

Copyright © 1979, 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.

SIMULATION OF OPTICALLY FORMED IMAGE PROFILES
IN POSITIVE PHOTORESIST

by

Michael Marson O'Toole

Memorandum No. UCB/ERL M79/42

June 1979

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SIMULATION OF OPTICALLY FORMED IMAGE PROFILES IN
POSITIVE PHOTORESIST

Michael Marson O'Toole

Copyright © 1979

Simulation of Optically Formed Image Profiles in Positive Photoresist

Michael Marson O'Toole

Ph.D.

Department of Electrical Engineering and Computer Sciences

ABSTRACT

A program to perform computer simulation of an optical projection printer was written in order to address the problems associated with replicating small lines in positive photoresist. The optical section of the simulation calculates the image intensity due to a mask pattern of periodic lines and spaces. The simulation considers the numerical aperture of the objective lens, multiple wavelength imaging, the presence of focus error, and partially coherent illumination of the mask. A novel algorithm for imaging with partially coherent light, relying heavily on intuitive insight, is presented in detail. A set of routines for the exposure and development of positive photoresist has been adapted and modified for the simulation. The final output of the simulation is a two-dimensional line-edge profile in resist.

The shape and position of the line-edge profile vary significantly with process parameters. Partially coherent mask illumination, an important parameter in projection printing, affects both the image intensity incident on the resist and the resulting developed profiles in the resist. Simulations show a reduced sensitivity of resist linewidth to exposure variation and an increased tolerance of the linewidth to focus error with partially coherent illumination.

ACKNOWLEDGEMENTS

I wish to thank the people associated with the SAMPLE program. Sharad Nandgaonkar, whose pleasant manner, patience, and attention to detail made working with him very enjoyable, is largely responsible for the structure of my portion of the program. I wish to thank Professor W.G Oldham for his many insightful comments, which added greatly to the direction of the project. Thanks are due also to Shankar Subramanian and John Reynolds for their help and suggestions.

I especially wish to thank my advisor, Professor Andy Neureuther, for his guidance, support, and advice throughout my graduate years. In discouraging times, his enthusiasm and encouragement were of great value to me.

Research sponsored by the National Science Foundation grant
ENG77-14660-A01.

TABLE of CONTENTS

Chapter		page
1	Introduction	1
	References for Chapter 1	4
2	The IMAGE Machine: Theory	5
	2.1 The Numerical Aperture and $F^\#$ of a Lens	5
	2.2 The Mask	6
	2.3 Imaging with Coherent Light	7
	2.3.1 The Coherent Transfer Function (CTF)	7
	2.3.2 The Effect of Focus Error on the CTF	8
	2.3.3 The Effect of Focus Error in the Space Domain	11
	2.3.4 Construction of the Image	11
	2.4 Imaging with Incoherent Light	11
	2.4.1 The Optical Transfer Function (OTF)	12
	2.4.2 The Effect of Focus Error on the OTF	12
	2.4.3 Construction of the Image	12
	2.5 Imaging with Partially Coherent Light	13
	2.5.1 The Condenser System	14
	2.5.2 The Imaging System	17
	2.5.3 Construction of the Image	18
	2.5.4 Polarization Effects	22
	References for Chapter 2	23
3	The IMAGE Machine: Code	24
	3.1 Standard Routines	24
	3.2 Partial Coherence Routines	26
	References for Chapter 3	31
4	The EXPOSE Machine	32
	4.1 Theory	32
	4.2 Code	33
	References for Chapter 4	36
5	The DEVELOP Machine	37
	5.1 Theory	37
	5.2 Code	37
	5.3 Problems with the DEVELOP Machine	42
	References for Chapter 5	43
6	The Influence of Partial Coherence on Projection Printing	44
	6.1 The Effects of Partial Coherence on the Image	44
	6.2 The Effects of Partial Coherence on the Resist Line Edge Profiles	45
	6.3 Further Considerations	47
	6.4 Conclusions	47
	References for Chapter 6	48

Appendix		page
A		Conventional Partial Coherence Theory 49
	A.1	Review of Partial Coherence Theory 49
		A.1.1 Some Definitions 49
		A.1.2 The van Cittert-Zernike Theorem 50
		A.1.3 Schnell's Theorem 51
		A.1.4 Method for Calculating the Image Intensity 51
	A.2	Further Discussion of Imaging with Partially Coherent Light 52
		A.2.1 Linear Systems Approach 53
		A.2.2 One Dimensional Imaging 55
		References for Appendix A 56
B		Resolution and The Uncertainty Principle 58
C		Common Block Documentation 59
D		SAMPLE Code (Lab) 70
E		The String Point Insertion Algorithm 98
F		SAMPLE Control Cards 99

Chapter 1

Introduction

The advent of projection mask aligners into integrated circuit fabrication has created a need to understand the various processing problems involved¹⁻¹⁸. Since controlled experimentation in the lab is both costly and time consuming, computer simulation of laboratory processes is quite useful. A simulation also has the inherent advantage of separating a particular parameter from the parameters of the entire process.

Simulation has been used by a number of authors to investigate the effect of a specific parameter on projection printing. In particular, the work of Hershel^{15,16} briefly addresses the role of partial coherence in projection printing. Narasimham and co-workers^{17,18} have simulated the effects of polychromatic exposure and of defocus. A key component in these simulations is the model for the photosensitive material; and much effort has been spent in characterizing the photoresists used by the integrated circuit industry¹⁹⁻²⁵. A useful process simulator must incorporate a valid model for the exposure and development of the resist.

A general process simulator called SAMPLE, simulation and modeling of profiles for lithography and etching, has been written as an ongoing group effort at the University of California at Berkeley. As shown in figure 1-1, SAMPLE consists of the Lab, containing the code for the simulation, and of the User-Interface²⁶, which controls the simulation with a simple set of user-oriented commands. The current version of the Lab is the subject of the first part of this dissertation. The second part of the dissertation uses SAMPLE to explore the effect of partial coherence on projection printing. SAMPLE is written in standard FORTRAN and is easily transferred to mini-computer systems with overlaying capability.

Depicted in figure 1-2, the first part of the Lab models a projection mask aligner. The condenser system of the printer illuminates the mask, and the imaging system projects the mask on a wafer coated with positive photoresist. Several layers between the resist and the substrate are allowed. After exposure, the wafer is removed from the aligner and developed.

The Lab consists of three main subsections called machines. Each machine is self-contained and produces output in common blocks accessible to the other machines. Each machine is controlled by a *sub-controller*, a subroutine with the same mnemonic name as the machine. The sub-controller sets various internal machine parameters and calls on the routines needed to solve the problem defined by the input data. Each machine has its own message routine, which contains system printouts of three classes--errors, warnings, and messages. The breaking up of the Lab simulation into separate machines provides a great deal of flexibility for adding additional features and minimizes computational redundancy.

The IMAGE machine of SAMPLE calculates the image intensity distribution incident on the resist, taking into account the effects of the numerical aperture of the objective lens, focus error, multiple wavelength exposure, and partially coherent mask illumination. The resist is then exposed via the EXPOSE machine, which uses the positive resist model of Dill and co-workers^{7,8,19}. The primary quantity of the Dill model is $M(x,z,t)$, which indicates the amount of exposure at the resist point (x,z) at time t . The resist is then developed by the DEVELOP machine, which uses a modified version of the string algorithm of Jewett²⁷. The development rate at point (x,z) in the resist is calculated via an experimentally determined rate function $R[M(x,z)]$ ^{7,20}. The final output is a two dimensional line-edge profile in resist. The shape and position of the line-edge profiles produced by the simulator vary significantly with process parameters.

See Appendix C: Common Block Documentation.

One of the important process parameters in projection printing is the degree of illumination coherence^{11,15,16}. In the laboratory the degree of coherence is controllable by inserting a stop in the condenser system of the projection printer. However, subsequent process variations, such as wafer bow and resist thickness, are not easily controllable in the laboratory and are even less controllable in production facilities. With the use of SAMPLE, the effect of partial coherence is separable from the other variables in the process. The effect of partially coherent mask illumination on the image intensity distribution and on the resulting resist profiles is discussed in Chapter 6. Simulations using SAMPLE show a reduced sensitivity of the resist linewidth to exposure variation and an increased tolerance of the linewidth to focus error with partial coherence. SAMPLE is used to indicate a reasonable operating value for the partial coherence parameter σ .

Chapters 2 through 5 describe each machine in detail. The theory is discussed first; and a detailed explanation of the code follows the theoretical discussion. The text should be read in conjunction with a copy of the code, appendix D, and of the Common Block Documentation, appendix C. Appendix F contains a list of User-Interface control cards that are compatible with the version of SAMPLE discussed in the thesis.

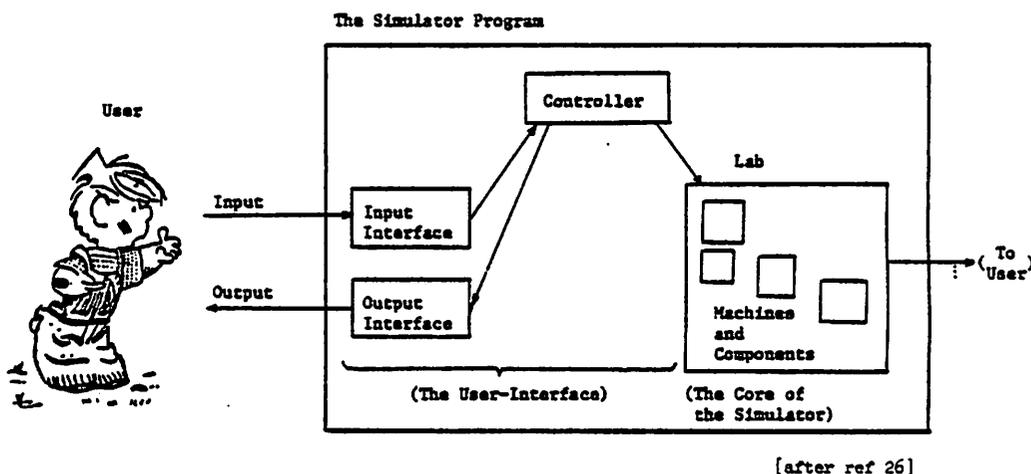


Figure 1-1

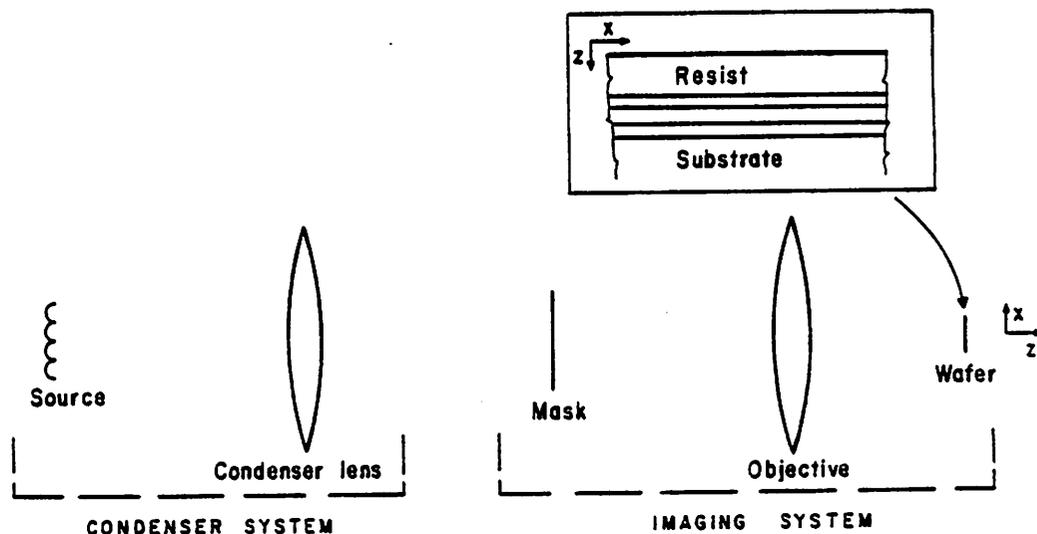


Figure 1-2

References for Chapter 1:

- 1) J.D. Cuthbert, "Optical Projection Printing", *Solid State Technology*, Aug 1977, p 59.
- 2) F.H. Dill, "Optical Lithography", *IEEE Trans. Elect. Dev.*, ED-22, no. 7, pp 440-444, July 1975.
- 3) J.W. Bossung, "Projection Printing Characterization", *Proc. of SPIE*, 100:80, April 1977.
- 4) W.T. Novak, "A New VLSI Printer", *SPIE Proceedings*, vol. 135, April 1978.
- 5) B.J. Lin, "Deep UV Lithography", *J. Vac. Sci. Tech.*, vol. 12, p 1317-1320, 1975.
- 6) M.A. Narasimham, "Projection Printed Photolithographic Images in Positive Photoresists", *IEEE Trans. on Elect. Dev.*, vol. ED-22, no. 7, July 1975.
- 7) F.H. Dill, A.R. Neureuther, J.A. Tuttle, and E.J. Walker, "Modelling Projection Printing of Positive Photoresists", IBM Research, February 2, 1975, RC 5261.
- 8) A.R. Neureuther and F.H. Dill, "Photoresist Modeling and Device Fabrication Applications", *Optical and Acoustical Micro-Electronics*, Polytechnic Press, N.Y., pp 233-249, 1974.
- 9) A. Brochet, G. Dubroeuq, "Modeling of Positive Resists-- Application the the Projection of Thin Structures on Silicon", *Proceedings of the International Conference on Microlithography*, Paris, France, June 1977.
- 10) A.R. Neureuther, M. O'Toole, W.G. Oldham, and S. Nandgaonkar, "Use of Simulation to Optimize Projection Printing Profiles", *153rd meeting of the Electrochemical Society*, Seattle, Wash., May 21-26 1978.
- 11) M.M. O'Toole and A.R. Neureuther, "The Influence of Partial Coherence on Projection Printing", *SPIE Developments in Semiconductor Microlithography IV*, San Jose, Ca., April 1979.
- 12) E.J. Walker, "Reduction of Photoresist Standing-Wave Effects by Post-Exposure Bake", *IEEE Trans. on Elect. Dev.*, vol ED-22, no 7, pp 464-466, July 1975.
- 13) B.J. Cavanello, M. Hatzakis, and J.M. Shaw, "Process for Obtaining Undercutting of a Photoresist to Facilitate Lift Off", *IBM Technical Disclosure Bulletin*, vol 19, no 10, March 1977.
- 14) D.A. McGillis and D.L. Fehrs, "Photolithographic Linewidth Control", *IEEE Trans. on Elect. Dev.*, vol. ED-22, no. 7, July 1975.
- 15) R. Hershel, "Partial Coherence in Projection Printing", *SPIE vol. 135, Developments in Semiconductor Microlithography III*, 1978.
- 16) Ron Hershel, "Effects of Partially Coherent Imagery on Resist Profiles", *Kodak Microelectronics Seminar, Interface '78*, Oct 1-3 '78, San Diego, Ca.
- 17) M.A. Narasimham and J.H. Carter, Jr., "Effects of Defocus on Photo- lithographic Images Obtained with Projection-Printing Systems", *SPIE Semiconductor Microlithography III*, 1978.
- 18) M.A. Narasimham, "Projection Printed Images in Positive Photoresists Under Polychromatic Exposures", *SPSE 30th Annual Conference*, May 1977, North Hollywood, Ca.
- 19) F.H. Dill, W.P. Hornberger, P.S. Hauge, and J.M Shaw, "Characterization of Positive Photoresist", *IEEE Transactions on Electron Devices*, vol ED-22, no. 7, July '75, p 445.
- 20) J.M. Shaw and M. Hatzakis, "Performance Characteristics of Diazo-Type Photoresists Under e-Beam and Optical Exposure", *IEEE Trans. on Electron Devices*, vol ED-25, no 4, pp 425-430.
- 21) J.M. Shaw and M. Hatzakis, "Developer Temperature Effects on Positive Photoresists", *Proceedings of the Kodak Microelectronics Seminar*, San Diego, October 1978.
- 22) F.H. Dill and J.M Shaw, "Thermal Effects on the Photoresist AZ1350J", *IBM J. Res. Dev.*, vol 21, May 1977, pp 210-218.
- 23) P.D. Blais, "Edge Acuity and Resolution in Positive Type Photoresist Systems", *Solid State Technology*, pp 75-80, August 1977.

- 24) H.N. Keller, "Functional Testing of Positive Photoresist for Manufacture of Film Integrated Circuits", *Solid State Technology*, pp 45-53, June 1978.
- 25) W.G. Oldham, "In Situ Characterization of Positive Resist Development", *SPIE Proceedings*, vol 135, April 1978.
- 26) S.N. Nandgaonkar, "Design of a Simulator Program(SAMPLE) for IC Fabrication", M.S. Report, Dept. of EECS, U.C. Berkeley, 1978.
- 27) R.E. Jewett, P.I. Hagouel, A.R. Neureuther, and T. Van Duzer, "Line-Profile Resist Development Simulation Techniques", *Polymer Engineering and Science*, vol. 14, no. 6, p 381-384, June 1977.

Chapter 2

The IMAGE Machine: Theory

The first machine of the Lab, IMAGE, images a mask pattern on the surface of the resist for a specified numerical aperture, degree of illumination coherence, and focus error. The mask pattern, consisting of a rectangular grating in one dimension, is reduced to a set of spatial frequencies using Fourier analysis. The mask spatial frequencies are modified by the optical system and used to construct an image on the resist surface.

The type of illumination coherence and the presence or absence of focus error determine the manner in which the mask spatial frequencies are modified. Coherent illumination of the mask with no focus error is modeled by weighting the frequencies of the field at the mask according to the coherent transfer function, or CTF¹⁻⁵. If the resist surface is displaced from the image plane (focus error), a phase factor proportional to the square of the spatial frequency is added to the CTF. For incoherent mask illumination with the resist in the plane of perfect focus, the frequency weightings are multiplied by the optical transfer function, OTF¹. For a defocused imaging system, the OTF is changed accordingly². Partially coherent illumination is modeled by adding the contribution of each source point to the resulting intensity pattern on the resist. Under perfect focus conditions, sets of adjacent source points can be grouped together to reduce computation time. Multiple wavelength exposures are modeled by computing an image pattern for each wavelength separately and combining the patterns in proportion to the relative intensities of the wavelengths.

The output of IMAGE is a one dimensional image in x of the normalized intensity distribution incident on the resist surface; the section of the image displayed is user-definable. User-commands can suppress the image plot, request an output of the OTF, request punched cards for the image data, and request diagnostics.

2.1 The Numerical Aperture and $F^\#$ of a Lens

The terms *numerical aperture* and $F^\#$ are used frequently in the optical simulation discussion. An explanation of these terms will clarify what is meant. Figure 2.1-1 depicts the physical situation under discussion.

The $F^\#$ of the objective is a description of the light gathering ability, or "speed", of the lens.

$$F_\infty^\# = \frac{f}{D} \quad (2.1-1)$$

where f is the focal length of the lens and D is the diameter of the lens. The $F^\#$ usually refers to the image side of the lens, since it is used as a measure of how quickly the image can expose a photosensitive material. The subscript ∞ means that the object is at infinity.

The numerical aperture of the lens specifies the highest spatial frequency accepted by the lens. One can refer to the *image-space* numerical aperture or to the *object-space* numerical aperture. Both numerical apertures vary with the magnification M of the system. However, as with the $F^\#$, simply referring to the numerical aperture generally means the numerical aperture with the image or the object at infinity. The image-space or object-space reference is usually understood in context. For example, image-space is meant in a camera specification and object-space is meant in a microscope specification. The notation used here for the numerical aperture of the lens with the image at infinity is $(NA)_\infty$.

$$(NA)_{\infty} = \frac{D}{2f} = \frac{1}{2F_{\infty}^{\#}} \quad (2.1-2)$$

When neither the object nor the image are at infinity, the equation for the numerical aperture takes on a slightly different form. From figure 2.1-1, the numerical aperture in object-space can be defined as

$$(NA)_{ob} = \sin \alpha_{ob} \approx \frac{D}{2d_{ob}} \quad (2.1-3)$$

From the lens law,

$$d_{ob} = \frac{d_{im}f}{d_{im}-f} \quad (2.1-4)$$

Substituting for d_{ob} in equation (2.1-3) and defining the magnification M as

$$M \equiv \frac{d_{im}}{d_{ob}} \quad (2.1-5)$$

equation (2.1-3) becomes

$$(NA)_{ob} = \left[\frac{D}{2f} \right] \cdot \frac{d_{im}-f}{d_{im}} = \left[(NA)_{\infty} \right] \cdot \frac{1}{1+\frac{1}{M}} \quad (2.1-6)$$

By a similar derivation, the image-space numerical aperture is

$$(NA)_{im} = \sin \alpha_{im} \approx \frac{(NA)_{\infty}}{1+M} \quad (2.1-7)$$

Note that

$$(NA)_{ob} = M \cdot (NA)_{im} \quad (2.1-8)$$

Similar relations hold for the $F^{\#}$ of a lens.

$$F^{\#} = F_{\infty}^{\#} (M+1) \quad (2.1-9)$$

where $F^{\#}$ refers to the image side of the lens. In later text, (NA_o) and (NA_c) refer to the numerical apertures of the objective and condenser lenses, respectively. Note that the magnification M is less than 1 for an image that is reduced in size from the object.

In the IMAGE machine, the numerical aperture of the objective lens in image space with a magnification M is an input parameter. If the objective lens numerical aperture is given by the lens manufacturer as $(NA_o)_{\infty}$, the manufacturer's specification should be divided by $M+1$ as per equation (2.1-7). Projection mask aligners always have a magnification less than 1. As a detailed Fourier optics analysis shows (ref 1, pp 110-3), the object lines are considered to be the same period as the desired image lines, and the transfer function for the system is given in terms of the image-space numerical aperture $(NA_o)_{im}$.

2.2 The Mask

The mask patterns used by the optical simulation are one dimensional periodic bars in x , shown in figure 2.2-1. The mask pattern simulates the mask of chromium on glass used in the manufacture of integrated circuits. A pattern is considered a *space* if the transparent width of the mask pattern is specified by the user, a *line* if the opaque part is specified, and a *linespace* if the transparent and the opaque parts are both specified. In all cases the pattern is periodic, and the specification of a line(or space) generates a periodic pattern of the specified line(or space) and a large space(or line). The distance between lines(or spaces) is determined by the

Figure 2.1-1

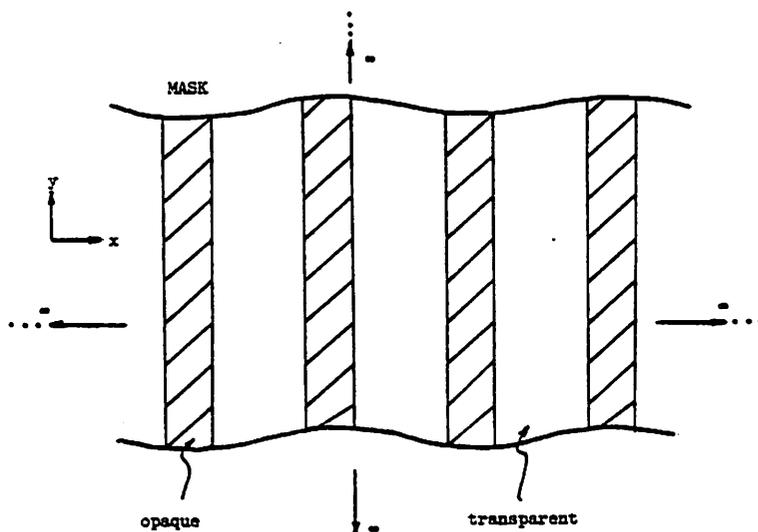
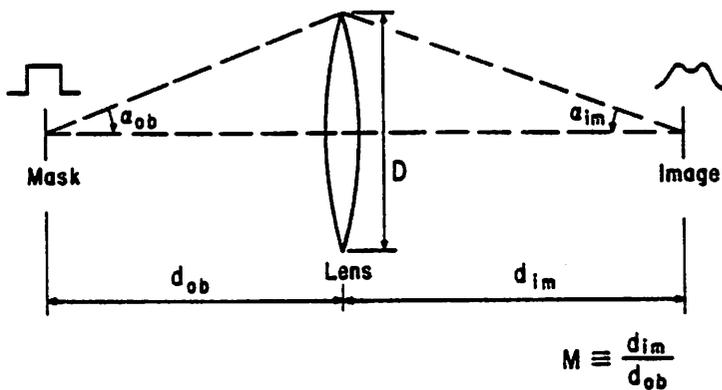
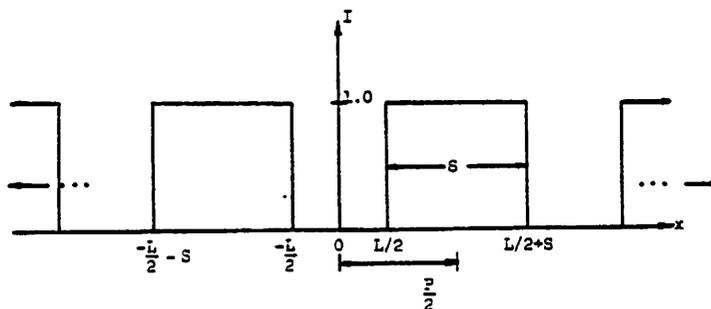


Figure 2.2-1

Figure 2.2-2



maximum number of spatial frequencies the program can handle--currently 41.

In order to determine the spatial frequencies of the mask pattern, a Fourier analysis is carried out in the x dimension. Figure 2.2-2 shows the intensity at the mask of a typical line of linewidth L , spacewidth S , and period $P (=L+S)$. The pattern is even around the $x=0$ axis; only cosine terms will be present in the pattern. Using standard Fourier series equations, the pattern is described by $f(x)$.

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos \frac{n\pi x}{L} \quad (2.2-1a)$$

$$\text{where } a_n = \frac{4}{P} \int_0^{\frac{P}{2}} f(x) \cos \frac{2n\pi x}{P} dx \quad n=0, 1, 2, \dots \quad (2.2-1b)$$

From figure 2.2-1,

$$a_n = \frac{4}{P} \int_{\frac{L}{2}}^{\frac{L+S}{2}} \cos \frac{2n\pi x}{P} dx \quad n=1, 2, 3, \dots$$

Working out the integral gives

$$a_n = \frac{-2}{n\pi} \sin \left[n\pi \frac{L}{P} \right] \quad n=1, 2, 3, \dots \quad \text{and} \quad \frac{a_0}{2} = \frac{S}{P} \quad (2.2-2)$$

Figure 2.2-3 shows the pattern for a space. The coefficients are given by

$$a_n = \frac{4}{P} \int_0^{\frac{S}{2}} \cos \frac{2n\pi}{P} x dx = \frac{2}{n\pi} \sin \frac{n\pi S}{P} \quad n=1, 2, \dots \quad (2.2-3)$$

$$\text{and} \quad \frac{a_0}{2} = \frac{S}{P}$$

The spatial frequencies ν_{m_i} of the mask are 0(dc) and multiples of the fundamental frequency $\nu_{m_1} = \frac{1}{P}$; they are shown in figure 2.2-4. Depending on whether the illumination coherence is coherent or incoherent, the spatial frequencies represent either the electric field or the intensity in the plane of the mask.

2.3 Imaging with Coherent Light

Imaging with coherent light is quite straightforward when viewed in terms of linear systems concepts. The coherent wave illuminating the mask gives rise into a set of diffracted beams. The objective lens of the imaging system acts as a low pass filter for the spatial frequencies of the electric field of the diffracted beams.

2.3.1 The Coherent Transfer Function (CTF)

For coherent illumination of the mask, the spatial frequencies of the mask represent the frequency in x of the electric field components at the mask. The spatial frequencies of the field at the mask can be translated to a set of travelling waves propagating at particular angles with respect to the x axis. Figure 2.3-1 shows the diffracted beams due to a source frequency of $\nu_s=0$ (dc). Since the mask is periodic in x , the spatial frequencies of the diffracted beams are ν_{m_1} apart, where ν_{m_1} is the fundamental spatial frequency of the mask. Since the diffracted

¹Due to the role that the magnification plays in the derivation of the CTF, the d referred to in figure 2.3-1 is really the lens to image distance d_{im} . However, for illustrative purposes, the magnification M is considered 1

Figure 2.2-3

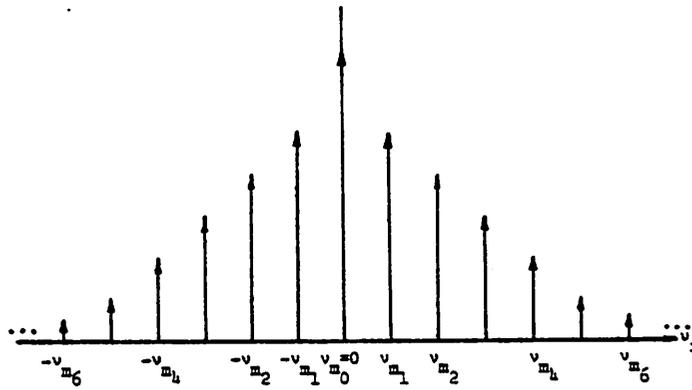
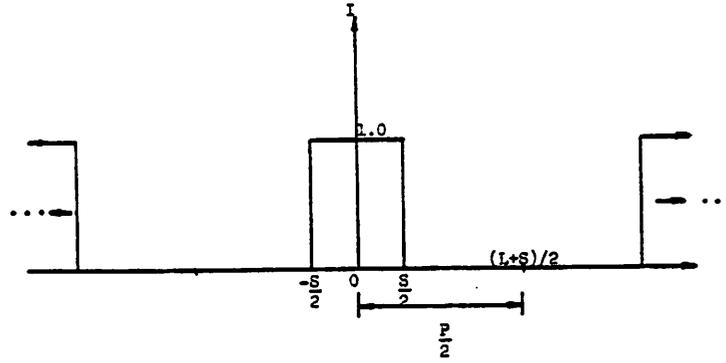
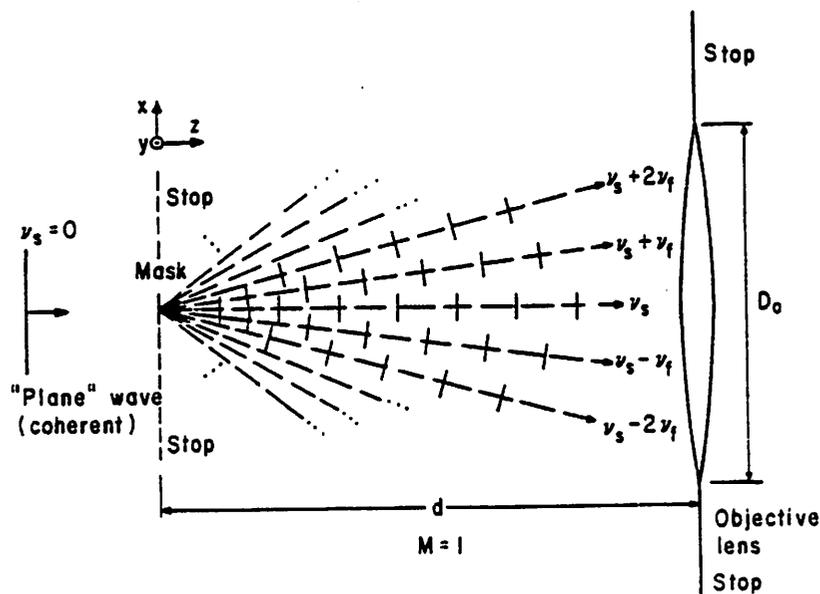


Figure 2.2-4

Figure 2.3-1



beams are assumed small in spatial extent, the beams are either intercepted or missed by the lens aperture. If $D_o \ll d$, the small angle approximation holds and $\nu_z \approx \nu = 1/\lambda$. The maximum spatial frequency in x , ν_{o_c} , intercepted by the objective lens is determined by

$$\frac{\nu_{o_c}}{\nu} \approx \frac{\nu_{o_c}}{\nu_z} = \frac{D_o/2}{d} \quad \text{or} \quad \nu_{o_c} = \frac{D_o}{2\lambda d} \quad (2.3-1)$$

The sharp cutoff of the diffracted beams in the spatial frequency domain gives rise to the coherent transfer function, or CTF. Figure 2.3-2 shows the one dimensional CTF for the lens in figure 2.3-1. The CTF is a rectangle function of the form

$$H(\nu_x) = \text{rect} \left[\frac{\lambda d}{D_o} \nu_x \right] \quad (2.3-2)$$

The negative frequencies refer to the negatively travelling beams in x . The two dimensional CTF for a circular lens, shown in figure 2.3-3, is given by

$$H(\nu_x, \nu_y) = \text{circ} \left[\frac{2\lambda d}{D_o} \rho \right] \quad (2.3-3)$$

where $\rho^2 = \nu_x^2 + \nu_y^2$

If the mask is large, the analysis only holds for patterns near the center of the mask. Patterns near the edge of the mask would see a CTF shifted in ν_x .

Figure 2.3-1 depicts *on-axis* ($\nu_{s_y} = \nu_{s_x} = 0$) coherent illumination of the mask. On-axis coherent illumination results in both a positive and a negative diffracted beam, $\pm \nu_{m_i}$, in x for each spatial frequency ν_{m_i} of the mask; and the one dimensional CTF is symmetric around $\nu_x = 0$. *Off-axis* coherent illumination occurs when the illuminating beam has a non-zero spatial component in x ($\nu_{s_y} = 0, \nu_{s_x} \neq 0$); and the CTF for the mask frequencies is shifted in x by ν_{s_x} . The shifted CTF corresponds to the angularly shifted diffracted beams of the mask.

Although the objective aperture is two dimensional, a one dimensional CTF for on-y-axis coherent illumination ($\nu_{s_y} = 0$) is accurate. Off-y-axis coherent illumination ($\nu_{s_y} \neq 0$), however, shifts the CTF in the spatial frequency domain a distance ν_{s_y} . Off-y-axis coherent illumination necessitates the use of the two dimensional CTF, since the the CTF in the x direction may be different for $\nu_{s_y} \neq 0$ than for $\nu_{s_y} = 0$, as with a circular objective.

2.3.2 The Effect of Focus Error on the CTF

Defocus effects can be represented by a modification of the in-focus coherent transfer function (CTF). The condition for focus is determined by the lens law

$$\frac{1}{d_i} + \frac{1}{d_o} = \frac{1}{f_o} \quad (2.3-4)$$

where d_i is the lens to in-focus image distance, d_o is the lens to object distance, and f_o is the focal length of the objective lens.

When the image plane is displaced from the plane of perfect focus, the more general relation (ref. 1, p. 123)

$$\frac{1}{d_i'} + \frac{1}{d_o} = \frac{1}{f_o} + \epsilon \quad (2.3-5)$$

holds, where d_i' is the lens to image distance.

and $d_{im} = d_{ob} = d$.

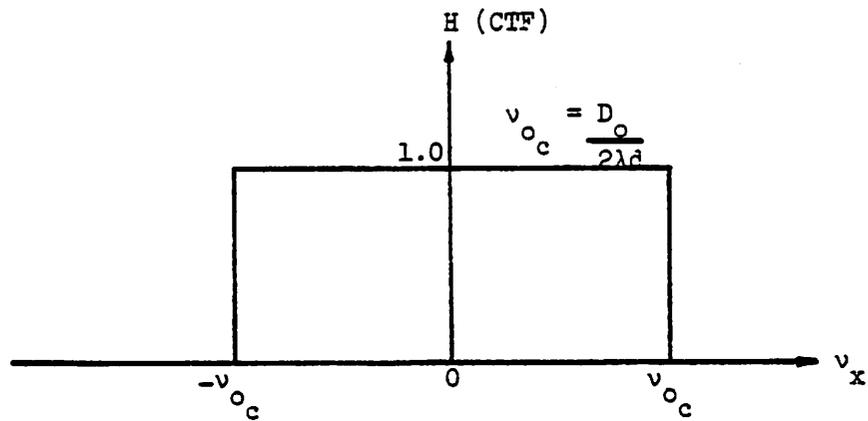


Figure 2.3-2

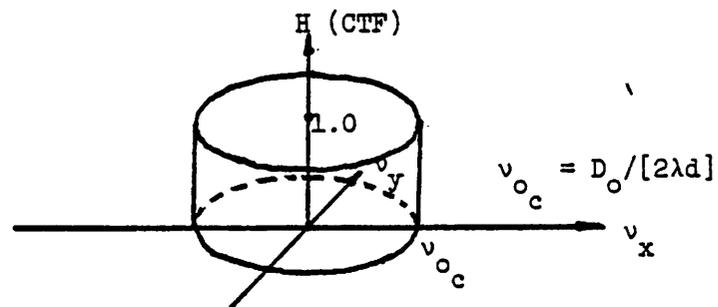


Figure 2.3-3

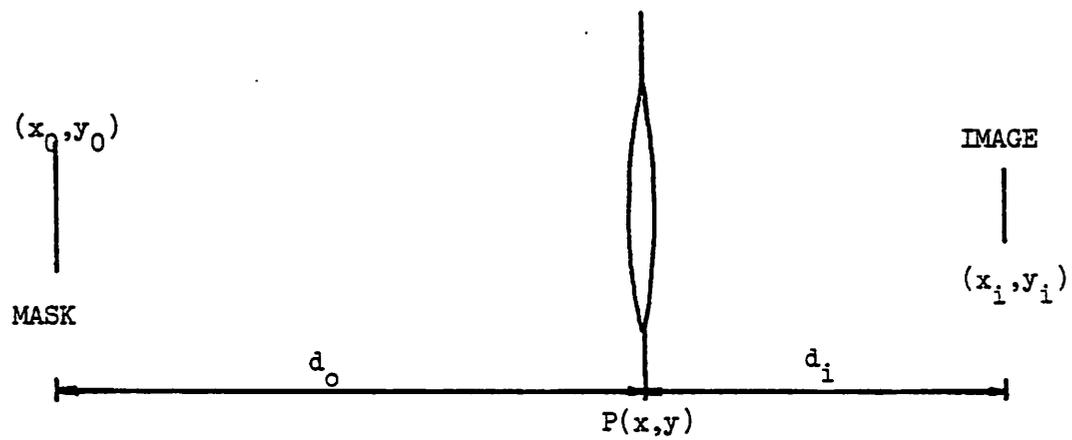


Figure 2.3-4

In linear system theory, a low pass filter gives rise to an impulse response--the response in the image plane to a point in the object plane. The impulse response(ref 1, p 93) of the two dimensional optical system shown in figure 2.3-4 is

$$h(x_i, y_i; x_o, y_o) = \frac{1}{\lambda^2 d_o d_i} \iint P(x, y) e^{i\frac{k}{2} \left[\frac{1}{d_o} + \frac{1}{d_i} - \frac{1}{f_o} \right] (x^2 + y^2)} e^{-ik \left[\left(\frac{x_o}{d_o} + \frac{x_i}{d_i} \right) x + \left(\frac{y_o}{d_o} + \frac{y_i}{d_i} \right) y \right]} dx dy \quad (2.3-6)$$

where the standard phase factor approximation has been assumed. The assumption is valid for any reasonable imaging system--one in which the low frequency components are preserved. Changing variables in equation (2.3-6) several times and taking the Fourier transform, the transfer function of the system becomes

$$F\{\tilde{h}\} = H(\nu_x, \nu_y) = P_g(\lambda d_i \nu_x, \lambda d_i \nu_y) \quad (2.3-7)$$

Equation (2.3-7) shows that, in a diffraction limited imaging system, the impulse response \tilde{h} is the Fourier transform of the exit pupil $P(x, y)$, where $x = \lambda d_i \nu_x$ and $y = \lambda d_i \nu_y$. Thus, a point in the object plane creates a spherical wave at the pupil. The spatial limitation imposed on the spherical wave by the pupil gives rise to an impulse response in the image plane that is not a point.

The effect of phase aberrations of the lens on the impulse response can be included in the analysis by defining a generalized pupil function

$$P_g(x, y) = P(x, y) e^{ikW(x, y)} \quad (2.3-8)$$

Departures from an ideal spherical wave incident on the pupil are represented by a phase addition, given by $W(x, y)$, to the pupil function. An error in focus can be treated as an aberration. A comparison of equations (2.3-5) and (2.3-6) reveals that $W(x, y)$ is given by

$$W(x, y) = \frac{\epsilon(x^2 + y^2)}{2} \quad (2.3-9)$$

For a circular aperture of diameter D_o , the maximum phase error at the aperture edge is

$$w = \frac{\epsilon D_o^2}{8} \quad (2.3-10)$$

The number w is called the Rayleigh distance and is a measure of the defocus severity². The Rayleigh distance is usually specified in wavelengths, such as $\frac{\lambda}{2}$. A Rayleigh distance of $\lambda/4$ corresponds to one incoherent(Rayleigh) depth of field, or

$$D_f = \frac{\lambda}{2(N A_o)_{im}^2} \quad (2.3-11)$$

The focus error, given by ϵ , is incorporated into generalized pupil function P_g such that

$$H(\nu_x, \nu_y) = P_g = P(\lambda d_i \nu_x, \lambda d_i \nu_y) e^{ikW(\lambda d_i \nu_x, \lambda d_i \nu_y)} \quad (2.3-12)$$

where $W(x, y)$ is given in equation (2.3-9). For a circular pupil of diameter D_o , the generalized pupil function under defocus conditions is

$$P_g(x, y) = \text{circ} \left[\frac{\sqrt{x^2 + y^2}}{\frac{D_o}{2}} \right] \cdot e^{ik\frac{\epsilon}{2}(x^2 + y^2)} \quad (2.3-13)$$

The two dimensional coherent transfer function(CTF) for a circular pupil with focus error is then

$$H(\nu_x, \nu_y) = \text{circ} \left[\frac{\sqrt{\nu_x^2 + \nu_y^2}}{\frac{D_o}{2\lambda d_i}} \right] \cdot e^{ik\frac{\epsilon}{2}\lambda^2 d_i^2 (\nu_x^2 + \nu_y^2)} \quad (2.3-14)$$

If the pattern has variations in only the x direction, ν_y is zero and the one dimensional transfer function

$$H(\nu_x) = \text{circ} \left(\frac{2\lambda d_i}{D_o} \nu_x \right) \cdot e^{ik\frac{\epsilon}{2}(\lambda d_i \nu_x)^2} \quad (2.3-15)$$

may be used.

Neither ϵ nor the Rayleigh distance are particularly useful designations of focus error. The distance between the plane of perfect focus and the out-of-focus image plane is more useful. Define δ_{d_i} equal to the distance between the plane of perfect focus and the out-of-focus image plane.

$$\delta_{d_i} \equiv d_i' - d_i \quad (2.3-16)$$

δ_{d_i} is positive when the image plane is further from the lens than the in-focus plane. It is negative when the image plane is closer to the lens than the perfect-focus plane. After manipulating equations (2.3-4) and (2.3-5),

$$d_i' = \frac{d_i}{1 + \epsilon d_i} \quad (2.3-17)$$

Using the definition of δ_{d_i} in equation (2.3-16),

$$\delta_{d_i} = d_i \left[\frac{1}{1 + \epsilon d_i} - 1 \right] = - \frac{d_i^2 \epsilon}{1 + \epsilon d_i} \quad (2.3-18)$$

If δ_{d_i} is expressed in terms of the Rayleigh distance w , then the numerator of equation (2.3-18) becomes

$$\epsilon = \frac{8w}{D_o^2} \rightarrow \epsilon d_i^2 = 2w \left[\frac{2d_i}{D_o} \right]^2 = \frac{2w}{(NA_o)_{im}^2}$$

The denominator of equation (2.3-18) becomes

$$1 + \epsilon d_i = 1 + \frac{8wd_i}{D_o^2} = 1 + \frac{4w}{D_o (NA_o)_{im}}$$

Typically w is on the order of λ , D_o is on the order of $10^5 \lambda$, and $(NA_o)_{im}$ is on the order of .2. Thus $\epsilon d_i \ll 1$ and can be neglected in the denominator of equation (2.3-18). The final expression for δ_{d_i} in terms of the Rayleigh distance is

$$\delta_{d_i} = \frac{-2w}{(NA_o)_{im}^2} \quad (2.3-19)$$

$$\text{where } (NA_o)_{im} = \frac{(NA_o)_{\infty}}{M+1} \quad (2.3-20)$$

and where M is the magnification of the system. Combining equations (2.3-10) and (2.3-19) and substituting for ϵ in equation (2.3-15) gives the one dimensional coherent transfer function for focus error.

$$H(\nu_x) = \text{circ} \left(\frac{2\lambda d_i}{D_o} \nu_x \right) e^{-i\pi\lambda\delta_{d_i}\nu_x^2} \quad (2.3-21)$$

Note that the cutoff frequency ν_{o_c} is given by

$$\begin{aligned} \nu_{o_c} &= \frac{D_o}{2\lambda d_i} = \frac{(NA_o)_{im}}{\lambda} \\ &= \frac{(NA_o)_{\infty}}{2\lambda} \quad \text{for } M=1 \end{aligned} \quad (2.3-22)$$

2.3.3 The Effect of Focus Error in the Space Domain

Equation (2.3-21) gives the CTF under the condition of focus error. A spatial frequency domain phase and weight (1 or 0) is added to the amplitude frequency components of the mask by the CTF. For convenience, define

$$a \equiv \frac{\lambda \delta_{d_f}}{2} \quad (2.3-23)$$

Then equation (2.3-21) becomes

$$H(\nu_x) = \text{circ}\left(\frac{2\lambda d_f}{D_o} \nu_x\right) \cdot e^{-i2\pi a \nu_x^2} \quad (2.3-24)$$

Define the Fourier transform of $g(x)$ as

$$F\{g(x)\} = G(\nu) = \int_{-\infty}^{\infty} g(x) e^{+i2\pi \nu x} dx \quad (2.3-25)$$

Then

$$\begin{aligned} F\{g(x-a\nu)\} &= \int_{-\infty}^{\infty} g(x-a\nu) e^{+i2\pi \nu x} dx \\ &= \int_{-\infty}^{\infty} g(x-a\nu) e^{+i2\pi \nu (x-a\nu)} e^{+i2\pi a \nu^2} d(x-a\nu) \end{aligned}$$

and letting $x' = x-a\nu$,

$$= e^{+i2\pi a \nu^2} F\{g(x)\} = e^{+i2\pi a \nu^2} G(\nu) \quad (2.3-26)$$

Thus

$$F\{g(x-a\nu)\} = e^{+i2\pi a \nu^2} G(\nu) \quad (2.3-27)$$

Note that, for the development above, the frequency ν is a single positive or a single negative frequency. Expressions such as $\cos(2\pi \nu x)$ must be broken into exponentials of frequencies $\pm \nu$. Equation (2.3-27) relates the effect of focus error in the spatial frequency domain to the effect in the space domain.

2.3.4 Construction of the Image

In order to arrive at the image intensity under coherent illumination, the spatial frequencies passed by the lens are added and then squared. For example, if the spatial frequencies 0 , ν_1 , and ν_2 are passed by the lens, and if the Fourier series weightings for these frequencies are a_0 , a_1 , a_2 , then the intensity $I(x)$ in the image plane is given by

$$I(x) = E^2(x) = (a_0 + a_1 \cos 2\pi \nu_1 x + a_2 \cos 2\pi \nu_2 x)^2 \quad (2.3-28)$$

Thus the determination of the image, given on-axis coherent illumination, is quite straightforward using linear systems concepts.

2.4 Imaging with Incoherent Light

Fortunately, incoherent illumination admits to an equally simple linear systems formulation, using an optical transfer function--or OTF. For incoherent illumination, the mask spatial frequencies refer to Fourier components of the intensity of the object mask. Unlike the spatial frequencies in the coherent formulation, the intensity spatial frequencies have no physical

significance. Rather, the intensity spatial frequencies are a result of a mathematical formulation for the special case of incoherent light. The MTF is defined as the absolute value of the OTF and is the same as the OTF when no aberrations are present. When defocus conditions prevail, the OTF may become negative.

2.4.1 The Optical Transfer Function (OTF)

The OTF is the normalized autocorrelation of the CTF (ref 1, p 114).

$$H_{otf}(\nu_x, \nu_y) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\xi, \eta) H^*(\xi - \nu_x, \eta - \nu_y) d\xi d\eta}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |H(\xi, \eta)|^2 d\xi d\eta} \quad (2.4-1)$$

For a two dimensional circular pupil, the OTF can be found in closed form to be

$$H_{otf}(\rho) = \frac{2}{\pi} \cos^{-1} \frac{\rho}{2\rho_0} - \frac{\rho}{\pi\rho_0} \sqrt{1 - \frac{\rho^2}{4\rho_0^2}} \quad \rho \leq 2\rho_0$$

$$= 0 \quad \text{otherwise} \quad (2.4-2)$$

$$\text{where } \rho_0 = \frac{D_o}{2\lambda d_i} = \nu_{oc} \text{ and } \rho = \sqrt{\nu_x^2 + \nu_y^2}$$

The OTF is shown in figure 2.4-1. Since the mask is one dimensional, ρ can be replaced with ν_x . Note that the two dimensional OTF for a circular lens and an infinite source has been used. If the lens were square, the OTF would have the form of figure 2.4-2.

$$H_{otf}(\nu_x, \nu_y) = (D_o - \lambda d_i |\nu_x|)(D_o - \lambda d_i |\nu_y|) \quad |\nu_{xy}| \leq \frac{l}{\lambda d_i}$$

$$= 0 \quad \text{otherwise} \quad (2.4-3)$$

2.4.2 The Effects of Focus Error on the OTF

For incoherent illumination with defocus, an optical transfer function can be deduced for the OTF of the circular objective of equation (2.4-2). Levi² reduced the integral to the form

$$H_{otf}(\nu_r; d) = \frac{4}{\pi} \int_{\nu_r}^1 \sqrt{1-u^2} \cos[2\pi\nu_r d(u-\nu_r)] du \quad (2.4-4a)$$

$$\text{where } d = \frac{2(NA_o)_{im}^2}{\lambda} \cdot \delta_{d_i} \text{ and } \nu_r = \frac{\lambda}{2(NA_o)_{im}} \quad (2.4-4b)$$

and δ_{d_i} is the distance between the observation plane and the plane of true focus. The parameter ν_r is called the reduced spatial frequency. The OTF may become negative for severe focus error of $w > \frac{\lambda}{2}$.

2.4.3 Construction of the Image

In order to determine the image intensity pattern, the Fourier component weightings a_i are multiplied by the values of the OTF at the frequencies ν_i . The weighted spatial frequencies are added to form the image. For example, if the intensity spatial frequencies 0, ν_1 , and ν_2 are passed by the OTF with weightings b_0, b_1 , and b_2 , and if the Fourier series weightings for these frequencies are a_0, a_1 , and a_2 , then the intensity in the image plane is given by

$$I(x) = a_0 b_0 + a_1 b_1 \cos 2\pi\nu_1 x + a_2 b_2 \cos 2\pi\nu_2 x \quad (2.4-5)$$

Figure 2.4-1

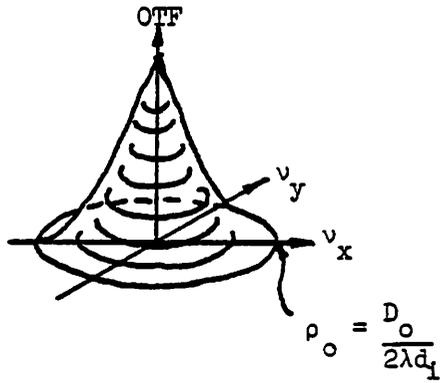
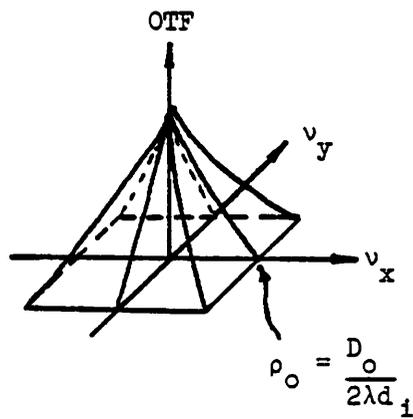


Figure 2.4-2



Thus the determination of the image intensity given incoherent illumination is also quite straightforward using linear systems concepts.

2.5 Imaging with Partially Coherent Light

Although the theory of coherent and incoherent imaging is straightforward, imaging with partially coherent light is difficult and abstract. The original investigations into imaging with partially coherent light was done by Hopkins⁶⁻⁹ in the 1950's; and a number of authors have refined the original formulation¹⁰⁻¹⁵. Although the standard formulation is well known, it lacks intuitive insight. The approach taken here to imaging with partially coherent light emphasizes intuitive insight and is somewhat novel. The intuition is translated into mathematics and then into computer code. The results agree well with the standard formulation.

Partial Coherence

Partial coherence refers to the ability of points on a wavefront to interfere with each other. A wavefront may be *temporally* coherent, *spatially* coherent, or coherent in both the temporal and spatial sense. A wavefront must be temporally and spatially coherent in order to form a sharp fringe pattern. If the wavefront consists of a single color (or narrow spectral range), the wavefront is termed temporally coherent. Only temporally coherent sources will be considered here; and the term *partial coherence* refers to the degree of spatial coherence of the wavefront.

A mercury vapor lamp with an interference filter passes as a temporally coherent source for projection imaging. Each point on the source radiates independently from every other point. Thus an opaque screen with two pinholes P_1 and P_2 placed adjacent to the source produces a pattern on a screen z to the right of the pinholes equal to the sum of the intensities of the pinholes; and no interference pattern results, figure 2.5-1. No phase correlation exists between the fields at separate points on an incoherent wavefront.

Although the fields of the points on an incoherent wavefront are uncorrelated, some correlation between the points develops as the field propagates. A rough intuitive description of how phase correlation is introduced can be seen by the following. Suppose the two pinholes in figure 2.5-1 are placed a distance d to the right of the incoherent source, as in figure 2.5-2. If the pinholes were considered an aperture of diameter s , the smallest resolvable diameter S on the source would be on the order of

$$S = \frac{\lambda d}{s} \quad (2.5-1)$$

by the Rayleigh resolution criterion.

A source much smaller than S behaves like a point source and appears coherent to the pinholes; and the intensity pattern observed on a screen z to the right of the pinholes is the square of the sum of the fields of the pinholes, yielding an interference pattern with nulls of zero intensity. A source several times larger than S produces several interference patterns shifted with respect to one another. The patterns add in intensity on the screen, giving nulls of finite intensity. The field at $x=d$ produced by the incoherent field at $x=0$ is said to be partially coherent. Note that the source size and the distance between the source and the field at $x=d$ is related to the degree of coherence of the field at $x=d$.

In an illumination system, source size also relates to the degree of coherence. The illumination system of a projection printer can be modeled as in figure 2.5-3. The source is separated from the condenser lens by the focal distance of the condenser lens. A particular point on the source produces a plane wave of a particular angle with respect to the z axis to the right of the condenser lens. The endpoints of the source produce the plane waves with the largest angle α_c . Each plane wave causes a fixed phase relation for that wave between the field of the points in the mask plane. The exact phase relation due to a wave depends on the distance between the

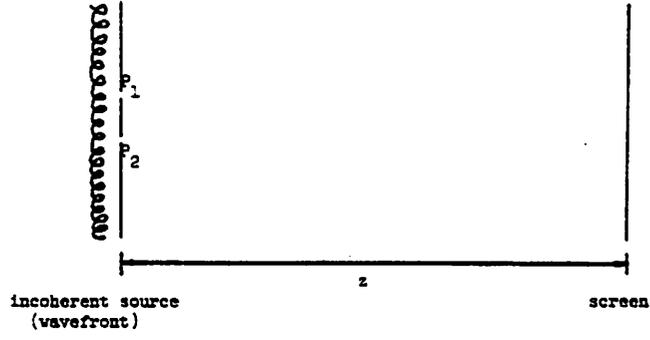


Figure 2.5-1

Figure 2.5-2

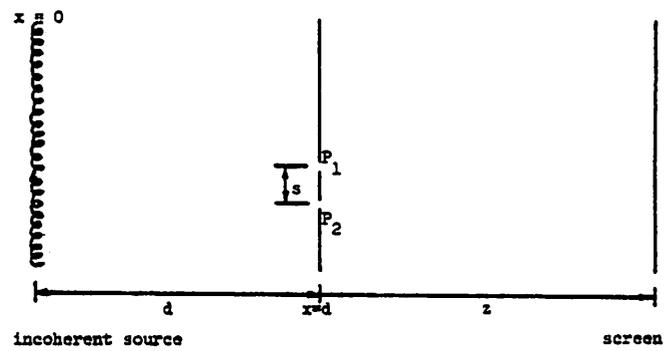
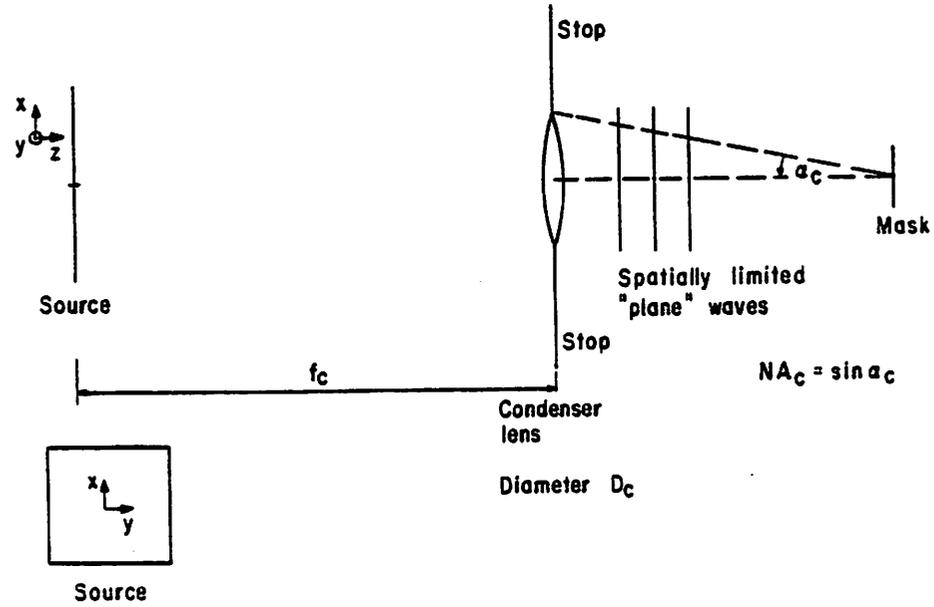


Figure 2.5-3



points and the angle of the wave, figure 2.5-4.

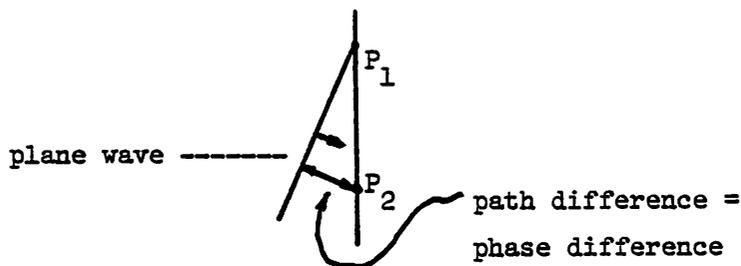


figure 2.5-4

Many plane waves at different angles cause many incoherent sets of phase relations between P_1 and P_2 , washing out the correlation of the fields at P_1 and P_2 . Thus the angular set of plane waves illuminating the mask determines the degree of correlation (or coherence) of the field at the mask.

An intuition for partially coherent imaging can be constructed by considering an object consisting of an amplitude transmission mask. The condenser system illuminates the object with a continuous set of mutually incoherent, uniformly weighted plane waves with a maximum spatial frequency of $\nu_{c_c} = \frac{(NA_c)}{\lambda}$. Each source spatial frequency interacts with the mask independently from the other source spatial frequencies. The one dimensional periodic mask diffracts a set of discrete frequencies in x due to each source point. The objective lens accepts all frequencies below the cutoff frequency $\nu_{o_c} = \frac{(NA_o)_{im}}{\lambda}$, as shown in figure 2.3-1. The beams intercepted by the objective aperture form the intensity distribution in the image plane due to each source wave. The separate intensity distributions due to each source wave are linearly added in order to determine the complete image.

The discussion that follows explains the condenser system in detail. The effect of the size and shape of the source and the shape of the objective lens is discussed with respect to image formation. An intuitive understanding of image formation with partially coherent light is developed.

2.5.1 The Condenser System

The degree of coherence of the light illuminating the mask depends on the configuration of the condenser system. A condenser system where the source is in the focal plane of the condenser lens is called a Köhler system, shown in figure 2.5-3, and is used throughout the analysis. The condenser system can be characterized by the condenser numerical aperture

$$(NA_c)_\infty = \sin \alpha_c \approx \frac{D_c}{2f_c} \quad (2.5-2)$$

where D_c is the diameter of the condenser lens and f_c is the focal length. α_c designates the maximum angle of the angular wedge of plane waves created by the condenser system.

Approximations in the Condenser System Analysis

In order to make the theory of imaging with partially coherent light tractable, two assumptions about the condenser system are usually made. The first assumption is that each source point acts like a coherent source and that adjacent source points are mutually incoherent. It can be shown that such an incoherent wavefront contains all of its energy in spatial frequencies greater than ν_x , where

$$\nu_x = \frac{1}{\lambda} = \nu \quad (2.5-3)$$

Then ν_z is imaginary via the equation $\nu_x^2 + \nu_z^2 = \nu^2$. Thus waves in z are damped exponentials and do not propagate. An exact mathematical proof using rigorous partial coherence theory (complex degree of coherence) can be found in reference 16, p 51. For cases of practical interest, the source may be assumed incoherent point by point.

Once the source is assumed incoherent point by point, the assumption that adjacent points of the source produce mutually incoherent plane waves to the right of the condenser lens is commonly made. The assumption is an approximation, due to the effect of the finite size of the condenser lens. The condenser system in figure 2.5-5 consists of a source point a focal distance away from the condenser lens at $(x,y) = (-a,0)$. If the lens were of infinite diameter, the field to the immediate right of the lens would be

$$E = e^{-i2\pi \frac{a}{\lambda f_c} x} \quad (2.5-4)$$

where the z variation has been omitted. The field E consists of one spatial frequency in x . In the x spatial frequency domain,

$$F\{E\} = \delta(\nu_x - \nu_0) \quad (2.5-5)$$

where $\nu_0 = \frac{a}{\lambda f_c}$. The finite lens size adds a multiplicative factor to the field of equation (2.5-4).

$$E = \text{circ} \left[\frac{\sqrt{x^2 + y^2}}{\frac{D_c}{2}} \right] \cdot e^{-i2\pi \frac{a}{\lambda f_c} x} \quad (2.5-6)$$

The finite lens is considered an infinite lens with a stop of diameter D_c adjacent to it. The Fourier transform of equation (2.5-6) gives the frequency components in x of the field in the aperture.

$$F\{E\} = \frac{D_c}{2} \cdot \frac{J_1 \left[2\pi \frac{\sqrt{\nu_x^2 + \nu_y^2}}{\frac{D_c}{2}} \right]}{\sqrt{\nu_x^2 + \nu_y^2}} * \delta(\nu_x - \nu_0) \quad (2.5-7)$$

where the $*$ means convolution.

Equation (2.5-7) is graphed in the frequency domain in figure 2.5-6. The frequency half-width is $\delta\nu_x = \frac{1.22}{D_c}$. A larger lens diameter gives a narrower half-width. The frequency graph is shifted by the carrier frequency $\nu_0 = \frac{a}{\lambda f_c}$. If $\nu_0 \gg \delta\nu_x$, the angular spread of the set of plane waves can approximately be considered zero, so that the set can be considered a single plane wave of x spatial frequency $\nu_0 = \frac{a}{\lambda f_c}$. The condition corresponds to

$$a \gg \frac{1.22\lambda f_c}{D_c} \quad (2.5-8)$$

Note that the right side of equation (2.5-8) is the Rayleigh criterion for the minimum distance δ that two points can be separated and still be resolved using an aperture of diameter D_c at a distance f_c . Plane waves generated by source points much closer together than the Rayleigh criterion are said to be highly correlated (somewhat coherent). A large condenser diameter D_c insures a small angular spread in the plane waves produced by a particular source point. Thus the assumption that angularly adjacent illuminating waves are uncorrelated becomes reasonable

See Appendix B, Resolution and The Uncertainty Principle.

Figure 2.5-5

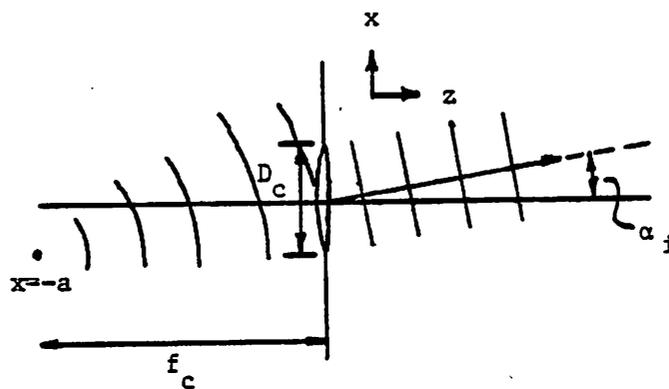
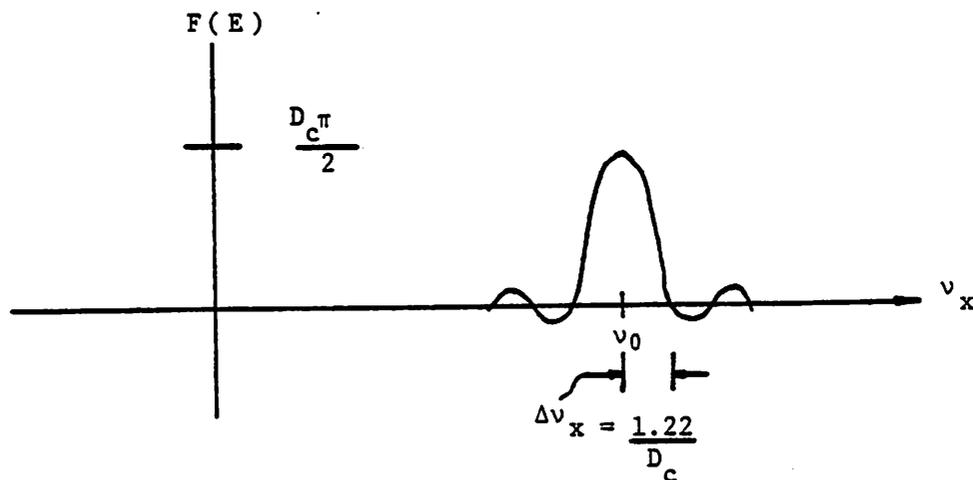
condenser lens
and stop

Figure 2.5-6



for large D_c .

Another physical interpretation of the preceding is worth discussing. The condenser lens can only resolve a certain area size on the source. By the Rayleigh criterion, the radius of the area size is $1.22 \frac{\lambda f_c}{D_c}$. The area that the condenser lens can resolve is called a *coherence area*.

From a point to the right of the condenser lens, a photon emitted by the central point of the coherence area has a probability of being emitted from one of the other points in the coherence area. The field spatial frequencies resulting from many photons being emitted from a coherence area whose central point creates the plane wave of frequency ν_x looks like the graph of $F\{E(x)\}$ in figure 2.5-6. If $\Delta\nu_x \ll \nu_0$, the coherence area on the source can be said to produce a plane wave of frequency ν_0 .

In summary, the source is assumed incoherent point by point; and adjacent source points produce mutually incoherent plane waves to the right of the condenser lens. The assumptions are the usual approximations made in partial coherence theory.

Shape of the Source

Although the mask is one dimensional, both the source and the objective lens are two dimensional. A one dimensional analysis of image formation requires careful attention to the assumptions involved, since the size and shape of the source and of the objective aperture affect the one dimensional pattern of the image. For simplicity, a uniform intensity source is used in the code, although the computer algorithm can be easily modified to handle a non-uniform source of any shape. With a more complicated modification (discussed in section 2.5-2), the algorithm can handle an arbitrarily shaped objective lens. The following discussion centers on the source, and a square objective lens is assumed.

For a particular condenser lens, the two dimensional size and shape of the source determine the angular set of plane waves illuminating the mask. For purposes of discussion, consider the two dimensional source of figure 2.5-7. The source is infinite in extent along y and limited in spatial extent along x , where $(x,y) = (0,0)$ is defined as the center of the source. For purposes of calculation, the source is divided into bars lying parallel to the x axis. The source bars may have varying widths, Δx_i , in the x dimension. Associated with each bar is a set of source positions in y . The source point (x_0, y_0) gives rise to a plane wave of the form

$$e^{-i2\pi(\nu_{s_x}x + \nu_{s_y}y)} \quad \text{where} \quad \nu_{s_x} = \frac{-x_0}{\lambda f_c}, \quad \nu_{s_y} = \frac{-y_0}{\lambda f_c}$$

The plane wave is described by the source spatial frequencies in x and y of ν_{s_x} and ν_{s_y} . A large source frequency indicates that the source wave illuminates the mask at a steep angle with respect to the z axis. The largest spatial frequency in x emitted by the condenser system is created by the x boundary point of the source and given by

$$\nu_{c_x} = \frac{(NA_c)}{\lambda} \quad (2.5-9)$$

The mask contains of a discrete set of spatial frequencies ν_{m_i} in x . A source wave illuminates the mask, giving rise to a discrete set of diffracted beams. The source frequency ν_{s_x} acts as a spatial carrier in x for the mask frequencies ν_{m_i} . Since the mask is periodic in x and has no spatial variation in y , the spatial frequencies ν_x and ν_y diffracted by the mask due to a source frequency of (ν_{s_x}, ν_{s_y}) are of the form

$$\nu_x = \nu_{s_x} + \nu_{m_i} \quad (2.5-10a)$$

and

Judicious choice of the widths results in computational efficiency, as will be seen in Chapter 3.

Figure 2.5-7

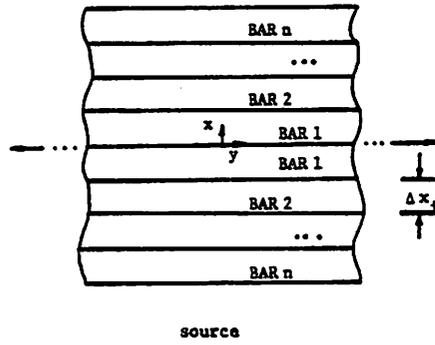


Figure 2.5-8

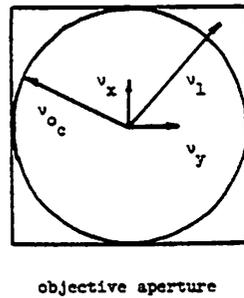
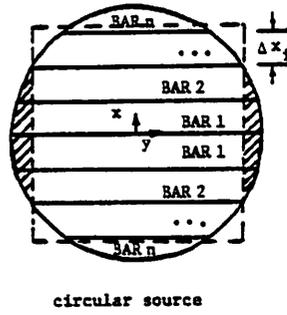


Figure 2.5-9

$$\nu_y = \nu_{s_y} \quad (2.5-10b)$$

As with the source frequencies, the spatial frequency of a diffracted beam represents its direction of propagation with respect to the z axis. A large spatial frequency in x or y propagates at a steep angle with respect to the z axis. A small frequency propagates at a small angle to the z axis. The spatial frequency in x of the diffracted beams are not affected by the source wave propagation in y , as long as ν_y is less than $\sqrt{\nu^2 - \nu_x^2}$ ^{**}. Larger values of ν_y result in a damped exponential field in x and z .

If the objective lens is square, the set of x spatial frequencies intercepted by the objective is independent of the y position of the source point. The set of spatial frequencies in x used to form the electric field of the image is the same for all y source points along the bar at $x=x_0$. Beams with source spatial frequencies in y that are greater than the y cutoff of the objective, however, are not intercepted by the objective and will not contribute to the image. Therefore, the image due to the i th bar need only be calculated for the source point at $y=0$. A source of non-uniform intensity can easily be modeled by assigning to each source bar a weighting dependent on the integrated intensity in y along the bar. In the current analysis and algorithm, the source is considered a line source of uniform intensity in x . For a square objective aperture, a line source is equivalent to a square source.

Although at present two dimensional sources of arbitrary shape are not used in the computer algorithm, the situation can be handled by weighting the source bars by the relative areas contained within them. Source points that generate source waves in y outside of the acceptance angle in y of the square objective lens would not be included in the weighting. In other words, the area of a source bar that is outside of the objective ($\sigma_y > 1$)^{*} is not included in the weighting of the bar, and the bar would be weighted as "1". Figure 2.5-8 illustrates the approach for a circular source and a square objective lens. The dashed square superimposed on the source represents the points on the source that generate source waves accepted by the objective in the absence of a mask. The shaded areas in figure 2.5-8 indicate the areas of the source that have no effect on the image. Bars 1 and 2 would have relative weighting of "1", assuming that the source is uniform in intensity. Again, a source of non-uniform intensity is handled by additionally weighting each source bar according to its integrated intensity.

2.5.2 The Imaging System

In the last section it was pointed out that, if the objective aperture is square, a two dimensional source can be reduced to a line source in x . In general, however, the objective lens is not square and the size and shape of the objective can affect the image significantly. The size of the objective in any particular dimension affects the largest spatial frequency accepted in that dimension, as per equation (2.3-1). The inclusion of higher spatial frequencies results in greater image detail.

Although less important than size, the shape of the objective lens also affects the image of the mask pattern. Figure 2.5-9 shows a square objective and a circular objective; the circular objective has a diameter equal to the side of the square. A diffracted spatial frequency in x of ν_{d_x} is intercepted by the square pupil as long as ν_{s_y} is not greater than ν_{o_c} , where ν_{o_c} is the cutoff spatial frequency in the x and y direction for the square pupil. Whereas, a diffracted spatial frequency in x of ν_{d_x} is intercepted by the circular pupil when the source spatial frequency in y of ν_{s_y} is less than

$$\sqrt{\nu_{o_c}^2 - \nu_{d_x}^2}$$

^{**}The condition for propagation is not restrictive in practice, since the numerical aperture of the objective is smaller than .5.

^{*}The partial coherence parameter σ is equal to the {NA of the condenser system}/{NA of the objective lens in image space}.

where ν_c^2 is the cutoff spatial frequency of the objective lens in all directions. As before,

$$\nu_{d_x} = \nu_{m_i} + \nu_{s_x}$$

The peripheral areas of a source bar that cause illumination beams of high spatial frequencies in y may cause diffracted beams due to high mask frequencies in x to be missed by the circular objective lens, even though the same mask spatial frequencies may be passed by the square objective lens. The spatial frequency ν_1 of figure 2.5-9 is an example of a spatial frequency of a diffracted beam that is intercepted by the square pupil and not by the circular pupil. For a circular objective, *each* spatial frequency of the mask must be weighted according to the x source position of the illuminating wave. As an approximation, the mask frequency can be weighted for an entire source bar by using the average y position of the source bar.

2.5.3 Construction of the Image

As discussed in previous sections, the mask diffracts an incoming plane wave into a set of spatially limited "plane waves", which are either inside or outside of the acceptance angle of the objective. There are two diffracted beams of spatial frequencies $\pm\nu_{m_i}$ for each mask frequency ν_{m_i} . When combined in the image plane, the two beams $\pm\nu_{m_i}$ form a sinusoidal intensity pattern in x of frequency ν_{m_i} . If the illuminating plane wave is at an angle with the mask, the two diffracted beams caused by the mask frequency ν_{m_i} also create an electric field interference pattern in x in the image plane of frequency ν_{m_i} . However, the angle of the intensity peaks with the z axis is not zero. At least two diffracted beams need to be inside the acceptance angle of the objective in order to create a spatial variation in the image plane. The spatial variation of the image intensity is a standing wave in x and z . If both $\pm\nu_{m_i}$ are outside the acceptance angle, no light reaches the image. If one is inside and the other outside, the image consists of a travelling wave in x , a dc intensity pattern.

Through the investigation of some simple cases, an intuition for the construction of the image due to one source point can be obtained. In this section, several cases will be considered. The following analysis uses the $e^{-i2\pi\nu_x x + i\omega t}$ convention for waves travelling in the positive x direction.

Case I

Case I assumes that a source wave of zero spatial frequency (dc) in x and of unit intensity impinges on a mask consisting of a single spatial frequency ν_m , where the subscript i has been dropped for convenience. A unit intensity weighting of the source wave gives an electric field weighting of $\sqrt{2}$. Figure 2.5-10 shows the physical situation. The spatial frequencies of the two diffracted beams in image space are $\pm\nu_m$. The two beams each have a focus error described in the spatial frequency domain by the phase term $e^{+i2\pi a(\pm\nu_m)^2}$. For perfect focus, $a=0$ and the phase term equals 1. As in equation (2.3-23), the variable a contains the defocus information.

$$a \equiv \frac{\lambda \delta_{d_i}}{2}$$

In the spatial frequency domain, a dc source wave is represented by $\delta(\nu)$. The diffracted field is determined by multiplying the source field E_s by the mask's field transmission t_m , where

$$E_s = \sqrt{2} e^{-i2\pi\nu_s x}, \quad \nu_s = 0 \quad (2.5-11a)$$

$$t_m = \cos(2\pi\nu_m x) = \frac{1}{2} e^{-i2\pi\nu_m x} + \frac{1}{2} e^{+i2\pi\nu_m x} \quad (2.5-11b)$$

Note that t_m implies an infinite mask in x and y with spatial variation in the x direction. Equations (2.5-12) give the spatial frequency domain equivalent of equations (2.5-11).

¹² Kinzly's results for a line edge shows similar results for square and circular objective apertures.

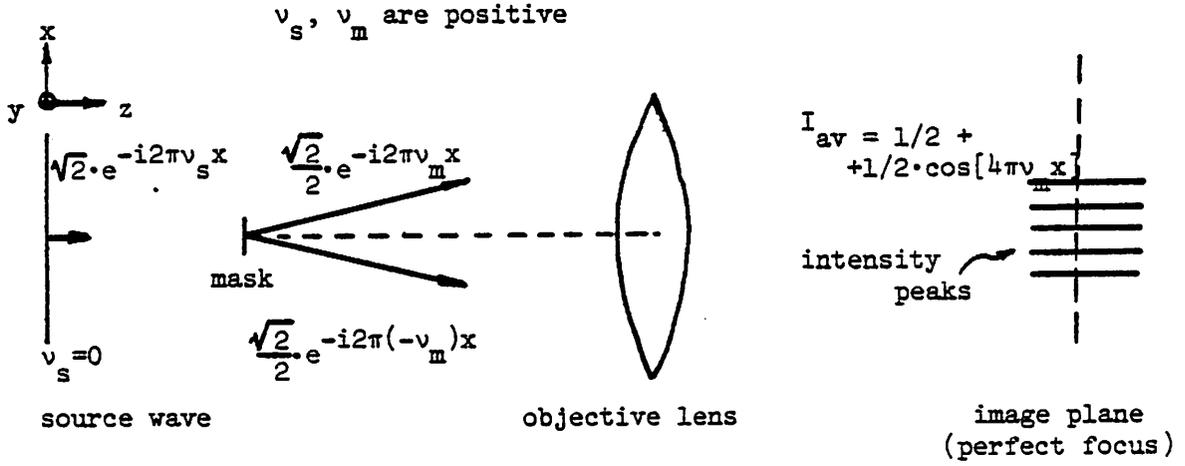
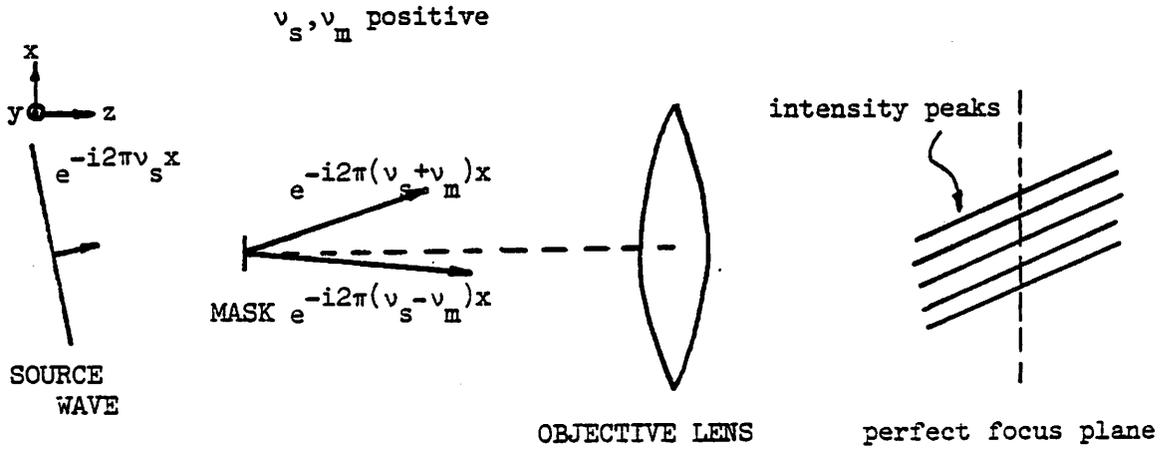


Figure 2.5-10

Figure 2.5-11



$$F\{E_s\} = \sqrt{2} \delta(\nu) \quad (2.5-12a)$$

$$F\{t_m\} = \frac{1}{2} \delta(\nu - \nu_m) + \frac{1}{2} \delta(\nu + \nu_m) \quad (2.5-12b)$$

Since multiplication in the space domain is convolution in the frequency domain, convolving equation (2.5-12a) with equation (2.5-12b) gives the diffracted field in the spatial frequency domain. Equations (2.5-13) give the frequency domain and the space domain (inside ()) expressions for the two diffracted beams. Using the definition for the Fourier transform given in equation (2.3-25),

$$F\{e^{-i2\pi\nu_m x}\} = \frac{\sqrt{2}}{2} \delta(\nu - \nu_m) \quad (2.5-13a)$$

$$F\{e^{+i2\pi\nu_m x}\} = \frac{\sqrt{2}}{2} \delta(\nu + \nu_m) \quad (2.5-13b)$$

The diffracted beams are then intercepted by an infinite lens, which adds the defocus factor $e^{+i2\pi a \nu^2}$ to the frequency domain expression. The final space domain expressions for the two beams is deduced using equation (2.3-27).

$$F\left\{\frac{\sqrt{2}}{2} e^{-i2\pi\nu_m(x - a\nu_m)}\right\} = \frac{\sqrt{2}}{2} e^{+i2\pi a \nu_m^2} \delta(\nu - \nu_m) \quad (2.5-14a)$$

and

$$F\left\{\frac{\sqrt{2}}{2} e^{-i2\pi(-\nu_m)(x - a(-\nu_m))}\right\} = \frac{\sqrt{2}}{2} e^{+i2\pi a (-\nu_m)^2} \delta(\nu + \nu_m) \quad (14b)$$

Thus the image plane electric field E_i is given by

$$\begin{aligned} E_i &= \frac{\sqrt{2}}{2} e^{-i2\pi\nu_m(x - a\nu_m)} + \frac{\sqrt{2}}{2} e^{+i2\pi\nu_m(x + a\nu_m)} \\ &= \sqrt{2} e^{+i2\pi\nu_m a^2} \cos 2\pi\nu_m x \end{aligned} \quad (2.5-15)$$

The average intensity is given by $I_{av} = \frac{1}{2} \text{Real}(E_i \cdot E_i^*)$.

$$I_{av} = \cos^2(i2\pi\nu_m x) = \frac{1}{2} + \frac{1}{2} \cos 2\pi(2\nu_m)x \quad (2.5-16)$$

For Case I, focus error makes no difference in the image intensity pattern because the intensity pattern doesn't vary in z for beams that are equal and opposite in angle around the z axis.

The assumption of an infinite mask and lens results in infinite plane waves throughout the discussion. Practically, small masks can be considered to diffract spatially limited "plane waves", or *beams*, which are then either entirely intercepted or entirely missed by a spatially limited lens. The intercepted beams are again considered "plane waves" for the purpose of forming the image in the image plane. For large masks the analysis is accurate for the beams (spatial frequencies) diffracted from the center of the mask. The spatial frequencies of the extreme portions of a large mask see a shifted CTF.

Case II

Case II considers the same mask spatial frequency as Case I, ν_m , and looks at the two intersecting plane waves generated by a non-dc source wave of spatial frequency ν_s . Since the current discussion's purpose is to build intuition, the constant multipliers related to the intensity normalization of Case I are dropped for convenience. Figure 2.5-11 shows the physical situation for Case II. As before, ν_s and ν_m are positive quantities. The procedure outlined in Case I is followed to give the spatial frequency domain expressions for the two image beams in equations (2.5-17).

$$F\{e^{-i2\pi(\nu_s - \nu_m)(x - (\nu_s - \nu_m)a)}\} = e^{+i2\pi a(\nu_s - \nu_m)^2} \delta(\nu - (\nu_s - \nu_m))$$

and

$$(2.5-17)$$

$$F\{e^{-i2\pi(\nu_s+\nu_m)(x-a(\nu_s+\nu_m))}\} = e^{+i2\pi a(\nu_s+\nu_m)^2} \delta(\nu - (\nu_s+\nu_m))$$

As before, both ν_s and ν_m are positive quantities.

$$\begin{aligned} E_i &= e^{+i2\pi(\nu_m-\nu_s)^2 a} e^{+i2\pi(\nu_m-\nu_s)x} + e^{+i2\pi(\nu_m+\nu_s)^2 a} e^{-i2\pi(\nu_m+\nu_s)x} \\ &= e^{+i2\pi(\nu_s^2+\nu_m^2)a} e^{-i2\pi\nu_s x} \\ &\quad \cdot \left[e^{-i2\pi(2\nu_s\nu_m)a} e^{+i2\pi\nu_m x} + e^{+i2\pi(2\nu_s\nu_m)a} e^{-i2\pi\nu_m x} \right] \\ &= e^{+i2\pi(\nu_s^2+\nu_m^2)a} e^{-i2\pi\nu_s x} \left[\cos\left[2\pi\nu_m(x-2\nu_m\nu_s a)\right] \right] \end{aligned} \quad (2.5-18)$$

The average intensity is

$$I_{av} = \cos^2\left[2\pi\nu_m(x-2\nu_s a)\right] = 1 + \cos\left[2\pi\nu_m(2x-4\nu_s a)\right] \quad (2.5-19)$$

The pattern is cosinusoidal with frequency $2\nu_m$ in x and phase term $-4\nu_m\nu_s a$. The phase term is really the *intensity* spatial frequency in z . Using the definition of a , the phase term becomes $-2\nu_s\nu_m\lambda\delta_{d_i}$, where δ_{d_i} is the displaced distance along z . The intensity spatial frequency for the z direction can be considered to be

$$-2\nu_s\nu_m\lambda \quad (2.5-20)$$

and describes the phase shift in z encountered by going δz to the right of the slanted intensity pattern created by the two intersecting plane waves.

Two Interfering Plane Waves

The fact that equation (2.5-20) represents the appropriate z phase shift can be verified by considering the equation for the average intensity of two unit-field interfering plane waves¹⁷, figure 2.5-12. The total field amplitude E is

$$E = e^{-i2\pi(\eta_s x + \zeta_s z)} + e^{-i2\pi(\eta_r x + \zeta_r z)} \quad (2.5-21a)$$

where

$$\eta_{s,r} = \frac{\sin\Omega_{s,r}}{\lambda} \quad \text{and} \quad \zeta_{s,r} = \frac{\sin\Omega_{s,r}}{\lambda} \quad (2.5-21b)$$

and

$$\psi_i = \frac{\Omega_s + \Omega_r}{2} \quad (2.5-21c)$$

in figure 2.5-12. I_{av} is equal to $\frac{1}{2} \text{Real}(E \cdot E^*)$.

$$I_{av} = 1 + \cos\left[2\pi(\eta_r - \eta_s)x + (\zeta_r - \zeta_s)z\right] \quad (2.5-22)$$

In order to correspond to Case II, let the two plane waves have x spatial frequencies of $\pm\nu_m$, and let them be shifted by a source wave ν_s . Thus

$$\eta_r = \nu_m + \nu_s \quad (2.5-23a)$$

$$\eta_s = -\nu_m + \nu_s \quad (2.5-23b)$$

$$\eta_r - \eta_s = 2\nu_m \quad (2.5-23c)$$

$2\nu_m$ agrees with the x intensity spatial frequency of equation (2.5-19). By the Helmholtz equation,

$$\zeta_r = \sqrt{\nu^2 - (\nu_m + \nu_s)^2} \quad \text{and} \quad \zeta_s = \sqrt{\nu^2 - (-\nu_m + \nu_s)^2} \quad (2.5-24)$$

Using the binomial expansion and throwing out higher order terms, the intensity spatial

Figure 2.5-12

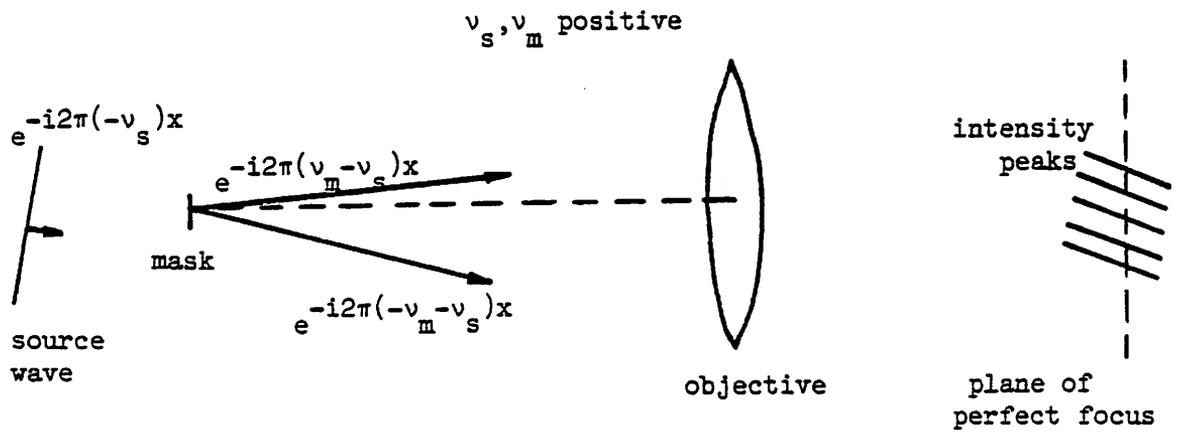
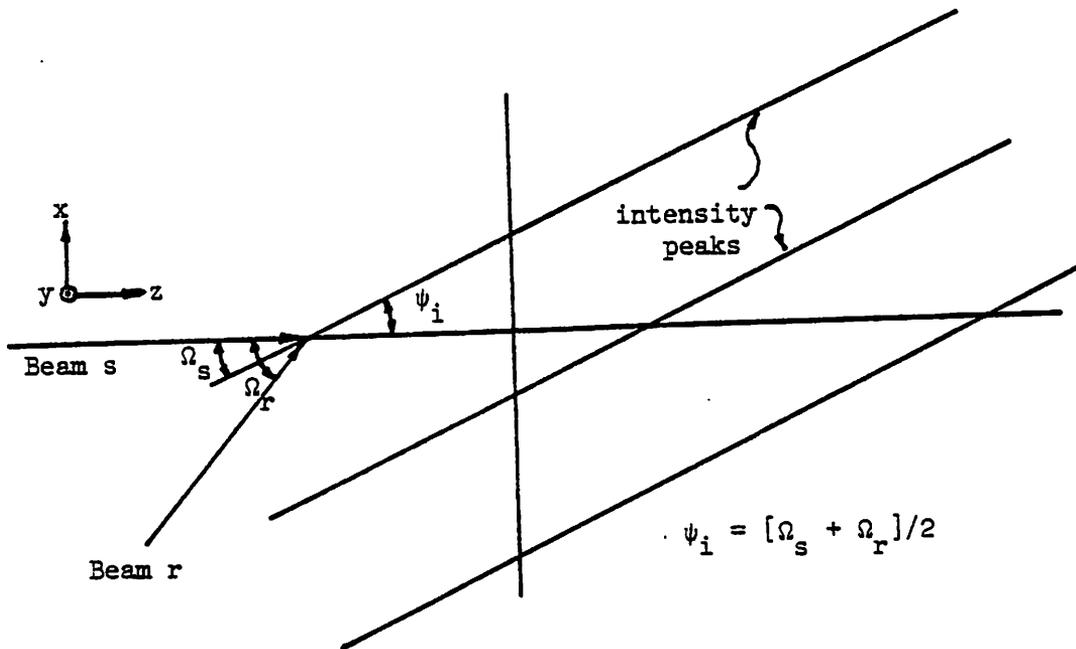


Figure 2.5-13

frequency in z is

$$\eta_r - \eta_s \approx \nu - \frac{(\nu_m + \nu_s)^2}{2\nu} - \left[\nu - \frac{(-\nu_m + \nu_s)^2}{2\nu} \right] = -2 \frac{\nu_m \nu_s}{\nu} \quad (2.5-25)$$

But

$$-2 \frac{\nu_m \nu_s}{\nu} = -2 \nu_m \nu_s \lambda \quad (2.5-26)$$

which is the same as expression (2.5-20).

Case III

Case III is identical to Case II, except that the source wave is of spatial frequency $-\nu_s$. The source wave is coming from the top at the same angle that Case II was coming from the bottom. Figure 2.5-13 shows the physical situation for Case III. The relations between the space domain and the frequency domain of the diffracted beams are

$$F\{e^{+i2\pi(\nu_m + \nu_s)(x+a(\nu_m + \nu_s))}\} = e^{+i2\pi a [-(\nu_m + \nu_s)]^2} \delta(\nu + (\nu_m + \nu_s))$$

and

$$F\{e^{-i2\pi(\nu_m - \nu_s)(x-a(\nu_m - \nu_s))}\} = e^{+i2\pi a (\nu_m - \nu_s)^2} \delta(\nu - (\nu_m - \nu_s)) \quad (2.5-27)$$

The image field E_i becomes

$$\begin{aligned} E_i &= e^{+i2\pi(\nu_m + \nu_s)(x+a(\nu_m + \nu_s))} + e^{-i2\pi(\nu_m - \nu_s)(x-a(\nu_m - \nu_s))} \\ &= e^{+i2\pi(\nu_m + \nu_s)^2 a} e^{+i2\pi(\nu_m + \nu_s)x} + e^{+i2\pi(\nu_m - \nu_s)^2 a} e^{-i2\pi(\nu_m - \nu_s)x} \\ &= e^{+i2\pi(\nu_s^2 + \nu_m^2)a} e^{+i2\pi\nu_s x} \\ &\quad \cdot \left[e^{+i2\pi(2\nu_s \nu_m)a} e^{+i2\pi\nu_m x} + e^{-i2\pi(2\nu_s \nu_m)a} e^{-i2\pi\nu_m x} \right] \\ &= e^{+i2\pi(\nu_s^2 + \nu_m^2)a} e^{+i2\pi\nu_s x} \left[\cos\left[2\pi\nu_m(x+2\nu_m\nu_s a)\right] \right] \end{aligned} \quad (2.5-28)$$

Without the constant multipliers, the average intensity is

$$I_{av} = \cos^2\left\{2\pi\nu_m(x+2\nu_s a)\right\} = 1 + \cos\left\{2\pi\nu_m(2x+4\nu_s a)\right\} \quad (2.5-29)$$

The average intensity of Case III is similar to that of Case II. The phase term has changed sign, denoting that the angle of the image intensity peaks with the z axis for Case III is equal in magnitude and opposite in sign to the angle for Case II. The intensity patterns for Case III and for Case II are identical in the $z=0$ plane, the perfect focus plane.

Note that if the source is symmetric, equations (2.5-19) and (2.5-29) can be added to get a total intensity of

$$\begin{aligned} I_T &= \cos^2(c+d) + \cos^2(c-d) \quad \text{where } c=2\pi\nu_m x, \quad d=2\pi(2\nu_m\nu_s a) \\ &= \frac{1}{2} + \frac{1}{2}\cos(2c+2d) + \frac{1}{2} + \frac{1}{2}\cos(2c-2d) \\ &= 1 + \cos 2c \cdot \cos 2d \end{aligned}$$

Thus the intensity due to two points (coherence areas) on the line source equidistant from the central point and due to one mask spatial frequency $\pm\nu_m$ is

$$I_T = 1 + \cos(2\pi(2\nu_m x)) \cdot \cos(2\pi(4\nu_m\nu_s a)) \quad (2.5-30)$$

Although the above three cases display considerable insight about the formation of the

image intensity pattern, the consideration of two diffracted beams is not sufficient for determining a general image. In general, a source wave results in an image consisting of many diffracted beams of varying spatial frequencies. The source wave of frequency ν_s is assumed to have unit intensity. Each real spatial frequency of the mask ν_m is assumed to have a Fourier component weighting of c_m ; therefore, each frequency $\pm\nu_m$ has weighting $\frac{c_m}{2}$. Figure 2.5-14 shows the $+\nu_m$ diffracted beam. As before, the spatial frequency domain description of the $+\nu_m$ image beam under defocus conditions is

$$\sqrt{2} \cdot \frac{c_m}{2} \cdot e^{+i2\pi a(\nu_m + \nu_s)^2} \delta(\nu - (\nu_m + \nu_s)) \quad (2.5-31)$$

The inverse transform of (2.5-31) is

$$\frac{\sqrt{2}}{2} \cdot c_m e^{-i2\pi(\nu_m + \nu_s)(x - a(\nu_m + \nu_s))} \quad (2.5-32)$$

and gives the space domain representation of the electric field due to mask frequency ν_m and to source frequency ν_s . Expression (2.5-32) is used in the computer algorithm that implements partial coherence. The phase factor $e^{-i2\pi\nu_s x}$ may be deleted when calculating the intensity due to a source wave, since all diffracted beams have this factor in common and it will drop out when calculating $E \cdot E^*$.

2.5.4 Polarization Effects

The mask is considered a thin amplitude mask consisting of lines made out of a perfect conductor and laid on a perfectly transmitting material. The field at the mask due to a source wave of intensity 1 is zero on the conductor and $\sqrt{2}$ between the conductors, as assumed in the analysis. It is assumed that the mask does not change the polarization of the illuminating light. If the incident wave is unpolarized, a typical diffracted beam will have the form

$$\mathbf{E} = \mathbf{E}_0 e^{-i2\pi\nu x} \quad (2.5-33)$$

$$\text{where } \mathbf{E}_0 = E_p \bar{p} + E_l \bar{l} \quad (2.5-34)$$

The equation for the average intensity becomes

$$I_{av} = \frac{1}{2} \text{Real}\{\mathbf{E} \cdot \mathbf{E}^*\} \quad (2.5-35)$$

The \bar{p} polarization (perpendicular to the plane of incidence) forms interference planes independent of the \bar{l} polarization (parallel to the plane of incidence). Each polarization interferes with itself to form an intensity pattern. The intensity patterns of the two polarizations are identical. In other words, there are no polarization effects under the assumptions of the analysis.

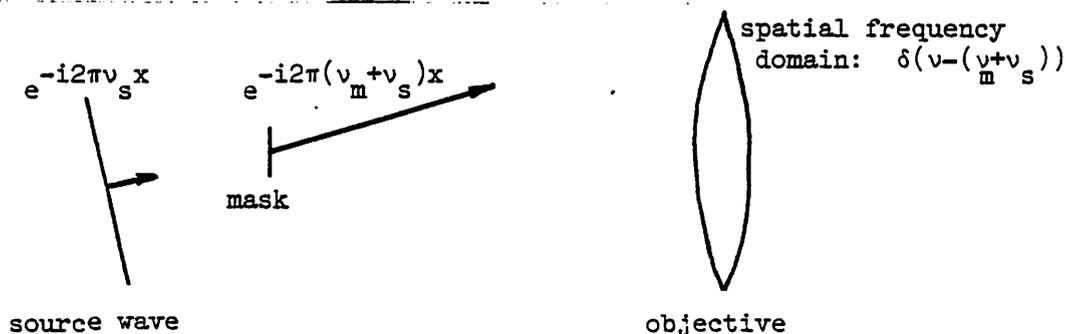


Figure 2.5-14

References for Chapter 2:

- 1) J.W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, 1968.
- 2) L. Levi, *Applied Optics*, Wiley, New York, 1968.
- 3) M. Born and E. Wolf, *Principle of Optics*, 5th edition, Pergamon Press, 1975.
- 4) A. Papoulis, *Systems and Transforms with Applications in Optics*. McGraw-Hill, 1968.
- 5) R. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 1968.
- 6) Hopkins, H.H., "The Concept of Partial Coherence in Optics", *Proc. Roy. Soc. (London)*, A208,1951,p. 263.
- 7) Hopkins, H.H., "On the Diffraction Theory of Optical Images", *Proc. Roy. Soc. (London)*,A217,1953,p. 408.
- 8) Hopkins, H.H., "The Frequency Response of a Defocused Optical System", *Proc. Roy. Soc. (London)*,1956.
- 9) Hopkins, H.H., "Applications of Coherence Theory in Microscopy and Interferometry", *JOSA*, vol 47, no 6, June '57, p 508.
- 10) E. Wolf, "A macroscopic theory of interference and diffraction of light from finite sources I. Fields with a narrow spectral range", *Proc. of the Roy. Soc. of London*, A(225), p 96, 1954.
- 11) E. Wolf, "A macroscopic theory of interference and diffraction of light from finite sources II. Fields with a spectral range of arbitrary width", *Proc. of the Royal Society of London*, A(230), p 247, 1955.
- 12) R.E. Kinzly, "Investigations of the Influence of the Degree of Coherence upon Images of Edge Objects", *JOSA*, vol 55, # 8, Aug '65, p 1002.
- 13) R.E. Kinzly, "Images of Coherently Illuminated Edged Objects Formed by Scanning Optical Systems", *JOSA*, vol 56, no 1, p 9, Jan 1966.
- 14) E.C. Kintner, "Method for the Calculation of Partially Coherent Imagery", *Applied Optics*, vol 17, no 17, pp2747-2753, Sept 1978.
- 15) A. Offner and J. Meiron, "The Performance of Optical Systems with Quasi-Monochromatic Partially Coherent Illumination", *Applied Optics*, vol 8, no 1, p 183-7, Jan 1969.
- 16) M.J. Beran and G.B. Parrent,Jr., *Theory of Partial Coherence*, Prentice Hall, Englewood Cliffs,N.Y., 1960.
- 17) R.J. Collier, C.B. Burckhardt, L.H. Lin, *Optical Holography*, Academic Press, 1971, p 228.

Chapter 3

The Image Machine: Code

Written in standard FORTRAN, the code for the IMAGE machine is portable to most minicomputers. The code is divided into a number of subroutines, which are organized in the hierarchical manner depicted in figure 3-1. At the top of the hierarchy is the sub-controller, subroutine IMAGE. The Controller of the User-Interface calls on IMAGE, which in turn calls on the various routines needed to solve the problem inputted by the user. Information is passed between the Controller and IMAGE through common blocks.

The following text describes the IMAGE routines in detail. It is divided into two sections, Standard Routines and Partial Coherence Routines. With the exception of the partial coherence routines, PARCOH and FASTPC, the code is straightforward. The following text should be read in conjunction with the Common Block Documentation and a listing of the SAMPLE code, Appendices C and D.

3.1 Standard Routines

IMAGE

Subroutine IMAGE is the sub-controller for the IMAGE machine. It begins by echoing the common block variables passed to the IMAGE machine by the Controller. Next it clears various arrays and checks a few variables for the correct range. Loop 10 calls subroutines CLCMTF and PLTPAT, if requested. CLCMTF calculates the horizontal intensity pattern, and PLTPAT outputs the pattern on the line printer. CLCMTF is called for each of the five possible wavelengths. If a OTF lineprinter plot is requested, subroutine IMAGE calls on subroutine PLTOTF. IMAGE then returns control to the Controller.

CLCMTF

Subroutine CLCMTF (*calculate MTF*) calculates the value of the OTF for incoherent light or the CTF for coherent light. For partially coherent light, CLCMTF calls on the partial coherence routines. The passed parameter, ILMBD, is the wavelength for the current image pattern. A diagram of CLCMTF is shown in figure 3.1-1. The projection system section comprises most of the routine. Sharad Nandgaonkar¹ is responsible for the code of the contact printing section.

The projection section begins by calculating the variable VMAX. VMAX contains the cutoff frequency for coherent or incoherent illumination. For partially coherent illumination, VMAX contains the highest spatial frequency intercepted by the objective lens, or

$$\frac{(NA_o)_{im}(1+\sigma)}{\lambda}$$

where σ is the partial coherence parameter.

After VMAX is determined, the routine branches according to the type of mask pattern requested. In each pattern section, the spacing between horizontal points, DELTX, is calculated via

$$\text{DELTX} = \frac{\text{WINDOW}}{\text{FLOAT}(\text{NMHPTS}-1)}$$

The variable WINDOW specifies the width in microns of the lineprinter plot.

¹The Standard Routines have been released to the general public as of April 1979.

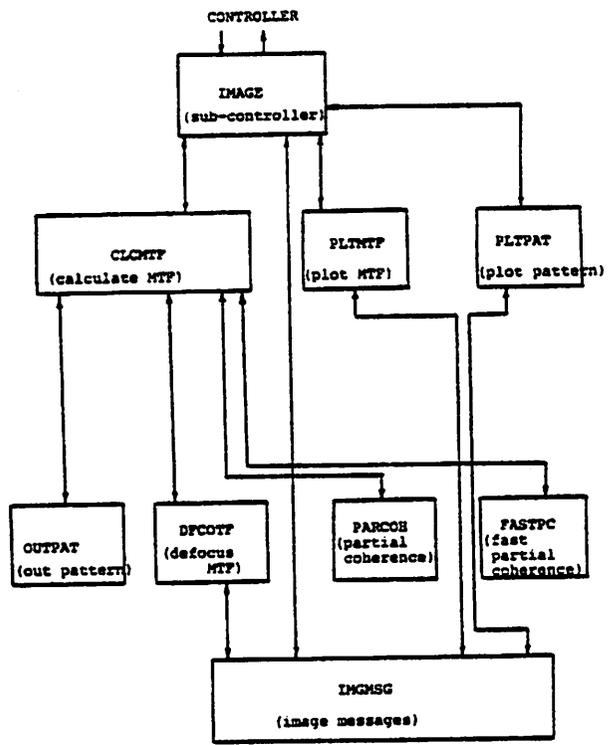
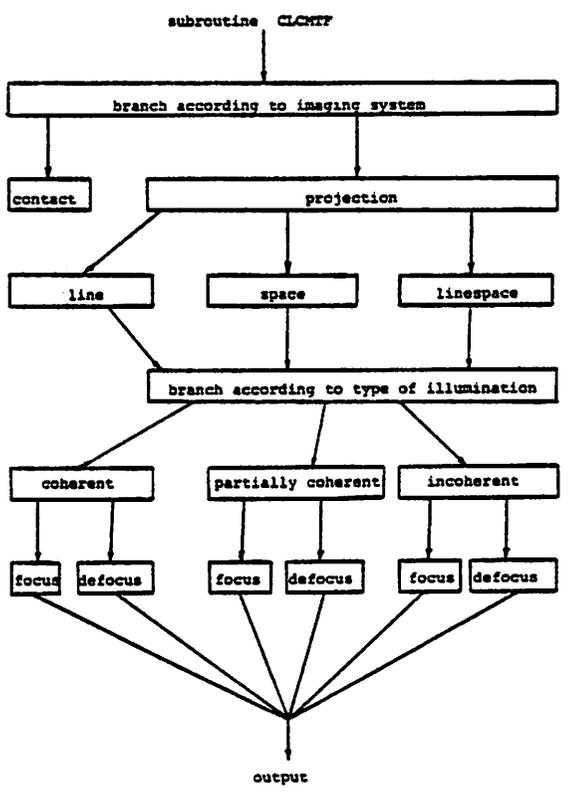


Figure 3-1

Figure 3.1-1



WNDORG (*window origin*) specifies the x value of the left side of the window and is calculated by CLCMTF. The $x=0$ position for the Fourier analysis¹ is in the middle of the line(space) for the line(space) pattern. Next the Fourier component weightings are calculated for the pattern. The maximum number of frequency components, MXNMFR, is used to construct the pattern for a line(or space), in order to place the next nearest line(or space) as far away as possible. A system message prints the distance to the next nearest line(or space). For a linespace pattern, only the needed frequency components are calculated. A system message is printed if the number of frequencies needed exceeds MXNMFR. Since all frequency components of the pattern are either 0 or multiples of the fundamental, only the fundamental frequency of the pattern, V_1 , is calculated and stored. The frequency component weightings are stored in the array FSAMSK (*Fourier series at mask*). FSAMSK(1) contains the dc weighting. FSAMSK(i) contains the weighting of the i th frequency component, $\nu_i = (i-1)*V_1$.

With the frequency weightings of the requested pattern calculated and stored, the routine branches according to the type of illumination coherence. The focus error in microns is indicated by the value of DFDIST; DFDIST=0 indicates perfect focus. Coherent light without defocus is handled by copying the frequency weightings in FSAMSK to the array FSAIMG (*Fourier series at image*). Subroutine OUTPAT (*out-pattern*) is called to construct the image intensity pattern. Coherent light with defocus is handled by subroutine PARCOH. Partially coherent light is handled by subroutine PARCOH (*partial coherence*) for the defocus case and by the routine FASTPC (*fast partial coherence*) for the in-focus case. Both routines construct the image intensity pattern internally; OUTPAT is not called for partially coherent illumination. Incoherent light is handled similarly to coherent light. The image frequency weightings FSAIMG are weighted by the OTF. The OTF is calculated in CLCMTF for the in-focus case and in subroutine DFCOTF for the defocus case. OUTPAT is called in both cases. After calculating the image pattern, CLCMTF returns control to subroutine IMAGE.

DFCOTF

Subroutine DFCOTF (*defocus OTF*) calculates the incoherent OTF for a focus error distance given by DFDIST. The wavelength of the current calculation is passed in the parameter list. DFCOTF is the code for the integral equation given by Levi² and discussed in the theory section. Twenty iterations give about three decimal places of accuracy for all possible input values; the routine's results were checked against Levi's results. The OTF calculated for frequency ν_i is multiplied by FSAMSK(i). The result is stored in FSAIMG(i). Control is then returned to routine CLCMTF.

OUTPAT

Subroutine OUTPAT calculates the image intensity pattern for each of the horizontal points. The current wavelength is passed through the parameter list. The horizontal pattern for the *current* wavelength is stored in the array THORIN. The composite horizontal intensity pattern for of the all wavelengths is stored in array HORINT (*horizontal intensity*). HORINT weights the pattern generated by each wavelength by the relative intensity of that wavelength. The sum of the relative intensities for all the wavelengths is 1. OUTPAT is called by routine CLCMTF.

PLTPAT

Subroutine PLTPAT gives a lineprinter plot of the normalized horizontal image intensity for each of the wavelengths. It also plots the composite(weighted) image intensity for all the wavelengths. Setting the appropriate flag in common block IMGFLG punches cards containing the (x,I) coordinates of the image plot--for ease of transport to a plotter. For the HP plotter routine used at UCB, the first punched card contains four numbers. The second card contains one number, the number of sets of points to come--# of wavelengths + composite(if # > 1). The third card contains one number, the number of (x,I) pairs to come. The following cards contain the (x,I) pairs sequentially, eleven per card. Card three and "point pair" cards follow in

¹The origin($x=0$) for the printout is defined at the mask line edge.

t groups, where t is the number on the second card.

Since the plotting routines in IMAGE and in DVELOP may soon be replaced with routines requiring less memory, only the highlights of the routine will be discussed. The plot information is contained in the array IPLT(123,99), where plot points (x,I) are stored as a holerith in the appropriate cell in the array. Curves for different wavelengths are stored as different letters in the array. On the CDC6400 printers at UC Berkeley, 123 by 99 results in a square plot. IPLT is stored in blank common (along with other arrays) in order to save memory during compilation. The vertical axis pertains to the intensity and the horizontal axis pertains to the x dimension. The extremes of the array IPLT contain the character "*". The center of cell IPLT(2,2) contains the intensity 1.3. The center of IPLT(2,98) contains intensity=0. 0 and 1.3 are printed on the plot to indicate these points. A horizontal line of dashes indicates the intensity of 1.0. Note that the center of the cells just within the *'s contain the boundary of the problem. HORINT(NMHPTS) is contained in the center of IPLT(122,2). NX and NZ contain the indices of IPLT for the horizontal point IHPT being calculated in loop 12. Numbers such as 2.0001 have the .0001 in order to avoid roundoff errors.

PLTOTF

Subroutine PLTOTF plots the OTF for the incoherent defocus case, the incoherent in-focus case, and the coherent in-focus case. A request for an OTF plot for partial coherence results in a system error message. PLTOTF can take significant time because various other subroutines, such as DFCOTF, are called for 31 spatial frequencies. As in subroutine PLTPAT, punched cards may be requested for the optical transfer function(OTF) for both in-focus and defocus cases. The plot is $\frac{1}{4}$ the area of the image plot, although it uses the same blank common array IPLT.

IMGMSG

Subroutine IMGMSG (*image messages*) contains the system message, warning, and error statements. There is an obvious order to the message routines of all machines. The common block variables used by IMAGE are echoed by IMGMSG at the start of each run of machine IMAGE. Most of the messages used by IMAGE are printed out by IMGMSG, with the exception of the partial coherence routines.

3.2 Partial Coherence Routines

The algorithms for partial coherence can accept four different cases: partial coherence without defocus, partial coherence with defocus, complete coherence without defocus, and complete coherence with defocus^(*). Subroutine PARCOH is the code corresponding to the partial coherence formulation discussed in previous pages. However, shortcuts in the algorithm, resulting in substantial savings in run time, can be made under the condition of perfect focus. Subroutine FASTPC takes advantage of these shortcuts. Comparison of the run times for the various routines is discussed.

PARCOH

Subroutine PARCOH calculates the intensity pattern on the resist for partially and fully coherent light under various degrees of defocus. The common blocks CBWIND, FOUUSER, HORIMG, IMG2PR, IMG3PR, IMGFLG, IO1, SPECTR, OPTIC, and PCTERM are needed by PARCOH. The array TERM(82,3) in block PCTERM is shared by FASTPC in order to save memory space. Documentation on all of the blocks' variables is contained in the Common

(*)The case of complete coherence without defocus can be handled in a more efficient manner than in the partial coherence formulation. The capability of complete coherence without defocus is included for the purpose of checking the partial coherence algorithm.

Block Documentation, Appendix C. The statement

COMPLEX ALL(82)

may or may not be included, depending on the particular image intensity calculation used. The number 82 reserves memory for the maximum number of mask spatial frequencies that can be intercepted per source wave by the objective lens: CLCMTF allows 41 frequencies and PARCOH processes the \pm components of the 41 frequencies. In addition to the blocks, PARCOH is passed the parameter ILMBD, which gives the index of the current wavelength RLAMBDA(index). The current wavelength is used to find the coherent cutoff frequency of the objective lens, VCOBJ, in the first executable statement of the routine.

At present, the routine assumes the source of figure 2.5-7(Chapter 2). The routine divides the source into assumed "coherence areas", corresponding to the bars in the figure. Each coherence area is responsible for an intensity pattern on the resist. The summation of the intensity patterns from all of the coherence areas on the source are added to form the total intensity pattern. For perfect focus(DFDIST=0.), symmetric coherence areas can be calculated at the same time, since the intensity patterns produced by coherence areas equidistant from the optic axis of a line source are identical in the plane of focus. Symmetric coherence areas correspond to illuminating plane waves(source waves) of $\pm\nu_s$. The in-focus capability is included in PARCOH for comparison with the faster algorithm in routine FASTPC. The spatial frequencies of the illuminating source beams are separated by DSFRQ(the discrete frequency separation), which is determined by the number of coherence areas in the top half of the source(NCOHAR). NCOHAR is determined by the number of coherence areas in the top half of the source for $\sigma=1$, given by the variable AREAS, times the current σ , given by the variable SIGMA. The added .0001 is to make sure that NCOHAR=AREAS if $\sigma=1$. NCOHAR has a minimum value of 8. SIGMA is the ratio of the numerical aperture of the condenser lens to the numerical aperture of the objective lens. The size of the source, the condenser lens size, and the focal length of the condenser lens determine the number of coherence areas needed. These numbers are not usually readily available or meaningful for many condensers. In any case, the image intensity converges to a pattern after ten coherence areas or so. *The run time of the routine is approximately proportional to the number of coherence areas chosen.* Run time becomes excessive for SIGMA > 3. Presently NCOHAR=15 for SIGMA=1 and can be changed by altering statement 1 in the routine. A comparison study of an efficient number of coherence areas to be taken will be done later. NI sets the number of iterations for loop 75. DSFRQ>0 and NI=2 indicates perfect focus and that only the positive source wave intensity patterns are calculated. DSFRQ=0 and NI=1 indicates the coherent formulation.

Closely related to the number of coherence areas is the normalization constant RNORML. The normalization constant is equal to the fraction of the acceptance angle of the objective lens that the source wave incremental frequency(DSFRQ) occupies. RNORML contains the weighting to be given to each coherence area. When all 2·NCOHAR source coherence area contributions are calculated for the final image(partially coherent light with defocus).

$$RNORML = \frac{DSFRQ}{2 \cdot AMIN1(VCOBJ, VCCOND)}$$

For partial coherence without defocus,

$$RNORML = \frac{DSFRQ}{AMIN1(VCOBJ, VCCOND)}$$

Only half of the source coherence areas need be calculated, since the other symmetric half just adds a factor of two. Note that the normalization factor divides the accumulated intensity pattern by the intensity incident on the resist given a clear mask.

The variable FACTOR is used to save multiplications in the intensity section.

$$FACTOR = .5 \cdot \sqrt{RNORML}$$

^{*} See Cases II and III in section 2.5.3(Chapter 2).

An incoming source wave with field E is defined to have an intensity weighting of 1. Then

$$I_{av} = \frac{1}{2} \text{Real}(E \cdot E^*) = 1$$

and the magnitude of E is equal to $\sqrt{2}$. The $\sqrt{2}$ weighting need not be explicitly written in the equation for FACTOR, since RNORML normalizes the intensity due to DSFRQ by the intensity due to a clear mask. In other words, the $\sqrt{2}$ field weighting (or 1 intensity weighting) cancels due to the normalization of RNORML. The .5 in FACTOR is due to the mask spatial frequency ν_m diffracting two beams of frequencies $\pm \nu_m$. That is,

$$\cos 2\pi \nu_m x = \frac{1}{2} e^{+i2\pi \nu_m x} + \frac{1}{2} e^{-i2\pi \nu_m x}$$

Each diffracted beam has a weight factor of .5 times the mask weighting.

After NCOHAR, RNORML, DSFRQ, FACTOR, and NI are set to the values appropriate to partial or full coherence with or without defocus, the main part of the routine is entered. Statement 25 defines the variable DFOCPH, or "defocus phase". DFOCPH is the variable a ,

$$a = \frac{\lambda \delta_{d_i}}{2}$$

defined in section 2.3.3. Several do-loops are included in the major section. The outer loop, 100, is responsible for iterating the intensity pattern effects due to the source spatial frequency ν_s , FORTRAN variable VS. Loop 100's index ICOHAR refers to a symmetric set of source coherence areas. Note that the first coherence area corresponds to a source wave of spatial frequency $VS = \frac{DSFRQ}{2}$ and not $VS=0$ (dc). Fully coherent light is an exception with $VS=0$.

The next loop, 75, is responsible for iterating the intensity pattern effects due to the symmetric source waves $\pm \nu_s$. The index I equals one for perfect focus and for full coherence. I equals two for partial coherence with defocus, since the effect on the image due to each area of the symmetric pair of source areas is not the same.

Loop 50 stores information about the spatial frequencies of the image intensity pattern due to the current source wave of spatial frequency ν_s . Source frequency $\nu_s = VS$ impinges on the mask frequency $\nu_m = VM$ causing diffracted spatial frequencies of $\nu_s + \nu_m$ and $\nu_s - \nu_m$, which are stored in the variable SHFREQ. If the diffracted beams (frequencies) are within the acceptance angle of the objective lens and if the diffracted beams are bright enough to include in the calculation (Fourier weighting $\geq .0001$), the beam frequency weighting, the mask frequency, and the defocus phase are stored in TERM(KOUNT, 1-2-3 respectively). KOUNT indexes the values due to one diffracted beam and can be incremented twice, corresponding to the $\pm \nu_m$, for each of the IFRCP of loop 50. Note that the Fourier series at the mask (FSAMSK) is multiplied by FACTOR and stored in TERM(KOUNT, 1). Loop 50 is repeated for the number of mask frequency components NMFRCP.

After the frequency information due to a source wave is stored, the intensity pattern for the source wave is calculated. There are several methods of calculation; the fastest of which will be discussed now. The following intensity algorithm relies on the paraxial ray approximation and thus assumes small numerical apertures. Loops 53 and 52 comprise the intensity calculation. The inner loop 52 stores the electric field for the point x in the variable AL. The field consists of all the diffracted beams formed into fields, as in Case IV of section 2.5.2. The normalized intensity pattern for the present source area is calculated via the equation

$$I_{av} = .5 \cdot \text{RNORML} \cdot \text{Real}(EE^*)$$

The intensity pattern is then added to the accumulated intensity pattern due to the previously calculated source areas.

A "high- σ " intensity algorithm, which departs somewhat from the small angle approximation, is also available (but not used in the current version). For $\sigma \geq 2$ the beam angles can become large. The energy flow of the beat patterns are no longer perpendicular to the resist

surface. A weighted average of the two beating beams is then taken. The energy in each beam is assumed to flow independently.

After all source area intensity contributions are calculated, the accumulated intensity is stored in HORINT and the routine returns control to CLCMTF.

Since the optical transfer function(OTF) formulation is used by IMAGE to treat patterns illuminated with incoherent light, a theoretical link is needed between the OTF formulation and the partial coherence formulation. The results of the two formulations are directly comparable if the normalization methods are the same. As discussed previously, the partial coherence formulation normalizes the intensity by dividing by the intensity at the resist surface with no mask present--or a perfectly clear mask present. The OTF is normalized by dividing the frequency weightings by

$$\iint |\bar{h}(x_i, y_i)|^2 dx_i dy_i$$

The quantity represents the dc intensity incident on the resist surface with no mask present³. Thus the normalization for the two formulations is the same.

FASTPC

Subroutine FASTPC calculates the intensity pattern incident on the resist for partially coherent light without defocus. The common blocks CBWIND, FOUUSER, HORIMG, IMG2PR, IMG3PR, IMGFLG, IO1, OPTIC, SPECTR, and PCTERM are needed by FASTPC. The current wavelength ILMBD is passed as a parameter from routine CLCMTF to routine FASTPC. The variables VCOND and VCOBJ contain the coherent cutoff frequencies of the condenser lens and of the objective lens, respectively. FASTPC runs about four times faster than PARCOH with NCOHAR=15.

As discussed in the Chapter 2, many coherence areas produce the same intensity pattern in the in-focus image plane. A particular "coherence area" A gives rise to a set of diffracted beams within the acceptance angle of the objective lens. An adjacent "coherence area" B that neither shifts new beams into nor present beams out of the acceptance angle produces the same intensity pattern as A. Thus it is pointless to calculate B's pattern. It should be noted that, although A and B give rise to the same pattern in the image plane, *the patterns add in intensity and are not coherent with each other*. Therefore, diffraction rings from dust and dirt in the optical system is lessened by many coherence areas, regardless of whether or not they give rise to the same intensity patterns in the image plane.

The routine begins by calculating the variables VCOND and VCOBJ, which contain the coherent cutoff frequencies of the condenser lens and of the objective lens, respectively. Three variables, VADSKP("ν add skip"), IENDFL("end flag"), and IFCINC("frequency component increment") are also initialized. VADSKP, IENDFL, and IFCINC are used to prevent redundant calculations. Do loop 10 searches for the largest positive mask spatial frequency included within the acceptance angle of the objective ν_i^+ . V1 is the fundamental frequency of the mask pattern, and thus the mask frequencies are multiples of V1. The difference between the objective lens cutoff frequency and the largest mask frequency is stored in VTMP1. VTMP2 stores the difference between V1 and VTMP1. Figure 3.2-1a shows the diffracted beams for the dc source wave along with the acceptance angle of the objective lens, the angle between the \pm VCOBJ lines. The minimum of (VTMP1,VTMP2) represents the angular distance one can shift the diffracted mask frequencies without affecting beams within the acceptance angle; that is, no new beams are shifted into the angle and no old beams are shifted out of the angle. Source waves smaller than $\nu_{s1} = VS1$ contribute to the same intensity pattern in the in-focus image plane, where

$$VS1 = \text{AMIN1}(VTMP1, VTMP2)$$

In loop 10, IFCPSV("index-frequency-component-save") saves the index of array FSAMSK of

³ Diffraction rings become a problem for $\sigma < .15$.

the highest frequency component that is included in the acceptance angle of the objective lens. Before the routine reaches do-loop 100, IFCPSV contains the index of the frequency component that will either be shifted out of the acceptance angle or be shifted into the acceptance angle during the second coherence area calculation. INCFAC("increment factor") determines whether IFCPSV is decremented or incremented after the first pass through the routine. IFCPSV and INCFAC are used to prevent redundant calculations.

Depending on whether $VTMP2 > VTMP1$ or whether $VTMP1 > VTMP2$, the block of statements beginning with 15 or with 13 is executed. VS1, VS2, and VS3 are the first three source wave spatial frequency increments used by the routine. They are multiplied by .99999 to take care of computer roundoff error in the comparison statements of loop 50. Figures 3.2-1 demonstrate how VS1,2,3 are chosen. Figure 3.2-1a shows the diffracted mask beams in their relation to the acceptance angle for the case $VTMP1 < VTMP2$ and $\nu_s = 0$. For source waves ν_s in the range

$$|\nu_s| < VTMP1 ,$$

the spatial frequencies accepted by the objective remain the same. Thus the first intensity calculation takes place with $VS1 = VTMP1$, figure 3.2-1b. ν_i^+ is considered to be just within the acceptance angle; and ν_{i+1}^- is still outside of the acceptance angle. The source wave frequency is then incremented by VS2, figure 3.2-1c, such that ν_i^+ is shifted outside the angle and that ν_{i+1}^- is outside the angle. For the case depicted in figure 3.2-1a,

$$VS2 = VTMP2 - VTMP1$$

The intensity pattern contribution is calculated for the set of beams within the angle and added to the previous pattern. For the next intensity pattern calculation, beam frequency ν_{i+1}^- is shifted inside the acceptance angle, figure 3.2-1d. Thus the third shift is $VS3 = V1 - (VTMP2 - VTMP1) = V1 - VS2$ and is shown in figure 3.2-1d. All future shift increments toggle between VS3 and VS2. For example, $VS4 = V1 - VS3 = VS2$. The routine continues to increment the source wave frequency $\nu_s = VS$ until it exceeds the angle of rays produced by the condenser lens. Note that, if $VTMP1$ is greater than $VTMP2$, IFCPSV is incremented by 1 for the reason explained above.

Although the central part of the routine is nearly identical to that discussed in subroutine PARCOH, several important differences should be noted. The normalization in FASTPC is based on the magnitude of the shift increments, $VSADD$ (present increment) + $VADSKP$ (past increments producing the same pattern). The normalization constant RNORMAL is equal to the fraction of the acceptance angle of the objective lens that the source wave incremental area occupies. If the mask were completely transparent, the normalization constant represents the fraction of the intensity incident on the resist surface due to the present source wave incremental area.

$$RNORMAL = \frac{VSADD + VADSKP}{(VCOBJ * AMIN1(SIGMA, 1.))}$$

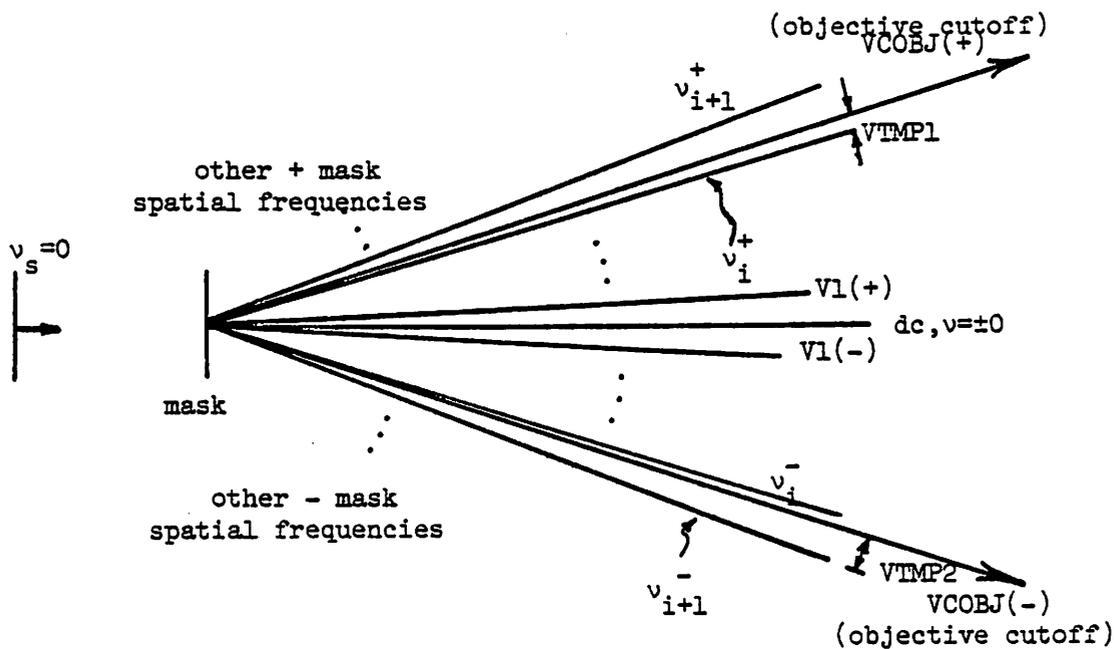
VADSKP adds the weighting from a previous calculation that was skipped; see text below. Note only the positive frequency increments are considered, since VCOBJ is the positive frequency cutoff. As in PARCOH, a variable FACTOR is defined as

$$FACTOR = .5 \cdot \sqrt{2} \cdot \sqrt{.5 \cdot RNORMAL}$$

The $.5 \cdot \sqrt{2}$ refers to a field of weighting $\sqrt{2}$ incident on a mask spatial frequency with two beams, each with weight .5. The added factor $\sqrt{.5 \cdot RNORMAL}$ refers to the intensity calculation,

where the .5 comes from the $\frac{1}{2} \text{Re}\{E \cdot E^*\}$ and RNORMAL is the normalization constant.

Before the normalization constant is calculated, a check for redundant calculations is made. If the mask frequency that is on the verge of being shifted out of (or into) the acceptance angle of the objective has a zero (or close to zero) weighting, the present coherence area effect is identical to the next coherence area effect. Therefore, the program proceeds to the next coherence area without calculating the effect of the present coherence area. The present area's weighting, VSADD, is simply added to the next area's weighting. VADSKP stores the

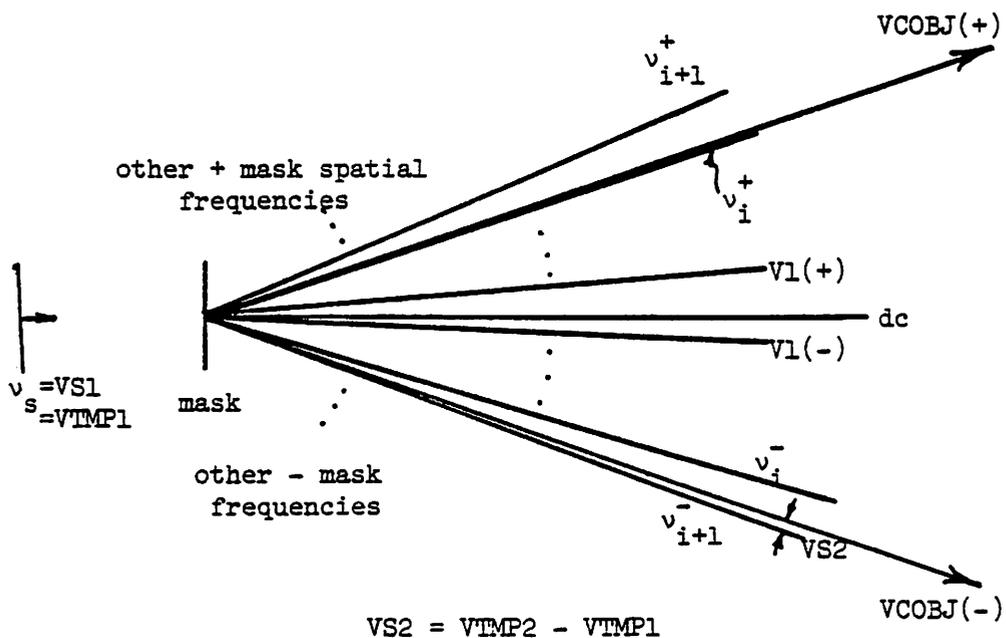


$$VTMP1 < VTMP2$$

$$VS1 = VTMP1$$

Figure 3.2-1a

Figure 3.2-1b



$$VS2 = VTMP2 - VTMP1$$

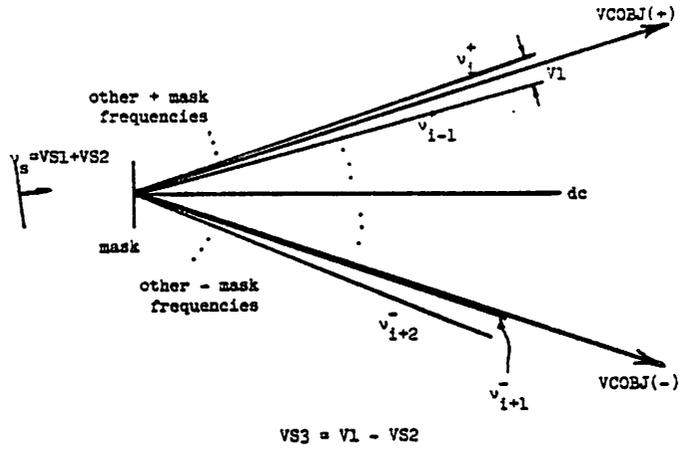


Figure 3.2-1c

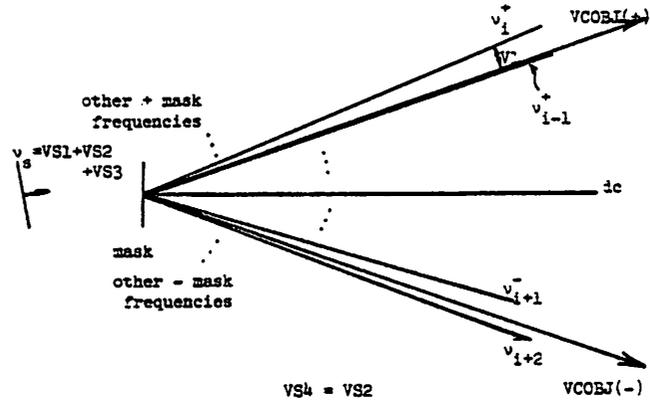
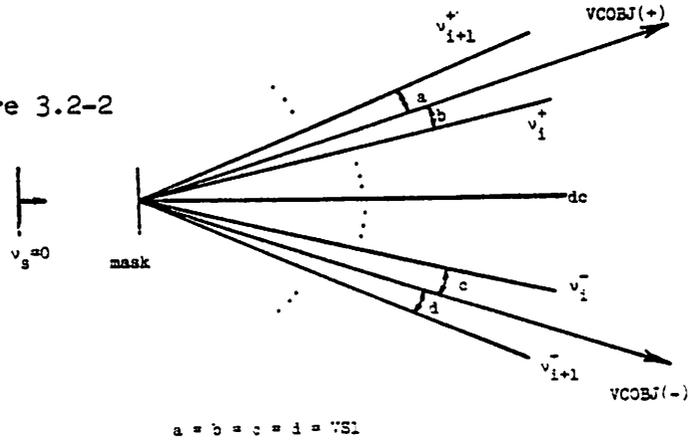


Figure 3.2-1d

figure 3.2-2



present area's weighting. However, if the present area is the final area calculated, the routine may not delay calculation of the present area intensity contribution. IENDFL checks for the final area calculation. When the index IFCPSV ≤ 0 , ICHECK maintains the proper index for array FSAMSK.

Perhaps another difference between PARCOH and FASTPC worthy of mention is the importance of the IF statements in loop 50. As in PARCOH, mask frequencies of less than .001 are simply ignored. The shifted positive mask frequencies are checked to insure that they are included in the acceptance angle. The shift frequency increments VSi were multiplied by .99999 in order to compensate for computer roundoff error. A problem can develop if VTMP2=VTMP1, as in figure 3.2-2. During the first intensity calculation, beam ν_i^+ should be included and beam ν_i^- should not be included. Because of roundoff error, both may be included. Multiplying the shift increments VS1,2,3 by .99999 alleviates the problem. The same comparison principle is used for the shifted negative mask frequencies. Negative shifted frequencies(SHFREQ) are kept if they are greater than -VCOBJ. Positive shifted frequencies are kept if they are less than VCOBJ. If the cutoff frequency of the objective has the same value as a multiple of the mask fundamental, it will be included in the first intensity calculation. Because of the assumption of a small mask, the multiple of the mask fundamental should only be included for patterns near the center of the mask.

The shift increments are toggled between VS2 and VS3 in loop 100 with the use of MOD(I,2) as a toggle. The variable IFCPSV is decremented or incremented, depending on the value of INCFAC*IFCINC. Note that if VTMP1>VTMP2, ICHECK(=IFCPSV for $i \geq 1$) goes as i, i+1, i-2, i+3, i-4, ..., where i is the index of the last beam inside the acceptance angle. If VTMP2>VTMP1, i represents the index of the first beam outside the acceptance angle. For IFCPSV ≤ 0 , ICHECK is equal to 2-IFCPSV. Statements 65 through 99 determine the last shift increment. If the temporary shifted frequency VSTEMP= ν , is less than or equal to the condenser lens cutoff frequency VCCOND, the routine continues as planned. If the shifted frequency is greater than VCCOND, the routine calculates the final shift increment VSADD=VCCOND-VS and sets the present source frequency to VCCOND. Also, the variable IENDFL is set equal to 1, in order to indicate the final coherence area calculation. If VSADD is less than .001, the routine doesn't bother to calculate the the intensity contribution. A negative VSADD means that the routine should jump out of loop 100 and is detected by the "less than .001 IF" statement. The factor of .99999 in the shift increments protects against roundoff error.

The final intensity is stored in HORINT as before.

References for Chapter 3:

- 1) S.N. Nandgaonkar, "Design of a Simulator Program(SAMPLE) for IC Fabrication", M.S. Report, Dept. of EECS, U.C. Berkeley, 1978.
- 2) L. Levi, *Applied Optics*, vol. 1, John Wiley and Sons, Inc., New York, 1968, p 456.
- 3) J.W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, 1968, p 114.

Chapter 4

The EXPOSE Machine

EXPOSE provides an array of M values as a function of position (x,z) in the resist for a particular exposure energy incident on the resist surface. Since the method of exposure used by the EXPOSE machine is identical to the method used by Neureuther and Dill^{1,2}, only a brief description of the theory will be given here. Several wrinkles in the code make a longer description of the code appropriate.

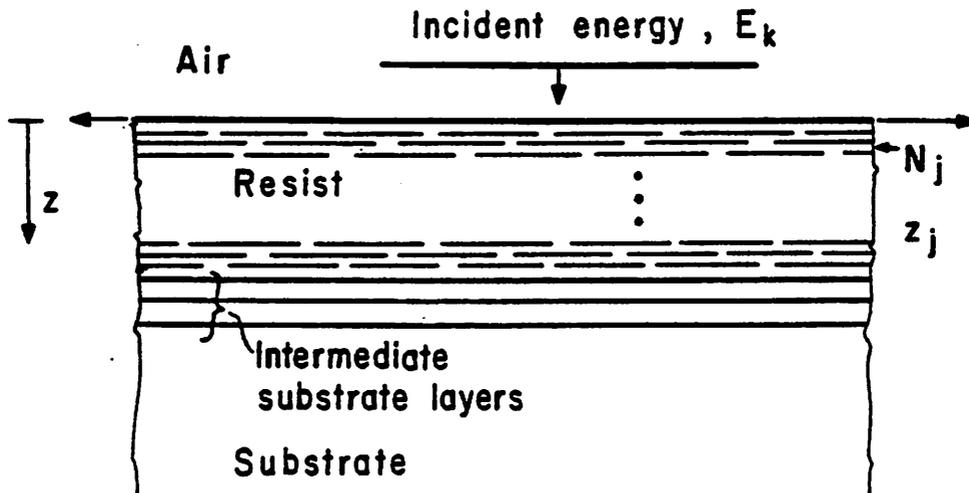
4.1 Theory

Although the exposure of the resist is a two dimensional problem, it has artificially been separated into two one dimensional problems. The one dimensional problem in z assumes that the exposure energy is perpendicularly incident on the surface of the resist, figure 4.1-1. The resist is divided into layers, each with a complex index of refraction N_j , which depends on the resist properties and the relative inhibitor concentration M_j of the layer. Initially all z_j layers start with $M_j=1$ and with the same value of N_j . The resist is then exposed with incident energy E_1 . Each layer receives a different amount of exposure energy, $(E_1)_j$, due to standing wave effects and due to attenuation of the fields in the resist. A new N_j and M_j for the layer j is calculated as a function of $(E_1)_j$. A second energy increment E_2 is used to expose the resist further, and new M_j 's and N_j 's are calculated for each layer. The process continues until a heavy exposure energy, or *dose* is reached; and a exposure table of M versus incident energy and z_j layer values, $M(z_j, E_k)$, is created. The energy increments $E_{k+1}-E_k$ are picked such that the resulting incremental changes in M are roughly the same. The one dimensional algorithm used to determine the energy absorbed for each layer is described by Berning³. Multiple wavelengths are handled by sharing each incremental exposure of the resist between the different wavelengths in proportion to their relative intensities. The dose is the energy in $\frac{mJ}{cm^2}$ of all of the wavelengths incident on the surface of the resist, i.e., the energy measured by a perfectly absorbing light meter.

If we assume that the intensity distribution produced by IMAGE does not spread while propagating through the resist, the distribution can be broken up into vertical columns, x_i , in x of cells in z . The exposure, or M_j value, of each cell in a particular column is determined by the exposure energy E_i incident on the top cell in the column. A two dimensional array of M versus resist position (x_i, z_j) is created by using the one dimensional energy profile in x created by IMAGE in conjunction with the energy table of EXPOSE, which is one dimensional in z . The dashed vertical line in figure 4.1-2 indicates the incident exposure energy that the column of resist has received. The actual M_j values in the column of resist due to exposure E_i are interpolated from the columns E_k and E_{k+1} in the exposure table $M(z_j, E_k)$, where E_i is between E_k and E_{k+1} . The maximum dimensions of the array $M(x_i, z_j)$ correspond to the depth of the resist and the width of the intensity image plus a linearly extrapolated boundary layer.

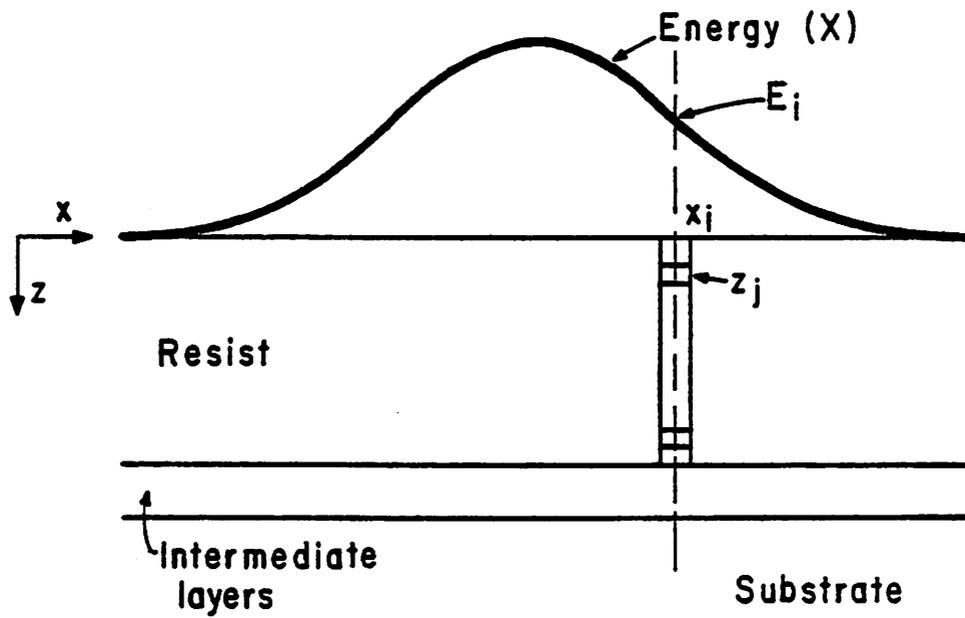
$M(x,z)$ can be massaged to model various post-exposure processes. A diffusion routine simulating post exposure bake has been written and plans for further additions have been made. The $M(x,z)$ values are easily translated into the development rates needed by the developing routines, DVELOP.

³written at U.C. Berkeley by W.G. Oldham and S. Nandgaonkar.



ONE DIMENSION : z

Figure 4.1-1



TWO DIMENSIONS : x, z

Figure 4.1-2

4.2 Code

The code for the EXPOSE machine is contained in four subroutines, diagrammed in figure 4.2-1. The code is written in standard FORTRAN and is portable to most minicomputers. The following text, the comments in the code, the Common Block Documentation, and reference 1 should enable one to understand the code in enough detail to permit modifications.

EXPOSE

Subroutine EXPOSE is the sub-controller for the EXPOSE machine. The variable IRUMZD(*re-use M of z and dose*) in block RUCOMP(*re-use computation*) decides whether routine CLCMZD(*calculate M vs. z and dose*) or routine CLCMXZ(*calculate M vs. x and z*) is called. CLCMZD calculates the exposure table $M(z_j, E_k)$. Using the exposure table and the intensity distribution from IMAGE, CLCMXZ calculates the array $M(x_i, z_j)$. The routines were separated in order to save computer time; and sub-controller EXPOSE attempts to use the existing exposure table for subsequent problems given to it by the Controller. If the doses required by CLCMXZ are outside of the exposure table calculated by CLCMZD, CLCMZD is recalled by sub-controller EXPOSE to expand the exposure(z-dose) table. The exposure table has a maximum recalculation limit. Beyond the limit, an error message, suggesting that the user used a ridiculously high dose, is printed out. Control is then returned to the Controller. The expanded table is used until the Controller asks for subroutine CLCMZD to be run again, which occurs the first time EXPOSE is run during a batch.

CLCMZD

Subroutine CLCMZD produces a two dimensional array of M versus z and dose, RMZDOS(J,IENDIV). Variables passed to and from subroutine CLCMZD are explained in the Common Block Documentation, appendix C. The first section of CLCMZD initializes the matrices SLBNDX and THIC. DELTZ is the thickness in microns of each resist layer; all the layer thicknesses are stored in THIC() except for the substrate. THIC(1) is the air and is left uninitialized. Initially, SLBNDX(ILMBD,IZPOS) holds the complex index of each wavelength, calculated on the basis of $M(x,z)=1$, for each resist layer.

The next section initializes CUENAB(J,IENDIV) and parts of RMZDOS. The cumulative energy absorbed(CUENAB) is really an "effective" cumulative energy. It represents the summation over all wavelengths of the (relative intensity at that wavelength)*(actual intensity)*(C at that wavelength). The sum of the relative intensities is equal to 1.

The bulk of the remainder of the routine encodes the equations found in reference 3. RMZDOS gives M at the position J for the energy division IENDIV. The actual *incident* energy on the resist surface, without subtracting reflected energy, is contained in the array EXPOS(IENDIV). The successive energy increments EXPOS(IENDIV)-EXPOS(IENDIV-1) are picked such that the resulting M increments are roughly the same. The array EXPOS contains the total relative intensity of the various wavelengths in millijoules per cm-squared. Physical conventions, shown in figure 4.2-2, are as follows: $J=1$ is the air "layer"; $J=56$ is the maximum value of the substrate "layer"; $J=55,54,53,52$ are the maximum values of the four possible intermediate layers; currently there is a maximum of 50 photoresist layers; the variable IZPOS or J equal to j for an interface refers to the $j,j+1$ interface.

The algorithm for the successive bleaching divides the resist into layers, each with a complex index $N=n-ik$ stored as a function of wavelength in SLBNDX(ILMBD,J), where ILMBD is the wavelength number* and J is the z position number. SLBNDX is calculated for each energy increment. The k th energy increment is stored in EXPOS(k). EXPOS(1)=0 and corresponds to no exposure. For example, if there were two wavelengths and the fifth energy increment update of SLBNDX had just been completed, the program would calculate the energy absorbed in each layer for the first wavelength using the fifth energy increment update for SLBNDX for that wavelength and then repeat the calculation for the second wavelength and

* A maximum of ten wavelengths are allowed.

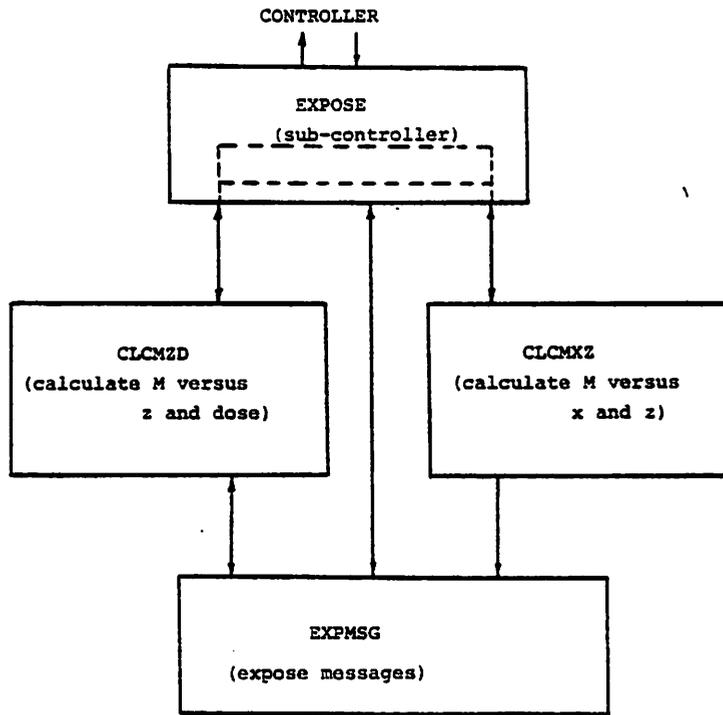


Figure 4.2-1

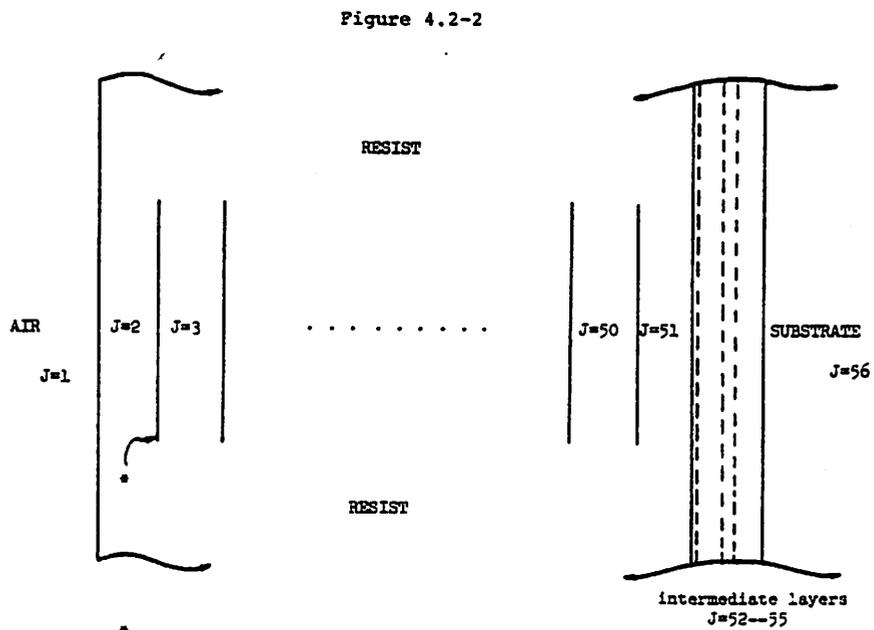


Figure 4.2-2

* J,J+1 interface--
referred to as J or IZPOS interface

its fifth update for SLBNDX.

The section of code containing the linearization of M with exposure energy is located in two parts in CLCMZD. The linearization section attempts to set the exposure energy increments of the exposure table such that M changes roughly the same with each energy increment. The equations used are as follows:

$$M_{k-1} + \delta M_k = e^{(E_{k-1} + \delta E_k) \cdot K_d} \quad (4.2-1a)$$

$$\text{where } K_d = - \sum_l I_{rel_l} \cdot C_l e^{-(A_l + B_l)d} \quad (4.2-1b)$$

d the thickness of the resist. M_{k-1} is the value of M at the end of the $k-1$ th energy exposure increment. δM_k is the change in M that the k th energy increment δE_k produces. A_l , B_l , C_l are the A,B,C's at the l th wavelength. I_{rel_l} refers to the relative intensity at the l th wavelength. Solving equations (4.2-1) gives

$$\delta E_k = \frac{\ln(M_{k-1} + \delta M_k)}{- \sum_l I_{rel_l} \cdot C_l e^{-(A_l + B_l)d}} - E_{k-1} \quad (4.2-2)$$

In the actual algorithm E_{k-1} is divided by the variable PWRAT1, in order to recognize the reflection loss at the resist-air interface. PWRAT1 is the fraction of the incident energy passed into the resist from the air for the wavelength located in the first dimension of WLABC--WLABC(1,n,o,p). The correspondence between the FORTRAN variables and the variables used here are:

- | | |
|--|----------|
| $M_{k-1} \rightarrow$ RM | (4.2-3a) |
| $\delta M_k \rightarrow$ DELTM | (b) |
| $K_d \rightarrow$ TEMPEX | (c) |
| $E_{k-1} \rightarrow$ EXPOS(IENDIV-1) | (d) |
| $E_k \rightarrow$ EXINC(IENDIV) | (e) |
| $I_{rel_l} \rightarrow$ RELINT(ILMBD) | (f) |
| $d \rightarrow$ THICK(1) | (g) |
| $A_l - B_l - C_l \rightarrow$ WLABC(ILMBD,2-3-4) | (h) |

The linearization algorithm solves for the incremental amount of k th incident energy needed to expose a resist of thickness d to an M of $M_{k-1} + \delta M_k$. The algorithm is approximate--as the assumed attenuation of the resist is $A+B$, the reflection from the front surface is determined from the primary wavelength only, and the standing wave effects are ignored. The "completeness" of exposure of the resist--as reflected in the exposure table RMZDOS--is controlled in the Controller through the constant picked in the DELTM assignment statement. The default value for the constant is $.7/(NENDIV-1)$ --giving an approximate resist exposure at the bottom of the resist of $.7$. NENDIV refers to the number of energy divisions the table is divided into. A large value for NENDIV results in a fine exposure table, and a small value for NENDIV results in a coarse exposure table. The Controller sets the original value of NENDIV. Sub-controller EXPOSE increases these values as necessary.

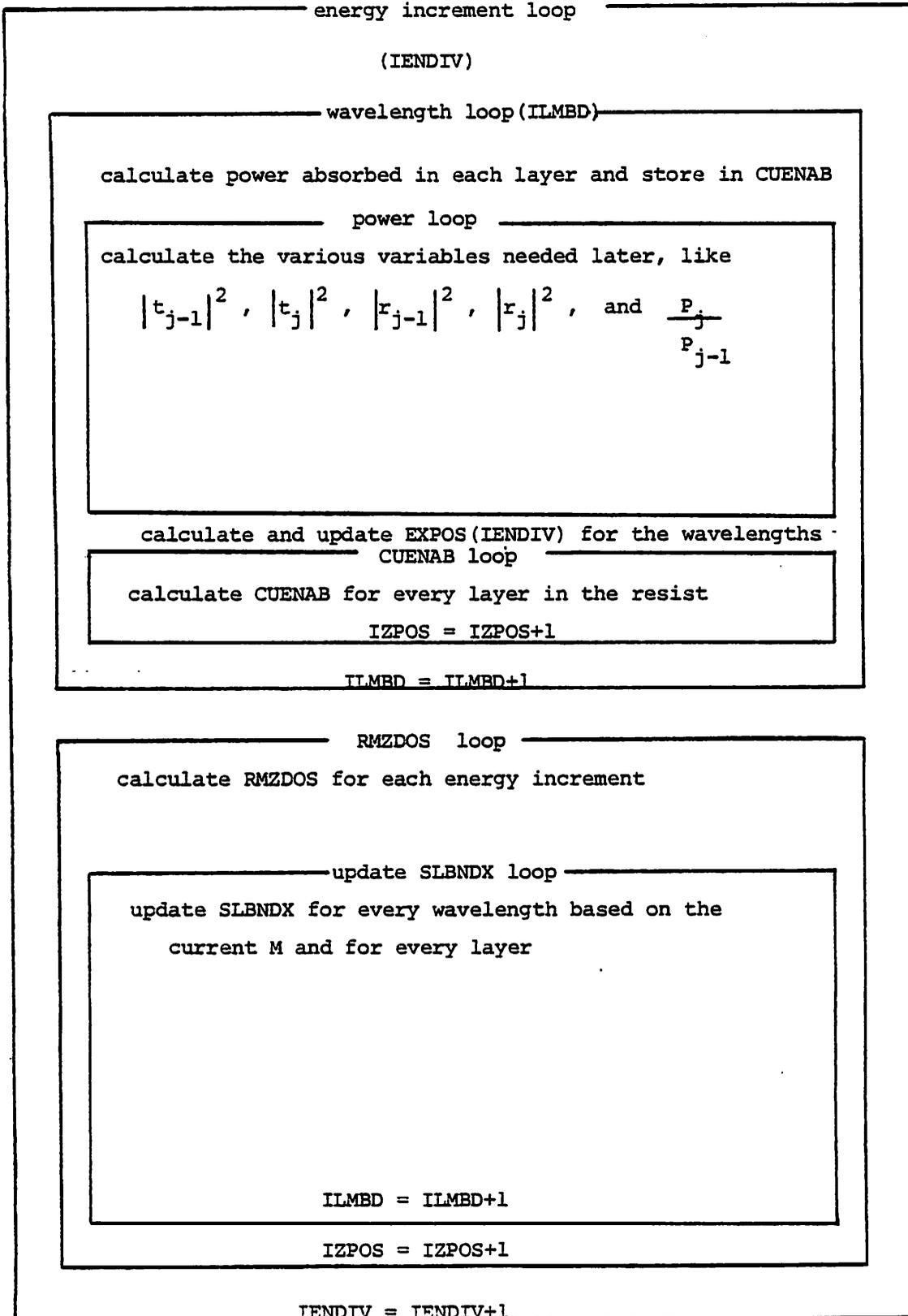
The rest of the routine runs the multiple wavelength exposure for the present energy increment IENDIV. Figure 4.2-3 depicts the loops of this section.

The basic equations that were programmed from Berning³ are

$$r_{j-1} = \frac{e^{-i2\Phi_j}(F_j - r_j) - F_j(1 - F_j r_j)}{F_j e^{-i2\Phi_j}(F_j - r_j) - (1 - F_j r_j)} \quad (4.2-4a)$$

$$t_{j-1} = \frac{(F_j^2 - 1)t_j e^{-i\Phi_j}}{F_j e^{-i2\Phi_j}(F_j - r_j) - (1 - F_j r_j)} \quad (b)$$

Figure 4.2-3



$$t_m = \frac{2\sqrt{n_0 \text{Real}(N_{m+1})}}{n_0 + N_{m+1}} \quad (c)$$

where $n_0 = 1$ = the real index of refraction of the medium (air) above the photoresist and N_{m+1} is the complex index of refraction of the layer $m+1$.

$$r_m = F_{m+1} \quad (d)$$

$$\Phi_j = i \left[\frac{2\pi}{\lambda_0} N_j l_j \right] \quad (e)$$

where l_j is the thickness of the j th layer.

$$F_j = \frac{n_0 - N_j}{n_0 + N_j} \quad (f)$$

Layer $m+1$ is the substrate. The ratio of the power absorbed in the j th layer to the power absorbed in the $j-1$ th layer is given by

$$\frac{P_j}{P_{j-1}} = \frac{|t_{j-1}|^2 (1 - |r_j|^2)}{|t_j|^2 (1 - |r_{j-1}|^2)} \quad (g)$$

The power absorbed in the j th layer is A_j

$$A_j = \frac{P_{j-1} - P_j}{P_{1-}^{(i)}} = (1 - R) \left[1 - \frac{P_j}{P_{j-1}} \right] \prod_{2 \leq q \leq j-1} \frac{P_q}{P_{q-1}} \quad (h)$$

where $P_{1-}^{(i)}$ is the incident power. The relations between the variables in the above equations and the FORTRAN variables in the program are as follows:

$$\begin{aligned} r_j &= \text{RJ} & t_j &= \text{TJ} \\ r_{j-1} &= \text{RJM1} & t_{j-1} &= \text{TJM1} \\ |r_j|^2 &= \text{RJS} & |t_j|^2 &= \text{TJS} \\ |r_{j-1}|^2 &= \text{RJM1S} & |t_{j-1}|^2 &= \text{TJM1S} \\ F_j &= \text{FJ} & n_0 &= (1., 0.) \\ \frac{P_j}{P_{j-1}} &= \text{PJDJM1(J)} & r_m &= \frac{1 - \text{SLBNDX}(\text{subs index})}{1 + \text{SLBNDX}(\text{subs index})} \\ \Phi_j &= \text{PHIJ} & 1 - |r_0|^2 &= \text{PWRAT1} \end{aligned} \quad (4.2-5)$$

CLCMXZ

Subroutine CLCMXZ creates a two dimensional array RMXZ that contains M as a function of x and z . The maximum dimension of RMXZ is (52,52) corresponding to the maximum dimension (50,50) of resist points plus a linearly extrapolated boundary layer. The resist point $(x,z) = (0.,0.)$ is the middle of the upper boundary of cell RMXZ(2,2). The resist point $(x,z) = (\text{XMAX}, \text{ZMAX})$ is the middle of the lower boundary of cell RMXZ(NMHPTS+1, NPRPTS). RMXZ is calculated using the horizontal energy array HOREN, the exposure energy array EXPOS, and the array RMZDOS containing M as a function of energy and z position. The section that degrades M simply subtracts the value of DGRADM from every cell in the matrix. The energy in HOREN and EXPOS is the summation of (relative intensity) · intensity · time for all of the wavelengths.

EXPMSG

Subroutine EXPMSG is the message subroutine for machine EXPOSE. It is similar in format to IMGMSG.

References for Chapter 4:

- 1) A.R. Neureuther and F.H. Dill, "Photoresist Modeling and Device Fabrication Applications", *Optical and Acoustical Micro-Electronics*, Polytechnic Press, N.Y., pp 233-249, 1974.
- 2) F.H. Dill, A.R. Neureuther, J.A. Tuttle, and E.J. Walker, "Modelling Projection Printing of Positive Photoresists", IBM Research, February 2, 1975, RC 5261.
- 3) Berning, Peter H.; "Theory and Calculations of Optical Thin Films", *Physics of Thin Films*, vol 1, 1963.

Chapter 5

The DEVELOP Machine

Machine DEVELOP develops the resist exposed by machine EXPOSE. The basic algorithms were originally conceived and written by Bob Jewett¹. Enough changes have been made in the code to warrant fairly complete documentation of the present machine.

5.1 Theory

The present version of DEVELOP follows a developing contour by keeping track of the position of the points on the developing string. At time $t = 0$, the developing string points are evenly spaced and situated at the top of the resist, as in figure 5.1-1. z equal to zero specifies the top of the resist; the particular zero time x values of the string points depend on the WINDOW and EDGE parameters chosen by the user. As development progresses, the string proceeds to define the resist-developer boundary. The rate that a string point travels is determined by the M value at the string point's position via the equation

$$\text{rate} = R(M) = a \cdot e^{E_1 + E_2 M + E_3 M^2} \quad (5.1-1)$$

The constant a serves to give the rate in microns/second. E_1 , E_2 , and E_3 are predetermined constants of the resist, the developer, the temperature, and the processing conditions that one wishes to include. The direction of a string point is determined by averaging the directions of the perpendicular bisectors of the two adjacent string segments, as in figure 5.1-2. The endpoints of the string maintain the direction initialized at the start of the routine.

Several problems occur using the string point approach. Loops may develop in the string, as in figure 5.1-3a. A point is added at the intersection of the intersecting string segments; and the loop is removed, as in figure 5.1-3b. Part of the string may develop outside of the resist boundary, as in figure 5.1-4a. In order to maintain string boundary integrity, points in the boundary area develop at the same rate as the boundary point. Points beyond the boundary area are slowed to a low rate in order to prevent the string from curving around and re-entering the resist. Periodically, the points outside of the boundary are deleted, as in figure 5.1-4b; a new boundary point is defined where the string crosses the boundary; the new endpoint direction is the same as the previously defined direction for the boundary point. The lower boundary also has a boundary area, figures 5.1-5. Strings in the lower boundary area have development rates equal to their resist counterparts; in other words, the lower boundary condition is reflective. If the boundary point develops into the lower boundary area, much of the string in the lower boundary area is deleted in order to save computer time. The various boundary conditions are an attempt to give reasonable results for the majority of patterns.

5.2 Code

Written in standard FORTRAN, the code for the DEVELOP machine is portable to most minicomputers. Figure 5.2-1 diagrams the relations between the routines. DEVELOP requires a large set of internal parameters, which it attempts to set itself. User-requested machine options include continuation of development, punched cards for the string points, diagnostics, and a higher accuracy run (additional points on the starting string).

Figure 5.1-1

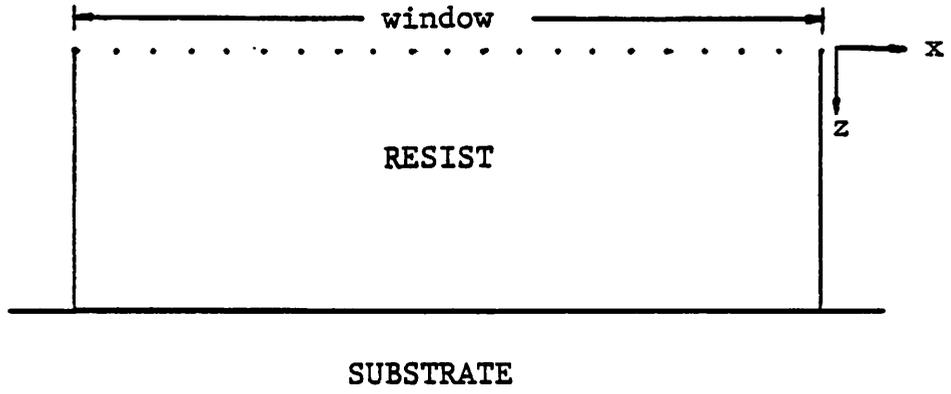


Figure 5.1-2

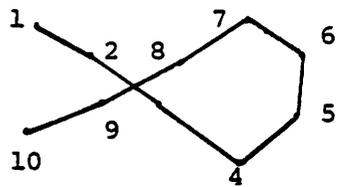
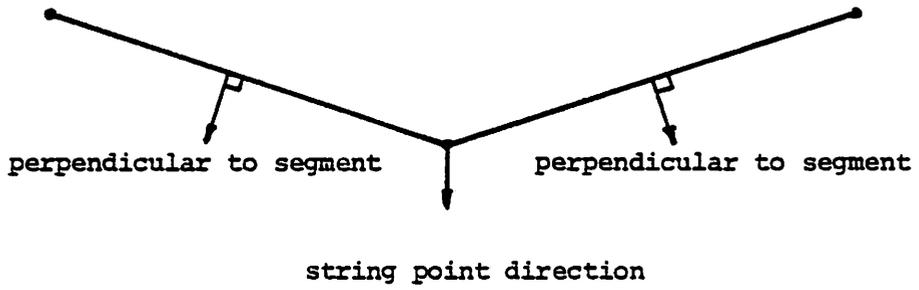


Figure 5.1-3a

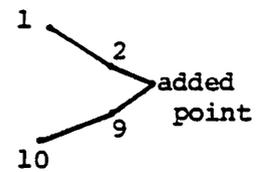
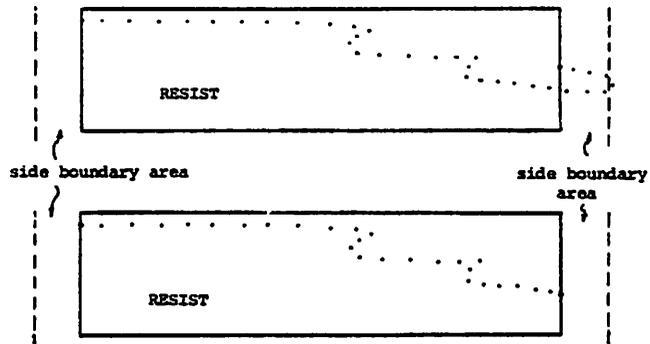


Figure 5.1-3b



Figures 5.1-4a,b

Figures 5.1-5a,b

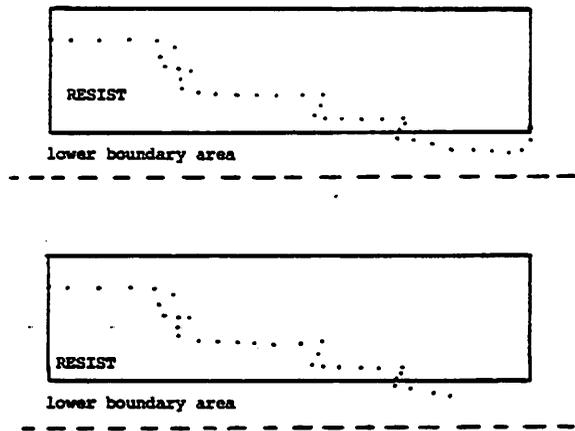
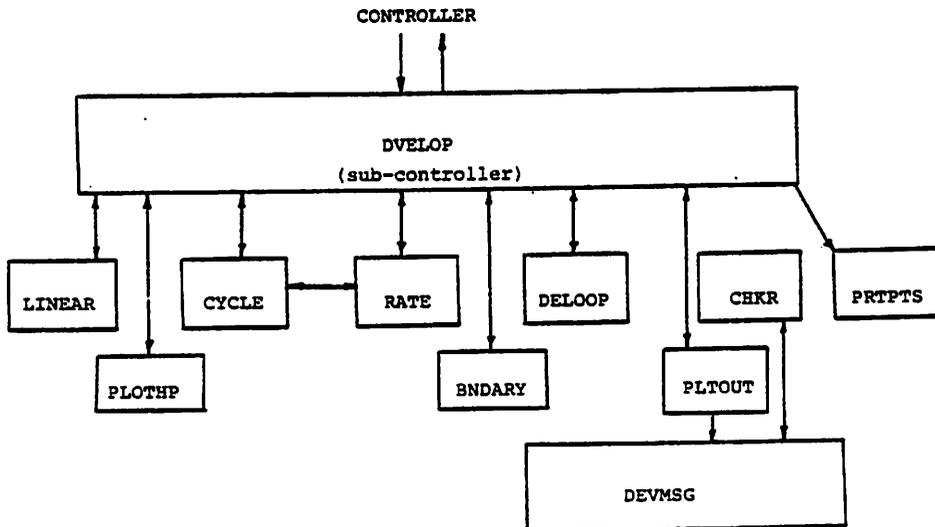


Figure 5.2-1



The following text should read in conjunction with a copy of the code and the Common Block Documentation, appendices C and D.

DVELOP

Subroutine DVELOP is the sub-controller for the DEVELOP machine. DVELOP lays claim to a modicum of intelligence, although improvements in the internal parameter assignments would be desirable. Machine DEVELOP advances the points on a string and remembers their current position. During advances, the string segments may become too long, resulting in poor accuracy, or too short, resulting in wasted computer time. Subroutine CHKR is called periodically to review and to correct string segment length. The number of string advances between calls to subroutine CHKR is given by the variable NADCHK (*number of advances between checks*). The variable NCKOUT stores the number of checks, or calls to CHKR, between outputs. The development time of the first output, the development time of the last output, and the time between intermediate developments are user designated and stored in the variables DEVSRT, DEVEND, and DEVINC, respectively. The number of advances per output can be calculated by multiplying $NCKOUT \cdot NADCHK$. Both the variable NCKOUT and NADCHK are important internally set parameters. Initially they are set by the resist breakthrough section of the sub-controller for the initial development time. They are reset for the intermediate development times. They are reset again after resist breakthrough to reflect lower development rates. The two intermediate values for NCKOUT are saved in NCKSV1 and NCKSV2 for optional printout. The intermediate (original) value for NADCHK is stored in NADSAV for optional printout. The breakthrough estimation section of the program attempts to estimate the time between string advances necessary to achieve accuracy for the highest exposure intensity in the pattern (worst case). The time between advances and not the number of advances per micron is the important parameter.

The first statements in DVELOP output the user specifications and set the maximum boundaries of the resist. Next, the breakthrough estimation section estimates the time taken by the developer to clear an opening in the resist. First, it finds the column of the RMXZ matrix, excluding the boundary cells, with the minimum M value in the $z=0$ cell (the first layer of the resist). Then it develops the column based on time increments of $4 \cdot \text{FACTOR}$ seconds, where FACTOR is an internal parameter of the breakthrough estimation section. The estimated time until breakthrough, TBREAK, is calculated. The development of the column is repeated with a time increment that is FACTOR of the previous increment. If TBREAK is within 5 percent of the previously calculated TBREAK, saved in TBSAVE, then NCNVRG (*number of convergences*) is incremented. The process is repeated until NCNVRG equals four, until four consecutive calculations for TBREAK are within 5 percent of the previous calculation. The time between string advances in the estimation section (DELTTM) is set to the value used for $NCNVRG=1$. $NCNVRG=4$ insures that TBREAK has converged to the correct time until breakthrough. It is necessary to check for convergence thoroughly, since an incorrect estimation for TBREAK will still cause breakthrough in about TBREAK seconds, due to "self-fulfilling prophecy". FACTOR is initially set to .6. In order to insure against a false convergence for high doses, FACTOR is set to .4 if the second estimate for TBREAK is less than 15 seconds. NADVAN is the number of advances needed per output, as calculated by the breakthrough estimation section. The time between string advances used in the program (TADV) will be recalculated using NCKOUT. NCKOUT is an unnecessary intermediate variable; it has been retained in the program for historical reasons and for flexibility. The number of advances between checks, NADCHK, is set equal to 3 and NCKOUT is set via the equation

$$NCKOUT = NADVAN/NADCHK + 1$$

NCKOUT is set liberally, in order to insure accuracy.

The next section of the program initializes various internal parameters. The number of string points, NPTS, is set according to the number of horizontal points, NMHPTS, requested by the Controller (currently 50). NPTS is increased if the journal paper flag, IDEVFL(3), is set. The variable MAXPTS is set; the delooping routine is called when NPTS exceeds MAXPTS.

MAXPTS lets the string get approximately 1.5 times as long as a linear string from (0.,0.) to (XMAX,ZMAX). The direction(CXZL and CXZR) and position(XZ(1) and XZ(NPTS)) are set for the string endpoints. Presently the direction for both points is set pointing directly into the resist. The endpoint directions are then normalized. Note that the position of the left endpoint is always (0.,0.), and position of the right endpoint is always (XMAX,0.). The *real x* positions of the string points bear no relation to either the Fourier series coordinate system in IMAGE or to the printout coordinate system with $x=0$ corresponding to the edge. Subroutine LINEAR is called to initialize the middle string points along the surface of the resist. The minimum and maximum string segment lengths are set for the x direction(SMINX,SMAXX), and the maximum segment length is set for the z direction(SMAXZ). SMINX, SMAXX, and SMAZZ are used by subroutine CHKR. If the journal paper flag is set, the maximum distances are reduced. TCHK and TADV are the time between "checks"(calls to CHKR) and the time between advances, respectively. Checks are made on some of the parameters before the main part of subroutine DVELOP is executed.

Loop 5 is the main controlling section of DVELOP. Loop 5 processes each output in sequence. NZFLG is a flag common to DVELOP and CYCLE that detects when the resist has developed through to the substrate(or oxide). After resist breakthrough, the NADCHK and NCKOUT are reset to lower values, reflecting the slower string development progression. TTOTSV(total time, save) records the exact breakthrough time. NFLG simply skips the present section for the remainder of the run. MAXPTS is set to 1.8 times the number of points in the present "delooped" string. Loop 3 cycles through the number of checks between outputs. Subroutine CYCLE advances the string NADCHK times. Subroutine CHKR is called if improper string segment lengths were detected in CYCLE. The short version(1) of DELOOP is used if NPTS exceeds MAXPTS. The long version(2) of DELOOP is called just before every output. The line printer plotting subroutine, PLTOUT, is called to record the present string point locations. Cards are punched by subroutine PLOTHP, if requested. After the number of outputs requested by the user is completed, control is returned to the Controller.

A temporary feature that allows for a continuation of development or for a descum operation has been included. If the last flag(#5) in the develop machine is 0, the normal run takes place(complete with initialization). It is assumed that the normal is run before the intermediate(no initialization) runs are made. If the last flag is 1, the development is continued with the current string point positions. It is assumed that the user will respecify E_1 , E_2 , E_3 , and possibly(probably) the initial development time(zero time is now the beginning of the continuation of development), the final development time, and the development time increment. A warning: the time between string advances is set by the initialization run(the normal run) and may be either too small or too large for the new E 's. The time between string advances will probably not be too large for the descumming operation with uniform etch rate. However, it may be too large for a continuation of development run, if the development rate with the new E 's is substantially greater than the rate with the old E 's. If the time between string advances is too small, the program burns up computer time unnecessarily(but no inaccuracies result). In order to save computer time for descum, make sure that the developer has broken through the resist before calling the develop machine with flag#5 set =1(i.e., before descumming). This insures that the string advance time is reduced.

LINEAR

Subroutine LINEAR initializes the string at time equal zero. The initialized string consists of a straight line between the endpoints set in subroutine DVELOP.

CYCLE

Subroutine CYCLE advances the string NADCHK times. Since the routine is very close to the code created by Bob Jewett, and since the current code contains many comment statements, only a brief discussion of CYCLE will appear here. The directions of the left and right endpoint advances are determined by the values of CXZL and CXZR, set in sub-controller DVELOP. The direction of a middle string point advance is determined by the average of the directions of the adjacent segments. The normalized direction of the point M, with (x,z)

position equal to $(\text{real}(XZ(M)), \text{imag}(XZ(M)))$, is found in the following manner. The directions of the normals of segments $M-1, M$ and $M, M+1$ are

$$\sqrt{-1} \cdot (XZ(M) - XZ(M-1)) \text{ and } (XZ(M+1) - XZ(M)) \cdot \sqrt{-1}$$

, respectively. The directions are added vectorially (added using complex arithmetic), giving the complex variable DT. DT is then normalized by dividing by its length $|DT|$. IFLAG is set equal to 1 if one or more string segments are outside of the length bounds. NZFLG indicates that the resist has broken through in TTOTSV seconds.

RATE

Function RATE calculates a rate for the (x, z) position designated by the complex parameter CZ by appropriately weighting the M values of the surrounding four points. The (i, j) indices of the matrix RMXZ refer to the (x, z) cell numbers; the value of M for the cell is contained in RMXZ(i, j). The rate is calculated using E_1, E_2, E_3 , and the weighted M value. Since the E_i values are valid only for $M \geq .4$, rates for M values $< .4$ are set equal to the $M = .4$ rate. Figure 5.2-2 shows the cells of RMXZ. Note that $(x, z) = (0, 0)$ is designated by the top-center of the cell RMXZ(2, 2). The upper and lower boundary areas consist of one cell each. The side boundaries consist of one and one half cells. Reflective boundary conditions are used for the lower boundary for 3 cells beyond the resist (beyond ZMAX). String points developing into the top boundary cell layer, beyond the side boundary cell layer, or into the fourth boundary cell beyond ZMAX are given an etch rate of $.0001 \mu/\text{sec}$. Points in the boundary layers are deleted in an appropriate manner in subroutine CHKR.

BNDARY

Subroutine BNDARY deletes points outside of the resist boundary in an intelligent manner. Presently, BNDARY is called by the sub-controller each time subroutine CHKR is called. String points are allowed to develop through the lower boundary, designated by ZM, and into the lower buffer. The maximum z of the lower buffer is designated by

$$ZM = ZMAX + 3 \cdot \text{DELTA}$$

For the point $(x; z = ZMAX + .01)$, the lower buffer returns the rate for the point $(x; z = ZMAX - .01)$. Points developing further than the $Z = ZM$ boundary receive a very low develop rate. String points that develop into the buffer areas are deleted, in order to save computer time. A string section that contains the left endpoint and that has developed into the left buffer area is deleted. The new endpoint is placed at the intersection of the buffer boundary ($x = 0$) and the string. Figures 5.2-3 depict the deletion of a left string segments. Left string sections that develop into the lower buffer area and that contain the left string point are deleted, except for four points in the lower buffer. The remaining four points insure string integrity above the lower boundary. Figures 5.2-4 depict the deletion of a lower left buffer string section. Right string sections that develop into the buffers are deleted in a similar manner.

CHKR

Subroutine CHKR modifies the string segment lengths to conserve computer time and to maintain accuracy. CHKR consists of two parts. The first part detects the number of segment lengths that are too long or too short and stores the number in the variable NCHNGE. The first part also stores the XZ array index of the right endpoint of each segment in the array INDEX. Thus XZ(INDEX(NCHNGE)) contains the (x, z) coordinates of the right endpoint of a segment needing length modification. The second part, loop 40, actually makes the necessary length corrections. The first section of loop 40 determines whether the segment in question, designated by the XZ index of $M = \text{INDEX}(N) + \text{NADD}$, is too long or too short. NADD stores the accumulated number of points loop 50 has added so far; the stored XZ index M is modified

Note: A change in the number of cells in the reflective boundary in function RATE necessitates a similar change in subroutine CHKR.

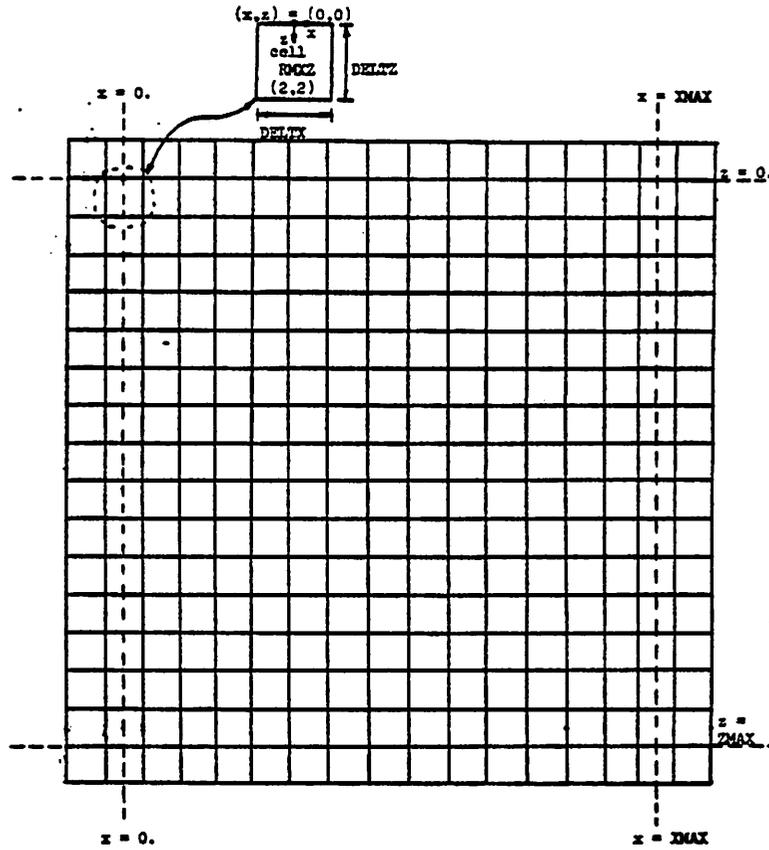
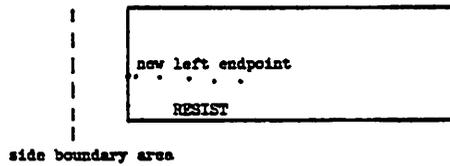
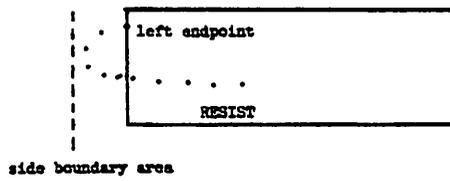
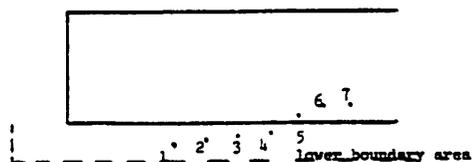
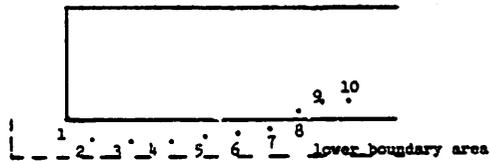


Figure 5.2-2



Figures 5.2-3a,b

Figures 5.2-4a,b



accordingly. If a segment is too long, as in figures 5.2-5, a point is added between M and $M-1$. The exact position of the added point is determined by a curve fit described in appendix E. The old point M becomes index $M+1$, the new point becomes index M , the old point $M+1$ becomes index $M+2$, etc.

If the segment is too short, as in figure 5.2-6a, the left and right points of the segment, designated by $M-1$ and M , are replaced by one point in the middle of the segment. The new point, newly designated as $M-1$, is half way between the old points $M-1, M$. $MLAST$ contains the right endpoint index of the previously lengthened segment. If the index M of the right endpoint of the present segment to be lengthened is equal to $MLAST$, the present segment is not lengthened since it was lengthened by previous action. In other words, if both segments $M-1, M$ and $M, M+1$ in figure 5.2-6a require lengthening, the lengthening of segment $M-1, M$ also lengthens segment $M, M+1$.

DELOOP

Subroutine DELOOP removes loops that develop in the string. DELOOP is passed the parameter $ITYPE$. $ITYPE$ equal to 2 removes all loops and called just before an output. $ITYPE$ equal to 1 is an attempt to save computer time and is called between outputs to delete intermediate loops. Intermediate loops contain many points, which increase the run time of subroutines $CHKR$ and $CYCLE$. The present frequency of intermediate calls to DELOOP was found to be a good compromise between saving time in $CHKR$ and $CYCLE$ and losing time in DELOOP. Subroutine $CHKR$ is called at the beginning of every call to DELOOP in order to increase accuracy. Although calling $CHKR$ before delooping usually results in an increase in string points and therefore a significant increase in computer time spent in delooping, the shape of the string is significantly improved².

Figures 5.2-7 show a typical loop intersection at segment $N, N+1$ and segment $M, M+1$. The base point of the loop deletion routine is designated by N . The point M begins $L2+1$ points from the point N . The routine checks to see if the points N and M are within the distance SX in x or SZ in z . If they are within both the x and z distances, the segments $N, N+1$ and $M, M+1$ are checked for intersection. The code for segment intersection is straightforward real algebra. If the segments intersect, a point is added at the intersection and the loop is deleted. It is very possible, especially when $ITYPE=2$, that the close proximity of M and N do not indicate intersection. M is then incremented, and the detection process continues until M reaches the end of the string. N is then incremented, a new M is defined, and the loop detection sequence is repeated.

The variables $L1$, $L2$, SZ , SX , and $NJUMP$ play an important part in saving computation time. $NJUMP+2$ specifies the number of points N jumps when it is incremented after an intermediate loop is deleted. $NJUMP$ must not be set too high, lest N be incremented past the origin of the next loop, which then will not be detected. $NJUMP$ is normally set equal to 2 for normal runs; $NJUMP$ is set equal to 4 for journal paper ($IDEVFL(3)=1$) runs, since there are more points on the string. The variable $L1$ stops loop searches when N is $L1$ points from the end of the string. $L1$ is not as critical a variable as $NJUMP$. The 2 determines the minimum loop size that is searched for by DELOOP. After N is incremented, M is initialized to $L2+1$ points beyond N . If the loop contains $L2$ points or fewer, the routine never detects the loop. For output calls to DELOOP ($ITYPE=2$), $L2$ is set equal to 1. If $L2$ is set too large for intermediate calls to DELOOP ($ITYPE=1$), large loops will be left upon leaving the subroutine. These loops generally enter regions of low M values and expand rapidly; they can cause serious problems. SX and SZ are the loop detection distances. If point M is within SX of point N in x and within SZ of N in z , then the segments $N, N+1$ and $M, M+1$ will be checked for intersection. The intermediate call to DELOOP sets SX and SZ to $SMAXX$ and $SMAZZ$, the maximum string length tolerances for x and z . The output call to DELOOP sets SX to $1.75*SMAXX$ and SZ to $1.75*SMAZZ$, in order to decrease the possibility of missing a loop.

²Very occasionally a loop is missed.

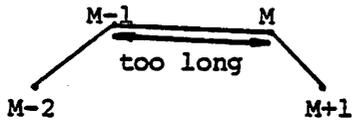


Figure 5.2-5a

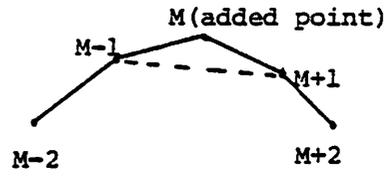


Figure 5.2-5b

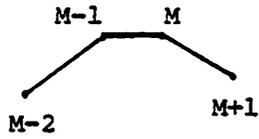


Figure 5.2-6a

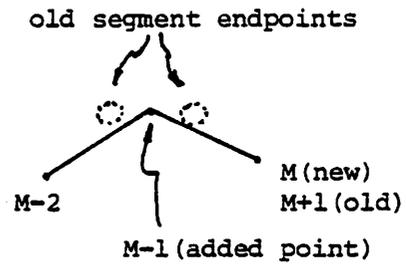


Figure 5.2-6b

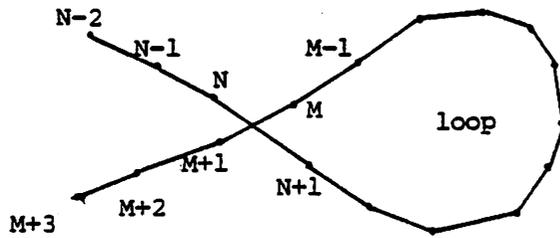


Figure 5.2-7a

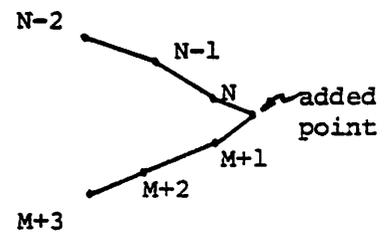


Figure 5.2-7b

Note that larger values for SX and SZ may cause nearby segments to be checked for intersection. Tightening the values of NJUMP, L1, L2, SZ, and SX causes DELOOP to detect loops with more certainty. However, the run time of the subroutine increases. The present values assigned to these parameters appear to be a reasonable compromise between accuracy and run time. Missing a large loop during an intermediate call to DELOOP can be disastrous in some cases, although most of the time the string has a tendency to correct itself.

PLTOUT

Subroutine PLTOUT plots the output strings on the lineprinter. IOUTPT designates the number of the output, or indicates initialization (IOUTPT=26), or indicates that the development program has dumped (IOUTPT=25) and that the output strings generated before the dump should be printed. Presently the routine relies on a large 123 by 99 array IPLT. The graph is scaled in x and z. The center of the cell IPLT(2,2) represents the point resist. The center of IPLT(122,98) represents the resist point (XMAX,ZMAX). The intersection of the resist and substrate is calculated by detecting the segment that intersects the substrate. Contours that have broken through the resist have additional information printed out. The resist-substrate intersection (or resist-oxide or whatever) for these contours is printed out. The routine attempts to locate the line segment crossing the resist-substrate boundary and calculates the intersection point. The resist-substrate intersection is given in microns with reference to the mask edge. The routine also attempts to print the resist sidewall angles (the angle between the substrate and the resist sidewall) for the contours that have broken through the resist. The routine detects the tips of the standing waves or the insides of the standing waves--depending on whether the resist is on the left or on the right side of the edge. A linear regression algorithm is used to form the straight line that defines the sidewall angle. The sidewall angle code will not produce a result if it detects fewer than two points, which may occur for thin resists with only one standing wave.

PLOTHP PRTPTS DEVMSG

Subroutines PLOTHP, PRTPTS, and DEVMSG punch cards for the HP plotter, print the string point positions, and print the messages for the develop machine, respectively.

5.3 Problems with the DEVELOP Machine

A number of problems have appeared in the develop routines. Although the problems seem to be solved, a brief review will serve to alert one to some of the difficulties. Double loops form frequently during development, as in figure 5.3-1a. Normally the call to DELOOP deloops them correctly. If DELOOP misses the first intersection q due to the fact that SX and SZ in DELOOP were not great enough to detect the intersection, and if the intersection r is detected, an incorrect deloop occurs and causes the pattern shown in figure 5.3-1b. If the problem occurs during an intermediate call to DELOOP, it probably will not be detected by the user. Fortunately, the error has a tendency to be self-correcting. Outputs like 5.3-1c are an indication that incorrect delooping has occurred. Increasing SX and SZ in subroutine DELOOP will insure that the problem doesn't occur. However, a small increase in SX and SZ results in a significant increase in computer run time. The present values of SX(Z) are 1.5*SMAXX(Z) for intermediate calls to DELOOP and 1.75*SMAXX(Z) for output calls to DELOOP. SX and SZ have been set conservatively (I hope). If, after exhaustive runs of the develop machine, incorrect delooping does not seem to be a problem, then one should reset them to 1.4 and 1.6 for intermediate calls and output calls, respectively. As with all of the internal parameters, be careful about resetting them to save computer time. Make sure that the machine has been run many times first. My experience has shown me that one should set the internal parameters *conservatively* and loosen after many runs rather than set the internal parameters to save computer time and tighten them when problems arise (because that improbable run WILL occur).

Figure 5.3-1a

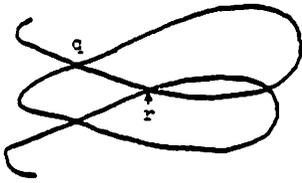


Figure 5.3-1b

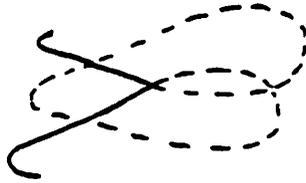


Figure 5.3-1c

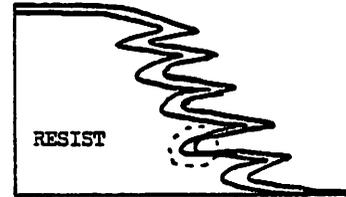


Figure 5.3-2

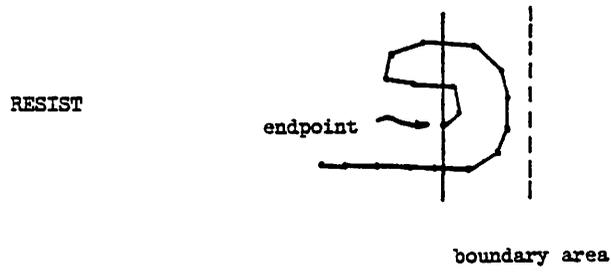


Figure 5.3-3a

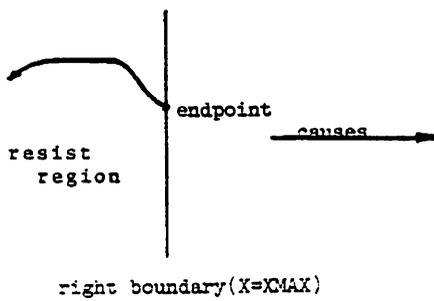
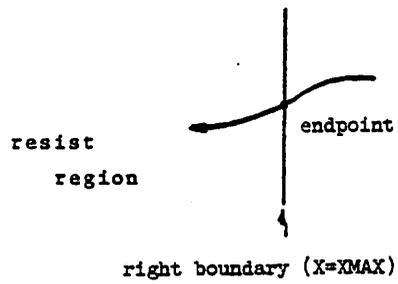


Figure 5.3-3b

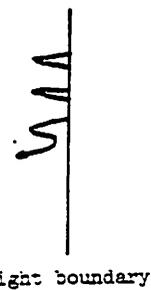


Figure 5.3-3c

Another potentially fatal problem can be described as "boundary area excursions". Figure 5.3-2 shows a string that has developed into the boundary and back out into a high development region. Subroutine BNDARY will not correct this problem once the string has reentered the resist. Currently the number of advances between checks(NADCHK) is 3, which means that points outside of the boundary are searched for every third advance. If every third advance does not seem to be enough, or if NADCHK is raised to save computer time, one might consider calling BNDARY more often.

The slope of the intensity curve should be near zero at the x boundaries. Figures 5.3-3 depict what happens if the slope is significant at the x boundaries. The intensity in figure 5.3-3a causes inaccuracies due to the fact that the higher intensity tends to pull the endpoint to the left. The window gives a downward direction to the endpoint. Figure 5.3-3b shows an intensity pattern that causes the standing waves to develop outside of the boundary, figure 5.3-3c. Eventually the string sections will reenter the resist and cause the program to dump.

NADCHK and NCKOUT are reset in the output after resist breakthrough. With very low contrast intensity patterns, they might be set too low for good accuracy. Furthermore, a high contrast pattern results in wasted computer time due to NCKOUT being set overly high for the slow development rate. Subcontroller DVELOP isn't very smart here.

Problems can occur if the standing wave peaks develop outside of the side boundaries. The string may reflect off of the boundary and reenter the resist. The window should be picked to insure that this condition does not occur.

References for Chapter 5:

- 1) R.E. Jewett, P.I. Hagouel, A.R. Neureuther, and T. Van Duzer, "Line-Profile Resist Development Simulation Techniques", *Polymer Engineering and Science*, vol 14, no 6, pp 381-384, June 1977.
- 2) M.M. O'Toole, "A Brief Report on the Rework of MACHINE4", written communication between SAMPLE group at U.C. Berkeley, dated 6/29/78.

Chapter 6

The Influence of Partial Coherence on Projection Printing

One of the important controllable parameters in projection printing is the degree of illumination coherence¹⁻⁵. Partially coherent mask illumination affects both the image intensity incident on the resist^{6,7,13} and the resulting developed profiles in the resist³. This chapter discusses the effects of partial coherence on projection printing with and without the presence of focus error. Section 6.1 explores the effects of partial coherence on the image. The image of a typical mask pattern under perfect focus conditions is discussed. The effect focus error, a major cause of degradation of the resist image⁸, is also discussed in section 6.1. The image features of edge slope and toe intensity are defined and incorporated into a piecewise linear model of the image intensity. Section 6.2 uses the piecewise linear intensity distribution to determine the relative importance of edge slope and toe intensity in resist linewidth control. Developed resist profiles⁹⁻¹¹ of a typical mask pattern are used to indicate a reasonable operating value for the partial coherence parameter σ --the ratio of the numerical aperture of the condenser system to the numerical aperture of the objective lens $[(NA_o)_{im}]$, figure 6-1.

The masks considered in the following analysis consist of periodic opaque lines in x with no variation in y . The source is a line source along x . For simplicity, the objective lens aperture is assumed square and the illumination light is assumed quasi-monochromatic.

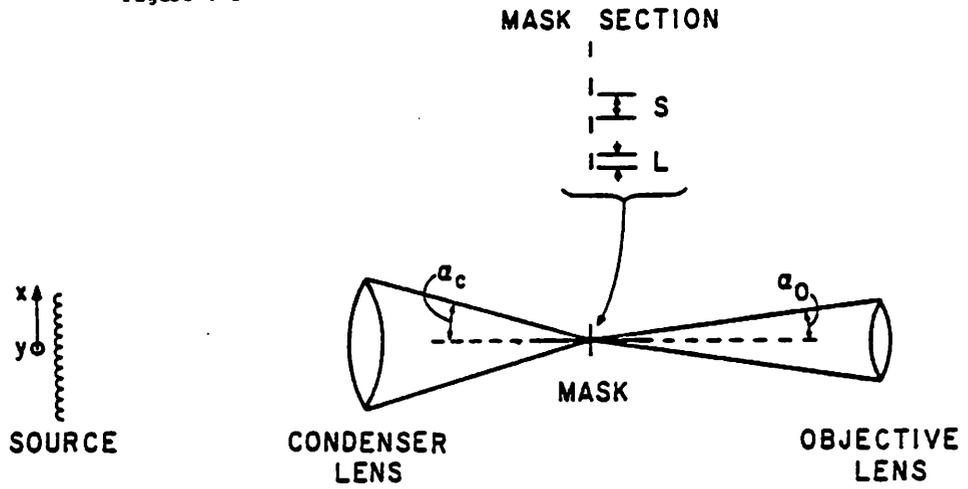
6.1 The Effects of Partial Coherence on the Image

The basic effects of imaging with partially coherent light are seen in figure 6.1-1, which shows the image intensity near the edge of a mask pattern consisting of 10 micron opaque bars interspersed with 10 micron clear spaces. The mask pattern was imaged at $.436\mu$ through a lens with a numerical aperture of $.28$. Figure 6.1-1 closely resembles the pattern produced by a isolated edge⁶. The position of the mask edge occurs at $x=0$; $\sigma = \infty$ indicates incoherent illumination. Three significant variations occur in the image as a function of partial coherence. First, the edge intensity is approximately $.25$ for $\sigma \leq .8$. The edge intensity increases to approximately $.5$ with increasing σ , denoting a shift towards the mask edge for $\sigma > .8$. Second, ringing occurs at the edge for $\sigma < .7$. Third, the intensity of the low portion of the curve increases with increasing σ . Although figure 6.1-1 only shows image intensity distributions for the case of perfect focus, similar variations hold for images with focus error.

Figures 6.1-2 show the effect of focus error on the image of a typical mask pattern consisting of 2μ lines interspersed with 6μ spaces for σ equal to $.3, .5, .7, .9, 1.2,$ and 1.5 . Since the mask is periodic, figures 6.1-2 are reflexive around both the $x = -1\mu$ and $x = 3\mu$ axes. As in the previous figure, the numerical aperture of the imaging lens is $.28$ and the wavelength is $.436\mu$. The focus error for the curves is taken in increments of $.4$ Rayleigh units; one Rayleigh(incoherent) depth of field is equal to $2.78\mu = \lambda/[2(NA_o)^2]$. The focus error d is defined as the distance in microns between the surface of the resist and the plane of perfect focus. Increasing focus error causes a decrease in the edge slope and an increase in the low intensity portion of the image. The low intensity portion of the image also increases with σ for all values of focus error. Figures 6.1-1 and 6.1-2 suggest that the image features of edge slope and the low portion of the image intensity may be useful in discussing how trends in the image affect the resist profiles.

In order to quantitatively discuss trends in the image intensity distribution, precise definitions for edge slope and low intensity value are needed. Edge slope S may simply be defined as the slope of the image at the mask edge, or

Figure 6-1

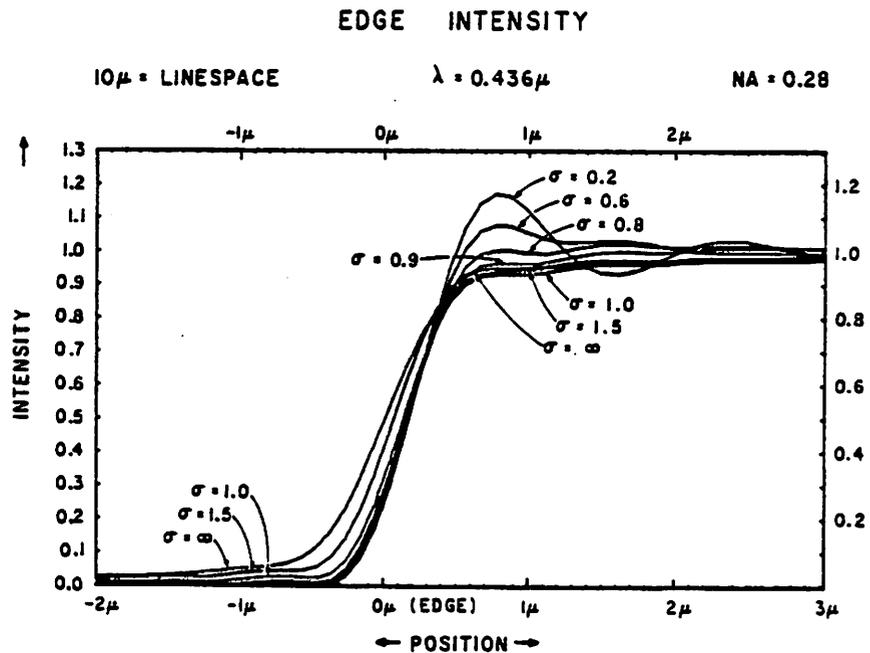


$$(NA)_c = \sin \alpha_c$$

$$(NA)_o = \sin \alpha_o$$

$$\sigma = \frac{(NA)_c}{(NA)_o}$$

Figure 6.1-1



PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 0.3$

2 μm LINE, 6 μm SPACE NA=0.28 $\lambda = 0.436 \mu\text{m}$ IR.u.=2.78 μm
x(0.4) R.u.

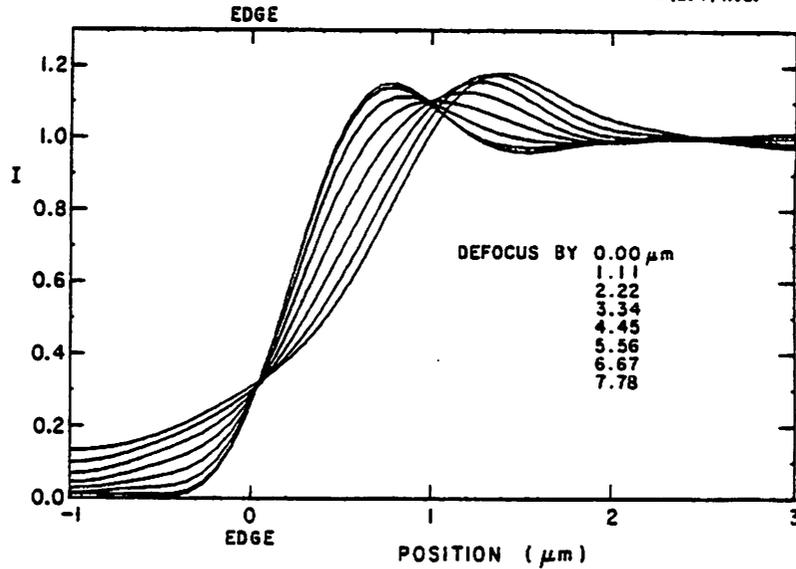


Figure 6.1-2a

PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 0.5$

2 μm LINE, 6 μm SPACE NA=0.28 $\lambda = 0.436 \mu\text{m}$ IR.u.=2.78 μm
x(0.4) R.u.

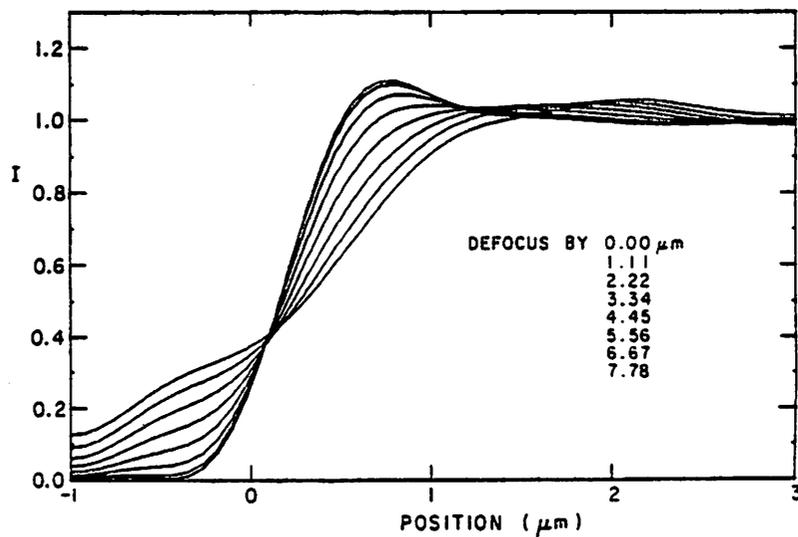


Figure 6.1-2b

PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 0.7$

2 μm LINE, 6 μm SPACE NA=0.28 $\lambda = 0.436 \mu\text{m}$ IR.u.=2.78 μm
z(0.4) R.u.

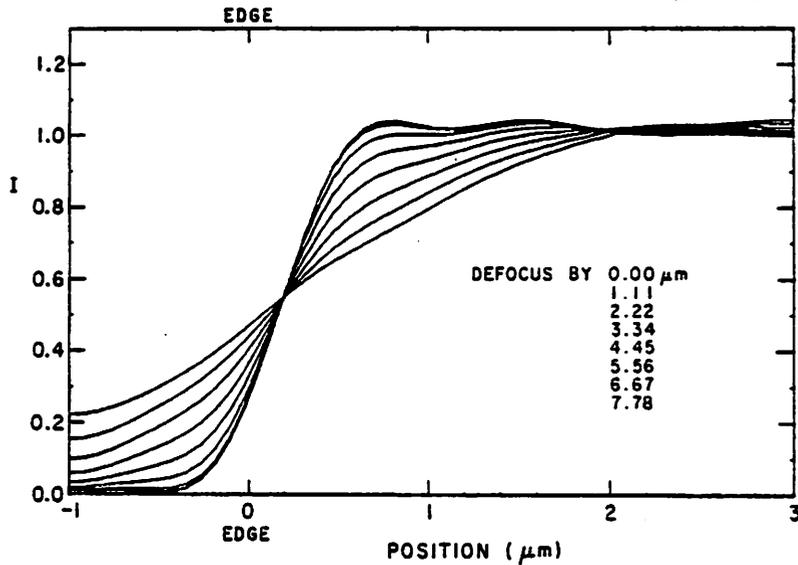


Figure 6.1-2c

PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 0.9$

2 μm LINE, 6 μm SPACE NA=0.28 $\lambda = 0.436 \mu\text{m}$ IR.u.=2.78 μm
z(0.4) R.u.

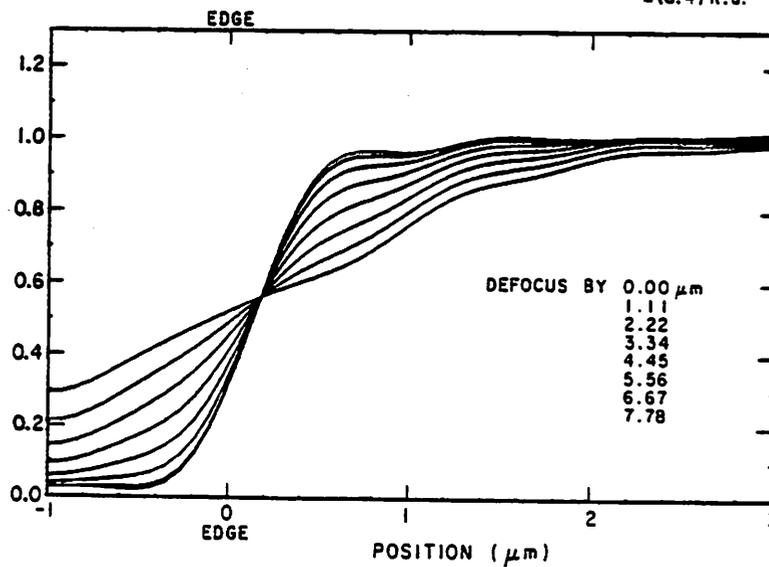


Figure 6.1-2d

PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 1.2$

2 μ m LINE, 6 μ m SPACE NA=0.28 $\lambda = 0.436 \mu$ m IR.u.=2.78 μ m
 $z(0.4)$ R.u.

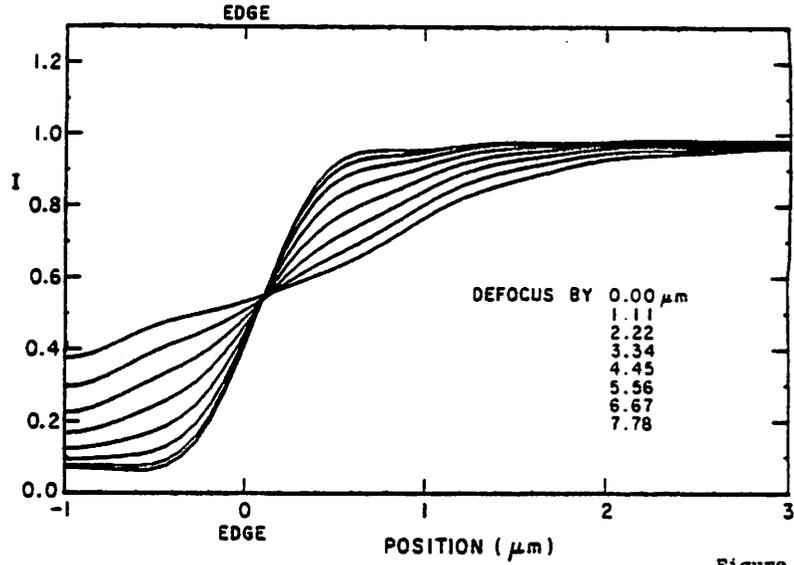


Figure 6.1-2e

PARTIAL COHERENCE WITH DEFOCUS : $\sigma = 1.5$
 EDGE INTENSITY

2 μ LINE, 6 μ SPACE NA = 0.28 $\lambda = 0.436 \mu$ 1 RU = 2.78 μ
 0.4 x RU

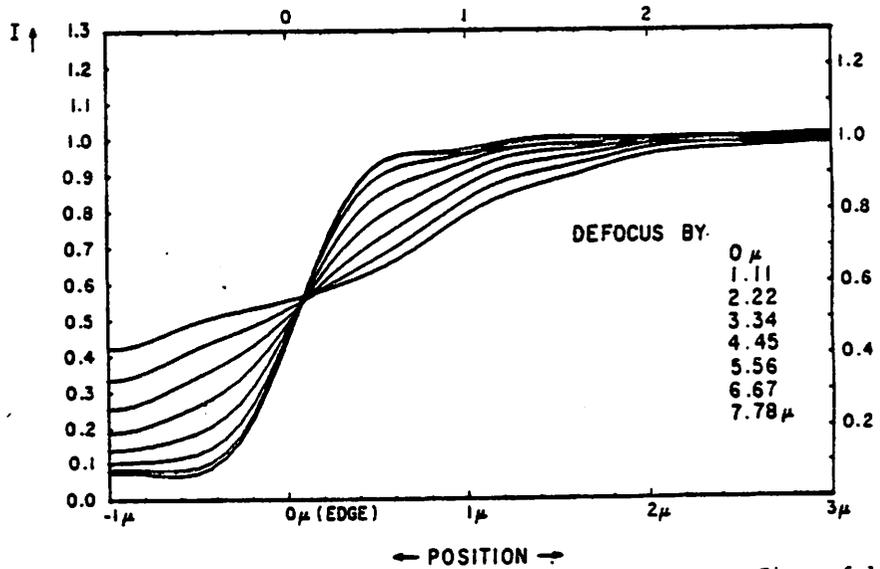


Figure 6.1-2f

$$S = \left. \frac{dI}{dx} \right|_{x=0} \quad (6.1-1)$$

Generally, the slope of the image near the edge does not vary significantly from the slope at the edge. Figure 6.1-3 shows the edge slope of the typical pattern as a function of focus error for a variety of σ 's. The broken vertical lines represent 2.78μ , 5.56μ , and 8.34μ of focus error (or 1, 2, and 3 incoherent depth of fields). Although the edge slope varies significantly with focus error, it is fairly constant as a function of σ . Most mask patterns of interest in projection printing generate similar sets of curves to those of figure 6.1-3, indicating that the edge slope of the image is not significantly improved with the use of partial coherence.

Unlike the edge slope, the low portion of the image intensity distribution does show significant variation with σ . Unfortunately the low portion of the image is not quite as easily defined as edge slope, for a severely defocused image may not have a constant low portion. However, a working definition can be adopted for the low portion of the image intensity, hereafter referred to as the *toe* intensity. Let the toe intensity T be defined as the intensity of the image at a distance from the mask edge of Δx , or

$$T = \left. I \right|_{x=\pm\Delta x} \quad (6.1-2a)$$

where

$$\Delta x = \pm \frac{1}{2} \left[\frac{1.22 \cdot \lambda}{2(NA_o)} \right] \quad (6.1-2b)$$

and where \pm refers to the low intensity side of the mask edge (towards the resist line). Δx is chosen as half the Rayleigh resolution limit for incoherent light.

Figure 6.1-4 shows the toe intensity of the typical pattern as a function of focus error for a variety of σ ' and for a Δx of $-.5\mu$, as defined above. The toe intensity rises rapidly with focus error and is strongly affected by σ . Toe intensities for $\sigma \geq .9$ are significantly greater than those for smaller σ for all cases of focus error. Toe intensities for $\sigma \leq .7$ and $d \leq 4\mu$ are very close, indicating that little is gained by reducing σ below .7 for focus errors less than 4 microns. Reduction of σ results in less light output from the condenser system and, therefore, increases exposure time.

The effects of edge slope and toe intensity can be artificially separated using the piecewise linear approximation of the image intensity distribution shown in figure 6.1-5. The three line segments of the approximation represent the toe, the edge slope, and the high intensity (taken = 1.0) portion of the image intensity distribution. The effect of overshoot is ignored in the approximation, since the development rate usually has saturated for the normalized intensity of 1.0. A comparison of the simple piecewise linear approximation of figure 6.1-5 with figures 6.1-2 shows that the approximation only holds for $d < 4.25\mu$. The piecewise linear image is used in the next section to separate the effects of edge slope and toe intensity on the resist profiles.

6.2 The Effects of Partial Coherence on the Resist Line Edge Profiles

The effect of partial coherence on the image intensity distribution results in a corresponding effect on the developed resist profiles. A typical resist profile generated by SAMPLE is shown in figure 6.2-1. The image of the standard pattern for $\sigma = .6$ and 2.22μ focus error was used to expose .94 microns of resist on .068 microns of oxide on a silicon substrate at an

^{*}A σ of 1.5 gives results close to the incoherent results.

Figure 6.1-3

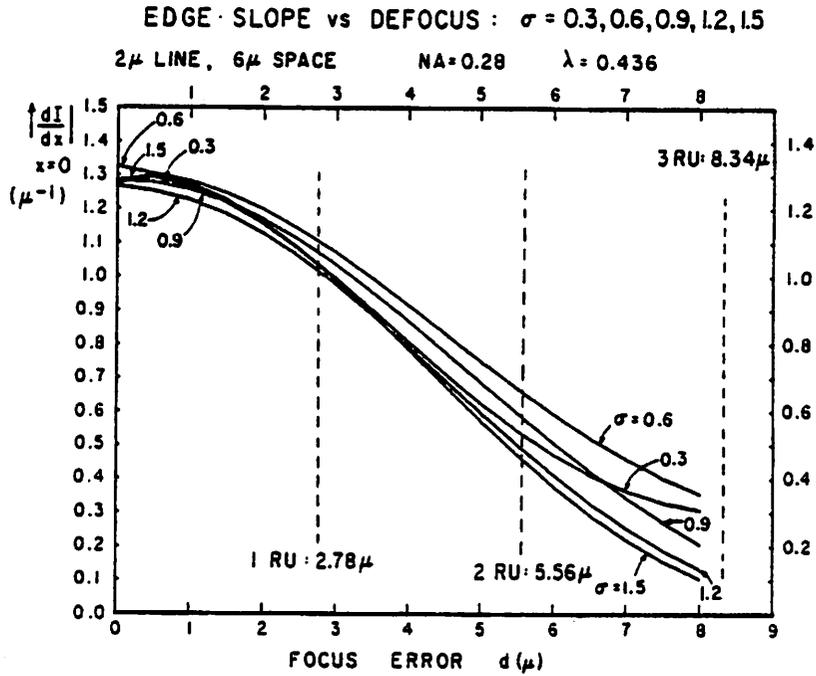


Figure 6.1-4

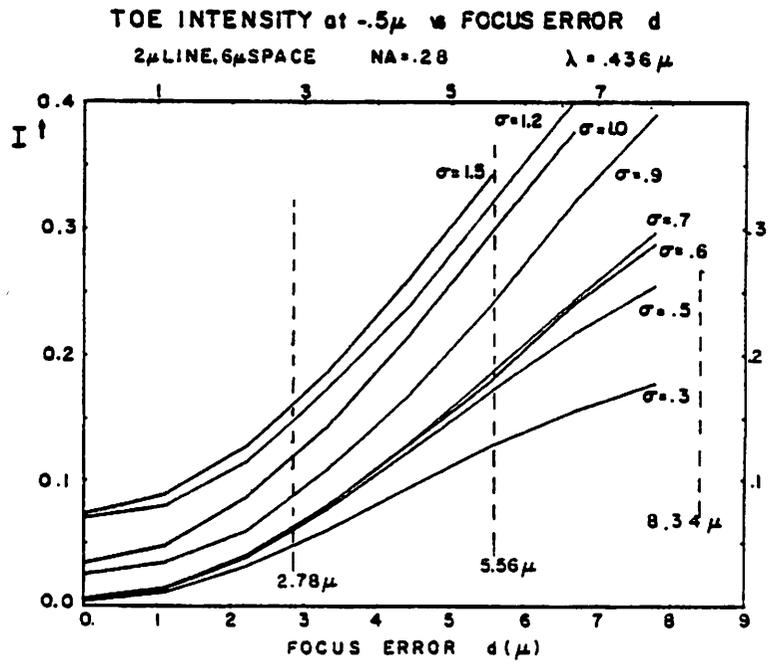
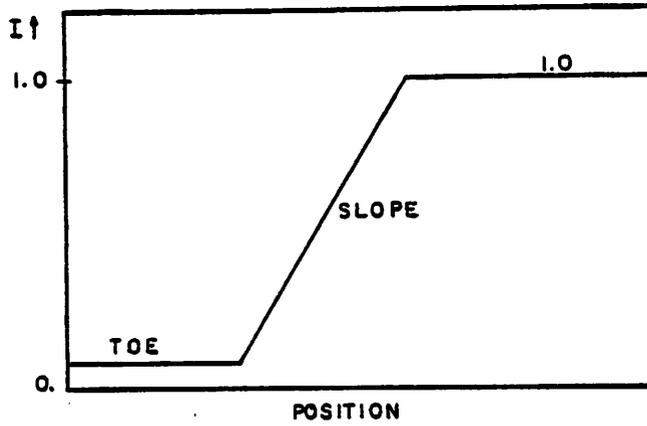


Figure 6.1-5

PIECEWISE LINEAR MODEL for INTENSITY



RESIST PROFILE

2 μ LINE 6 μ SPACE NA = 0.28 $\lambda = 0.436 \mu$

60 sec DEVELOPMENT

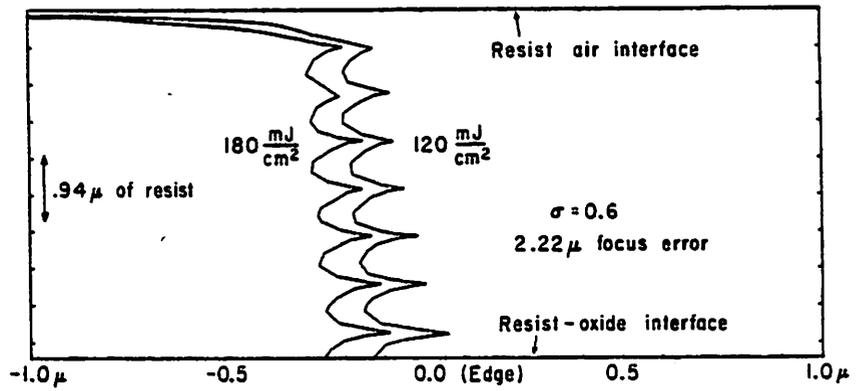


Figure 6.2-1

exposure wavelength of $.436\mu$. The physical configuration results in a near standing wave maximum at the oxide-resist interface. The development model parameters of $A(.551cm^{-1})$, $B(.058cm^{-1})$, $C(.01\frac{cm^2}{mJ})$ and $E_1(5.63)$, $E_2(7.43)$, $E_3(-12.6)$ were taken from reference 10 and from measurements made at UC Berkeley. The linewidth is defined as the width of the line at the resist-oxide interface. Figure 6.2-1 shows the 60 second development curves for exposure energies of $120\frac{mJ}{cm^2}$ and $180\frac{mJ}{cm^2}$. $120\frac{mJ}{cm^2}$ is the energy needed at $.436\mu$ to clear large areas of $.94\mu$ of resist in about 30 seconds, a typical development method. The heavy exposure of $180\frac{mJ}{cm^2}$ was chosen to represent process variations. The linewidth and linewidth control studies that follow use the physical configuration and model parameters described above.

The ability to produce lines of a fixed width under a wide range of processing conditions is referred to as good linewidth control. Small variations in resist or oxide thickness can cause the exposure energy coupled into the resist to vary by about a factor of two¹. Illumination non-uniformity can cause an additional 5-10 percent exposure variation over the optical field. Obviously, large exposure variations can cause substantial changes in the resist linewidth.

The relative effects of edge slope and toe intensity on linewidth control can be somewhat separated using the piecewise linear model for the image discussed in section 6.1 and shown in figure 6.1-5. The half-linewidth variation δ as a function of toe intensity (60 second development) is shown in figure 6.2-2 for edge slopes of (a) $1.3\mu^{-1}$, (b) $1.0\mu^{-1}$, and (c) $.8\mu^{-1}$. From figure 6.1-3, the smallest edge slope of $.8\mu^{-1}$ corresponds to a focus error of 4.25μ --approximately the largest focus error for which the piecewise linear model is a reasonable approximation. The roughness of the curves in figure 6.2-2 is due to the fact that δ is taken at intervals of approximately .15 in toe intensity and to anomalies in the development simulation. The half-linewidth variation is due to an exposure factor of $1\frac{1}{2}$. For toe intensities less than .08, the linewidth variation is independent of toe value and increases inversely with slope. For toe intensities greater than .1, linewidth variation increases rapidly with toe value and is independent of slope. Since partial coherence generally doesn't have a great effect on edge slope, the linewidth variation for images with toe intensities $<.08$ (zero or small focus error) is not controllable with the use of partial coherence. However, the linewidth variation of images with high toe intensities may be significantly reduced with partial coherence.

In order to verify the above ideas, a study of linewidth variation and linewidth as a function of σ was carried out for the typical pattern of 2μ lines interspersed with 6μ spaces; the image intensity distributions and resist profiles were calculated by SAMPLE. Figure 6.2-3 shows the half-linewidth variation δ as a function of σ for the cases of a) perfect focus, b) 2.78μ focus error, and c) 4.17μ focus error. For the perfect focus case reasonable linewidth control is obtained for all values of σ , corresponding to the fact that the toe intensity is smaller than .08 for all σ (figure 6.1-4). For 2.78μ of focus error, curve 6.2-3b shows a significant jump in δ for $\sigma \geq .9$. Correspondingly, figure 6.1-4 shows a toe intensity greater than .1 for $\sigma > .9$ and 2.78μ focus error. For 4.17μ of focus error, figure 6.1-4 shows a toe intensity equal to approximately .1 for σ between .3 and .7. For $\sigma \geq .7$ the toe intensity increases abruptly, reflected in figure 6.2-3c by the jump in δ at $\sigma = .8$. Figure 6.2-3 demonstrates that changes in linewidth control are highly correlated with the toe intensity of the image pattern.

In addition to indicating changes in linewidth control, the toe intensity of the image can also be used to predict the behavior of the resist linewidth. A low toe intensity in the image indicates a low development rate in the corresponding position in the resist, whereas a high toe intensity indicates a somewhat higher development rate. Once the resist has developed into the region of the toe, the loss of linewidth with further development depends on the intensity of the image in the toe region. Figure 6.2-4 shows the half-linewidth W for the typical pattern versus σ for the cases of a) perfect focus with an exposure energy of $120\frac{mJ}{cm^2}$, b) 2.78μ focus

Figure 6.2-2

1/2 LINEWIDTH VARIATION δ vs. TOE INTENSITY
 PIECEWISE LINEAR MODEL, 60sec DEVELOPMENT

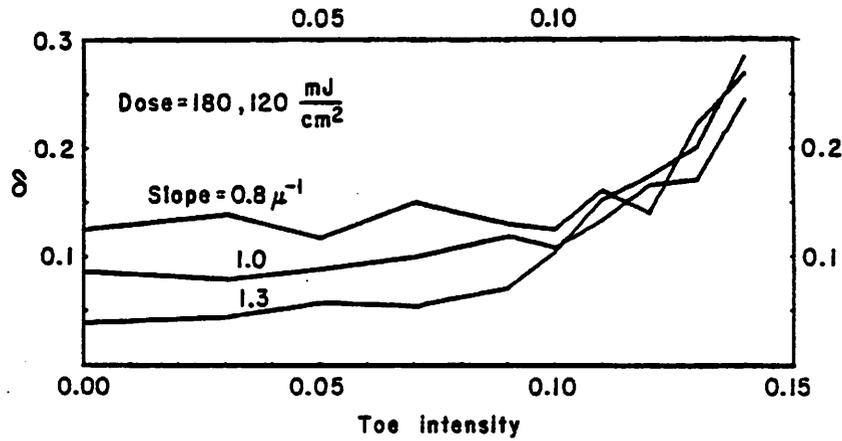


Figure 6.2-3

1/2 LINEWIDTH VARIATION δ vs. σ
 2 μ LINE 6 μ SPACE NA = 0.28 $\lambda = 0.436 \mu$

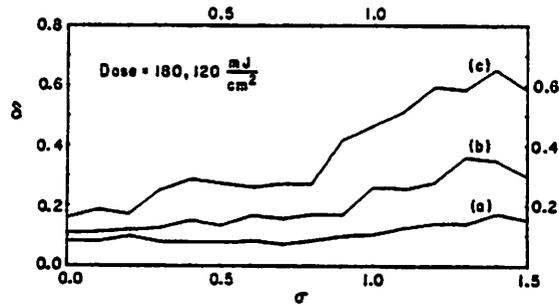
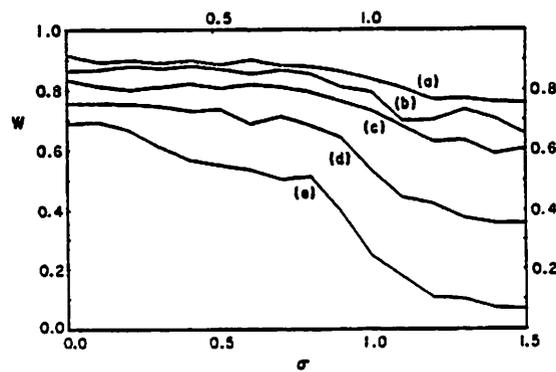


Figure 6.2-4

1/2 LINEWIDTH W vs. σ
 2 μ LINE 6 μ SPACE NA = 0.28 $\lambda = 0.436 \mu$



error at $120 \frac{mJ}{cm^2}$, c) perfect focus at $180 \frac{mJ}{cm^2}$, d) 2.78μ focus error at $180 \frac{mJ}{cm^2}$, and e) 4.17μ focus error at $180 \frac{mJ}{cm^2}$. The heavy exposure ($180 \frac{mJ}{cm^2}$) curves show severe linewidth degradation for σ greater than .75. From figure 6.1-4, a σ of .75 corresponds to a toe intensity near .08 for 2.78μ of focus error and a toe intensity of .12 for 4.17μ of focus error. The behavior of the linewidth curves in figure 6.2-4 roughly follow the predictions of figure 6.1-4 using the toe intensity criterion discussed previously. The relatively stable linewidth for $\sigma \leq .75$ indicates a greater tolerance to focus error with low σ values than with high σ values.

Figures 6.2-5 demonstrate the increased linewidth control under extreme exposure variation for a σ of .6 versus a σ of 1.5. Note that the resist linewidth is closer to the mask linewidth for $\sigma = .6$. Figures 6.2-6 demonstrate the increased tolerance to focus error with lower σ values. A σ of .6 with 4.25μ of focus error gives approximately the same results as a σ of 1.5 with 2μ of focus error.

6.3 Further Considerations

The result that σ values less than .7 offer little advantage with respect to linewidth control holds for mask linewidths larger than about $1 \frac{1}{4}\mu$ (for $\lambda = .436\mu$ and $NA = .28$). For linewidths around 1 micron, increased linewidth control¹² can be achieved with a σ between .2 and .3. The increased linewidth control is due to two factors. First, for linewidths close to the resolution limit of the objective lens, the toe values will not be zero--even for low σ . Second, the edge ringing concurrent with very low σ increases the contrast of the image intensity. Figures 6.3-1 show the intensity distribution for a 1 micron line and space for $\sigma = .6$ and .3. The increased contrast at $\sigma = .3$ is evident. Note, however, that the tolerance to focus error is *less* for $\sigma = .3$ than for $\sigma = .6$.

Multiple wavelength exposure also changes the results somewhat. First, for most oxide and resist thicknesses, the change of energy coupled into the resist is not as great as it is with single wavelength exposure. Frequently, the coupling-out of one wavelength with resist thickness variation offset by the coupling-in of another wavelength. Second, the linewidths are closer to the mask linewidths with multiple wavelength exposure. A σ of .7 is still a reasonable operating point, however.

6.4 Conclusions

Partially coherent mask illumination reduces the sensitivity of resist linewidth to exposure variations, primarily by suppressing the toe value of the image intensity distribution. The low toe intensities concurrent with smaller σ result in a resist linewidth closer to the mask linewidth than with larger σ (i.e., increased tolerance to focus error). A σ of .7 has been shown to maintain reasonably low toe intensities under focus error. A σ less than .7 reduces the light output from the condenser system without a significant increase in linewidth control for most patterns of interest.

¹² σ values below .15 cause diffraction rings in the image due to dust particles on the mask or optics.

Figure 6.2-5a

RESIST PROFILES

2 μ LINE, 6 μ SPACE NA = 0.28 λ = 0.436 μ 0 FOCUS ERROR

60 sec DEVELOPMENT CURVES

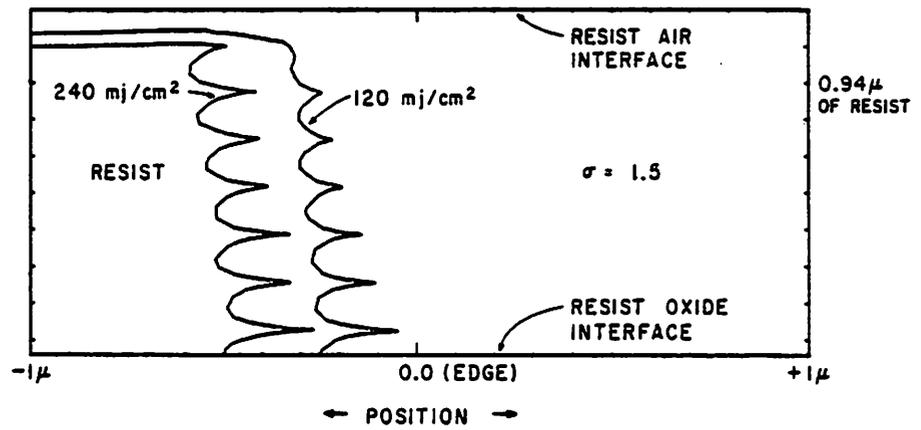
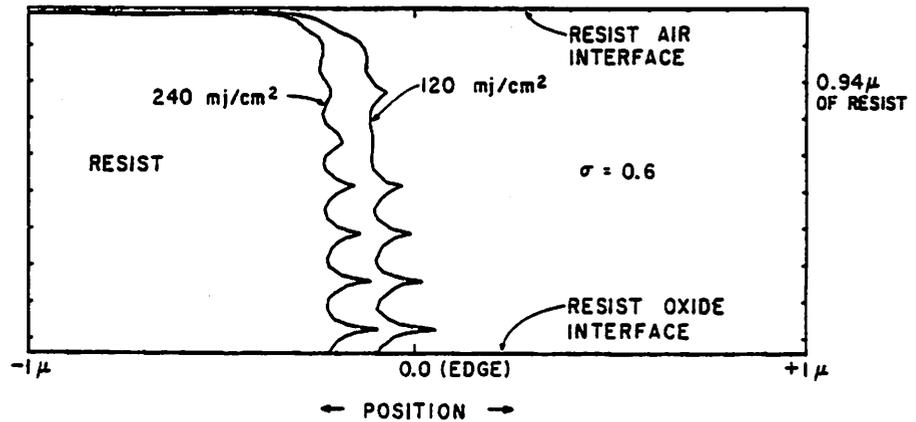


Figure 6.2-5b

RESIST PROFILES

2 μ LINE, 6 μ SPACE NA = 0.28 λ = 0.436 μ 0 FOCUS ERROR

60 sec DEVELOPMENT CURVES



RESIST PROFILES

2 μ LINE, 6 μ SPACE NA = 0.28 $\lambda = 0.436$ 0.94 μ RESIST

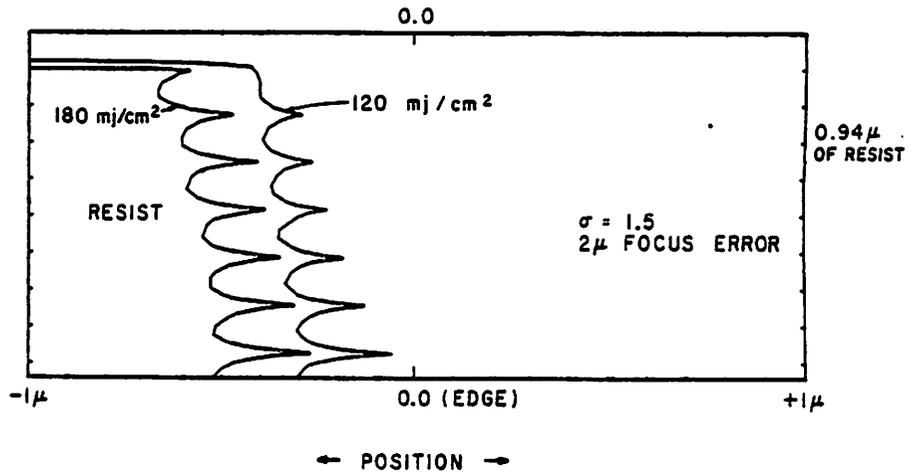


figure 6.2-6a

RESIST PROFILES

2 μ LINE, 6 μ SPACE NA = 0.28 $\lambda = 0.436$ 0.94 μ RESIST

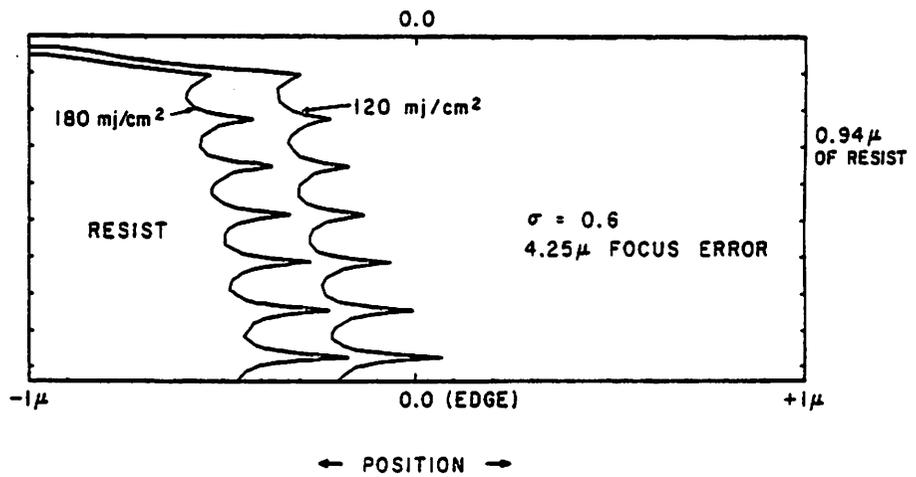
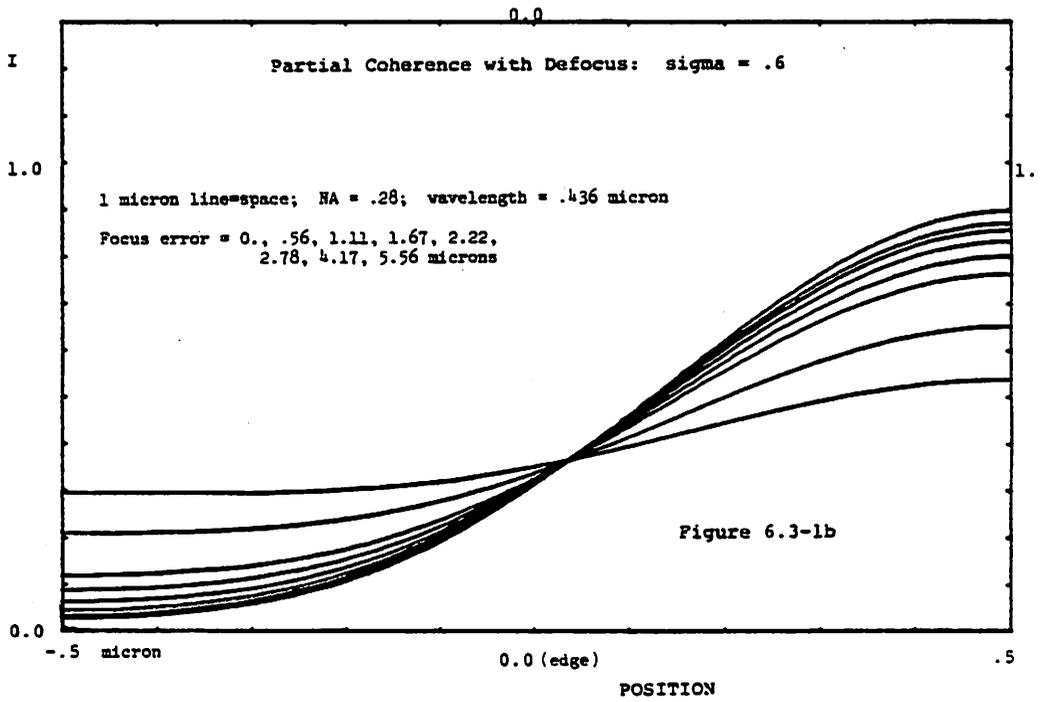
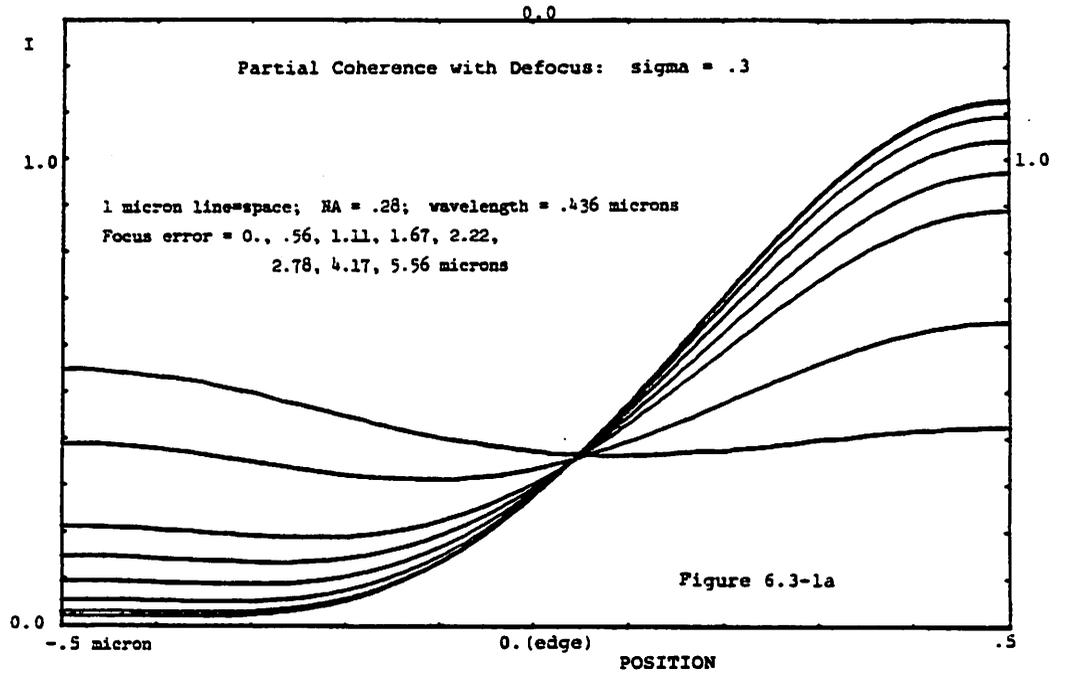


Figure 6.2-6b



References for Chapter 6:

- 1) J.D. Cuthbert, "Optical Projection Printing", *Solid State Technology*, Aug '77, p 59.
- 2) R. Hershel, "Partial Coherence in Projection Printing", SPIE vol. 135, *Developments in Semiconductor Microlithography III*, 1978.
- 3) Ron Hershel, "Effects of Partially Coherent Imagery on Resist Profiles", *Kodak Microelectronics Seminar, Interface '78*, Oct 1-3 '78, San Diego, Ca.
- 4) A.R. Neureuther, M. O'Toole, W.G. Oldham, and S. Nandgaonkar, "Use of Simulation to Optimize Projection Printing Profiles", *153rd meeting of the Electrochemical Society*, Seattle, Wash., May 21-26 1978.
- 5) W.G. Oldham, S. Nandgaonkar, A.R. Neureuther, and M. O'Toole, "A General Simulator for VLSI Lithography and Etching Processes, Part 1--Application to Projection Lithography", *Trans. on Electron Devices*, Apr '79.
- 6) R.E. Kinzly, "Investigations of the Influence of the Degree of Coherence upon Images of Edge Objects", *JOSA*, vol 55, no. 8, Aug '65, p 1002.
- 7) D. Nyyssonen, "Linewidth Measurement with an Optical Microscope: the Effect of Operating Conditions on the Image Profile", *Appl. Opt.*, vol 16, no. 8, Aug '77, p 2223.
- 8) M.A. Narasimham and J.H. Carter, Jr., "Effects of Defocus on Photo-lithographic Images Obtained with Projection-Printing Systems", SPIE vol 135, *Developments in Semiconductor Microlithography III*, 1978.
- 9) F.H. Dill, A.R. Neureuther, J.A. Tuttle, and E.J. Walker, "Modelling Projection Printing of Positive Photoresists", IBM Research, February 2, 1975, RC 5261.
- 10) F.H. Dill, W.P. Hornberger, P.S. Hauge, and J.M Shaw, "Characterization of Positive Photoresist", *IEEE Transactions on Electron Devices*, vol ED-22, no. 7, July '75, p 445.
- 11) R.E. Jewett, P.I. Hagouel, A.R. Neureuther, and T. Van Duzer, "Line-Profile Resist Development Simulation Techniques", *Polymer Engineering and Science*, vol. 14, no. 6, pp. 381-384, June 1977.
- 12) M. Lacombat and G.M. Dubroeuq, "Coherent Illumination Improves Step and Repeat Printing on Wafers", SPIE *Developments in Semiconductor Microlithography IV*, San Jose, Ca., April 1979.
- 13) Bern J. Lin, "Deep-UV Lithography", short course on the Fundamentals of High-Resolution Lithography, University Extension at UC Berkeley, San Francisco, Ca., February 23, 1979.

Appendix A

Conventional Partial Coherence Theory

A.1 Review of Partial Coherence Theory

Imaging with partially coherent light was investigated by Hopkins⁷⁻¹⁰ in the 1950's. The theory has been applied and simplified by a number of other authors¹¹⁻²¹; and several books¹⁻⁴ (or book chapters) have been published on the subject.

A.1.1 Some Definitions

The key function in the theory of partial coherence is the *mutual coherence function*, $\Gamma_{12}(\tau)$, defined by

$$\Gamma_{12}(\tau) = \Gamma(x_1, x_2, \tau) = \langle U^*(x_1, t) U(x_2, t + \tau) \rangle \quad (\text{A-1})$$

where $U(x, t)$ is the analytic signal associated with a cartesian component of the electric field at point x and time t , where τ is the time delay, and where $\langle \rangle$ denotes the time average

$$\langle f(t) \rangle \equiv \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t) dt \quad (\text{A-2})$$

The normalized mutual coherence function $\gamma_{12}(\tau)$ is defined as

$$\gamma_{12}(\tau) = \frac{\Gamma_{12}(\tau)}{\sqrt{\Gamma_{11}(0)\Gamma_{22}(0)}} \quad (\text{A-3})$$

and is known as the *complex degree of coherence*.

$\gamma_{12}(\tau)$ represents the correlation between the electric field component at point x_1 and the same component at point x_2 a time τ later. The mutual coherence function follows a pair of wave equations

$$\nabla_i^2 \Gamma_{12}(\tau) = \frac{1}{c^2} \cdot \frac{\partial^2 \Gamma_{12}(\tau)}{\partial \tau^2} \quad i=1, 2 \quad (\text{A-4})$$

The intensity can be found from $\Gamma_{12}(\tau)$ via

$$I(x) = \Gamma(x_1, x_1, 0) \quad (\text{A-5})$$

The mutual coherence function can be further simplified under the conditions for quasi-monochromatic light

$$\Delta\nu \ll \bar{\nu} \quad (\text{A-6a})$$

$$\Delta\nu \ll \frac{1}{|\tau|} \quad (\text{A-6b})$$

where $\Delta\nu$ is the spectral width and $\bar{\nu}$ is the mean frequency. The second condition simply states that the loss of correlation of a field with itself in time τ , $\Gamma_{11}(\tau)$, due to differences in the component temporal frequencies (color) of the field is not an important consideration in the present problem. Under quasi-monochromatic conditions, $\Gamma_{12}(\tau)$ can be written as

$$\Gamma_{12}(\tau) \approx J_{12} e^{-i2\pi\bar{\nu}\tau} \quad (\text{A-7})$$

where $J_{12} = \Gamma_{12}(0)$ and where J_{12} is known as the *mutual intensity* of points x_1 and x_2 . The mutual intensity also follows a pair of wave equations.

$$\nabla_i^2 J_{12} + k^2 J_{12} = 0 \quad i=1, 2 \quad (\text{A-8})$$

The normalized mutual intensity is known as the *complex coherence factor* μ_{12}

$$\mu_{12} = \frac{J_{12}}{\sqrt{J_{11}J_{22}}} \quad (\text{A-9})$$

The complex coherence factor is equal to the visibility V of the fringe pattern in a Young's experiment if the pinholes are illuminated with the same intensity. In general,

$$\mu_{12} = \left[\frac{I_1 + I_2}{2\sqrt{I_1 + I_2}} \right] \cdot V \quad (\text{A-10})$$

where I_1 and I_2 are the intensities at pinholes P_1 and P_2 .

If the mutual intensity is known in one plane, Σ_1 , then it is known in any other plane, Σ_2 . Figure A-1 shows the physical situation.

$$J(x_1, x_2) = \int_{\xi} \int_{\xi} \int_{\xi} J(\xi_1, \xi_2) e^{-i2\pi \frac{r_2 - r_1}{\lambda}} \frac{\Psi_1(\theta_1)}{\lambda r_1} \frac{\Psi_2(\theta_2)}{\lambda r_2} d\xi_1 d\xi_2 \quad (\text{A-11})$$

where the Ψ 's are the appropriate inclination factors.

A.1.2 The van Cittert-Zernike Theorem

The van Cittert-Zernike Theorem relates the mutual intensity in the (x, y) plane to the intensity of the source in the Σ plane. Figure A-2 shows the physical situation. For a quasi-monochromatic incoherent source, the mutual intensity J_{12} is given by

$$J(P_1, P_2) = \kappa I(P_1) \delta(|P_1 - P_2|) \quad (\text{A-12})$$

Using (A-12) in (A-11), and making the assumptions that the extent of the source and observation region are much less than the distance z separating them and that $\Psi(\theta_i) \approx 1$, the complex coherence factor in the observation plane becomes

$$\mu(x_1, y_1; x_2, y_2) = e^{-i\Omega} \frac{\int_{\Sigma} I(\xi, \eta) e^{\frac{i2\pi}{\lambda z} (\Delta x \xi + \Delta y \eta)} d\xi d\eta}{\int_{\Sigma} I(\xi, \eta) d\xi d\eta} \quad (\text{A-13})$$

$$\text{where } \Omega = \frac{\pi}{\lambda z} \left[(x_2^2 + y_2^2) - (x_1^2 + y_1^2) \right] = \frac{\pi}{\lambda z} (\rho_2^2 - \rho_1^2) \quad (\text{A-13a})$$

and

$$\Delta x = x_2 - x_1 \quad \Delta y = y_2 - y_1 \quad (\text{A-13b})$$

The integral in the numerator of equation (A-13) is the Fourier transform of the source intensity with

$$\nu_x = \frac{\Delta x}{\lambda z} \quad \nu_y = \frac{\Delta y}{\lambda z}$$

If the source is uniform in intensity, μ represents the visibility (contrast) V of the fringe pattern produced in the primed plane, z_2 to the right of the observation plane, by two pinholes in the observation plane $(\Delta x, \Delta y)$ apart.

$$I_{z_2} = I_1 + I_2 + 2I_1 I_2 |\mu_{12}| \cos \left[\frac{2\pi}{\lambda z_2} (\Delta x x' + \Delta y y') - \Omega \right] \quad (\text{A-14})$$

Note that, although we have started with an incoherent source, points (x_1, y_1) and (x_2, y_2) in the observation plane are correlated to each other. The correlation (or fringe contrast) varies with

For a uniform circular source of radius a , μ_{12} is proportional to a first order Bessel function. The distance in the observation plane between the first zeros of the Bessel function is $1.22 \frac{\lambda z}{a}$, the Rayleigh resolution criterion.

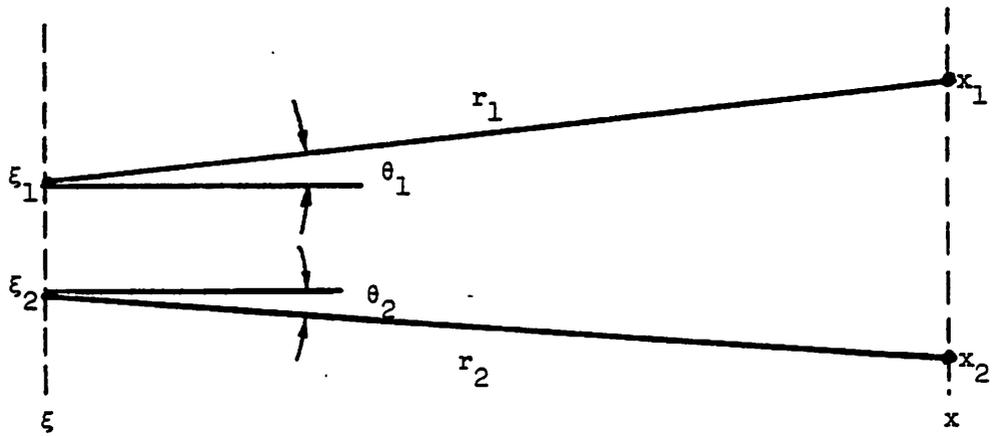
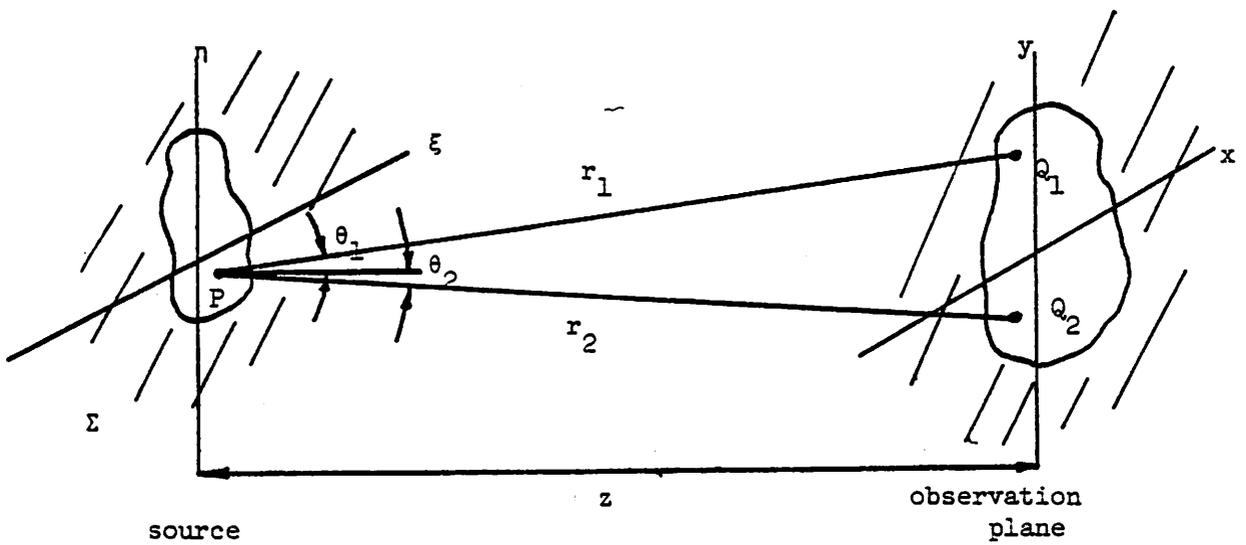


Figure A-1

Figure A-2



the spacing between the points in the observation region. For spacings $\gg 1.22 \frac{\bar{\lambda}z}{a}$, the correlation is very near zero. A large spacing between (x_1, y_1) and (x_2, y_2) allow the points to resolve many "coherence areas" on the source. Each coherence area produces its own fringe pattern in the primed plane, and the sum of many patterns due to many coherence areas averages out. A small spacing that cannot resolve the source into several units gives a high correlation between the points. The source *appears* coherent to the pinholes, and a pattern with visibility ≈ 1 is created in the primed plane. The size of the source determines the number of coherence areas resolved, and thus the fringe visibility due to the pinholes is affected by the size of the source.

A.1.3 Schnell's Theorem

Schnell's Theorem relates the intensity distribution $I(x, y)$ in an observation plane to the mutual intensity in the Σ plane, figure A-2. The Σ plane is assumed to consist of an aperture described by $P(\xi, \eta)$, which may contain a complex valued amplitude transmittance function. The quasi-monochromatic light incident on the aperture is assumed to have a mutual intensity function of the form

$$J(\xi_1, \eta_1; \xi_2, \eta_2) = I_0 \mu(\Delta\xi, \Delta\eta) \quad (\text{A-15})$$

If

$$z > \frac{2D_c d_{coh}}{\lambda} \quad (\text{A-16})$$

where d_{coh} represents the maximum diameter of a coherence area in the aperture and where D_c is the diameter of the aperture, then

$$I(x, y) \approx \frac{I_0}{\lambda z^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\Delta\xi, \Delta\eta) \mu(\Delta\xi, \Delta\eta) e^{i \frac{2\pi}{\lambda z} (x\Delta\xi + y\Delta\eta)} d\Delta\xi d\Delta\eta \quad (\text{A-17})$$

The intensity in the (x, y) plane is the two dimensional Fourier transform of the product of the functions P and μ , or I is the convolution of the transform of P with the transform of μ . Note that, if the coherence area of the wave incident on the aperture is very small in comparison to the area of the aperture, the intensity in the (x, y) plane is similar to the $I(x, y)$ that would be generated by an incoherent source filling the pupil P in the Σ plane. The requirement (A-16) can be relaxed with the appropriate use of a positive lens in the Σ plane.

The condenser system in figure A-3 demonstrates the effects of partial coherence on the intensity $I(x, y)$ in the observation plane. The partial coherence effects originate from the size of the source, the partial coherence of the source (usually considered incoherent), and the diameter of the condenser lens. The size of the source and the distance z_0 determine the expression for μ in equation (A-17). The diameter of the condenser lens determines the expression for P in (A-17). For large condenser lenses P may be considered unity, and the effects of partial coherence in the observation plane is dependent only on the source size and the z distance. For a Köhler condenser system ($z_0 = f$), the phase contribution of the lens to the expression P simply removes the requirement (A-16).

A.1.4 Method for Calculating the Image Intensity

There are several methods for calculating the image intensity. A good review of the methods can be found in Goodman¹. The analysis here follows Goodman.

Although not the most common method, integration over the source will be reviewed here, since it bears the closest resemblance to the method used in SAMPLE. Figure A-4 shows the physical situation. Quasi-monochromatic conditions are assumed to prevail, and the source is assumed incoherent point by point. $F(\xi, \eta; \alpha, \beta)$ and $K(u, v; \xi, \eta)$ represent the amplitude

¹An imaging system consisting of a single lens, an object of maximum dimension L_o a distance z_o from the lens, and an image of maximum dimension L_i a distance z_i from the lens must satisfy the relation

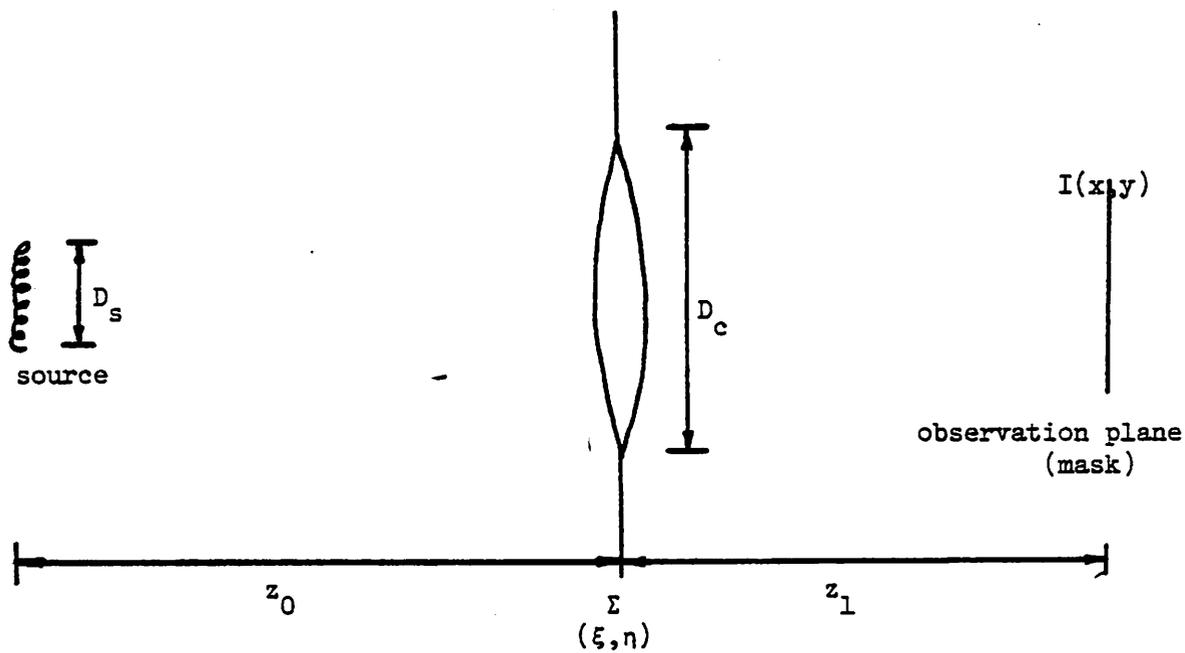


Figure A-3

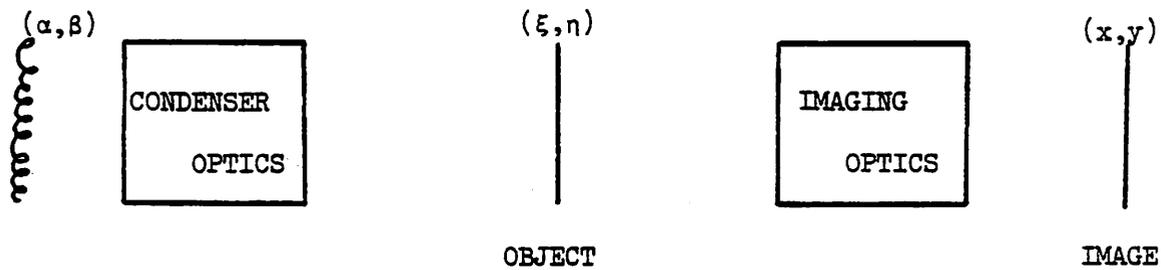


Figure A-4

spread functions of the illuminating and the imaging systems, respectively. Each point on the source produces an image in the (u, v) plane.

The source point (α, β) emits light represented by the time-varying phasor $A_s(\alpha, \beta; t)$. It then illuminates an object of transmittance $t_o(\xi, \eta)$ and produces a field amplitude A_o to the right of the object.

$$A_o(\xi, \eta; \alpha, \beta; t) = F(\xi, \eta; \alpha, \beta) t_o(\xi, \eta) A_s(\alpha, \beta; t - \delta_1) \quad (\text{A-18})$$

where δ_1 is a time delay that depends on (ξ, η) and (α, β) . The field amplitude A_{im} at the image point (u, v) due to the source point (α, β) is given by

$$A_{im}(u, v; \alpha, \beta; t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(u, v; \xi, \eta) t_o(\xi, \eta) F(\xi, \eta; \alpha, \beta) A_s(\alpha, \beta; t - \delta_1 - \delta_2) d\xi d\eta \quad (\text{A-19})$$

where δ_2 is a time delay that depends on (u, v) and (α, β) . The intensity at (u, v) due to source point (α, β) is given by

$$\begin{aligned} I_{im} &= \langle |A_{im}|^2 \rangle \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(u, v; \xi_1, \eta_1) K^*(u, v; \xi_2, \eta_2) \\ &\quad \cdot t_o(\xi_1, \eta_1) t_o^*(\xi_2, \eta_2) F(\xi_1, \eta_1; \alpha, \beta) F^*(\xi_2, \eta_2; \alpha, \beta) \\ &\quad \cdot \langle A_s(\alpha, \beta; t - \delta_1 - \delta_2) A_s^*(\alpha, \beta; t - \delta_1 - \delta_2) \rangle d\xi_1 d\eta_1 d\xi_2 d\eta_2 \end{aligned} \quad (\text{A-20})$$

Under quasi-monochromatic conditions,

$$\langle A_s(\alpha, \beta; t - \delta_1 - \delta_2) A_s^*(\alpha, \beta; t - \delta_1 - \delta_2) \rangle = I_s(\alpha, \beta) \quad (\text{A-21})$$

where $I_s(\alpha, \beta)$ is the intensity of the source at (α, β) . Integrating over the source points gives the final expression for image intensity.

$$\begin{aligned} I_{im}(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_s(\alpha, \beta) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(u, v; \xi_1, \eta_1) K^*(u, v; \xi_2, \eta_2) \\ &\quad \cdot F(\xi_1, \eta_1; \alpha, \beta) F^*(\xi_2, \eta_2; \alpha, \beta) t_o(\xi_1, \eta_1) t_o^*(\xi_2, \eta_2) d\xi_1 d\eta_1 d\xi_2 d\eta_2 d\alpha d\beta \end{aligned} \quad (\text{A-22})$$

A.2 Further discussion of Imaging with Partial Coherence

A more common method of calculating the image intensity uses the mutual intensity function to describe the illumination incident on the object. Frequently, the left part of the condenser system of figure A-5a is replaced by the mutual intensity function

$$J_l(x_1, y_1; x_2, y_2) = \frac{\kappa}{(\lambda z_l)^2} |P_c(x_1, y_1)|^2 \delta(x_1 - x_2, y_1 - y_2) \quad (\text{A-23})$$

of figure A-5b. Equation (A-23) states that the incoherent light of the source remains essentially incoherent while travelling the distance z_l . The source and condenser lens are replaced by

$$\frac{L_o^2}{4z_o} + \frac{L_l^2}{4z_l} \ll l_c$$

where l_c is the coherence length of the light, for quasi-monochromatic conditions to hold.

The approximation is valid if P_c is much greater than the coherence area of the light in the plane (x, y) .

Figure A-5a

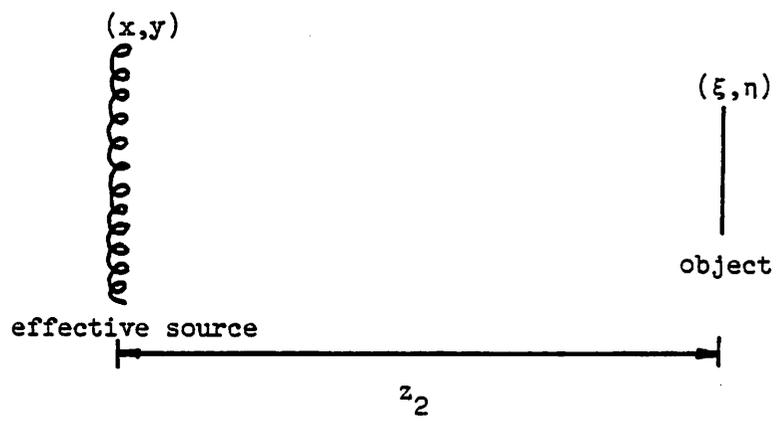
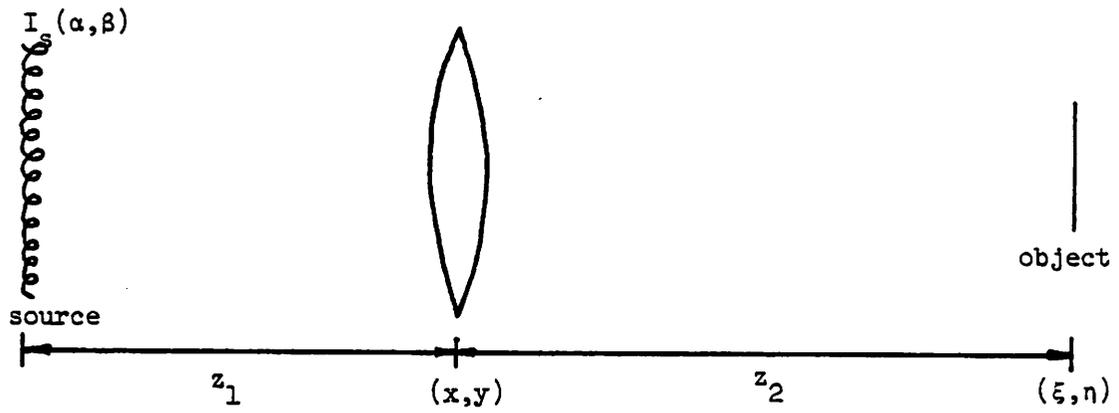


Figure A-5b

an effective source with mutual intensity J_i in the (x, y) plane. P_c represents the pupil function of the condenser lens, and δ is the Dirac delta function.

The mutual intensity incident on the object, J_o , is calculated using the van Cittert-Zernike theorem with the effective source.

$$J_o(\xi_1, \eta_1; \xi_2, \eta_2) = \frac{\kappa' e^{-i\frac{\pi}{\lambda z_2}[(\xi_1^2 + \eta_1^2) - (\xi_2^2 + \eta_2^2)]}}{(\lambda z_2)^2} \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P_c(x_1, y_1)|^2 e^{i\frac{2\pi}{\lambda z_2}(\Delta \xi x_1 + \Delta \eta y_1)} dx_1 dy_1 \quad (\text{A-24})$$

Phase aberrations of the condenser lens can be incorporated into the pupil function as

$$P_c'(x_1, y_1) = P_c e^{-iW(x, y)} \quad (\text{A-25})$$

Since J_o depends on $|P_c'|^2$, the mutual intensity incident on the object is independent of any aberrations that exist in the condenser lens.

The image mutual intensity can then be determined from the relation of the object plane mutual intensity to the image plane mutual intensity, given by

$$J_i(u_1, v_1; u_2, v_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J_o'(\xi_1, \eta_1; \xi_2, \eta_2) \cdot K(u_1, v_1; \xi_1, \eta_1) K^*(u_2, v_2; \xi_2, \eta_2) d\xi_1 d\eta_1 d\xi_2 d\eta_2 \quad (\text{A-26})$$

where J_o' is the mutual intensity of the object after transillumination.

$$J_o'(\xi_1, \eta_1; \xi_2, \eta_2) = t_o(\xi_1, \eta_1) t_o^*(\xi_2, \eta_2) J_o(\xi_1, \eta_1; \xi_2, \eta_2) \quad (\text{A-27})$$

where J_o is the mutual intensity *incident* on the object plane and t_o is the object transmittance.

Combining equations (A-26) and (A-27) and letting $(u_1, v_1) = (u_2, v_2) = (u, v)$ gives the image intensity $I_i(u, v)$.

$$I_i(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(u, v; \xi_1, \eta_1) K^*(u, v; \xi_2, \eta_2) \cdot t_o(\xi_1, \eta_1) t_o^*(\xi_2, \eta_2) J_o(\xi_1, \eta_1; \xi_2, \eta_2) d\xi_1 d\eta_1 d\xi_2 d\eta_2 \quad (\text{A-28})$$

A.2.1 Linear Systems Approach

Equation (A-26) implies that the mutual intensity in the image plane and the mutual intensity in the object plane can be related using a linear systems approach. The expression KK^* in equation (A-26) is the impulse response of the four dimensional linear system. KK^* is the mutual intensity observed at the image coordinates $(u_1, v_1; u_2, v_2)$ due to an object mutual intensity consisting of an impulse at coordinates $(\xi_1, \eta_1; \xi_2, \eta_2)$. The object plane is described by the variables (ξ, η) , and the image plane is described by the variables (u, v) . If the object coordinates have been normalized such that the magnification between the object and image planes is unity, if the effects of image inversion have been removed from the mathematics, and if K is devoid of space-variant phase factors, the linear system is termed *isoplanatic* or *space invariant*. Equation (A-26) can then be written as a convolution integral of the form

$$J_i(u_1, v_1; u_2, v_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J_o'(\xi_1, \eta_1; \xi_2, \eta_2) \quad (\text{A-29})$$

$$\cdot K(u_1 - \xi_1, \nu_1 - \eta_1) K^*(u_2 - \xi_2, \nu_2 - \eta_2) d\xi_1 d\eta_1 d\xi_2 d\eta_2$$

The four dimensional transfer function of the space invariant system is accordingly defined as

$$\begin{aligned} \mathbf{H}(\nu_1, \nu_2, \nu_3, \nu_4) &= F\{K(x_1, x_2) K^*(x_3, x_4)\} \\ &= \mathbf{K}(\nu_1, \nu_2) \mathbf{K}^*(-\nu_3, -\nu_4) \end{aligned} \quad (\text{A-30})$$

where

$$F\{\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \{\} e^{i2\pi(\nu_1 x_1 + \nu_2 x_2 + \nu_3 x_3 + \nu_4 x_4)} dx_1 dx_2 dx_3 dx_4 \quad (\text{A-31})$$

The spectrums of the object and image mutual intensities are defined by

$$\mathbf{J}_o' = F\{J_o'\} \quad (\text{A-32a})$$

$$\mathbf{J}_i = F\{J_i\} \quad (\text{A-32b})$$

Thus, in the frequency domain,

$$\mathbf{J}_i(\nu_1, \nu_2, \nu_3, \nu_4) = \mathbf{K}(\nu_1, \nu_2) \mathbf{K}^*(-\nu_3, -\nu_4) \mathbf{J}_o'(\nu_1, \nu_2, \nu_3, \nu_4) \quad (\text{A-33})$$

Since the impulse response is related to the pupil function via

$$\mathbf{K}(\nu_1, \nu_2) = P(\bar{\lambda} z_1 \nu_1, \bar{\lambda} z_2 \nu_2) \quad (\text{A-34})$$

equation (A-33) becomes

$$\begin{aligned} \mathbf{J}_i(\nu_1, \nu_2, \nu_3, \nu_4) &= P(\bar{\lambda} z_1 \nu_1, \bar{\lambda} z_2 \nu_2) \\ &\quad \cdot P^*(-\bar{\lambda} z_1 \nu_3, -\bar{\lambda} z_2 \nu_4) \mathbf{J}_o'(\nu_1, \nu_2, \nu_3, \nu_4) \end{aligned} \quad (\text{A-35})$$

Note that the transfer function of equation (A-34) drops to zero when ν_1 , ν_2 , ν_3 , or ν_4 exceed the limits imposed by the pupil function.

Although equation (A-35) appears simple, \mathbf{J}_o' depends both on the properties of the illumination and of the object. Combining equations (A-27) and (A-31), changing variables, and simplifying gives

$$\begin{aligned} \mathbf{J}_o'(\nu_1, \nu_2, \nu_3, \nu_4) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{T}_o(p + \nu_1) \\ &\quad \cdot \mathbf{T}_o^*(p - \nu_3, q - \nu_4) \mathbf{J}_o(p, q) dp dq \end{aligned} \quad (\text{A-36})$$

where \mathbf{T}_o is the Fourier transform of the object transmittance t_o . Equation (A-36) can be used with equation (A-35) to find the spectrum of the image intensity $I_i(\nu_u, \nu_v)$.

$$\begin{aligned} I_i(\nu_u, \nu_v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{T}_o(z_1, z_2) \mathbf{T}_o^*(z_1 - \nu_u, z_2 - \nu_v) \\ &\quad \cdot \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{K}(z_1 - p, z_2 - q) \mathbf{K}^*(z_1 - p - \nu_u, z_2 - q - \nu_v) \mathbf{J}_o(p, q) dp dq \right] dz_1 dz_2 \end{aligned} \quad (\text{A-37})$$

where the quantity in the brackets is called the *transmission cross-coefficient* and is independent of the object. The transmission cross-coefficient, T , can be simply evaluated by integrating the area of three partially overlapping circles⁶. When the illumination is incoherent, T reduces to the optical transfer function. For coherent illumination, it becomes the product of the pupil function P with its conjugate P^* .

A.2.2 One Dimensional Imaging

With the following change of variables, equation (A-37) is explicitly separated into the transmission cross-coefficient and the image intensity integral. Let

$$(\nu_u, \nu_v) \rightarrow (\xi, \eta) \quad (\text{A-38})$$

$$(z_1, z_2) \rightarrow (\xi + \xi', \eta + \eta') ; dz_1 = d\xi', dz_2 = d\eta'$$

$$(p, q) \rightarrow (-\bar{\xi}, -\bar{\eta})$$

$$J_o \rightarrow J = F\{\text{mutual intensity incident on object}\}$$

$$T_o \rightarrow F = F\{\text{transmittance of object}\}$$

Equations (A-38) are the variable changes needed to go from Goodman's to Kintner's equations. Equation (A-37) becomes

$$I(\xi, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(\xi + \xi', \eta + \eta'; \xi', \eta') \cdot F(\xi + \xi', \eta + \eta') F^*(\xi', \eta') d\xi' d\eta' \quad (\text{A-39a})$$

where

$$T(\xi_1, \eta_1; \xi_2, \eta_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J(\bar{\xi}, \bar{\eta}) K(\xi_1 + \bar{\xi}, \eta_1 + \bar{\eta}) \cdot K^*(\xi_2 + \bar{\xi}, \eta_2 + \bar{\eta}) d\bar{\xi} d\bar{\eta} \quad (\text{A-39b})$$

If the object varies in one dimension only, equation (A-39) becomes

$$I(\xi) = \int_{-\infty}^{\infty} T(\xi + \xi', \xi') F(\xi + \xi') F^*(\xi') d\xi' \quad (\text{A-40a})$$

where

$$T(\xi_1, \xi_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J(\bar{\xi}, \bar{\eta}) K(\xi_1 + \bar{\xi}, \bar{\eta}) K^*(\xi_2 + \bar{\xi}, \bar{\eta}) d\bar{\xi} d\bar{\eta} \quad (\text{A-40b})$$

Equation (A-40b) allows for a two dimensional optical system.

A further simplification is possible if the object consists of a periodic object with period $\frac{1}{\zeta_p}$ is represented by

$$f(\zeta) = \sum_{n=-\infty}^{\infty} a_n \delta(n\zeta_p - \zeta) \quad (\text{A-41})$$

in the frequency domain. Equation (A-41) can be inserted into equation (A-40a) to give

$$I(\zeta) = \sum_{n=-\infty}^{\infty} c_n \delta(n\zeta_p - \zeta) , (c_n = -c_n) \quad (\text{A-42a})$$

where

$$c_n = a_n a_0^* T(n\zeta_p; 0) + \sum_{n'=1}^{\infty} [a_{n+n'} a_n^* T[(n+n')\zeta_p; n'\zeta_p] + a_{n-n'} a_{-n}^* T[(n-n')\zeta_p; -n'\zeta_p]] \quad (\text{A-42b})$$

Therefore the intensity in the image plane is given by

$$I(x) = \sum_{n=-\infty}^{\infty} c_n \cos(2\pi n \zeta_p x) \quad (\text{A-43})$$

The coefficients c_n are sometimes difficult to calculate, since equation (A-40b) is difficult to evaluate. If the illumination system's numerical aperture is much smaller than the objective lens's numerical aperture (i.e., a small source), equation (A-40b) can be reduced to one dimension. Equation (A-40b) becomes

$$T(\xi_1; \xi_2) = \int_{-\infty}^{\infty} J(\bar{\xi}) K(\xi_1 + \bar{\xi}) K^*(\xi_2 + \bar{\xi}) d\bar{\xi} \quad (\text{A-44})$$

Equation (A-40b) can be evaluated as a generalized convolution of three circles, if aberrations (including focus error) in the objective pupil and in the source aperture are suppressed.

References for Appendix A:

- 1) J.W. Goodman, *Theory of Partial Coherence*, (to be published)
- 2) L. Mandel and E. Wolf, editors; *Selected Papers on Coherence and Fluctuations of Light, vols 1 and 2*, Dover Publications, N.Y., 1970.
- 3) M.J. Beran and G.B. Parrent, Jr., *Theory of Partial Coherence*, Prentice Hall, Englewood Cliffs, N.Y., 1960.
- 4) M. Born and E. Wolf, *Principle of Optics*, 5th edition, Pergamon Press, 1975.
- 5) A. Papoulis, *Systems and Transforms with Applications in Optics*, McGraw-Hill, 1968.
- 6) E.C. Kintner, "Method for the Calculation of Partially Coherent Imagery", *Applied Optics*, vol 17, no 17, pp 2747-2753, Sept 1978.
- 7) Hopkins, H.H., "The Concept of Partial Coherence in Optics", *Proc. Roy. Soc. (London)*, A208, 1951, p. 263.
- 8) Hopkins, H.H., "On the Diffraction Theory of Optical Images", *Proc. Roy. Soc. (London)*, A217, 1953, p. 408.
- 9) Hopkins, H.H., "The Frequency Response of a Defocused Optical System", *Proc. Roy. Soc. (London)*, 1956.
- 10) Hopkins, H.H., "Applications of Coherence Theory in Microscopy and Interferometry", *JOSA*, vol 47, no 6, June '57, p 508.
- 11) E.E.H. Linfoot, *Recent Advances in Optics*, Clarendon Press, 1958.
- 12) B.J. Thompson, "Image Formation with Partially Coherence Light", *Progress in Optics, vol VII*, North Holland Publishing Co., Amsterdam, 1969.
- 13) B.M. Watrasiewicz, "Theoretical Calculations of Images of Straight Edges in Partially Coherent Illumination", *Opt Acta*, vol. 12, p. 391, 1965.
- 14) R.E. Kinzly, "Investigations of the Influence of the Degree of Coherence upon Images of Edge Objects", *JOSA*, vol 55, # 8, Aug '65, p 1002.
- 15) R.E. Kinzly, "Images of Coherently Illuminated Edged Objects Formed by Scanning Optical Systems", *JOSA*, vol 56, no 1, p 9, Jan 1966.
- 16) A. Offner and J. Meiron, "The Performance of Optical Systems with Quasi-Monochromatic Partially Coherent Illumination", *Applied Optics*, vol 8, no 1, p 183-7, Jan 1969.
- 17) R. Hershel, "Partial Coherence in Projection Printing", SPIE vol. 135, *Developments in Semiconductor Microlithography III*, 1978.

In this case, off axis source points do not appreciably change the acceptance or rejection by the objective lens of beams diffracted by the mask. A square source (line source) also allows a one dimensional analysis. See the illumination section of the partial coherence description.

- 18) D. Nyssonen, "Linewidth Measurement with an Optical Microscope: the Effect of Operating Conditions on the Image Profile", *Appl. Opt.*, vol 16, No. 8, Aug '77, p 2223.
- 19) P.S. Considine, "Effects of Coherence on Imaging Systems", *JOSA*, vol 56, no 8, Aug 1966.
- 20) E. Wolf, "A macroscopic theory of interference and diffraction of light from finite sources I. Fields with a narrow spectral range", *Proc. of the Roy. Soc. of London*, A(225), p 96, 1954.
- 21) E. Wolf, "A macroscopic theory of interference and diffraction of light from finite sources II. Fields with a spectral range of arbitrary width", *Proc. of the Royal Society of London*, A(230), p 247, 1955.

Appendix B

Resolution and The Uncertainty Principle

The limit of resolution is simply a restatement of the uncertainty principle.

$$\Delta x \Delta p_x \approx h \quad (\text{B-1})$$

where h is Planck's constant. Figures B-1a and B-1b show a one dimensional source and lens system. A source segment of length Δx emits a photon of momentum $|p| = \frac{h}{\lambda}$. The uncertainty in the position of the atom that emitted the photon is given by Δx , and the uncertainty of the momentum in the x direction is given by $|\Delta p_x|$. From figure B-1a,

$$\sin \alpha \approx \frac{|\Delta p_x|}{|p|} \quad (\text{B-2})$$

But

$$|\Delta p_x| = \frac{\Delta p_x}{2} = \frac{h}{2\Delta x} \quad (\text{B-3})$$

And $|p| = \frac{h}{\lambda}$, giving

$$\sin \alpha \approx \frac{\frac{h}{2\Delta x}}{\frac{h}{\lambda}} = \frac{\lambda}{2\Delta x} \quad (\text{B-4})$$

From figure B-1b,

$$\sin \alpha \approx \frac{\frac{D}{2}}{z} \quad (\text{B-5})$$

Combining equations (B-4) and (B-5) gives

$$\Delta x \approx \frac{\lambda z}{D}$$

which is the familiar Rayleigh criterion for the resolution of two points.

Appendix C

COMMON BLOCK DOCUMENTATION

The following text lists the common blocks of the Lab of SAMPLE in alphabetical order. In blocks common to the Controller, the variables are set in the Controller unless otherwise specified.

/CBWIND/

---"common block window" is common to the Controller and to the IMAGE machine---

WINDOW WINDOW---the x (horizontal) window length in microns; it is set by the Controller with the default value of RSW for a "space", RLW for a "line", and $\min(\text{RSW}, \text{RLW})$ for a "linespace".

EDGE EDGE---the distance in microns from the left of the window that the mask edge is positioned; it is set by the Controller with the default value of $\text{WINDOW}/2$.

WNDORG WNDORG---the position of the left edge of the window in microns. The true $x=0$ position is determined by the Fourier series routine that analyzes the mask. WNDORG is set by the IMAGE machine and is an internal parameter that the user never observes. Output x positions are referenced to the line edge, defined as $x=0$.

/COPYWD/

---"copy window" is common to the Controller and to the IMAGE and DEVELOP machines---

xxxx xxxx---CPWIND, CPEDGE, and CPWORG correspond to WINDOW, EDGE, and WNDORG in common block CBWIND. The window specifications used in the IMAGE machine are used by the DEVELOP machine. The variables are copied to block COPYWD by subroutine IMAGE, in case the user errs by respecifying the window without re-running IMAGE. Although machine DEVELOP does not use the variables in block CBWIND(or COPYWD) directly, the variables are used for the output graphs.

/DEVFLG/

---"develop flag" is common to the Controller and to machine DEVELOP---

IDEVFL IDEVFL(5)---a one dimensional array containing the flags for machine DEVELOP. As for the flags in all the common blocks, 1 means "yes" and 0 means "no". IDEVFL(1)--print the no. of string pts.,etc. and CHKR point totals? IDEVFL(2)--cards for the etch plot? IDEVFL(3)---set equal to more points for publication runs(more time-consuming and slightly more accurate runs). IDEVFL(3) set equal to 0 gives sufficient accuracy for valid conclusions. The main advantage in publication runs is smoother HP plots. IDEVFL(4)---not used presently. IDEVFL(5)---continuation of development?

/DEVPAR/

---"development parameters" is common to the resist development routines and to the Controller---

E1,2,3 E1, E2, E3---parameters used by function RATE. Currently they are set in the Controller; eventually they will be looked up in a table as a function of prebake temperature, developer temperature, and resist type.

/DEVTIM/

---"development times" is common to the resist development routines and to the Controller---

MXNDEV MXNDEV---the maximum number of development contours that can be requested by the user--currently 20.

DEVSRT DEVSRT---the development time in seconds of the first development contour(output).

DEVEND DEVEND---the development time in seconds of the final development contour.

DEVINC DEVINC---the time in seconds between the intermediate development contours.

/DOSPAR/

---"dose parameters" common to EXPOSE and the Controller---

DOS DOS--the total energy(all wavelengths) per unit area incident on the mask in mJ/cm**2.

/DVELP1/

---internal to the resist develop routines(machine DEVELOP)---

XZ XZ(450)---complex array containing the x and z positions of the 450 possible development string-points.
XMAX XMAX---the real(not including the boundary) maximum value of x .
ZMAX ZMAX---the real maximum value of z .
NPTS NPTS---the current number of development string-points.
CXZL CXZL---the normalized left endpoint direction of development.
CXZR CXZR---the normalized right endpoint direction.
NADCHK NADCHK---the number of advances(calls on subroutine CYCLE) between checks(calls on subroutine CHKR).
NCKOUT NCKOUT---the number of checks between outputs; thus the number of advances between outputs is $NADCHK * NCKOUT$.

/DVELP2 /

---internal to the resist develop routines(machine DEVELOP)---

TADV TADV---the current time-between-advances(for subroutine CYCLE).
TCHK TCHK---the current time-between-checks.
TTOT TTOT---contains the total current development time during the execution of the DEVELOP machine.
IFLAG IFLAG---flag set in cycle that tells the sub-controller(DVELOP) that some string segments are too long or too short. CHKR is then called from DVELOP.
SMAXX SMAXX---the maximum x string segment length allowed by CHKR(& CYCLE).
SMINX SMINX---the minimum x string segment length allowed by CHKR.
SMAZX SMAZX---the maximum z string segment length allowed by CHKR.

/DVELP3/

---internal to the resist develop routines(machine DEVELOP)---

- NZFLG** NZFLG---a flag used between subroutines DVELOP and CYCLE in order to signal when the string has broken through the resist. The number of checks between outputs is then reduced.
- NADFLG** NADFLG---the output number of the string that broke through the resist.

/DVELP4/

---"develop 4" is internal to machine DEVELOP---

- BREAK** BREAK---the estimated time to resist breakthrough. Found in the sub-controller of DEVELOP.
- MAXPTS** MAXPTS---when the number of string points exceeds MAXPTS, DELOOP is called. Found in the sub-controller of DEVELOP.
- NADSAV** NADSAV---contains the NADCHK before resist breakthrough.
- NCKSV1** NCKSV1---contains NCKOUT before resist breakthrough for the first development output.
- NCKSV2** NCKSV2---contains NCKOUT before resist breakthrough for the intermediate outputs.
- NOUT** NOUT---the number of development outputs requested; it is calculated in subroutine DVELOP.

/EXPFLG/

---"expose flag" is common to the Controller and to machine EXPOSE---

- IEXPFL** IEXPFL(5)---a one dimensional array containing the flags for EXPOSE. As for all the flags in the common blocks, 1 means "yes" and 0 means "no". IEXPFL(1)--request for an M versus energy and z array. IEXPFL(2)--request for an M versus x and z array. IEXPFL(3)-not used at present. IEXPFL(4)--not used at present. IEXPFL(5)--not used at present.

/EXPINT/

---"expose internal" is internal to machine EXPOSE---

- PWRBGN** PWRBGN(10)---gives the reflected power from the unexposed resist for the ten possible wavelengths; subroutine CLCMXZ passes PWR to subroutine EXPMSG.

PWREND PWREND(10)---gives the reflected power for a resist exposed to the table maximum of EXPOS(NENDIV) mJ/cm**2.

KFLAG KFLAG---If subroutine CLCMXZ requests an energy outside of the exposure table RMZDOS(51,21), KFLAG is set equal to 1 in order the signal the sub-controller for machine EXPOSE to recalculate RMZDOS(51,21) by calling on subroutine CLCMZD with a higher value of DELTM.

/EXPTBL/

---"expose table" is common to machine EXPOSE and to the Controller---

EXPOS EXPOS(21)---a one dimensional array containing the accumulated normalized energy of the energy incident on the resist. It is the summation of energy times relative intensity for all the wavelengths. The maximum number of elements EXPOS may have is 21; normally it has NENDIV+1 elements--since EXPOS(1)=0. EXPOS is not set by the Controller; it passes information from subroutine CLCMZD to subroutine CLCMXZ.

RMZDOS RMZDOS(51,21)---a two dimensional array containing the M values as a function of layer i and energy division j. Maximum i=51 and maximum j=21. Position i=1 refers to the air and is not used. Position j=1 refers to no exposure. RMZDOS is used to pass information from subroutine CLCMZD to subroutine CLCMXZ; it is not set in the Controller.

/FOUSER/

---"fourier series" is common to the optical simulation machine(IMAGE routines) and to the Controller---

MXNMFR MXNMFR---the maximum number of spatial frequencies used to form the mask--currently 41 and set in the Controller.

NMFRCP NMFRCP---the number of spatial frequencies used to form the mask(set in subroutine CLCMTF).

FSAMSK FSAMSK(41)---a one dimensional array containing the weightings of each spatial frequency component of the mask(set in subroutine CLCMTF).

FSAIMG FSAIMG(41)---a one dimensional array containing the weightings of the spatial frequency components of the image(set in the IMAGE machine).

/HORIMG/

---"horizontal image" is common to machines IMAGE, EXPOSE, and DEVELOP and to the Controller---

- DELTX** DELTX---the distance between horizontal points x in microns(set in the IMAGE machine subroutine CLCMTF).
- MNHPTS** MNHPTS---the maximum number of horizontal points in x that may be requested by the user--currently 50. Two boundary cells are added by the EXPOSE machine--giving a total of 52 points in x . Each point is the center of a cell; Cells 1 and "right-end" are boundary cells. The midpoint(in x) of cell 2 is the left side of the physical boundary. The midpoint(in x) of the second cell from the right represents the maximum x .
- NMHPTS** NMHPTS---the number of horizontal points in x .
- HORINT** HORINT(50)---a one dimensional array containing the normalized energy incident on the horizontal plane. A maximum of 50 points is allowed. The normalized energy is equal to the summation of the energy*(relative intensity) for all the wavelengths.

/IMGFLG/

---"image flag" is common to the Controller and machine IMAGE---

- IMGFL** IMGFL(5)---a one dimensional array containing the flags for the IMAGE machine. As for the flags in all of the common blocks, 1 means "yes" and 0 means "no". IMGFL(1)--Request for an intensity pattern plot. IMGFL(2)--Request for an MTF pattern plot. IMGFL(3)--Request for punched cards. Cards are only punched for the plots requested. IMGFL(4)--triggers the diagnostics in the partial coherence routines. IMGFL(5)--circular(=1) or square(default; =0) objective lens — not implemented yet.

/IMGSYS/

---first "image system block" common to the Controller and to machine IMAGE---

- IMSPRO** IMSPRO---signals a projection system.
- IMSCON** IMSCON---signals a contact system.
- IMSYTY** IMSYTY---is compared with the above to signal the system desired.

/IMG2PR/

---"image system, 2nd block, pertinent to projection" is common to the Controller and to machine IMAGE---

- RNA RNA---the numerical aperture of the objective lens in image space.
- MXNWTF MXNWTF---maximum number of weighting frequencies--not used.
- NMWTF NMTWTF---number of weighting frequencies--not used.
- RMTFWT RMTFWT(41)---MTF weights for the 41 possible frequencies, 40 frequencies plus dc--not used.

/IMG3PR/

---"image system, 3rd block, pertinent to projection" is common to the Controller and to machine IMAGE---

- IINCOH IINCOH---incoherent light.
- IPARCO IPARCO---partially coherent light.
- IFULCO IFULCO---fully coherent light.
- ICOHER ICOHER---the case considered by IMAGE. ICOHER is compared with the previous block variables to determine the type of light requested for projection printing.
- SIGMA SIGMA---the partial coherence parameter(= {NA of condenser system}/{NA of objective lens}). NA stands for "numerical aperture".
- DFDIST DFDIST---the distance in microns that the air-resist interface is displaced from the perfect focus plane. A positive DFDIST indicates that the wafer is further from the lens than the focal plane. A negative DFDIST indicates that the wafer is closer to the lens than the focal plane. Default value is 0. Positive and negative distances give the same result.

/IMG4CO/

---"image system, 4th block, contact image" is common to IMAGE machine and to the Controller---

- SEPMTC SEPMTC---contact system parameter: separation of mask and resist surface.
- C1,2 C1,2---constants that determine image intensity $I(x)$ via

$$I(x) = C_1 I_0 e^{C_2 \sqrt{\frac{2}{\lambda \cdot \text{SEPMTC}}}}$$

/IO1 /

---"input-output block 1" is common to all of the machines and to the Controller---

XXXX XXXX---all of the variables in the common block are assigned numbers appropriate to the computer system in use. The machines access the printer with the FORTRAN statement "WRITE(IPRINT,#) etc.". The program(Lab) on the CDC 6400 at UC Berkeley only uses IPRINT and IPUNCH.

/MVSP0S/

---"M versus position" is common to machines IMAGE and DEVELOP---

RMXZ RMXZ(52,52)---a two dimensional array containing M as a function of position (x,z). Maximum dimension is (52,52) corresponding to the maximum 50 by 50 position points in the resist plus a boundary layer extrapolated by subroutine CLCMXZ. CLCMXZ creates RMXZ from arrays RMZDOS and HORINT for use in function RATE(in machine DEVELOP). RMXZ is the output of the EXPOSE machine.

/OBJMSK/

---"object mask" is common to the Controller and to machine IMAGE---

MLINE MLINE---the code number for a line.

MSPACE MSPACE---the code number for a space.

MLNSPA MLNSPA---the code number for a line-space.

MASKTY MASKTY---the mask type chosen(one of the three above).

RLW RLW---the line width in microns(need not be specified for MASKTY=MSPACE).

RSW RSW---the space width in microns(need not be specified for MASKTY=MLINE).

/OPTCHP/

---"optical chip parameters" is common to machine EXPOSE and to the Controller--

RINDEX RINDEX(10,6,2)---a three dimensional array giving the complex index of the various layers as a function of wavelength. The first dimension refers to the ten possible exposure wavelengths. The first position of the second dimension refers to the resist layer; the second, third, fourth, and fifth positions of the second dimension

refer to the first, second, third, and fourth intermediate substrate layers; and the last position refers to the substrate itself. If there are fewer than four intermediate substrate layers, the latter array positions may remain uninitialized. For example, oxide on silicon would be represented by the second position referring to the oxide and the third position referring to the silicon. The fourth, fifth, and sixth array positions remain uninitialized. The third dimension contains the real and imaginary parts of the index in the first and second positions, respectively. $RINDEX(h,j,1) = n$ and $RINDEX(h,j,2) = -k$, where the complex index $N = n - ik$.

/OPTIC /

---"optic block" is internal to machine IMAGE---

- CURWVL** CURWVL---the current wavelength IMAGE is processing(used in the message subroutine).
- V1** V1---the fundamental spatial frequency of the mask.
- VMAX** VMAX---the cutoff spatial frequency of the coherent or incoherent system. VMAX for a partially coherent system is $(1 + \text{SIGMA}) * (\text{VMAX for a coherent system})$; all the spatial frequencies needed for the partially coherent system are calculated and stored in FSAMSK(41).
- THORIN** THORIN(50)---a one dimensional array storing the horizontal image intensity for one wavelength(like HORINT, but for one wavelength).

/PCTERM/

---"partial coherence term" is internal to the partial coherence routines FASTPC and PARCOH in the IMAGE machine for the purpose of saving memory---

- TERM** TERM(82,3)---the 82 refers to the 82 possible frequencies(the plus and minus of the 41 possible frequencies). The 3 refers to frequency weighting(position 1), frequency(position 2), and phase consideration(position 3--not used in FASTPC).

/PREBAK/

---"prebake" is common to the Controller and to the EXPOSE routines---

- DGRADM** DGRADM---M is degraded by this number(default=0) throughout the $M(x,z)$ array. It is an approximate prebake since the RMZDOS table is different due to the different A, B, C after prebake and due to the different starting value of M. DGRADM may be deleted or corrected in the future. There is no Input-Interface card to access this feature(except TRIAL).

/RESPAR/

---"resist parameters" is common to machine EXPOSE and to the Controller---

WLABC WLABC(10,4)---a two dimensional array containing resist information for the ten wavelengths. The first dimension refers to the ten wavelergths. The first position of the second dimension refers to the wavelength in microns; the second, third, and fourth positions are A, B, and C for that wavelength, respectively. The exposure linearization section in subroutine CLCMZD has the greatest accuracy when the dominant wavelength is in the first position of the first wavelength; it is not necessary unless there is a wide difference in C*(rel inten) between the exposure wavelengths. The dominant wavelength is defined as the wavelength with the greatest (relative intensity)*C .

/RUCOMP/

---"reuse computation" is common to machine EXPOSE and to the Controller.

IRUMZD IRUMZD---saves the computation of the M versus z and dose table if the table need not be recomputed. IRUMZD = 0 if the table is to be computed and = 1 if the table is not to be computed. The Controller initializes IRUMZD to 0; it also resets it to 0 whenever the relevant input parameters have been changed since the last run of EXPOSE. EXPOSE sets IRUMZD to 1 just before returning control to the Controller.

/SIMPAR/

---"simulation parameters" is common to machines IMAGE, EXPOSE, and DEVELOP and to the Controller---

NPRLYR NPRLYR---the number of layers the resist is to be divided into.

NPRPTS NPRPTS---NPRLYR + 1

NENDIV NENDIV---the number of energy divisions the exposure of the resist is broken into. Currently the maximum value allowed is 21--with the first division referring to no exposure.

DELTM DELTM---the incremental M for the linearization of incremental energy exposure section in subroutine CLCMZD. The default value for DELTM is .7/float(NENDIV-1), where NENDIV=15 . DELTM gives control over the range of exposure of the resist and is initialized in the Controller. The value of NENDIV determines the "coarseness" of the exposure table, and the value of the numerator determines the energy "range" of the table. If subroutine CLCMXZ calls for a value outside of the exposure table, control is returned to subroutine EXPOSE, where subroutine CLCMZD is called to expand the exposure table. The table remains expanded for the remainder of this batch's runs.¹ Machine EXPOSE internally handles the value of DELTM, except for initialization.

DELTZ DELTZ---the distance between the resist layers in microns. It is calculated in subroutine CLCMZD and used in routine CLCMZD and function RATE.

/SPECTR/

---"spectrum" is common to machines IMAGE and EXPOSE and to the Controller---

MXNLMD MXNLMD---the maximum number of wavelengths allowed(currently 10).

NMLMBD NMLMBD---the number of wavelengths used in exposure.

RLAMBD RLAMBD(10)---a one dimensional array containing the ten wavelengths in microns.

RELINT RELINT(10)---a one dimensional array containing the relative intensities of the ten wavelengths. The relative intensities are normalized to one; thus the sum of the values contained in RELINT is one.

/THKNES/

---"thickness" is common to machine EXPOSE and to the Controller---

MNLYRS MNLYRS---the maximum number of layers allowed in the problem(currently 6). The first layer is the resist layer; the last layer is the substrate. Four intermediate layers are allowed.

NMLYRS NMLYRS---the number of layers in the problem.

THICK THICK(6)---a one dimensional array containing the thicknesses of the various layers described in the RINDEX section of common block OPTCHP. The substrate thickness is assumed infinity and is represented by THICK(6). THICK(6) is not initialized; it will be used by the etching routines when they are implemented. The substrate is considered a layer. If there are fewer than four layers, the latter positions may remain uninitialized.

Appendix D

SAMPLE Code (Lab)

```

SUBROUTINE IMAGE

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /COPYWD/ CPWIND,CPEDGE,CPWORG
COMMON /IMGFLG/ IMGFL(5)
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IO1 / ITERM,IBULK,IROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)

C IMAGE GENERATES A HORIZONTAL IMAGE AT THE SURFACE OF THE RESIST

CALL IMGMSG(1)
C CLEAR THE HORIZONTAL IMAGE ARRAY
DO 2 IHPT=1,NMHPTS
HORINT(IHPT) = 0.
2 CONTINUE
DO 3 IFRCP=1,MXNMFR
FSAMSK(IFRCP) = 0.
FSAIMG(IFRCP) = 0.
3 CONTINUE

IF(EDGE.LT.(.25*WINDOW)) CALL IMGMSG(12)
IF(EDGE.GT.(.75*WINDOW)) CALL IMGMSG(13)
DO 10 ILMBD = 1,NMLMBD
DO 5 IHPT=1,NMHPTS
THORIN(IHPT) = 0.
5 CONTINUE
IDUMMY = ILMBD
CALL CLCMTF(IDUMMY)
CALL IMGMSG(11)
IF(IMGFL(1).EQ.1)CALL PLTPAT(IDUMMY)
10 CONTINUE
IF(IMGFL(2).EQ.1)CALL PLTOTF

WRITE(IPRINT,22222)
22222 FORMAT(1H1)
CPWIND = WINDOW
CPEDGE = EDGE
WNDORG = CPWORG
RETURN

END

SUBROUTINE CLCMTF(ILMBD)

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IMGSYS/ IMSPRO,IMSCON,IMSYTY
COMMON /IMG2PR/ RNA,MXNWTF,NMWTF,RMTFT(41)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /IMG4CO/ SEPMTC,C1,C2
COMMON /OBJMSK/ MLINE,MSPACE,MLNSPA,MASKTY,RLW,RSW
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)

C CLCMTF COMPUTES THE DIFFRACTION LIMITED FOURIER COMPONENTS FOR THE
C VARIOUS CASES. A MAXIMUM OF 41 FREQUENCIES IS TAKEN FOR THE LINESPACE
C AND 41 FREQUENCIES ARE TAKEN FOR LINE AND FOR SPACE.
C NOTE - EACH HORIZONTAL POINT IS THE CENTER OF A CELL.

CURWL = RLAMB(ILMBD)
C BRANCH ACCORDING TO IMAGING SYSTEM
IF(IMSYTY.EQ.IMSPRO)GO TO 10
IF(IMSYTY.EQ.IMSCON)GO TO 100

```

```

CALL IMGMSG(10)
RETURN

C PROJECTION SYSTEM
C SET MAXIMUM FREQUENCY FOR COHERENT, PARTIALLY COHERENT,
C OR INCOHERENT CASE.
10 IF(ICOHER.EQ.IINCOH) VMAX = 2.*RNA/RLAMB(ILMBD)
   IF(ICOHER.EQ.IPARCO) VMAX = (RNA/RLAMB(ILMBD))*(1.+SIGMA)
   IF(ICOHER.EQ.IFULCO) VMAX = RNA/RLAMB(ILMBD)

   IF(MASKTY.EQ.MLINE)GO TO 20
   IF(MASKTY.EQ.MSPACE)GO TO 30
   IF(MASKTY.EQ.MLNSPA)GO TO 40
   CALL IMGMSG(2)

C MASK IS A LINE
20 DELTX = WINDOW/FLOAT(NMHPTS-1)
   WNDORG = .5*RLW-EDGE
   NMFRCP = MXNMFR
   V1 = VMAX/FLOAT(MXNMFR-1)
   RSW = 1./V1 - RLW
   CALL IMGMSG(4)
   GO TO 42

C MASK IS A SPACE
30 DELTX = WINDOW/FLOAT(NMHPTS-1)
   WNDORG = .5*RSW-EDGE
   NMFRCP = MXNMFR
   V1 = VMAX/FLOAT(MXNMFR-1)
   RLW = 1./V1 - RSW
   CALL IMGMSG(5)
   FSAMSK(1) = RSW*V1
   ARG = RSW*V1*3.14159
   DO 35 IFRCP=2,NMFRCP
   RN = FLOAT(IFRCP-1)
   FSAMSK(IFRCP) = (.63662/RN)*SIN(RN*ARG)
35 CONTINUE
   GO TO 50

C MASK IS A LINESPACE
40 DELTX = WINDOW/FLOAT(NMHPTS-1)
   WNDORG = .5*RLW - EDGE
C FIND THE FUNDAMENTAL FREQUENCY FOR THE LINESPACE CASE
   V1 = 1./(RLW + RSW)
C FIND THE NUMBER OF FREQUENCY COMPONENTS THAT NEED TO BE TAKEN
   ITEMP = INT(VMAX/V1)
   NMFRCP = ITEMP + 1
   IF((VMAX/V1-FLOAT(ITEMP)).GE..95) CALL IMGMSG(9)
   IF(NMFRCP.LT.MXNMFR) GO TO 42
   CALL IMGMSG(3)
   NMFRCP = MXNMFR
42 ARG = RLW*V1*3.14159
   FSAMSK(1) = RSW*V1
   DO 45 IFRCP=2,NMFRCP
   RN = FLOAT(IFRCP-1)
   FSAMSK(IFRCP) = (-.63662/RN)*SIN(RN*ARG)
45 CONTINUE

C GO TO THE COHERENT CASE, THE PARTIALLY COHERENT CASE, AND
C THE INCOHERENT CASE.
50 IF(ICOHER.EQ.IFULCO) GO TO 60
   IF(ICOHER.EQ.IPARCO) GO TO 70
   IF(ICOHER.EQ.IINCOH) GO TO 80
   STOP

60 IF(DFDIST.NE.0.0) GO TO 66
C COHERENT LIGHT WITHOUT DEFOCUS.
61 DO 62 IFRCP=1,NMFRCP
   FSAIMG(IFRCP) = FSAMSK(IFRCP)
62 CONTINUE
   CALL OUTPAT(ILMBD)
   RETURN
C COHERENT LIGHT WITH DEFOCUS.
66 CALL PARCOH(ILMBD)
   RETURN

70 IF(SIGMA.GT.3.) CALL IMGMSG(14)
   IF(DFDIST.NE.0.0) GO TO 76
C PARTIALLY COHERENT LIGHT WITHOUT DEFOCUS.
71 CALL FASTPC(ILMBD)
   RETURN

```

C PARTIALLY COHERENT LIGHT WITHOUT DEFOCUS

```
76 CALL PARCOH(ILMBD)
RETURN
```

```
80 CALL IMGMSG(15)
IF(DFDIST.NE.0.0) GO TO 86
```

C INCOHERENT LIGHT WITHOUT DEFOCUS.

```
81 FSAIMG(1) = FSAMSK(1)
DO 82 IFRCP=2,NMFRCP
RN = FLOAT(IFRCP-1)
TEMP1 = V1*RN/VMAX
FSAIMG(IFRCP) = FSAMSK(IFRCP)*.63662*( ACOS(TEMP1) -
TEMP1*SQRT(1. - TEMP1**2) )
```

```
82 CONTINUE
CALL OUTPAT(ILMBD)
RETURN
```

C INCOHERENT LIGHT WITH DEFOCUS.

```
86 CALL DFCOTF(ILMBD)
CALL OUTPAT(ILMBD)
RETURN
```

C CONTACT SYSTEM

C BRANCH ACCORDING TO THE TYPE OF THE MASK

```
100 IF (MASKTY .EQ. MLINE) GOTO 120
IF (MASKTY .EQ. MSPACE) GOTO 140
IF (MASKTY .EQ. MLNSPA) GOTO 160
CALL IMGMSG(2)
RETURN
```

```
120 RSW = 4.0 * RLW
```

```
GO TO 160
```

```
140 RLW = 4.0 * RSW
```

```
160 DELTX = WINDOW/FLOAT(NMHPTS-1)
```

```
WNDORG = RSW/2. - EDGE
```

```
T2BYLL = 2.0 / ( SEPMTCLAMBDA(ILMBD) )
```

```
S2BYLL = SQRT(T2BYLL)
```

```
XCLIP = (RSW/2.0) + (ALOG(C1))/(C2*S2BYLL)
```

C CREATE THE OUTPUT PATTERN.

```
DO 190 IHPT = 1, NMHPTS
```

```
VIHPT = (IHPT - 1)
```

```
XDIST = VIHPT*DELTX + WNDORG
```

```
HORIZI = 1.0
```

```
IF(XDIST.GT.XCLIP) HORIZI=C1*EXP(-C2*(XDIST-RSW/2.0)*S2BYLL)
```

```
THORIN(IHPT) = THORIN(IHPT) + HORIZI
```

```
HORINT(IHPT) = HORINT(IHPT) + HORIZI*RELINT(ILMBD)
```

```
190 CONTINUE
```

```
RETURN
```

```
END
```

SUBROUTINE DFCOTF(ILMBD)

```
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
```

```
COMMON /IMG2PR/ RNA,MXNWF,NMWFTR,MTFWT(41)
```

```
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
```

```
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
```

```
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMBDA(10),RELINT(10)
```

C SUBROUTINE DFCOTF CALCULATES THE INCOHERENT OTF FOR THE DEFOCUS

C PARAMETER DFDIST(= DISTANCE IN MICRONS OF FOCUS ERROR).

C TWENTY ITERATIONS GIVES ABOUT 3 DECIMAL PLACES IN THE MTF CALCULATION.

```
NMITER = 20
```

```
IF(SIGMA.LT.3.01)CALL IMGMSG(7)
```

```
DELTA = 2.*RNA*RNA*DFDIST/RLAMBDA(ILMBD)
```

```
VREDUC = .5*V1*RLAMBDA(ILMBD)/RNA
```

```
FSAIMG(1) = FSAMSK(1)
```

```
DO 20 IFRCP = 2,NMFRCP
```

```
RN = FLOAT(IFRCP-1)
```

```
TEMP = 0.
```

```
VR = RN*VREDUC
```

```
DU = (1.-VR)/FLOAT(NMITER)
```

```
U = VR + DU/2.
```

```
DO 10 ITER = 1,NMITER
```

```
TEMP = TEMP+1.2732*SQRT(1.-U*U)*COS(6.2832*VR*DELTA*(U.-VR))*DU
```

```
U = U + DU
```

```
10 CONTINUE
```

FSAIMG(IFRCP) = TEMP*FSAMSK(IFRCP)
20 CONTINUE

RETURN
END

SUBROUTINE PARCOH(ILMBD)

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IMGFLG/ IMGFL(5)
COMMON /IMG2PR/ RNA,MXNWF,NMWTF,RMTFWT(41)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /IO1 / ITERM,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /PCTERM/ TERM(82,3)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)
COMPLEX AL

C PARCOH CALCULATES THE INTENSITY PATTERN ON THE RESIST FOR PARTIALLY
C COHERENT LIGHT WITH VARIOUS DEGREES OF DEFOCUS AND FOR
C COHERENT LIGHT WITH VARIOUS DEGREES OF DEFOCUS.

IF(IMGFL(4).EQ.1) WRITE(IPRINT,1111)
1111 FORMAT(///9X,47HDIAGNOSTICS FOR THE PARTIAL COHERENCE ROUTINES:./)

1 AREAS = 15.
VCOBJ = RNA/RLAMB(ILMBD)
VCCOND = SIGMA*VCOBJ

IF(ICOHER.EQ.IPARCO) GO TO 20
C FULL COHERENCE SECTION.

10 NCOHAR = 1
DSFRQ = 0.
RNORML = 1.
FACTOR = .5
NI = 1
GO TO 25

C PARTIAL COHERENCE WITH DEFOCUS SECTION.

20 NCOHAR = INT(SIGMA*AREAS + .0001)
IF(NCOHAR.L.E.7)NCOHAR=8
DSFRQ = VCCOND/FLOAT(NCOHAR)
RNORML = DSFRQ/(2.*AMIN1(VCCOND,VCOBJ))
FACTOR = .5*SQRT(RNORML)
NI = 2

IF(DFDIST.NE.0.) GO TO 25
C PARTIAL COHERENCE WITH PERFECT FOCUS SECTION.

RNORML = 2.*RNORML
FACTOR = .5*SQRT(RNORML)
NI = 1

C A POSITIVE DFDIST MEANS WAFER IS FURTHER AWAY FROM THE LENS THAN
C THE FOCAL PLANE.

25 DFOCPH = RLAMB(ILMBD)*DFDIST/2.
DO 100 ICOHAR=1,NCOHAR
VS = DSFRQ*FLOAT(2*ICOHAR-1)/2.

IF(IMGFL(4).EQ.1) WRITE(IPRINT,401) ICOHAR,RNORML,VS,FACTOR
401 FORMAT(/1X,7HICOHAR=,I3./5X,9HRNORML = ,F7.5,7H VS= ,F7.5,
* 12H FACTOR = ,F7.5)

C DO ONE SET OF SYMMETRIC COHERENCE AREAS OF THE SOURCE.
DO 75 I=1,NI

C DETERMINE WHICH FREQUENCIES MAKE IT THRU THE LENS.

KOUNT = 0
FRINCR = 0.
IFR = 1
DO 50 J=1,NMFRCP

C SHIFT THE POSITIVE MASK FREQUENCIES

VM = V1*FRINCR
SHFREQ = VS + VM
IF((ABS(SHFREQ).GT.VCOBJ).OR.(ABS(FSAMSK(IFR)).LT..001))GO TO 40
KOUNT = KOUNT + 1
TERM(KOUNT,1) = FACTOR*FSAMSK(IFR)
TERM(KOUNT,2) = VM
TERM(KOUNT,3) = SHFREQ*SHFREQ*DFOCPH

C SHIFT THE NEGATIVE MASK FREQUENCIES

```

40 SHFREQ = VS - VM
   IF((ABS(SHFREQ).GT.VCOBJ).OR.(ABS(FSAMSK(IFR)).LT..001))GO TO 45
   KOUNT = KOUNT + 1
   TERM(KOUNT,1) = FACTOR*FSAMSK(IFR)
   TERM(KOUNT,2) = -VM
   TERM(KOUNT,3) = SHFREQ*SHFREQ*DFOCPH
45 FRINCR = FRINCR + 1.
   IFR = IFR + 1
50 CONTINUE
   IF(IMGFL(4).EQ.0) GO TO 51
   WRITE(IPRINT,58) (TERM(II,1),II=1,KOUNT)
   WRITE(IPRINT,58) (TERM(II,2),II=1,KOUNT)
   WRITE(IPRINT,58) (TERM(II,3),II=1,KOUNT)
58 FORMAT(1X,12(F10.5))

C CALCULATE THE INTENSITY PATTERN FOR THE HORIZONTAL POINTS BY
C MULTIPLYING ALL COMBINATIONS OF FREQUENCY TERMS PRODUCED BY THE
C PRESENT COHERENCE AREA.
51 DO 53 IHPT=1,NMHPTS
   X = FLOAT(IHPT-1)*DELTX + WNDORG
   AL = (0.,0.)
   DO 52 K=1,KOUNT
   ARG = -TERM(K,2)*X + TERM(K,3)
   AL = AL + CMPLX(TERM(K,1),0.)*CEXP( (0.,6.2832)*CMPLX(ARG,0.) )
52 CONTINUE
   THORIN(IHPT) = REAL(AL*CONJG(AL)) + THORIN(IHPT)
53 CONTINUE

   VS = -VS
75 CONTINUE

100 CONTINUE

C ACCUMULATE THE INTENSITY IN HORINT.
DO 200 IHPT=1,NMHPTS
   HORINT(IHPT) = HORINT(IHPT) + RELINT(ILMBD)*THORIN(IHPT)
200 CONTINUE

   RETURN
   END

SUBROUTINE FASTPC(ILMBD)

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /FOUSER/ MXNMF, NMFRC, FSAMSK(41), FSAIMG(41)
COMMON /HORIMG/ DELTX, MNHPTS, NMHPTS, HORINT(50)
COMMON /IMGFLG/ IMGFL(5)
COMMON /IMG2PR/ RNA, MXNWTF, NMWTF, RMTFWT(41)
COMMON /IMG3PR/ IINCOH, IPARCO, IFULCO, ICOHER, SIGMA, DFDIST
COMMON /IOI / ITERM, IBULK, IPROUT, IRESV1, IIN, IPRINT, IPUNCH
COMMON /OPTIC / CURWL, V1, VMAX, THORIN(50)
COMMON /PCTERM/ TERM(82,3)
COMMON /SPECTR/ MNLMBD, NMLMBD, RLAMB(10), RELINT(10)
COMPLEX AL

C FASTPC CALCULATES THE INTENSITY PATTERN ON THE RESIST FOR PARTIALLY
C COHERENT LIGHT WITHOUT DEFOCUS. IT RUNS MUCH FASTER THAN PARCOH.

   IF(IMGFL(4).EQ.1) WRITE(IPRINT,1111)
1111 FORMAT(///9X,47HDIAGNOSTICS FOR THE PARTIAL COHERENCE ROUTINES:/)

   VCOBJ = RNA/RLAMB(ILMBD)
   VCOND = VCOBJ*SIGMA
   VADSKP = 0.
   IENDFL = 0
   IFCINC = 1

   DO 10 I=1,NMFRC
   IFCPSV = I
   VTMP1 = FLOAT(I-1)*V1
   VTMP1 = VCOBJ-VTMP1
   IF(VTMP1.LE.V1) GO TO 11
10 CONTINUE
11 VTMP2 = (V1-VTMP1)
   IF(VTMP2.GT.VTMP1) GO TO 15
13 VS1 = (VTMP2 )*.99999
   VS2 = (VTMP1-VTMP2)*.99999
   VS3 = (V1-VS2 )*.99999

```

```

IFCPSV = IFCPSV + 1
INCFAC = -1
GO TO 17
15 VS1 = (VTMP1 )*.99999
VS2 = (VTMP2-VTMP1)*.99999
VS3 = (V1-VS2 )*.99999
INCFAC = 1
17 VS = VS1
VSADD = VS1
IF(IMGFL(4).EQ.1) WRITE(IPRINT,401) VS1,VS2,VS3,IFCPSV

DO 100 I=3,100

C CHECK FOR REDUNDANT CALCULATIONS.
ICHECK = IFCPSV
IF(IENDFL.EQ.1) GO TO 30
IF(IFCPSV.LE.0) ICHECK = 2-IFCPSV
IF(ABS(FSAMSK(ICHECK)).GT..0001) GO TO 30
VADSKP = VSADD + VADSKP
IF(IMGFL(4).EQ.1) WRITE(IPRINT,402) I,VS,VSADD,VADSKP,
* IFCPSV,IFCINC,INCFAC,ICHECK
GO TO 60

C DETERMINE WHICH FREQUENCIES MAKE IT THRU THE LENS.
30 KOUNT = 0
FRINCR = 0.
RNORML = (VSADD+VADSKP)/(VCOBJ*AMIN1(SIGMA,1.))
IF(RNORML.GT.1.) RNORML=1.
FACTOR = .5*SQRT(RNORML)
IF(IMGFL(4).EQ.1) WRITE(IPRINT,403) I,RNORML,VS,VSADD,VADSKP,
* IFCPSV,IFCINC,INCFAC,ICHECK
IFR = 1
DO 50 J=1,NMFRCP
C SHIFT THE POSITIVE MASK FREQUENCIES
VM = V1*FRINCR
SHFREQ = VS + VM
IF( (SHFREQ.GE.VCOBJ) .OR. (ABS(FSAMSK(IFR)).LT..001) ) GO TO 40
KOUNT = KOUNT + 1
TERM(KOUNT,1) = FACTOR*FSAMSK(IFR)
TERM(KOUNT,2) = VM
C SHIFT THE NEGATIVE MASK FREQUENCIES
40 SHFREQ = VS - VM
IF(ABS(FSAMSK(IFR)).LT..001) GO TO 45
IF(ABS(SHFREQ).GE.VCOBJ) GO TO 45
KOUNT = KOUNT + 1
TERM(KOUNT,1) = FACTOR*FSAMSK(IFR)
TERM(KOUNT,2) = -VM
45 FRINCR = FRINCR + 1.
IFR = IFR + 1
50 CONTINUE
IF(IMGFL(4).NE.1) GO TO 51
WRITE(IPRINT,404) (TERM(I,1),I=1,KOUNT)
WRITE(IPRINT,404) (TERM(I,2),I=1,KOUNT)

C CALCULATE THE INTENSITY PATTERN FOR THE HORIZONTAL POINTS BY
C MULTIPLYING ALL COMBINATIONS OF FREQUENCY TERMS PRODUCED BY THE
C PRESENT COHERENCE AREA.
51 DO 53 IHPT=1,NMHPTS
X = FLOAT(IHPT-1)*DELTX + WNDORG
AL = (0.,0.)
DO 52 K=1,KOUNT
ARG = -TERM(K,2)*X
AL = AL + CMPLX(TERM(K,1),0.)*CEXP( (0.,.6.2832)*CMPLX(ARG,0.) )
52 CONTINUE
THORIN(IHPT) = REAL(AL*CONJG(AL)) + THORIN(IHPT)
53 CONTINUE
VADSKP = 0.

60 IFCPSV = IFCPSV + INCFAC*IFCINC
INCFAC = -INCFAC
IFCINC = IFCINC + 1
IF(MOD(I,2).NE.0) GO TO 62
61 VSADD = VS3
GO TO 65
62 VSADD = VS2

65 VSTEMP = VS + VSADD
IF(VSTEMP.LE.VCCOND) GO TO 99
VSADD = VCCOND - VS

```

```

VS = VCCOND*.99999
IENDFL = 1
IF(VSADD.LT..001) GO TO 199
GO TO 100
99 VS = VSTEMP

```

```
100 CONTINUE
```

C ACCUMULATE THE INTENSITY IN HORINT.

```

199 DO 200 IHPT=1,NMHPTS
HORINT(IHPT) = HORINT(IHPT) + RELINT(ILMBD)*THORIN(IHPT)
200 CONTINUE

```

```
RETURN
```

```

401 FORMAT(1X,4HVS1=,F8.4,6H VS2=,F8.4,5H VS3=,F8.4,9H IFCPSV=,I2)
402 FORMAT(/1X,2HI=,I3,/10X,22HSKIP CALCULATION: VS=,F7.5,
*8H VSADD=,F7.5,9H VADSKP=,F7.5,11H IFCPSV =,I2,
*10H IFCINC =,I2,10H INCFAC =,I2,11H ICHECK =,I2)
403 FORMAT(/1X,2HI=,I3,/5X,7HRNORML=,F7.5,5H VS=,F7.5,8H VSADD=,
*F7.5,9H VADSKP=,F7.5,8H IFCPSV=,I2,8H IFCINC=,I3,8H INCFAC=,I2,
*8H ICHECK=,I2)
404 FORMAT(1X,I2(F10.5))

```

```
END
```

SUBROUTINE OUTPAT(ILMBD)

```

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMBDA(10),RELINT(10)

```

C STORE THE MTF IN HORINT(ALL LAMBDA) AND IN THORIN(CURRENT LAMBDA).

```

5 DO 10 IHPT=1,NMHPTS
VALMTF = FSAIMG(1)
X = FLOAT(IHPT-1)*DELT X + WNDORG
DO 20 IFRCP=2,NMFRCP
RN = FLOAT(IFRCP-1)
TEMP = COS(6.2832*V1*RN*X)
VALMTF = FSAIMG(IFRCP)*TEMP + VALMTF
20 CONTINUE
IF(ICOHER.EQ.IFULCO) VALMTF = VALMTF*VALMTF
THORIN(IHPT) = VALMTF
HORINT(IHPT) = HORINT(IHPT) + RELINT(ILMBD)*VALMTF
10 CONTINUE

```

```

RETURN
END

```

SUBROUTINE PLTPAT(ILMBD)

```

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /IMGFLG/ IMGFL(5)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IO1 / ITERML,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMBDA(10),RELINT(10)
COMMON IPLT(123,99),X(450),Z(450)
DIMENSION XPT(50)
DIMENSION IALPHA(6)
DATA IALPHA /1HA,1HB,1HC,1HD,1HE,1HF/
DATA KSTAR,KMINUS,KPOINT,KBLANK,K0,K1,K3
* /1H*,1H-,1H.,1H,1H0,1H1,1H3/

```

```

C IPLT(2,2) IS (X,INTENSITY) = (0.,1.3)
C IPLT(2,93) IS (X,INTENSITY) = (0.,0.)
IFLG = 0
IF(ILMBD.NE.1)GO TO 10

```

```

XLTH = DELTX*FLOAT(NMHPTS-1)
DO 2 L=2,99
DO 2 K=1,123
IPLT(K,L) = KBLANK
2 CONTINUE
IO = INT( (EDGE/XLTH)*120. + 2.00001 )
DO 3 K=1,123
IPLT(K,1) = KSTAR
IPLT(K,94) = KSTAR
IPLT(K,23) = KMINUS
3 CONTINUE
DO 4 L=2,93
IPLT(1,L) = KSTAR
IPLT(123,L) = KSTAR
IPLT(10,L) = KPOINT
4 CONTINUE
IPLT(10,93) = K0
IPLT(10-1,2) = K1
IPLT(10+1,2) = K3
IPLT(10,23) = K1

10 DO 12 IHPT=1,NMHPTS
RNK = FLOAT(IHPT-1)*DELTX
NX = INT( (RNK/XLTH)*120. + 2.0001 )
NZ = 95-INT( (THORIN(IHPT)/1.3)*91. + 2.0001 )
C STORE THE VALUES TO BE PUNCHED ON CARDS.
IPLT = (ILMBD-1)*60 + IHPT
X(IPLT) = RNK + WNDORG
Z(IPLT) = THORIN(IHPT)
IF((NZ.GE.95).OR.(NZ.LE.0)) GO TO 11
IPLT(NX,NZ) = IALPHA(ILMBD)
GO TO 12
11 IFLG = 1
12 CONTINUE
IF(NMLMBD.EQ.1) GO TO 15
IF(ILMBD.NE.NMLMBD)RETURN
DO 14 IHPT=1,NMHPTS
NX = INT( (FLOAT(IHPT-1)*DELTX/XLTH)*120. + 2.0001 )
NZ = 95-INT( (HORINT(IHPT)/1.3)*91. + 2.0001 )
IF((NZ.GE.95).OR.(NZ.LE.0))GO TO 13
IPLT(NX,NZ) = IALPHA(ILMBD+1)
GO TO 14
13 IFLG = 1
14 CONTINUE
15 WRITE(IPRINT,16)
16 FORMAT(/////45X,34H--INTENSITY PATTERN FROM IMAGE--)
WRITE(IPRINT,18) (IALPHA(ILMBD),RLAMBDA(ILMBD),RELINT(ILMBD),
* ILMBD=1,NMLMBD)
18 FORMAT(/,20X,44HSYMBOL WAVELENGTH RELATIVE INTENSITY,
* /,(22X,A1,7X,F6.4,8H MICRONS,7X,F6.4) )
IF(ICOHER.EQ.IFULCO) WRITE(IPRINT,19)
19 FORMAT(/20X,21HCOHERENT ILLUMINATION)
IF(ICOHER.EQ.IPARCO) WRITE(IPRINT,20) SIGMA
20 FORMAT(/20X,27HPARTIAL COHERENCE: SIGMA =,F7.2)
IF(ICOHER.EQ.IINCOH) WRITE(IPRINT,21)
21 FORMAT(/20X,23HINCOHERENT ILLUMINATION)
WRITE(IPRINT,22) DFDIST
22 FORMAT(/20X,11HDEFOCUS BY ,F5.2,8H MICRONS)
IF(IFLG.EQ.1) WRITE(IPRINT,23)
23 FORMAT(/20X,44HINTENSITY PATTERN IS OUTSIDE PLOT AT POINTS.)
WRITE(IPRINT,25) XLTH,EDGE,IPLT
25 FORMAT(/,20X,10HX WINDOW =,F7.4,14H MICRONS IN X.,5X,
*11HTHE EDGE IS,F7.4,39H MICRONS FROM THE LEFT WINDOW BOUNDARY.,
*/1HX/, (5X,I23A1) )
IF(NMLMBD.EQ.1)GO TO 30
WRITE(IPRINT,27) IALPHA(NMLMBD+1)
27 FORMAT(45X,A1,31H IS COMPOSITE INTENSITY PATTERN)
30 WRITE(IPRINT,31) NMHPTS
31 FORMAT(/,32X,49HINTENSITY VS X VALUES FOR THE COMPOSITE PATTERN--
* ,12.7H POINTS)
DO 32 I=1,NMHPTS
XPT(I) = DELTX*FLOAT(I-1) - EDGE
32 CONTINUE

C OUTPUT X AND I(X)--THE CODE IS VALID FOR ANY VALUE OF MNHPTS, AS
C LONG AS MNHPTS >= NMHPTS.
NTIMES = NMHPTS/18 + 1
IF((MOD(NMHPTS,18).EQ.0).AND.(NTIMES.NE.1)) NTIMES=NTIMES-1
JBEGIN = 1
JEND = 18
DO 36 J=1,NTIMES

```

```

IF(NTIMES.NE.J) GO TO 35
JBEGIN = (NTIMES-1)*18 + 1
JEND = NMHPTS
35 WRITE(IPRINT,33) (XPT(JJ) , JJ=JBEGIN,JEND)
WRITE(IPRINT,34) (HORINT(JJ) , JJ=JBEGIN,JEND)
33 FORMAT(/3X,3HX :.18(F6.3,1X))
34 FORMAT(1X,5HI(X):.18(F6.3,1X))
JBEGIN = JBEGIN + 18
JEND = JEND + 18
36 CONTINUE

40 IF(IMGFL(3).EQ.0)RETURN
C PUNCH OUT CARDS FOR HP PLOTTER.
XL = WNDORG
XR = XLTH+WNDORG
ZT = 1.3
ZB = 0.0
WRITE(IPUNCH,50) XL,XR,ZB,ZT
50 FORMAT(/,1X,4(F8.5,1X))
RNPT = FLOAT(NMHPTS)
RMLMBD = FLOAT(NMLMBD)
WRITE(IPUNCH,55) RMLMBD
55 FORMAT(1X,F8.5)
DO 70 IL = 1,NMLMBD
WRITE(IPUNCH,55) RNPT
ISTART = (IL-1)*60 + 1
ISTOP = ISTART + NMHPTS - 1
WRITE(IPUNCH,60) ( X(I),Z(I), I=ISTART,ISTOP )
60 FORMAT(11(1X,F6.3))
70 CONTINUE
CALL IMGMSG(6)

RETURN
END

```

SUBROUTINE PLTOTF

```

COMMON /IMGFLG/ IMGFL(5)
COMMON /IMG2PR/ RNA,MXNWTF,NMWTF,RTMFWT(41)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /IO1 / ITERM1,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)
COMMON IPLT(123,99),X(450),Z(450)
DIMENSION VM(5)
DIMENSION IALPHA(6)
DATA IALPHA /1HA,1HB,1HC,1HD,1HE,1HF/
DATA KSTAR,KMINUS,KPOINT,KBLANK,K0,K1,K3
  /1H*,1H-,1H.,1H ,1H0,1H1,1H3/

```

C PLTOTF PLOTS THE OTF FOR ALL THE CASES. THE ROUTINE TAKES MORE TIME
 C THAN THE REGULAR CALCULATIONS BECAUSE MORE THAN THE PASSED FREQUENCIES
 C ARE CALCULATED.
 C NOTE (X,Z) = (0.,1.) IS (NX=31.5,NZ=2.)

C STORE THE CUTOFF FREQUENCIES FOR EACH WAVELENGTH IN VM(ILMBD).

```

VMMAX = VMAX
DO 1 ILMBD=1,NMLMBD
VM(ILMBD) = VMAX*(RLAMB(NMLMBD)/RLAMB(ILMBD))
IF(ILMBD.EQ.1)GO TO 1
VMMAX = AMAX1(VM(ILMBD),VM(ILMBD-1))
1 CONTINUE
DO 2 L=2,99
DO 2 K=2,123
IPLT(K,L) = KBLANK
2 CONTINUE
DO 3 K=1,63
IPLT(K,1) = KSTAR
IPLT(K,51) = KSTAR
IPLT(K,26) = KSTAR
3 CONTINUE
DO 4 L=1,51
IPLT(1,L) = KSTAR
IPLT(63,L) = KSTAR
4 CONTINUE

```

C CREATE THE MTF PLOT ONE POINT(ALL WAVELENGTHS) AT A TIME.
 DO 12 NX=2,62,2

```

      TMPCOM = 0.
      FPOS = FLOAT(NX-2)*VMMA/60.
      DO 11 ILMBD=1,NMLMBD
      TEMP = 0.
      IF(FPOS.GE.VM(ILMBD))GO TO 10
C CHECK FOR THE TYPE OF COHERENCE.
      IF(ICOHER.EQ.IFULCO) GO TO 5
      IF(ICOHER.EQ.IPARCO) GO TO 6
      IF(ICOHER.EQ.IINCOH) GO TO 7
      5 TEMP = 1.
      GO TO 10
      6 WRITE(IPRINT,1000)
      1000 FORMAT(//////,15X,78HERROR: CALLING FOR AN MTF PLOT FOR THE PARTI
      *AL COHERENCE CASE MAKES NO SENSE.)
      RETURN
      7 IF(DFDIST.NE.0.0) GO TO 8
      ARG = FPOS/VM(ILMBD)
      TEMP = .63662*(ACOS(ARG)-(ARG)*SQRT(1.-(ARG)**2))
      GO TO 10
      8 DELTA = 2.*RNA*RNA*DFDIST/RLAMB(D(ILMBD)
      VR = .5*FPOS*RLAMB(D(ILMBD)/RNA
      DU = (1.-VR)/10.
      U = VR + DU/2.
      DO 9 ITER=1,10
      TEMP = TEMP+1.2732*SQRT(1.-U*U)*COS(6.2832*VR*DELTA*(U-VR))*DU
      U = U + DU
      9 CONTINUE
      10 IF(TEMP.GT.0.) NZ=27-INT(TEMP*24.+1.5)
      IF(TEMP.LE.0.) NZ=25-INT(TEMP*24.-1.5)
      IPLT(NX,NZ) = IALPHA(ILMBD)
C STORE THE MTF TO BE PUNCHED ON CARDS.
      IPLOT = (ILMBD-1)*60 + NX/2
      X(IPLOT) = FPOS
      Z(IPLOT) = TEMP
      11 CONTINUE
      12 CONTINUE
      IPLT(2,26) = K0
      IPLT(1,2) = K1
      IPLT(1,62) = K1

      WRITE(IPRINT,15)
      15 FORMAT(//////,20X,32H--THEORETICAL MTF FROM IMAGE--
      WRITE(IPRINT,18) (IALPHA(ILMBD),(RLAMB(D(ILMBD),RELINT(ILMBD),
      * ILMBD=1,NMLMBD)
      18 FORMAT(/,20X,44HSYMBOL WAVELENGTH RELATIVE INTENSITY,
      * /,(22X,A1,7X,F6.4,8H MICRONS,7X,F6.4) )
      IF(ICOHER.EQ.IFULCO) WRITE(IPRINT,19)
      19 FORMAT(/20X,21HCOHERENT ILLUMINATION)
      IF(ICOHER.EQ.IINCOH) WRITE(IPRINT,21)
      21 FORMAT(/20X,23HINCOHERENT ILLUMINATION)
      WRITE(IPRINT,22) DFDIST
      22 FORMAT(/20X,11HDEFocus BY ,F5.2,8H MICRONS)
      WRITE(IPRINT,25) VMMA,((IPLT(I,J),I=1,63),J=1,51)
      25 FORMAT(/29X,17HFREQ SPAN(VMAX) =,F6.2,8H CYC/MIC,
      * /1HX,/,10X,63A1))

      40 IF(IMGFL(3).EQ.0)GO TO 100
C PUNCH OUT CARDS FOR THE HP PLOTTER.
      XL = -.05
      XR = VMMA
      ZT = 1.05
      ZB = -1.05
      WRITE(IPUNCH,50) XL,XR,ZB,ZT
      50 FORMAT(/,1X,4(F8.4,1X))
      RNPT = 31.
      RMLMBD = FLOAT(NMLMBD)
      WRITE(IPUNCH,55) RMLMBD
      55 FORMAT(1X,F8.5)
      DO 70 IL=1,NMLMBD
      WRITE(IPUNCH,55) RNPT
      ISTART = (IL-1)*60 + 1
      ISTOP = ISTART + 30
      WRITE(IPUNCH,60) ( X(I),Z(I),I=ISTART,ISTOP )
      60 FORMAT(11(1X,F6.3))
      70 CONTINUE
      CALL IMGMSG(6)

      100 RETURN
      END

```

SUBROUTINE IMGMSG(NUMB)

```

COMMON /CBWIND/ WINDOW,EDGE,WNDORG
COMMON /FOUSER/ MXNMFR,NMFRCP,FSAMSK(41),FSAIMG(41)
COMMON /IMG2PR/ RNA,MXNWF,NMWTFC,RTFWT(41)
COMMON /IMG3PR/ IINCOH,IPARCO,IFULCO,ICOHER,SIGMA,DFDIST
COMMON /IMGFLG/ IMGFL(5)
COMMON /IO1 / ITERMI,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /OBJMSK/ MLINE,MSPACE,MLNSPA,MASKTY,RLW,RSW
COMMON /OPTIC / CURWL,V1,VMAX,THORIN(50)
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)

```

C IMGMSG IS THE MESSAGE SUBROUTINE FOR THE IMAGE MACHINE.

GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NUMB

```

1 WRITE(IPRINT,101) ( RLAMB(I),RELINT(I),RNA), I=1,NMLMBD )
IF(MASKTY.EQ.MLNSPA) WRITE(IPRINT,102) RLW,RSW
IF(MASKTY.EQ.MLINE) WRITE(IPRINT,103)RLW
IF(MASKTY.EQ.MSPACE) WRITE(IPRINT,104)RSW
IF(ICOHER.EQ.IFULCO) WRITE(IPRINT,105) DFDIST
IF(ICOHER.EQ.IPARCO) WRITE(IPRINT,106) SIGMA,DFDIST
IF(ICOHER.EQ.IINCOH) WRITE(IPRINT,107) DFDIST
WRITE(IPRINT,108) WINDOW,EDGE
RETURN
2 WRITE(IPRINT,2000)
WRITE(IPRINT,20)
STOP
3 WRITE(IPRINT,3000)
WRITE(IPRINT,30) CURWL,NMFRCP
RETURN
4 WRITE(IPRINT,1000)
WRITE(IPRINT,40) RLW,RSW
RETURN
5 WRITE(IPRINT,1000)
WRITE(IPRINT,50) RSW,RLW
RETURN
6 WRITE(IPRINT,1000)
WRITE(IPRINT,60)
RETURN
7 WRITE(IPRINT,3000)
WRITE(IPRINT,70) SIGMA
RETURN
8 RETURN
9 WRITE(IPRINT,3000)
WRITE(IPRINT,90) NMFRCP,CURWL
RETURN
10 WRITE(IPRINT,2000)
WRITE(IPRINT,100)
STOP
11 IF(IMGFL(4).EQ.0) RETURN
WRITE(IPRINT,1000)
IF(ICOHER.EQ.IPARCO)GO TO 113
WRITE(IPRINT,110) NMFRCP,CURWL
DO 111 I=1,NMFRCP
J=I-1
WRITE(IPRINT,112) FSAMSK(I),FSAIMG(I),J,V1
111 CONTINUE
RETURN
113 WRITE(IPRINT,114) NMFRCP,CURWL
DO 115 I=1,NMFRCP
J=I-1
WRITE(IPRINT,116) FSAMSK(I),J,V1
115 CONTINUE
RETURN
12 WRITE(IPRINT,3000)
WRITE(IPRINT,120)
RETURN
13 WRITE(IPRINT,3000)
WRITE(IPRINT,130)
RETURN
14 IF(IMGFL(4).EQ.0) RETURN
WRITE(IPRINT,2000)
WRITE(IPRINT,140)
STOP
15 WRITE(IPRINT,1000)
DEPFLD = .5*CURWL/(RNA*RNA)
WRITE(IPRINT,150) DEPFLD,CURWL
RETURN
16 RETURN

```

```

101 FORMAT(///,SX,15(1H*),/5X,15H* RUN IMAGE */5X,15(1H*),
* ///,32X,23HIMAGE PARAMETER VALUES:,,/20X,
*56HWAVELENGTH RELATIVE INTENSITY NUMERICAL APERTURE,/,
* (18X,F5.4,8H MICRONS,10X,F6.4,17X,F5.4) )
102 FORMAT(/1X,36HPATTERN IS PERIODIC WITH LINEWIDTH =,F7.4,
* 8H MICRONS,17H AND SPACEWIDTH =,F7.4,8H MICRONS)
103 FORMAT(/1X,36HPATTERN IS PERIODIC WITH LINEWIDTH =,F7.4,
* 8H MICRONS)
104 FORMAT(/1X,37HPATTERN IS PERIODIC WITH SPACEWIDTH =,F7.4,
* 8H MICRONS)
105 FORMAT(1X,40HMASK IS COHERENTLY ILLUMINATED -----
*19HTHE IMAGE PLANE IS ,F5.2,32H MICRONS FROM THE IN-FOCUS PLANE)
106 FORMAT(/1X,49HMASK IS ILLUMINATED WITH PARTIALLY COHERENT LIGHT,
*9H-SIGMA =,F4.2,2X,4H---,2X,
*19HTHE IMAGE PLANE IS ,F5.2,30H MICRONS FROM THE FOCAL PLANE.,
*/1X,47HSQUARE OBJECTIVE, INFINITE(OR LINE IN X) SOURCE)
107 FORMAT(1X,42HMASK IS INCOHERENTLY ILLUMINATED -----
*19HTHE IMAGE PLANE IS ,F5.2,33H MICRONS FROM THE IN-FOCUS PLANE.
*/1X,35HCIRCULAR OBJECTIVE, INFINITE SOURCE)
108 FORMAT(1X,20HINTENSITY WINDOW IS ,F7.4,19H MICRONS WIDE --- ,
*21HMASK EDGE IS LOCATED ,F7.4,29H MICRONS FROM THE LEFT WINDOW,
* 9H BOUNDARY)
20 FORMAT(15X,59HUNRECOGNIZABLE PATTERN---MUST BE LINE, SPACE, OR LIN
*SPACE.)
30 FORMAT(15X,97HMAXIMUM NUMBER OF FREQUENCY COMPONENTS IN GENERATING
* THE MASK HAS BEEN EXCEEDED FOR WAVELENGTH = ,F5.4,11H MICRONS---,
*/20X,13,24H FREQUENCIES ARE NEEDED.)
40 FORMAT(15X,17HMASK PATTERN IS A,F8.4,6H LINE.,
* /15X,25HTHE NEXT NEAREST LINE IS ,F9.4,14H MICRONS AWAY.)
50 FORMAT(15X,17HMASK PATTERN IS A,F8.4,7H SPACE.,
* /15X,26HTHE NEXT NEAREST SPACE IS ,F9.4,14H MICRONS AWAY.)
60 FORMAT(25X,38HCARDS WERE PUNCHED FOR THE HP PLOTTER.)
70 FORMAT(3X,8HSIGMA = ,F4.2, 113H, INDICATING SOME COHERENCE. THE
* DEFOCUS ROUTINE HAS BEEN CALLED AND ASSUMES COMPLETE INCOHERENCE
*OF THE SOURCE.)
90 FORMAT(1X,52HTHE LENS IS CLOSE TO PASSING ONE MORE FREQ.--TO SEE.
*54H THE EFFECT OF THE ADDITIONAL COMPONENT, ADD 5 PERCENT,
*/10X,41H TO THE NUMERICAL APERTURE AND RUN AGAIN.,
*/1X,41HTHE PRESENT NO. OF SPATIAL FREQ. USED IS ,I2,
*23H: CURRENT WAVELENGTH IS ,F6.4,8H MICRONS)
100 FORMAT(15X,62HUNRECOGNIZABLE IMAGING SYSTEM---MUST BE PROJECTION O
*R CONTACT.)
110 FORMAT(10X, I2,56H FREQUENCY COMPONENTS CONSTITUTE THE MASK
*AT WAVELENGTH ,F5.4, 8H MICRONS)
112 FORMAT(5X,20HFREQ. WT. OF MASK = ,F6.4,
* 26H FREQ. WT. OF IMAGE = ,F6.4,13H : FREQ. = ,I2,
* 1H*,F5.4,8H CYC/MIC)
114 FORMAT(5X,I2, 83H FREQUENCY COMPONENTS OF THE MASK ARE USED IN CON
*STRUCTING THE IMAGE AT WAVELENGTH ,F5.4,8H MICRONS)
116 FORMAT(20X,12HFREQ. WT. = ,F6.4, 13H : FREQ. = ,I2,
* 1H*,F5.4,8H CYC/MIC)
120 FORMAT(20X,47HTHE EDGE IS IN THE FIRST QUARTER OF THE WINDOW.)
130 FORMAT(20X,46HTHE EDGE IS IN THE LAST QUARTER OF THE WINDOW.)
140 FORMAT(5X,46HTHE ACCEPTANCE VALUE OF SIGMA FOR THE PARTIAL ,
*54HCOHERENCE ROUTINES HAS BEEN ARTIFICIALLY LIMITED TO 3./5X,
*31HPLEASE RUN THE INCOHERENT CASE.)
150 FORMAT(10X,32HTHE INCOHERENT DEPTH OF FIELD IS.F6.2,
*24H MICRONS FOR WAVELENGTH ,F5.4)

1000 FORMAT(///,20X, 43H----- SYSTEM MESSAGE(IMAGE) -----,/)
2000 FORMAT(///,20X,40H----- FATAL ERROR(IMAGE) -----,/)
3000 FORMAT(///,20X,36H----- WARNING(IMAGE) -----,/)

```

END

SUBROUTINE EXPOSE

```

COMMON /EXPINT/ PWRBGN(10),PWREND(10),KFLAG
COMMON /EXPFLG/ IEXPFL(5)
COMMON /EXPTBL/ EXPOS(21), RMZDOS(51,21)
COMMON /IO1 / ITERMLIBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /RUCOMP/ IRUMZD
COMMON /SIMPAN/ NPRLYR, NPRPTS, NENDIV, DELTM, DELTZ

```

C EXPOSE COMBINES FORMER MACHINES 2 AND 3 AND PERFORMS BLEACHING
C AND EXPOSURE.
C IRUMZD IS INITIALIZED TO 0 IN THE CONTROLLER AND DECIDES
C IF SUBROUTINE CLCMZD (FORMERLY RUNM2) IS CALLED.

C IRUMZD IS SET=1 UPON LEAVING THE EXPOSE MACHINE.
 C THE CONTROLLER RESETS IRUMZD TO 0 IF RELEVANT INPUT DATA IS CHANGED.
 C KFLAG IS USED TO RECALCULATE RMZDOS(.) IF REQUESTED EXPOSURES ARE
 C OUTSIDE OF THE TABLE.

```

    IF(IRUMZD.EQ.0) GO TO 1
    CALL EXPMSG(1)
    CALL EXPMSG(2)
    GO TO 3
  1 NFLG = 0
    NENDIV = 15
    DELTM = .7/FLOAT(NENDIV-1)
    CALL CLCMZD
    CALL EXPMSG(1)
    CALL EXPMSG(2)

    3 KFLAG = 0
    CALL CLCMXZ
  C RETURN TO THE CONTROLLER IF THE Z-DOSE TABLE WAS SUFFICIENTLY LARGE.
    IF(KFLAG.EQ.0) GO TO 10
    CALL EXPMSG(7)
  C WHEN DOSES ARE OUTSIDE OF THE EXPOSURE(Z-DOSE) TABLE.
  C THE MAXIMUM OF THE TABLE IS EXTENDED AND SUBROUTINE
  C CLCMZD IS RECALLED TO CALCULATE A NEW TABLE.
    ROLD=DELTM*FLOAT(NENDIV-1)
    RNEW=ROLD+0.175
    IF(RNEW.LE..95) GO TO 4
  C IF THE TABLE HAS BEEN EXPANDED AS FAR AS IT CAN BE, CALL FATAL ERROR.
    IF(NFLG.EQ.1) CALL EXPMSG(10)
    RNEW = .98
    NFLG = 1
  C THE NUMBER OF ENERGY DIVISIONS IS INCREASED SO THAT ACCURACY
  C WILL NOT BE SACRIFICED BY A DELTM THAT IS TOO LARGE.
    4 NENDIV=NENDIV+3
    IF(NENDIV.GT.21)NENDIV=21
    DELTM=RNEW/FLOAT(NENDIV-1)
  C RETURN TO CLCMZD TO GENERATE AN EXPANDED TABLE.
    CALL CLCMZD
    CALL EXPMSG(2)
    GOTO 3

  10 CALL EXPMSG(4)
    CALL EXPMSG(3)
    CALL EXPMSG(5)
    CALL EXPMSG(6)
    CALL EXPMSG(8)
    WRITE(IPRINT,22222)
  22222 FORMAT(1H1)
    IRUMZD = 1
    RETURN

  END

```

SUBROUTINE CLCMZD

```

COMMON /EXPTBL/ EXPOS(21),RMZDOS(51,21)
COMMON /EXPINT/ PWRBGN(10),PWREND(10),KFLAG
COMMON /OPTCHP/ RINDEX(10,5,2)
COMMON /RESPAR/ WLABC(10,4)
COMMON /SIMPAR/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMMON /SPECTR/ MNLMBD,NMLMBD,RLAMB(10),RELINT(10)
COMMON /THKNES/ MNLYRS,NMLYRS,THICK(6)
DIMENSION SLBNDX(10,56),THIC(55),PJDJM1(54),EXINC(21)
DIMENSION CUENAB(51,21)
COMPLEX SLBNDX,TEMPC1,TEMPC2,TEMPC3,TEMPC4,RJMI,TJMI,FJ,PHIJ1,
* PHIJ2,WL02,RJ,TJ

  C CLCMZD IS THE BLEACHING SUBROUTINE.
    DELTZ = THICK(1)/FLOAT(NPRLYR)
  C INITIALIZE SLBNDX, THIC, AND RINDEX FROM THE INFORMATION GIVEN.
    DO 5 ILMBD=1,NMLMBD
    RINDEX(ILMBD,1,2) = -( WLABC(ILMBD,2)+WLABC(ILMBD,3) ) *
    * WLABC(ILMBD,1)/12.5664
    DO 1 IZPOS=2,NPRPTS
    THIC(IZPOS) = DELTZ
    SLBNDX(ILMBD,IZPOS) = CMPLX(RINDEX(ILMBD,1,1),RINDEX(ILMBD,1,2))

```

```

1 CONTINUE
  NM = NMLYRS-1
  DO 2 ILYR=2,NM
    THIC(NPRLYR+ILYR) = THIC(ILYR)
2 CONTINUE
  SLBNDX(ILMBD,1) = (1.,0.)
  DO 3 ILYR = 2,NMLYRS
    J = NPRLYR + ILYR
    SLBNDX(ILMBD,J) = CMPLX(RINDEX(ILMBD,ILYR,1),RINDEX(ILMBD,ILYR,2))
3 CONTINUE
C J IS THE SUBSTRATE LAYER AT THIS POINT.
5 CONTINUE

C INITIALIZE CUENAB ARRAY AND PARTS OF RMZDOS.
9 DO 18 IENDIV=1,NENDIV
  DO 17 IZPOS=2,NPRPTS
    CUENAB(IZPOS,IENDIV) = 0.
17 CONTINUE
18 CONTINUE
  DO 19 IZPOS=2,NPRPTS
    RMZDOS(IZPOS,1) = 1.
19 CONTINUE

  JSAVE = J
C CALCULATE THE VERTICAL M VALUES AS A FTM. OF EXPOSURE ENERGY.

C FIRST LINEARIZATION OF M SECTION.
  TEMPEX = 0.
  DO 12 ILMBD=1,NMLMBD
    TEMPEX = TEMPEX - (RELINT(ILMBD)*WLABC(ILMBD,4))*
      * EXP(-(WLABC(ILMBD,2) + WLABC(ILMBD,3))*THIC(1))
12 CONTINUE
  RM = 1.
  EXPOS(1) = 0.
  EXINC(1) = 0.

  DO 90 IENDIV=2,NENDIV

C RUN MULTIPLE WAVELENGTH EXPOSURE FOR THE PRESENT CUENAB INCREMENT.
  DO 45 ILMBD=1,NMLMBD
    WLOP = WLABC(ILMBD,1)/6.2832
    WL02 = CMPLX(0.,-2./WLOP)
    J = JSAVE
    RJ = ((1.,0.)-SLBNDX(ILMBD,J))/((1.,0.)+SLBNDX(ILMBD,J))
    TJ = (CMPLX(2.*SQRT(REAL(SLBNDX(ILMBD,J))),0.)) /
      * ((1.,0.) + SLBNDX(ILMBD,J))
    J = J - 1
  C NOW J IS THE FIRST LAYER ON TOP OF THE SUBSTRATE.
    JM1 = J-1

    DO 30 KOUNT=1,JM1
      TEMPC4 = SLBNDX(ILMBD,J)*CMPLX(THIC(J),0.)
      PHIJ1 = CEXP((WL02/(2.,0.))*TEMP4)
      PHIJ2 = CEXP(WL02*TEMP4)
      FJ = ((1.,0.)-SLBNDX(ILMBD,J))/((1.,0.)+SLBNDX(ILMBD,J))
      TEMPC1 = FJ - RJ
      TEMPC2 = (1.,0.) - FJ*RJ
      TEMPC3 = FJ*PHIJ2*TEMP1-TEMP2
      RJM1 = (PHIJ2*TEMP1-FJ*TEMP2)/TEMP3
      TJM1 = (FJ*FJ-(1.,0.))*TJ*PHIJ1/TEMP3
      TJM1S = (REAL(TJM1))**2 + (AIMAG(TJM1))**2
      RJM1S = (REAL(RJM1))**2 + (AIMAG(RJM1))**2
      TJS = (REAL(TJ))**2 + (AIMAG(TJ))**2
      RJS = (REAL(RJ))**2 + (AIMAG(RJ))**2
      PJDJM1(J) = (TJM1S/TJS)*(1.-RJS)/(1.-RJM1S)
    C ENDS HERE AT J=2---PJDJM1(1) REMAINS UNINITIALIZED.
    J = J-1
    RJ = RJM1
    TJ = TJM1
30 CONTINUE
  C PWRATI = (POWER PRESENT AT J=1 INTERFACE)/(INCIDENT POWER)
  C FOR THE WAVELENGTH INFO STORED IN WLABC(1,I,J,K).
  C SAVE INITIAL POWER REFLECTION FOR PRINTOUT.
  IF(IENDIV.EQ.2) PWRBGN(ILMBD) = RJM1S
  IF(IENDIV.EQ.NENDIV) PWREND(ILMBD) = RJM1S
  IF(ILMBD.NE.1) GO TO 27
  C SECOND LINEARIZATION OF M SECTION.
  PWRATI = 1. - RJM1S
  EXINC(IENDIV) = (ALOG(RM-DELTM))/TEMPEX - EXPOS(IENDIV-1)/PWRATI
  EXPOS(IENDIV) = EXINC(IENDIV) + EXPOS(IENDIV-1)

```

```

      RM = RM - DELTM
      27 TEMPR1 = RELINT(ILMBD)*EXINC(IENDIV)
      TEMPR = 1.-RJMIS
      C CALCULATE CUMULATIVE ENERGY ABSORBED FOR THE PR LAYERS.
      DO 40 J=2,NPRPTS
      AJ = (1.-PJDJMI(J))*TEMPR
      TEMPR = TEMPR * PJDJMI(J)
      IF(ILMBD.NE.1)GO TO 29
      28 CUENAB(J,IENDIV) = CUENAB(J,IENDIV-1) + AJ*WLABC(ILMBD,4)*TEMPR1
      * / (DELTZ*(RMZDOS(J,IENDIV-1)*WLABC(ILMBD,2)+WLABC(ILMBD,3)))
      GO TO 40
      29 CUENAB(J,IENDIV) = CUENAB(J,IENDIV) + AJ*WLABC(ILMBD,4)*TEMPR1
      * / (DELTZ*(RMZDOS(J,IENDIV-1)*WLABC(ILMBD,2)+WLABC(ILMBD,3)))
      C NOTE THAT CUENAB(1,IENDIV) REMAINS UNINITIALIZED.

      40 CONTINUE

      45 CONTINUE

      C CALCULATE M AND UPDATE THE INDICES OF EACH PR LAYER.
      DO 50 IZPOS=2,NPRPTS
      RMZDOS(IZPOS,IENDIV) = EXP(-CUENAB(IZPOS,IENDIV))

      DO 55 ILMBD=1,NMLMBD
      TP = -(WLABC(ILMBD,2)*RMZDOS(IZPOS,IENDIV)+WLABC(ILMBD,3))
      * *WLABC(ILMBD,1)/12.5664
      SLBNDX(ILMBD,IZPOS) = CMPLX(REAL(SLBNDX(ILMBD,IZPOS)),TP)

      55 CONTINUE
      50 CONTINUE

      90 CONTINUE

      RETURN

      END

```

SUBROUTINE CLCMXZ

```

      COMMON /DOSPAR/ DOS
      COMMON /EXPINT/ PWRBGN(10),PWREND(10),KFLAG
      COMMON /EXPFLG/ IEXPFL(5)
      COMMON /EXPTBL/ EXPOS(21),RMZDOS(51,21)
      COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
      COMMON /MVSPOS/ RMXZ(52,52)
      COMMON /PREBAK/ DGRADM
      COMMON /SIMPAN/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
      DIMENSION HOREN(50)

      C CLCMXZ SETS M AS A FUNCTION OF X AND Z---INCLUDING A BOUNDARY LAYER.
      C THE RESIST POINT (X,Z)=(0.,0.) IS THE MIDDLE OF THE UPPER BOUNDARY OF
      C CELL RMXZ(2,2). THE RESIST POINT (XMAX,ZMAX) IS THE MIDDLE OF THE
      C LOWER BOUNDARY OF CELL RMXZ(NMHPTS+1,NPRPTS).

      C NOTE: HOREN CONSISTS OF THE SUM OF THE EFFECTIVE INTENSITIES OF
      C ALL THE WAVELENGTHS. = SUM-OVER-I( RELINT(I)*WLENERGY )

      IFLG = 0
      DO 5 IHPT=1,NMHPTS
      HOREN(IHPT) = HORINT(IHPT)*DOS
      5 CONTINUE

      DO 40 IHPT=1,NMHPTS
      IENDIV = 2
      10 IF(EXPOS(IENDIV).GT.HOREN(IHPT))GO TO 20
      IENDIV = IENDIV + 1
      IF(IENDIV.GT.NENDIV)GOTO 100
      GO TO 10
      20 FRAC = (HOREN(IHPT) - EXPOS(IENDIV-1)) /
      * (EXPOS(IENDIV) - EXPOS(IENDIV-1))
      DO 30 IZPOS=2,NPRPTS
      RMXZ(IHPT+1,IZPOS) = RMZDOS(IZPOS,IENDIV-1) +
      * FRAC*(RMZDOS(IZPOS,IENDIV)-RMZDOS(IZPOS,IENDIV-1))
      IF(RMXZ(IHPT+1,IZPOS).LT..4) IFLG=1
      30 CONTINUE
      40 CONTINUE

```

```

C FILL IN THE BOUNDARIES OF RMXZ
  NHP1 = NMHPTS + 1
  NHP2 = NMHPTS + 2
  NPP1 = NPRPTS + 1
C EXTRAPOLATE THE TOP AND BOTTOM ROWS.
C M VALUES LESS THAN .4 ARE TESTED FOR IN FUNCTION RATE.
  DO 60 IX=2,NHP1
    RMXZ(IX,1) = RMXZ(IX,2) + RMXZ(IX,2)-RMXZ(IX,3)
    IF(RMXZ(IX,1).GT.1.) RMXZ(IX,1)=1.
    RMXZ(IX,NPP1) = RMXZ(IX,NPRPTS) + RMXZ(IX,NPRPTS)-RMXZ(IX,NPRLYR)
    IF(RMXZ(IX,NPP1).GT.1.) RMXZ(IX,NPP1)=1.
  60 CONTINUE
C EXTRAPOLATE THE SIDE COLUMNS.
  DO 70 IZ=1,NPP1
    RMXZ(1,IZ) = RMXZ(2,IZ) + RMXZ(2,IZ)-RMXZ(3,IZ)
    IF(RMXZ(1,IZ).GT.1.) RMXZ(1,IZ)=1.
    RMXZ(NHP2,IZ) = RMXZ(NHP1,IZ) + RMXZ(NHP1,IZ)-RMXZ(NMHPTS,IZ)
    IF(RMXZ(NHP2,IZ).GT.1.) RMXZ(NHP2,IZ)=1.
  70 CONTINUE

C IF M IS DEGRADED BY POSTBAKE, MODIFY RMXZ.
  DGRADM = ABS(DGRADM)
  IF(DGRADM.LT..001)GO TO 90
  DO 85 IX=1,NHP2
    DO 80 IZ=1,NPP1
      TEMP = RMXZ(IX,IZ) - DGRADM
      IF(TEMP.LT.0.)GO TO 80
      RMXZ(IX,IZ) = TEMP
    80 CONTINUE
  85 CONTINUE

  90 IF(IFLG.EQ.1) CALL EXPMSG(8)
  RETURN

100 KFLAG=1
  RETURN

  END

```

SUBROUTINE EXPMSG(NUMB)

```

COMMON /DOSPAR/ DOS
COMMON /EXPFLG/ IEXPFL(5)
COMMON /EXPINT/ PWRBGN(10),PWREND(10),KFLAG
COMMON /EXPTBL/ EXPOS(21),RMZDOS(51,21)
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IO1 / ITERMI,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /MVSP0S/ RMXZ(52,52)
COMMON /OPTCHP/ RINDEX(10,6,2)
COMMON /PREBAK/ DGRADM
COMMON /RESPAR/ WLABC(10,4)
COMMON /SIMPAP/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMMON /SPECTR/ MNLMDB,NMLMBD,RLAMB(10),RELINT(10)
COMMON /THKNES/ MNLYRS,NMLYRS,THICK(6)

```

C EXPMSG IS THE ERROR SUBROUTINE FOR SUBROUTINE EXPOSE.

```

  GO TO (1,2,3,4,5,6,7,8,9,10),NUMB

1 WRITE(IPRINT,101) ((WLABC(I,J), J=1,4), I=1,NMLMBD)
  WRITE(IPRINT,102)
  N = NMLYRS - 1
  DO 104 J=1,N
    WRITE(IPRINT,103) ( WLABC(I,1),J,THICK(J),RINDEX(I,J,1),
      * RINDEX(I,J,2), I=1,NMLMBD )
  104 CONTINUE
  WRITE(IPRINT,105) ( WLABC(I,1),RINDEX(I,NMLYRS,1),
    * RINDEX(I,NMLYRS,2), I=1,NMLMBD )
  WRITE(IPRINT,106) DOS
  IF(DGRADM.NE.0.) WRITE(IPRINT,107) DGRADM
  RETURN
2 IF(IEXPFL(1).EQ.0) RETURN
  WRITE(IPRINT,1000)
  WRITE(IPRINT,20) (EXPOS(IZ), IZ=1,21)
  DO 202 IZPOS=2,NPRPTS
    WRITE(IPRINT,201) IZPOS,(RMZDOS(IZPOS,J),J=1,NENDIV)
  202 CONTINUE
  RETURN

```

```

3 WRITE(IPRINT,1000)
DO 301 ILMBD=1,NMLMBD
TEMP = .5*WLABC(ILMBD,1)/RINDEX(ILMBD,1,1)
WRITE(IPRINT,30) WLABC(ILMBD,1),TEMP
301 CONTINUE
RETURN
4 WRITE(IPRINT,1000)
DO 402 ILMBD=1,NMLMBD
WRITE(IPRINT,40) RLAMBDA(ILMBD),PWRBGN(ILMBD)
WRITE(IPRINT,401) EXPOS(NENDIV),PWREND(ILMBD)
402 CONTINUE
RETURN
5 WRITE(IPRINT,1000)
WRITE(IPRINT,50) DELTX,DELTZ
RETURN
6 IF(IEXPFL(2).EQ.0) RETURN
WRITE(IPRINT,1000)
WRITE(IPRINT,60)
N = NPRPTS + 1
DO 606 IZPOS=1,N
WRITE(IPRINT,601) IZPOS,(RMXZ(J,IZPOS),J=1,42,2)
606 CONTINUE
RETURN
7 WRITE(IPRINT,1000)
WRITE(IPRINT,70) EXPOS(NENDIV)
RETURN
8 WRITE(IPRINT,1000)
WRITE(IPRINT,80)
RETURN
9 RETURN
10 WRITE(IPRINT,2000)
WRITE(IPRINT,100)
STOP

101 FORMAT(///,5X,16(1H*),/5X,16H* RUN EXPOSE *,/5X,16(1H*),
* ///,32X, 19HRESIST PARAMETERS ://,
*( 8X,13HWAVELENGTH: ,F5.4,8H MICRONS,8X,4HA = ,F6.4,5H 1/CM,8X,
*4HB = ,F6.4,5H 1/CM,8X,4HC = ,F6.4,9H CM**2/MJ )
102 FORMAT(/,34X,17HCHIP PARAMETERS :/)
103 FORMAT(8X,13HWAVELENGTH: ,F5.4,8H MICRONS,8X,10H LAYER NO. ,I1,8X,
*12H THICKNESS = ,F6.4,8H MICRONS,8X,14H INDEX(A+IB) = ,F5.2,2X,F6.3)
105 FORMAT(8X,13HWAVELENGTH: ,F5.4,8H MICRONS,26X,9H SUBSTRATE,26X,
* 14H INDEX(A+IB) = ,F5.2,2X,F6.3)
106 FORMAT(/,30X,21HEXPOSURE PARAMETERS ://,8X, 6HDOSE = ,F5.1,
* 9H MJ/CM**2)
107 FORMAT(/10X,16HM IS DEGRADED BY,F6.3,16H DUE TO PREBAKE.)
20 FORMAT(25X, 71HMATRIX OF M VS. Z(VERTICAL) AND THE ENERGY DOSE I
*NCREMENTS(HORIZONTAL)..//,1X,5HDOSE:.,7X,21(F5.1),/)
201 FORMAT(1X,5HZPOS=,I2,5X,21(F5.3))
30 FORMAT(10X,38HPERIOD OF STANDING WAVE FOR WAVELENGTH,F6.4,
* 12H MICRONS IS ,F5.4,9H MICRONS.)
40 FORMAT(10X,38HINITIAL REFLECTED POWER AT WAVELENGTH ,F5.4,
* 12H MICRONS IS ,F4.3)
401 FORMAT(15X,42HREFLECTED POWER FOR END OF EXPOSURE TABLE(
* F6.1,15H MJ/CM**2 ) IS ,F4.3)
50 FORMAT(10X,25HSPACING BETWEEN X CELLS =,F7.4,8H MICRONS,
* 5H -- 25HSPACING BETWEEN Z CELLS =,F7.4,8H MICRONS)
60 FORMAT(10X,95HMATRIX OF M AS A FUNCTION OF (X,Z) FOR THE BOUNDARY
* LAYER AND EVERY OTHER GRID POINT(UP TO 40)../)
601 FORMAT(1X,5HZPOS=,I2,5X,21(F5.3))
70 FORMAT(//,10X,46H THE EXPOSURE IS OUTSIDE OF THE EXPOSURE TABLE.,
* /,10X,36H THE CURRENT MAXIMUM IN THE TABLE IS ,F5.1,
* 10H MJ/CM**2.,/,
* 10X,50H RECALCULATE THE EXPOSURE TABLE WITH A NEW MAXIMUM.)
80 FORMAT(10X,44HM IS SMALLER THAN .4 AT POINTS IN THE RESIST)

100 FORMAT(//,8X,47H THE EXPOSURE DOSE CANNOT BE ACCOMODATED IN THE
* 48HEXPOSURE TABLE--EVEN WITH TABLE RECALCULATIONS.)

1000 FORMAT(///,20X,44H----- SYSTEM MESSAGE(EXPOSE) -----,/)
2000 FORMAT(///,20X,41H----- FATAL ERROR(EXPOSE) -----,/)
3000 FORMAT(///,20X,37H----- WARNING(EXPOSE) -----,/)

```

END

SUBROUTINE DVELOP

COMMON /DEVFLG/ IDEVFL(S)

```

COMMON /DEVPAR/ E1,E2,E3
COMMON /DEVTIM/ MXNDEV,DEVSRT,DEVEND,DEVINC
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
COMMON /DVELP3/ NZFLG,TTOTSV
COMMON /DVELP4/ BREAK,MAXPTS,NADSAV,NCKSV1,NCKSV2,NOUT
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /IO1 / ITERMI,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /MVSP0S/ RMXZ(52,52)
COMMON /SIMPAN/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMPLEX XZ,CXZL,CXZR,CTZ

```

C DEVELOP IS THE SUB-CONTROLLER FOR THE PR DEVELOP ROUTINES.
C IDEVFL(3)=1 FOR PUBLICATION RUNS, WHICH ARE MORE COSTLY.

C THE WINDOW HAS NO EFFECT ON THE CODE FOR THE DEVELOP ROUTINES.
C IF ALL OF THE SLOPING PORTION OF THE INTENSITY PATTERN IS NOT
C INSIDE OF THE WINDOW, THE LINE EDGE PROFILE WILL BE IN ERROR.

```

CALL DEVMSG(1)
CALL DEVMSG(11)
IF(IDEVFL(5).EQ.1) GO TO 2
XMAX = DELTX*FLOAT(NMHPTS-1)
ZMAX = DELTZ*FLOAT(NPRLYR)
C NOTE THERE IS NO POSITIONAL CORRESPONDENCE BETWEEN THE HORIZONTAL
C IMAGE POINTS AND THE POINTS ON THE DEVELOPING STRING.

```

C BREAKTHRU ESTIMATION SECTION

```

JMIN = 2
RMMIN = RMXZ(2,2)
NMHP1 = NMHPTS + 1
DO 10 J=3,NMHP1
IF(RMMIN.LE.RMXZ(J,2)) GO TO 10
JMIN = J
RMMIN = RMXZ(J,2)
10 CONTINUE
XJMIN = DELTX*FLOAT(JMIN-2)
DELTTM = 4.
FACTOR = .6
TBREAK = 1.E18
KOUNT = 0
NCNVRG = 0
11 TBSAVE = TBREAK
KOUNT = KOUNT + 1
IF(KOUNT.GT.20) CALL DEVMSG(4)
IF( (KOUNT.NE.3).OR.(TBREAK.GE.15.) ) GO TO 13
FACTOR = .4
NCNVRG = 0
13 NCNVRG = NCNVRG + 1
TBREAK = 0.
DELTTM = FACTOR*DELTTM
Z = 0.
12 CTZ = CMPLX(XJMIN,Z)
Z = Z + DELTTM*RATE(CTZ)
TBREAK = TBREAK + DELTTM
IF(Z.LT.ZMAX) GO TO 12
IF( ABS(TBSAVE-TBREAK).GE.(.05*TBSAVE) ) NCNVRG = 0
IF(NCNVRG.EQ.4) GO TO 14
IF(NCNVRG.EQ.1) BREAK = TBREAK
GO TO 11
14 DELTTM = DELTTM/(FACTOR*FACTOR*FACTOR)
NADVAN = INT(DEVSRT/DELTTM + .5)
NADCHK = 3
NCKOUT = NADVAN/NADCHK + 1
NCKSV1 = NCKOUT
NADSAV = NADCHK

NPTS = NMHPTS
IF(NMHPTS.LT.30) NPTS=INT(1.5*FLOAT(NMHPTS))
IF(NMHPTS.LT.20) NPTS=2*NMHPTS
IF(NMHPTS.LT.14) NPTS=3*NMHPTS
IF(IDEVFL(3).EQ.1) NPTS=INT(1.4*FLOAT(NPTS))
NPTTMP = NPTS
MAXPTS = INT( 1.6*SQRT(ZMAX*ZMAX+ XMAX*XMAX)*FLOAT(NPTTMP) )
CXZL = (0.,1.)
CXZR = (0.,1.)
XZ(1) = (0.,0.)
XZ(NPTS) = CMPLX(XMAX,0.)

```

C NORMALIZE THE STARTING STRING ENDPOINT DIRECTIONS.

```

CXZL=CXZL/CMPLX(CABS(CXZL),0.)
CXZR=CXZR/CMPLX(CABS(CXZR),0.)
C SET MIN AND MAX STRING SEGMENT LENGTHS FOR X; SET MAX STRING
C LENGTH FOR Z.
CALL LINEAR
SDISTX = ABS( REAL(XZ(2)) - REAL(XZ(1)) )
SMINX = .70*SDISTX
SMAXX = 1.8*SDISTX
SMAZX = 1.8*DELTZ
IF(IDEVFL(3).NE.1) GO TO 1
SMAZX = 1.5*SDISTX
SMAZX = 1.5*DELTZ
1 TTOT=0.

C INITIALIZE THE PLTOUT ROUTINE.
NFLG = 0
NZFLG = 0
TTOTSV = 0.
2 NOUT = INT( (DEVEND-DEVSRT)/DEVINC ) + 1
IF(NOUT.GT.MXNDEV) CALL DEVMSG(5)
IF(NOUT.LT.1) CALL DEVMSG(6)
IF(NCKOUT.LT.1) CALL DEVMSG(10)
C TIME BETWEEN CHECKS = TCHK AND TIME BETWEEN ADVANCES = TADV
TCHK = DEVSRT/FLOAT(NCKOUT)
TADV = TCHK/FLOAT(NADCHK)
CALL DEVMSG(2)
CALL PLTOUT(26)
DO 5 IOUT=1,NOUT
IF( (NZFLG.EQ.1).OR.(IOUT.EQ.1) ) GO TO 30
NADVAN = INT(DEVINC/DELTTM + .5)
NCKOUT = NADVAN/NADCHK + 1
NCKSV2 = NCKOUT
TCHK = DEVINC/FLOAT(NCKOUT)
TADV = TCHK/FLOAT(NADCHK)
30 DO 3 ICKOUT=1,NCKOUT
CALL CYCLE
IF(IFLAG.EQ.1) CALL CHKR
IF(IFLAG.EQ.1) CALL BNDARY
C CALL DELOOP WHEN NPTS GETS LARGE AND AFTER EVERY OUTPUT.
C DELOOP(2) REMOVES ALL LOOPS, TAKES MORE TIME, AND IS RESERVED
C FOR THE OUTPUTS.
IF(NPTS.GE.MAXPTS) CALL DELOOP(1)
3 CONTINUE
IF((NZFLG.EQ.0).OR.(NFLG.EQ.1)) GO TO 4
NADCHK = 3
NCKOUT = 8
IF(NCKSV1.LE.16) NCKOUT = 4
TCHK = DEVINC/FLOAT(NCKOUT)
TADV = TCHK/FLOAT(NADCHK)
NFLG = 1
CALL DEVMSG(3)
MAXPTS = INT(1.8*FLOAT(NPTS))
4 CALL DELOOP(2)
IOUTTP = IOUT
IF(IDEVFL(1).EQ.1) CALL PRTPTS(IOUTTP)
IF(IDEVFL(2).EQ.1) CALL PLOTTP(IOUTTP)
CALL PLTOUT(IOUTTP)
5 CONTINUE
IF(IDEVFL(2).EQ.1) CALL DEVMSG(13)

WRITE(IPRINT,22222)
22222 FORMAT(/1H1)

RETURN
END

FUNCTION RATE(CZ)

COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DEVPAR/ E1,E2,E3
COMMON /DEVTIM/ MXNDEV,DEVSRT,DEVEND,DEVINC
COMMON /HORIMG/ DELTX,MNHPTS,NMHPTS,HORINT(50)
COMMON /MVSPOS/ RMXZ(52,52)
COMMON /SIMPAN/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMPLEX CZ,XZ,CXZL,CXZR

C FUNCTION RATE CALCULATES A RATE FOR THE X,Z POSITION GIVEN.
C NOTE (X,Z)=(0.,0.) IS (X=2.,Z=1.5)--CENTER OF UPPER BOUNDARY OF
C CELL (2,2).
C RATE LETS THE STRING DEVELOP OUTSIDE THE BOUNDARY AT A
C MUCH REDUCED RATE. IN ORDER TO KEEP THE STRING LENGTH DOWN.

```

C CHKR DELETES THE POINTS OUTSIDE OF (0.,XMAX).

```

      TRUEX = REAL(CZ)
      TRUEZ = AIMAG(CZ)
      X = 2.00001 + TRUEX/DELTX
      Z = 1.50001 + TRUEZ/DELTZ
      IX = INT(X)
      IZ = INT(Z)
      IF (IZ.LT.1) GO TO 40
      IF ((IX.GE.(NMHPTS+2)).OR.(IX.LE.0)) GO TO 40
      IF (TRUEZ.GT.ZMAX) GO TO 50
      FRACX = X - FLOAT(IX)
      FRACZ = Z - FLOAT(IZ)
      SFRACX = 1. - FRACX
      SFRACZ = 1. - FRACZ
      RM = RMXZ(IX,IZ)*SFRACX*SFRACZ + RMXZ(IX+1,IZ)*FRACX*SFRACZ
      * + RMXZ(IX,IZ+1)*SFRACX*FRACZ + RMXZ(IX+1,IZ+1)*FRACX*FRACZ
C ACCORDING TO THE HORNBERGER PAPER, THE R(M) CURVE FIT USED HERE IS
C NOT VALID FOR M<.4—LET THE RATE BE EQUAL TO THAT OF .4 FOR VALUES
C OF M<.4
      IF (RM.LT..4) GO TO 20
      RATE = EXP(-9.21034+E1+RM*(E2+E3*RM))
      RETURN

20 RATE = EXP(-9.21034+E1+.4*E2+.16*E3)
      RETURN
C NEXT STMT PREVENTS THE STR FROM DEVELOPING TOO FAR OUTSIDE THE BOUND.
40 RATE = .0001
      RETURN
C USE REFLECTIVE BOUNDARY CONDITIONS FOR 3 LAYERS BEYOND THE RESIST.
50 IF (TRUEZ.GT.(ZMAX+3.*DELTZ)) GO TO 40
      TRUEZ = ZMAX - (TRUEZ-ZMAX)
      GO TO 5
END

```

SUBROUTINE LINEAR

```

      COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
      COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
      COMPLEX XZ, CXZL, CXZR
      COMPLEX XZSTRT, XZSTEP
C SUBROUTINE LINEAR SETS UP THE FIRST STRING GIVEN THE NO. OF POINTS IN
C THE STRING AND THE LOCATION OF THE XZ ENDPOINTS. NOTE ALL X AND Z
C MANIPULATIONS ARE DONE SIMULTANEOUSLY USING COMPLEX ARITHMETIC.

      XZSTEP=(XZ(NPTS)-XZ(1))/CMLPX(FLOAT(NPTS-1),0.)
      XZSTRT=XZ(1)-XZSTEP
      NSTOP=NPTS-1
C CREATE THE MIDDLE POINT LOCATIONS XZ(N).
      DO 1 IPT=2,NSTOP
      XZ(IPT)=XZSTRT+CMLPX(FLOAT(IPT),0.)*XZSTEP
1 CONTINUE

      RETURN
      END

```

SUBROUTINE CYCLE

```

      COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
      COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
      COMMON /DVELP3/ NZFLG,TTOTSV
      COMPLEX XZ,CXZL,CXZR,DL,DR,DT
C CYCLE TAKES THE STRINGS THRU THE NO. OF ADVANCES IN BETWEEN CHECKS.

      IFLAG=0
      TMINX = 1.E38
      TMAXX = 0.
      TMAXZ = 0.
      ZPOSMX = 0.
      NSAVE = 0

      NSTOP=NPTS-1
      DO 1 N=1,NADCHK
C CALCULATE THE ADVANCE OF THE LEFT ENDPOINT.

```

```

DL=XZ(2)-XZ(1)
XZ(1) = XZ(1)+CXZL*CMLX(TADV*RATE(XZ(1)),0.)
C T IS THE LENGTH BETWEEN STRING POINTS--TMAX(X,Z) AND TMIN(X,Z) ARE
C THE MAX AND MIN LENGTH OF THE STRING SEGMENTS FOR X AND Z.
T = CABS(DL)
DL=DL/CMLX(T,0.)
C CALCULATE THE ADVANCE FOR THE MIDDLE STRING POINTS BASED ON THE
C DIRECTION OF THE SEGMENTS ON EITHER SIDE OF THE POINTS.
DO 2 M=2,NSTOP
DR=XZ(M+1)-XZ(M)
TMINX = AMIN1( TMINX,ABS(REAL(DR)) )
TMAXX = AMAX1( TMAXX,ABS(REAL(DR)) )
TMAXZ = AMAX1( TMAXZ,ABS(AIMAG(DR)) )
T=CABS(DR)
DR=DR/CMLX(T,0.)
DT=DL+DR
DT = DT/CMLX(CABS(DT),0.)
C DT*SQRT(-1) IS THE NORMALIZED DIRECTION FOR THE PRESENT POINT.
XZ(M) = XZ(M) + DT*CMLX(0.,TADV*RATE(XZ(M)))
ZPOSMX = AMAX1(ZPOSMX,AIMAG(XZ(M)))
DL=DR
2 CONTINUE
C CALCULATE THE ADVANCE POSITION FOR THE RIGHT ENDPOINT.
XZ(NPTS)=XZ(NPTS)+CXZR*CMLX(TADV*RATE(XZ(NPTS)),0.)
IF(ZPOSMX.GE.ZMAX) NSAVE=N
1 CONTINUE
C IF THE SEGMENT LENGTHS ARE TOO LONG/SHORT, DELETE/ADD POINTS(CHKR).
IF( (TMAXX.GT.SMAXX).OR.(TMINX.LT.SMINX).OR.(TMAXZ.GT.SMAXZ) )
* IFLAG=1
TTOT=TTOT+TCHK
C IF BREAKTHROUGH HAS OCCURED, SET FLAG.
IF( (NSAVE.EQ.0).OR.(NZFLG.EQ.1) ) RETURN
NZFLG = 1
TTOTSV = TTOT-TCHK+FLOAT(NSAVE)*TADV
RETURN

END

```

SUBROUTINE CHKR

```

COMMON /DEVFLG/ IDEVFL(S)
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
COMMON /IO1 / ITERM,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /SIMPAN/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMPLEX XZ,CXZL,CXZR,P0,P1,P2,P3,POSITN
DIMENSION INDEX(450)

C CHKR CHECKS FOR SEGMENTS THAT ARE TOO LONG OR TOO SHORT.

NPTOLD = NPTS
SMIDLE = (SMINX + SMAXX)/2.
NCHNGE = 0
C FIND HOW MANY SEGMENT LENGTHS ARE TOO LONG OR TOO SHORT.
DO 20 M=2,NPTS
SEGMTX = ABS( REAL(XZ(M)) - REAL(XZ(M-1)) )
SEGMTZ = ABS( AIMAG(XZ(M)) - AIMAG(XZ(M-1)) )
IF( (SEGMTX.LT.SMAXX).AND.(SEGMTZ.GT.SMINX).AND.
* (SEGMTZ.LT.SMAXZ) ) GO TO 20
NCHNGE = NCHNGE+1
INDEX(NCHNGE)=M
20 CONTINUE
IF(NCHNGE.EQ.0) GO TO 100

NADD=0
MLAST=1
C CORRECT THE SEGMENT LENGTHS.
DO 50 N=1,NCHNGE

C DETERMINE IF SEGMENT TO LEFT OF M IS TOO LONG OR SHORT.
M=INDEX(N)+NADD
SEGMTX = ABS( REAL(XZ(M)) - REAL(XZ(M-1)) )
SEGMTZ = ABS( AIMAG(XZ(M)) - AIMAG(XZ(M-1)) )
IF( (SEGMTX.LT.SMAXX).AND.(SEGMTZ.LT.SMAXZ) ) GO TO 30
C IF THE SEGMENT IS TOO LONG, ADD A POINT BETWEEN M-1 AND M;
C AND MAKE THE NEW POINT M. THE OLD-M MAKE NEW-(M+1), ETC.
IF(NPTS.GE.450) CALL DEVMSG(12)
NADD=NADD+1

```

```

C SET THE POSITION OF THE ADDED POINT XZ(M-NEW).
  IF (M.LT.3) P0=XZ(1)-CMLX(SMIDLE,0.)
  IF(M.GE.3) P0=XZ(M-2)
  P1 = XZ(M-1)
  P2 = XZ(M)
  IF(M.EQ.NPTS) P3=P2+CMLX(SMIDLE,0.)
  IF(M.LT.NPTS) P3= XZ(M+1)
  POSITN = (P1+P2)/CMLX(2.,0.) + (P1+P2-P0-P3)/CMLX(16.,0.)
C UPDATE THE POINTS FROM THE ADDED POINT ON.
  DO 25 I=M,NPTS
  J=NPTS+M-I
  XZ(J+1)=XZ(J)
  25 CONTINUE
  XZ(M) = POSITN
  NPTS=NPTS+1
  GO TO 50

  30 IF(SEGMTX.GT.SMINX) GO TO 50
C IF THE SEGMENT IS TOO SHORT, DELETE A POINT.
  IF(NPTS.LT.5) CALL DEVMSG(5)
C SINCE THE SEGMENT TO BE LENGTHENED
C IS TO THE LEFT OF THE CURRENT M, AND IF IT ALREADY HAS BEEN
C LENGTHENED BY THE FORMER SEGMENT CORRECTION (MFORMER=MCURRENT),
C DON'T LENGTHEN THE CURRENT SEGMENT.(SEE NEXT STATEMENT).
  IF(MLAST.EQ.M) GO TO 50
  NADD=NADD-1
  NPTS=NPTS-1
  MLAST=M
C REPLACE THE LEFT AND RIGHT POINTS OF THE SHORT SEGMENT WITH ONE POINT
C IN THE MIDDLE.
  XZ(M-1)=CMLX(.5,0.)*(XZ(M-1)+XZ(M))
  DO 40 K=M,NPTS
  XZ(K)=XZ(K+1)
  40 CONTINUE
C THE FORMER POINT M HAS BEEN DELETED AND THE NEW POINT M IS THE
C FORMER POINT M+1. THE NEW M-1 IS HALF WAY BETWEEN THE FORMER POINTS
C M,M-1.

  50 CONTINUE

100 IF(IDEVFL(1).EQ.1) WRITE(IPRINT,101) TTOT,NPTOLD,NPTS
101 FORMAT(6X,17H  CHR: TTOT = ,F5.1,4H SEC,10X,10HOLD-PTS = ,
* 13.3X,10HNEW-PTS = ,I3)
  RETURN
  END

SUBROUTINE BNDARY

COMMON /DEVFLG/ IDEVFL(5)
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /IO1 / ITERMI,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON /SIMPAN/ NPRLYR,NPRPTS,NENDIV,DELTM,DELTZ
COMPLEX XZ,CXZL,CXZR

C SUBROUTINE BNDARY DELETES POINTS IN THE BOUNDARY AREAS.

  NPTOLD = NPTS
  NBNDL = 0

C DELETE LEFT BUFFER STRING SECTIONS.
C NOTE--CHANGE ZM IN RATE IF YOU CHANGE ZM HERE.
  1 IFLG = 0
  ZM = ZMAX + 3.*DELTZ
  NPTDEL = 0
  DO 2 IPT=2,NPTS
  IF(REAL(XZ(IPT)).GE.0.) GO TO 3
  NPTDEL = NPTDEL + 1
  NBNDL = NBNDL+1
  NPTS = NPTS-1
  2 CONTINUE
  3 IF(NPTDEL.EQ.0) GO TO 5
  ZPOS = ( AIMAG(XZ(NPTDEL+1))+AIMAG(XZ(NPTDEL+2)) )/2.
  DO 4 N=2,NPTS
  XZ(N) = XZ(NPTDEL+N)
  4 CONTINUE
C RESET THE ENDPOINT POSITION AND LEAVE CXZL THE SAME AS BEFORE.
  XZ(1) = CMLX(0.,ZPOS)

```

```

COMMON /DVELP2/ TADV.TCHK.TTOT.IFLAG.SMAXX.SMINX.SMAXZ
COMMON /DVELP4/ BREAK.MAXPTS.NADSAV.NCKSV1.NCKSV2.NOUT
COMMON /IO1 / ITERML.IBULK.IPROUT.IRESV1.IIN.IPRINT.IPUNCH
COMMON IPLT(123,99).X(450).Z(450)
DIMENSION OUTTIM(20),XCROSS(20),ANGEST(20)
DIMENSION IALPHA(21)
COMPLEX XZ,CXZL,CXZR,ANGPTS(5)
DATA IALPHA /1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM
* ,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU/
DATA KSTAR,KMINUS,KPOINT,KBLANK,K0,K1,K3
* /1H*,1H-,1H..1H ,1H0,1H1,1H3/

C THE GRAPH IS 121 POINTS IN X AND 97 POINTS IN Z. * DENOTES THE
C BOUNDARY. XZ(1) IS THE UPPER LEFT CORNER OF THE GRAPH-NOT INCLUDING
C THE BOUNDARY. XMAX AND ZMAX DETERMINE THE DIMENSIONS OF THE GRAPH.
C THE CENTER OF IPLT(2,2) REPRESENTS THE (0.,0.) IN THE RESIST.
C THE CENTER OF IPLT(122,98) REPRESENTS THE (XMAX,ZMAX) RESIST POINT.

C IF THIS IS THE FIRST TIME THRU. SET UP THE BOUNDARY FOR IPLT AND
C INITIALIZE VARIOUS PARAMETERS.
IF(IOUTPT.EQ.25)GO TO 20
IF(IOUTPT.NE.26)GO TO 5
NZFLAG = 0
NADFLG = 0
ZT = 0.
ZB = ZMAX
XR = XMAX-CPEDGE
XL = -CPEDGE
XAXIS = (ZMAX/XMAX)*96.
IF(XAXIS.GT.96.)XAXIS = 96.
NXBND = INT(XAXIS+3.5)
DO 2 L=1,99
DO 2 K=1,123
IPLT(K,L) = KBLANK
2 CONTINUE
IO = INT( (CPEDGE/CPWIND)*120. + 2.00001 )
DO 3 K=1,123
IPLT(K,1) = KSTAR
IPLT(K,NXBND) = KSTAR
3 CONTINUE
NXBNDM = NXBND-1
DO 4 L=2,NXBNDM
IPLT(1,L) = KSTAR
IPLT(123,L) = KSTAR
IPLT(10,L) = KPOINT
4 CONTINUE
IPLT(10,2) = K0
RETURN

5 OUTTIM(1) = DEVSRT
IF(IOUTPT.NE.1) OUTTIM(IOUTPT) = DEVSRT+DEVINC*FLOAT(IOUTPT-1)
DO 8 K=1,NPTS
TEMPX = REAL(XZ(K))
NX = INT( (TEMPX/(XR-XL))*120. + 2.00001 )
TEMPZ = AIMAG(XZ(K)) - ZT
IF(TEMPZ.GE.ZMAX)NZFLAG=1
NZ = INT( (TEMPZ/(ZB-ZT))*XAXIS + 2.00001 )
IF((NX.GT.1).AND.(TEMPX.LE.XMAX).AND.(NZ.GT.1).AND.(TEMPZ.LE.ZMAX)
* )IPLT(NX,NZ) = IALPHA(IOUTPT)
8 CONTINUE
IF(NZFLAG.EQ.0) GO TO 19
IF(NADFLG.NE.0) GO TO 9
NADFLG = IOUTPT
ADVPR = FLOAT((NCKSV1+(NADFLG-1)*NCKSV2)*NADSAV)/ZMAX

C FOLLOWING SECTION DETERMINES THE X VALUE OF THE RESIST-SUBSTRATE
C INTERSECTION FOR THE CONTOURS.
N = NPTS -1
9 DO 10 I=1,N
DZI = ZMAX - AIMAG(XZ(I))
DZIPI = ZMAX - AIMAG(XZ(I+1))
IF((DZI*DZIPI).GT.0.) GO TO 10
ISV = I
GO TO 14
10 CONTINUE
XCROSS(IOUTPT) = XMAX - CPEDGE
GO TO 16
14 TEMP = REAL(XZ(ISV)) - REAL(XZ(ISV+1))
IF(ABS(TEMP).LT.1.E-30) GO TO 15
SLOPE = (AIMAG(XZ(ISV))-AIMAG(XZ(ISV+1)))/TEMP
B = AIMAG(XZ(ISV))-SLOPE*REAL(XZ(ISV))

```

```

XCROSS(IOUTPT) = (ZMAX-B)/SLOPE - CPEDGE
GO TO 16
15 XCROSS(IOUTPT) = REAL(XZ(ISV)) - CPEDGE

C SIDEWALL ANGLE ESTIMATION SECTION
16 INDEX = 0
DO 17 I=1,NPTS
  IF(INDEX.GE.5) GO TO 17
  IF(AIMAG(XZ(I+1)).GT.(1.01*ZMAX)) GO TO 17
  IF(REAL(XZ(I)).LE.REAL(XZ(I+1))) GO TO 17
  IF(REAL(XZ(I+1)).GE.REAL(XZ(I+2))) GO TO 17
  INDEX = INDEX + 1
  ANGPTS(INDEX) = XZ(I+1)
17 CONTINUE
C DO A LINEAR FIT TO THE POINTS IN ANGPTS
  IF(INDEX.LE.1)GO TO 19
  SUMX = 0.
  SUMXSQ = 0.
  SUMY = 0.
  SUMXY = 0.
  DO 18 I=1,INDEX
    SUMXY = SUMXY + AIMAG(ANGPTS(I))*REAL(ANGPTS(I))
    SUMX = SUMX + REAL(ANGPTS(I))
    SUMY = SUMY + AIMAG(ANGPTS(I))
    SUMXSQ = SUMXSQ + REAL(ANGPTS(I))*REAL(ANGPTS(I))
18 CONTINUE
  TEMP = (SUMXSQ-SUMX*SUMX)/FLOAT(INDEX)
  IF(TEMP.EQ.0.) TEMP = .00001
  TMP = SUMXY - SUMX*SUMY/FLOAT(INDEX)
  ANGEST(IOUTPT) = (360.*ATAN2(TMP,TEMP))/6.283

19 IF(IOUTPT.NE.NOUT) RETURN
20 WRITE(IPRINT,21)
21 FORMAT(//////,45X,27H--- DEVELOPED PATTERN ---,/)
  WRITE(IPRINT,22) XL,XR,ZT,ZB
22 FORMAT(1HX,9HX LEFT = , F8.4,8H MICRONS./
  * 1X,9HX RIGHT = , F8.4,8H MICRONS./
  * 1X, 9HZ TOP = , F8.4, 8H MICRONS./
  * 1X, 11HZ BOTTOM = , F6.4,8H MICRONS.
  *//,1X,17HSYMBOL: TIME: ,4X,30HRESIST-SUBSTRATE INTERSECTION:
  *8X,53HSIDEWALL ANGLE ESTIMATE(POSSIBLY INACCURATE-CHECK FOR.
  * 15H PLAUSIBILITY):)
  NF = NADFLG-1
  IF(NADFLG.EQ.0) NF=NOUT
  IF(NADFLG.EQ.1) GO TO 26
  WRITE(IPRINT,25) (ALPHA(IOUT),OUTTIM(IOUT),
  * IOUT=1,NF)
25 FORMAT(4X,A1,2X,F7.1,4H SEC)
  IF(NADFLG.EQ.0) GO TO 30
26 WRITE(IPRINT,28) (ALPHA(IOUT),OUTTIM(IOUT),XCROSS(IOUT),
  * ANGEST(IOUT), IOUT=NADFLG,NOUT)
28 FORMAT(4X,A1,2X,F7.1,4H SEC,9X,4HX = .F8.4,8H MICRONS,
  * 15X,F6.1,8H DEGREES)
30 WRITE(IPRINT,31) CPWIND,CPEDGE,( (IPLT(I,J),I=1,123) J=1,NXBND)
31 FORMAT(1HX,/14X,13HTHE WINDOW IS,F7.4,19H MICRONS WIDE IN X..5X.
  *11HTHE EDGE IS,F7.4,42H MICRONS FROM THE LEFT SIDE OF THE WINDOW.
  *./,(8X, 123A1))
  IF(NZFLAG.EQ.1) WRITE(IPRINT,34) OUTTIM(NADFLG)
34 FORMAT(8X,79HTHE RESIST HAS DEVELOPED THROUGH TO THE SUBSTRATE AT
  *ONE OR MORE POINTS IN THE ,F5.1,12H SEC OUTPUT.)
  IF(IDEVFL(5).EQ.1) RETURN
  NADV = NCKSV1+NCKSV2*(NADFLG-1)*NADSAV
  IF(NADFLG.NE.0) WRITE(IPRINT,36) ADVPR,IALPHA(NADFLG),NADV
36 FORMAT(8X,37HTHE APPROXIMATE NUMBER OF ADV/MIC IS ,F7.2./,
  *8X, 7HOUTPUT ,A1, 5H TOOK,14,17H STRING ADVANCES.)

RETURN
END

```

SUBROUTINE PLOTHP(IOUTPT)

```

COMMON /DEVTIM/ MXNDEV,DEVSRT,DEVEND,DEVINC
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAZX
COMMON /IO1 / ITER,MI,BULK,IROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON IPLT(123,99),X(450),Z(450)
COMPLEX XZ, CXZL,CXZR

```

C DELETE LOWER BUFFER LEFT STRING SECTIONS-LEAVE 4 PTS FOR BOUND COND.

```

5 NPTDEL = 0
DO 6 IPT=1,NPTS
  IF(AIMAG(XZ(IPT)).LE.ZM) GO TO 7
  NPTDEL = NPTDEL + 1
6 CONTINUE
7 IF(NPTDEL.LE.4) GO TO 9
  NPTDEL = NPTDEL - 4
  NPTS = NPTS - NPTDEL
DO 8 N=1,NPTS
  XZ(N) = XZ(N+NPTDEL)
8 CONTINUE

```

C NOTE THERE IS SOME ERROR FOR STANDING WAVES DEVELOPING OUTSIDE OF
C THE BOUNDARY.
C NOTE ALSO THAT THE 0. BOUNDARY MUST BE CHANGED IF WE DEVELOP SELECTED
C PORTIONS OF THE STRING.

C DELETE THE RIGHT BUFFER STRING SECTIONS.

```

9 NPTDEL = 0
  NTEMP = NPTS
  NN = NPTS-1
DO 10 I=1,NN
  IPT = NTEMP - I
  IF(REAL(XZ(IPT)).LE.XMAX) GO TO 11
  NPTDEL = NPTDEL + 1
  NBNDEL = NBNDEL+1
  NPTS = NPTS - 1
10 CONTINUE
11 IF(NPTDEL.EQ.0) GO TO 12
  ZPOS = ( AIMAG(XZ(NPTS))+AIMAG(XZ(NPTS-1)) )/2.
  XZ(NPTS) = CMPLX(XMAX,ZPOS)

```

C DELETE LOWER BUFFER RIGHT STRING SECTIONS.

```

12 NPTDEL = 0
  NTEMP = NPTS + 1
DO 13 I=1,NPTS
  IPT = NTEMP - I
  IF(AIMAG(XZ(IPT)).LE.ZM) GO TO 14
  NPTDEL = NPTDEL + 1
13 CONTINUE
14 IF(NPTDEL.LE.4) GO TO 19
  NPTS = NPTS - (NPTDEL-4)

```

```

19 IF(IDEVFL(1).EQ.1) WRITE(IPRINT,20) NBNDEL
20 FORMAT(20X,9HBOUNDARY: ,I3,24H BOUNDARY POINTS DELETED)

```

```

RETURN
END

```

SUBROUTINE DELOOP(ITYPE)

```

COMMON /DEVFLG/ IDEVFL(5)
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
COMMON /IO1 / ITERM,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMPLEX XZ, CXZL, CXZR

```

C DELOOP REMOVES ANY LOOP THAT DEVELOPS IN THE STRING.
C ITYPE=2 REMOVES ALL LOOPS FOR FINAL OUTPUT.
C ITYPE=1 IS FOR INTERMEDIATE CALLS AND IS LESS COMPREHENSIVE.

```

  IF(IDEVFL(1).EQ.1) WRITE(IPRINT,10) NPTS
  10 FORMAT(2X,19HDELOOP CALLED WITH ,I3,9H POINTS :)
C CALL CHKR IN ORDER TO HAVE STRING LENGTHS APPROXIMATELY CORRECT.
  CALL CHKR
  NJUMP = 2
  IF(IDEVFL(3).EQ.1) NJUMP=4
  NATEMP = 0
  N = 0
  L1 = 7
  L2 = 4
  SZ = 1.5*SMAXZ
  SX = 1.5*SMAXX
  IF(ITYPE.EQ.1) GO TO 1
  SZ = 1.75*SMAXZ
  SX = 1.75*SMAXX
  L1 = 4
  L2 = 1

```

C 1.75 IS ARBITRARY--DELETING ALL LOOPS HAS THE DISADVANTAGE OF CHECKING

```

C ALL ADJACENT SEGMENTS FOR INTERSECTION.
  1 N=N+1
C IF M IS AT THE ENDPOINT, RETURN.
  IF(N.GT.(NPTS-L1)) GO TO 100
  M = N+L2
C M IS 5(OR 2) POINTS BEYOND N TO START WITH AND IS INCREASED UNTIL
C N IS INCREMENTED. A LOOP CONSISTS OF THE POINTS M AND N (N IS
C TO THE LEFT OF M).
  2 M=M+1
C IF THE X(OR Z) DISTANCES BETWEEN THE POINTS M AND N IS GREATER THAN
C 1.5(1.75) TIMES THE ORIGINAL X(Z) STRING SEGMENT LENGTHS, INCREMENT M
  IF(M.GE.NPTS) GO TO 1
  IF(ABS(REAL(XZ(M))-REAL(XZ(N))).GT.SX) GO TO 2
  IF(ABS(AIMAG(XZ(M))-AIMAG(XZ(N))).GT.SZ) GO TO 2
  NATEMP = NATEMP + 1
C DO THE SEGMENTS N,N+1 AND M,M+1 INTERSECT--IF NOT, INCREMENT M.
  XM = REAL(XZ(M))
  ZM = AIMAG(XZ(M))
  XMP1 = REAL(XZ(M+1))
  ZMP1 = AIMAG(XZ(M+1))
  XN = REAL(XZ(N))
  ZN = AIMAG(XZ(N))
  XNP1 = REAL(XZ(N+1))
  ZNP1 = AIMAG(XZ(N+1))
C ATTEMPT TO GUARD AGAINST OVERFLOW.
  RUN = (XMP1-XM)
  IF(ABS(RUN).LT.1.E-38)GO TO 2
  SLOPEM = (ZMP1-ZM)/RUN
  RINCPM = (XMP1*ZM-XM*ZMP1)/RUN
  RUN = (XNP1-XN)
  IF(ABS(RUN).LT.1.E-38)GO TO 2
  SLOPEN = (ZNP1-ZN)/RUN
  RINCPN = (XNP1*ZN-XN*ZNP1)/RUN
  SLOPE = SLOPEN-SLOPEM
  IF(ABS(SLOPE).LT.1.E-38)GO TO 2
  XINTER = (RINCPM-RINCPN)/SLOPE
  ZINTER = SLOPEN*XINTER + RINCPN
C (XINTER,ZINTER) IS THE POINT WHERE THE TWO LINES INTERSECT.
CHECK TO SEE THAT (XINTER,ZINTER) IS INCLUDED IN THE LINE SEGMENT N,N+1
  IF( (XINTER.LT.AMAX1(AMIN1(XN,XNP1),AMIN1(XM,XMP1))) .OR.
    * (XINTER.GT.AMIN1(AMAX1(XN,XNP1),AMAX1(XM,XMP1))) .OR.
    * (ZINTER.LT.AMAX1(AMIN1(ZN,ZNP1),AMIN1(ZM,ZMP1))) .OR.
    * (ZINTER.GT.AMIN1(AMAX1(ZN,ZNP1),AMAX1(ZM,ZMP1))) ) GO TO 2
  XZ(N+1) = CMPLX(XINTER,ZINTER)
C DELETE THE LOOP.
  JSTART = N+2
  JSTOP = NPTS - (M-(N+1))
  DO 3 J=JSTART, JSTOP
  JJ=J+M-N-1
  XZ(J) = XZ(JJ)
  3 CONTINUE
  NOLD=NPTS
  NPTS=JSTOP
  IF(IDEVFL(1).EQ.1) WRITE(IPRINT,4) NOLD,NPTS,TTOT,NATEMP
  4 FORMAT(7X,20HDELOOP: OLD-NPTS = ,I3,8X,11HNEW-NPTS = ,I3,8X,
  C 7HTIME = ,F5.1,4H SEC,8X,I3,9H ATTEMPTS)
  N = N+1
C JUMP AHEAD FOR INTERMEDIATE DELOOPS.
  IF(ITYPE.EQ.1) N=N+NJUMP
  NATEMP = 0
  GO TO 1

100 IF(IDEVFL(1).EQ.1) WRITE(IPRINT,101) NPTS
101 FORMAT(2X,21HLEAVE DELOOP--NPTS = ,I3)
  RETURN

```

END

SUBROUTINE PLTOUT(IOUTPT)

```

COMMON /COPYWD/ CPWIND.CPEDGE.CPWORG
COMMON /DEVFLG/ IDEVFL(5)
COMMON /DEVPAR/ E1.E2.E3
COMMON /DEVTIM/ MXNDEV.DEVSRT.DEVEND.DEVINC
COMMON /DVLEP1/ XZ(450),XMAX.ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT

```

```

XL = .0
ZB = -ZMAX
ZT = .0
IF(IOUTPT.NE.1)GO TO 8
NOUT = INT((DEVEND-DEVSRT)/DEVINC) + 1
RNOUT = FLOAT(NOUT)
WRITE(IPUNCH,1) XL,XMAX,ZB,ZT,RNOUT
1 FORMAT(/,4(1X,F8.5),/,1X,F8.5)
8 DO 10 I=1,NPTS
X(I) = REAL(XZ(I))
Z(I) = -AIMAG(XZ(I))
10 CONTINUE
RNPTS = FLOAT(NPTS)
WRITE(IPUNCH,2) RNPTS
2 FORMAT(1X,F9.5)
WRITE(IPUNCH,3) ( X(I),Z(I) ,I=1,NPTS )
3 FORMAT(11(1X,F6.3))

RETURN
END

```

SUBROUTINE PRTPTS(IOUTPT)

```

COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
COMMON /IO1 / ITERM,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMMON IPLT(123,99),X(450),Z(450)
COMPLEX XZ, CXZL,CXZR

DO 10 I=1,NPTS
X(I) = REAL(XZ(I))
Z(I) = -AIMAG(XZ(I))
10 CONTINUE
WRITE(IPRINT,2) NPTS
2 FORMAT(1X,I3,15H STRING POINTS:)
WRITE(IPRINT,3) ( X(I),Z(I) ,I=1,NPTS )
3 FORMAT(11(1X,F6.3))

RETURN
END

```

SUBROUTINE DEVMSG(NUMB)

```

COMMON /DEVFLG/ IDEVFL(5)
COMMON /DEVPAR/ E1,E2,E3
COMMON /DEVTIM/ MXNDEV,DEVSRT,DEVEND,DEVINC
COMMON /DVELP1/ XZ(450),XMAX,ZMAX,NPTS,CXZL,CXZR,NADCHK,NCKOUT
COMMON /DVELP2/ TADV,TCHK,TTOT,IFLAG,SMAXX,SMINX,SMAXZ
COMMON /DVELP3/ NZFLG,TTOTSV
COMMON /DVELP4/ BREAK,MAXPTS,NADSAV,NCKSV1,NCKSV2,NOUT
COMMON /IO1 / ITERM,IBULK,IPROUT,IRESV1,IIN,IPRINT,IPUNCH
COMPLEX XZ,CXZL,CXZR

```

C DEVMSG IS THE MESSAGE SUBROUTINE FOR THE DEVELOP ROUTINES.

```

GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15),NUMB

1 WRITE(IPRINT,101) E1,E2,E3,DEVSRT,DEVINC,DEVEND
IF(IDEVFL(3).EQ.1) WRITE(IPRINT,102)
IF(IDEVFL(5).EQ.0) WRITE(IPRINT,103)
IF(IDEVFL(5).EQ.1) WRITE(IPRINT,104)
RETURN
2 IF(IDEVFL(1).EQ.0) RETURN
WRITE(IPRINT,1000)
WRITE(IPRINT,20) XMAX,ZMAX,NPTS,MAXPTS,CXZL,CXZR,XZ(1),XZ(NPTS)
*,NADSAV,NADCHK,NCKSV1,NCKSV2,NCKOUT,BREAK,TADV,SMAXX,SMINX,SMAXZ
RETURN
3 WRITE(IPRINT,1000)
WRITE(IPRINT,30) TTOTSV
IF(IDEVFL(1).EQ.1) WRITE(IPRINT,301) NADCHK,NCKOUT
RETURN
4 WRITE(IPRINT,2000)
WRITE(IPRINT,40)
RETURN
5 WRITE(IPRINT,2000)

```

```

WRITE(IPRINT,50)
STOP
6 WRITE(IPRINT,2000)
WRITE(IPRINT,60)
STOP
7 RETURN
8 RETURN
9 RETURN
10 WRITE(IPRINT,3000)
WRITE(IPRINT,100)
RETURN
11 WRITE(IPRINT,1000)
RM1 = .0001*EXP(E1+E2+E3)
RM75 = .0001*EXP(E1+.75*E2+.5625*E3)
RM5 = .0001*EXP(E1+.5*E2+.25*E3)
WRITE(IPRINT,110) RM1,RM75,RM5
RETURN
12 WRITE(IPRINT,2000)
WRITE(IPRINT,120)
CALL PLTOUT(25)
STOP
13 WRITE(IPRINT,1000)
WRITE(IPRINT,130)
RETURN
14 RETURN
15 RETURN

101 FORMAT(///,5X,17(1H*),/5X,17H* RUN DEVELOP *,/5X,17(1H*),
* ///,32X,18HPARAMETER VALUES ://,10X,5HE1 = ,F6.2,8X,
* 5HE2 = ,F6.2,8X,5HE3 = ,F6.2,
* /10X,26HFIRST DEVELOPMENT OUTPUT = ,F6.1,4H SEC.
* /10X,45HTIME INCREMENT BETWEEN DEVELOPMENT OUTPUTS = ,F5.1,
* 4H SEC./10X,26HFINAL DEVELOPMENT OUTPUT = ,F6.1,4H SEC)
102 FORMAT(10X,29HEXTRA STRING POINTS REQUESTED)
103 FORMAT(10X,23HINITIAL DEVELOPMENT RUN)
104 FORMAT(10X,46HINTERMEDIATE DEVELOPMENT RUN(WARNING: NUMBER .
*54HOF ADVANCES PER OUTPUT SET BY INITIAL DEVELOPMENT RUN))
20 FORMAT(1X, 20HINTERNAL PARAMETERS://,
* 1X, 6HX MAX , F10.4,/,
* 1X, 6HZ MAX , F10.4, /,
* 1X, 36HNUMBER OF POINTS IN STARTING STRING , I4, /,
* 1X,37HSTRING POINT LIMIT FOR DELOOP CALL ,I3,/,
* 1X, 46HDIRECTION OF LEFT ENDPOINT, X AND Z COMPONENTS, 2F7.2,/,
* 1X, 47HDIRECTION OF RIGHT ENDPOINT, X AND Z COMPONENTS,2F7.2,/,
* 1X, 25HX AND Z OF LEFT ENDPOINT , 2F7.4, /,
* 1X, 25HX AND Z OF RIGHT ENDPOINT, 2F7.2, /,
* 1X, 51HNUMBER OF ADVANCES BETWEEN CHECKS BEFORE BREAKTHRU ,I3,/,
* 1X, 42HPRESENT NUMBER OF ADVANCES BETWEEN CHECKS , I3, /,
* 1X, 34HNUMBER OF CHECKS FOR FIRST OUTPUT ,I3,/,
* 1X, 50HNUMBER OF CHECKS BETWEEN OUTPUTS BEFORE BREAKTHRU ,I3,/,
* 1X, 33HNUMBER OF CHECKS BETWEEN OUTPUTS , I3, /,
* 1X,42HESTIMATED TIME UNTIL RESIST BREAKTHROUGH = ,F6.1,4H SEC. /,
* 1X,24HTIME BETWEEN ADVANCES = ,F4.2,4H SEC./,
* 1X,39HMAXIMUM STRING-POINT SEPARATION IN X = ,F6.4,8H MICRONS./,
* 1X,39HMINIMUM STRING-POINT SEPARATION IN X = ,F6.4,8H MICRONS./,
* 1X,39HMAXIMUM STRING-POINT SEPARATION IN Z = ,F6.4,8H MICRONS./)
30 FORMAT(5X,46HTHE DEVELOPER HAS BROKEN THROUGH THE RESIST IN,
*F5.1,9H SECONDS.)
301 FORMAT(///,10X,45HTHE NUMBER OF ADVANCES BETWEEN CHECKS IS NOW .
*12,/,10X,45HTHE NUMBER OF CHECKS BETWEEN OUTPUTS IS NOW ,I2)
40 FORMAT(10X,50HBREAKTHROUGH-ESTIMATION SECTION UNABLE TO CONVERGE)
50 FORMAT(10X,50HTHE NUMBER OF REQUESTED OUTPUTS PER RUN OF DEVELOP,
*22H MUST BE LESS THAN 21.)
60 FORMAT(20X, 45HNO OUTPUTS WERE REQUESTED FROM RUN OF DVELOP.)
100 FORMAT(40X,10HNCKOUT < 1)
110 FORMAT(40X,26HBACKGROUND DEVELOP RATE = ,F7.5,12H MICRONS/SEC./,
* 40X,26H M=.75 DEVELOP RATE = ,F6.4,12H MICRONS/SEC./,40X,
* 26H M=.5 DEVELOP RATE = ,F6.4,12H MICRONS/SEC)
120 FORMAT(20X, 56HIN CHKR: THE NUMBER OF DEVELOPMENT STRING POINTS
*> 450./,20X,46HTHE PLOTS GENERATED BEFORE THE FATAL ERROR ARE,
*17H OUTPUTTED BELOW.)
130 FORMAT(25X,38HCARDS WERE PUNCHED FOR THE HP PLOTTER.)

1000 FORMAT(///,20X,44H----- SYSTEM MESSAGE(DVELOP) -----./)
2000 FORMAT(///,20X,41H----- FATAL ERROR(DVELOP) -----./)
3000 FORMAT(///,20X,37H----- WARNING(DVELOP) -----./)

```

END

Appendix E

The String Point Insertion Algorithm*

SAMPLE uses an algebraically derived formula for the position of the interpolated string-point, given by

$$P_{2.5} = \frac{P_2 + P_3}{2} + \frac{P_2 - P_1 + P_4 - P_2}{16} \quad (\text{E-1})$$

where P_j represents the complex position in (x, z) of the j th string point. The algebraic approach uses the Lagrangian interpolation polynomial $Q(j)$ to approximate $P(j)$, where

$$Q(j) = \sum_{i=1}^4 \delta_i(j) P(i) \quad (\text{E-2a})$$

and where

$$\delta_i(j) = \prod_{k=1, k \neq i}^4 \frac{j-k}{i-k} \quad (\text{E-2b})$$

so that

$$Q(i) = P(i) \quad \text{for } i=1, 2, 3, 4. \quad (\text{E-2c})$$

Carrying out equations (E-2) for the insertion of a point between P_2 and P_3 gives equation (E-1), where $Q(2.5) = P(2.5)$.

*Appendix E has been provided by Sharad N. Nandgaonkar and is included here for documentation completeness.

Appendix F

SAMPLE Control Cards

Table I is an alphabetical listing of the control cards that work with the version of SAMPLE documented in this thesis. The parameter choice is listed between / /, and the units are indicated between (). Where several parameters are indicated, only the parameters applicable to the request need be specified. For example, the card

ILLUMINATION PARTIALLY COHERENT .7

specifies partially coherent mask illumination with a σ of .7 . If incoherent illumination is desired, the card should read

ILLUMINATION INCOHERENT

and σ is not specified. Specification of σ in the card changes the default value of σ for the current batch of runs.

Table I

control card	parameters
ACCURACY	/DEVELOP/ /ON,OFF/
CONTACT	/SEP(μm)/ /C1($\frac{\text{mJ}}{\text{cm}^2}$)/ /C2(μm)/
CONTINUE	/DEVELOP/
DEFOCUS	/D(μm)/
DEVTIME	/ t_1 (sec)/ / t_2 (sec)/ / t_3 (sec)/
DIFFUSION	/ σ_D (μm)/
DOSE	/E($\frac{\text{mJ}}{\text{cm}^2}$)/
END	
ILLUMINATION	/COHERENT,INCOHERENT,PARTIALLY COHERENT/ / σ /
LABEL	/74 character message on graphs/
LAMBDA	/ λ (μm)/ /relative intensity/
LAYER	/ n_i / / k_i / / t_i (μm)/
MASK	/LINE,SPACE,LINESPACE/ /L(μm)/ /S(μm)/
MODEL	/SUBSTRATE,RESIST,OXIDE,NITRIDE,SPECIAL #/ /OPTICAL,EXPOSE,DEVELOP/ /values/
PLOT	/IMAGE,MTF,CONTOURS,DEBUG/
PLOTOFF	/IMAGE,MTF,CONTOURS,DEBUG/
PRINT	/IMAGE,MTF,CONTOURS,DEBUG/
PRINTOFF	/IMAGE,MTF,CONTOURS,DEBUG/
PROJ	/numerical aperture/
RUN	/IMAGE,EXPOSE,DEVELOP,ALL/
TITLE	/74 character title/
TRIAL	/trial parameters/
WINDOW	/W(μm)/ /E(μm)/