

Copyright © 1986, 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 OPTICAL IMAGING
FOR PHOTOLITHOGRAPHY SIMULATION**

by

Philip D. Flanner III

Memorandum No. UCB/ERL M86/57

3 July 1986

TWO-DIMENSIONAL OPTICAL IMAGING
FOR PHOTOLITHOGRAPHY SIMULATION

by

Philip D. Flanner III

Memorandum No. UCB/ERL M86/57

3 July 1986

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TWO-DIMENSIONAL OPTICAL IMAGING
FOR PHOTOLITHOGRAPHY SIMULATION

by

Philip D. Flanner III

Memorandum No. UCB/ERL M86/57

3 July 1986

ELECTRONICS RESEARCH LABORATORY
College of Engineering
University of California, Berkeley
94720

Table of Contents

1. Introduction	1
2. Calculating images of two-dimensional features	1
2.1. Introduction	1
2.2. The Transmission Cross-Coefficient Formulation	2
2.2.1. The Transmission Function K	2
2.2.2. The Mutual Intensity Function J_0^-	4
2.3. The Program	4
3. Applications	7
3.1. Introduction	7
3.2. Square vs. Circular contact holes	8
3.3. Elbows	9
3.4. Defects	10
4. Conclusion	10

Appendix I — Figures

Appendix II — Twod user manual

Two-dimensional optical imaging for Photolithography Simulation

Philip D. Flanner III

1. Introduction

As optical lithography processing is pushed into the sub-micron region, the ability to construct a mask that produces the required image in photoresist becomes a limiting factor. As dimensions approach or go below the classical resolution limit, the interference of diffracted light between adjacent features is a factor that must be taken into account when designing layout rules. Computer simulation can be a valuable tool in the study of the imaging of small features and their proximity effects.

The SAMPLE process simulation program can simulate the aerial image and processing of one-dimensional features, basically straight parallel lines. However, most cases of feature proximity, such as contacts near contacts or lines, lines near pads, defects, and lines that bend are inherently two-dimensional and require a two-dimensional process simulation. The first step in such a process simulation program has been completed. This general two-dimensional aerial image program is the subject of this report.

2. Calculating images of two-dimensional features

2.1. Introduction

In order to be a useful tool, the simulation program should use models that accurately represent actual processes. Figure 1 shows the model system used. It consists of an incoherent quasimonochromatic light source, a light condensing system, a user definable object (the mask), and the imaging system. The program calculates the aerial image at the surface of the photoresist — the next step would be to simulate the resist exposure and development.

The transmission cross-coefficient method of Hopkins(1) is used to calculate the aerial image. In this section I will discuss the equations, terms, assumptions and actual implementation of this method in the program.

2.2. The Transmission Cross-Coefficient Formulation

A complete derivation of the transmission cross-coefficient method for partially coherent imaging is given in Born and Wolf(2). It is shown that the fourier transform of the image intensity is given by

$$I(f, g) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(f' + f, g' + g; f', g') F(f' + f, g' + g) F^*(f', g') df' dg' \quad (1)$$

where $F(f, g)$ is the fourier transform of the mask transmission function, which would usually be 1 in transparent areas and 0 in opaque areas of the mask. $T(f', g'; f'', g'')$ is the transmission cross coefficient, which contains all the information on the imaging system and illumination. It is derived to be

$$T(f', g'; f'', g'') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J_0^-(f, g) K(f + f', g + g') K^*(f + f'', g + g'') df dg \quad (2)$$

where $K(f, g)$ is the frequency response function of the imaging system and $J_0^-(f, g)$ is the fourier transform of the mutual intensity of the light at the object. These two functions are discussed below.

2.2.1. The Transmission Function K

The coherent transmission function $K(x_o, y_o; x_i, y_i)$ of the imaging system is defined to be the complex amplitude, per unit area of the object plane, at the point (x_i, y_i) in the Gaussian image plane due to a disturbance of unit amplitude and zero phase at the object point (x_o, y_o) . If we assume that the object is small in relation to the distance between the object and image planes, the transmission function to a good approximation is a function only of the displacement of the Gaussian image point; i.e.

$$K(x_o, y_o; x_i, y_i) = K(x_i - x_o, y_i - y_o) \quad (3)$$

Under these conditions it can be shown that the frequency response function is equal to the pupil function $G(\xi, \eta)$ at the point $\xi = aM\lambda/NAf$, $\eta = aM\lambda/NAg$ of the Gaussian reference sphere :

$$K(f, g) = G(aM\frac{\lambda}{NA}f, aM\frac{\lambda}{NA}g) \quad (4)$$

where a is the radius of the aperture (exit pupil), M the magnification of the imaging system, λ the wavelength of the light and NA the numerical aperture of the imaging system. The amplitude of G is a measure of the non-uniformity of the image forming wave (assumed uniform here). The phase of G is the aberration function Φ of the system. Since no light comes from outside the exit pupil, G is zero outside a circle of radius a , i.e.

$$G(\xi, \eta) = 0 \quad \xi^2 + \eta^2 > a^2 \quad (5a)$$

which corresponds to

$$K(f, g) = 0 \quad f^2 + g^2 > \frac{1}{M^2} \left(\frac{NA}{\lambda} \right)^2 \quad (5b)$$

Thus the frequency response function is non-zero only in a small circular area, so that the integrals in eqn (2) only formally extend to $\pm\infty$.

The simulation program assumes no aberrations in the imaging optics. However, the program can calculate the image at a plane other than the plane of best focus. According to the displacement theorem, this focus error can be represented as an aberration term in the imaging system aberration function :

$$\Phi_{\Delta z}(\xi, \eta) = \frac{1}{2} \Delta z \left(\frac{NA}{M} \right)^2 \frac{\xi^2 + \eta^2}{a^2} \quad (6)$$

Here Δz is the distance away from the plane of focus; positive Δz is above the plane.

Thus the frequency response function used in eqn. (2) is

$$K(f, g) = \begin{cases} \exp \left[\frac{2\pi i}{\lambda} \frac{1}{2} \Delta z \lambda^2 (f^2 + g^2) \right] & f^2 + g^2 < \left(\frac{NA}{\lambda M} \right)^2 \\ 0 & f^2 + g^2 \geq \left(\frac{NA}{\lambda M} \right)^2 \end{cases} \quad (7)$$

2.2.2. The Mutual Intensity Function J_0^-

The mutual intensity function $J(P_1, P_2)$ is defined to be the value of the cross-correlation function $\Gamma_{12}(\tau)$ of the complex disturbance (e.g. electric field) of the light emerging from points P_1 and P_2 at $\tau = 0$:

$$J(P_1, P_2) = \langle U_1(t) U_2^*(t) \rangle \quad (8)$$

In eqn. (2), J_0^- is the fourier transform of the mutual intensity function of the light from the condenser just before passing through the mask (i.e. just above the mask in Figure 1). If the size of the condenser entrance pupil and the area of the mask to be imaged are small compared to the distance from the source to the mask, then the mutual intensity function is proportional to the fourier transform of the intensity function of the source. Assuming a circular aperture and uniform intensity (=1), the fourier transform of the mutual intensity is constant inside a circle corresponding to the condenser aperture and given by

$$J_0^-(f, g) = \begin{cases} \frac{\lambda^2}{\pi s^2 NA^2} & f^2 + g^2 < \frac{s^2 NA^2}{\lambda^2} \\ 0 & f^2 + g^2 > \frac{s^2 NA^2}{\lambda^2} \end{cases} \quad (9)$$

The partial coherence factor s is the ratio of the numerical aperture of the condenser to the numerical aperture of the imaging system; thus sNA = the numerical aperture of the condenser.

2.3. The Program

In order to simplify the evaluation of the integrals in eqn. (1) I assume the mask transmittance function $F(x, y)$ is periodic with periods $2L_x$ and $2L_y$. It can be written as a Fourier series :

$$F(x, y) = \sum_m \sum_n c(m, n) \exp(imk_x x) \exp(ink_y y) \quad (10)$$

where $k_x = \frac{2\pi}{2L_x}$ and $k_y = \frac{2\pi}{2L_y}$. The Fourier transform of $F(x, y)$ is given by

$$F(f, g) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_m \sum_n c(m, n) \exp(imk_x x) \exp(ink_y y) \exp(2\pi i x f) \exp(2\pi i y g) dx dy \quad (11a)$$

$$= \sum_m \sum_n c(m, n) \delta(f, \frac{-mk_x}{2\pi}) \delta(g, \frac{-nk_y}{2\pi}) \quad (11b)$$

where δ is the Dirac delta function. Replacing $F(f, g)$ and $F^*(f, g)$ in eqn. (1) with the expression above gives

$$I(f, g) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(f' + f, g' + g; f', g') \sum_m \sum_n c(m, n) \delta(f' + f, \frac{-mk_x}{2\pi}) \delta(g' + g, \frac{-nk_y}{2\pi}) \times \quad (12)$$

$$\sum_p \sum_q c^*(p, q) \delta(f', \frac{-pk_x}{2\pi}) \delta(g', \frac{-qk_y}{2\pi}) df' dg'$$

The delta functions eliminate the integrals leaving

$$I(f, g) = \sum_r \sum_s I_{rs} \delta(f, \frac{-rk_x}{2\pi}) \delta(g, \frac{-sk_y}{2\pi}) \quad (13a)$$

where

$$I_{rs} = \sum_p \sum_q c^*(p, q) T(\frac{-pk_x}{2\pi} + f, \frac{-qk_y}{2\pi} + g, \frac{-pk_x}{2\pi}, \frac{-qk_y}{2\pi}) c(p + r, q + s) \quad (13b)$$

The intensity distribution at the image is then given by the inverse Fourier transform of $I(f, g)$ which is

$$I(x, y) = \sum_m \sum_n d(m, n) \exp(imk_x x) \exp(ink_y y) \quad (14a)$$

where

$$d(m, n) = \sum_p \sum_q c^*(p, q) T(\frac{-(p+m)k_x}{2\pi}, \frac{-(q+n)k_y}{2\pi}, \frac{-pk_x}{2\pi}, \frac{-qk_y}{2\pi}) c(p + m, q + n) \quad (14b)$$

The program forces the mask pattern to be an even function by reflecting the user input pattern about the x and y axes. Figure 2 shows the periodic extension; the square in the center is the original pattern. The program also allows only real mask transmittances (e.g. no phase shifting). Under these conditions, eqn. (10) for the mask transmittance can

be simplified to

$$F(x, y) = \sum_m \sum_n F_{mn} \cos(mk_x x) \cos(nk_y y) \quad (15)$$

where the coefficients F_{mn} are related to the $c(m, n)$ of eqn. (10) by

$$c(m, n) = c(-m, n) = c(m, -n) = c(-m, -n) = \frac{F_{mn}}{4} \quad (m, n \neq 0) \quad (16)$$

In eqn. (15) the sums are over non-negative m and n only.

The transmission cross coefficient $T(\dots)$ in eqn. (14b) can be evaluated using eqn. (2). I use dimensionless coordinates normalized by λ/NA and assume a magnification of 1:1. This simplifies equations (7) and (9):

$$T(f', g'; f'', g'') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J_0^-(f, g) K(f + f', g + g') K^*(f + f'', g + g'') df dg \quad (2)$$

$$K(f, g) = \begin{cases} \exp\left[\frac{2\pi i}{\lambda} \frac{1}{2} \Delta z NA^2 (f^2 + g^2)\right] & f^2 + g^2 < 1 \\ 0 & f^2 + g^2 \geq 1 \end{cases} \quad (7')$$

$$J_0^-(f, g) = \begin{cases} \frac{1}{\pi s^2} & f^2 + g^2 < s^2 \\ 0 & f^2 + g^2 > s^2 \end{cases} \quad (9')$$

Thus the integrand in eqn. (2) is non-zero only in the intersection of the three circles defined by J_0^- , K and K^* . If there is no defocus, the integral is just the area of intersection and has an analytic solution; when there is defocus the integral has to be evaluated numerically but can be reduced to a one-dimensional integral as described by Subramanian(3). If aberrations other than defocus are allowed, then in general the aberration function Φ will not be radially symmetric and a two-dimensional numerical integration technique is needed. Since the integral must be zero if the circles do not intersect, there is an upper limit on the number of terms that need to be considered. In eqn. (14b), the transmission cross coefficient will be zero if

$$\frac{mk_x}{2\pi} > 1+s \quad (17a)$$

$$m > 2 \cdot (1+s) L_x \frac{NA}{\lambda} \quad (17b)$$

and similarly for n. These are the maximum values of the indices m (and n) needed for computation in equations (10) and (15).

3. Applications

3.1. Introduction

The imaging program has been used to study aerial images of two-dimensional features. I will briefly present some results for square vs. round contact holes, elbows, defects (specifically intrusion defect to a clear line), and a cross pattern that shows the constructive interference effect when using light with higher than normal ($s \leq 0.3$) coherence. More detailed discussion of these and other applications can be found in papers presented by our research group (4)(5).

One problem with this program is the amount of computer time that is required to compute the image of a large field. A field $7\lambda/NA$ by $7\lambda/NA$ may take an hour or more of CPU time on a VAX or a couple minutes on an IBM 3081 mainframe. The run time is proportional to the square of the area so larger images take much more time. However, this program is primarily useful in studying corners and very small features, so that field areas need not be large. Also, the periodic extension done by the program can be exploited by using patterns with one or two axes of symmetry, reducing the field area by a factor of 2 or 4. None of the figures in this report took more than 4 minutes on the IBM 3081, including plotting time.

3.2. Square vs. Circular contact holes

Small square contact holes in a mask do not always print as squares. Figures 3 and 5 show the aerial images of $1.25\mu\text{m.}$ by $1.25\mu\text{m.}$ square contact holes (with GCA parameters, i.e. $NA = 0.28$ and $\lambda = .436\mu\text{m.}$). The square shape is completely lost in the image, which is nearly (but not perfectly) circular. Figures 4 and 6 show what happens when the contact size is increased to $2.50\mu\text{m.}$ Now the aerial images are square with rounded corners, although the corner rounding is about half of each side.

Figures 3 & 5 and 4 & 6 were imaged with two different coherence factors $\sigma = 0.3$ and $\sigma = 0.7$. In the case of the smaller feature, the reduced coherence factor (and thus increased coherence of the light) causes the peak intensity to be much higher and the intensity slope and contrast of the contacts to be higher, which could lead to better control in contact printing. The larger features are very similar in both cases except at the center, where constructive interference has added four intensity peaks at the corners of the square. Since these peaks occur in a region where the light intensity is >1 anyway, they would most likely not show up in the resist profile. Thus it appears that decreasing σ could increase the control over size of printed contacts for small features, but has little advantage for large features.

Figures 7 through 10 are analogous to figures 3 through 6 except that they are images of circular contact holes of the same area (actually octagonal approximations to circles since my program can only handle rectangular and triangular shapes). Theoretical considerations indicate that more light appears in the central bright area in the image of a circular aperture than a square one. This could lead to a) higher contrast images and b) less interference between adjacent features so that circular contacts could be packed closer together than square contacts. Since small square contacts image as circles anyway, it would be more efficient to design around circles than to attempt to keep square contacts (assuming that circular contacts meet electrical/electronic specifications!)

Comparison of figures 3 and 7 (or 5 and 9) show that indeed more light does appear in the center region of the image of the circular contact. However, the effect is not great and probably not worth the extra effort of making circular contacts on the mask unless there is less interference between adjacent contacts. Figures 8 and 10 show that, as one would expect, large circular contacts still image as circles. Comparing with the large squares shows that image intensity slope is lower for the circles.

3.3. Elbows

Elbows (lines that go around corners) are common features on a mask that can not be studied with a one-dimensional simulator like SAMPLE. Figure 11 is a contour plot of the image of a light $1.25\mu\text{m}$. elbow nested between two dark $1.25\mu\text{m}$. elbows. The peak intensity at the center of the line is 120% far from the corner and 140% at the corner. There is a large dip in intensity near the corner. At $\sigma = 0.3$ this effect is large enough to possibly affect the width of the lines in the resist when developed. Increasing σ to 0.7 nearly eliminates the problem as can be seen in figure 12.

Figure 13 is a contour plot of the image of an isolated light $1.25\mu\text{m}$. elbow with $\sigma = 0.3$. The intensity peak and valleys at the corner are nearly identical to those in the corresponding elbow in figure 11, showing that this effect is an intrafeature and not inter-feature effect, i.e. it arises from interference of light from different parts of the same feature rather than from different features.

Although these figures show only the aerial image (because we don't have a three-dimensional resist exposure/develop simulator) they can give a good qualitative idea of what the resist profile will look like. Figure 14 shows an SEM of an actual resist profile and a simulation from the imaging program. Since I couldn't simulate resist development, I just took the aerial image and flipped it over. The simulation pictured was run with the parameters used in the printing of the three elbows at the right of the SEM, which have $0.5\mu\text{m}$. widths. Several features, such as the resist "bumps" at the corners and the raised

valleys between the resist lines are at least qualitatively predicted by the aerial image simulation.

3.4. Defects

The printability of mask defects and effect of defects on other features is of great concern in photolithography. The two-dimensional imaging program is very useful in evaluating these effects. Figures 15 through 20 show the effect of an intrusion defect which could be caused by a speck of dirt or a piece of stuck chrome on the image of a $1.25\mu\text{m}$. wide transparent line. Figure 15 shows the image of the line with no defect and figures 16 through 20 show the image as the $0.5\mu\text{m}$. wide defect intrudes successively deeper into the line. At the center of the defect the contour lines are pushed in about half the intrusion distance of the defect. Based on theoretical considerations and simulations such as this a semi-empirical formula giving a "rule-of-thumb" effect of a defect as a function of size and distance from a feature can be developed(5).

4. Conclusion

A computer program has been written that uses the transmission cross coefficient approach to find the aerial images of masks containing arbitrary patterns constructed of rectangular and triangular patches. The program has been used in some studies of images of two-dimensional features. In general, interfeature and intrafeature proximity effects on small ($< 0.8\lambda/NA$) features can be minimized by keeping the partial coherence factor σ greater than 0.5. At these values of σ the interactions between features are similar to those between adjacent lines and a separation greater than $0.6 \lambda/NA$ is recommended.

An exception is in the printing of contacts. Reducing σ to 0.3 will increase the peak intensity and intensity slope of contact images which allows better control of contact hole size.

Acknowledgement

This research was supported by the National Science Foundation under grant ECS-8420688 and industrial contributions under the California MICRO program.

References

- [1] H.H. Hopkins. "On the diffraction theory of optical images." *Proc. Royal Soc. Series A*, vol. 217 No. 1131, pp. 408-432, 1953.
- [2] M. Born & E. Wolf, Principles of Optics, Pergamon Press, Ch 9 and 10.
- [3] S. Subramanian, "Rapid Calculation of defocused partially coherent images." *Appl. Optics*, vol. 20 No 10, pp. 1854-1857, 1981
- [4] P. Flanner III, S. Subramanian, A.R. Neureuther, "Two-dimensional Optical Proximity Effects." *Proc. SPIE Optical Microlithography V*, 1986.
- [5] A.R. Neureuther, P. Flanner III, "Coherence of Defect Interactions with Features in Optical Imaging." *Proc. Electron,Ion,Photon Beams Symposium*, Boston, May 27-30 1986.

Appendix I — Figures

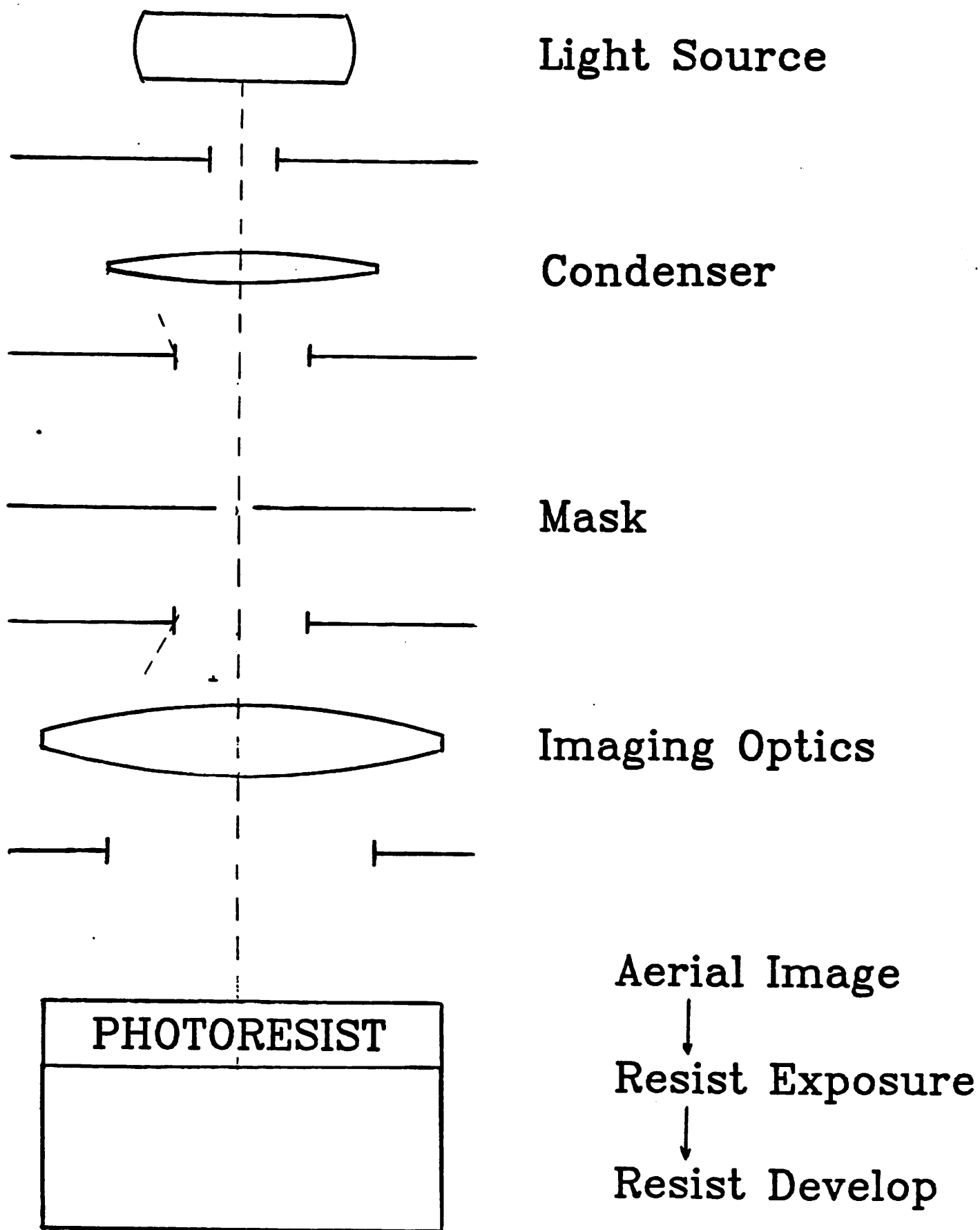
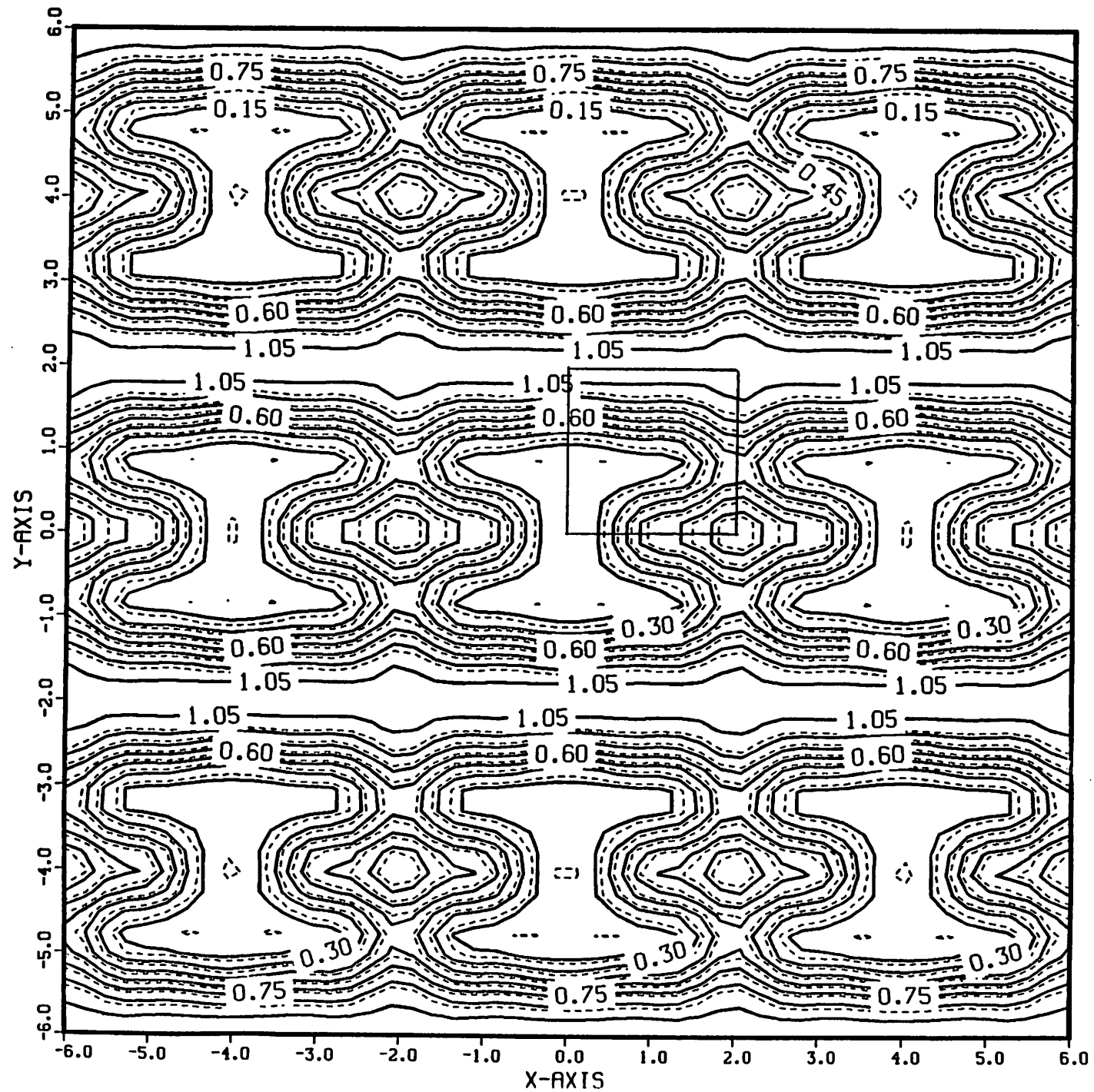


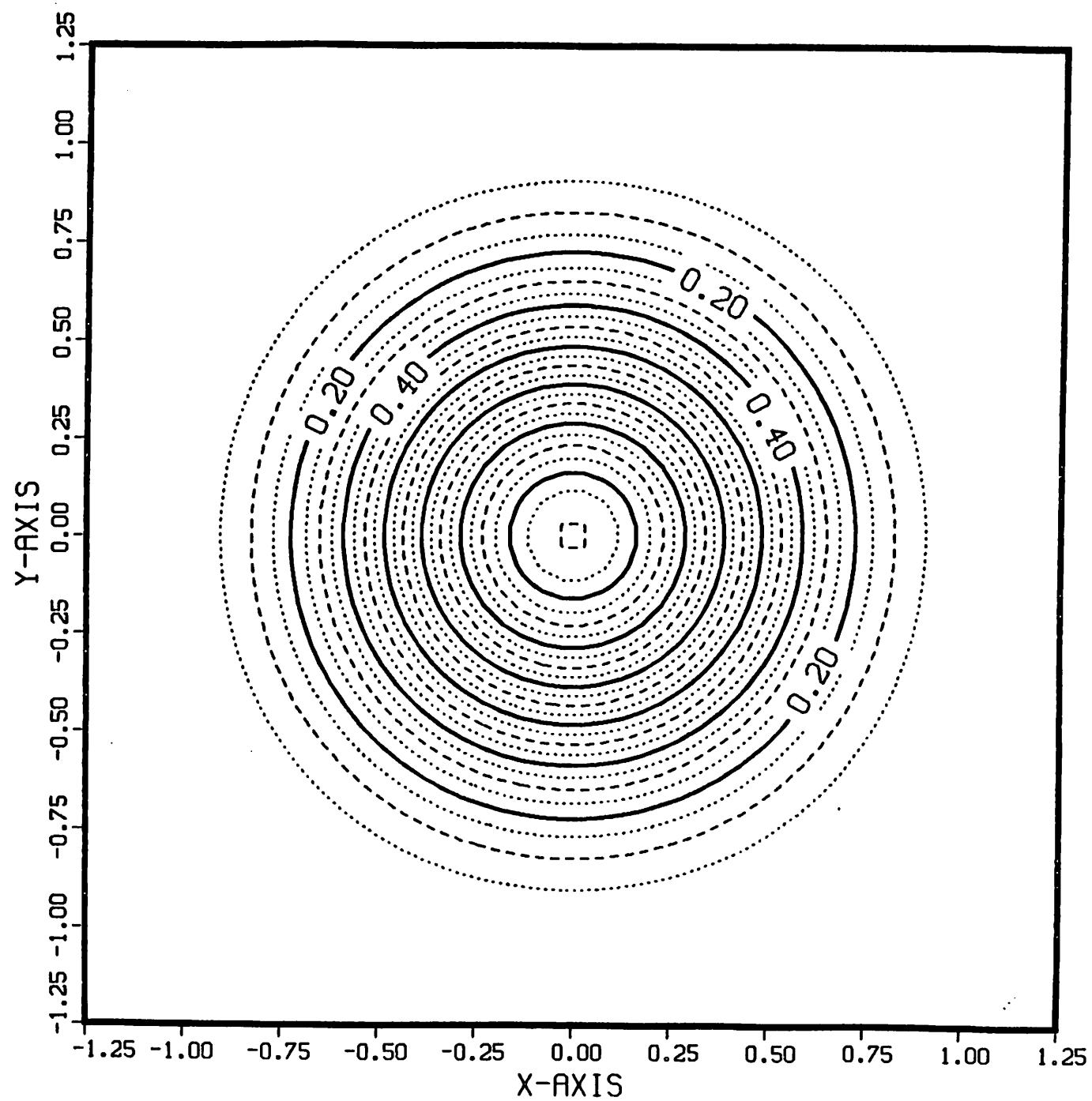
Figure 1

Figure 2



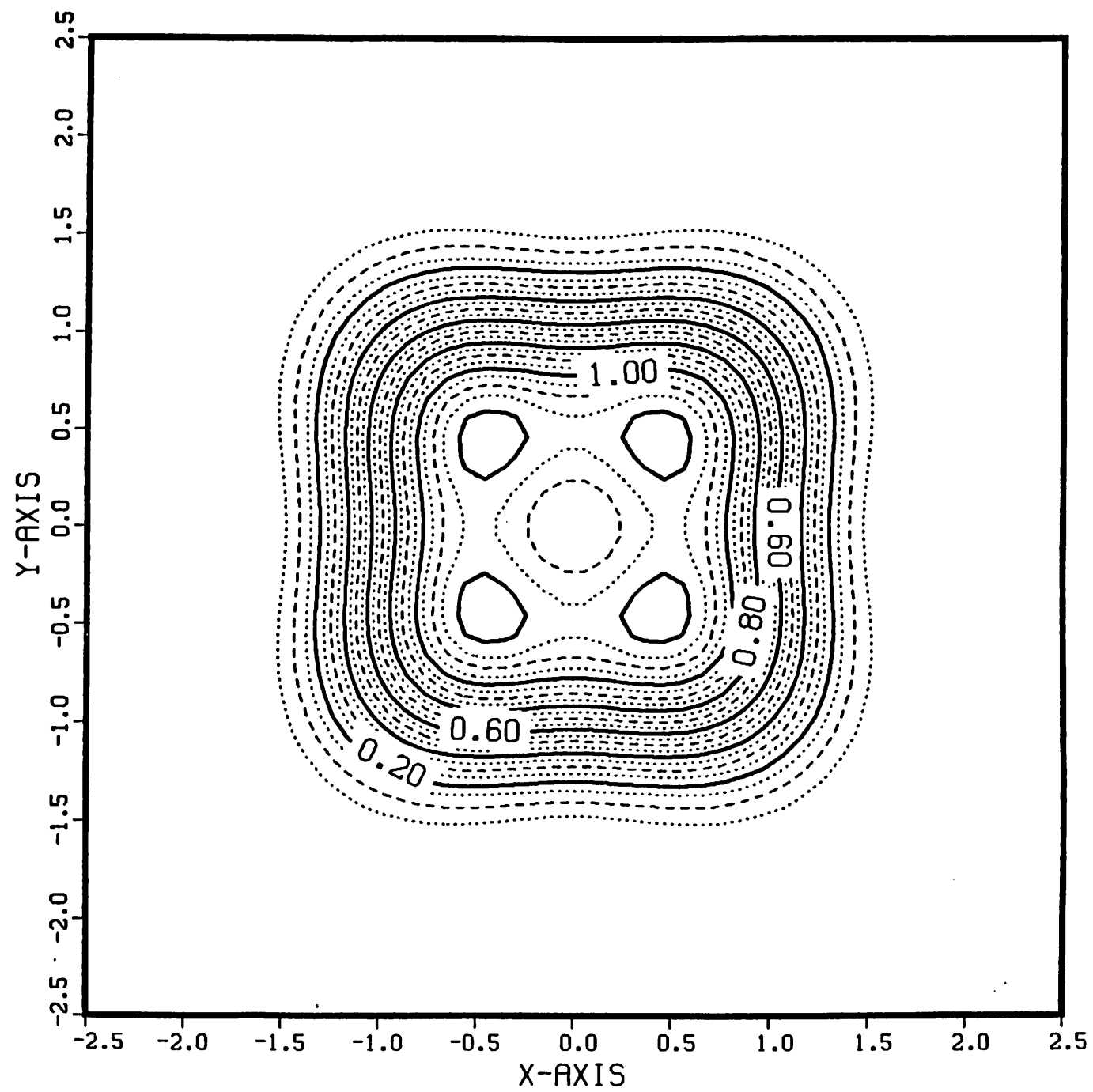
SQUARE CONTACT : SIGMA=0.3 SIZE=1.25 UM

Figure 3



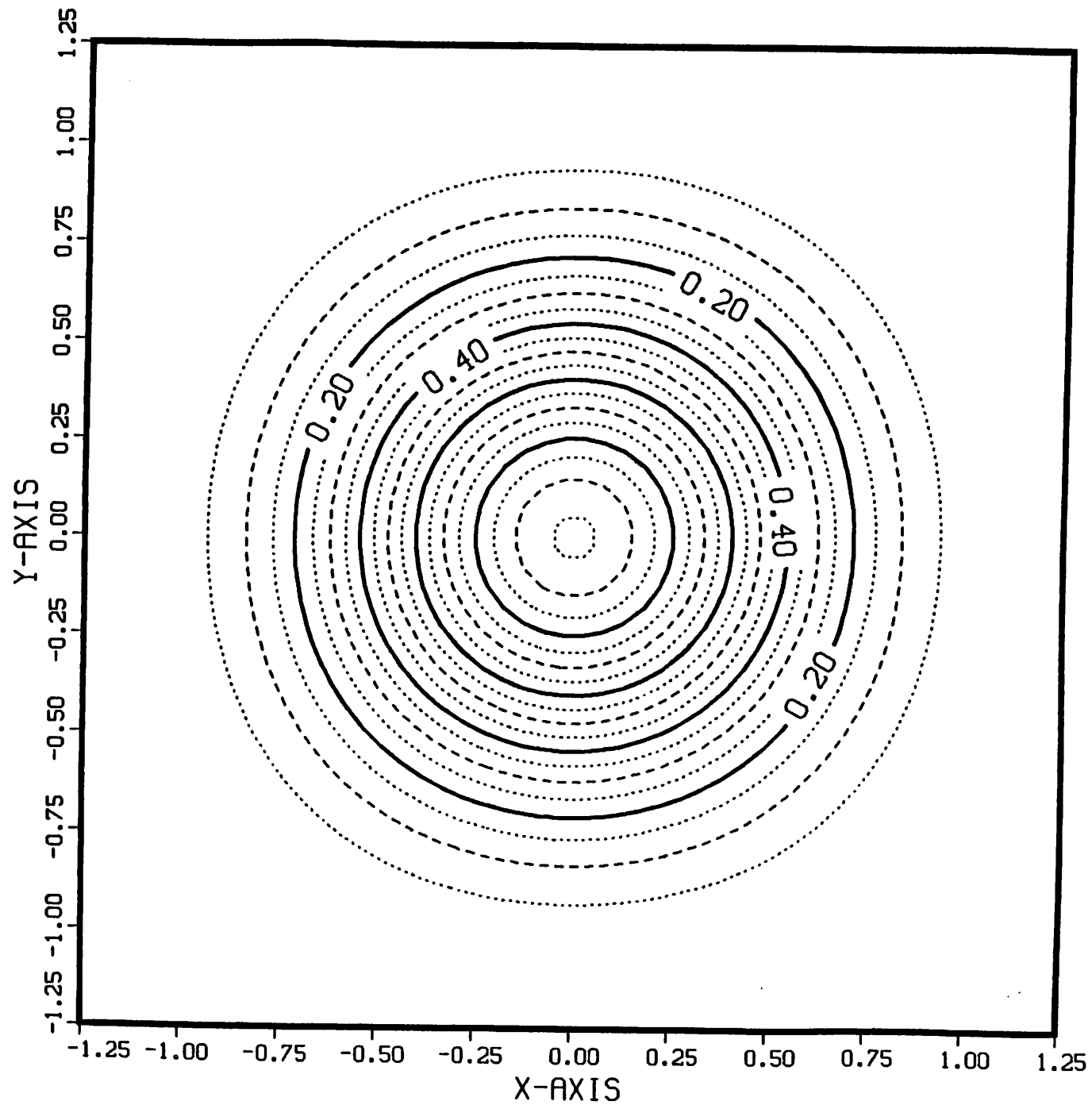
SQUARE CONTACT : SIGMA=0.3 SIZE=2.50 UM

Figure 4



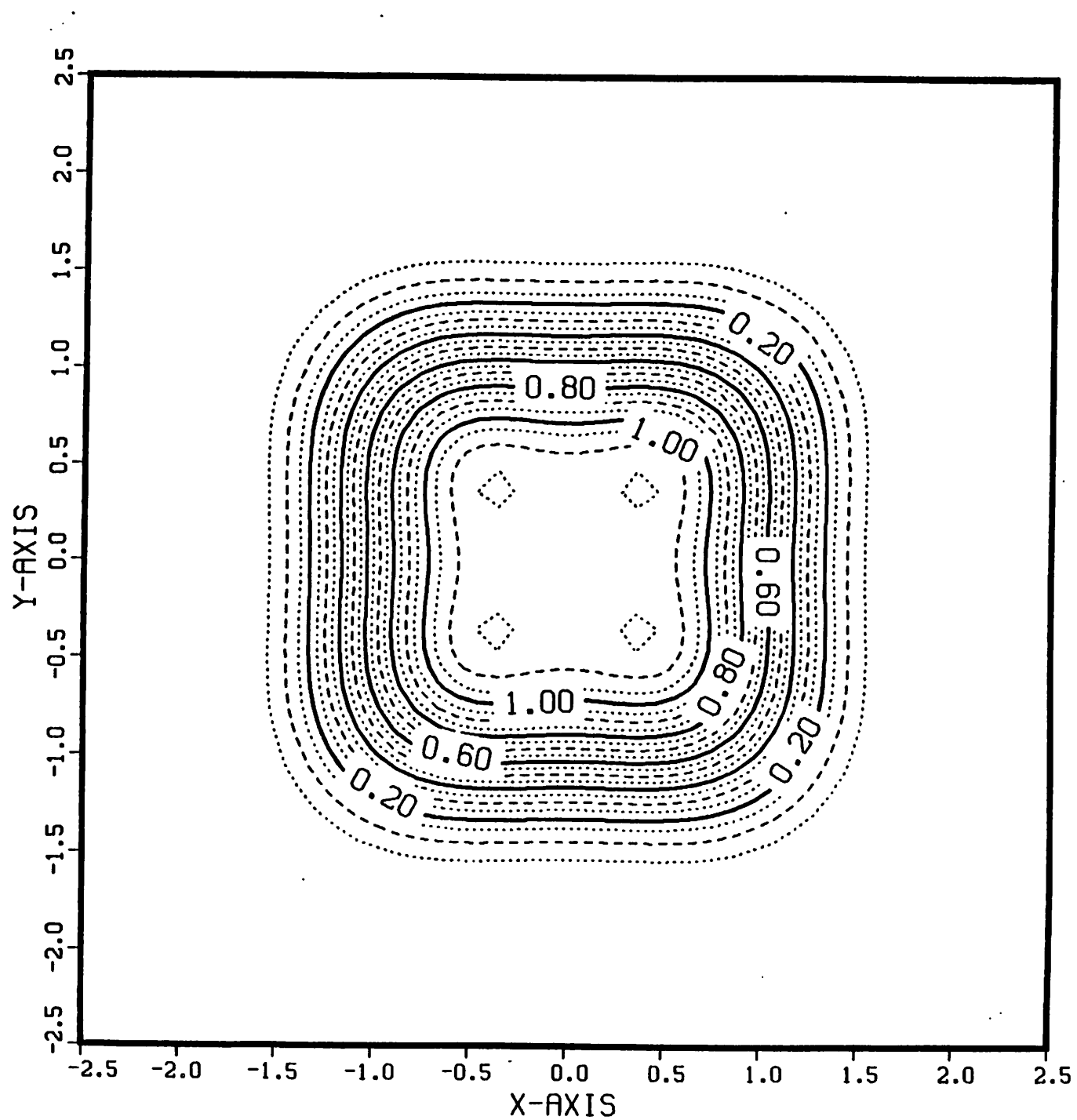
SQUARE CONTACT : SIGMA=0.7 SIZE=1.25 UM

Figure 5



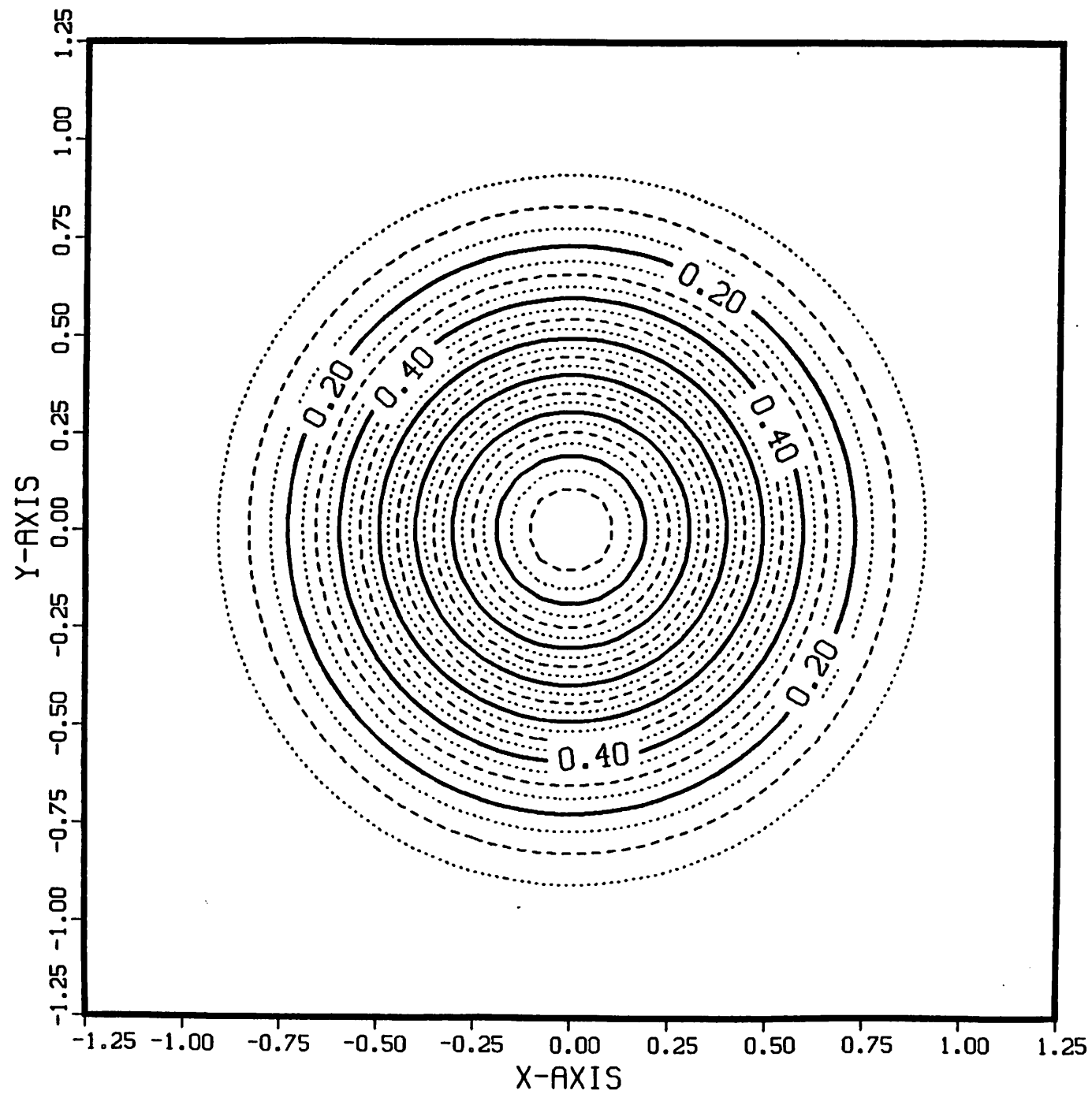
SQUARE CONTACT : SIGMA=0.7 SIZE=2.50 UM

Figure 6



CIRCLE CONTACT : SIGMA=0.3 SIZE=1.25 UM

Figure 7



CIRCLE CONTACT : SIGMA=0.3 SIZE=2.50 UM

Figure 8

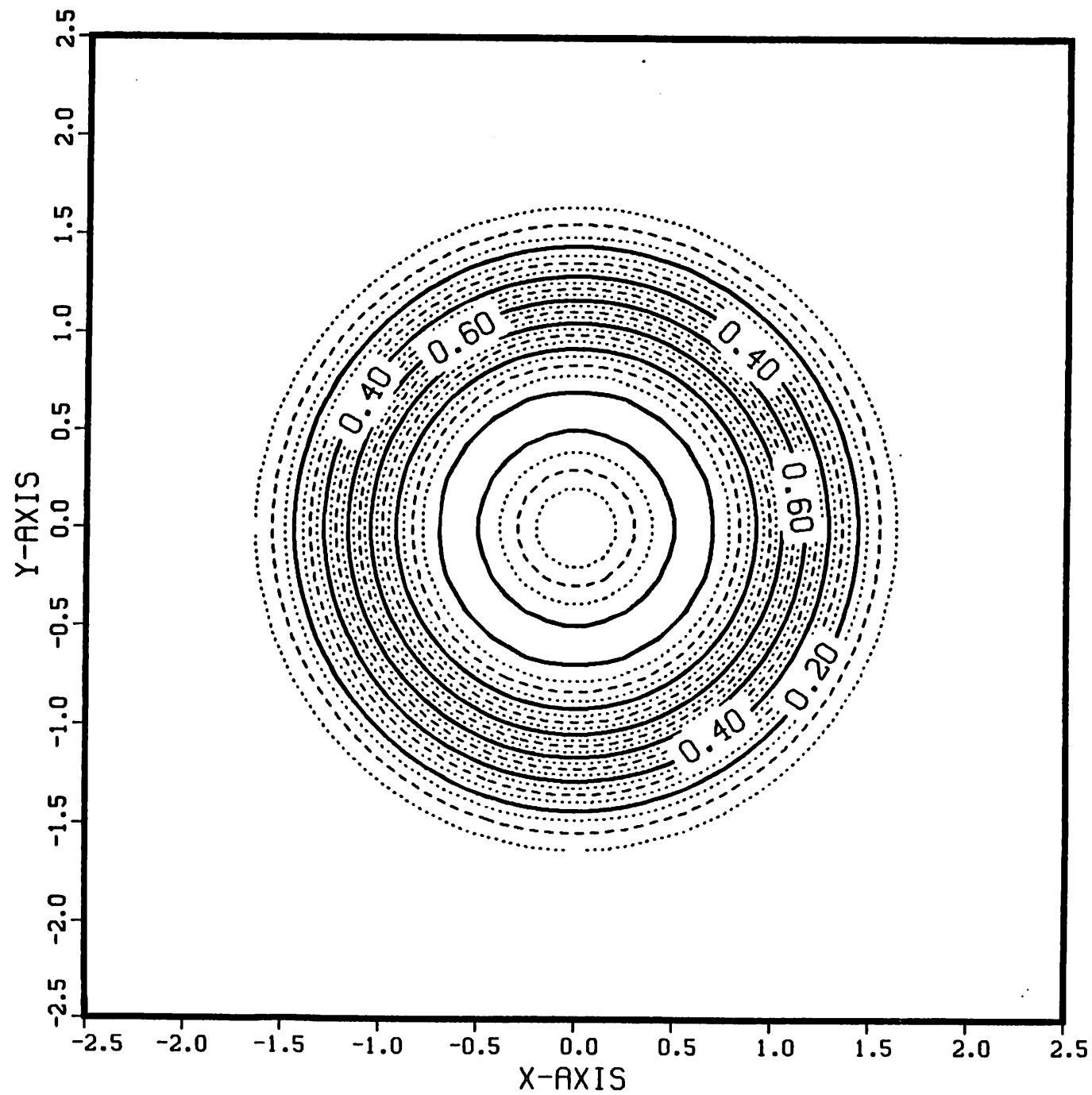
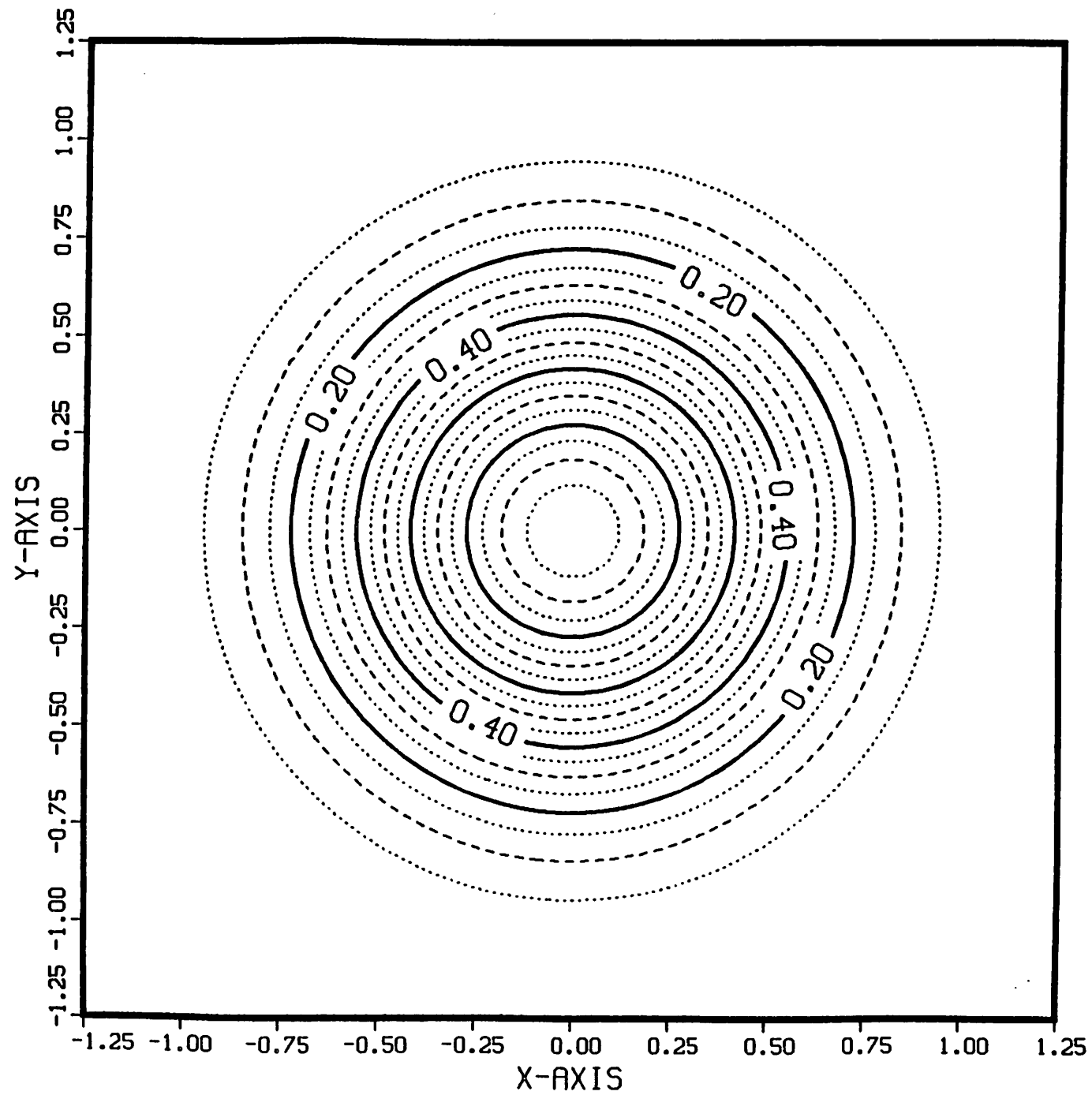


Figure 9



CIRCLE CONTACT : SIGMA=0.7 SIZE=2.50 UM

Figure 10

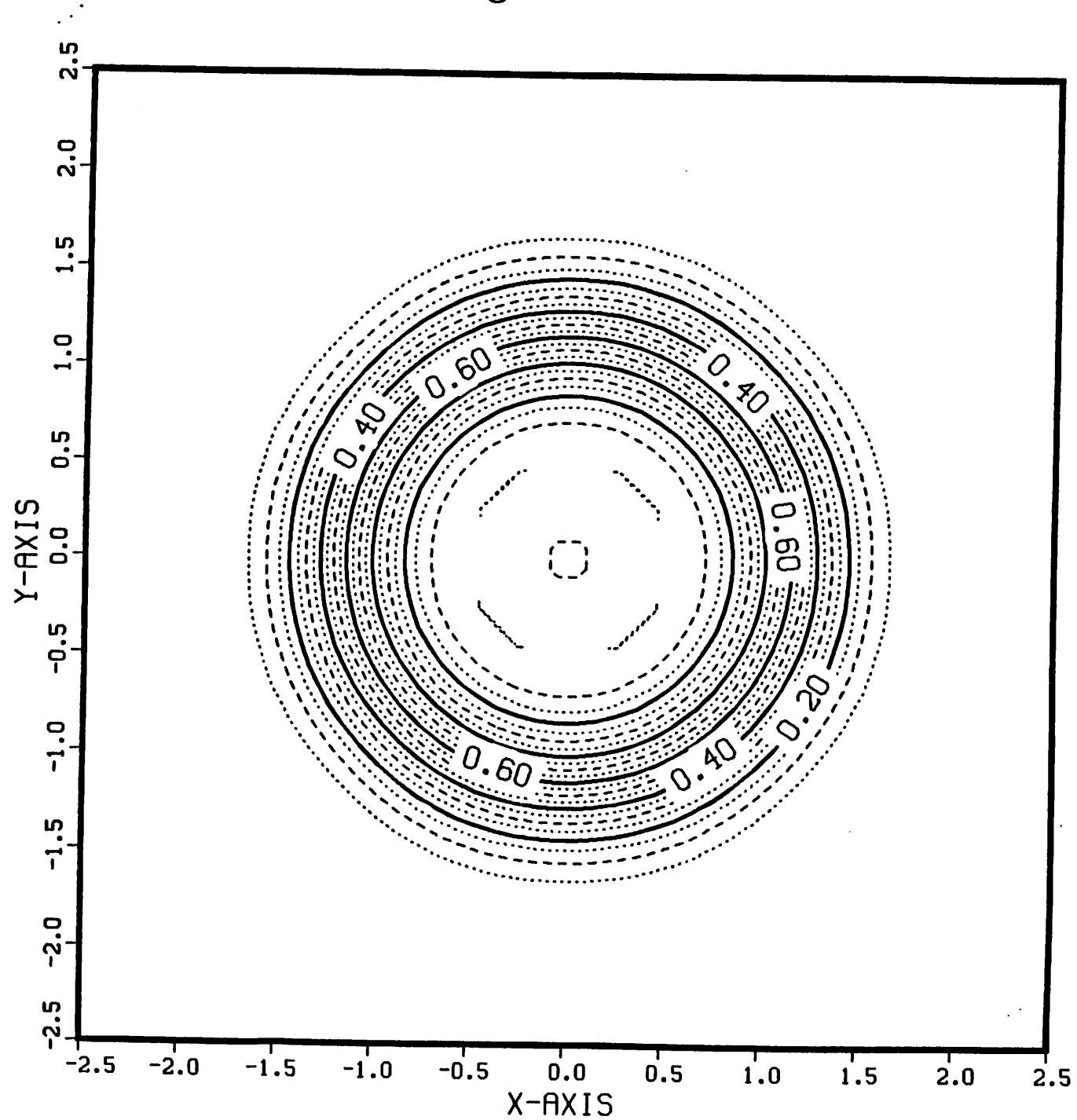
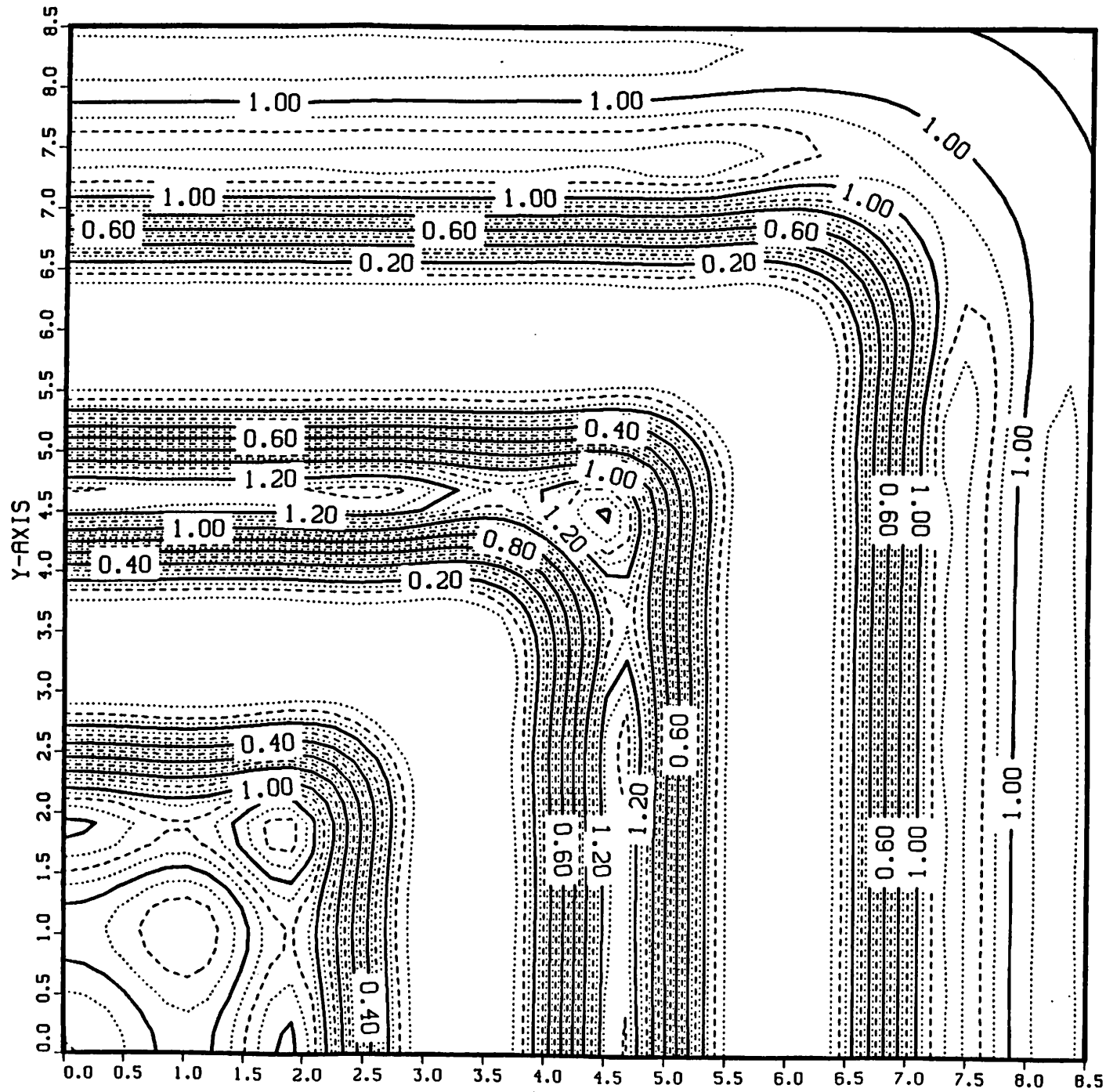


Figure 11



NESTED ELBOWS : SIGMA-0.7

Figure 12

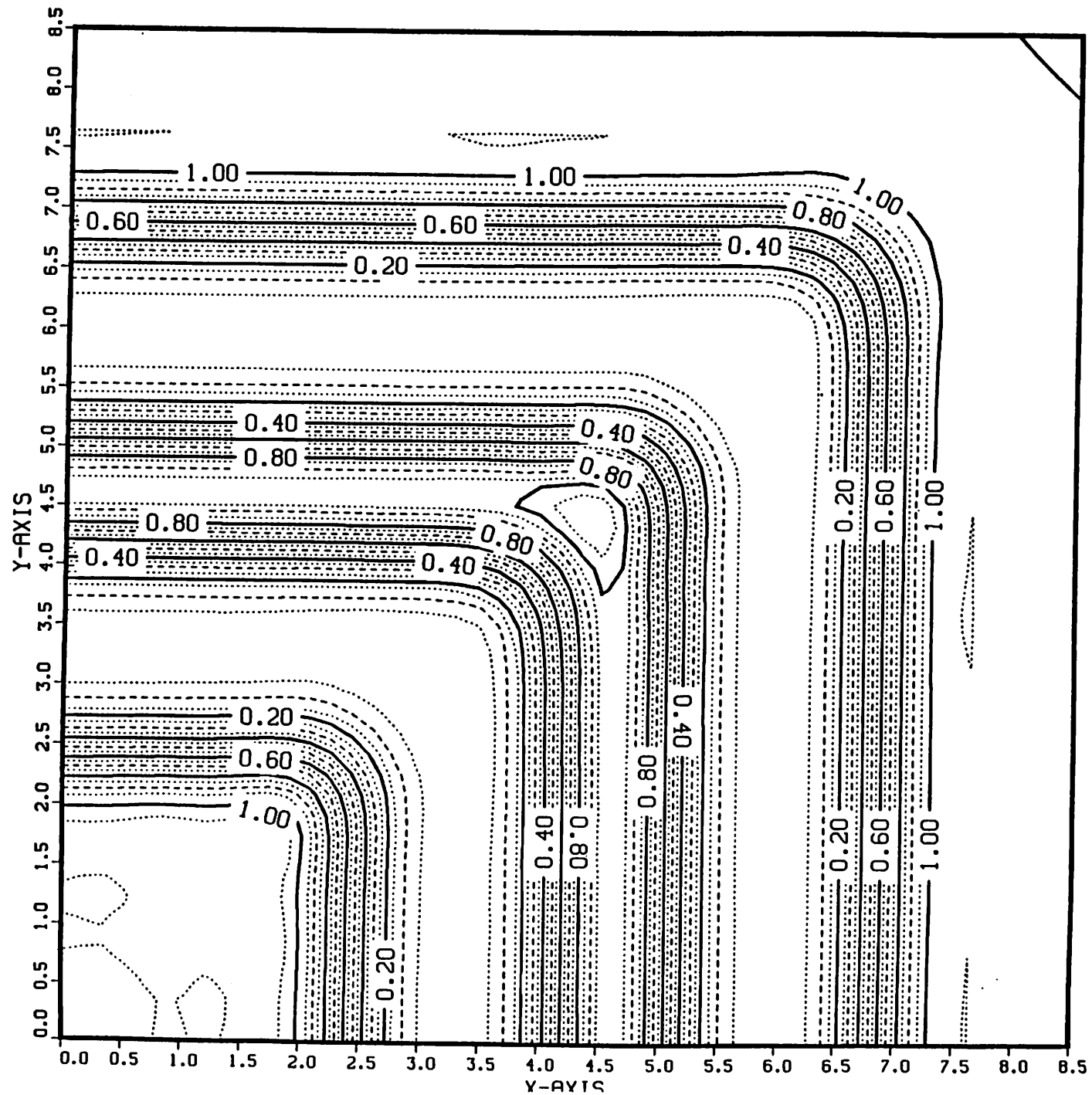
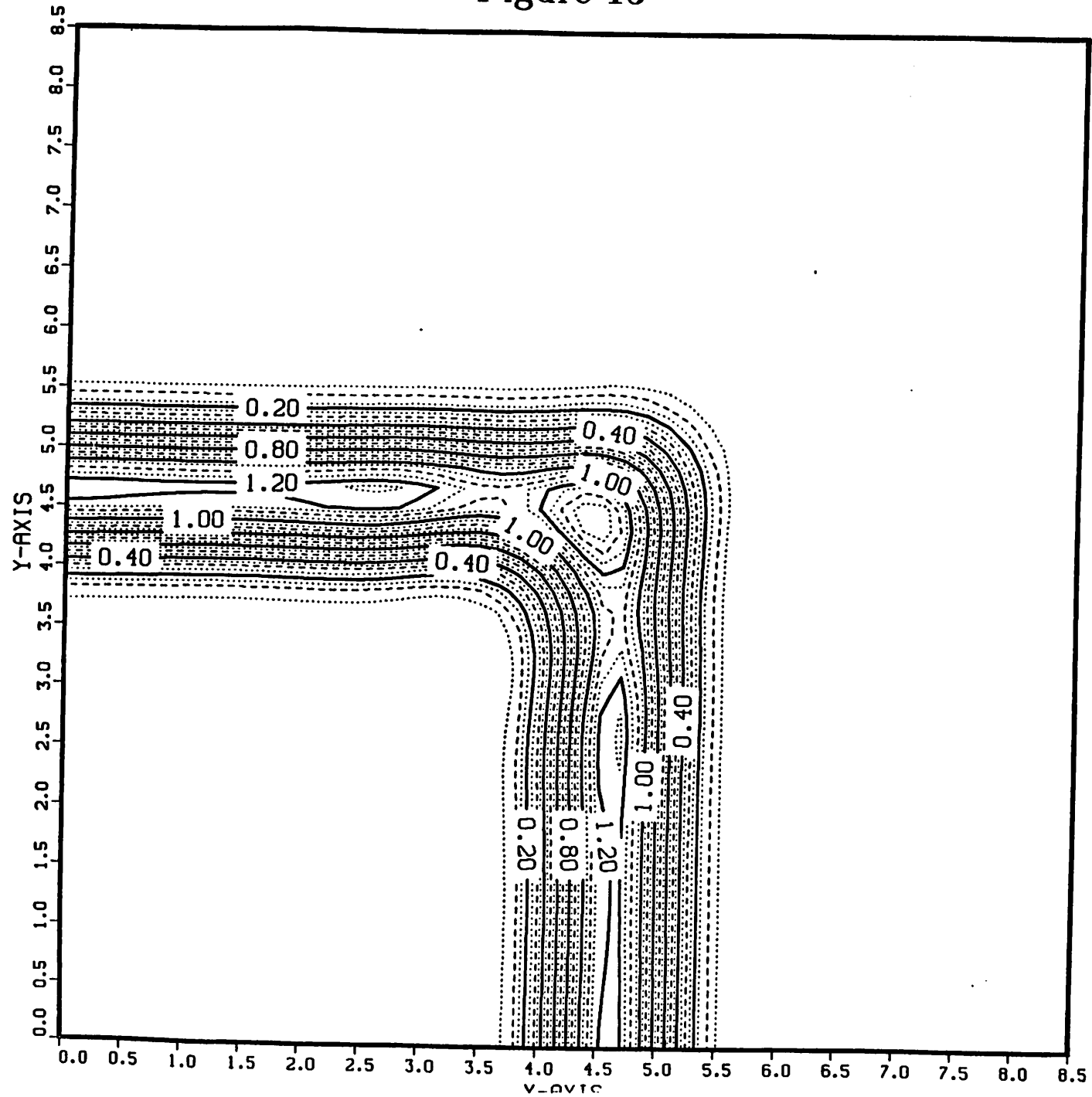
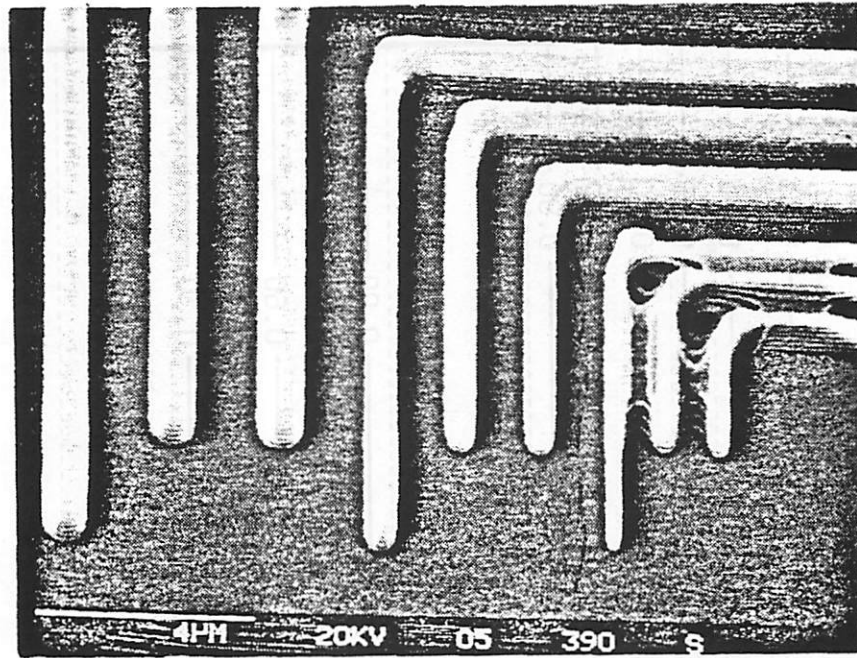


Figure 13



Elbows with 0.5 μm . line/space

Figure 14



#30 Mitte

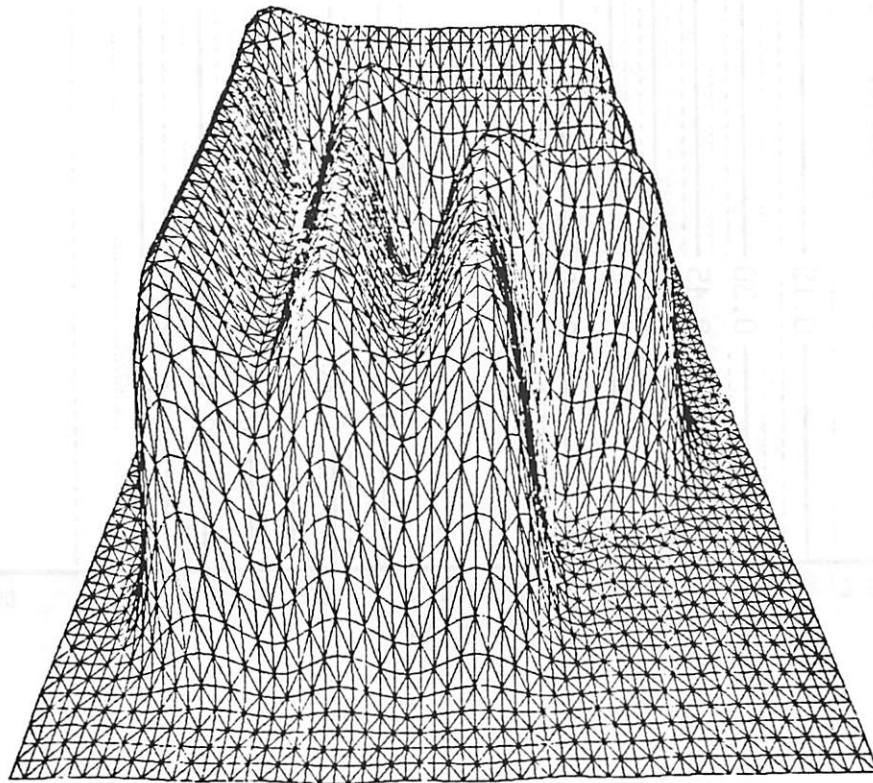
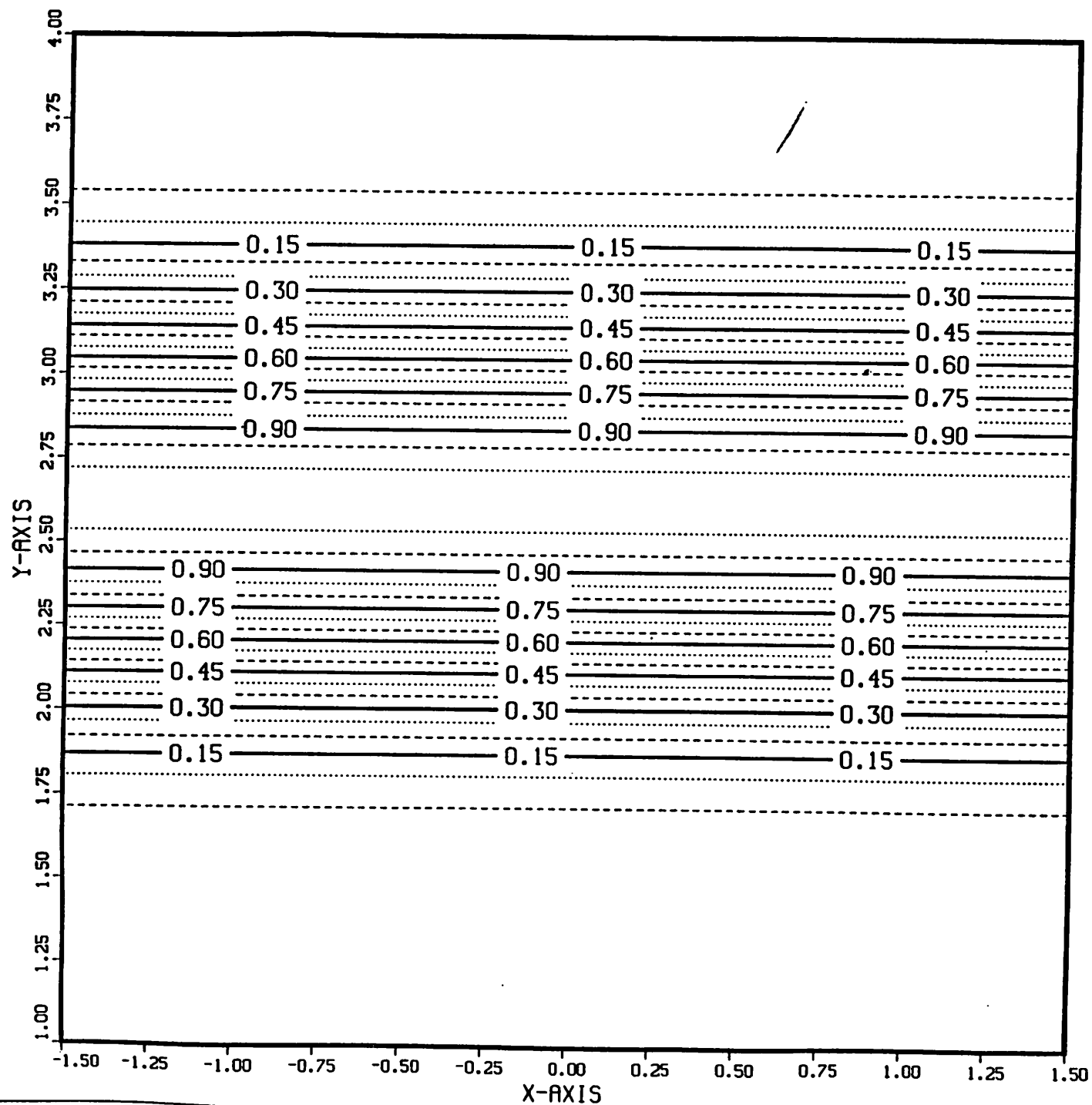


Photo courtesy H. Meyer and E. Loebach of Censor, Lichtenstein

Figure 15



1.25 MICRON LINE WITH 0.5X0.1 INTRUSION

Figure 16

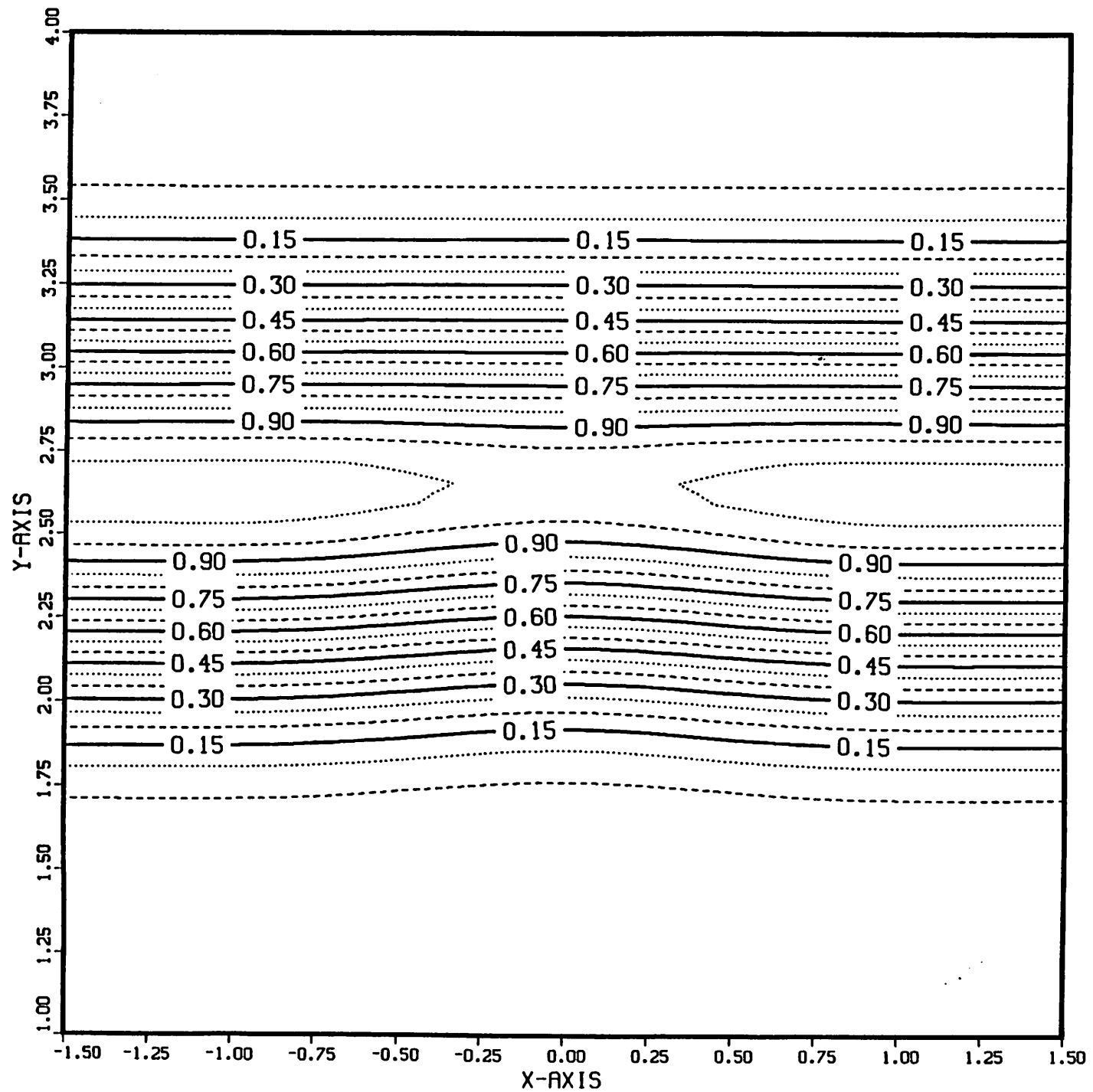
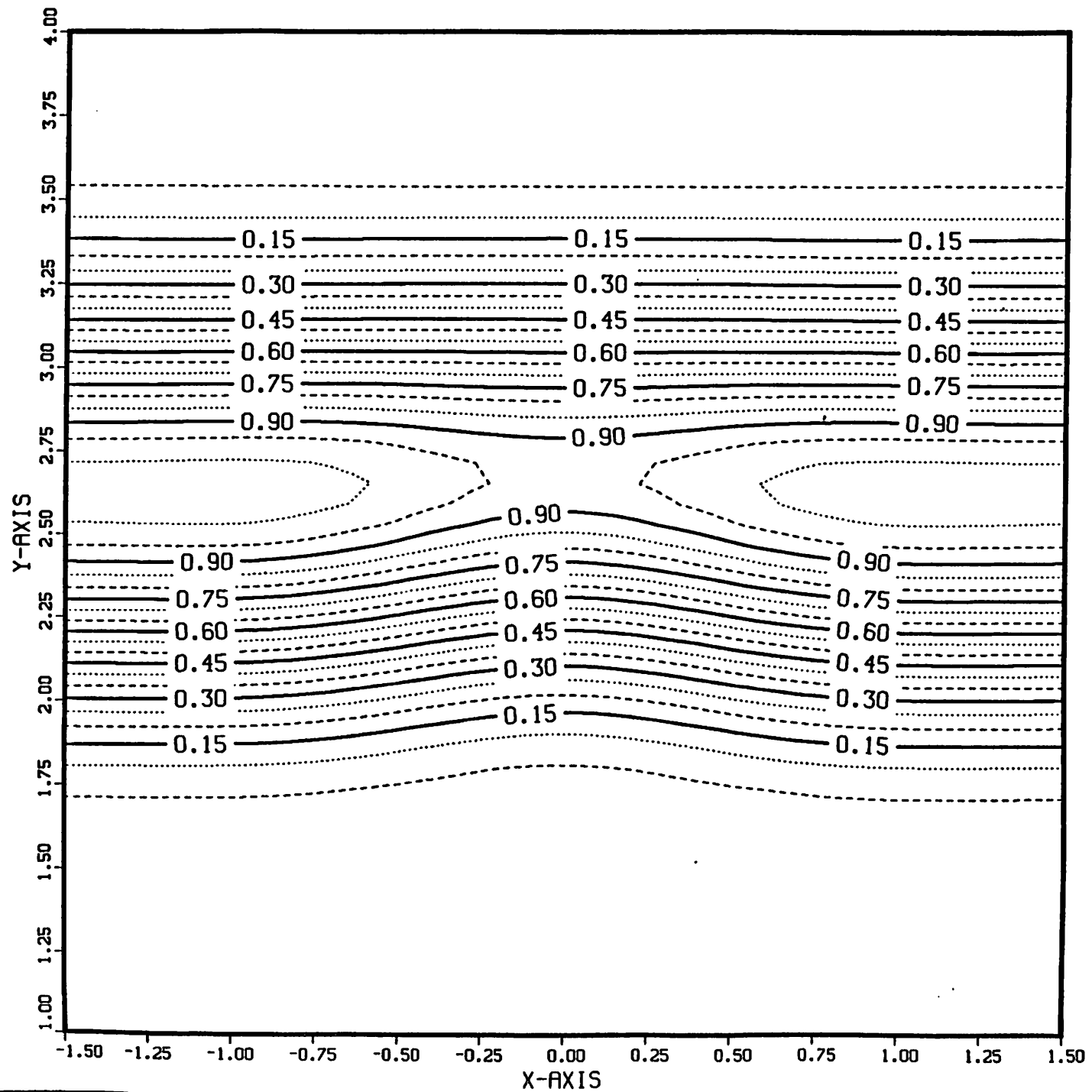
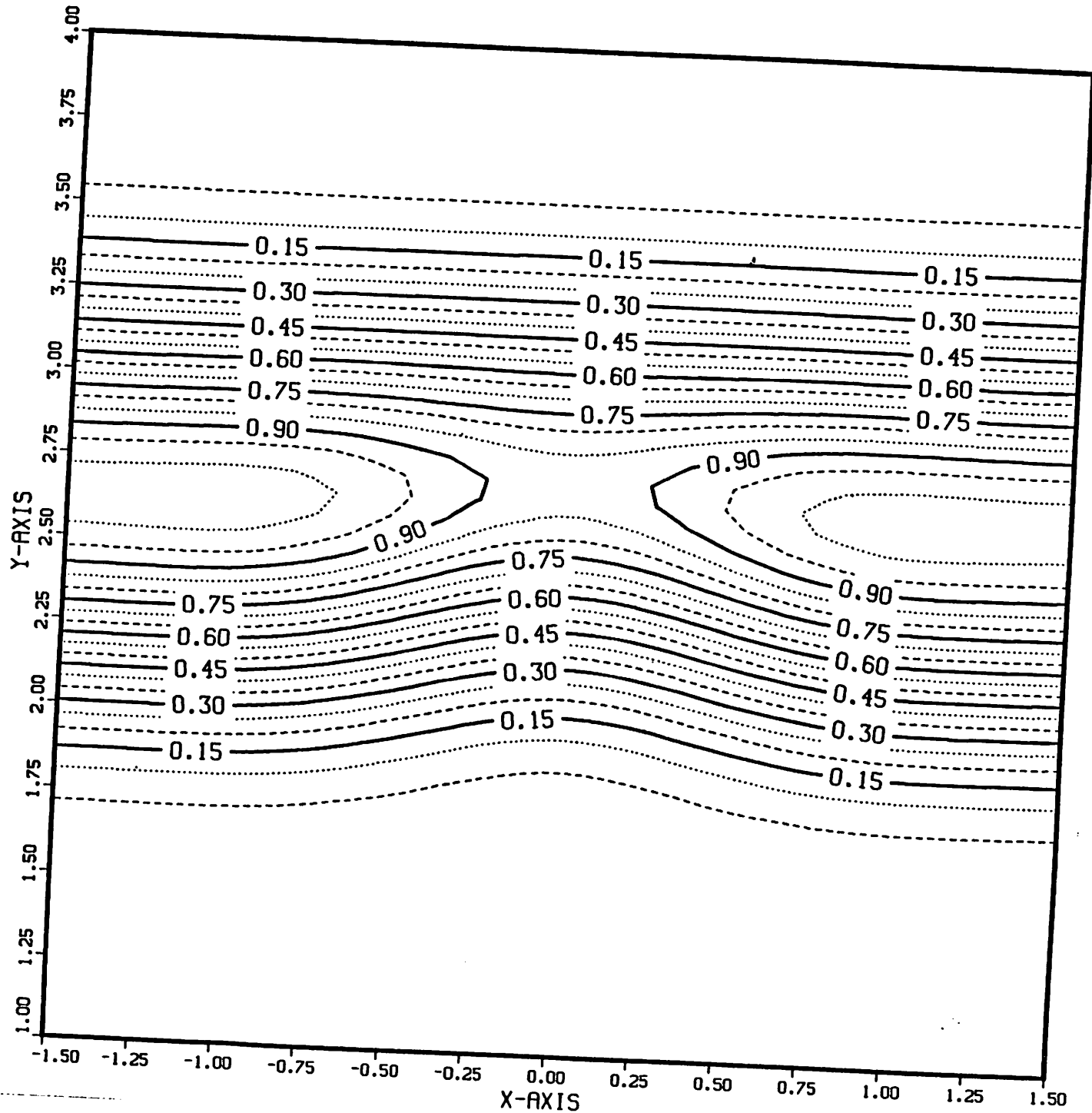


Figure 17



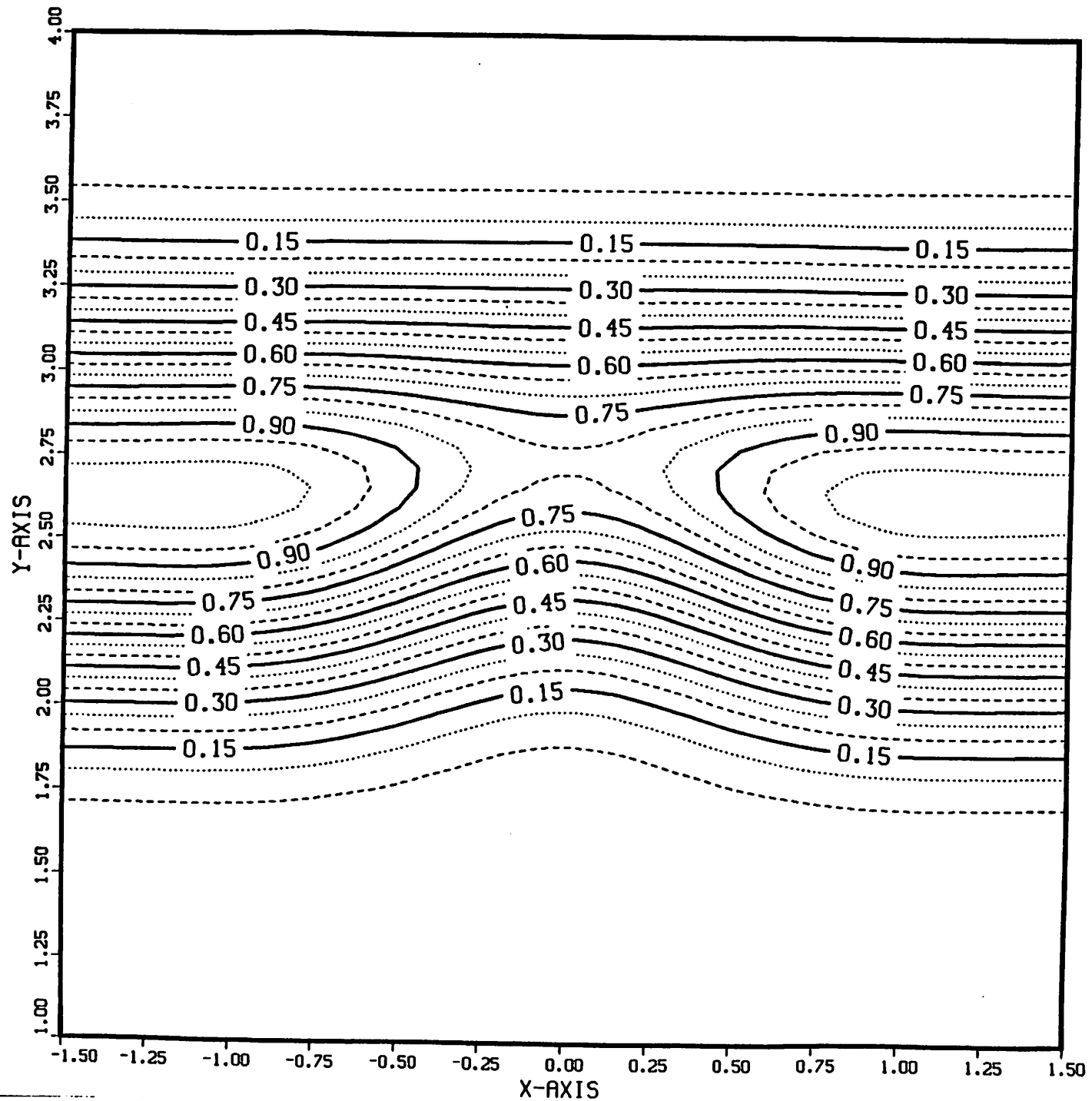
1.25 MICRON LINE WITH 0.5X0.3 INTRUSION

Figure 18



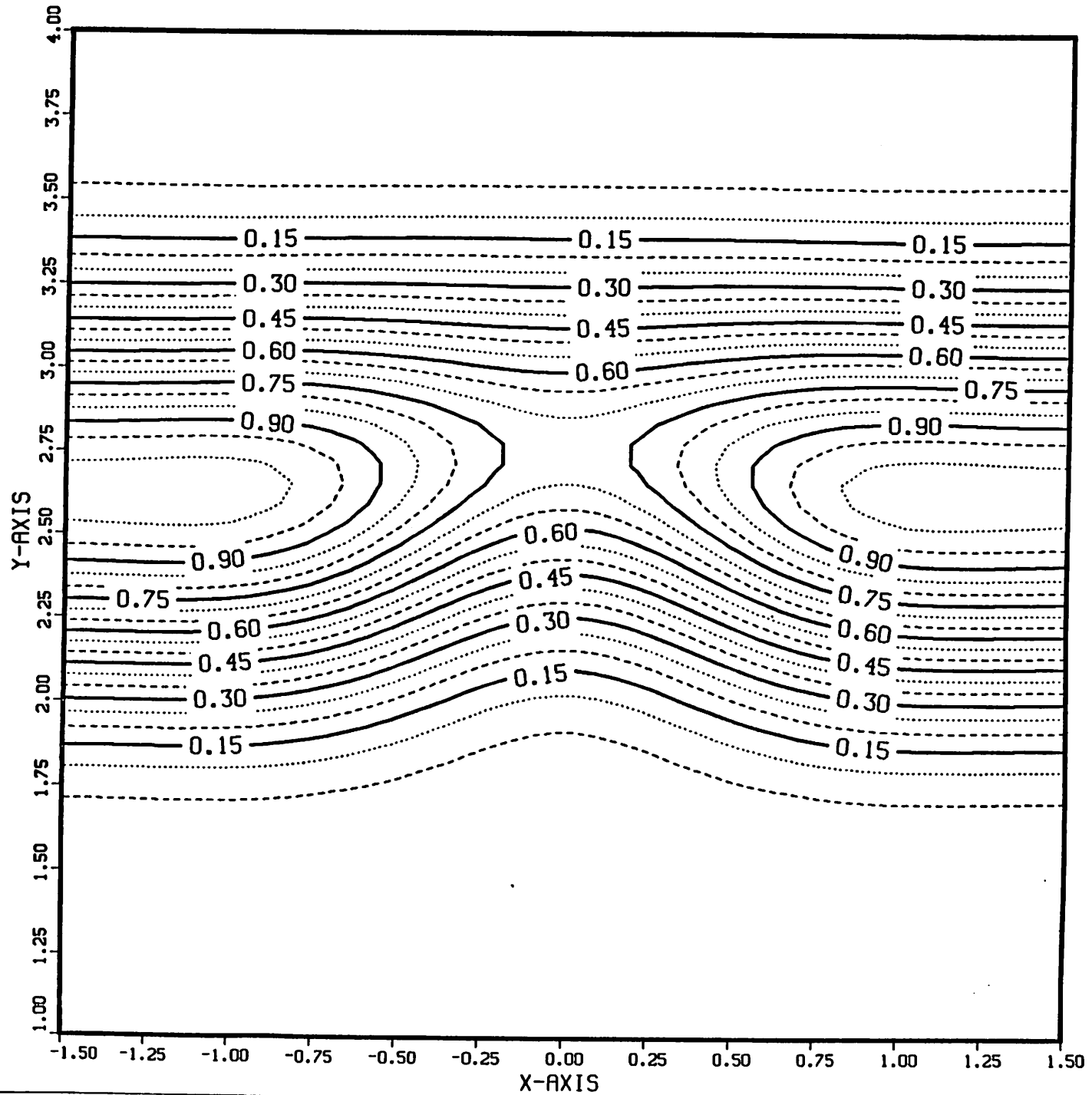
1.25 MICRON LINE WITH 0.5X0.4 INTRUSION

Figure 19



1.25 MICRON LINE WITH 0.5X0.5 INTRUSION

Figure 20



Appendix II — User manual

Using the 2D optical imaging program

The program ("2D") calculates the light intensity at the image plane relative to the [uniform] intensity at the mask using the same algorithms as SAMPLE, extended to handle two dimensional objects. The mask is specified as a set of rectangles and triangles whose size, position and relative transmittance (from -1 to 1) are specified by the user. Relative intensities of overlapping portions of rectangles add. See the example below.

To simplify calculations, the program makes an even periodic extension in both x and y directions of the user specified mask. See example below and Figure 2 for an example.

2D expects input in the form of trial statements like SAMPLE. There is no keyword mapping in 2D so all trial numbers must be specified. 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 sign (#). The pound sign indicates that the remainder of the line is a comment to be ignored by 2D. Example : All of the following inputs cause the same action (set light source wavelength to .436 microns and numerical aperture to .28) :

Example 1 2 .436; 3 .28;

Example 2 Trial 2 : wavelength = .436 microns Trial 3 : na = .28;

Example 3 Trial 2 : Set wavelength & to .436 microns; Trial 3 : .28 # Set numerical aperture to .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 the trial statement.

Trial Statements

A list of the trial statements recognized by 2D 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.

1 — Set print level
 Trial 1 : printlevel;

- * Printlevel is the level of diagnostic output generated by the program.
 - * For interactive use, you should set the printlevel to 3.
 - * Printlevel 2 causes the program to echo the input lines to the output.
 - * Printlevel 3 causes the program to report when trial statement execution is completed by displaying the trial parameters.
- 2 — Set lambda (wavelength) in micrometers
Trial 2 : lambda;
- * This tells the program the wavelength of the light used to illuminate the mask.
- 3 — Set numerical aperture
Trial 3 : NA;
- 4 — Set defocus in micrometers
Trial 4 : defocus;
- * Positive defocus means the focus is above the image plane.
- 5 — Set coherence factor s
Trial 5 : s;
- 6 — Set working area in micrometers
Trial 6 : x y [transmittance];
- * The working area is the size of the field which is to be imaged. 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.
 - * The default for transmittance is 0.
- 7 — Add rectangular aperture to mask
Trial 7 : xllc yllc xlen ylen transmittance;
- * Add a rectangular feature to the mask. Rectangles may overlap other
 - * rectangles - it is the users responsibility to make sure no part of
 - * the mask has transmittance greater than 1 or less than 0.
 - * xllc = x coordinate of lower left corner of rectangle
 - * yllc = 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.
- 8 — Add triangular aperture to mask
Trial 8 : xllc yllc xlen ylen transmittance;
Trial 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 users 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.
- * xllc = x coordinate of corner of right triangle
- * yllc = 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.

9 - Calculate n by n mask transmittance and save

Trial 9 : nx ny [xllc yllc xlen ylen] [diff] ['filename'];

- * Calculate the transmittance profile of the mask. The transmittance at
- * any point is the sum of the working area transmittance (see trial 6) and
- * the transmittances of all rectangles defined by trial 7 statements that
- * cover the point. The output is 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.
- * nx = number of divisions along x axis
- * ny = number of divisions along y axis
- * 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
- * diff = 1 to limit spatial frequencies (normally not used)
- * filename = name of file into which to store the calculated intensities.

* The format of the file is as follows :

* The first line has nx.ny.xllc.yllc and the x and y step size (= len/(n-1)).

* The remaining lines have the intensities, one per line, starting

* at (xllc,yllc). The x index moves fastest. An example :

```

  2 4 0.  0.  4.000000 1.333333
0.004                                     - intensity at (xllc+0*xs,yllc+0*ys)
0.011                                     - intensity at (xllc+1*xs,yllc+0*ys)
0.017                                     - intensity at (xllc+0*xs,yllc+1*ys)
0.005                                     - intensity at (xllc+1*xs,yllc+1*ys)
0.012                                     - intensity at (xllc+0*xs,yllc+2*ys)
0.010                                     - intensity at (xllc+1*xs,yllc+2*ys)
0.004                                     - intensity at (xllc+0*xs,yllc+3*ys)
0.004                                     - intensity at (xllc+1*xs,yllc+3*ys)

```

10 - Calculate image fourier coefficients

Trial 10 ::

- * The program calculates the Fourier coefficients of the image intensity profile. This trial statement can take a long, long time to execute!!
- 11 — Calculate n by n intensity at object and save
 Trial 11 : nx ny [xllc yllc xlen ylen] ['filename'];
 * Calculate the intensity profile in the image plane. The parameters and
 * output file format are the same as for trial 9.
- 12 — Save mask and image coefficients
 Trial 12 : 'filename';
 * Save fourier coefficients that describe mask transmittance and image
 * intensity. The coefficients can be re-loaded with trial 13 to generate
 * more plot files.
- 13 — Load mask and image coefficients
 Trial 13 : 'filename';
 * Load coefficients previously stored with trial 12.
- 14 — Calculate image intensity along a line (generate SAMPLE type plotfile)
 Trial 14 : xi yi xf yf [npts] ['filename'];
 * Calculate intensity profile along the line joining points (xi,yi) and
 * (xf,yf). Number of points is optional - default is 50. Intensities
 * are 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).
 * 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)
- 15 — Plot image intensity contours (IBM3081 w/DISSPLA only)
 Trial 15 : xllc yllc xlen ylen [xsiz ysiz [maj [min [lbls]]]] ['title']
 * Plot intensity contours. This is the DISSPLA version that we
 * run on an IBM 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
 * title = plot title to be printed on plot page

Example

The following example consists of two 2D input files. The first calculates the image for a pair of opaque elbows and stores the calculated coefficients. The second reads the coefficients back and calculates the intensity profiles along a diagonal through the elbow corners and a contour plot of the image intensity on an 8.5x11 inch page.

```
#
# Example 1 :
#
# Calculate image of 4 um. mask with 0.5 um. spaced elbows
#
trial 1: print = 3
trial 2: lambda = .436 um
trial 3: na = .28
trial 4: defoc = .0 um
trial 5: s = .7
trial 6: mask = 4x4 @0
trial 7: cutout = (1.0,0.5) 0.5x2.5 @1
trial 7: cutout = (1.5,2.5) 2.0x0.5 @1
trial 7: cutout = (2.0,0.5) 0.5x1.5 @1
trial 7: cutout = (2.5,1.5) 1.0x0.5 @1
trial 10
trial 12: save = 'out'
end 0
```

```
#
# Example 2:
#
# Read the coefficient file produced by example 1 and get
# a SAMPLE style intensity plotfile along a diagonal through
# the two elbows and a contour plot
#
trial 1 : 2;
trial 13: load file 'elbow4.cof';
trial 14: intensity (0.4) .. (3.1) to 'f77punch7';
trial 15: contours (0,0) 4x4 11x8.5 inches 'Example 2 : Elbows'
trial 0 :
```