagation?
- How many discrete phase screens are required to demonstrate a branch point?
- Kolmogorov-spectrum turbulence can generate branch points, but is this spectrum of spatial frequencies necessary?
- What is physically happening to a beam to generate the branch?

3 Branch Detection

Before starting this investigation, a method for detecting a phase tear was needed. The most common way of detecting a branch point is to examine the local loop path integral for non-zero terms. In this case, the phase difference between four adjacent phase samples in a complex field was calculated, wrapped back to the range of $-\pi$ to π with the $\text{angle}(\exp(1j*\Delta\phi))$ functions in Matlab, and summed. To leverage the high speed processing routines in Matlab, the following code was used to identify branch points in the phase surface:

```
po = angle(Eout2);
dpo1 = angle(exp(1j*(po( 2:(nxy-0), 1:(nxy-1)) - po( 1:(nxy-1),1:(nxy-1)))));
dpo2 = angle(exp(1j*(po( 2:(nxy-0), 2:(nxy-0)) - po( 2:(nxy-0),1:(nxy-1)))));
dpo3 = angle(exp(1j*(po( 1:(nxy-1), 2:(nxy-0)) - po( 2:(nxy-0),2:(nxy-0)))));
dpo4 = angle(exp(1j*(po( 1:(nxy-1), 1:(nxy-1)) - po( 1:(nxy-1),2:(nxy-0)))));
Branches = dpo1 + dpo2 + dpo3 + dpo4;
```

where nxy is the number of mesh sample points in one dimension across the sampled field.

We limited the branches that were displayed to the area around the aperture of interest with the following code:

```
apBranch = aperture(nxy,dxy,rBranch);
Branches = Branches .* apBranch(2:nxy,2:nxy);
```

where aperture produces a simple circular 2D aperture with nxy samples in each axis and a mesh spacing of dxy and a radius of rBranch. Simplified code for this is: $[x=(-nxy/2:1:nxy/2-1)*dxy;$
 $[xx,yy]=\text{meshgrid}(x,x); r=\text{sqrt}(xx.^2+yy.^2); \text{apBranch} = (r \leq r\text{Branch});].$

We plotted the branch points on top of the phase of the beam using the following Matlab code. The index reversal in the plotting is due to the (row,column) vs (x,y) indexing of the arrays.

```
show(g.x,g.y,angle(EoutF));
[inds]=find((Branches)>0.5);
if (length(inds)>0)
    [xind,yind]=ind2sub(size(Branches),inds);
    hold on;
    plot(g.y(yind),g.x(xind),'g.','MarkerSize',20);
end;
[inds]=find((Branches)<-0.5);
if (length(inds)>0)
    [xind,yind]=ind2sub(size(Branches),inds);
    hold on;
    plot(g.y(yind),g.x(xind),'b.','MarkerSize',15);
end;
```

4 Numerical Model – Generation of Phase Tears & Branches

For this study, a 10-cm diameter 1- μm wavelength top-hat intensity profile beam was generated and then aberrated by a phase screen. The beam was then propagated a distance of 250 meters, which is equivalent to a Fresnel number of 10. For the two-screen studies, we aberrated the beam again with a phase screen and propagated it again the 250 meter distance. In all the studies, the amplitude of the aberration was slowly increased by a multiplicative scale factor until the branch points were observed. The following Matlab code made up the core of the modeling:

```
for sf=0:sfStep:sfMax;
    E = ap .* exp(1j*sf*phs1); Ein = E;
    [Eout1,dxout,xout]=fresnelShort2D(E,dxy,z,wavelength);
% And then optionally:
    E = Eout1 .* exp(1j*sf*phs2);
    [Eout2,dxout,xout]=fresnelShort2D(E,dxy,z,wavelength);
    EoutF = Eout2; ...
```

4.1 Phase Aberrations

We began by generating a set of phase aberration shapes which are illustrated below. The Matlab code we used for this, dependent on a user selection variable `phstype`, is:

```
if (phstype==1 || phstype==4)
    roofPower = 1.0; %roof is linear if 1.0, parabolic if 2.0
    phs1 = -1.5*(abs(g.xx)./Dap + abs(g.yy)./Dap).^roofPower);
    if (phstype==1)
        phs1 = phs1 .* (g.xx>0);
        if (ppt) TitleSlidePowerPoint('Half Pyramid'); end;
    else
        if (ppt) TitleSlidePowerPoint('Full Pyramid'); end;
    end;
end;
if (phstype==2)
    offsetx = Dap/5; offsety = Dap/5;
    fs1 = Dap/2.0; %spatial frequency of the phase aberration
    phs1 = exp(-1*((g.xx-offsetx).^2.0*pi/fs1).^2);
    phs1 = phs1.*exp(-1*((g.yy-offsety).^2.0*pi/fs1).^2);
    if (ppt) TitleSlidePowerPoint('Gaussian Phase Bump'); end;
end;
```

```

if (phstype==3)
    fs1 = Dap*0.5; %spatial frequency of the phase aberration
    phs1 = sin(2.0*pi/fs1*g.xx); %not clear
    if (ppt) TitleSlidePowerPoint('1D Sine'); end;
end;
if (phstype==5)
    fs1 = Dap*0.75; %spatial frequency of the phase aberration
    phs1 = sin(g.xx*2.0*pi/fs1).* sin(g.yy*2.0*pi/fs1);
    if (ppt) TitleSlidePowerPoint('2D Sine'); end;
end;
if (phstype==6 || phstype==7 || phstype==8)
    roofPower = 1.0; %roof is linear if 1.0, parabolic if 2.0
    phs1 = -3*( abs(g.xx)./Dap).^roofPower );
    if (phstype==7)
        phs1 = phs1 .* (g.yy>0);
        if (ppt) TitleSlidePowerPoint('Half Ramp'); end;
    elseif (phstype==8)
        phs1 = phs1 .* (g.xx>0);
        if (ppt) TitleSlidePowerPoint('Half Ramp - Other Way'); end;
    else
        if (ppt) TitleSlidePowerPoint('Full Ramp'); end;
    end;
end;
if (phstype==9)
    Nzern = 14;
    v = randn(1,Nzern);
    phs1 = ZernikeCompose2(v,nxy,dxy,Dap/2,0,0);
    sfStep = 0.5;
    sfMax = 5;%1.9;
    if (ppt) TitleSlidePowerPoint(sprintf('Gauss Normalized Random Sum of
%i Zernikes',Nzern)); end;
end;

```

$g.xx$ and $g.yy$ are the 2D x and y coordinate of the grid with the origin on the mesh point at the center($x = (-nxy/2+1:nxy/2-1)*dxy$; $[g.xx,g.yy]=meshgrid(x,x)$).

4.2 Propagation Parameters

The aperture diameter (Dap) of interest for the modeling was set to 10 cm. The wavelength was 1 μm . The propagation grid was setup at 512 samples per axis (nxy). There was a fixed 4x guard band on the mesh established by setting the mesh sample spacing to $4*Dap/nxy$. The nominal Fresnel number (N_f or N_f) was set to 10, meaning that the propagation distance was set to $[(Dap/2)^2] / \text{wavelength} / N_f = 2.5 \text{ km}$. The phase multiplier was set to scan from 0 to 30 in increments of 1.0. The branch check diameter was set to double the aperture diameter. The propagation was setup with the following code:

```

Dap = 0.1;
nxy = 512;
dxy = Dap*4/nxy;
wavelength = 1.0e-6;
Nf = 10.0;
sfStep=1.0;
sfMax=30;%1.9;
z = (Dap/2)^2 / wavelength / Nf;
rBranch = Dap./1.0;

```

Using the propagation parameter theory we developed, the minimum number of mesh points is 256. The maximum mesh spacing is 1.3 mm. We are using 512 and 0.781 mm respectively, so our propagation is adequately sampled.

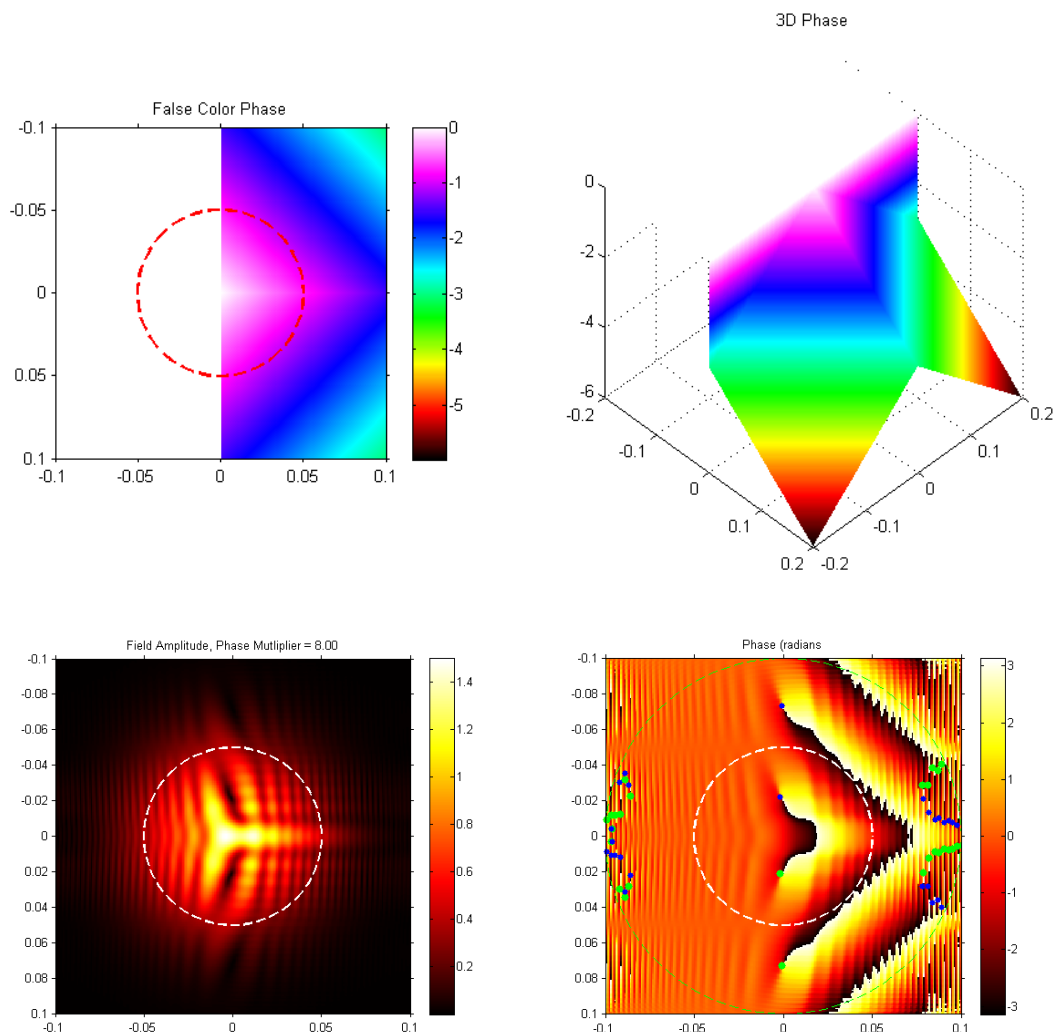
The Gaussian beam amplitude (not intensity) was created with the code:

```
ap = Gaussian(nxy,dxy,Dap/2);
```

4.3 $N_f=10$ Propagation Results

Each of the following sections shows a plot of the phase profile in 2D (false color) and 3D projection and one select result after a single $N_f=10$ propagation. The smaller dotted circle is the aperture diameter (red in the input phase and white in the output phase). The larger green dashed circle in the phase indicates the branch check radius.

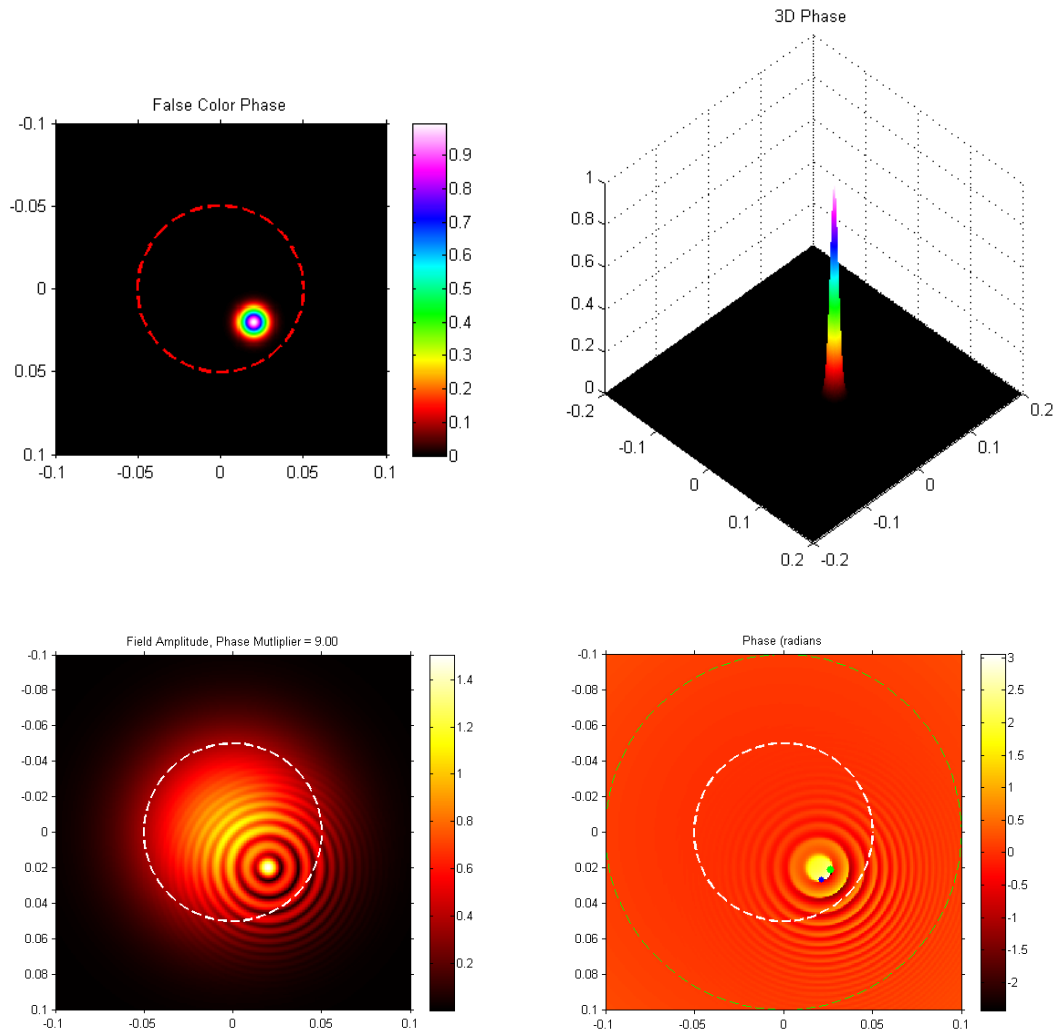
4.3.1 Half-Pyramid



This surface was discontinuous, but demonstrated an interesting form of a branch cut in the phase. The light from the slanted faces brought together to form a continuous cut but was interrupted by the discontinuity formed where the flat section met the slanted faces. This was a demonstration of a single phase screen

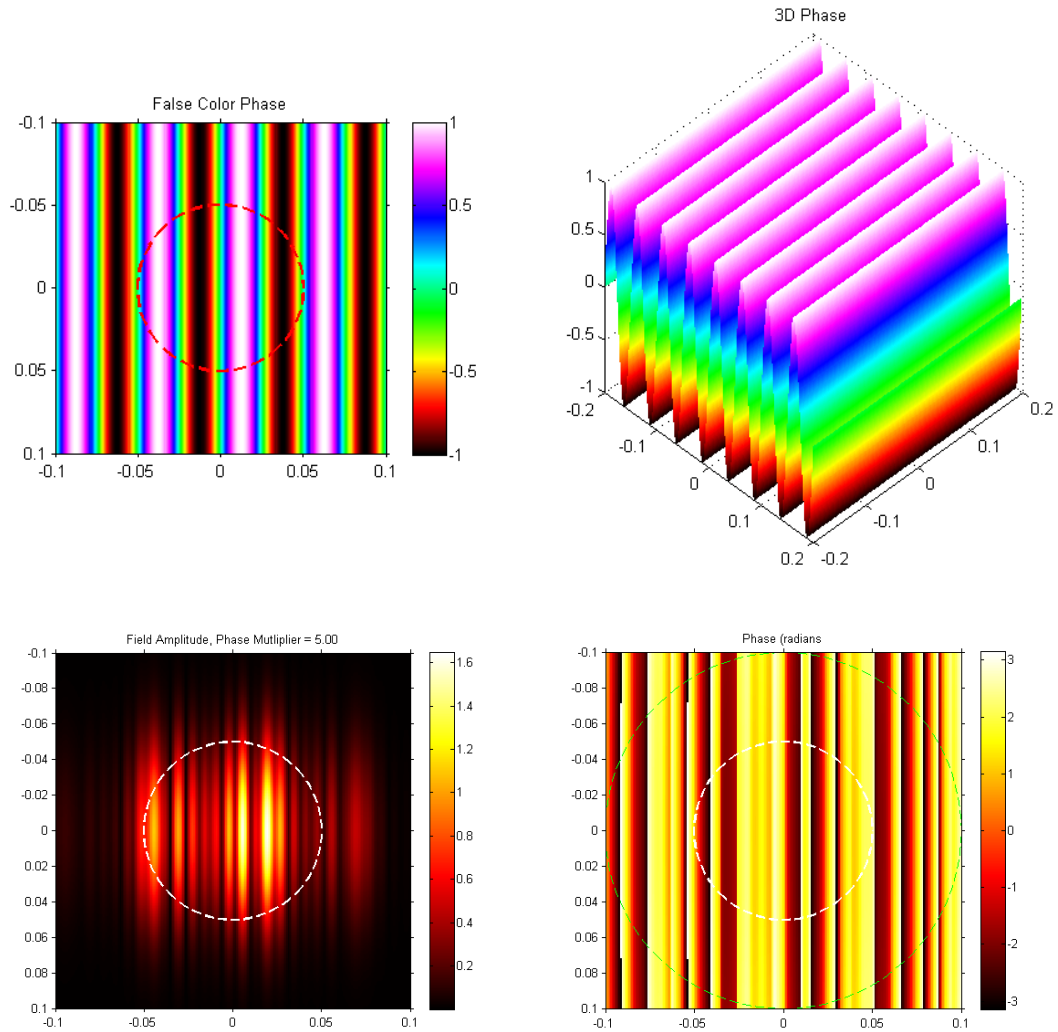
creating a branch after propagation (or before it in this case due to the discontinuity), but this does not meet our desired criteria of a continuous phase screen creating a tear after propagation.

4.3.2 Gaussian Bump



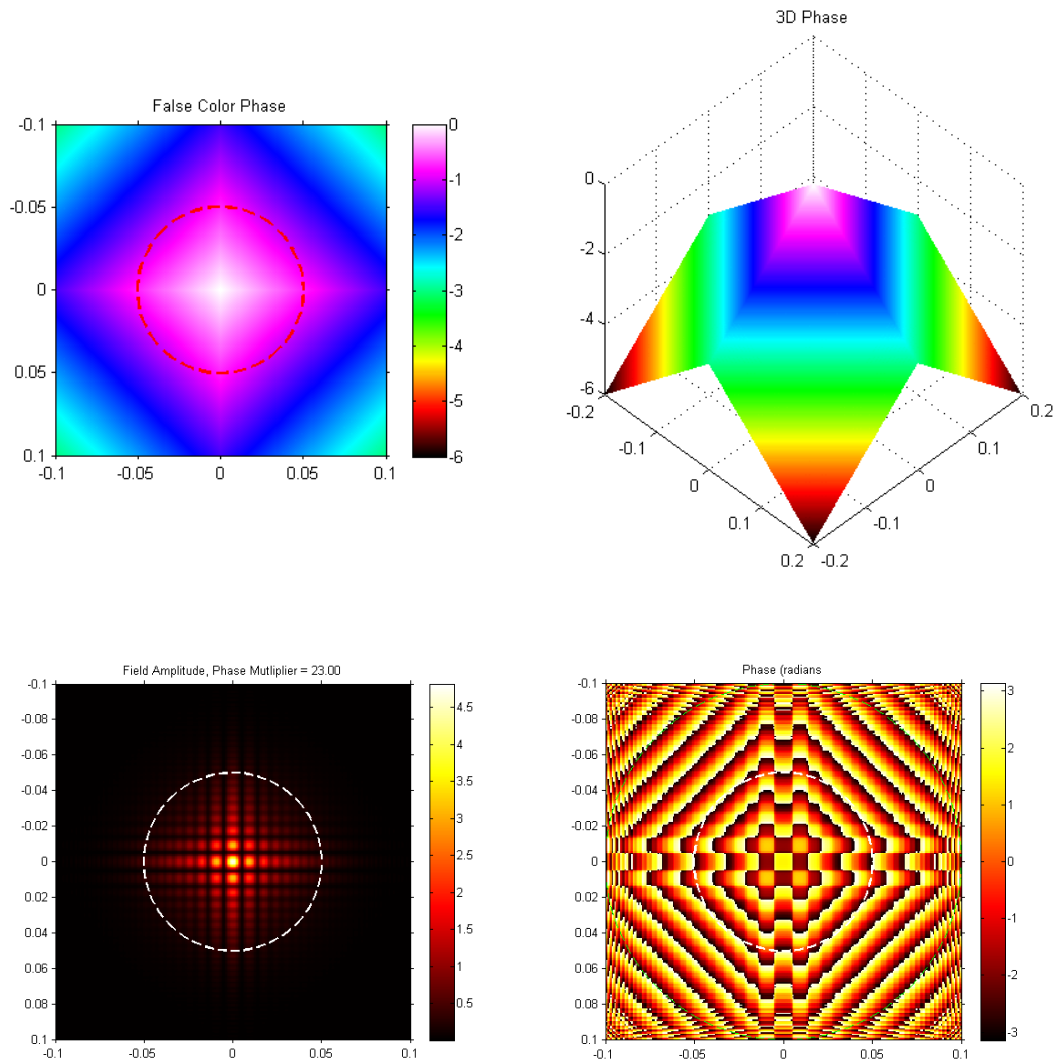
The Gaussian shape used here has a great deal of high spatial frequency phase content, but is continuous. This is a similar shape to a deformable mirror influence function. The Gaussian shape looks like a local focusing of the beam after propagation. As the amplitude of the phase profile increased to ~ 9 radians, evidence of a branch cut in the phase became evident at the first null of the local focusing. This is a good example of a single continuous phase screen producing a branch cut.

4.3.3 One-Dimensional Sinusoid



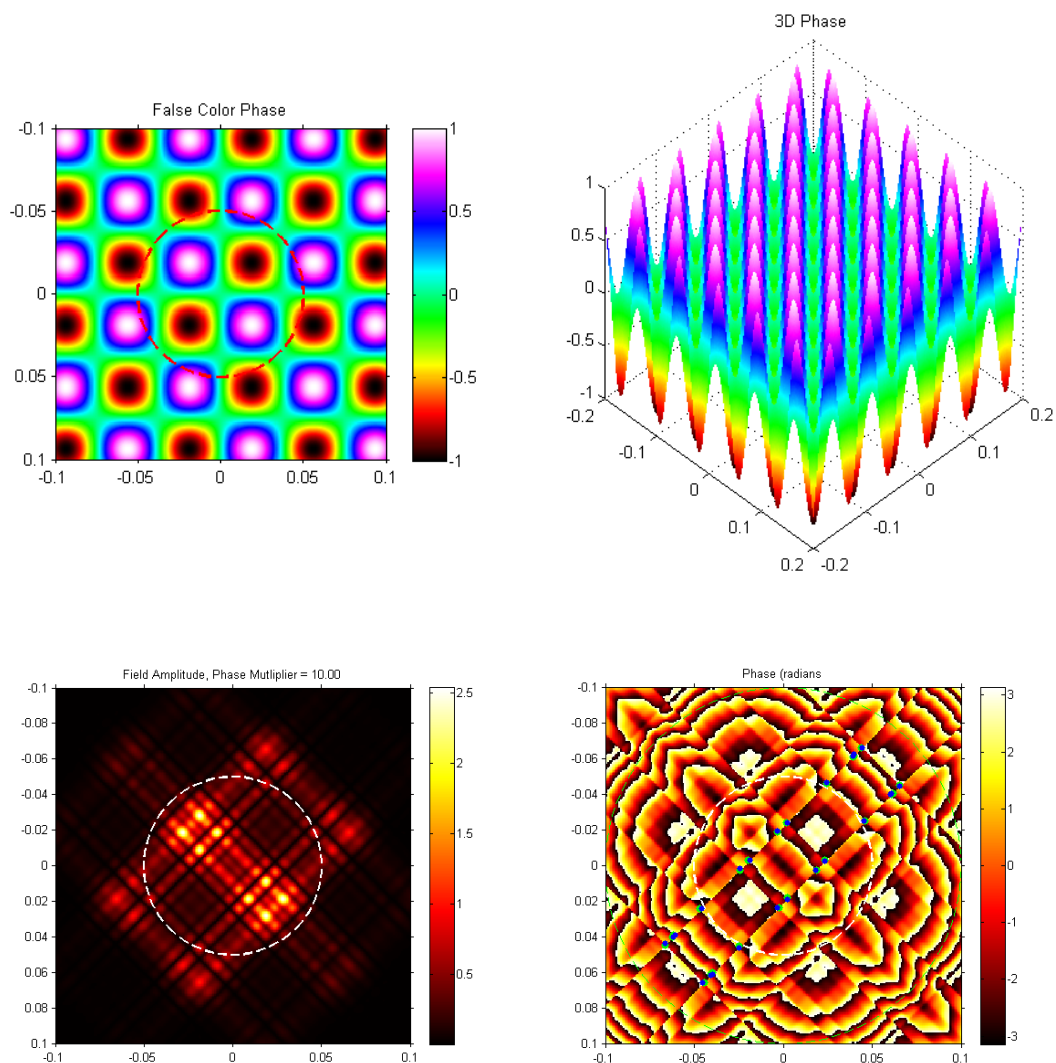
The 1D sinusoidal grating appeared to produce some branch cuts, but the end points were likely outside the aperture of the modeling.

4.3.4 Full Pyramid



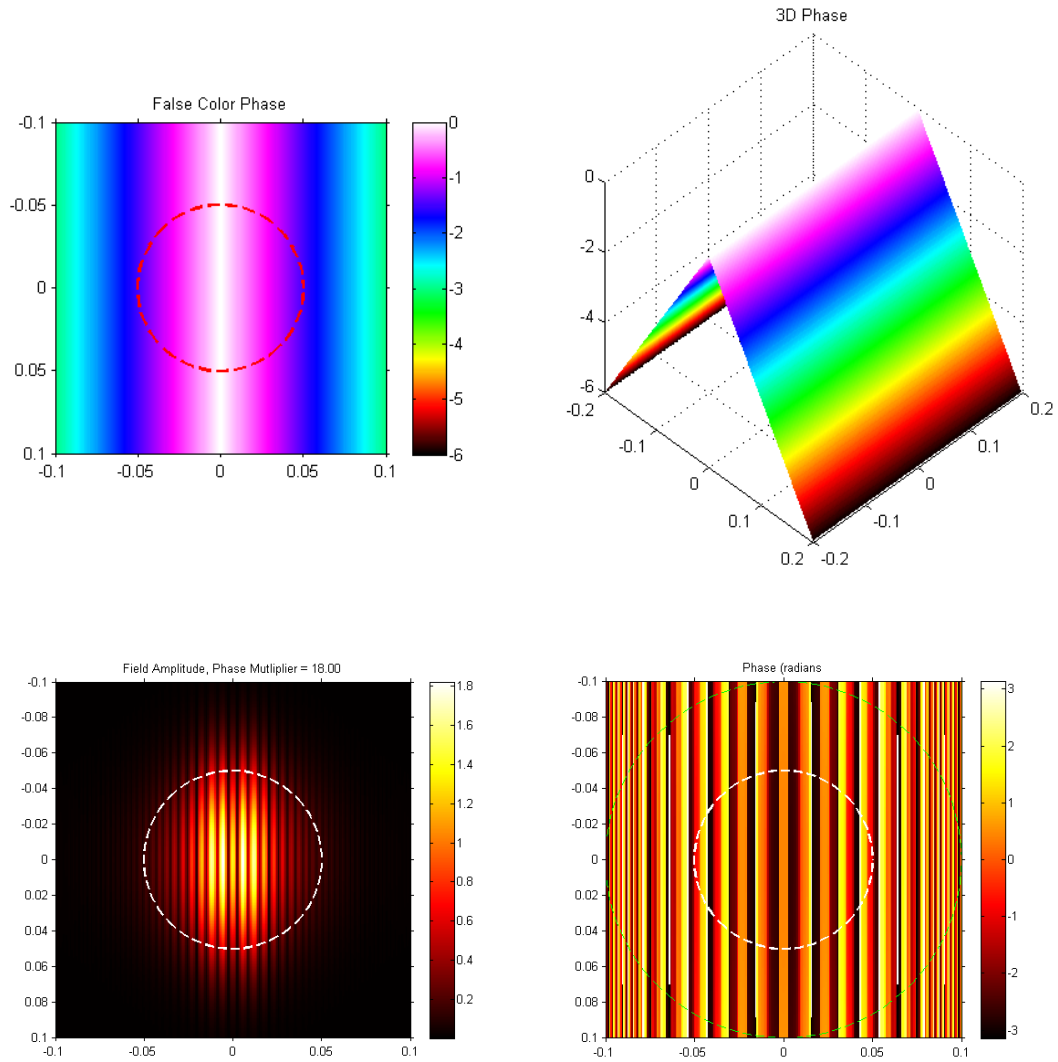
The full pyramid structure showed evidence of self-interference due to the clear intensity nulls in the propagated beam, but did not appear to produce any branch cuts or points in the phase after propagation. This phase profile is essentially equivalent to the interference of four beams.

4.3.5 Two-Dimensional Sinusoidal Grating

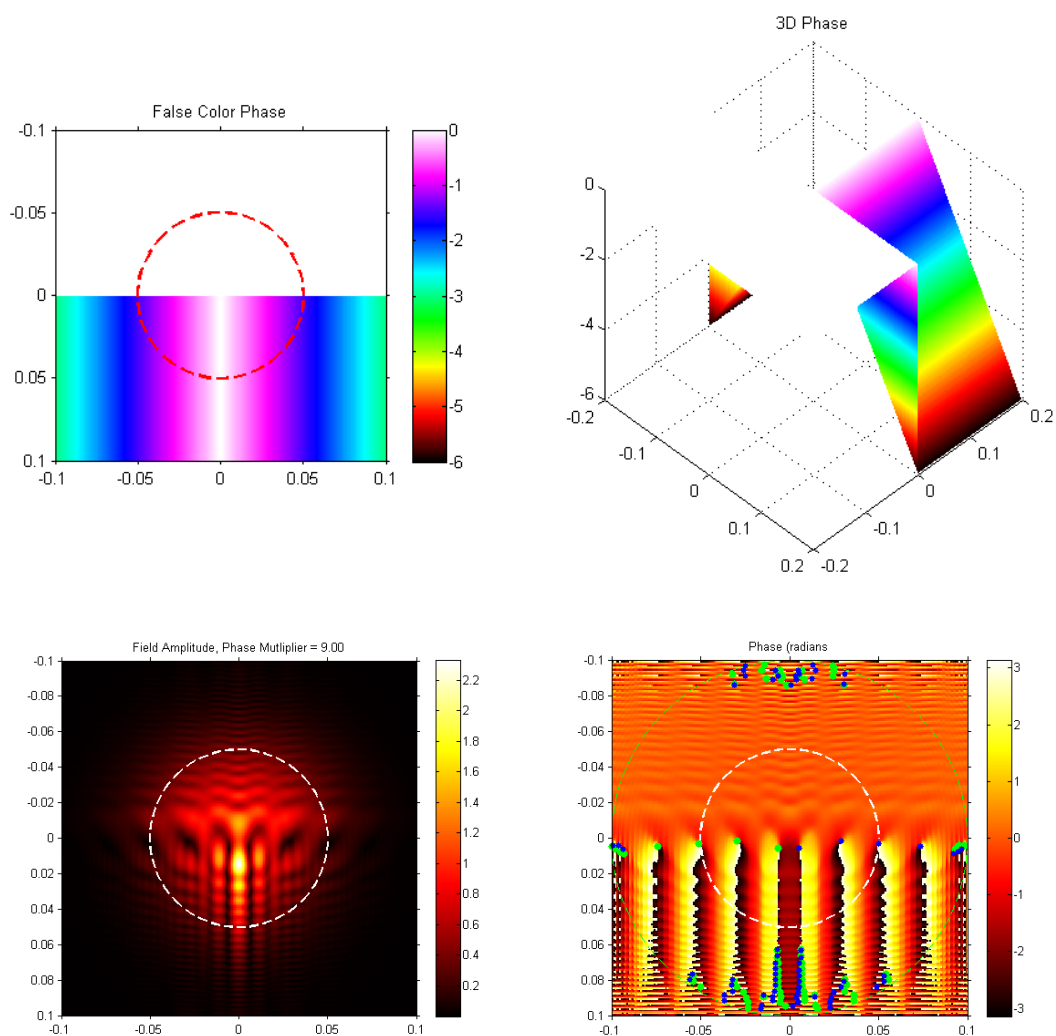


The 2D sinusoidal phase (grating) clearly produced branches.

4.3.6 Full Ramp

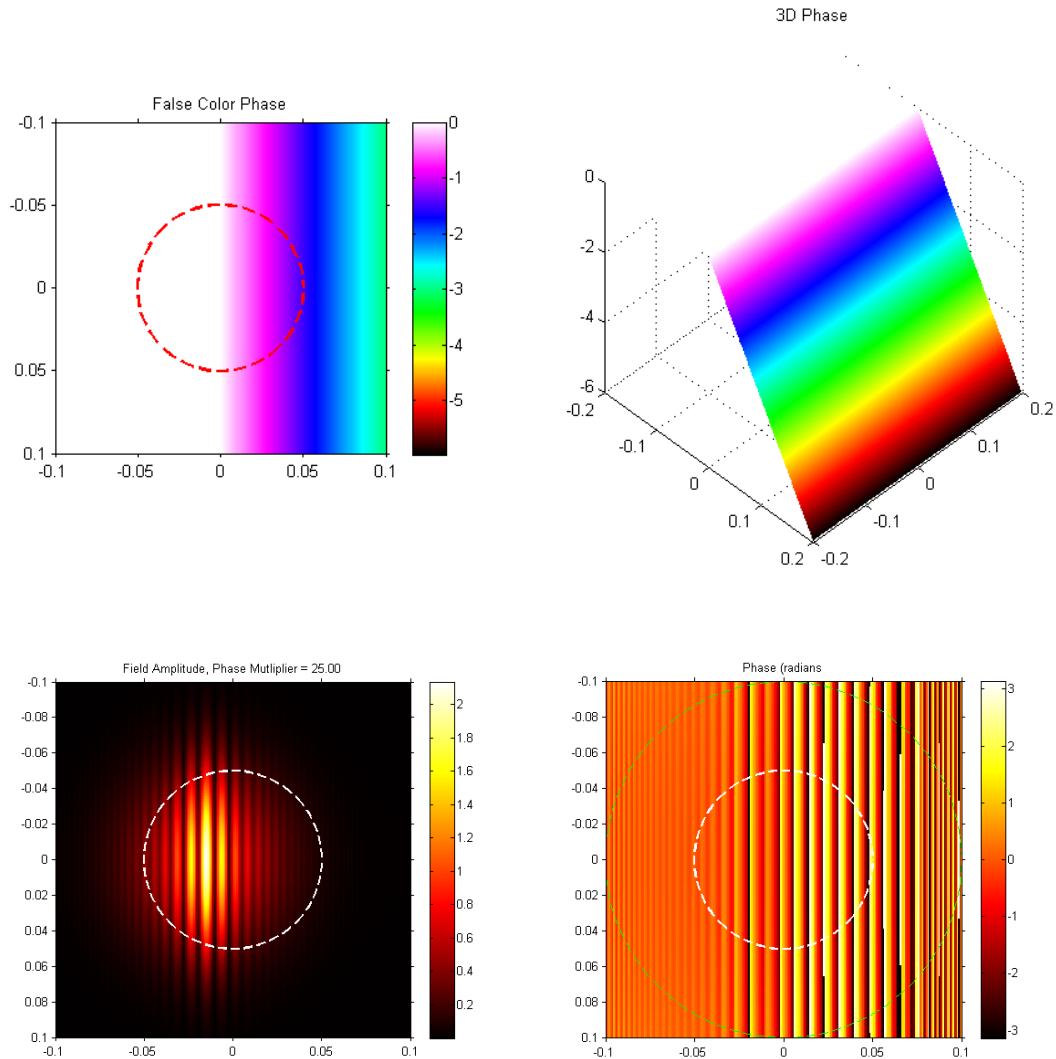


The full-ramp or pitched-tent phase profile produced a phase profile similar to the 1D grating, but did not show branch points. This is likely due to any branches being started outside the modeling aperture.

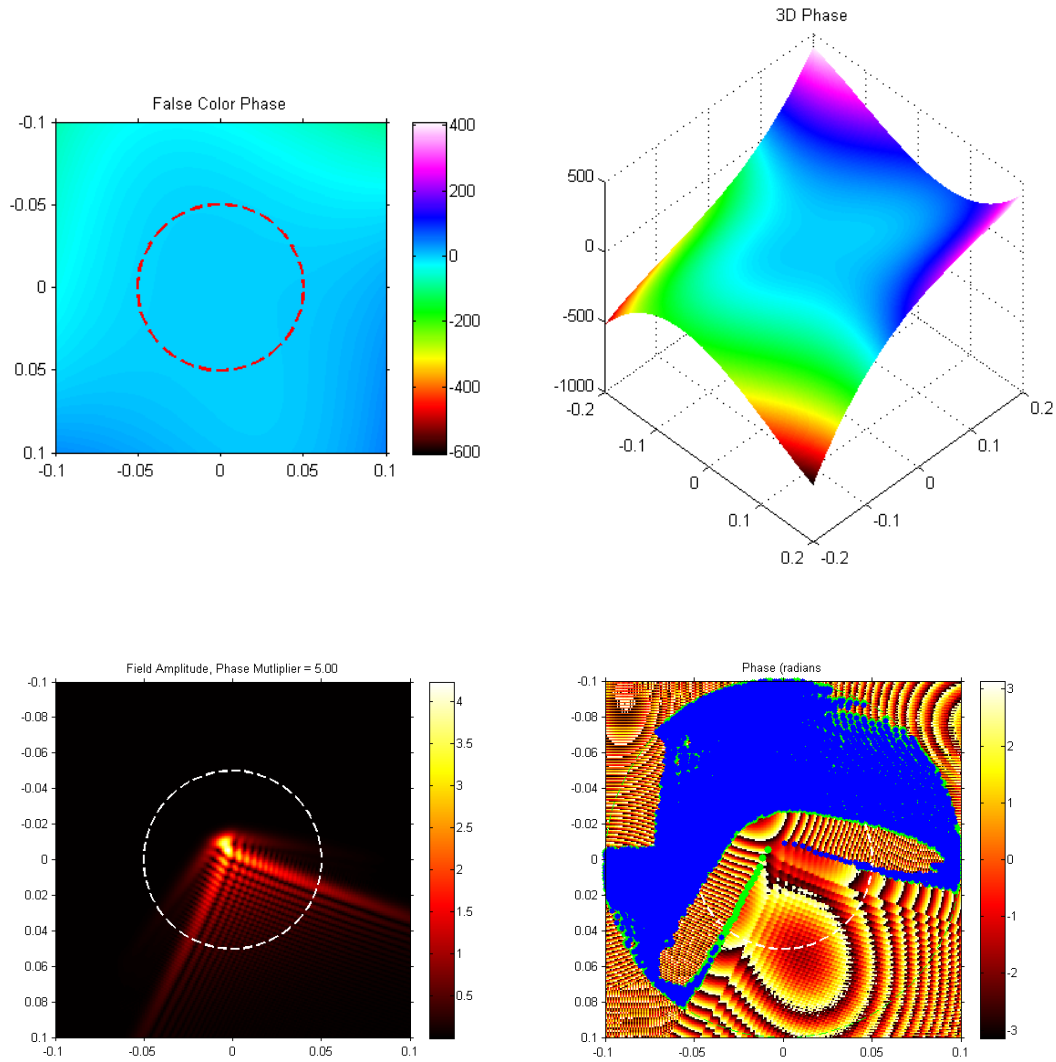


This half-ramp phase profile produced branches very similar to the half-pyramid phase. The clearest branch points were again at the discontinuity. The branch cuts followed the phase profile.

4.3.7 Half Ramp – Other Axis

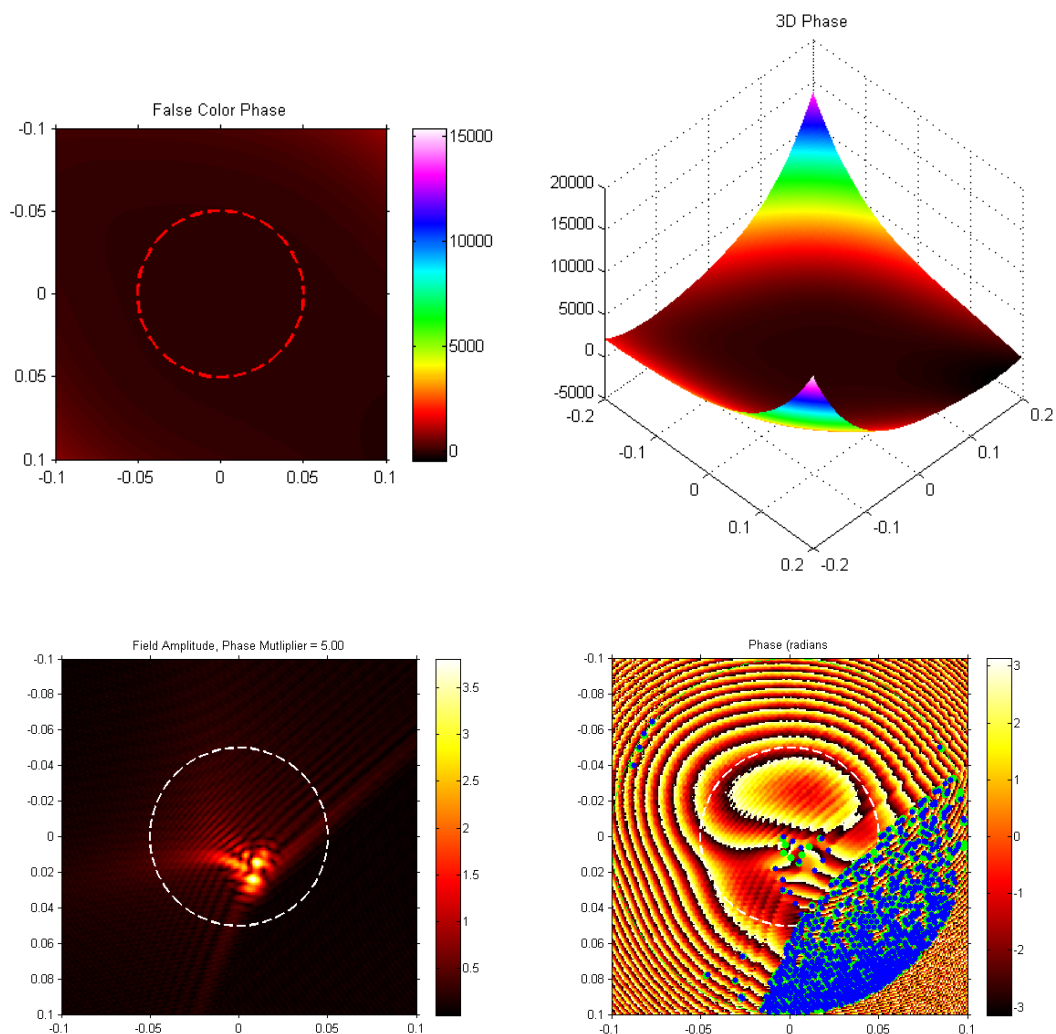


This half-ramp is similar to the pyramid phase profile in that the overall beam was broken into sub-beams and interfered. This produced intensity nulls that could correspond to branches, but it was unclear because the terminating branch points were outside the modeling aperture. Gauss Normalized Random Sum of 9 Zernikes



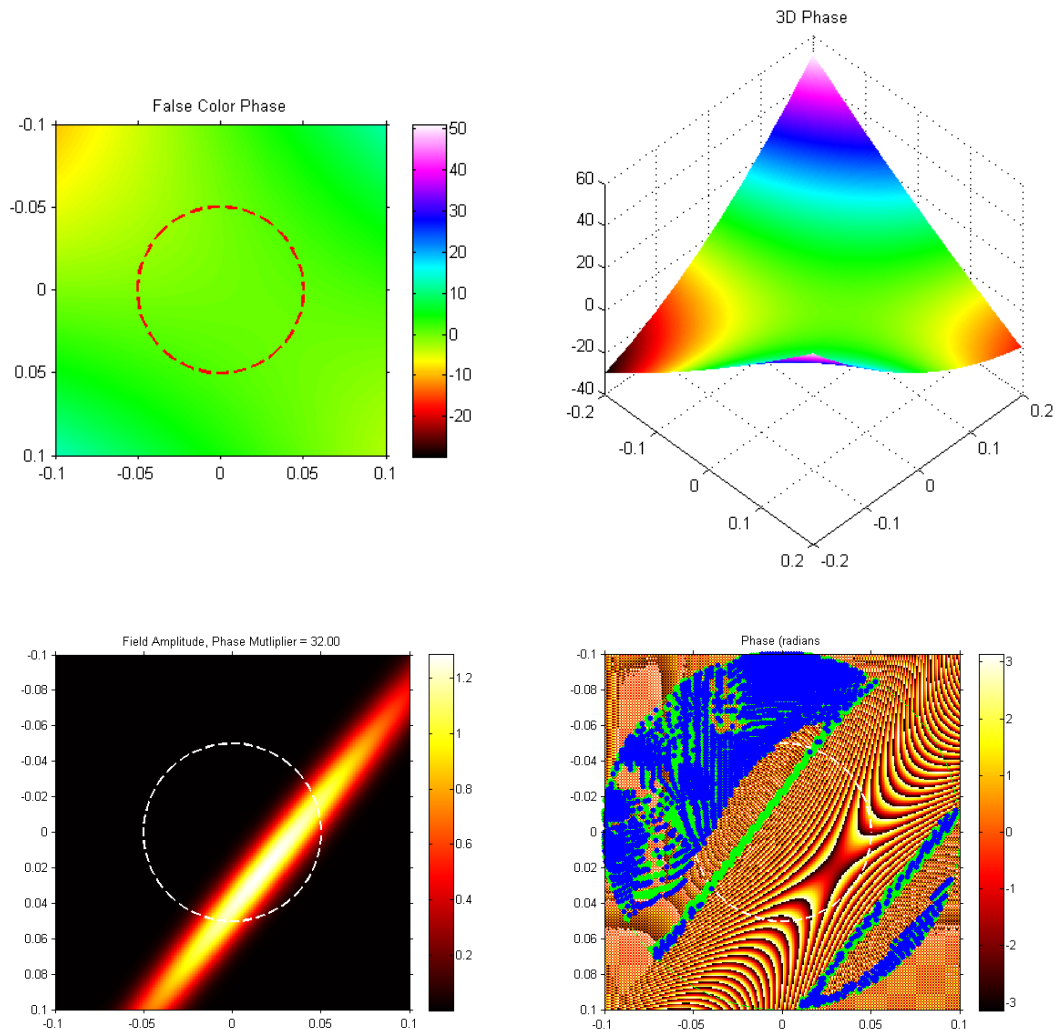
We ran a series of Zernike sums throughout this modeling. In this case we included the linear (x and y tilt), quadratic (astigmatism and focus), and the third-order terms (coma and trefoil). We were easily able to see branch points and cuts and the generation of intensity nulls.

4.3.9 Gauss Normalized Random Sum of 14 Zernikes



In this case, we included all the Zernikes up to 4th order. There is again clear evidence of intensity nulls and branch points and cuts.

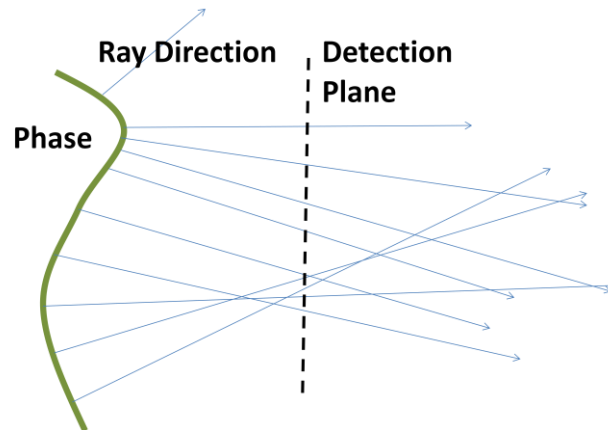
4.3.10 Gauss Normalized Random Sum of 5 Zernikes



We then moved on to see if only tilt, astigmatism, and focus could create a phase branch. Here is an example where this was observed. In this case, we did not see self-interference lobes of the beam like we did in examples with higher-order content, but we did see the formation of a very dark region and the phase ambiguity associated with that region. It is unclear whether this is a modeling artifact for this case because the regions with branch points shown in the phase were very dark and the beam did not break-up into lobes of intensity like we saw in other examples. I would conjecture from our modeling results that we are seeing a deep fade phase ambiguity, not the self-interference that would be associated with a branch.

4.4 Discussion of Modeling Results

In the modeling above, we showed that it was clearly possible to create a branch cut with a single continuous phase screen as long as the phase contained at least 3rd-order phase content. Based on these observations, we hypothesize that the branch points and cuts are likely generated by self-interference, like that shown in the figure below. This is consistent with Fried's description above, but goes back one step in the process to describe the nature of the generation of these intensity nulls (or deep fades). If a beam comprised of a set of rays is sufficiently bent to cause the rays to cross and self-interfere, the result will likely be a branch in the phase.



5 Conclusions

We have presented what we think is the simplest way of generating a branch in the phase using only a single continuous phase screen and a relatively short propagation. We would like to consider approaches for using a continuous phase sheet DM to create these branch points in a dual-DM AO system in the future.

6 Acknowledgements

This work was done with technical assistance from Dr. Donald J. Link from MZA's office in Jupiter, FL. Special thanks to Jeremy Bos for his input and comments as well.

7 References

-
- ¹ D. Fried and J. Vaughn, "Branch cuts in the phase function," Appl. Opt. 31, 2865–2882 (1992).
 - ² Todd M. Venema and Jason D. Schmidt, "Optical phase unwrapping in the presence of branch points", 12 May 2008 / Vol. 16, No. 10 / OPTICS EXPRESS p. 6985-6998. (<http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA504245>)
 - ³ D. Fried, "Branch point problem in adaptive optics," J. Opt. Soc. Am. A 15, 2759–2768 (1998).