Copyright © 1988, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

# TWO-DIMENSIONAL IMAGES WITH EFFECTS OF LENS ABERRATIONS IN OPTICAL LITHOGRAPHY

by

Kenny K. H. Toh

Memorandum No. UCB/ERL M88/30

20 May 1988

# TWO-DIMENSIONAL IMAGES WITH EFFECTS OF LENS ABERRATIONS IN OPTICAL LITHOGRAPHY

by

Kenny K. H. Toh

Memorandum No. UCB/ERL M88/30

20 May 1988

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

# TWO-DIMENSIONAL IMAGES WITH EFFECTS OF LENS ABERRATIONS IN OPTICAL LITHOGRAPHY

by

Kenny K. H. Toh

Memorandum No. UCB/ERL M88/30

20 May 1988

## ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# Two-Dimensional Images with Effects of Lens Aberrations in Optical Lithography

*Kenny K.H. Toh*

Department of Electrical Engineering and Computer Sciences,

University of California at Berkeley

*ABSTRACT*

A FORTRAN program associated with SAMPLE that simulates a two-dimensional optical image from a projection printer has been upgraded to include the effects of arbitrary lens aberrations. This program has been used to study general issues in projection printing, including the optical proximity effect and defect interactions with features. Basic studies of projection printed images are presented to identify the types of patterns which are most susceptible to residual lens aberrations and to establish test structures which may be used to monitor the presence of critical types of residuals. These effects are explored by including arbitrary lens optical path difference (OPD) aberration functions in the optical image simulation program. The lens aberration function is expressed either in Zernike polynomials or a series expansion. The intensity is calculated from Hopkin's transmission cross-coefficient formulation with a self-checking algorithm. A catalog of results is presented here for the dominant primary aberrations of coma and astigmatism for a fixed maximum OPD of 0.4 $\lambda$. Contact holes are shown to be much more susceptible to astigmatism than coma and the traditional checkerboard test pattern is verified as a sensitive diagnostic pattern. An alternative structure consisting of thin lines with a short break is shown to be even more sensitive to astigmatism and useful for distinguishing it from coma. A further improvement in sensitivity is obtained through the use of small nonprintable defect-like features in proximity to regular features which coherently interact with the blurred image of the feature. A test target of this type is recommended for monitoring coma.

May 12, 1988

Author Kenny K.H. Toh

Title Two-Dimensional Images with Effects of Lens Aberrations in Optical Lithography

RESEARCH PROJECT

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, in partial satisfaction of the requirements for the degree of Master of Science, Plan II.

Approval for the Report and Comprehensive Examination:

Committee: [signature], Research Adviser

May 12, 1988 Date

[signature]

May 9, 1988 Date

# Table of Contents

Chapter 1 : Introduction ..... 1

1.1. Overview of Optical Image Simulation ..... 1

1.2. Projection Lens Aberrations ..... 1

Chapter 2 : The Simulation Program ..... 3

2.1. Review of the Simulation Program ..... 3

2.2. Evaluating the TCC ..... 4

2.3. Program Verification and Capabilities ..... 5

Chapter 3 : General Applications of SPLAT ..... 7

3.1. Introduction ..... 7

3.2. Defects ..... 7

3.2.1. Transparent Square Defects ..... 8

3.2.2. Opaque Square Defects ..... 8

3.3. Defect Interaction with Features ..... 9

3.3.1. Transparent Defect Attached to an Isolated Feature ..... 9

3.3.2. Different Defect-Feature Configurations ..... 10

3.4. The Proximity Effect ..... 11

3.4.1. Tapers in Elbows ..... 11

3.4.2. Equally Spaced Lines ..... 11

3.5. Pattern Misalignment ..... 12

Chapter 4 : Identifying and Monitoring Effects of Lens Aberrations ..... 13

4.1. The Wave Aberration function $\Phi(f, g)$ ..... 13

4.2. Systematic Studies of Test Patterns ........ 13

4.3. Traditional Patterns ........ 14

4.4. Alternative Patterns ........ 16

4.5. Defect-Like Patterns ........ 16

4.6. The Role of Partial Coherence ........ 17

4.7. Defocus Amplification of Aberrations ........ 17

Conclusion ........ 19

Acknowledgement ........ 20

References ........ 21

Figures ........ 23

Appendices ........ 55

Appendix A - Some Mathematical Details ........ 56

Appendix B - SPLAT User's Guide ........ 61

# Chapter 1

# Introduction

## 1. Overview of Optical Image Simulation

As device dimensions are pushed deeper and deeper into the submicron regime, the task of ensuring good yield and throughput with optical lithography becomes more and more difficult. The optical simulation tool is almost indispensable as a method of exploring the limits of diffraction-limited photolithography. Optical simulation has been used to examine the interaction of light between closely spaced features, to design patterns optimized for best resolution, to study critical locations and sizes of defects, as well as to design patterns that can be used as visual or electrical test patterns for applications such as focus targets. Optical simulation is definitely a very valid tool in IC processing, since simulation runs are not only less expensive than running experiments, but also much faster.

This document continues the work done by the SAMPLE[1-8] simulation group at U.C. Berkeley, and presents some new results dealing with defects and optical proximity effects. The main emphasis of this document, however, is on projection lens aberrations, and the effect of these aberrations on image quality.

## 2. Projection Lens Aberrations

It is known that as device dimensions continue to shrink and the demand for greater image field information continues to grow, optical projection printing systems must hold tighter tolerances over larger image fields. As the image field increases, the aberrations from a single lens element grow rapidly, so minimizing the residual aberrations between many elements becomes increasingly difficult. The images of Pol et al[9] for a very early 248 nm lens in Figure 1 clearly show the effects of aberrations such as coma. While the lens manufacturers have sophisticated means of measuring residual aberrations, the impact of the residual aberrations on the aspects of image quality important in projection printing is still not well understood. It is also desirable to have a simple, independent means to verify that when the lens is incorporated in a projection printer it is still performing within acceptable tolerances.

The basic formulation of the aberration problem in terms of the circle polynomials of Zernike and the nature of the primary aberrations can be found in Born and Wolf[10] and Fincham and Freeman.[11] Lens manufacturers typically have their own programs for calculating effects of residual aberrations. One example of the effect of aberrations in projection printing is the work of Matsumoto et al.[12] The primary purpose of this document is to make an initial exploration of easily implemented diagnostic patterns and their sensitivity to various amounts of primary aberrations. To address these issues, the capability to include arbitrary lens aberrations has been added to a two-dimensional optical image simulation program associated with SAMPLE.[5]

# Chapter 2

# The Simulation Program

## 1. Review of the Simulation Program

SPLAT (Simulation of Projection Lens Aberrations via TCCs) is based on the Hopkins theory of partially coherent imaging,[13] which uses transmission cross-coefficients (TCCs) to calculate the image intensity pattern from the Fourier transform of the object (mask) transmission. Essentially, a projection printer can be characterized as having three basic components : an illumination system, a mask, and a lens to focus onto the image or Gaussian plane. As light in the form of a plane wave strikes the mask, it is diffracted, and as such, can be expressed as a simple sum of diffracted orders (assuming a periodic pattern for the mask). Each diffracted wave is then modulated by the lens before arriving at the image plane. Because of coherence effects, the mask-diffracted orders will also interact with each other. Thus, if the Fourier transform of the mask transmittance is given by $F(f,g)$, where $f$ and $g$ are spatial frequencies, the image intensity, according to Hopkins, can be expressed as a 4-fold integral in spatial frequency space:

$$I(x,y) = \iiiint TCC(f',g':f'',g'')\, F(f',g')\, F^*(f'',g'')\, e^{-2\pi i[(f'-f'')x+(g'-g'')y]} df'dg'df''dg'' \quad (2.1)$$

In the expression above, $x$ and $y$ are related to the geometrical coordinates of the mask/object, while the frequency pairs $(f',g')$ and $(f'',g'')$ represent two sets of diffracted plane waves that are interacting with some degree of coherence. $TCC(f',g':f'',g'')$ is known as a transmission cross-coefficient, and is the interaction between the light intensity (with a Fourier transform $J(f,g)$ ) and the lens. Following Hopkins, this transmission cross-coefficient can be written as follows :

$$TCC(f',g':f'',g'') = \iint J_0(f,g)\, K(f+f',g+g')\, K^*(f+f'',g+g'')\, df dg \quad (2.2)$$

Here, $J(f,g)$ represents the illumination cone, and for critical or Kohler illumination, is a constant within a radius proportional to $\sigma$, the partial coherence parameter of the system. $K(f,g)$ is the objective pupil function, and is given by

$$K(f,g) = e^{-i\frac{2\pi}{\lambda}\Phi(f,g)} \qquad f^2+g^2<1 \quad (2.3)$$

where $\Phi(f, g)$ is the wave aberration function which can be expressed as a simple power series in $f$ and $g$.

## 2. Evaluating the TCC

Given that the aberration function $\Phi(f, g)$ is known, it is possible to rewrite the expression for the TCC:

$$TCC(f', g' : f'', g'') = J_0 \iint e^{-i\frac{2\pi}{\lambda}\Phi(f+f', g+g')} e^{i\frac{2\pi}{\lambda}\Phi(f+f'', g+g'')} \, df \, dg \tag{2.4}$$

Evaluating the transmission cross-coefficient requires integrating the exponentiated phase variations due to the lens aberrations over the union of overlapping circles as shown in Figure 2. Here, the small circle represents the illumination cone and the large circles represent the acceptance cone of the imaging lens shifted by the wave vectors for the diffracted orders whose TCCs are being evaluated.

The evaluation of the TCC is carried out numerically using a two-dimensional adaptive quadrature method. First, the minimum ($f_{min}$) and maximum ($f_{max}$) limits of integration in the horizontal direction is found. The region between $f_{min}$ and $f_{max}$ is then divided into 16 vertical strips, according to a 16-point Gaussian quadrature. Each strip is then divided into 5 equally spaced intervals, and for each interval, the subinterval area $\delta f_i \delta g_j$ is multiplied by $\Phi$ evaluated at the midpoint of that particular subinterval. The total for each vertical strip is summed using both a 5-point Simpson and a 3-point Simpson, and compared.[14] If the difference is greater than a specified error tolerance, then the vertical strip is divided into smaller subintervals and reevaluated. The results for each vertical strip are added to obtain the final integral.

In summary, the adaptive quadrature method outlined above requires evaluating the aberration function $\Phi(f, g)$ at a minimum of 80 points within the region of integration, and has the redeeming feature that it is "self-checking", i.e. the computer is programmed to produce the approximation with optimal accuracy, thus relieving the user of the necessity of analyzing the accuracy of the result. Where the aberration function is "badly behaved", the program takes small step sizes, and where there is smooth behavior, the calculation is sped up with large steps. However, as might be expected, the minimum 80 function calls per TCC still does consume a considerable amount of computation time.

Because the number of TCCs that have to be evaluated is proportional to the mask area, even average-sized mask patterns with sizes on the order of 6λ/NA x 6 λ/NA require as many as 6 million function calls. Although the number of transmission cross-coefficients that have to be computed can be reduced by as much as a factor of 16 due to symmetry, the computation time needed for this program is still fairly large. Figure 3 shows a rough indication of the computing time on a Vax 11/780 versus mask size. The presence of arbitrary lens aberrations requires the use of the 2-dimensional integrating routines described earlier. In this case, the speed of computation depends on how many degrees of symmetry exist in the TCCs. In general, if there is only one lens aberration (such as coma only), there can be as many as 7 degrees of symmetry† in the TCCs, but combinations of aberrations (such as coma and defocus) will reduce or eliminate this symmetry and therefore increase the computation time. In comparison, the absence of aberrations will speed up the computation significantly. When only defocus is allowed,[8] the TCC integral still has to be evaluated numerically, but it can be reduced to a one-dimensional integral as described by Subramanian.[15] Simulations that do not require lens aberrations run the fastest, because analytical formulas can be used to evaluate the TCC.

### 3. Program verification and capabilities

Although as yet the results of this program have not been verified experimentally, there are three indications that this aberration program, SPLAT, is correct. First, for defocus only, the calculated intensity images agree to within 3 decimal places with the results from both 2D[8] (the original aerial image program) and SAMPLE.[5] The latter two programs use an entirely different one-dimensional qua-

---

† Symmetry here refers to the relationship between the TCCs for a given set of harmonics. The seven degrees of symmetry are :

$TCC(f',g':f'',g'') = TCC(-f',g':-f'',g'')$ *g-axis reflection*

$TCC(f',g':f'',g'') = TCC(f',-g':f'',-g'')$ *f-axis reflection*

$TCC(f',g':f'',g'') = TCC(-f',-g':-f'',-g'')$ *reflection around origin*

$TCC(f',g':f'',g'') = TCC(f'',g'':f',g')$ *commutative symmetry*

$TCC(f',g':f'',g'') = TCC(-f'',g'':-f',g')$ *g-axis reflection*

$TCC(f',g':f'',g'') = TCC(f'',-g'':f',-g')$ *f-axis reflection*

$TCC(f',g':f'',g'') = TCC(-f'',-g'':-f',-g')$ *reflection around origin*

Because the TCCs are complex variables, a check is also made to see if there is symmetry between the conjugates of the TCCs, e.g. $TCC(f',g':f'',g'') = TCC^*(-f',g':-f'',g'')$.

drature approach. Secondly, the TCC calculations for $\sigma = 1$ and varying degrees of third order aberrations agree well with the theoretical values tabulated by Born and Wolf.[10] And finally and perhaps most conclusively, the physical behavior of the images with aberrations, as computed by SPLAT, agrees with the description of Born & Wolf[10] : distortion moves the image position but does not affect its quality, coma moves and distorts the image and also leaves a small comet-like tail, astigmatism flattens the image, and so on.

Currently, SPLAT will accept values of coherence $0 < \sigma \leq 1.0$, and can compute the effects of any combination of the third order aberrations (spherical aberration, coma, astigmatism, curvature/defocus, and distortion). The mask patterns do have to be periodic, though, and execution is reasonably fast for masks with dimensions less or equal to $4\lambda/NA$ x $4\lambda/NA$. The simulated images produced can be represented either 2-dimensionally or 3-dimensionally (contour patterns); the 2-dimensional cuts can be taken along critical directions and then sent to SAMPLE for resist profile simulation. In this manner it is possible to see the impact of processing conditions and aberrations on the developed patterns. For simulation of images from large mask patterns, SPLAT takes much too long to execute, and so is impractical as a diagnostic tool for such masks. However, a solution has been implemented that saves all the non-zero TCCs calculated for a particular mask size. Since these transmission cross-coefficients depend only on the size of the field being simulated, the illumination and the lens of the imaging system, these coefficients can be reused for other patterns in the same field when the parameters mentioned are kept constant. Thus, additional pattern intensities can be calculated in less than 10% of the initial computation time.

# Chapter 3

## General Applications of SPLAT

### 1. Introduction

This two-dimensional optical imaging program, and its predecessor, 2D[8] , has, in the past, been used to study several important issues in projection printing, including defect interactions with features, and optical proximity effects. This section will discuss briefly some applications of this optical imaging program, and simulation results will be presented. For convenience and general applicability to different stepper models, all pattern sizes (except where stated otherwise) are specified in terms of the normalized parameter $\lambda$/NA.

### 2. Defects

The study of mask defects is useful as a starting point for understanding the mechanics of the interaction between defects and features. This is especially true for small defects which are normally hard to detect or to repair using conventional mask repair techniques. Figure 4 shows contour plots of the normalized intensity of the image caused by 0.4$\lambda$/NA transparent and opaque square defects. The important thing to note here is that the contours are circular in shape. This, of course, is due to diffraction - the 0.4$\lambda$/NA defect by itself is close to the resolution limit of the stepper, which is approximately 0.3$\lambda$/NA for partially coherent imaging.† Thus, if a defect has dimensions smaller than the resolution limit of the stepper, the intensity coming through the defect will be independent of the shape of the defect and will be proportional to the area of the defect. This is shown in Figure 5, where defects of different shapes but equal areas are simulated and their corresponding intensity contours are plotted. It can readily be seen that the contours look similar and are roughly circular for different shapes. A closer examination also shows that both the bar-bell and the rectangle have elliptical contours. This shape-dependence comes about when any of the defects' dimensions exceeds the resolution limit. So, in general, if the dimensions of the defect remain close to the resolution of the stepper, the

† The resolution limit is defined as the point where the contrast of an imaged pattern is zero. The resolution limit is 0.5$\lambda$/NA for fully coherent imaging, 0.25$\lambda$/NA for incoherent imaging, and between 0.25 and 0.5$\lambda$/NA for partially coherent imaging.

intensity of the defect will be roughly independent of shape.

### 2.1. Transparent Square Defects

The next step towards understanding defects is through the study of square transparent defects. It can be shown that if the dimensions of a defect are less than the wavelength of the illumination, the electric field transmitted through a transparent square defect will be proportional to the square of the defect length, i.e. $E_t \propto D^2$, where $E_t$ is the electric field transmitted through the transparent defect, and $D$ is the length of a side of the square defect. Since intensity $I = |E|^2$, the peak intensity through a small defect will be proportional to $D^4$. Figure 6 is a logarithmic plot of the peak intensity $I$ vs. the defect size $D$, obtained from simulations of transparent defects in opaque masks. This plot shows that as long as the defect size $D \leq 0.4\lambda/\mathrm{NA}$, the log-slope is approximately constant and has a value of approximately 4.0, which implies that $I_t \propto D^4$. From this set of simulations, it is also possible to pin down the exact relationship, and thus to obtain the following set of equations.

$$E_t = 2.9D^2, \qquad D \leq 0.4\lambda/\mathrm{NA} \tag{3.1}$$

$$I_t = |E_t|^2 = 8.5D^4, \qquad D \leq 0.4\lambda/\mathrm{NA} \tag{3.2}$$

This relationship is also independent of the partial coherence $\sigma$, as shown in Figure 7, where $I_t$ vs $D$ is plotted logarithmically for $\sigma = 0.3$ and $\sigma = 0.7$. It can clearly be seen that the slopes of the two curves are equal for defect sizes $D < 0.4\lambda/\mathrm{NA}$. The reason for this is that the defect size is small enough for it to behave like a point source, so the light coming through the defect will be fully coherent.

### 2.2. Opaque Square Defects

Using the equations above, it is possible to develop an analogous set of equations for the case of opaque square defect in transparent masks. The electric field blocked by the square defect, $E_o$, is related to the electric field passing through the transparent square defect, $E_t$, by the simple relationship

$$E_o = 1 - E_t \tag{3.3}$$

Since $E_t = 2.9D^2$ from equation (3.1),

$$E_o = 1 - 2.9D^2, \qquad D \leq 0.4\lambda/\mathrm{NA} \tag{3.4}$$

$$I_o = |E_o|^2 = \left[1 - 2.9D^2\right]^2 \qquad D \le 0.4\lambda/\text{NA} \tag{3.5}$$

Equations (3.2) and (3.5) are plotted linearly in Figure 8, together with the data from the simulations. It is evident that the above equations break down for defect sizes $D > 0.4\lambda/\text{NA}$. This is to be expected, since for these large defect sizes, the defect dimensions will be of the same magnitude as the illumination wavelength, so the light coming through or being blocked by the defect will no longer be fully coherent.

## 3. Defect Interaction with Features

From Figures 6-8, it can be seen that if a transparent defect in an opaque mask is small, with the defect size $D \le 0.4\lambda/\text{NA}$, then the peak intensity due to the defect alone will be less than 20% of the clear field intensity. Photoresists exposed and developed at zero bias will normally print at the 30% intensity level, so the small defect by itself will not print. However, when the defect is placed close to a feature, the light coming through the defect will interact coherently with the light from the feature. This interaction is known as the optical proximity effect, and as can be seen in Figure 9[7] , results in a significant linewidth variation. The linewidth change due to the interaction between the defect and the feature has been studied extensively with simulations.[2,3,6,7] Some of the key results are summarized below, for the case of a transparent defect attached to an isolated transparent feature.

### 3.1. Transparent Defect Attached to an Isolated Feature

When a small transparent defect is close to an isolated transparent feature, the interaction between the defect and the feature is nearly coherent and the defect slightly increases the intensity at the edge of the feature. At an intensity threshold of 30% (which corresponds roughly to the intensity at the line edge), the linewidth change in $\lambda/\text{NA}$ can be found by dividing the intensity change by the intensity slope at the line edge as [3]

$$\Delta L \approx \frac{\Delta I}{dI/dL} \approx \frac{I_c - I_f}{2.9 - 1.3\sigma} = \frac{2\mu_{eff}\sqrt{I_f}\sqrt{I_d} + I_d}{2.9 - 1.3\sigma} \tag{3.6}$$

where $I_c$ is the composite intensity, $I_f$ is the feature intensity, $I_d$ is the intensity due to the defect alone, and $\mu_{eff}$ is a dimensionless parameter describing the degree of coherent interaction between the

--

defect and the feature. For small defects the defect intensity $I_d \approx 8.5D^4$, (where $D$ is the defect size in λ/NA) so $\Delta L \; \alpha \; D^2$, and the linewidth change is a parabolic function of the defect size $D$. Evaluating Equation (3.6) with $\sigma = 0.45$, $\mu_{eff} = 1$ (purely coherent interaction for small attached defects), $I_f = 0.30$ and ignoring the $I_d$ term (defect intensity term << defect-feature interaction term) yields

$$\Delta L \approx 0.47\sqrt{I_d} = 0.47\sqrt{8.5D^4} = 1.4D^2 \tag{3.7}$$

For larger defects however, the above perturbational approach must be replaced by more rigorous image simulation. This is partly because the slope $dI/dL$ is no longer constant and the defect-feature interaction is no longer fully coherent ($\mu_{eff} < 1$). More importantly, for defect sizes $D > 0.4$ λ/NA, the defect peak intensity is greater that 30%, so the 30% intensity threshold of the composite pattern will move away from the feature and be pinned on the 30% threshold of the defect. In effect, the linewidth change is now determined by the defect size, as the defect itself is large enough to be classified as a feature. This means that the linewidth change becomes proportional to defect size. The following composite algebraic model has been found to apply for both large and small defects near a single feature.

$$\Delta L = \begin{cases} 1.4D^2 & D \le 0.4\ \lambda/\mathrm{NA} \\ 1.5D - 0.42 & 0.4\lambda/\mathrm{NA} < D < 0.8\ \lambda/\mathrm{NA} \end{cases} \tag{3.8}$$

The equation above has been compared to intensity threshold (30%) simulations as well as to rigorous resist simulations through SAMPLE, and the results are plotted in Figure 9. The image intensity simulation results agree with the more rigorous resist profile simulation results, which indicates that in this case, the resist appears to be acting as a threshold detector at 30% of the clear field intensity. The simple algebraic model from Equation (3.8) also shows an excellent fit to both the threshold and resist simulations.

### 3.2. Different Defect-Feature Configurations

Modelling the interaction between defects and features in different configurations is more difficult than modelling the interaction between a transparent defect attached to a transparent isolated feature, primarily because the coherent interaction between the defect and its surrounding neighbor or neighbors becomes much more difficult to characterize. However, the basic principle remains the same, even if

--

the defect is placed in between two features or if the mask polarity is reversed. Systematic simulations can and have been used [2,3,6,7] to characterize the defect-feature interaction, and thus, an understanding can be gained as to what defect dimensions or locations might cause critical damage to the developed pattern.

## 4. The Proximity Effect

The interaction of defects with features is but one aspect of the proximity effect. The proximity effect, as mentioned earlier, takes place when two features are close enough to each other that the light through both features will interact with some degree of coherence. In general, if the spatial coherence of the stepper is increased (lower $\sigma$), the optical proximity effect will be more severe. A few of the patterns in which the proximity effect occurs are discussed below.

### 4.1. Tapers in Elbows

The two-dimensional imaging program is especially well suited for examining two-dimensional structures such as elbows. A contour plot of the image intensity from a pair of $0.8\lambda/NA$ elbows is shown in Figure 11. This image was computed at $\sigma = 0.3$, and it can clearly be seen that the high partial coherence in the system causes quite severe ringing in the image, especially at the corners of the elbows. The linewidth variation at the corners might also be too large to be tolerated by certain processes. There are several methods of reducing the ringing or the linewidth variations in elbows. The most obvious is to reduce the coherence of the system, i.e., increase the $\sigma$. The effect of this is shown in Figure 12. However, a reduction in the coherence of the system will also result in a lower image slope and thus cause a small loss in resolution. A more elegant solution is to "smooth" out the sharp 90° bends in the elbows by introducing tapers at the corners of the elbows. Figure 13 uses 45° triangular tapers and an improvement can clearly be observed in the contour smoothness. This illustrates the ease with which simulation can be used to reduce or repair the optical proximity effect.

### 4.2. Equally Spaced Lines†

† Here, a line is defined to be a line in resist; thus, if positive photoresist is used, there will not be any resist where the resist is exposed with a transparent feature in an opaque mask. So, the transparent feature is defined as a space (because of the opening in resist). Similarly, a opaque feature in a transparent line will produce a positive photoresist line,

Equally spaced lines with a single protruding line/space is a pattern that is sometimes used in visual tests for optical lithography. Figure 14 shows the results of the simulation of transparent 0.6λ/NA spaces in an opaque mask, with $\sigma = 0.7$. The width of the protruding space, judging by the 10% contours, is slightly greater than the width of the neighboring spaces. This, of course, is another manifestation of the optical proximity effect. The equally spaced features will interact to produce a higher intensity minimum, which will in turn cause the low intensity contours to move closer together.

The proximity effect is more pronounced in Figure 15, which is the same mask used in Figure 14 but with the polarity reversed. The protruding line, if developed at the 40% intensity contour, will be wider at the protruding end than at the center of the equal line-space array, because less light passes through the line-space array.

## 5. Pattern Misalignment

Two-dimensional optical lithography simulation can also be applied to pattern misalignment problems, specifically, the problems that occur when two stepped image fields are misaligned (e.g. in electron-beam mask-making). As might be expected, when two features are "pinched" together (butting error), the two features will be resolved as a single feature with a slight bulge where the features meet. This is illustrated in Figure 16, where the simulation was run with $\sigma = 0.5$, and a 0.2λ/NA overlap between 0.8λ/NA features. The situation however turns critical when the two features are separated from one another. Simulations show that if the distance between the two features is less than 0.2λ/NA (Figure 17), then the problem is tolerable; however, when the separation increases to 0.4λ/NA (Figure 18), a zero-bias process (30% intensity contour) would clearly result in a broken resist line.

---

so that feature is now a line.

# Chapter 4

# Identifying and Monitoring Effects of Projection Lens Aberrations

## 1. The wave aberration function $\Phi(f,g)$

From optics, it is known that if the aperture of an optical system is appreciable, the whole of the light from a given object point does not reunite at a common focus after refraction by a single lens. In practice, images suffer from aberrations or defects - however, by combining a number of lenses made of different glasses and of differing thicknesses, it is possible to eliminate or reduce these aberrations. For such "corrected" systems (which include projection printer lenses), the residual aberrations are typically expressed in terms of wavefront aberrations, where the the optical path difference (OPD) between the real wavefront and a perfect spherical wavefront is calculated. [11] In general, it is possible to quantify any type of projection lens aberration (with the exception of chromatic aberration) by expanding its wavefront aberration in a power series as:

$$\Phi(f,g) = \sum_{l,m,n} C_{lmn} (x^2+y^2)^l \, (xf+yg)^m \, (f^2+g^2)^n \qquad (4.1)$$

In the equation above, $(x,y)$ refer to the object locations in the field, relative to the lens center, while $(f,g)$ are the normalized polar pupil coordinates, $l,m$ and $n$ are three integers that describe the order of the aberrations (3rd order, 5th order, etc), and $C_{lmn}$ is a constant that determines the magnitude of the aberrations. In these studies the constant $C_{lmn}$ and the object coordinates $(x,y)$ were used as input parameters, The third-order aberrations, also known as Siedel or primary aberrations, typically have the greatest influence on images. There are five such aberrations: Spherical Aberration [$l$=0, $m$=0, $n$=2], Coma [$l$=0, $m$=1, $n$=1], Astigmatism [$l$=0, $m$=2, $n$=0], Curvature [$l$=1, $m$=0, $n$=1], and Distortion [$l$=1, $m$=1, $n$=0]. Once $\Phi(f,g)$ has been completely defined, it is inserted into the TCC equation, (2.4), and the TCC computations can be carried out.

## 2. Systematic Studies of Test Patterns

This document is primarily concerned with image quality and the extent to which the above primary aberrations play a role. Fortunately, distortion and curvature can quickly be eliminated. Distortion and even coma produce a lateral shift of the image but this placement error does not affect the

profile shape and will not be considered. Curvature basically describes a focus change as a function of position in the field, which more properly belongs in a discussion of the focus budget. Spherical aberration can have very pronounced effects but to first order, the effect can be mediated by a change in focus. Thus, the two remaining primary aberrations which dominate image quality are coma and astigmatism. Both of these effects are zero in the center of the field for a centered lens and increase with field height. These effects are also radial in nature so that the direction of the influence will also change with position in the field. It is interesting to note that astigmatism has a symmetrical OPD function while coma has an asymmetrical OPD function.

A variety of device features and test patterns have been used to characterize the effects of aberrations. Figure 19 shows the set of patterns used in this systematic study. These patterns range from contact holes and traditional checkerboards to alternative thin line and defect-like structures. A rigorous determination of the resist profile from these patterns would require extensive simulation of the dissolution process in three dimensions. Instead a simpler threshold interpretation of the intensity contour plots for images of these structures will be used to get a reasonable indication of the effects of aberrations. The intensity level which approximately corresponds to the developed opening at the substrate is from 30% to 10% depending on the amount of overexposure and overdevelopment used to gain linewidth control. Throughout this study the discussion will refer to the 20% contour.

**3. Traditional Patterns**

The presence of aberrations is often recognized by the fact that contact holes are distorted from a shape with 90 degree rotational symmetry. The contour plots of the image intensity for an isolated square under the influence of astigmatism and coma are shown in Figure 20 and 21 respectively. The contours are for 10% levels normalized to a clear field intensity of unity. The four positions correspond to the center, right hand edge, top edge and 45 degree top-right edge of a circular lens field. For the three edge positions the maximum value of the OPD is $0.4\lambda$.

Coma tends to produce an "ice cream cone" effect and movement along a radial line. The small amount of coma in Figure 20 produces a build-up of the intensity on the side of the feature away from the center of the field. For astigmatism, there is no movement and spreading is produced both inward

and outward along a radial line. This results in a double ended symmetrical "football" shape pointing in the radial direction which is the direction in which the maximum OPD occurs. For the same maximum OPD of 0.4λ astigmatism is much more pronounced than coma as can be seen by comparing the 20% (second outermost) contour in Figures 20 and 21. Defocus would preserve the 90 degree rotational symmetry and would tend to make the contour larger and more circular. Spherical aberration would produce an effect with the same symmetries as defocus and is partially compensated by and difficult to distinguish from defocus.

Small checkerboard patterns are known to be sensitive to aberrations and are frequently used in diagnostics. Examples of the effects of coma and astigmatism on a checkerboard pattern are shown in Figure 22 and 23 respectively. Here the amount of the aberration and the positions in the field are the same as those for Figures 20 and 21. The distortion of the contours of the checkerboard pattern is more pronounced than it is for a simple contact. The movement of the high intensity away from the center of the field is very evident with coma. However, the distortion of the 20% contour is not particularly large. Astigmatism, on the other hand, results in a very major stretching of the 20% contour along the diagonal direction for the 45 degree field position. Therefore, the checkerboard pattern is most sensitive for astigmatism in the diagonal direction.

The checkerboard patterns have been used as visual test patterns for the detection of aberrations on a GCA 6200 Wafer Stepper in the Electronics Research Laboratory MicroLab [Univ. of California, Berkeley], and a SEM of a 1.6 μm checkerboard pattern in 1.1 μm of Kodak 820 photoresist is shown in Figure 24. An image simulation, run at $\sigma = 0.7$ and 0.2λ of astigmatism along the (0.7,0.7) diagonal is shown alongside. It is easy to see the similarity between the SEM and the simulation. Surprisingly though, this SEM was taken at best focus and in the center of the field. This indicates that this particular GCA Stepper has some degree of astigmatism, and since astigmatism will not be present in a properly centered lens, it can also be concluded that the lens in this stepper is not centered correctly.

Unfortunately, relating simulations and interpreted SEMs to the actual lens condition is difficult if not impossible. This is because an actual projection lens might consist of as many as 15 separate elements, all of which will contribute to the overall aberration of the lens. In addition, higher order aber-

rations are often introduced deliberately to compensate for the dominant third order aberrations. Traditional lens characterization data consists of up to 30 Zernike polynomial coefficients, i.e,, the power series in Equation (4.1) might have up to 30 separate terms. However, it is still useful to develop targets such as the checkerboards discussed above, which can identify the presence and effective degree of third order aberrations. This, of course, is one of the more useful applications of an image simulation program.

### 4. Alternative Patterns

Given the basic tendencies of coma and astigmatism to blur images in particular ways it is not difficult to develop alternative test structures which are even more sensitive to their presence. Since astigmatism extends the image both inward and outward along the radial direction, small gaps in radial lines should be very sensitive to astigmatism. The equally spaced lines test structure shown in Figure 19 is a structure in this class. In this case the gaps are formed by the implicit periodic extension in the vertical direction. The effect of coma and astigmatism on this structure are shown in Figures 25 and 26 respectively. When the lines are radial (0,1) astigmatism fills in to form a continuous line at the 20% contour. When the lines are tangential (1,0) a large line with width equal to the original line length is formed. Coma tends to pile up the intensity at the top of the lines when they are radial and does not fill in between lines when they are horizontal. Thus the two effects can be distinguished.

### 5. Defect-Like Patterns

For optical projection printing the variation in intensity due to a defect in proximity to a feature is far greater than the image intensity produced by the same defect when it is isolated.[7] For example, an isolated pin hole which would only produce a 3% intensity by itself will produce a 17% intensity increase when it is adjacent to a feature. This large impact is due to a nearly coherent interaction of the electric fields of the defect with those of the feature. The existence of this highly coherent interaction effect can be used to develop test structures which are extremely sensitive to aberrations. The basic idea for this application is to place small nonprinting features near a feature in the directions in which the aberration tends to blur the feature.

--

An example of a defect-like test structure for detecting coma is shown in Figure 19. The resulting images with coma and astigmatism are shown in Figures 27 and 28 respectively. For the right edge field position (1,0) where coma tends to blur the feature to the right the interaction with the defect on the right hand side is much stronger than on the left hand side giving rise to a very asymmetrical image. Astigmatism, however, always gives a symmetrical image even though defects are present. When coma is in the vertical direction (0,1) the pattern is again symmetrical. Insight as to how the defect is interacting with the feature can be gained from Figure 29 which shows the effect of coma on the aerial image for a set of parallel lines. Here the image position movement by coma has been corrected by including distortion. The strong interaction with the defect on the right hand side is produced by the rise in the aerial intensity just outside the mask edge.

### 6. The Role of Partial Coherence

The images considered thus far have been for a partial coherence $\sigma$ of 0.3. This relatively high degree of coherence tends to emphasize the effects of aberrations and is thus recommended for lens diagnostic work. A larger value of $\sigma$ reduces these effects as shown in Figure 30. Thus knowledge of the partial coherence parameter is needed for quantitative determination of the amount of aberration. In production, less coherent illumination is commonly used not only to reduce effects of aberrations but also to increase the throughput and reduce optical proximity effects.

### 7. Defocus Amplification of Aberrations

The majority of the simulations thus far were run using a maximum OPD of $0.4\lambda$, and used only one aberration† at any one time. This optical path difference of $0.4\lambda$ is actually fairly large, and lens manufacturers claim that the projection len aberrations in a typical lens can be beat down to an OPD less than $0.1\lambda$. At best focus, however, it is almost impossible to see much less measure aberrations with such small OPDs. However, it is still possible to detect these aberrations by adding in a healthy amount of defocus. Defocus can be considered a lens aberration, and is defined as the distance of the

† This condition, which assumes a single dominant aberration, makes it easier to analyze the effects of coma or astigmatism alone, but is sometimes not realistic, since a lens could have a combination of aberrations none of which are dominant.

focal plane from the wafer plane. The presence of defocus adds a parabolic term to the OPD function, distorts the image even more, and so "amplifies" the lens aberration.

To illustrate this, a set of simulations were run for a Siemens Star pattern at $\sigma = 0.5$, maximum aberration OPDs of $0.2\lambda$, and defocus of 0 R.U., 1 R.U. and 2 R.U..† These 3 sets of defocus simulations are displayed in Figure 31. It is evident from the 0 R.U. Defocus simulations that the aberrations are virtually undetectable, as all 3 of the contour plots in the left column look similar. The picture changes somewhat for a programmed defocus of 1 R.U., and now there is noticeable asymmetry in the Coma simulation, as well as "footballing" for Astigmatism. The contrast between the three sets of simulations is even greater at 2 R.U. of defocus. Thus, the simulations show that it might be possible to quantitatively determine what dominant aberrations there might be in a stepper, simply by running a focus matrix and comparing the SEMs with simulations such as these. Preliminary studies have been made on a GCA Stepper which look promising, but there is still insufficient experimental data at this point.

† 1 Rayleigh Unit = $0.5\ \lambda/NA^2$

## Conclusion

A FORTRAN program that simulates a two-dimensional optical image from a projection printer has been upgraded to include the effects of arbitrary lens aberrations. This program, named SPLAT, has been used to study general issues in projection printing such as the optical proximity effect and defect interactions with features. A basic exploratory study of the effects of residual primary aberrations and test structures for monitoring their presence has also been made. This has been carried out by including an adaptive quadrature algorithm to integrate the phase effects due to the optical path difference (OPD) function.

The inclusion of a single primary aberration such as coma or astigmatism increases the computation time by about a factor of 20 over the original program, which had an algorithm for defocus only. Combinations of primary aberrations and general Zernike expansions of the OPD will increase the computation time by an additional factor of 8. Effects of the two dominant primary aberrations, coma and astigmatism, were evaluated for a maximum OPD of $0.4\lambda$. At this level astigmatism gives contact holes a "football" like shape. This degree of astigmatism can easily be found from diagonal effects in checkerboard patterns or from even more sensitive narrow gaps in thin lines. Diagnostic structures based on the coherence of interaction of nonprintable defect-like features appear to be the most sensitive for aberrations in general. This study is far from exhaustive but gives a good indication of the kinds of structures and techniques which might be used to quantitatively assess the the amount of coma and astigmatism present in an optical system.

## Acknowledgement

The author is grateful to Victor Pol for providing the optical micrograph in Figure 1. I would like to thank J. Bruning of GCA and A. Offner, F. Zernike of Perkin-Elmer, and W. Haller of UC Berkeley for stimulating discussions on this work. Thanks also to Prof. Andrew Neureuther for his help and encouragement throughout my stay here at UC Berkeley.

This work was supported by the National Science Foundation under grant ECS-8420688, and by the Industrial-SRC Consortium on Deep UV Lithography and the California State MICRO program under grants SRC 87-MP118 and MICRO 87-121.

## References


1. K.K.H. Toh and A. R. Neureuther, "Identifying and Monitoring Effects of Lens Aberrations in Projection Printing," *Proceedings of SPIE : Optical Microlithography VI*, vol. 772, March 1987.

2. V. Mastromarco, A.R. Neureuther, K.K.H. Toh, "Printability of Defects in Optical Lithography : Polarity and Critical Location Effects," *J. Vac. Sci. Technol. B.*, Jan/Feb 1988.

3. K.K.H. Toh, C.C. Fu, K.L. Zollinger, A.R. Neureuther, R.F.W. Pease, "Characterization of Voting Suppression of Optical Defects Through Simulation," *Proceedings of SPIE : Optical/Laser Microlithography*, March 1988.

4. B. Huynh, K.K.H. Toh, W.E. Haller, A.R. Neureuther, "Optical Printability of Defects in Two-Dimensional Patterns," *Electron, Ion, Photon Beams Conference*, June 1988. Accepted for Publication.

5. W. G. Oldham, S. N. Nandgaonkar, A. R. Neureuther, and M. M. O'Toole, "A General Simulator for VLSI Lithography and Etching Processes: Part I - Application to Projection Lithography," *IEEE Trans. on Electron Devices*, vol. ED-26, no. 4, pp. 717-722, April 1979.

6. P.D. Flanner III, S. Subramanian, A.R. Neureuther, "Two-Dimensional Optical Proximity Effects," *SPIE*, vol. 633, pp. 239-244, 1986.

7. A.R. Neureuther, P. Flanner III and S. Shen, "Coherence of Defect Interactions with Features in Optical Imaging," *J. Vac. Sci. Technol. B.*, pp. 308-312, Jan/Feb 1987.

8. P.D. Flanner III, Two-dimensional Optical Imaging for Photolithography Simulation, University of California, Berkeley, May 12, 1986. M.S. Thesis

9. V. Pol, J.H. Bennewitz, G.C. Escher, M. Feldman, V.A. Firtion, T.E. Jewell, B.E. Wilcomb, and J.T. Clemens, "Excimer Laser-Based Lithography: a Deep Ultraviolet Wafer Stepper," *Proceedings of SPIE*, vol. 633, pp. 6-16, 1986.

10. M. Born and E. Wolf, *Principles of Optics, Sixth Edition*, Pergamon Press, London, 1980. Chapters 5, 9 and 10.

11. W.H.A. Fincham & M.H. Freeman, *Optics, 9th Edition*, pp. 380-408, Butterworths, London, 1980.

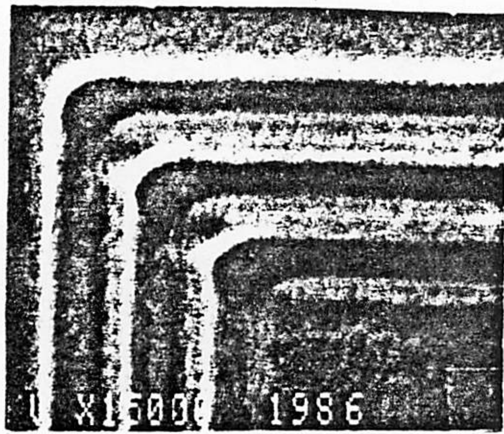
12. K. Matsumoto, K. Konno, K. Ushida, "Development and Application of Photolithography Simulation Program for Step-and-Repeat Projection System," *Kodak Microelectronics Seminar*, pp. 74-79, 1983.

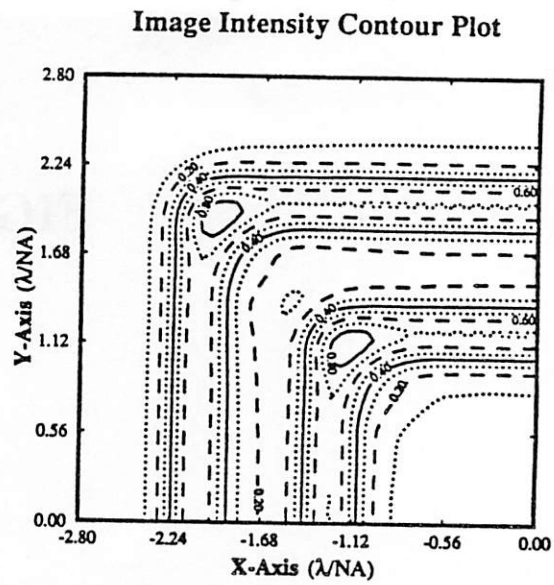
13. H.H. Hopkins, "On the diffraction theory of optical images," *Proc. Royal Soc. Series A.*, vol. 217, no. 1131, pp. 408-432, 1953.

14. L.W. Johnson, R.D. Riess, *Numerical Analysis, Second Edition*, pp. 313-318, Addison-Wesley Publishing Company, Reading, Massachusetts, 1982. Chapter 6

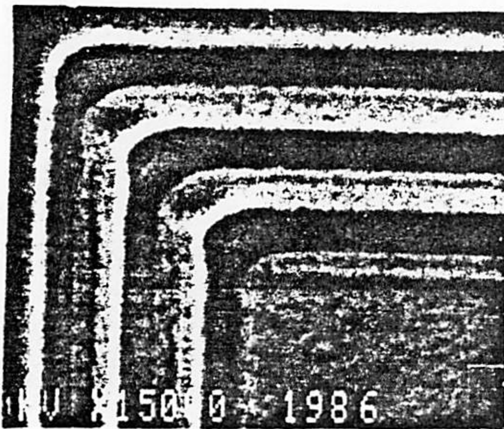
15. S. Subramanian, "Rapid Calculation of Defocused Partially Coherent Images," *Applied Optics*, vol. 20, no. 10, pp. 1854-57, 15 May 1981.

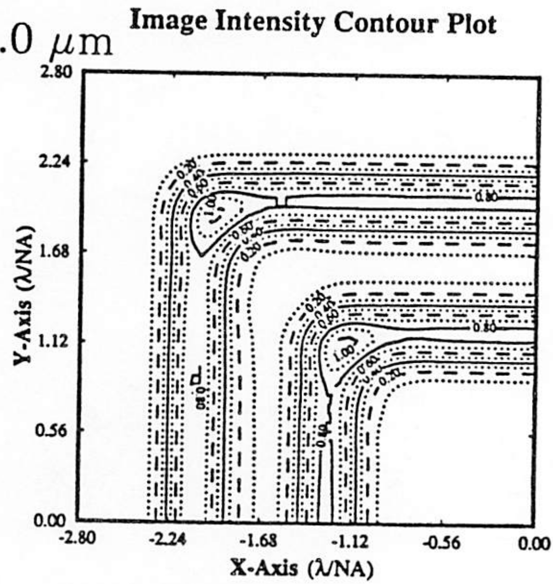
# FIGURES



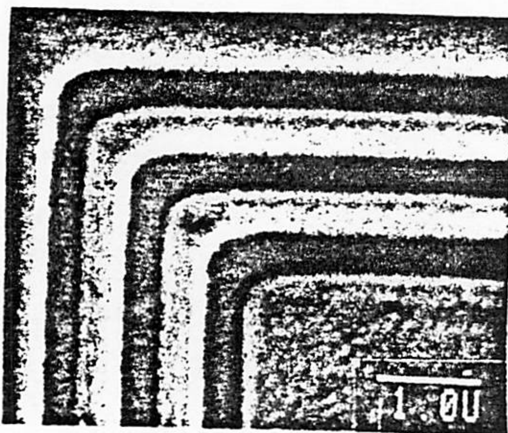



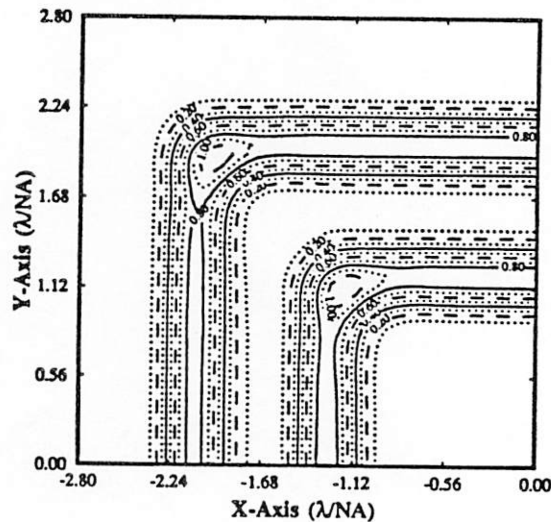

Best Focus + 1.0 $\mu$m




Best Focus + 0.5 $\mu$m




21 sec

Best Focus

**Figure 1.** The effect of aberrations on image quality for 0.4 μm elbows. The left column shows SEMs[1] with varying amounts of defocus, while the right column shows simulations that were run with coma at a maximum OPD of 0.2λ. λ = 0.248, NA = 0.38, and σ = 0.3.

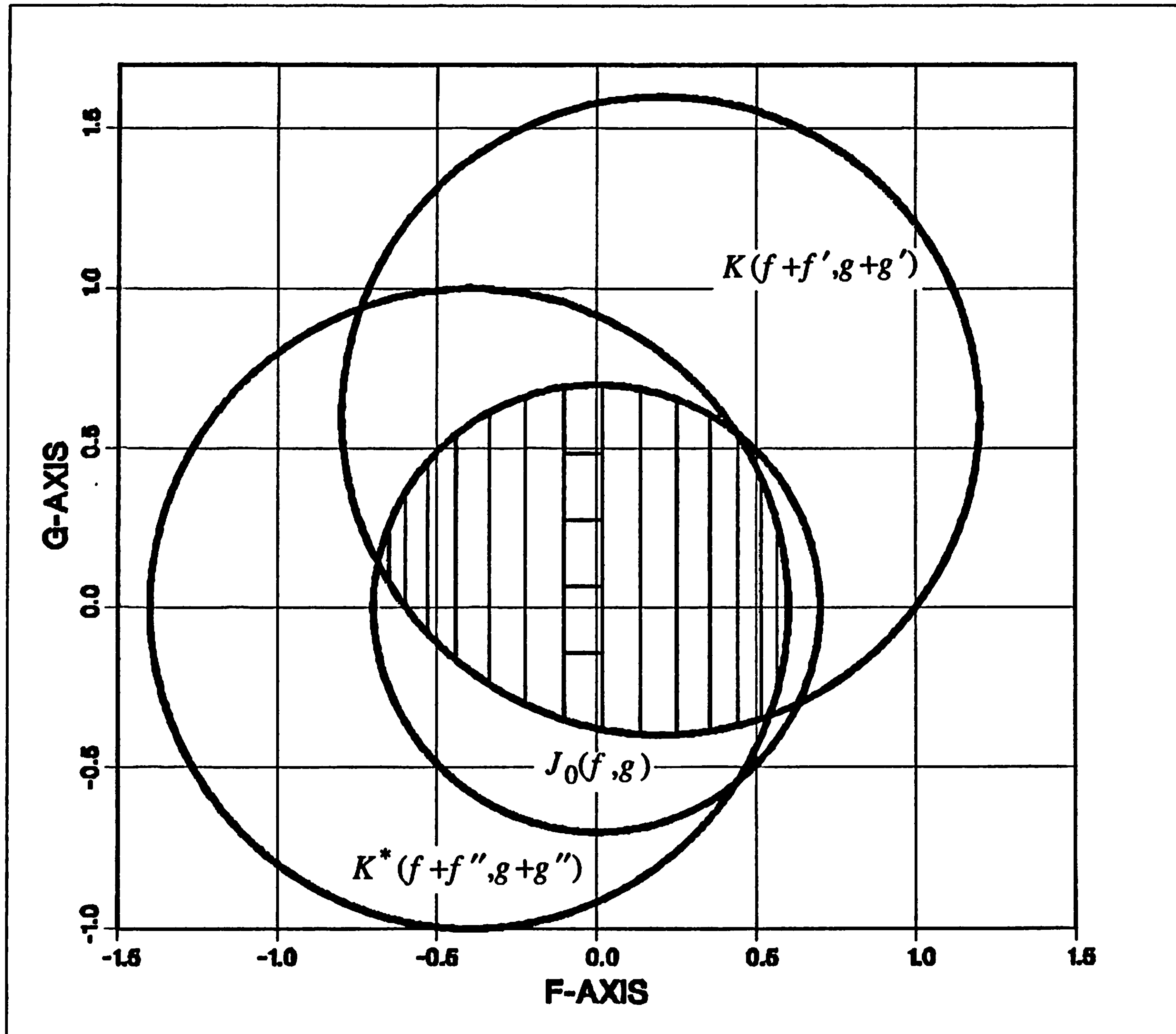
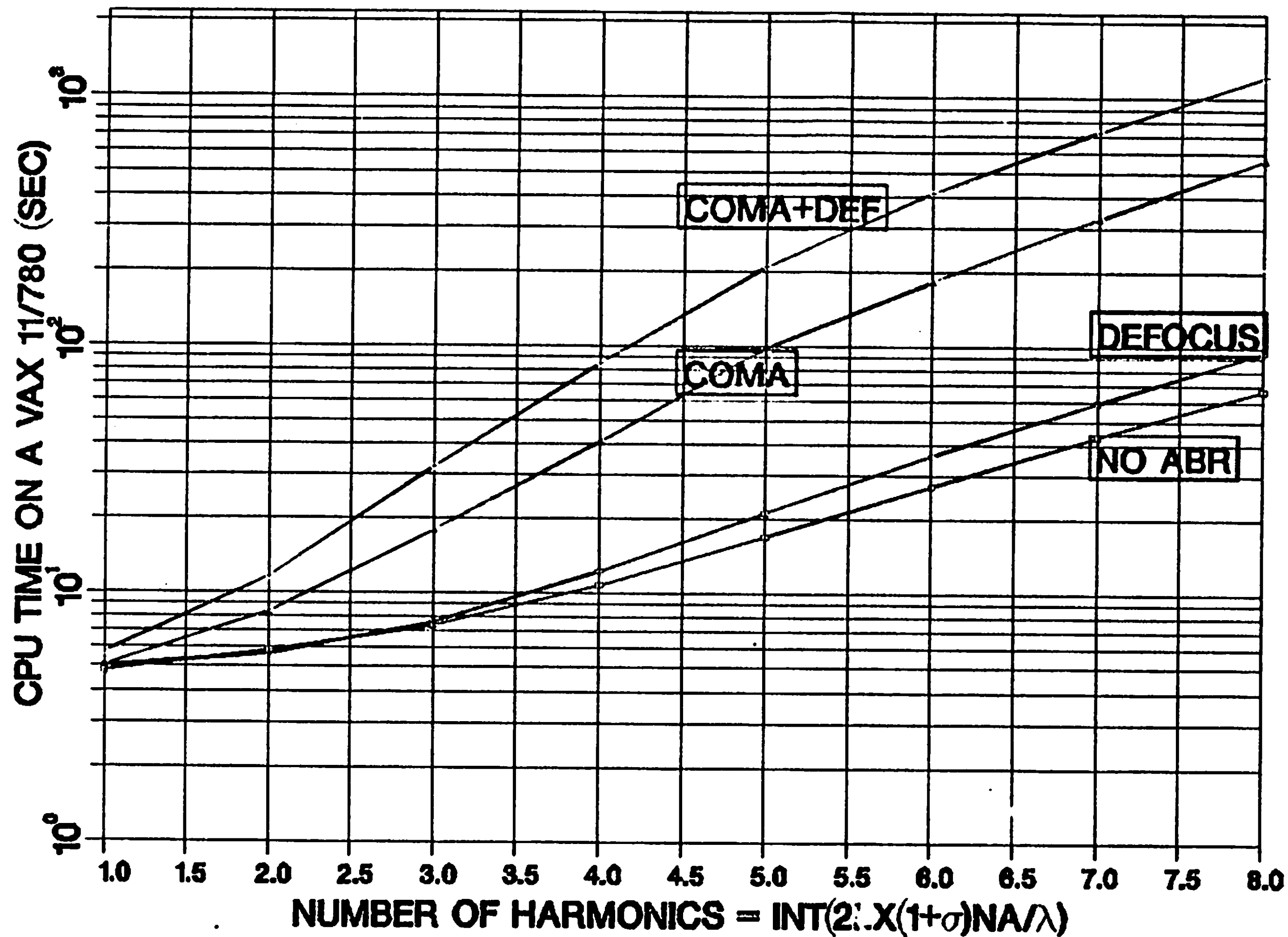
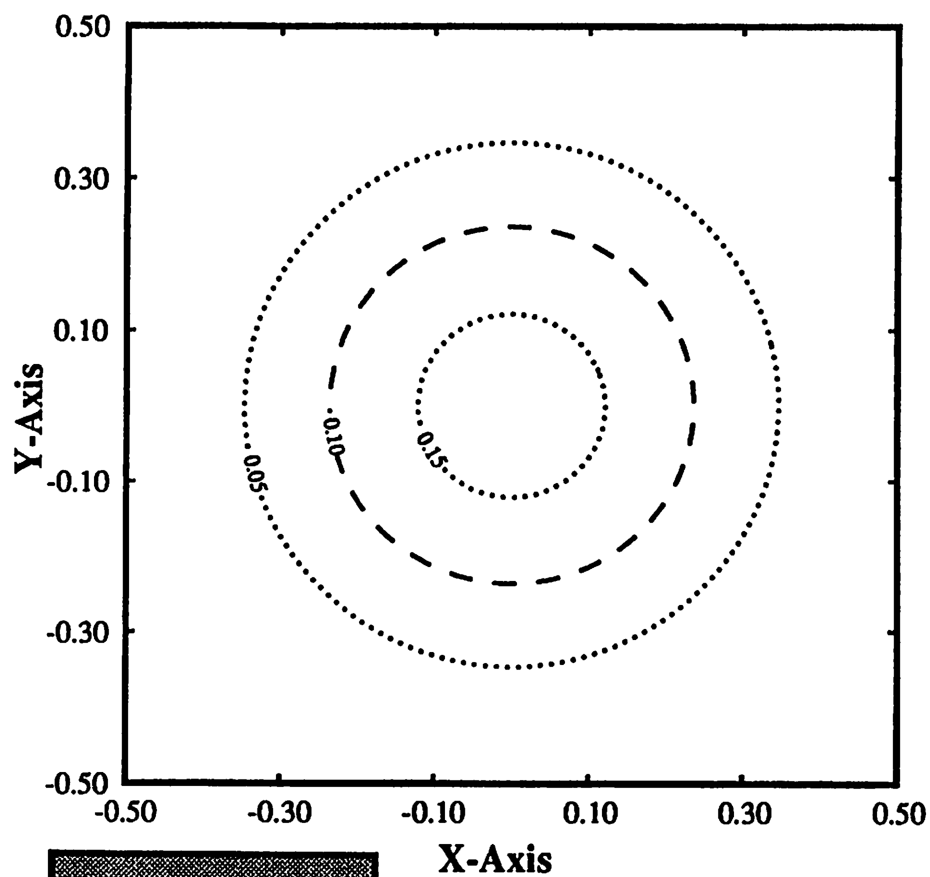


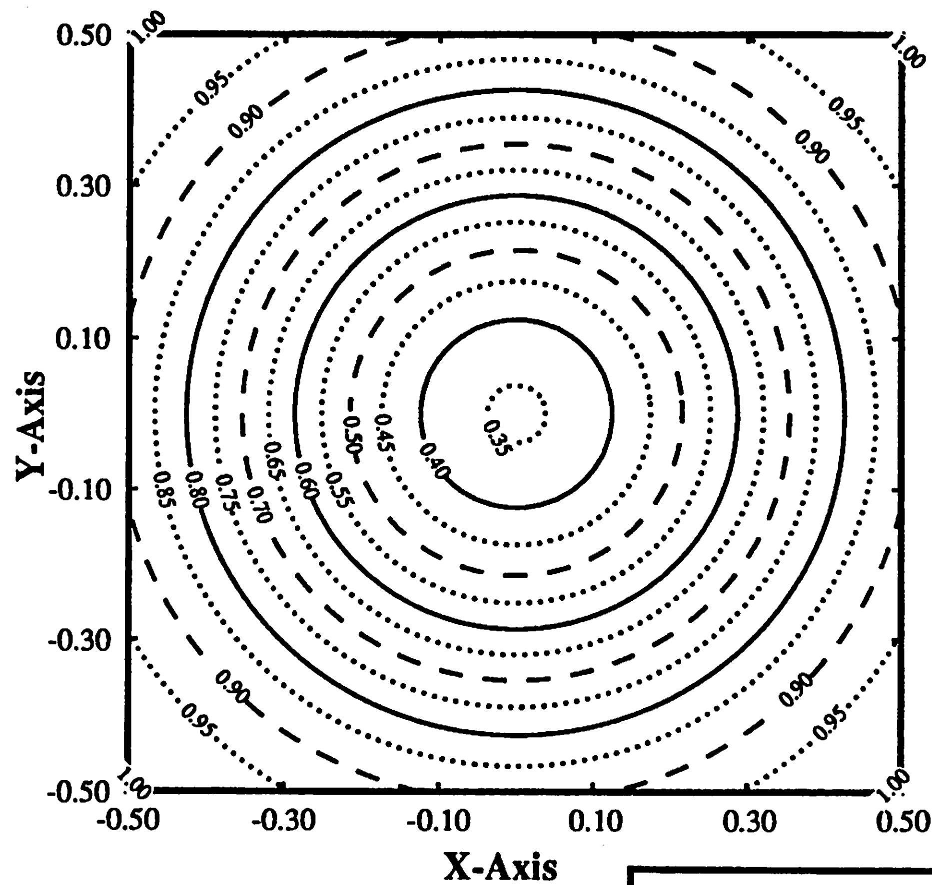

**Figure 2.** Region of integration of the $TCC(f',g':f'',g'')$ for circular apertures.




| LX in λ/NA | # Harmonics (σ = 0.3) | # Harmonics (σ = 0.7) |
|---|---|---|
| 0.25 | 0 | 0 |
| 0.50 | 1 | 1 |
| 0.75 | 1 | 2 |
| 1.00 | 2 | 3 |
| 1.25 | 3 | 4 |
| 1.50 | 3 | 5 |
| 1.75 | 4 | 5 |
| 2.00 | 5 | 6 |
| 2.25 | 5 | 7 |
| 2.50 | 6 | 8 |
| 2.75 | 7 | 9 |
| 3.00 | 7 | 10 |
| 3.25 | 8 | 11 |
| 3.50 | 9 | 11 |
| 3.75 | 9 | 12 |
| 4.00 | 10 | 13 |
| 4.25 | 11 | 14 |
| 4.50 | 11 | 15 |
| 4.75 | 12 | 16 |
| 5.00 | 12 | 17 |

**Figure 3.** Computation time versus number of harmonics for square masks. The relationship between the number of harmonics and the mask size is given in the attached table, where 2LX is equal to the total mask size.



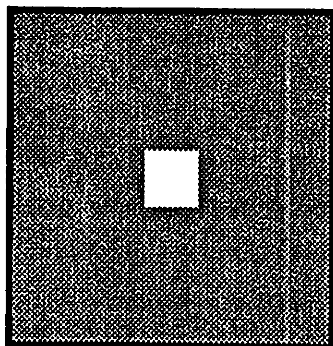
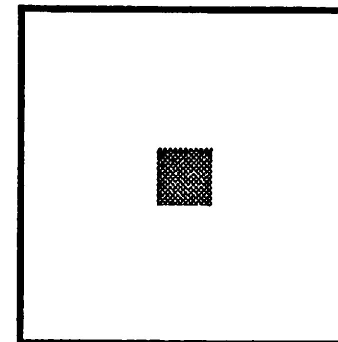



Figure 4. Image Intensity Contour plots of (a) a 0.4λ/NA x 0.4λ/NA square transparent defect in an opaque mask, (b) a 0.4λ/NA x 0.4λ/NA square opaque defect in a transparent mask. The defect dimensions are close to the resolution limit of the stepper, so the contours are circular.



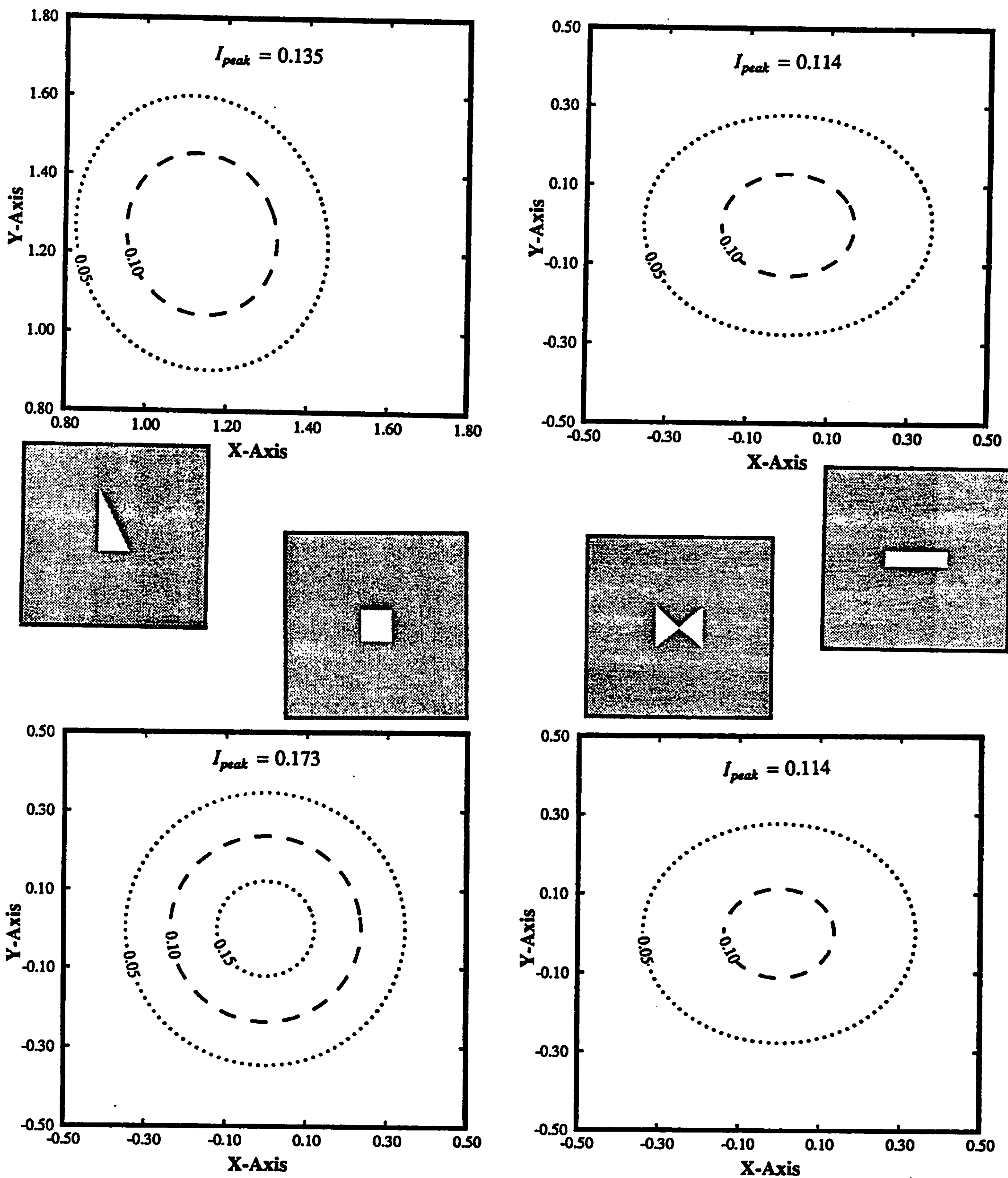



**Figure 5.** Image Intensity Contour plots for transparent defects of different shapes but equal areas (0.4λ/NA x 0.4λ/NA. The contours for all these shapes are circular, which shows that the image profile is roughly independent of shape.

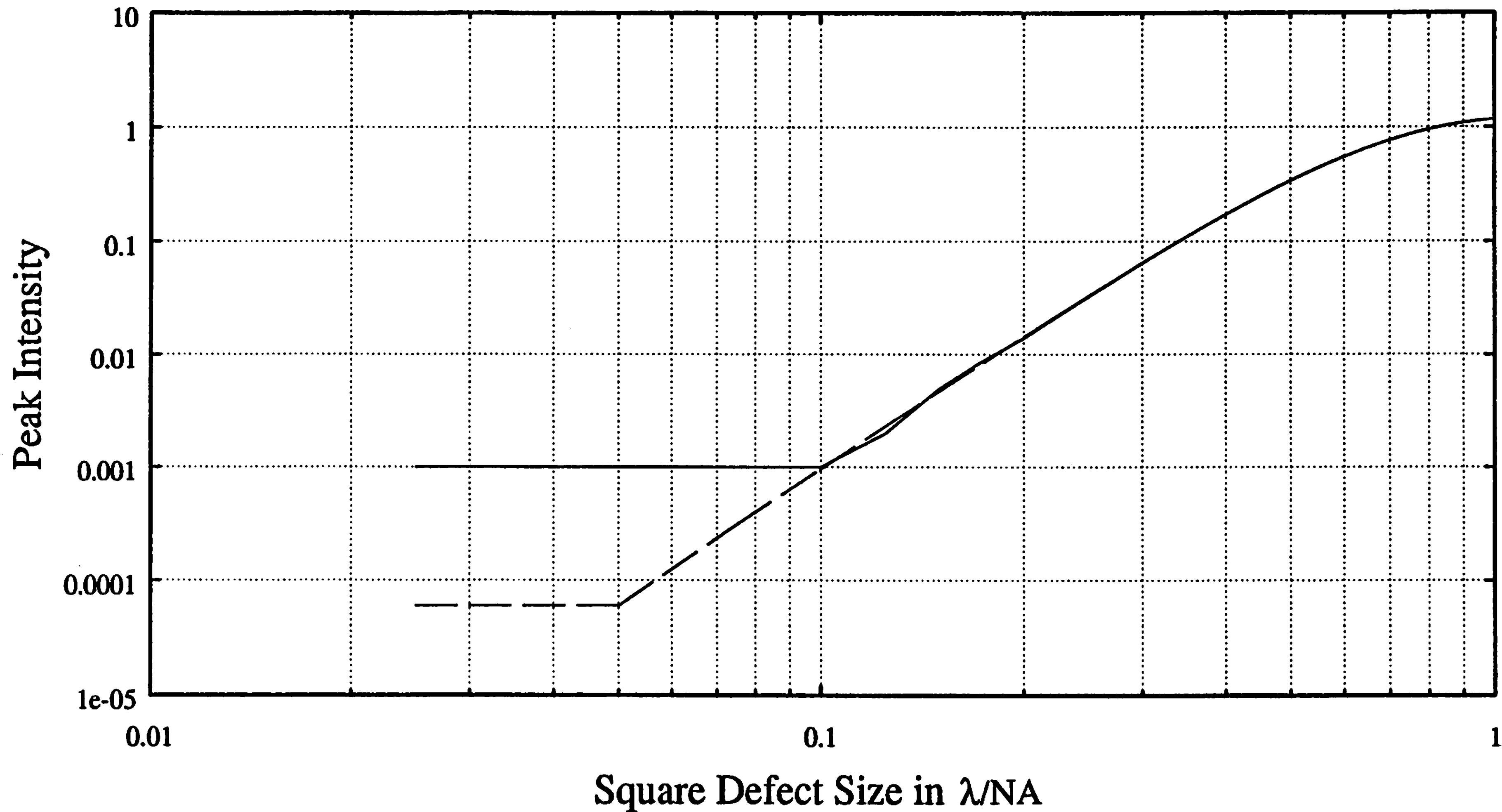



**Figure 6.** Logarithmic plot of the peak intensity $I_t$ vs defect size $D$ for transparent square defects. The solid lines come from simulations taken to 3 decimal paces, while the dashed lines are accurate to 5 decimal places. The log-slope for $0.1\lambda/NA \leq D \leq 0.4\lambda/NA$ is approximately 4.0, which implies that $I_t \propto D^4$.

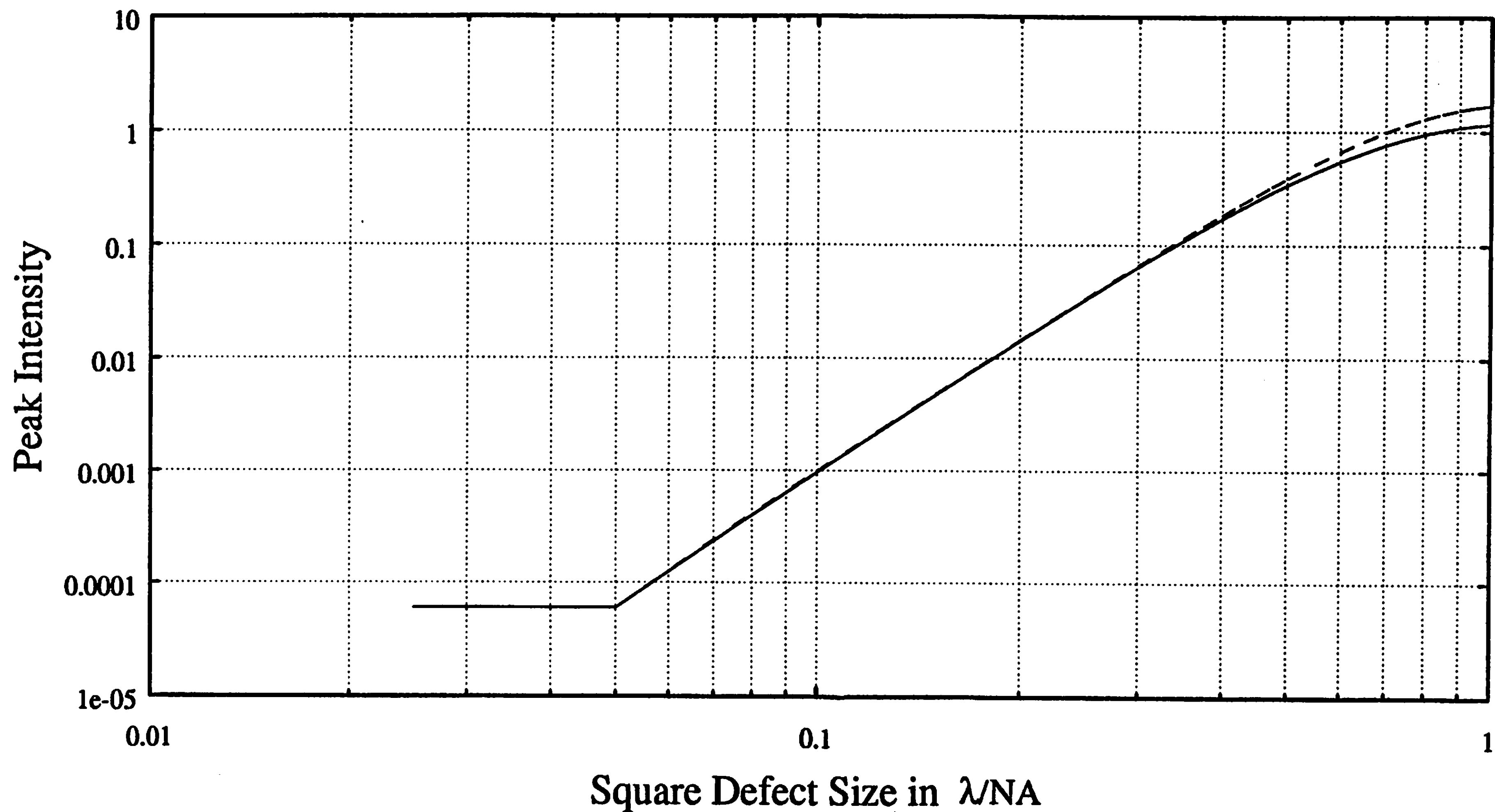



**Figure 7.** Logarithmic plot of the peak intensity $I_t$ vs defect size $D$ for transparent square defects for two different values of partial coherence $\sigma$. The solid lines are for simulations with $\sigma = 0.7$, and the dashed lines are for $\sigma = 0.3$. The two curves coincide for $D \leq 0.4\lambda/\mathrm{NA}$, which shows that the intensity of small defects is independent of $\sigma$.

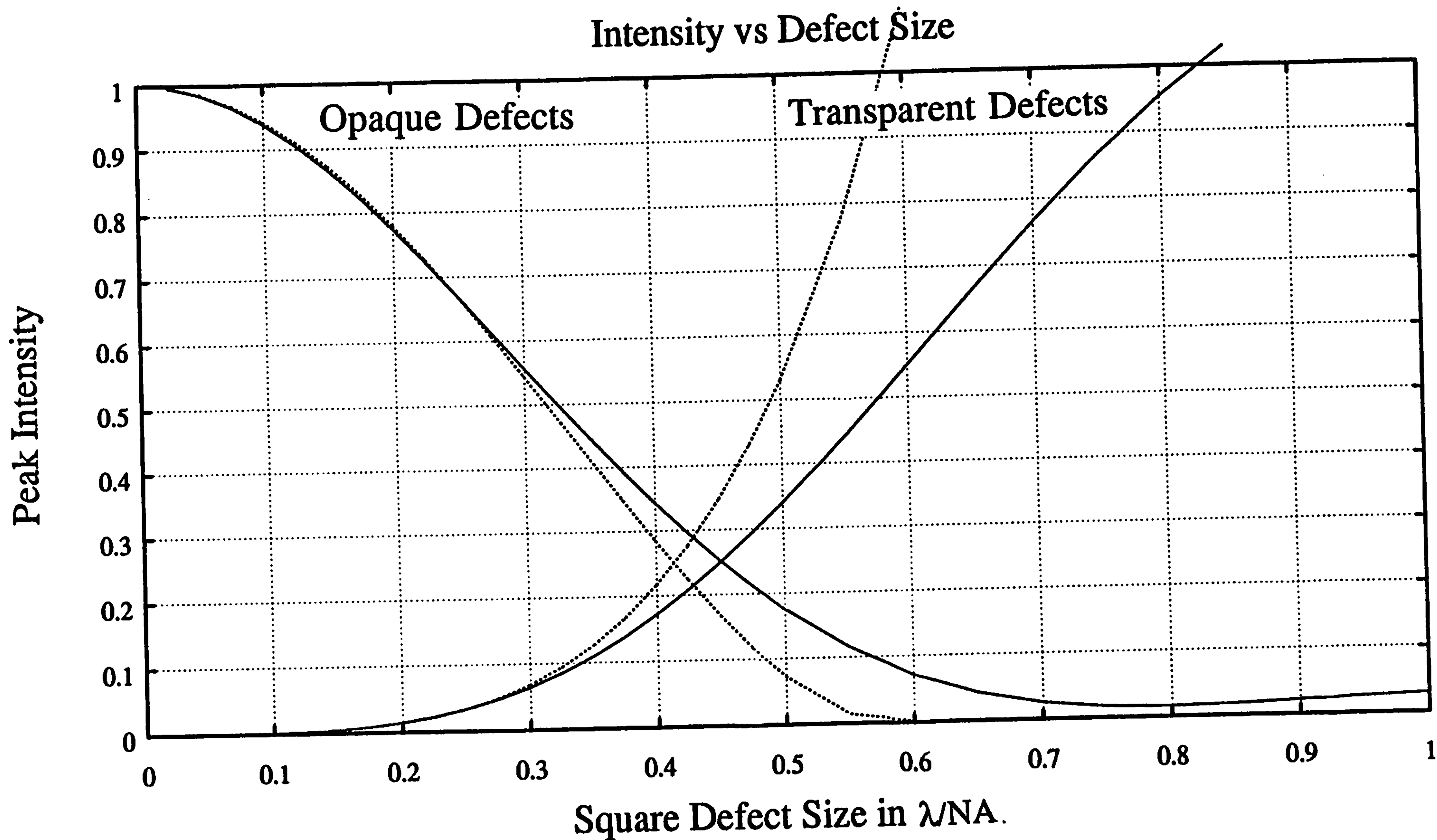



**Figure 8.** Intensity vs Defect Size plots for $\sigma = 0.7$, based on simulations of transparent and opaque square defects. The dotted lines come from Equations (3.2) and (3.5), and show that these equations only hold for $D \leq 0.4\lambda/NA$.

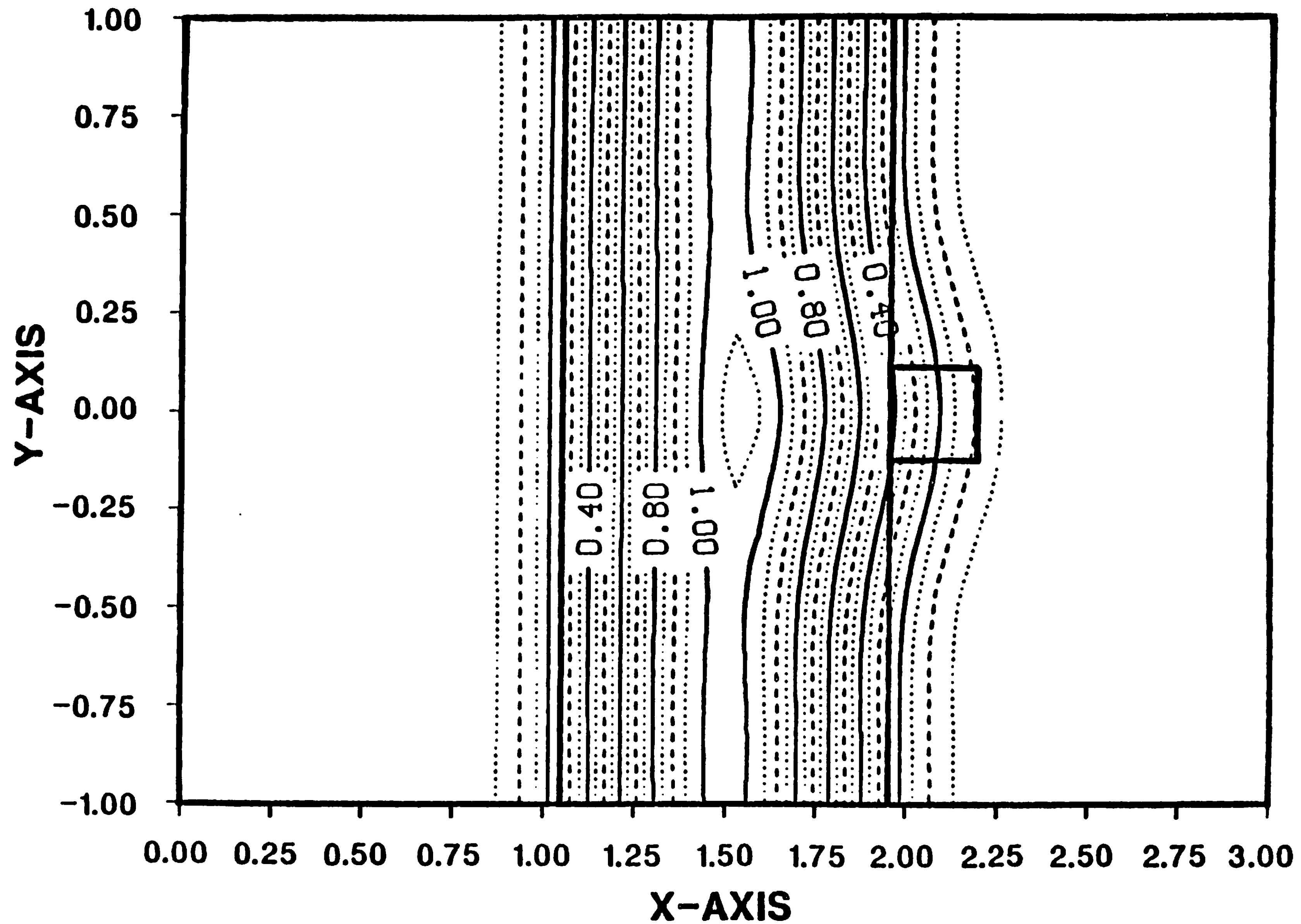



**Figure 9.** Transparent defect interaction with a transparent isolated feature. The 0.25 λ/NA x 0.25 λ/NA defect by itself has a peak intensity 3.3% of clear field, but when placed next to a 0.8λ/NA feature, it causes a 0.1λ/NA linewidth change.

**Figure 10.** Linewidth Change versus Defect Size for a transparent defect attached to a transparent isolated feature. The boxed points are from Equation (3.8), and compare well with 30% intensity threshold simulations (dashed lines) as well as to more rigorous resist simulations (solid lines).

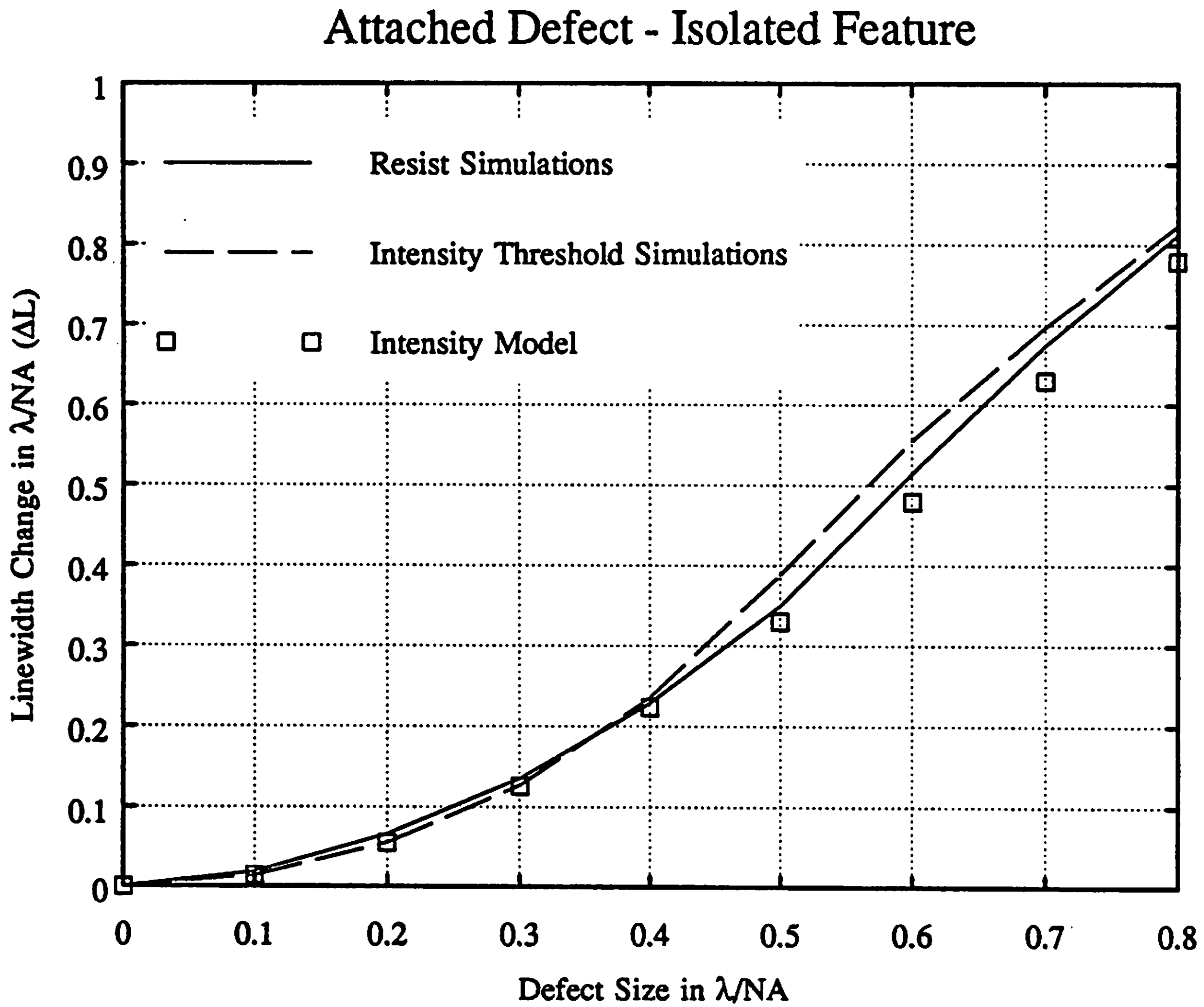


**Figure 11.** A pair of nested 0.8λ/NA transparent elbows, simulated at $\sigma = 0.3$. The ringing and linewidth variation at the corners of the elbows might not be tolerable in some applications.

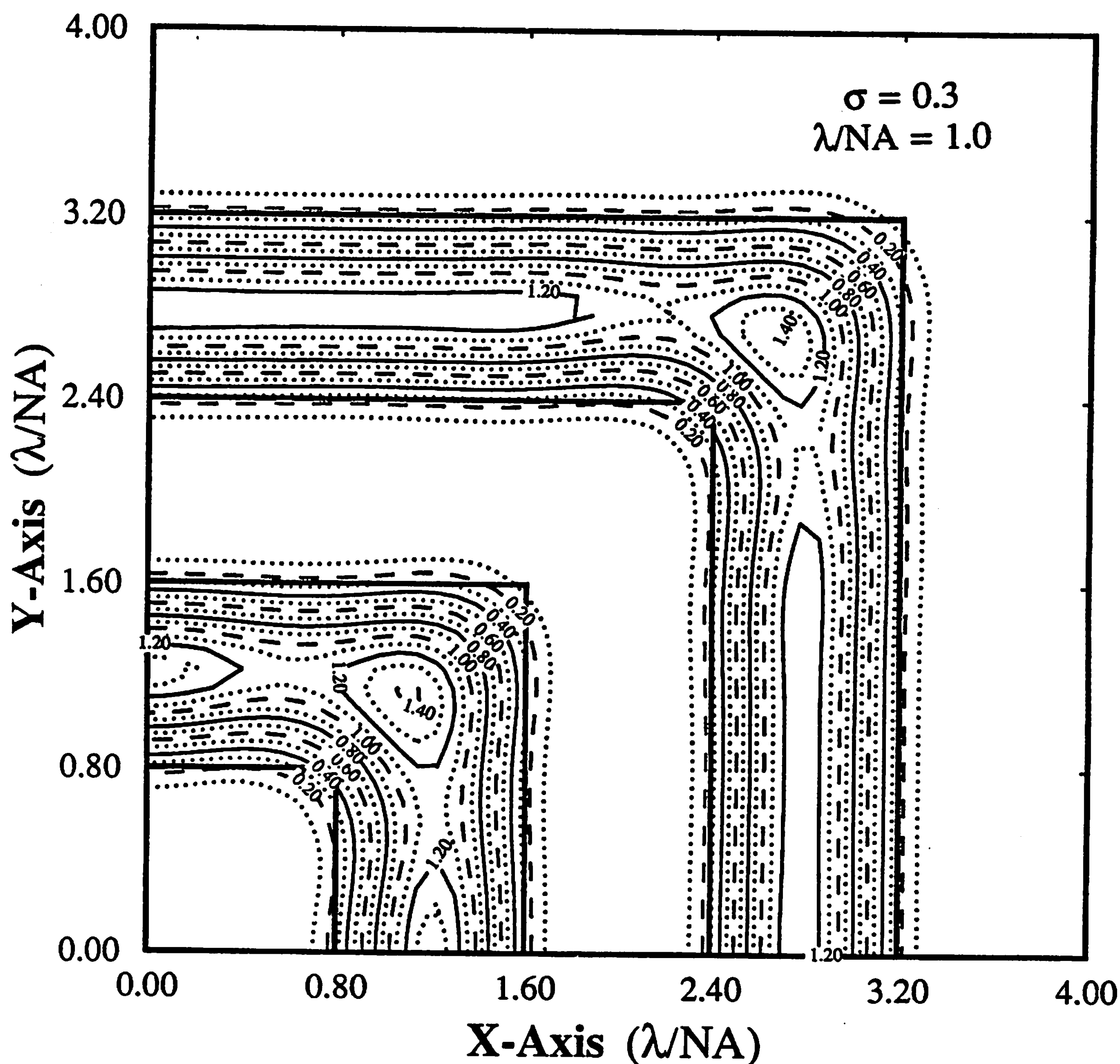


**Figure 12.** A pair of nested 0.8λ/NA transparent elbows, simulated at $\sigma = 0.7$. The lower spatial coherence results in smoother contours, but some resolution is lost.

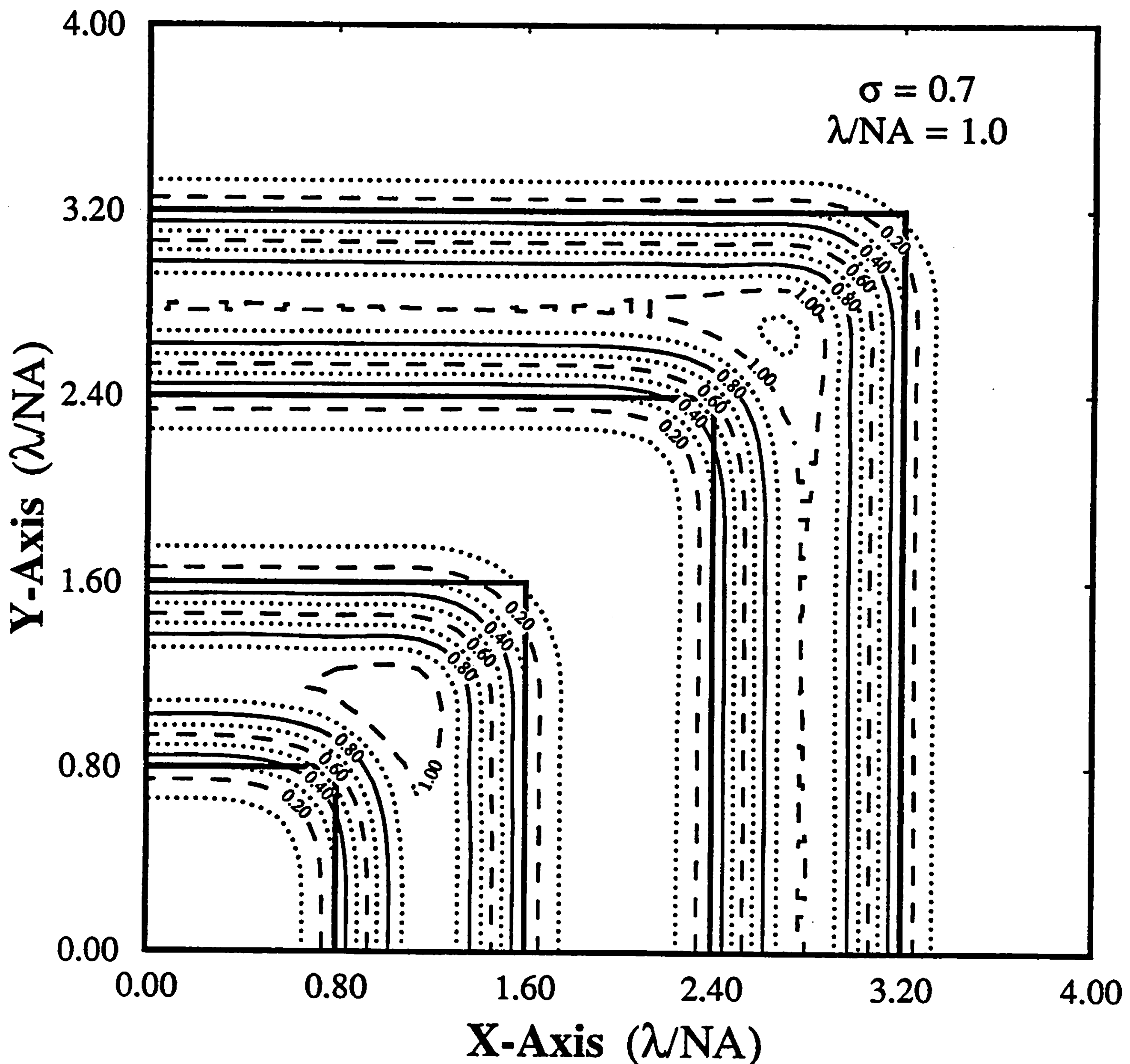


**Figure 13.** A pair of nested 0.8λ/NA transparent elbows, simulated at $\sigma = 0.3$. 45° triangular tapers have been used to round the elbows, and smooth intensity contours without ringing are obtained.

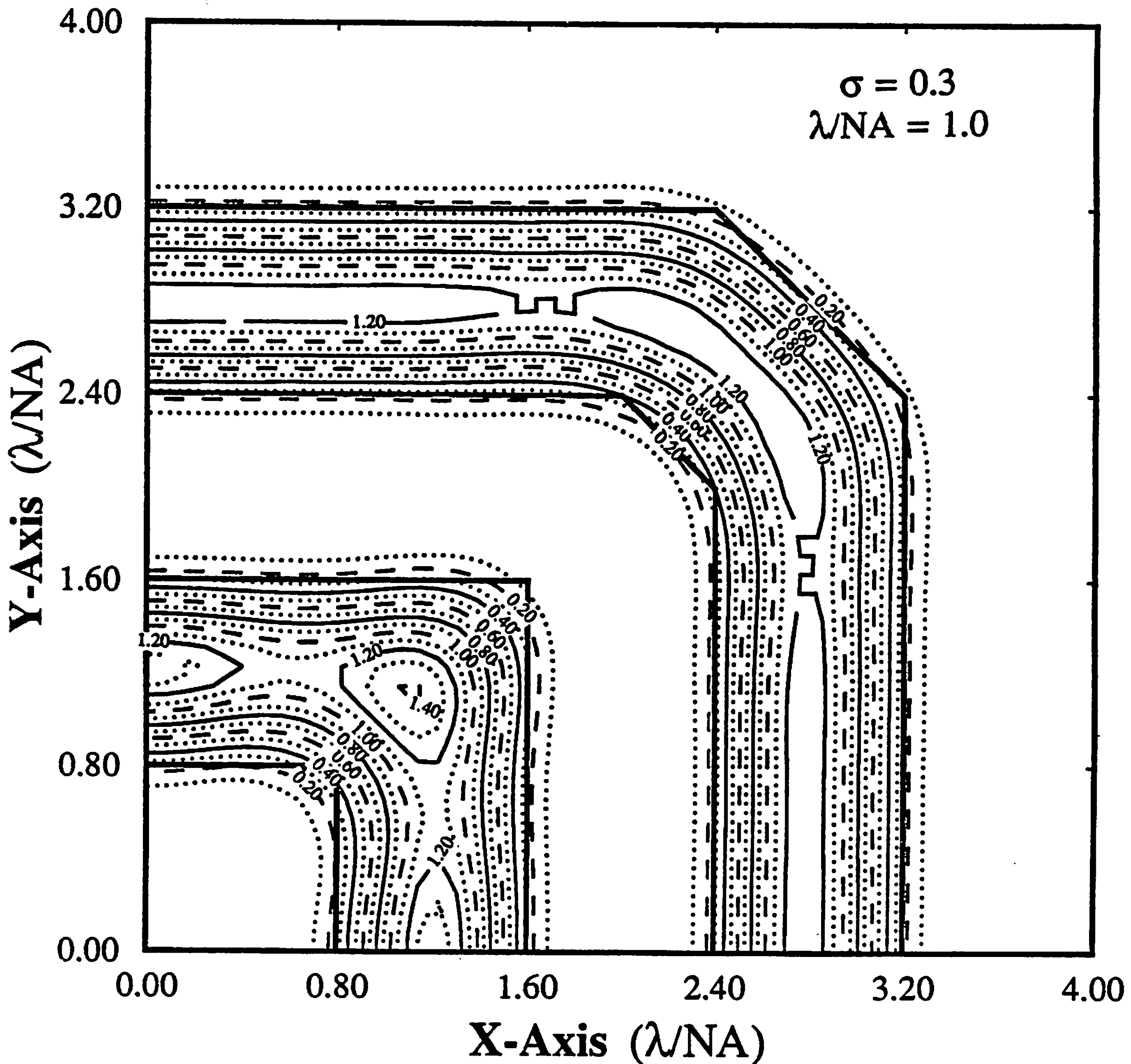


**Figure 14.** Equally spaced 0.6λ/NA features, with a protruding space, simulated at σ = 0.7.

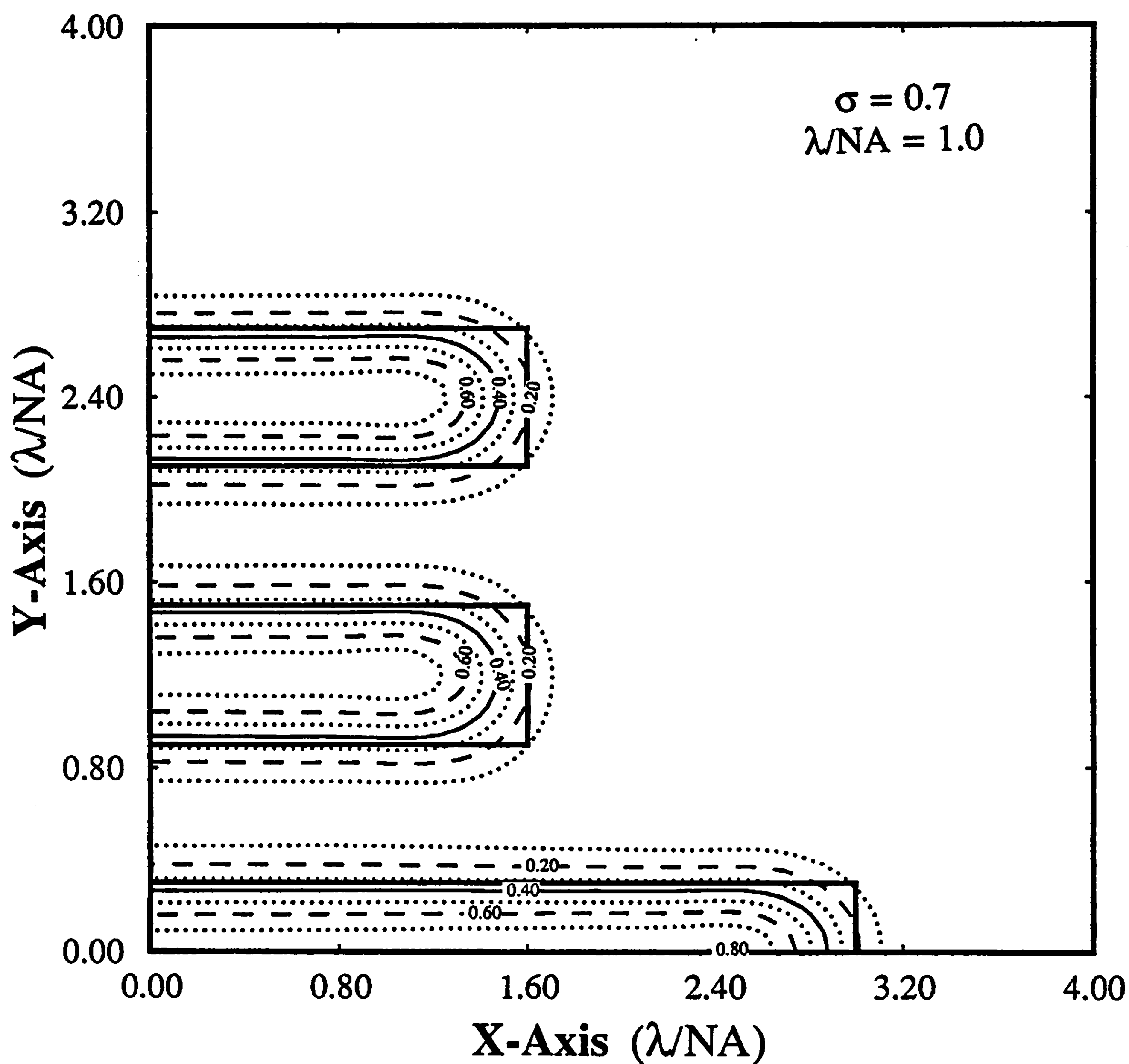


**Figure 15.** Equally spaced 0.6λ/NA features, with a protruding line, simulated at $\sigma = 0.7$.

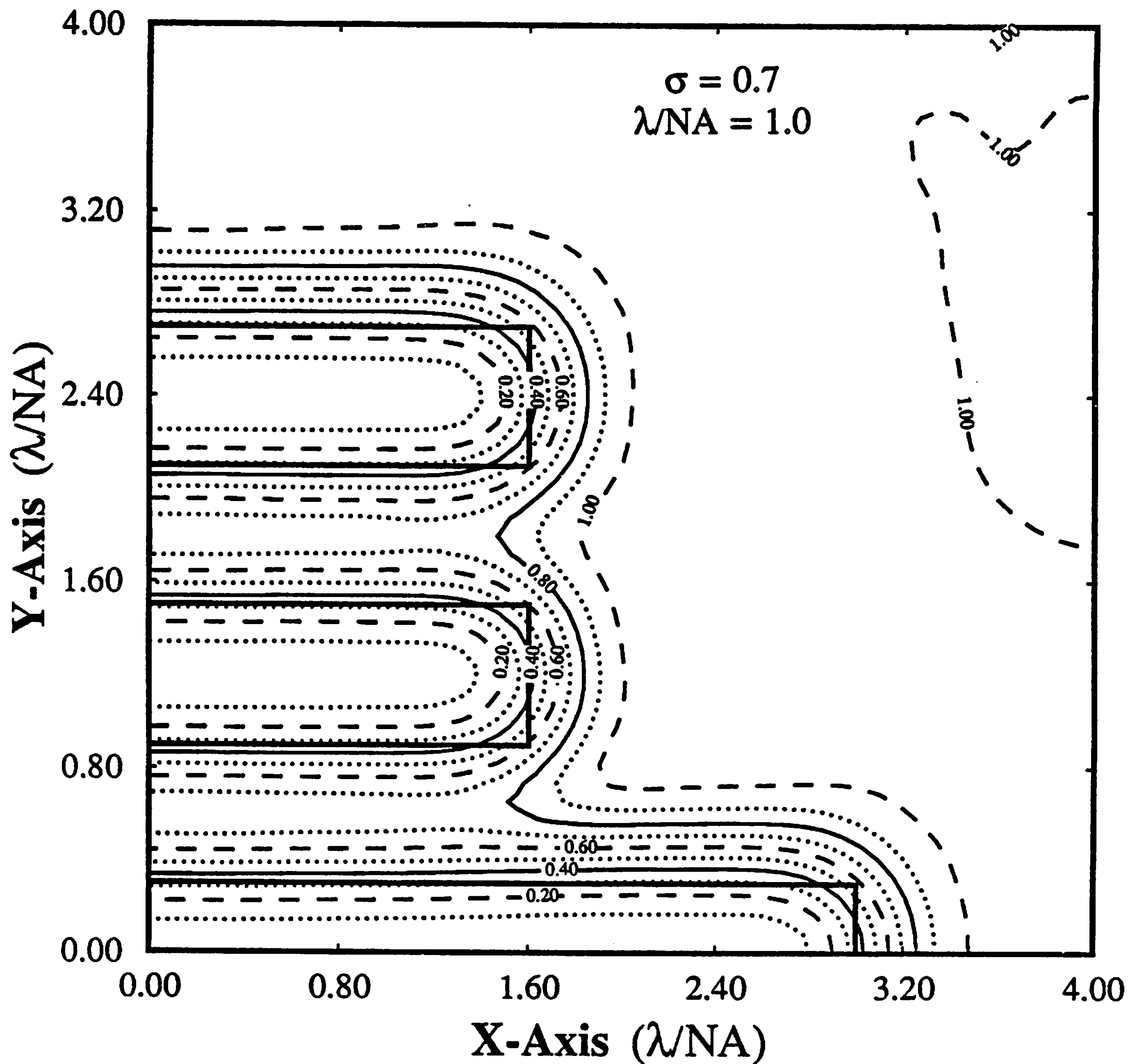


**Figure 16.** Butting Error - two 0.8λ/NA features are "pinched" together, and overlap by 0.2λ/NA. σ = 0.5. This situation is not critical.

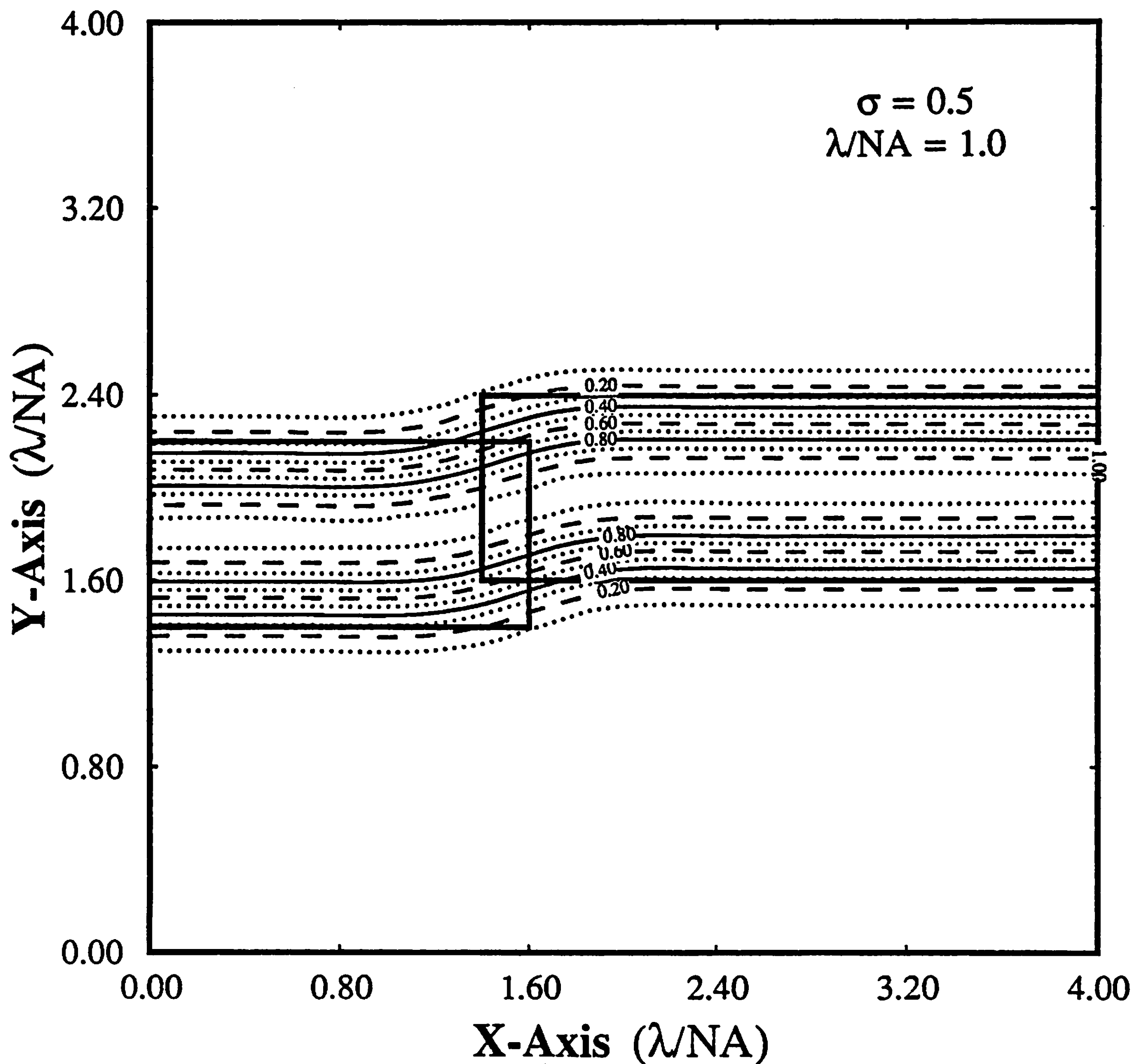


**Figure 17.** Stitching Error - two 0.8λ/NA features are separated by 0.2λ/NA. $\sigma = 0.5$. This situation too is not critical, provided the resist is not underexposed.

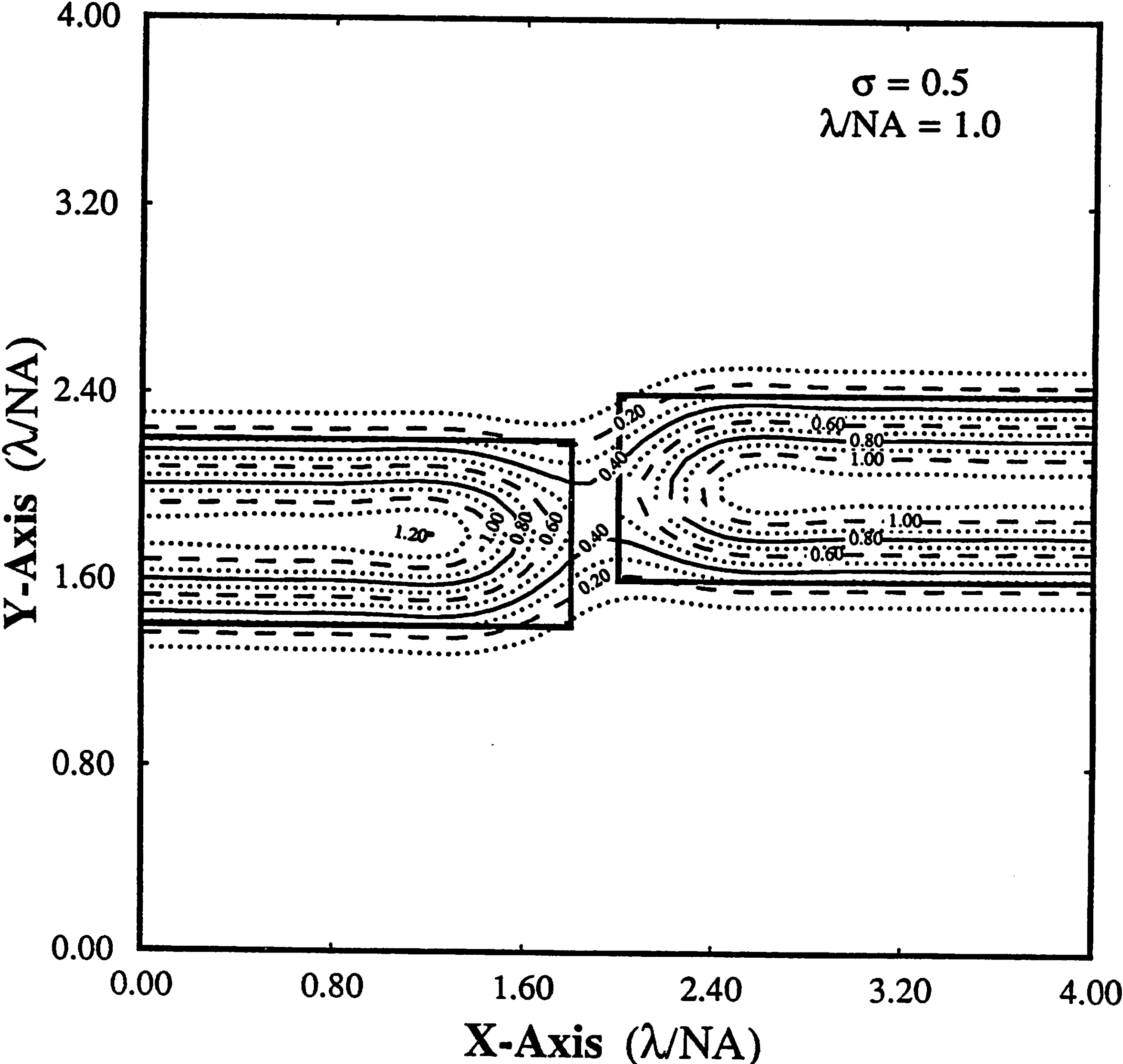
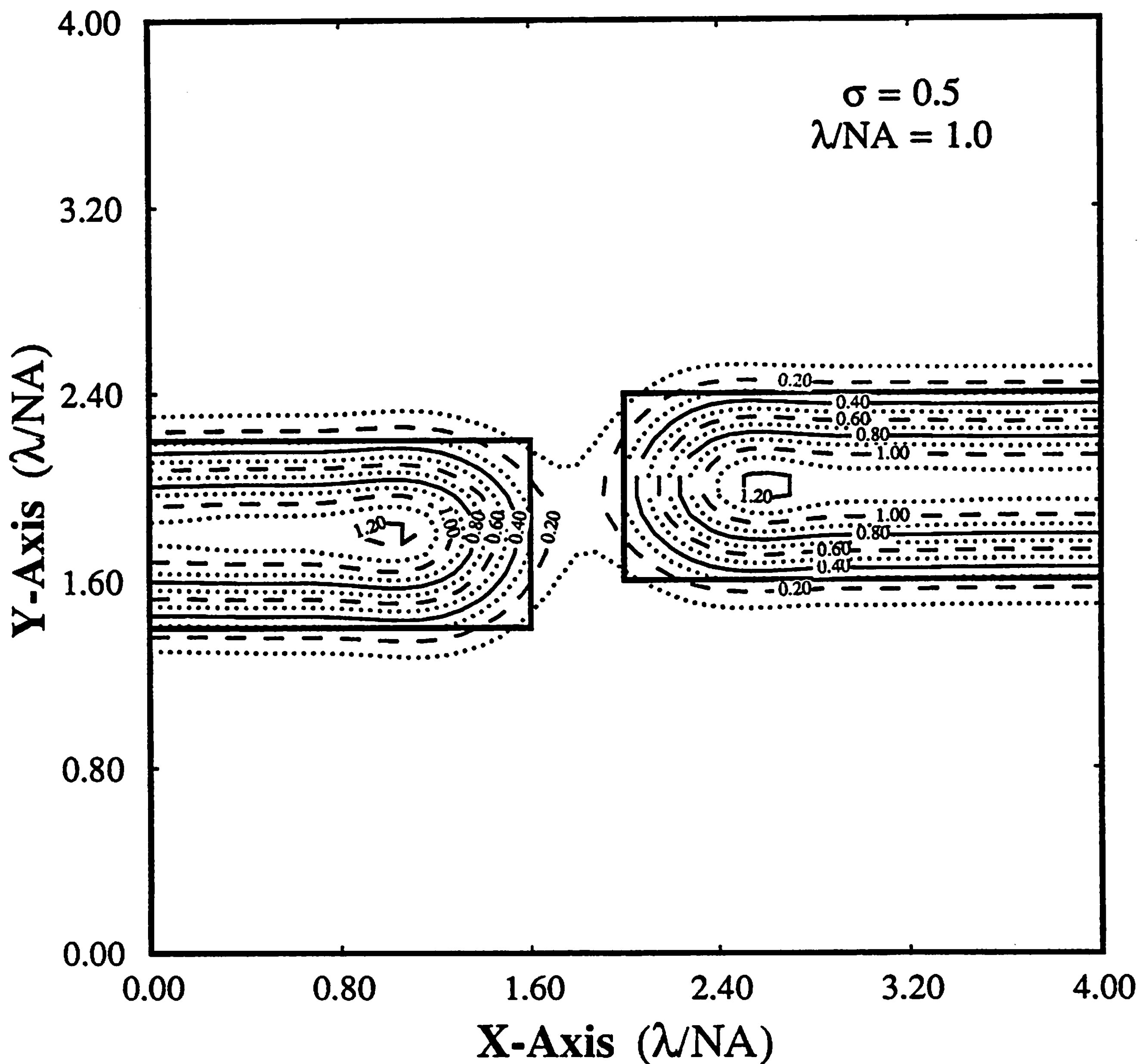
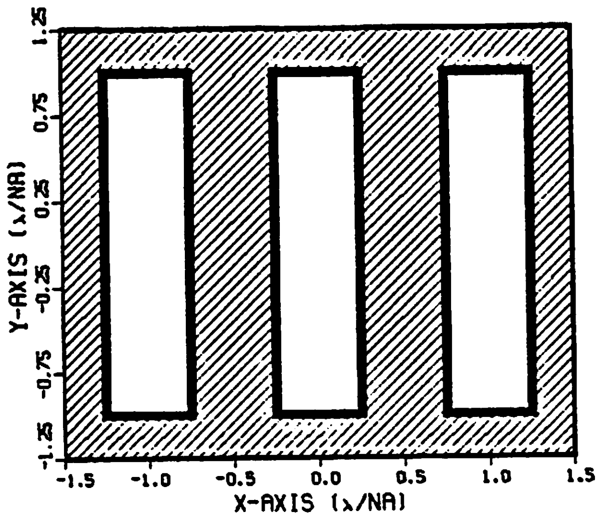
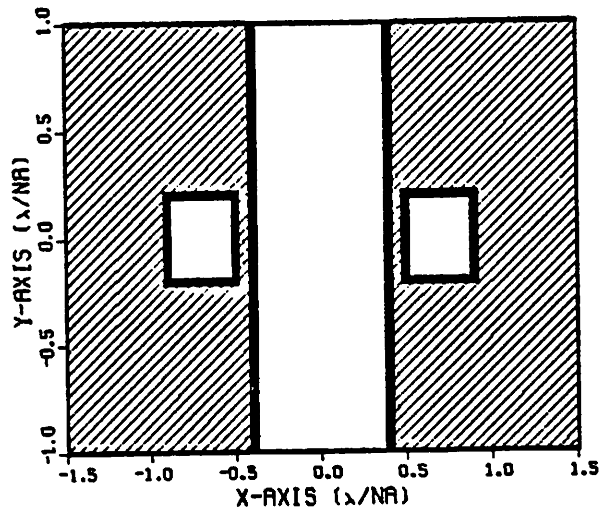


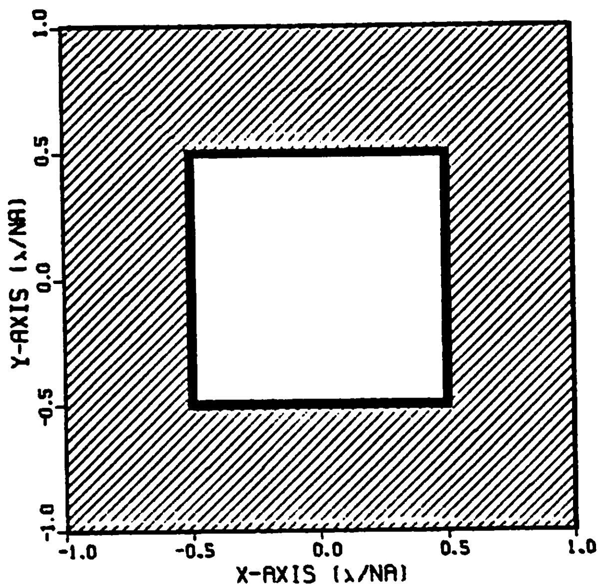
**Figure 18.** Stitching Error - two 0.8λ/NA features are separated by 0.4λ/NA. $\sigma = 0.5$. This situation is critical, and there will be a break between the two features unless the resist is severely overexposed.



**MASK PATTERNS**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**ALL MASKS USED ARE PERIODIC**
**AND SCALED IN DIMENSIONS OF $\lambda$/N.A.**



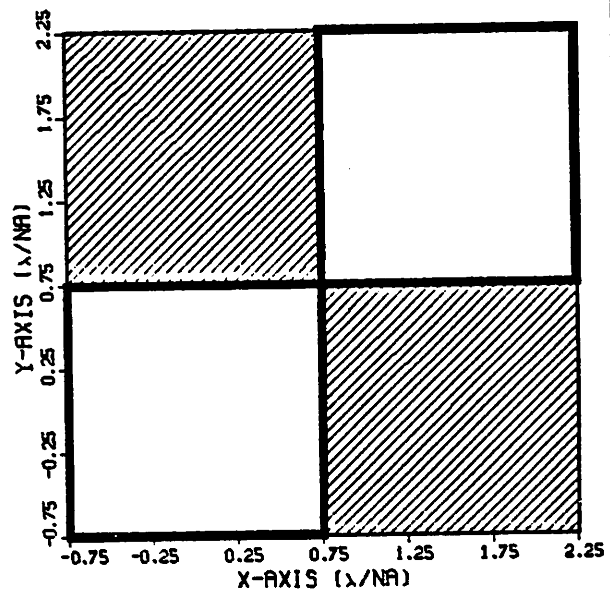
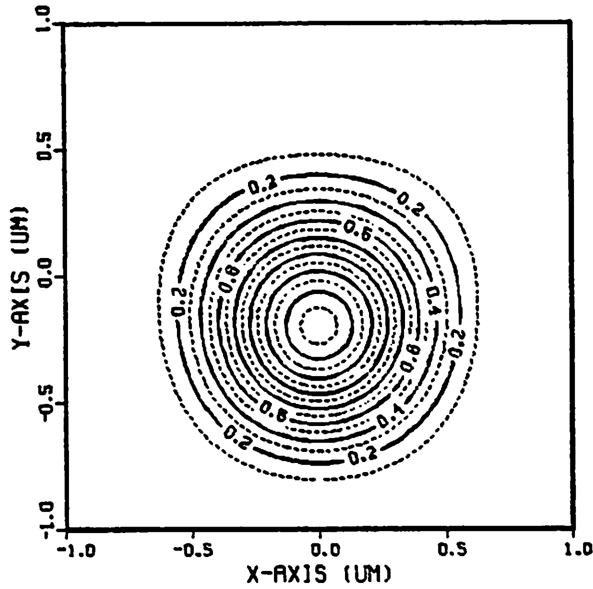



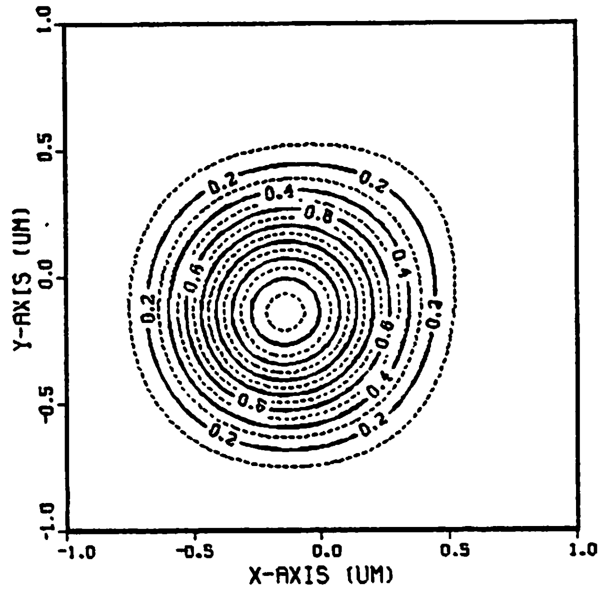

**Figure 19.** Mask test patterns simulated.

**ISOLATED SQUARE**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**SQUARE DIMENSIONS = 1.0 X 1.0 $\lambda$/N.A.**

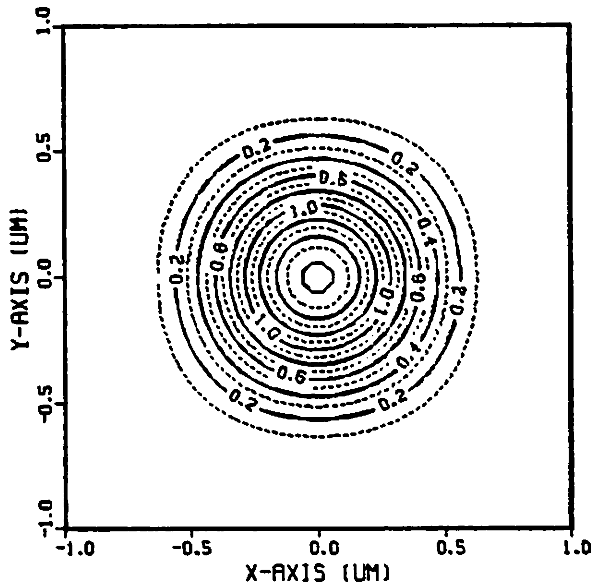
**COMA = 0.20 (0,1), OPD = 0.4$\lambda$**




**COMA = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**COMA = 0.20 (1,0), OPD = 0.4$\lambda$**

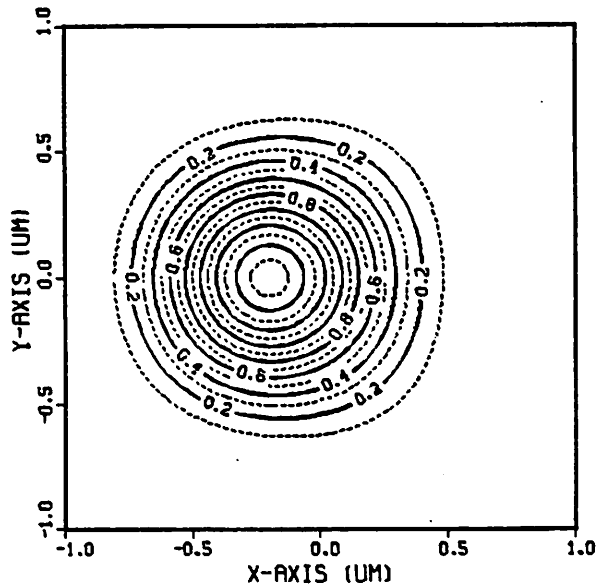
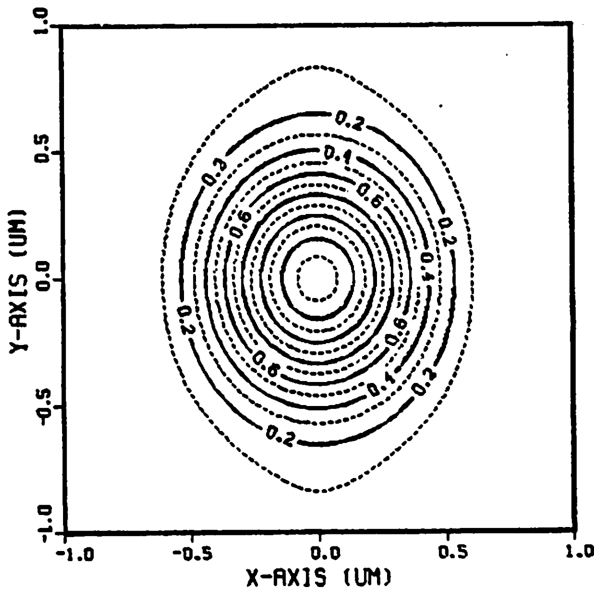


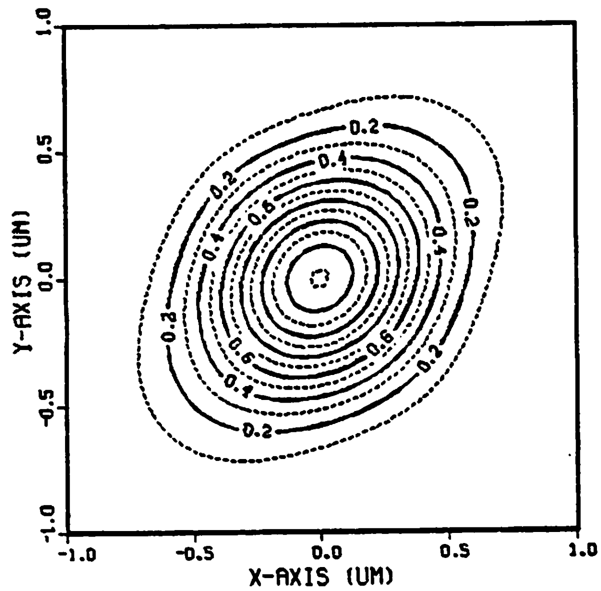

Figure 20. Intensity contours for the image of an isolated square with 0.4$\lambda$ coma.

**ISOLATED SQUARE**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**SQUARE DIMENSIONS = 1.0 X 1.0 $\lambda$/N.A.**

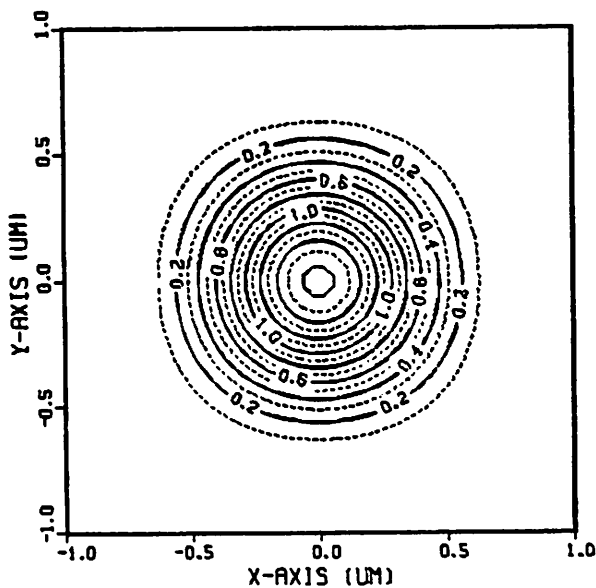
**AST = 0.20 (0,1), OPD = 0.4$\lambda$**




**AST = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**AST = 0.20 (1,0), OPD = 0.4$\lambda$**

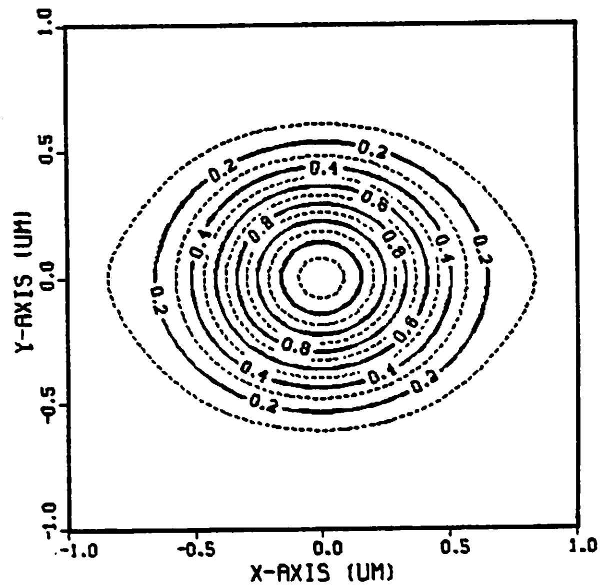
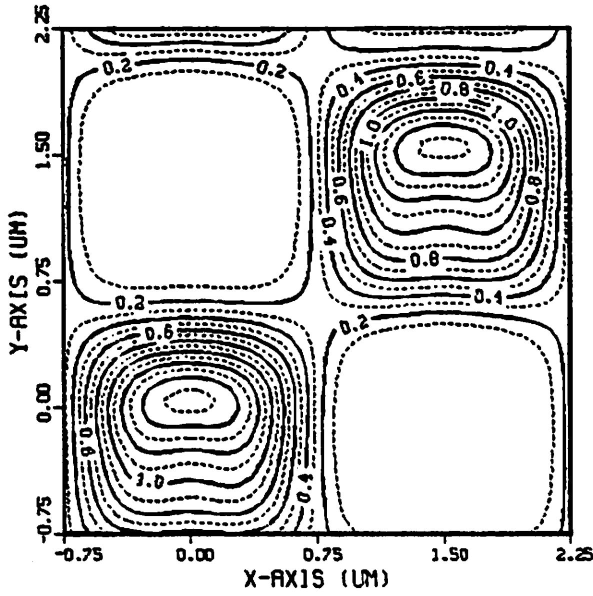


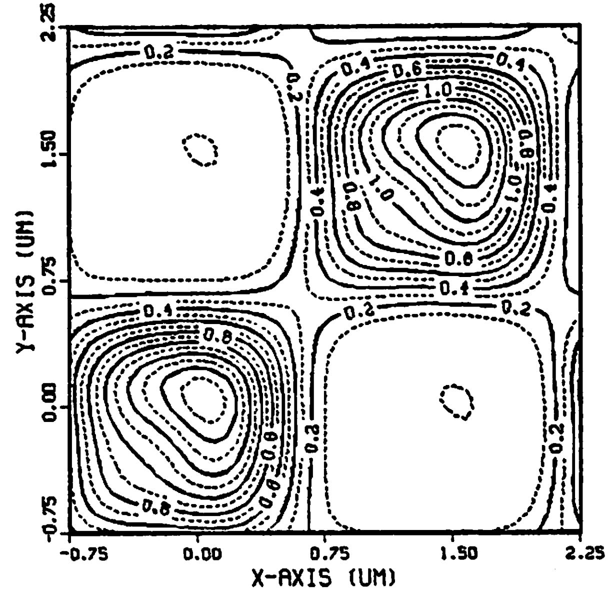

**Figure 21.** Intensity contours for the image of an isolated square with 0.4$\lambda$ astigmatism.

**CHECKERBOARD PATTERN**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**SQUARE LENGTH = 1.5 $\lambda$/N.A.**

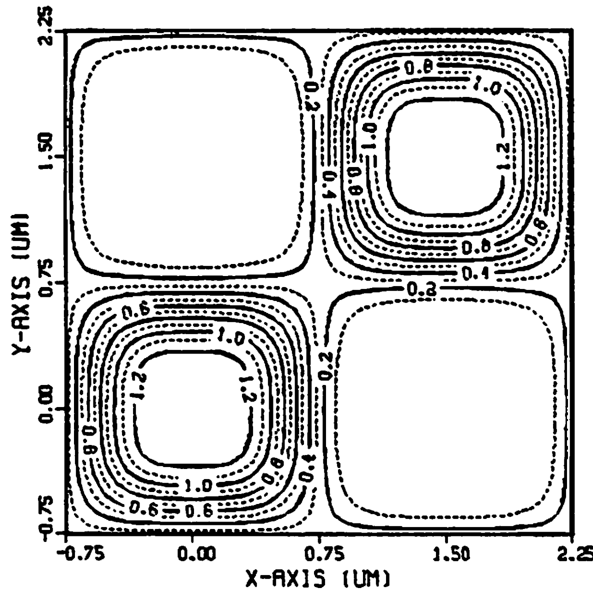
**COMA = 0.20 (0,1), OPD = 0.4$\lambda$**




**COMA = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**COMA = 0.20 (1,0), OPD = 0.4$\lambda$**

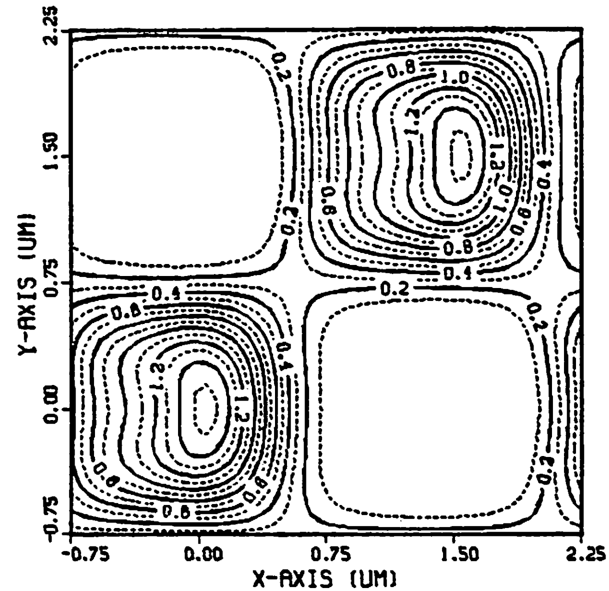
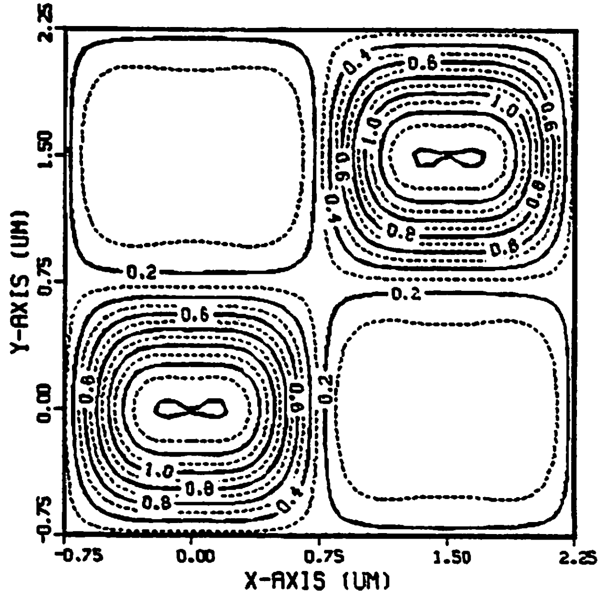


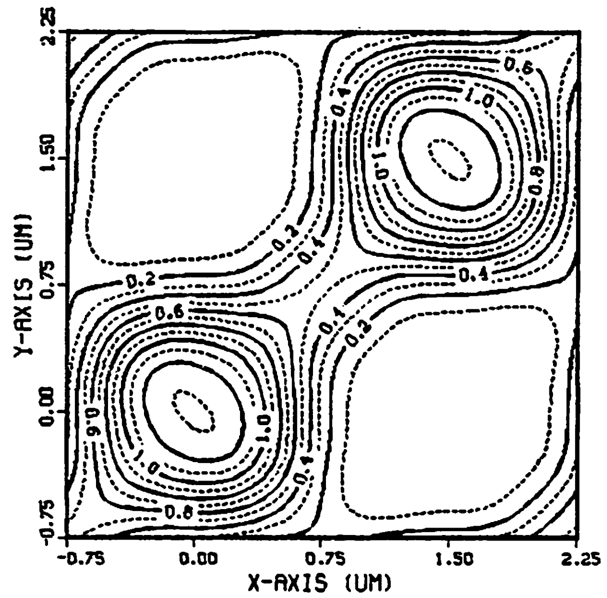

**Figure 22.** Image of a checkerboard pattern with 0.4$\lambda$ coma.

**CHECKERBOARD PATTERN**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**SQUARE LENGTH = 1.5 $\lambda$/N.A.**

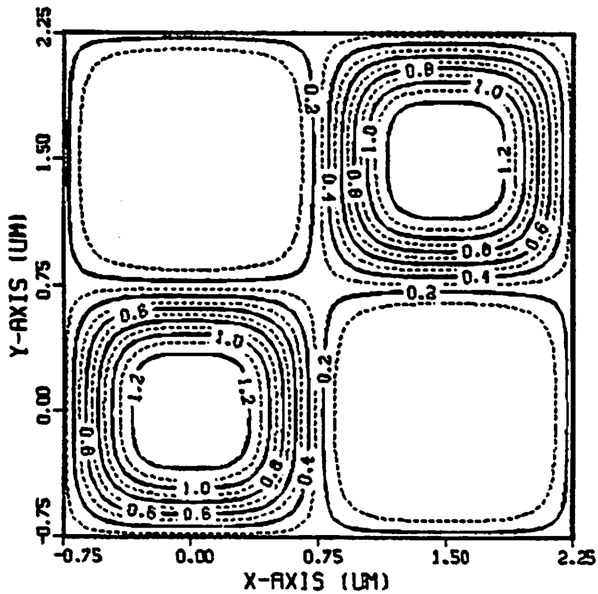
**AST = 0.20 (0,1), OPD = 0.4$\lambda$**




**AST = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**AST = 0.20 (1,0), OPD = 0.4$\lambda$**

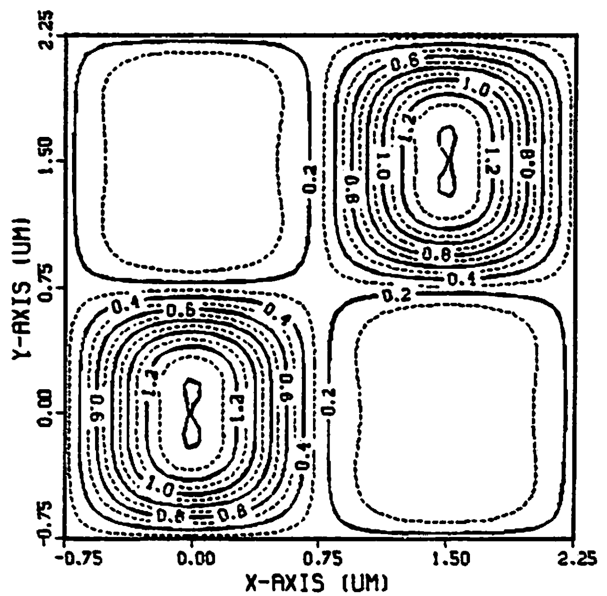
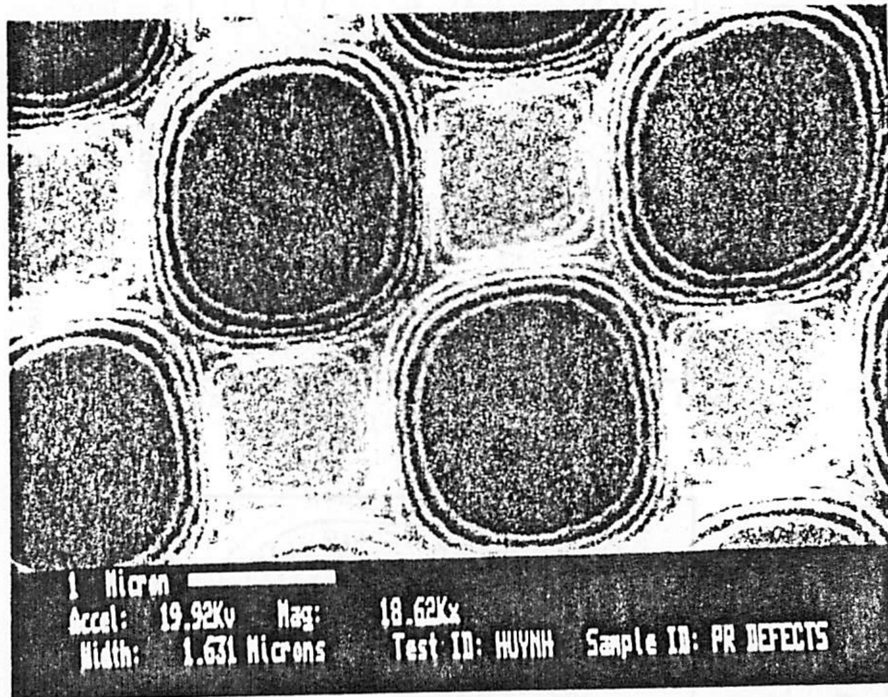



**Figure 23.** Image of a checkerboard pattern with 0.4$\lambda$ astigmatism.




## Image Intensity Contour Plot

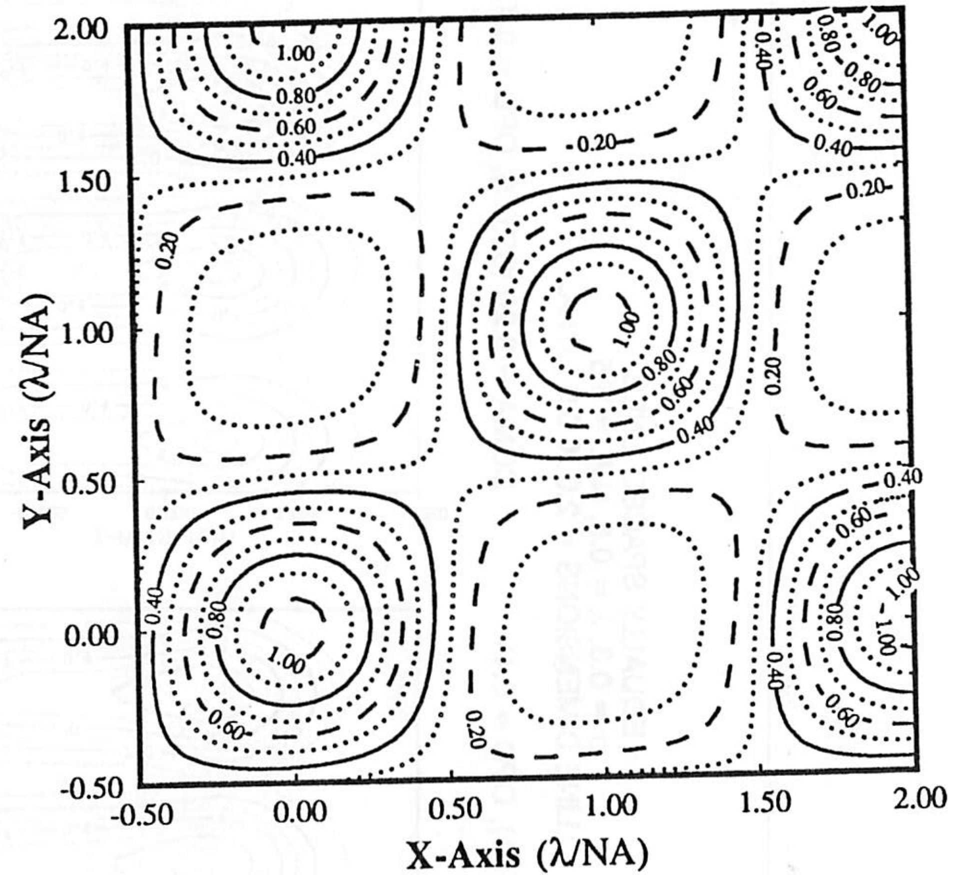
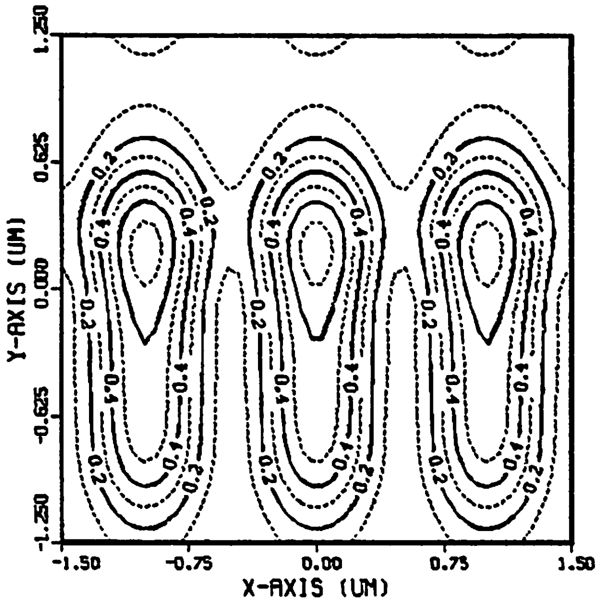


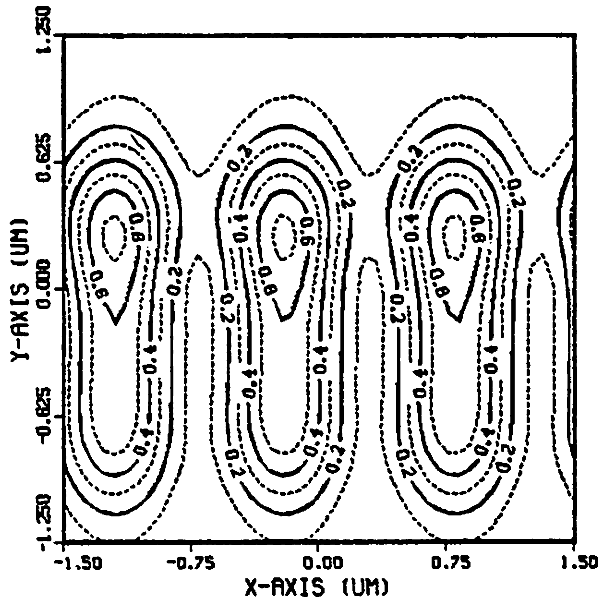

**Figure 24.** The effect of aberrations on image quality for 1.6 μm checkerboard patterns: a) experiment (λ/NA = 1.6 μm) b) simulation with 0.2λ of astigmatism along the diagonal.

**EQUALLY SPACED LINES**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**LINE DIMENSIONS = 2.0 X 0.5 $\lambda$/N.A.**

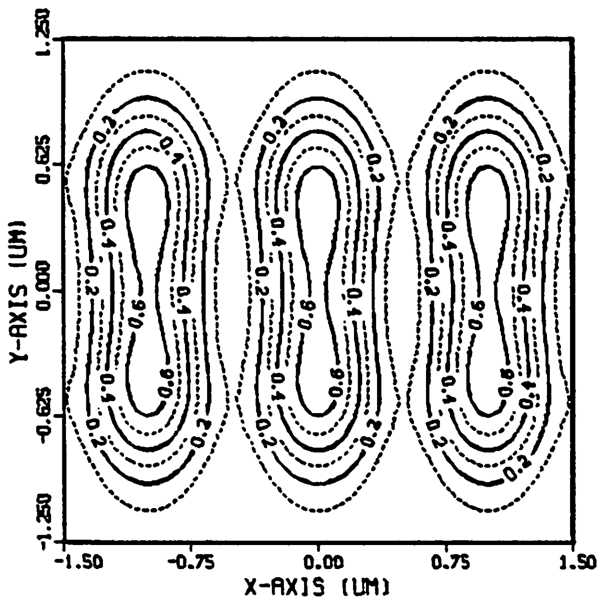
**COMA = 0.20 (0,1), OPD = 0.4$\lambda$**




**COMA = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**COMA = 0.20 (1,0), OPD = 0.4$\lambda$**

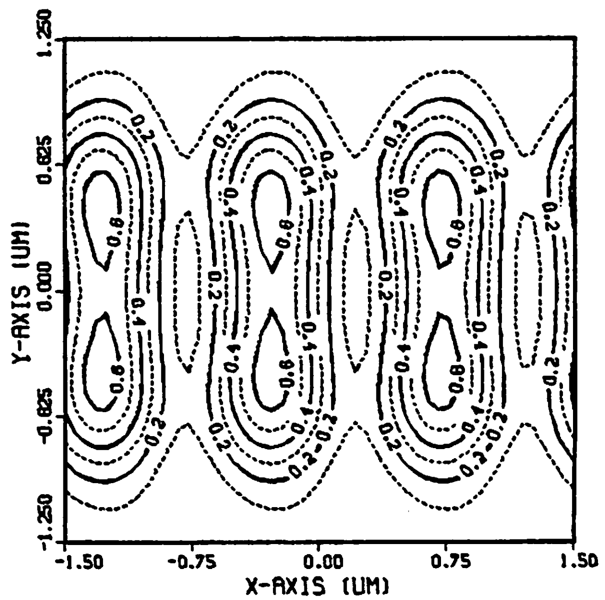
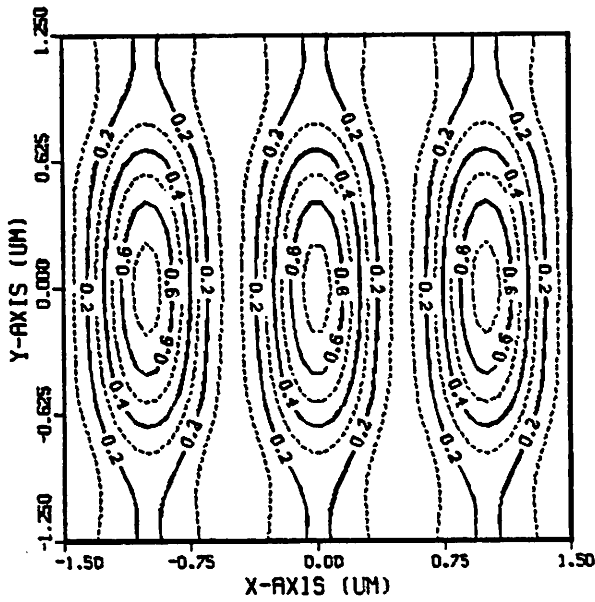


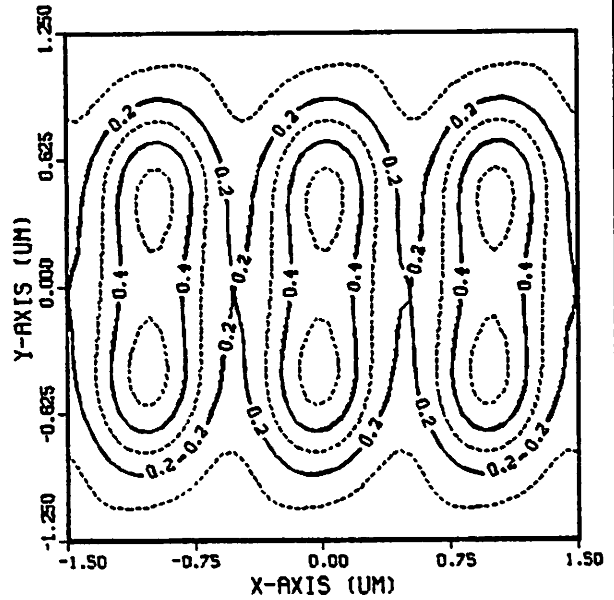

**Figure 25.** Image of a multi-line pattern with 0.4$\lambda$ coma.

**EQUALLY SPACED LINES**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**LINE DIMENSIONS = 2.0 X 0.5 $\lambda$/N.A.**

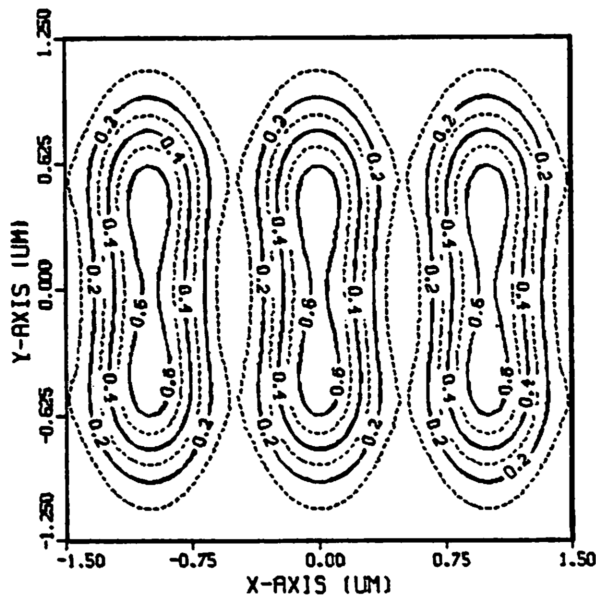
**AST = 0.20 (0,1), OPD = 0.4$\lambda$**




**AST = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**AST = 0.20 (1,0), OPD = 0.4$\lambda$**

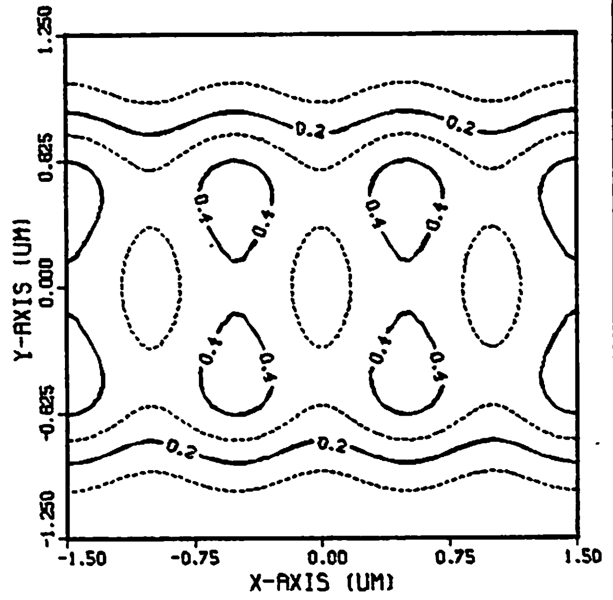
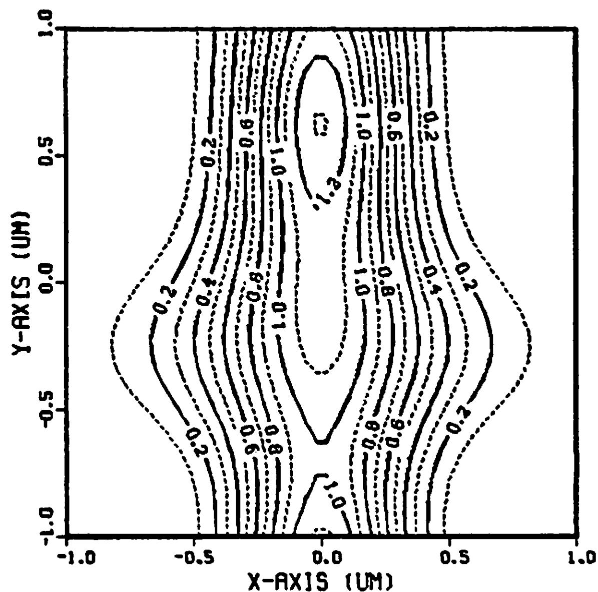


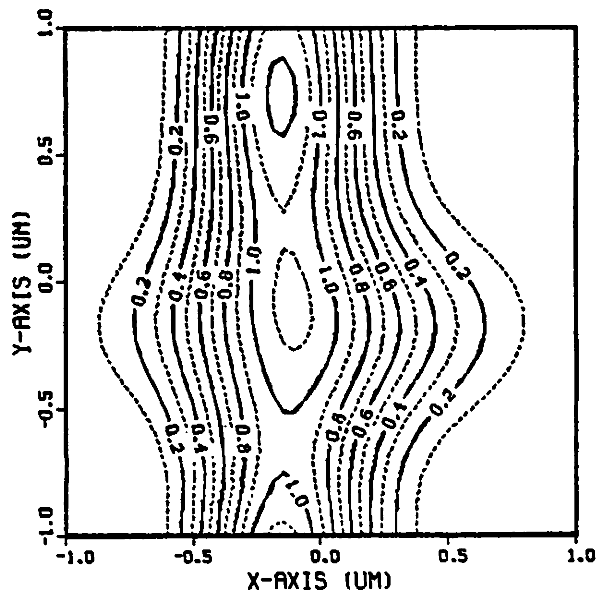

**Figure 26.** Image of a multi-line pattern with 0.4$\lambda$ astigmatism.

**TRANSPARENT FEATURE AND DEFECTS**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**LINE WIDTH = 0.8 $\lambda$/NA, DEFECT SIZE = 0.4 X 0.4 $\lambda$/NA**
**LINE-DEFECT SEPARATION = 0.1 $\lambda$/NA**

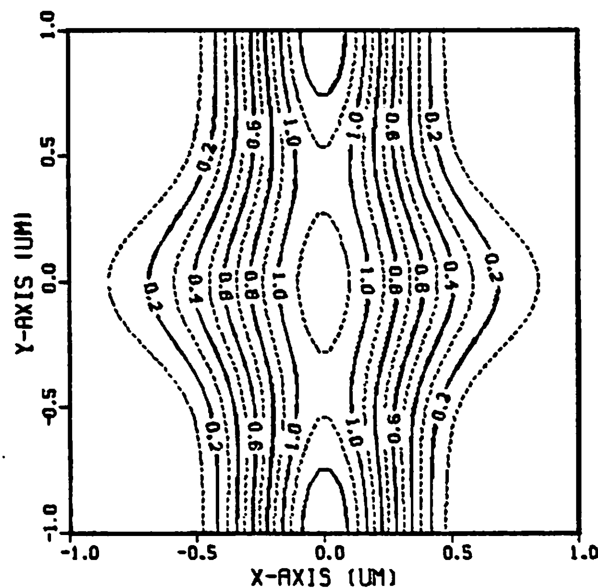
**COMA = 0.20 (0,1), OPD = 0.4$\lambda$**




**COMA = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**COMA = 0.20 (1,0), OPD = 0.4$\lambda$**

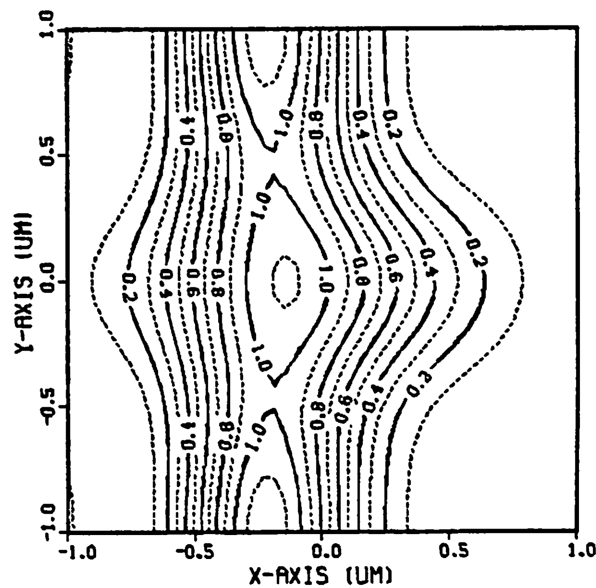
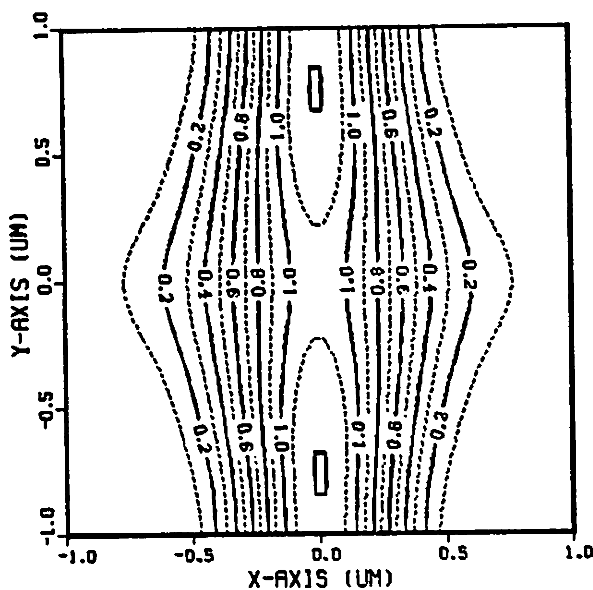


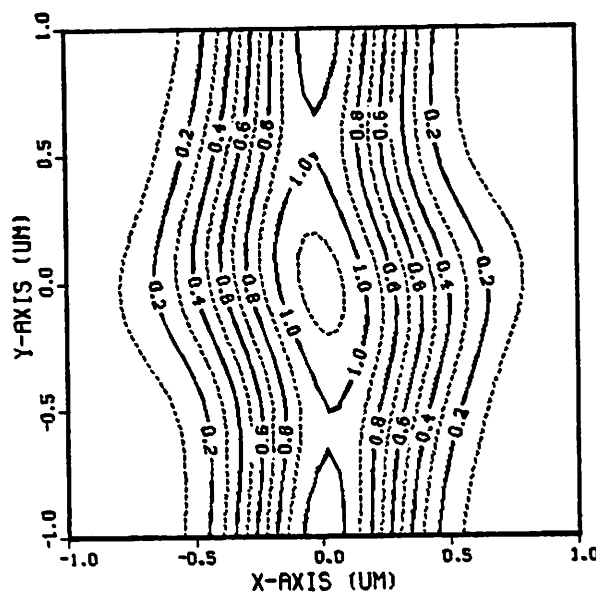

**Figure 27.** Image of a defect-like structure with 0.4$\lambda$ coma.

**TRANSPARENT FEATURE AND DEFECTS**
**$\sigma$ = 0.3, $\lambda$ = 0.5, N.A. = 0.5**
**LINE WIDTH = 0.8 $\lambda$/NA, DEFECT SIZE = 0.4 X 0.4 $\lambda$/NA**
**LINE-DEFECT SEPARATION = 0.1 $\lambda$/NA**

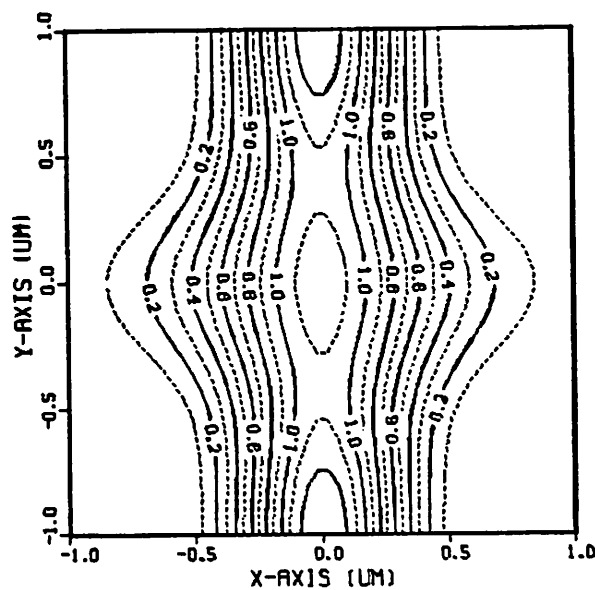
**AST = 0.20 (0,1), OPD = 0.4$\lambda$**




**AST = 0.20 (0.7,0.7), OPD = 0.4$\lambda$**




**NO ABERRATIONS**




**AST = 0.20 (1,0), OPD = 0.4$\lambda$**

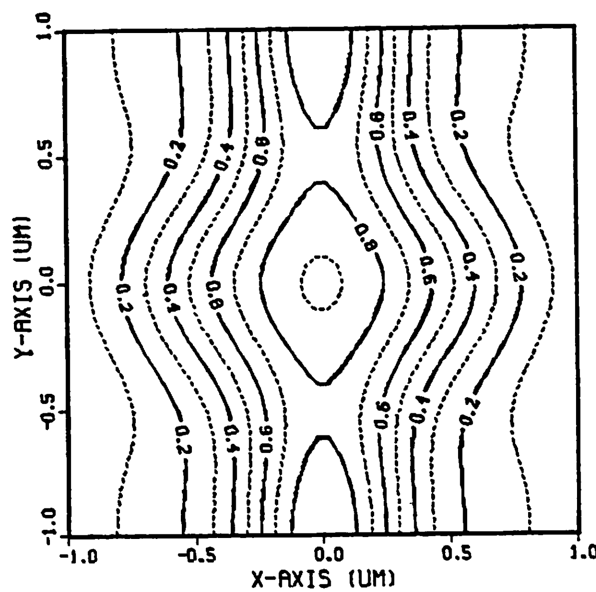
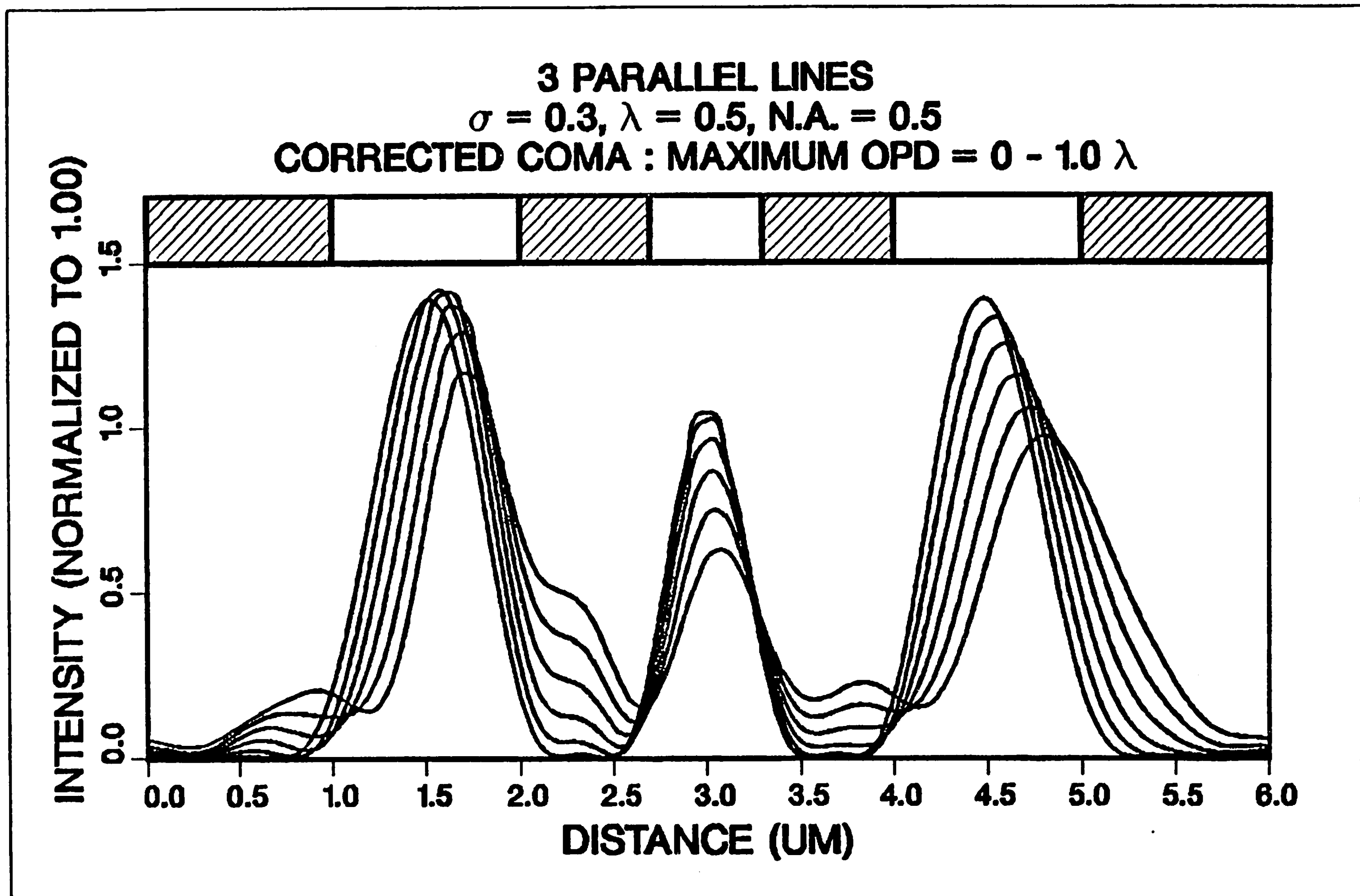


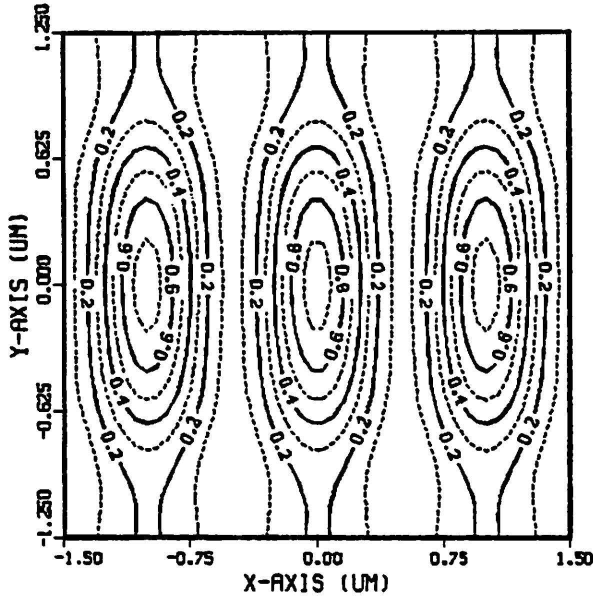

**Figure 28.** Image of a defect-like structure with 0.4$\lambda$ astigmatism.

**Figure 29.** Effect of coma corrected for distortion on the intensity versus horizontal distance for parallel lines.

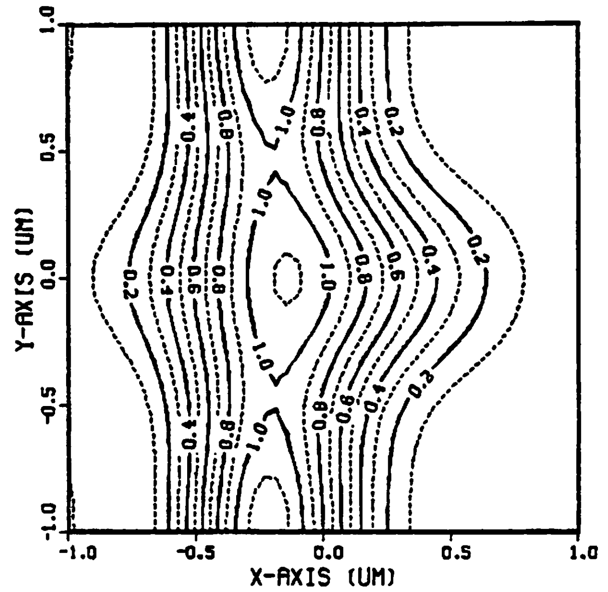


**EFFECTS OF COHERENCE σ ON TEST PATTERNS**
**σ = 0.3 - 0.7, λ = 0.6, N.A. = 0.5**

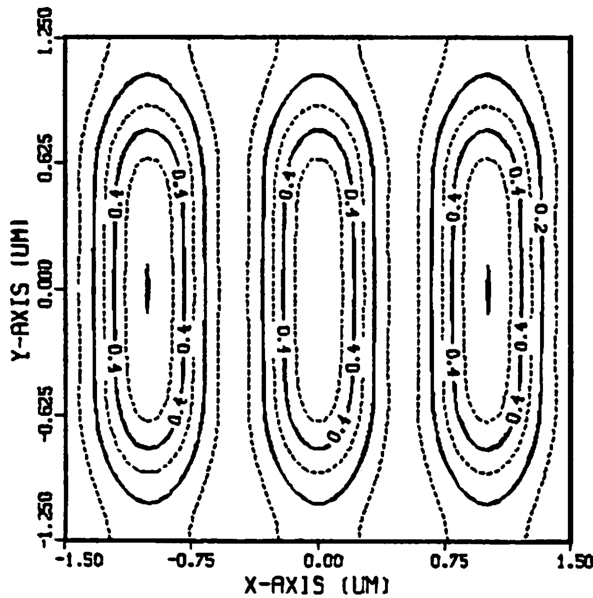
**PARALLEL LINES**
**AST = 0.20 (0,1), σ = 0.3**




**FEATURES AND DEFECTS**
**COMA = 0.20 (1,0), σ = 0.3**




**AST = 0.20 (0,1), σ = 0.7**




**COMA = 0.20 (1,0), σ = 0.7**

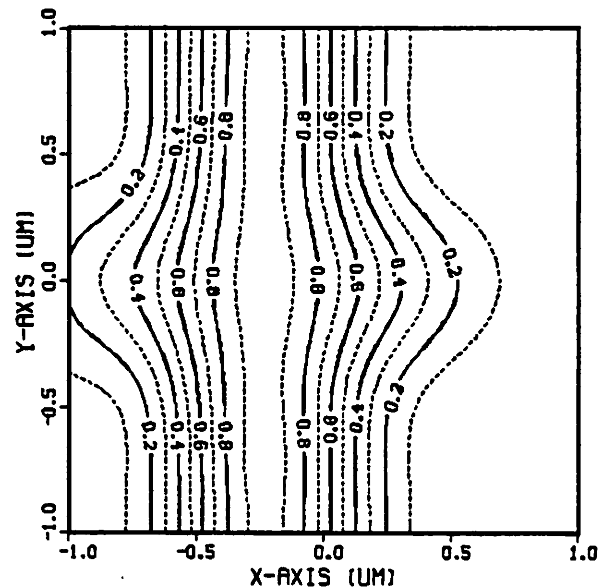
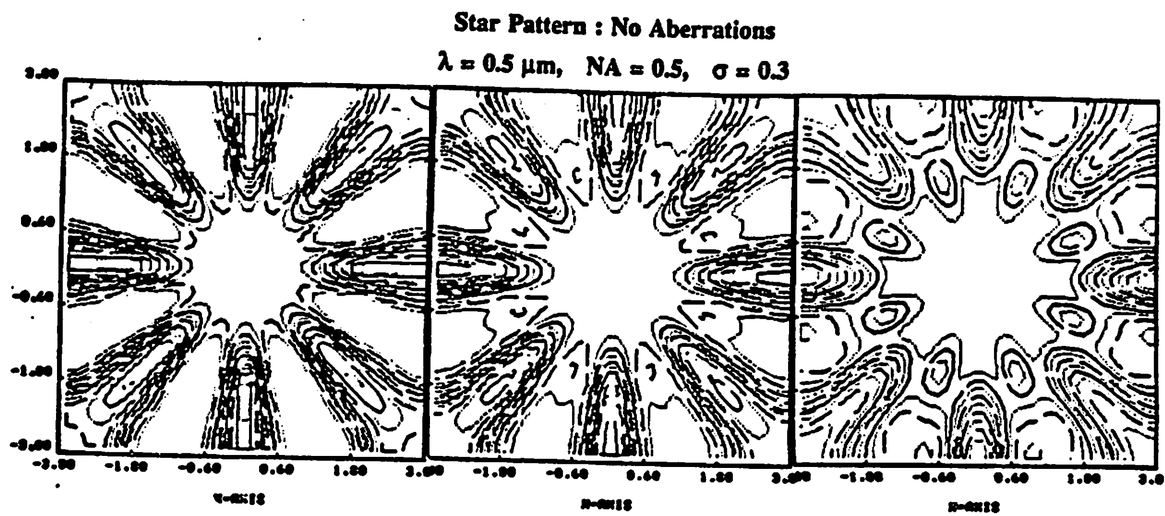
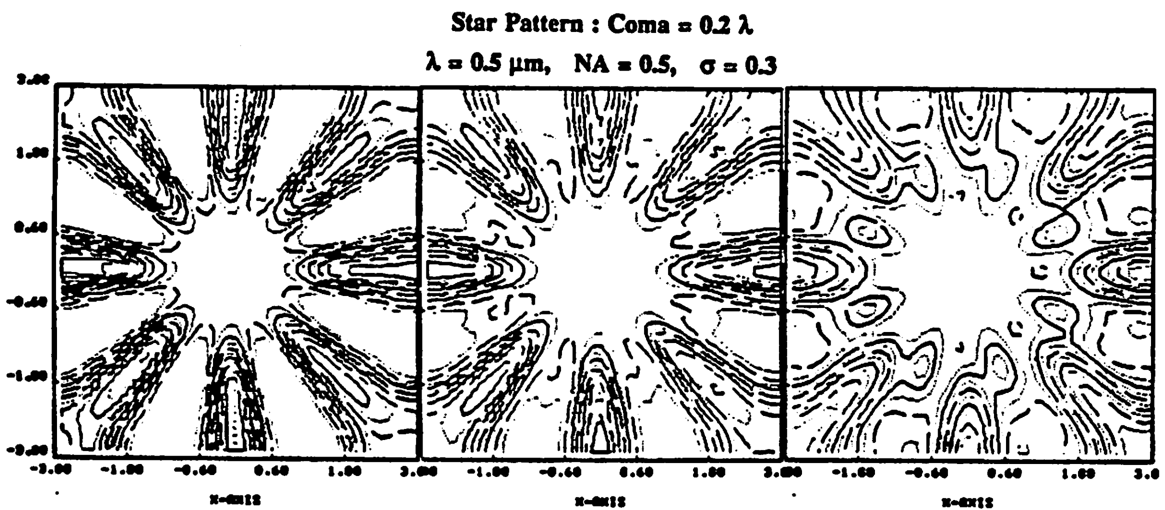


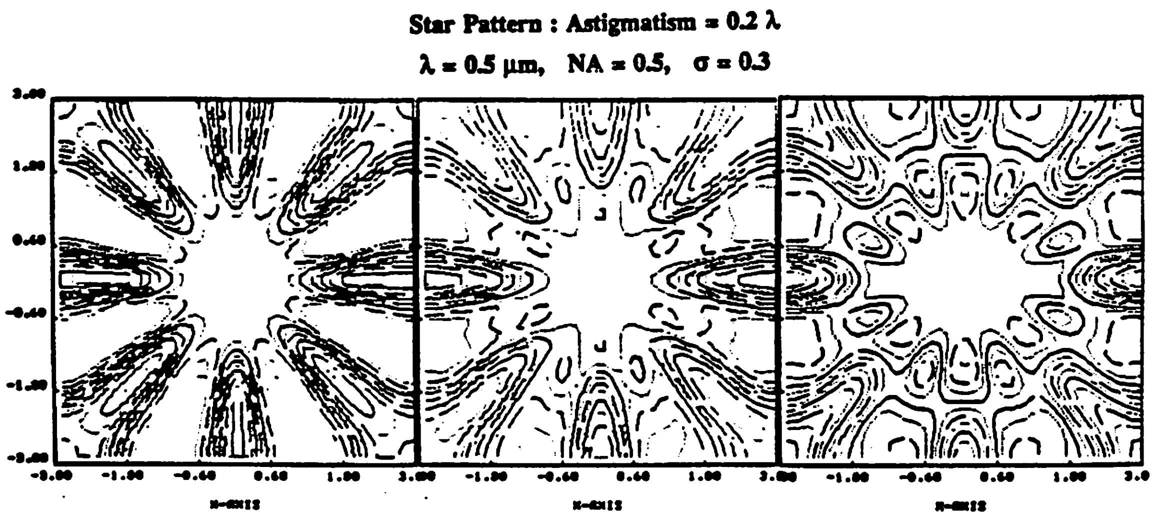

**Figure 30.** Effect of the partial coherence parameter σ on images with aberrations.




No Defocus | 1 R.U. Defocus | 2 R.U. Defocus




No Defocus | 1 R.U. Defocus | 2 R.U. Defocus




No Defocus | 1 R.U. Defocus | 2 R.U. Defocus

**Figure 31.** Simulations of Siemens Star patterns, run at defocus values of 0, 1, and 2 Rayleigh Units.

# APPENDICES

Appendix A - Some Mathematical Details

Appendix B - SPLAT User's Guide

# Appendix A

# Some Mathematical Details

## 1. Summary of the Hopkins formulation

The Hopkins formulation for imaging with partial coherence has been discussed in detail in Born and Wolf [Chapter 10], and is summarized below.

### 1.1. Partial Coherence

A system illuminated with a real physical quasi-monochromatic source consists of many point sources, all of which are mutually incoherent. However, because the source has a finite spectral linewidth, the wave fields due to a single point source at any two points $P_1$ and $P_2$ far from the source will not be independent. In other words, there will be interference between the waves at the two points. This, of course, is the basic definition of partial coherence, which states that for a wave field produced by a finite polychromatic source, a measure of correlation exists between the vibrations at different points $P_1$ and $P_2$ in the field. † The total intensity at any point P then is the summation of all the 2-point correlations all over the field, integrated over the total area of the light source.

### 1.2. Propagation of Light Intensity through the System.

The next building block in the Hopkins formulation comes directly from signals, systems, and Fourier Transforms, and uses the fact that in frequency space, the spectrum of a signal $Y(\omega)$ coming out of a system is equal to the spectrum of the incoming signal $X(\omega)$ multiplied by the frequency response of the system, i.e. $Y(\omega) = H(\omega)\, X(\omega)$. So, if the incoming illumination has an intensity described by a frequency spectrum $J_0(f, g)$, and the wave propagates through an object (mask) with a transmission function $F(f, g)$, and through a medium with a transmission function $K(f, g)$, then both the object and the medium will act as filters on the wave, and the resultant signal will be just the product of the incoming signal and the filter functions in frequency space. Therefore, in frequency space,

† If light at the points $P_1$ and $P_2$ comes from a single monochromatic point source, the correlation between the fields at the two points will be high. This condition is known as "coherent" illumination. In contrast, if the two points each receive light from a different physical source (low correlation) the illumination is described as "incoherent". Partially coherent illumination lies between these two extremes.

$$J_1(f,g) = J_0(f,g)\, F(f,g)\, K(f,g).$$

## 1.3. An Understanding of the Hopkins Formula

The Hopkins formula can be more easily understood by first examining a simplified projection system that only consists of a mask and a source. If the illumination is constant, and if the imaging is perfect, the intensity at the image plane is equal to the electric field multipled by its conjugate, $I = |E|^2$. The electric field, in turn, is proportional to the transmission of the mask, $f(x,y)$, and so at the image plane,

$$I(x,y) = I_0\, f(x,y)\, f^*(x,y) \tag{A.1}$$

where $I_0$ is a constant. In the frequency domain, the above can be written as a convolution,

$$I(f,g) = I_0 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(f'+f, g'+g)\, F^*(f',g')\, df'\, dg' \tag{A.2}$$

The above formula expresses $I(f,g)$ as the sum of contributions from each spatial frequency $(f',g')$ of the object structure. Now, the intensity of the illumination as well as the transmission function of the medium (which contains several lenses) has to be factored in. Each $F(f,g)$ in (A.2) above has to be multiplied by a corresponding $K(f,g)$ which represents the medium of propagation. The $K(f,g)$ functions in turn have to take into account the mutual intensity or the correlation between all the points in the wave field. With this information, it is now possible to understand the basic Hopkins formula below.

$$I(f,g) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} TCC(f'+f, g'+g : f', g')\, F(f'+f, g'+g)\, F^*(f',g')\, df'\, dg' \tag{A.3}$$

$$TCC(f',g':f'',g'') = \iint J(f,g)\, K(f+f', g+g')\, K^*(f+f'', g+g'')\, df\, dg \tag{A.4}$$

$TCC(f',g':f'',g'')$ is a function that describes the interaction between the light intensity (with a Fourier transform $J(f,g)$) and the lens. Comparing Equations (A.2) and (A.3), the reason for the $TCC$ now becomes clear - the $TCC$ is a function that modulates the wave, and takes into account both the diffraction of the wave, and the partial coherence of the system. For this reason, the $TCC$ is known as the Transmission Cross Coefficient. Equation (A.3) does look quite formidable, but luckily, it can be

evaluated with numerical techniques provided that a periodic mask is used.

## 2. Simplifying the Hopkins formula

Let $f(x,y)$ represent the transmittance of the mask at Cartesian coordinates $(x,y)$, with $T_x$ and $T_y$ representing the periods in the x and y-directions respectively. Then, $f(x,y)$ can be represented by a complex summation as follows :

$$f(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C(m,n)\, e^{i2\pi m\frac{x}{T_x}}\, e^{i2\pi n\frac{y}{T_y}} \tag{A.5}$$

where $C(m,n)$ is a complex constant related to the patterns on the mask. Taking the Fourier transform of the above yields

$$F(f,g) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{i2\pi xf}\, e^{i2\pi yg}\, dx\, dy \tag{A.6}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C(m,n)\, e^{i2\pi x\,(f+\frac{m}{T_x})}\, e^{i2\pi y\,(g+\frac{n}{T_y})}\, dx\, dy \tag{A.7}$$

$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C(m,n)\, \delta(f+\frac{m}{T_x})\, \delta(g+\frac{n}{T_y}) \tag{A.8}$$

Now, as mentioned earlier, the image that is formed at the wafer consists of several plane waves which are diffracted into different orders at the mask and then modulated by the lens, which acts as a filter. Each of these plane waves is characterized by 2 spatial frequencies, $f$ and $g$, corresponding to the degree of diffraction in the x and y directions. Each of these plane waves in turn interfere with each other at the lens and at the image plane. Using the expression for $F(f,g)$ developed above in the Hopkins formula (A.3),

$$I(f,g) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} TCC(f'+f,g'+g:f',g') \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C(m,n)\, \delta(f+f'+\frac{m}{T_x})\, \delta(g+g'+\frac{n}{T_y}) \times \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} C^*(p,q)\, \delta(f'+\frac{p}{T_x})\, \delta(g'+\frac{q}{T_y})\, df'\, dg' \tag{A.9}$$

This long equation reduces to a much simpler form due to the presence of the delta functions.

$$I(f,g) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} TCC(\frac{-m}{T_x},\frac{-n}{T_y}:\frac{-p}{T_x},\frac{-q}{T_y})\, C(m,n)\, C^*(p,q) \times \delta(f+\frac{m-p}{T_x})\, \delta(g+\frac{n-q}{T_y}) \tag{A.10}$$

To obtain the image intensity at the wafer, it is necessary to take the inverse Fourier Transform of this latest equation. The image intensity thus has the form

$$I(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(f,g)\, e^{-i2\pi xf}\, e^{-i2\pi yg}\, df\, dg \tag{A.11}$$

$$= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} TCC\left(\frac{-m}{T_x},\frac{-n}{T_y}:\frac{-p}{T_x},\frac{-q}{T_y}\right) C(m,n)C^*(p,q)\, e^{-i2\pi x\frac{p-m}{T_x}}\, e^{-i2\pi y\frac{q-n}{T_y}} \tag{A.12}$$

This final equation can be evaluated numerically once the $TCC$s are known. Rewriting Equation (A.4) with $J(f,g) = J_0(f,g)$,

$$TCC(f',g':f'',g'') = \iint J_0(f,g)\, K(f+f',g+g')\, K^*(f+f'',g+g'')\, df\, dg \tag{A.13}$$

As mentioned before, $J_0(f,g)$ represents the illumination cone, and for critical or Kohler illumination, is a constant within a radius proportional to σ, the partial coherence parameter of the system. $K(f,g)$ is the objective pupil function, and is given by

$$K(f,g) = e^{-i\frac{2\pi}{\lambda}\Phi(f,g)} \qquad f^2+g^2<1 \tag{A.14}$$

where $\Phi(f,g)$ is the wave aberration function which can be expressed as a simple power series in $f$ and $g$. It is now possible to rewrite the expression for the $TCC$ as follows.

$$TCC(f',g':f'',g'') = J_0 \iint e^{-i\frac{2\pi}{\lambda}\Phi(f+f',g+g')}\, e^{i\frac{2\pi}{\lambda}\Phi(f+f'',g+g'')}\, df\, dg \tag{A.15}$$

For non-zero $\Phi(f,g)$, Equation (A.15) can be evaluated using 2-dimensional numerical integration.

### 3. Some Important Fourier Transform Relationships

The 1-Dimensional Fourier Transform is most familiar in the following form :

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\, e^{-i\omega t}\, d\omega \tag{A.16a}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{i\omega t}\, dt \tag{A.16b}$$

With a simple change in variables ( $\omega = 2\pi f$, and $t=x$), the set of equations above can be rewritten as

$$f(x) = \int_{-\infty}^{\infty} F(f) e^{-i2\pi fx}\, df \tag{A.17a}$$

$$F(f) = \int_{-\infty}^{\infty} f(x) e^{i2\pi fx} dx \tag{A.17b}$$

The 2-Dimensional Fourier Transform is simply an extension of this 1-Dimensional Transform pair, and thus can be written out as follows.

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(f,g)\, e^{-i2\pi xf}\, e^{-i2\pi yg}\, df\, dg \tag{A.18a}$$

$$F(f,g) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{i2\pi xf}\, e^{i2\pi yg}\, dx\, dy \tag{A.18b}$$

Another important relationship in the Fourier Transform states that the Fourier Transform of a complex exponential is simply a delta function. The set of equations developed below is used in going from Equation (A.7) to (A.8).

$$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ifx}\, df \tag{A.19}$$

$$\delta(2\pi(x+b)) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i2\pi f(x+b)}\, df \tag{A.20}$$

$$2\pi\delta(2\pi(x+b)) = \delta(x+b) = \int_{-\infty}^{\infty} e^{-i2\pi f(x+b)}\, df \tag{A.21}$$

# Appendix B

# SPLAT User's Guide

# May, 1988

# SPLAT
## (Simulation of Projection Lens Aberrations via TCCs)

*Kenny K. H. Toh*
*(Professor Andrew R. Neureuther)*

Department of Electrical Engineering and Computer Sciences,
and Electronics Research Laboratory,
University of California, Berkeley, CA 94720
(415)642-8897

*ABSTRACT*

Basic studies of projection printed images have been made using a two-dimensional optical image simulation program associated with SAMPLE. With the original program, "2D", images of periodic masks could be simulated with varying wavelength $\lambda$, numerical aperture NA and spatial coherence $\sigma$.

Now, the program has been rewritten to include the effects of arbitrary primary lens aberrations (including defocus). In addition, the transmission cross-coefficients (TCCs) associated with a particular mask size and illumination condition can be saved and reused with different mask patterns.

For analysis of the resulting images, 2-D or 3-D image intensity data points can be saved. The 3-D data points can be plotted on a contour plot using a graphics package such as DISSPLA(ISSC). Alternatively, a contour plot routine included here as a separate program can also be used to obtain plots on a HP2648 (or an equivalent HP terminal), or on an Apple Laserwriter. The 2-D data points can be plotted using the SAMPLE plotting package, or can be fed into SAMPLE for simulation of processing steps based on that intensity profile.

This program has been compiled and run successfully on a Vax 11/780, an IBM 3090 mainframe, as well as on an IBM AT.

May 16, 1988

# SPLAT : AN OPTICAL IMAGING PROGRAM

---

"SPLAT" is a FORTRAN program, based on the Hopkins theory of partially coherent imaging, that simulates two-dimensional projection-printing with partial coherence. Transmission cross-coefficients are used to calculate the light intensity at the image plane relative to the [uniform] intensity at the mask. This program uses the same algorithms as SAMPLE, but is extended to handle two-dimensional objects as well as primary lens aberrations. The mask is specified as a set of rectangles and triangles whose size, position and relative transmittance are specified by the user. This allows the user to examine the effects of two-dimensional structures such as elbows and squares. Once the intensity has been computed at the image plane, the image intensity along a line can then be extracted and fed into SAMPLE for further processing. Alternatively, the image intensity in a specified rectangle can be extracted and examined through the use of a contour plotting package.

At present, this program can handle any combination of the five primary lens aberrations : Spherical Aberration, Coma, Astigmatism, Curvature/Defocus and Distortion (For descriptions of these aberrations, see Appendix II). To simplify and speed up the calculations, the program makes an even periodic extension in both x and y directions of the user-specified mask. Even so, care must be taken to limit the size of the mask used, because the computation time is proportional to the square of the area of the mask. A practical maximum, where the user would wait approximately 5 minutes for results, would be an unfolded(total) mask size of 4.0 $\lambda$/NA x 4.0 $\lambda$/NA.

This program uses a fairly primitive parser routine to read its input statements - this routine, in essence, recognizes numbers and characters, but not keywords. Statements can be separated by semicolons or placed on separate lines. A statement may be continued on more than one line by placing an ampersand (&) as the first character of each continuation line. All characters on the input line are ignored except numbers, quoted strings, semicolons, an ampersand in the first column, or pound signs. The pound sign (#) indicates that the remainder of the line is a comment to be ignored by the parser.

Example :

All of the following inputs cause the same action (set light source wavelength $\lambda$ = 0.436 microns and numerical aperture NA = 0.28) :

```
# Example 1
2 0.436; 3 0.28;

# Example 2
Statement 2 : wavelength = 0.436 microns
Statement 3 : na = 0.28;

# Example 3
Statement 2 : Set wavelength
&             to 0.436 microns;
Statement 3 : 0.28  # Set numerical aperture to 0.28
```

Note :

For interactive use of the program, semicolons should be used as statement separators. Without the semicolon, the program must read an extra line to see if it is a continuation line before acting on that input statement.

□

## INPUT STATEMENTS FOR SPLAT

---

A list of the input statements recognized by SPLAT is given below. Arguments shown in brackets are optional (the brackets are not necessary when using the program). Strings (such as filenames) must be inside single or double quotes. Note that the word STATEMENT at the beginning of the input statement is not necessary.

[STATEMENT] 1 printlevel

STATEMENT 1 sets the level of diagnostic output generated by the program. PRINTLEVEL=1 produces no diagnostic output, while PRINTLEVEL=2 causes the program to echo the input lines to the output. PRINTLEVEL=3, on the other hand, is primarily for interactive/diagnostic use, and causes the program to report when statement execution is completed, by displaying the input parameters. The default value of PRINTLEVEL is 1.

[STATEMENT] 2 wavelength

This statement sets the wavelength, in micrometers, of the light used to illuminate the mask. Currently, the program is set up to accept only a single wavelength. The default is WAVELENGTH = 0.436 μm.

[STATEMENT] 3 numerical.aperture

The imaging system is projection-type, with an imaging lens numerical aperture equal to NUMERICAL.APERTURE. Default NUMERICAL.APERTURE is 0.28.

[STATEMENT] 4 defocus

The image can be calculated at a plane other than the plane of best focus. The distance from the plane of best focus, DEFOCUS, is measured in micrometers, and defaults to 0.0 μm. Positive DEFOCUS is defined as being below the gaussian plane, while negative DEFOCUS is above it.

[STATEMENT] 5 sigma

STATEMENT 5 sets the partial coherence factor, SIGMA, of the imaging system. Only values of partial coherence that lie between 0.0 (full coherence) and 1.0 (partial coherence) are accepted by the program. SIGMA defaults to 0.7, a value common to most projection printers.

[STATEMENT] 6 xlength ylength [transmittance [scale]]

The working area is the size of the field which is to be imaged, and is specified by XLENGTH and YLENGTH in micrometers. This specified area is actually only the first quadrant in Cartesian coordinates. The total/unfolded area is bounded by the coordinates (-XLENGTH,-YLENGTH), (-XLENGTH,YLENGTH), (XLENGTH,YLENGTH), (XLENGTH,-YLENGTH), and a periodic extension in both the X and Y directions is assumed. The optional TRANSMITTANCE is the initial transmittance of the mask (usually either 1 for transparent or 0 for opaque). The image calculation time is approximately proportional to the square of the imaged area, and to avoid long computation times, field sizes with total areas greater than 16 micrometers squared should be avoided. One way to do this is to take advantage of symmetry - see STATEMENT 7 for details. The default for TRANSMITTANCE is 0, while XLENGTH and YLENGTH both default to 2.0 micrometers. SCALE is a positive real number that can be used to scale the mask. This scale factor will also apply to STATEMENTs 7 and 8 that come after STATEMENT 6.

[STATEMENT] 7 xcoord ycoord xlen ylen transmittance [phase]

This statement is used to add a rectangular feature to the mask. Rectangles may overlap other rectangles - it is the user's responsibility to make sure no part of the mask has transmittance greater than 1 or less than 0.

XCOORD = x coordinate of lower left corner of rectangle
YCOORD = y coordinate of lower left corner of rectangle
XLEN = length of rectangle in x direction
YLEN = length of rectangle in y direction
TRANSMITTANCE= transmittance of the rectangle relative to the current transmittance of the mask where it is to be placed.
PHASE = phase angle (normally 0 degrees). This option is used for phase shifted masks.

As an example, to specify a 2 μm x 2 μm transparent square in a 5 μm x 5 μm opaque mask, the following statements could be used :

STATEMENT 6 : 5 um x 5 um at 0 transmittance;
STATEMENT 7 : (2.0,2.0) 2.0 um x 2.0 um at 1 transmittance;

However, symmetry can and should be utilized to reduce the computation time, by locating the center of the rectangle at the origin of the coordinate system in the following manner:

STATEMENT 6 : 2 um x 2 um at 0 transmittance;
STATEMENT 7 : (0.0,0.0) 1.0 um x 1.0 um at 1 transmittance;

In the last two input statements above, a 1 μm x 1 μm transparent square is defined in a 2 μm x 2 μm opaque mask. Because the program automatically makes even periodic extensions in the x and y directions, the region specified by the user is actually only the first quadrant in the cartesian coordinate system; the other three quadrants are the mirror images of the first, reflected across the x and y-axes respectively. Therefore, the 1 μm x 1 μm square defined above is actually only part of a 2 μm x 2 μm square, with the center of symmetry of that larger square located at the origin. In a similar manner, an opaque square on a transparent mask can be defined using the statements below, where the transmittance is 1 for the background, and -1 for the square.

STATEMENT 6 : 2 um x 2 um at 1 transmittance;
STATEMENT 7 : (0.0,0.0) 1.0 um x 1.0 um at -1 transmittance;

[STATEMENT] 8 xcoord ycoord xlen ylen transmittance; or
[STATEMENT] 8 x1 y1 x2 y2 x3 y3 transmittance;

Add a triangular aperture to the mask. Triangles may overlap other triangles and rectangles - it is the user's responsibility to make sure no part of the mask has transmittance greater than 1 or less than 0. The triangle can be specified in two ways - as a right triangle defined by the corner (right angle) point, base and height; or as a general triangle defined by three points. Again, even periodic extensions in the x and y directions are assumed.

XCOORD = x coordinate of corner of right triangle
YCOORD = y coordinate of corner of right triangle
XLEN = base length of right triangle (may be negative to flip triangle)
YLEN = height of right triangle (may be negative to flip triangle)
X1 = x coordinate of point defining general triangle
Y1 = y coordinate of point defining general triangle
X2 = x coordinate of point defining general triangle
Y2 = y coordinate of point defining general triangle
X3 = x coordinate of point defining general triangle
Y3 = y coordinate of point defining general triangle
TRANSMITTANCE= transmittance of the triangle relative to the current transmittance of the mask where it is to be placed.

**Example : Mask Specification and Symmetry**

```
Input Statements :
        Statement 6 : XM x YM@0;
        Statement 7 : (0,0) (XL,YL) @1;

    (-XM,YM)                    (XM,YM)                    (3XM,YM)
       ##########################################################
       #...........|...........#...........|...........#
       #...........|...........#...........|...........#
       #...........|..(XL,YL)...#...........|...........#
       #......111111|111111......#......111111|111111......#
       #......1     |     1......#......1     |     1......#
       #----------(0,0)----------#-----------|-----------#
       #......1     |     1......#......1     |     1......#
       #......111111|111111......#......111111|111111......#
       #..(-XL,-YL) |...........#...........|...........#
       #...........|...........#...........|...........#
       #...........|...........#...........|...........#
    (-XM,-YM)#################(XM,-YM)##################(3XM,-YM)
       #...........|...........#...........|...........#
       #...........|...........#...........|...........#
       #...........|...........#...........|...........#
       #......111111|111111......#......111111|111111......#
       #......1     |     1......#......1     |     1......#
       #----------(0,-2YM)-------#-----------|-----------#
       #......1     |     1......#......1     |     1......#
       #......111111|111111......#......111111|111111......#
       #........... |...........#...........|...........#
       #...........|...........#...........|...........#
       #...........|...........#...........|...........#
    (-XM,-3YM)################(XM,-3YM)################(3XM,-3YM)

Opaque areas      = "....."
Transparent areas = "1   1"
Axes              = "--|--"
```

[STATEMENT] 9    nx ny [mode [xllc yllc xlen ylen [difflag]]] ['filename'];

STATEMENT 9 is used to calculate the transmittance profile of the mask. The transmittance at any point is the sum of the working area(mask) transmittance (see STATEMENT 6) and the transmittances of all rectangles or triangles defined by STATEMENTs 7 and 8 that cover that point. The output file consists of a list of transmittances at points on a grid defined by NX and NY. The default is to calculate for the entire working area, but the user can specify a smaller (or larger) area. MODE defines the type of output obtained from this statement. MODE = 0 produces output to a specified file, MODE = 1 sends a crude contour plot to the terminal, while MODE = 2 does both.

NX = number of divisions along x axis (default = 20)
NY = number of divisions along y axis (default = 20)
MODE = type of output desired
XLLC = x coordinate of lower left corner of rectangle to be plotted
YLLC = y coordinate of lower left corner of rectangle to be plotted
XLEN = length of rectangle in x direction
YLEN = length of rectangle in y direction
DIFFLAG = 1 to limit spatial frequencies (normally not used)
FILENAME= name of file into which to store the calculated intensities.

[STATEMENT] 10 [mode [force]] ['filename'];

STATEMENT 10 orders the program to calculate the transmission cross-coefficients (TCCs) as well as the Fourier coefficients of the image intensity profile. This statement can take a long, long time to execute. MODE = 1 stores the TCCs in *binary* form in the file FILENAME. FORCE is an integer flag which forces the program to choose which of the 3 integration routines should be used for TCC calculation. FORCE = 0 is the default, FORCE = 1 uses a 1-dimensional integration routine, while FORCE = 2 uses a 2-dimensional integration scheme. This last option, FORCE, should be used only for diagnostics.

[STATEMENT] 11 [xllc yllc xlen ylen] ['filename'];

Calculate the intensity at the image plane. The parameters are :

XLLC = x coordinate of lower left corner of rectangle to be plotted
YLLC = y coordinate of lower left corner of rectangle to be plotted
XLEN = length of rectangle in x direction
YLEN = length of rectangle in y direction
FILENAME= name of file into which to store the calculated intensities.

This statement calculates the image intensity within a rectangle specified by XLLC, YLLC, XLEN, YLEN. The intensities are calculated at each point of a 50 by 50 equally spaced grid within that rectangle. This output file may be used with a 3-dimensional plotting package to provide either a 3-dimensional intensity profile or an intensity contour plot. (See Appendix I)

[STATEMENT] 12 'filename'

Save the Fourier coefficients that describe mask transmittance and image intensity. This input line will produce a relatively large (approx. 30 kbytes) *binary* output file, which contains the Fourier coefficients as well as the imaging system parameters - numerical aperture, wavelength, coherence factor, mask size. The coefficients can be re-loaded with STATEMENT 13 to generate more plot files of the same mask pattern.

[STATEMENT] 13 'filename'

STATEMENT 13 is used to load the Fourier mask and image coefficients previously saved with STATEMENT 12. Therefore, for a given set of imaging system parameters (N.A., wavelength, sigma, mask size + pattern), the intensity profile need only be calculated once with STATEMENT 10. STATEMENT 12 saves the data computed by STATEMENT 10 for later use, and STATEMENT 13 reloads this data. Note that the TCCs saved via STATEMENT 10 can be used for any mask pattern, as long as the field size remains constant. On the other hand, STATEMENT 12-13 saves the Fourier coefficients only, so these can be used only for one particular mask pattern.

[STATEMENT] 14 xi yi xf yf [npts [mode]] ['filename'];

Calculate the image intensity profile along the line joining points (XI,YI) and (XF,YF). The number of points along that profile and is optionally specified by NPTS, which defaults to 50.

The intensity profile is output in the same format as a SAMPLE f77punch7 file. The 'x' values in the file represent the distance along the line from the point (XI,YI), while the 'y' values represent the light intensity normalized to 1.00. Again, MODE is an integer flag, normally 0, used to specify the format of the intensity profile. MODE = 1 outputs the intensities with 5 decimal spaces (as compared to 3 with MODE = 0). This option is particularly useful for analysis of small patterns such as defects, where the intensity is very low.

XI = x coordinate of initial plot point
YI = y coordinate of initial plot point
XF = x coordinate of final plot point
YF = y coordinate of final plot point
NPTS= number of points (default is 50)
MODE = accuracy flag (defaults to 0)

[STATEMENT] 15 xllc yllc xlen ylen [xsiz ysiz [maj [min [lbls [mode]]]]] ['title']

STATEMENT 15 produces a plot of the intensity contours using the DISSPLA(ISSC) graphics package. Essentially, the data produced by this statement is similar to that produced by STATEMENT 11, except that the output data here is formatted to conform with the input requirements of DISSPLA. STATEMENT 15 was developed for use at UC Berkeley for use on an IBM 3090 mainframe.

XLLC = x coordinate of lower left corner of area to be plotted
YLLC = y coordinate of lower left corner of area to be plotted
XLEN = length of area in x direction
YLEN = length of area in y direction
XSIZ = horizontal length of actual plot in inches
YSIZ = vertical length of actual plot in inches
MAJ = major (thick line) contour increment (default 0.20)
MIN = minor (dotted line) contour increment (default 0.05)
LABELS = request no contour labeling if 0
MODE = 0 for a plotfile, 1 for TEK4115, 2 for printer/non-graphics.
TITLE = plot title to be printed on plot page

WARNING:

The FORTRAN code containing STATEMENT 15 can only be compiled on an environment that runs DISSPLA. For this reason, STATEMENT 15 is not included as part of the compilable code, and the instructions in Appendix I.1 should be followed to include this statement as part of the code. STATEMENT 15 will normally be a blank statement (i.e. no commands implemented). For implementation of this subroutine on a DISSPLA environment, consult your local DISSPLA expert.

[STATEMENT] 16 empty

[STATEMENT] 17 empty

[STATEMENT] 18 empty

[STATEMENT] 19 'filename'

STATEMENT 19 loads the transmission cross-coefficients(TCC) saved from STATEMENT 10. These TCCs can be reused for different mask patterns because the TCCs are only dependent on the mask size and not upon the patterns contained within it. Using these TCCs a new image can be recomputed very speedily (in approx. 38 cpu seconds).

[STATEMENT] 20 xpos ypos

This statement is used to specify the coordinates of the object, relative to the lens axis. With this, it is possible to examine features at different points of the field. This statement is used primarily to determine how the object is situated (perpendicular, parallel, etc) relative to the lens axis. The object coordinates input through this statement will be normalized such that $XPOS^2 + YPOS^2 = 1$. For example, (XPOS,YPOS) = (1,0) means that the portion of the mask currently being simulated is located on the X-axis of the field. If the mask consists of vertical lines, then it is possible to say that the pattern is perpendicular to the lens axis. In contrast, if (XPOS,YPOS) = (0,1), the vertical pattern would be parallel to the lens axis. This distinction has been found to be important for aberrations such as coma and astigmatism, which are orientation-dependent.

[STATEMENT] 21 spherical_aberration

[STATEMENT] 22 coma

[STATEMENT] 23 astigmatism

[STATEMENT] 24 curvature

[STATEMENT] 25 distortion

Statements 21-25 can be used to specify any combination of primary lens aberrations. The numbers specified (COMA, ASTIGMATISM, etc) are used as multipliers to a predefined power series (Zernike polynomials), and as such, can be either positive or negative real values. It is easier to think of these multipliers as a measure of the deviations, measured in fractions of a wavelength, from the ideal spherical wavefront. In general, the deviation is related to the multiplier as $\Delta\lambda$ = multiplier / wavelength. For example, if $\lambda = 0.5$ µm, and COMA = 0.1, this would mean that the wavefront has a maximum deviation of $0.2\lambda$ from the ideal spherical wavefront.

□

## SPLAT EXAMPLES

Recall that in these, and all other examples which follow, all characters on the input line are ignored except for numbers, quoted strings, semicolons, the ampersand (&) in the first column, and the pound(#) sign. Furthermore, the word "Statement" is optional. Here, on each page, the input file is presented first, and followed by the line-printer output that results from that input file.

**Example 1 : Input File**

```
# Projection Lithography using SPLAT  -- test1
#
# 1.0um x 1.0um opaque diamond in a transparent mask.
#
Statement 1 : Printlevel 3              ;# set print level
Statement 2 : lambda = 0.436 um         ;# wavelength
Statement 3 : NA     = 0.28             ;# numerical aperture
Statement 4 : Defocus= 0.0 um           ;# defocus
Statement 5 : Sigma  = 0.7              ;# coherence factor
Statement 6 : mask   = 2um x 2um
&             at 1 transmittance        ;# define the working area
Statement 8 : cutout = (0.0, 0.0)        # triangular opaque pattern
&             0.5 x 0.5 at -1           ;# define the mask openings
Statement 10:                           ;# calculate the Fourier coeffs
Statement 14: intensity (-2,0) .. (2,0)
&             to 'test1.plot'           ;# 2-D intensity profiles
Statement 0 : end;
```

---

**Example 1 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                        5/10/87 -- KT

Trial  1:  Print level=  3
Trial  2:  Lambda      =  0.4360 microns
Trial  3:  N.A.        =  0.2800
Trial  4:  Defocus     =  0.     microns  [ 0.    Rayleigh Units ]
Trial  5:  Sigma       =  0.7000
Trial  6:  Field size =  2.0000 x  2.0000 @ 1.0000  scale =1.00
#      8:  Add triangle ( 0.   , 0.   )x( 0.500, 0.500) @-1.000
Trial  8:  Triangle   = ( 0.   , 0.   ) ( 0.   , 0.500) ( 0.500, 0.   )
Trial 10:  Calculate image Fourier Transform
#     10:  Symmetry :  T( f1, g1, f2, g2) =
#     10:              T(-f1,-g1,-f2,-g2)
#     10:              T( f1,-g1, f2,-g2)
#     10:              T(-f1, g1,-f2, g2)
#     10:              T( f2, g2, f1, g1)
#     10:  TCC computation time  =   4.76667 sec.
#     10:  Imaged with  4 by  4 harmonics
#     10:  TCC calls:     361  zeros:     520
Trial 14:  3-decimal plot line : ( -2.000,  0.   )..( 2.000,  0.   )
           50 points saved in "test1.plot"
Trial 0 :  End of session
User Time (CPU) =   10.00000 sec,  System Time =   1.80000 sec.
```

**Example 2 : Input File**

```
#   Projection Lithography using SPLAT  -- test2
#
#   0.4um square in close proximity to a 0.8um line, with 1 um defocus.
#
Trial 1 : Printlevel 3              ;# set print level
Trial 2 : lambda = 0.5 um           ;# wavelength
Trial 3 : NA     = 0.5              ;# numerical aperture
Trial 4 : Defocus= 1.0 um           ;# defocus
Trial 5 : Sigma  = 0.7              ;# coherence factor
Trial 6 : mask   = 2um x 2um
&         at 0 transmittance        ;# define the working area
Trial 7 : cutout = (0.0, 0.0)
&         0.4 x 2.0 at 1;
Trial 7 : cutout = (0.5, 0.0)
&         0.4 x 0.2 at 1            ;# define the mask openings
Trial 10:                           ;# calculate the Fourier coeffs
Trial 14: intensity (0,2) .. (2,2)
&         to 'f77line';
Trial 14: intensity (0,0) .. (2,0)
&         to 'f77combined'          ;# intensity profiles
Trial 0 : end;
```

---

**Example 2 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                    5/10/87 -- KT

Trial  1: Print level=  3
Trial  2: Lambda     = 0.5000 microns
Trial  3: N.A.       = 0.5000
Trial  4: Defocus    = 1.0000 microns  [ 1.0000 Rayleigh Units ]
Trial  5: Sigma      = 0.7000
Trial  6: Field size = 2.0000 x 2.0000 @ 0.     scale =1.00
Trial  7: Cutout     =( 0.    , 0.   )x( 0.4000, 2.0000) @ 1.0000 < 0.   >
Trial  7: Cutout     =( 0.5000, 0.   )x( 0.4000, 0.2000) @ 1.0000 < 0.   >
Trial 10: Calculate image Fourier Transform
#     10: Symmetry :  T( f1, g1, f2, g2) =
#     10:             T(-f1,-g1,-f2,-g2)
#     10:             T( f1,-g1, f2,-g2)
#     10:             T(-f1, g1,-f2, g2)
#     10:        conjg[T( f2, g2, f1, g1)]
#     10: TCC computation time  = 34.58333 sec.
#     10: Imaged with  6 by  6 harmonics
#     10: TCC calls:   1874  zeros:   1906
Trial 14: 3-decimal plot line : ( 0.   , 2.000)..( 2.000, 2.000)
          50 points saved in "f77line   "
Trial 14: 3-decimal plot line : ( 0.   , 0.   )..( 2.000, 0.   )
          50 points saved in "f77combine"
Trial 0 : End of session
User Time (CPU) =  48.00000 sec,  System Time =  2.61667 sec.
```

**Example 3 : Input File**

```
# The following two examples consist of two separate SPLAT input files. The
# first (Example 3) calculates the image for a pair of opaque elbows and
# stores the calculated coefficients in the file named "test3.cof".
# The second (Example 4 on the following page) reads the coefficients back
# and calculates the intensity profiles along a diagonal through the elbow
# corners.
#
#
# 4 um. mask with 0.75 um. wide elbows separated by 0.5 um. -- test3
#
stmt 1: print = 3
stmt 2: lambda = 0.436 um
stmt 3: na     = 0.28
stmt 4: defoc  = 0.0 um
stmt 5: s      = 0.7
stmt 6: mask   = 4x4 @0
stmt 7: cutout = (0.75,0.50) 0.75 x 2.75 @1
stmt 7: cutout = (1.50,2.50) 2.00 x 0.75 @1
stmt 7: cutout = (2.00,0.50) 0.75 x 1.50 @1
stmt 7: cutout = (2.75,1.25) 0.75 x 0.75 @1
stmt 10
stmt 12: save  = 'test3.cof'
end    0
```

---

**Example 3 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                         5/10/87 -- KT

Trial  1: Print level= 3
Trial  2: Lambda     = 0.4360 microns
Trial  3: N.A.       = 0.2800
Trial  4: Defocus    = 0.     microns  [ 0.    Rayleigh Units ]
Trial  5: Sigma      = 0.7000
Trial  6: Field size = 4.0000 x 4.0000 @ 0.     scale =1.00
Trial  7: Cutout     =( 0.7500, 0.5000)x( 0.7500, 2.7500) @ 1.0000 < 0.  >
Trial  7: Cutout     =( 1.5000, 2.5000)x( 2.0000, 0.7500) @ 1.0000 < 0.  >
Trial  7: Cutout     =( 2.0000, 0.5000)x( 0.7500, 1.5000) @ 1.0000 < 0.  >
Trial  7: Cutout     =( 2.7500, 1.2500)x( 0.7500, 0.7500) @ 1.0000 < 0.  >
Trial 10: Calculate image Fourier Transform
#     10: Symmetry : T( f1, g1, f2, g2) =
#     10:            T(-f1,-g1,-f2,-g2)
#     10:            T( f1,-g1, f2,-g2)
#     10:            T(-f1, g1,-f2, g2)
#     10:            T( f2, g2, f1, g1)
#     10: TCC computation time = 58.56667 sec.
#     10: Imaged with 8 by 8 harmonics
#     10: TCC calls: 5152 zeros: 5505
Trial 12: Fourier Coefficients saved in "test3.cof "
Program execution terminated.
User Time (CPU) = 69.41666 sec, System Time = 1.78333 sec.
```

**Example 4 : Input File**

```
#       Read the coefficient file produced by example 3 and get
#       a SAMPLE style intensity plotfile along a diagonal through
#       the two elbows.
#
#       Plot intensity -- test4
#
Trial 1 : 2;
Trial 13: load file   'test3.cof';
Trial 14: intensity   (0,4) .. (4,0)  to 'test4.plot';
Trial 0 :
```

---

**Example 4 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                              5/10/87 -- KT

 Trial  1:  Print level=  2
Trial 13: load file   'test3.cof';
 Trial 13:  Fourier Coeffs loaded from "test3.cof "
Trial 14: intensity   (0,4) .. (4,0)  to 'test4.plot';
 Trial 14:  3-decimal plot line : (  0.    ,  4.000)..(  4.000,  0.   )
          50 points saved in "test4.plot"
Trial 0 :
 Program execution terminated.
 User Time (CPU) =    8.11667 sec,  System Time =   0.90000 sec.
```

**Example 5 : Input File**

```
# The next two examples again consist of two separate SPLAT input files.
# The first (Example 5) calculates the image for a checker-board pattern,
# and stores the calculated transmission cross-coefficients in the file
# named "test5.tcc".
# The second (Example 6 on the following page) reads the coefficients back,
# and calculates the intensity profiles for a different pattern, this time
# an isolated square.
# To make things a little more interesting, a slight amount of coma is
# introduced.
#
# 1.50 um. mask with 1.50 um. squares. -- test5
#
stmt  1: print  = 3
stmt  2: lambda = 0.5 um
stmt  3: na     = 0.5
stmt  4: defoc  = 0.0 um
stmt  5: s      = 0.7
stmt  6: mask   = 1.5 x 1.5 @0
stmt  7: cutout = (0.00,0.00) 0.75 x 0.75 @1
stmt  7: cutout = (0.75,0.75) 0.75 x 0.75 @1
stmt 20: xpos = 1   ypos = 0
stmt 22: coma = 0.1
stmt 10: mode = 1   save in 'test5.tcc'
stmt 14: cutline from (-0.75,0.4) to (0.75,0.4)
&        with 100 points in 'test5.plot'
end    0
```

---

**Example 5 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                        5/10/87 -- KT

 Trial  1:  Print level=  3
 Trial  2:  Lambda      =  0.5000 microns
 Trial  3:  N.A.        =  0.5000
 Trial  4:  Defocus     =  0.      microns  [ 0.     Rayleigh Units ]
 Trial  5:  Sigma       =  0.7000
 Trial  6:  Field size =  1.5000 x  1.5000 @ 0.      scale =1.00
 Trial  7:  Cutout      =( 0.     , 0.     )x( 0.7500, 0.7500) @ 1.0000 < 0.    >
 Trial  7:  Cutout      =( 0.7500, 0.7500)x( 0.7500, 0.7500) @ 1.0000 < 0.    >
 Trial 20:  Relative x-coord (object) =  1.0000
 Trial 20:  Relative y-coord (object) =  0.
 Trial 22:  a031(coma) =  0.1000
 Trial 22:  b031(dist) =  0.
 Trial 22:  Maximum Coma OPD = 0.2000 Wavelengths
 Trial 10:  Calculate image Fourier Transform
 Trial 10:  Imaging System Parameters saved
 #      10:    Lambda     =  0.5000
 #      10:    N.A.       =  0.5000
 #      10:    Defocus    =  0.
 #      10:    Sigma      =  0.7000
```

```
Trial 10:  Mask Characteristics saved
#    10:    xlength       = 1.5000
#    10:    ylength       = 1.5000
#    10:    Transmittance = 0.
#    10:  Object Coordinates : ( 1.0000  0.    )
Trial 10:    1 Primary Lens Aberrations saved
#    10:    Com  = 0.1000    Dis  = 0.
#    10:  Symmetry :  T( f1, g1, f2, g2) =
#    10:       conjg[T(-f1,-g1,-f2,-g2)]
#    10:             T( f1,-g1, f2,-g2)
#    10:       conjg[T(-f1, g1,-f2, g2)]
#    10:       conjg[T( f2, g2, f1, g1)]
#    10:   4905 / 14641 TCCs saved in "test5.tcc "
#    10:  TCC computation time  =  97.16666 sec.
#    10:  Imaged with  5 by  5 harmonics
#    10:  TCC calls:     662  zeros:    1259
Trial 14:  3-decimal plot line : ( -0.750,  0.400)..( 0.750,  0.400)
        100 points saved in "test5.plot"
Program execution terminated.
User Time (CPU) =  108.95000 sec,  System Time =  24.75000 sec.
```

**Example 6 : Input File**

```
# This example uses previously computed TCCs to recalculate the image
# intensity profile for a different mask pattern. Note that the image
# parameters (lambda, na, sigma) should not be changed. Also, the mask size
# must remain constant. If any of these parameters are changed, the
# results obtained with STATEMENT 19 will not be valid.
#
# The new pattern is an isolated 0.5 lambda/NA square. -- test6
#
stmt  1: print  = 3
stmt  6: mask   = 1.5 x 1.5 @0
stmt  7: cutout = (0.00,0.00) 0.25 x 0.25 @1
stmt 19: read from 'test5.tcc'
stmt 14: cutline from (-0.75,0.0) to (0.75,0.0)
&        with 100 points in 'test6.plot'
end   0
```

---

**Example 6 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                         5/10/87 -- KT

Trial  1:  Print level=  3
Trial  6:  Field size =  1.5000 x  1.5000 @ 0.     scale =1.00
Trial  7:  Cutout     =( 0.    , 0.    )x( 0.2500, 0.2500) @ 1.0000 < 0.    >
Trial 19:  Imaging System Parameters Loaded
#     19:    Lambda    =  0.5000
#     19:    N.A.      =  0.5000
#     19:    Defocus   =  0.
#     19:    Sigma     =  0.7000
Trial 19:  Mask Characteristics Loaded
#     19:    xlength       =  1.5000
#     19:    ylength       =  1.5000
#     19:    Transmittance =  0.
#     19:  Object Coordinates : (  1.0000  0.    )
Trial 19:    1 Primary Lens Aberrations Loaded
#     19:    Com  =  0.1000    Dis  =  0.
Trial 19:   4905 TCCs read in from "test5.tcc ".
#     19:  Imaged with  5 by  5 harmonics
Trial 19:  Calculate image Fourier Transform
Trial 14:  3-decimal plot line : ( -0.750,  0.   )..( 0.750,  0.   )
           100 points saved in "test6.plot"
Program execution terminated.
User Time (CPU) =   14.33333 sec,  System Time =   1.46667 sec.
```

**Example 7 : Input File**

```
# And finally, a demostration of Statements 9 and 11.
# Statement 9 shows what the mask looks like after a discrete Fourier
# transform, followed by an inverse Fourier transform, has been
# performed on the mask.
#
# Triangular pattern -- test7
#
stmt 1: print = 3
stmt 2: lambda = 0.5 um
stmt 3: na     = 0.5
stmt 4: defoc  = 0.0 um
stmt 5: s      = 0.3
stmt 6: mask   = 1.5 x 1.5 @0
stmt 8: cutout = (1.00,1.00) -1.00 x -1.00 @1
stmt 9: 20 20 mode = 1  box : (-1.5,-1.5) 3 x 3;
stmt 10:
stmt 11: -1.5 -1.5 3.0 3.0 'test7.ctr'
stmt 14: cutline from (-1.5,-1.5) to (1.5,1.5)
&      with 100 points in 'test7.plot'
end   0
```

---

**Example 7 : Line Printer Output**

```
 2-D optical imaging with aberrations-- V2.0
                                 5/10/87 -- KT

Trial  1:  Print level=  3
Trial  2:  Lambda      =  0.5000 microns
Trial  3:  N.A.        =  0.5000
Trial  4:  Defocus     =  0.      microns   [ 0.      Rayleigh Units ]
Trial  5:  Sigma       =  0.3000
Trial  6:  Field size  =  1.5000 x  1.5000 @ 0.      scale =1.00
#      8:  Add triangle (  1.000,  1.000)x( -1.000, -1.000)  @ 1.000
Trial  8:  Triangle    = (  0.    ,  1.000) (  1.000,  1.000) (  1.000,  0.    )
Trial  9:  Transmittance  : 20x20 pts  ( -1.500, -1.500)x(  1.500,  1.500)
          Contour map of the mask transmittance
           Left bottom corner :  ( -1.500 -1.500)
           Right top corner   :  (  1.500  1.500)
           x-interval         :  0.158 microns
           y-interval         :  0.158 microns
           0000000000000000000
           0000000000000000000
           0000000000000000000
           0000      00      0000
           000 11111   11111 000
           000 1111 00 1111 000
           000 111 0000 111 000
           000 11 000000 11 000
           000 1 00000000 1 000
           0000 0000000000 0000
           0000 0000000000 0000
```

```
          000 1 00000000 1 000
          000 11 000000 11 000
          000 111 0000 111 000
          000 1111 00 1111 000
          000 11111  11111 000
          0000      00     0000
          00000000000000000000
          00000000000000000000
          00000000000000000000
Trial 10: Calculate image Fourier Transform
#     10: Symmetry :  T( f1, g1, f2, g2) =
#     10:             T(-f1,-g1,-f2,-g2)
#     10:             T( f1,-g1, f2,-g2)
#     10:             T(-f1, g1,-f2, g2)
#     10:             T( f2, g2, f1, g1)
#     10: TCC computation time  =   1.93333 sec.
#     10: Imaged with  3 by  3 harmonics
#     10: TCC calls:     263  zeros:      74
Trial 11: Image intensity contour data stored in "test7.ctr "
Trial 14: 3-decimal plot line : ( -1.500, -1.500)..( 1.500,  1.500)
         100 points saved in "test7.plot"
Program execution terminated.
User Time (CPU) =  247.81667 sec,  System Time =  13.56667 sec.
```

□

## APPENDIX I : PLOTTING CONTOUR PLOTS

The easiest way of analyzing SPLAT output is undoubtedly by looking at contour plots, and by noting the differences between the plots when certain parameters are changed. Of course, the program can be set up to spew out reams upon reams of data, but no self-respecting graduate student cum programmer would willingly spend hours analyzing endless data points. Therefore, a good plotting package is absolutely necessary here. Unfortunately, there exist very few good 3-D plotting packages, and to compound the problem, each package comes with a significant amount of documentation. Therefore, a relatively primitive contour plotting program has been written, intended for use with HP239X and HP2647/8 terminals. The output of this program can also be piped to an Apple Laserwriter for nice-looking plots. This program is included free, for the first time (isn't that a bargain!) as part of the SPLAT imaging package.

The plot program consists of the following modules :

contour.f - main program, searches out location of contours
axes.f - subroutine to draw the axes
plot.f - plot out each contour

Once compiled, the program will accept an intensity file such as that produced by STATEMENT 11 (SPLAT). This intensity file consists of 2500 intensity points in a 50 by 50 rectangular grid, and the file *has* to be named "image.ctr". The program will read in the 2500 data points, and search by vertical and horizontal scan, with linear interpolation, for the location of points which fall on a 0.1*N contour, where N is an integer ranging from 1 to approximately 12 (the maximum N depends on the intensity maxima). For each contour, a line is drawn connecting all the points found. The order of connection is determined by nearest neighbor : a search is made of all the existing points for the point closest to an initial point - once that point is found, a line is drawn connecting the two points, the first point is deleted from the storage array, a search is made for the point closest to the second point, and so on.

This program, named "contour", can be run as follows (% denotes the UNIX prompt) :

% contour

The program will then read in the 2500 intensity points, and will next prompt the user for the following options :

[1] Enter Contour Step Size [0.05 - 0.20]:

The program assumes that the intensity values are normalized to a clear-field intensity of 1.0, and can be asked to produce contours at different step sizes. For example, a step size of 0.1 will produce contours at 0.1, 0.2, 0.3, and so on.

[2] Enter scale factor [0.1 - 1.0]:

The contour plot can be scaled down linearly using this option.

[3] Do you want a HP2648 contour map? [y,n]

This option is used on a HP2648 or equivalent (e.g. HP2397) terminal.

[4] Do you want a POSTSCRIPT contour map? [y,n]

This option, if answered affirmatively, will produce a contour plot in POSTSCRIPT format. The program will produce a file named "image.ps", which can be sent to an Apple Laserwriter using the UNIX command "lpr -P[printer] image.ps".

[5] Do you want labels on odd contours? [y,n]

This is used only for the POSTSCRIPT contour plot; it will be ignored in HP2648 mode. A negative answer will result in only every other contour line being labeled. For example, if the contour step size is 0.1, then only the 0.2, 0.4, 0.6, ..., contour lines will be labeled.

There are, of course, discretization problems, as well as point connection problems, but on the whole, this program does do a reasonable job of producing a contour plot. The output looks nice, and has good resolution, but is not quite up to the standards of the professional plot packages.

**APPENDIX I.1 : CONTOUR PLOTS WITH DISSPLA**

The DISSPLA (Integrated Software Systems Corporation) graphics package can be used to produce very good quality contour plots. However, as most users will most likely not have access to DISSPLA, the module containing the DISSPLA subroutine is not included as part of the normal SPLAT package. If DISSPLA is available, though, this is what has to be done.

(a) Replace the SPLAT module "tr15.f" with the module "tr15ds.f" included with the source code.

(b) Somehow define the DISSPLA environment so that DISSPLA variables are included during compilation. This is similar to including external functions, and varys from site to site.

(c) Compile the program. The modules to be included are :
amask.f atrian.f bdat.f crosgd.f cross.f crossc.f files.unix.f funcc.f main.f opl.f spec.f specsym.f trial1.f trial2.f trial3.f tr15ds.f trial4.f

The DISSPLA subroutine included here has been used successfully on an IBM 3090 mainframe. However, because the use of DISSPLA is dependent upon site, help with installing this subroutine is best sought from your local DISSPLA expert.

**APPENDIX II : A FEW GOOD ABERRATIONS...**

At present, this program can handle any combination of the five primary lens aberrations, which are as follows :

1. Spherical Aberration

   This aberration is normally present on all uncorrected lenses, and is due mainly to the failure of the paraxial approximation in the ideal lens law. Rays that pass though the outer zones of a lens are deflected more than those that pass through the inner zones, and as a result, these rays do not pass through a common focus. Spherical aberration is the only aberration that is independent of the object position relative to the lens axis : all points on the image plane are affected similarly.

2. Coma

   Coma comes about from the unequal bending of parallel rays from an off-axis object. Rays parallel to the lens axis will come to a common focal point, but if these rays are shifted slightly (approx. 5 degrees) off axis, they will not focus, primarily because each ray "sees" a different amount of glass. In effect, coma is caused by unequal magnification in different zones of the lens, due to failure of the paraxial lens law.

3. Astigmatism

   Off-axis objects cause different focus points for different ray planes. The object sees a thicker lens width along the sagittal plane than the tangential plane, which results in different focal points for each of these planes. All spherical lenses have astigmatism for off-axis objects.

4. Curvature

   Strictly speaking, curvature is not an aberration at all, but comes about because flat images are desired instead of the natural spherical images. Curvature is similar to defocus.

5. Distortion

   Distortion only occurs when the other lens aberrations are present, and when the aperture is placed some distance away from the lens. This aberration is due mainly to different magnification at the outer zones of the lens.

These five primary aberrations, a.k.a. Seidel Aberrations, a.k.a. 3rd (lowest) order ray aberrations, can be expressed as functions of the object location (relative to the lens axis) and the image location or spatial frequency (and harmonics). For a more detailed treatment, please refer to the paper "Identifying and Monitoring Effects of Lens Aberrations in Projection Printing", *SPIE Vol 772 : Optical Microlithography VI*, Santa Clara, CA, March 1987, by Toh and Neureuther.

□

**APPENDIX III : HELP?**

This program is still decently young, and the aberrations capability is hardly a year old. Therefore, it should not surprise the user too much if this program crashes while simulating a complex pattern. This has been noticed in certain cases for images with defocus for large masks, and I am still trying to find the bug...

However, on the positive side, I have been running numerous test cases during the past few months, and my crash ratio is approximately 0.1%. If a program glitch does occur, feel free to call me up and ask for help. Debugging and mask specification help is available from

Kenny K.H. Toh,
550-C10 Cory Hall,
Dept of Electrical Engineering,
Cory Hall,
Univ of California at Berkeley,
Berkeley, CA 94720.
(415)-642-8897

□