Model calibration for optical lithography via semidefinite programming

Mehrdad Nouralishahi* Clive Wu† Lieven Vandenberghe‡

Abstract

We apply semidefinite programming to an estimation problem in optical lithography. In this problem, an approximate model of the imaging system is modified so that it satisfies a set of calibration measurements, with the purpose of incorporating into the model distortion effects due to diffusion and etching. The estimation problem is formulated as a semidefinite program, and several techniques are presented for exploiting problem structure in an interior-point method for solving it. These include efficient methods for handling upper bounds on the matrix variables, symmetry constraints on the variables, and low-rank structure in the coefficient matrices.

1 Introduction

We present a method for estimating a nonlinear model of the form

$$I(x) = \sum_{k=1}^{r} |(\phi_k * u)(x)|^2 , \tag{1}$$

based on measurements of the input u and the output I, and on prior knowledge of the underlying system. The independent variable x is typically one- or two-dimensional, the functions ϕ_k and u can be real- or complex-valued, and $\phi_k * u$ denotes the convolution of ϕ_k and u. The functions ϕ_k are the model parameters to be estimated.

The paper is motivated by an application in optical lithography for semiconductor manufacturing. In this context u is the input image, to be transferred from an integrated circuit mask to a silicon wafer by the lithographic process, I is the intensity of the output image on the wafer, and x is a two-dimensional vector of coordinates. Models of the form (1) were introduced by Hopkins to describe partially coherent optical systems [Hop51]. In the problem of interest here, we assume that an accurate Hopkins model of the optical system is available, and our goal is to adjust the model parameters to make it consistent with a set of

*Mentor Graphics, San Jose, California (Mehrdad_Nouralishahi@mentor.com).
†Blaze-DFM, Inc., Sunnyvale, California (cwu@blaze-dfm.com).
‡Electrical Engineering Department, University of California, Los Angeles (vandenbe@ee.ucla.edu).

measured data. This adjusted or calibrated model is more accurate because it incorporates additional distortions due to etching and diffusion that are difficult to model analytically. More details and background about the application are presented in section 2.

The method we present is based on a formulation of the problem as a *semidefinite program* (SDP), *i.e.*, a convex optimization problem with a linear cost function and linear matrix inequality constraints. Although several high-quality packages for solving SDPs are available, the SDPs that arise in the lithography application are typically very large, and non-standard techniques are necessary to solve them. We present a customized primal-dual interior-point method that exploits three types of problem structure: symmetries in the matrix variable, upper bounds on the matrix variable, and rank-one structure in the coefficient matrices. The symmetry constraints result from the invariance of the output intensity at a point with respect to reflections of the input image about the two coordinate axes through the point. Using elementary arguments from linear algebra, we exploit this invariance property to reduce the dimension of the optimization variable. The same result can be obtained by more general algebraic techniques due to Gatermann and Parrilo [GP04].

Second, we describe an efficient method for handling upper bounds on the variables in a standard form SDP. For dense, unstructured SDPs this results in a method with roughly the same computational cost as for the corresponding problem without upper bounds. This technique was also obtained independently by Toh *et al.* [TTT07].

A final reduction in complexity is achieved by exploiting rank-one structure in the SDP coefficient matrices [BYZ00, RV06].

The paper is organized as follows. Section 2 describes the lithography application in more detail. In section 3 we formulate the model calibration problem as an SDP. In order to simplify the notation, we first consider one-dimensional systems, and then extend the formulation to two dimensions. In section 4 we discuss the implications of symmetry constraints of the form $X = PXP$ where P is a symmetric permutation matrix, and show that they allow us to reduce the size of the problem by a change of variables. In section 5 we present a fast solution method for standard form SDPs with upper bounds on the variables. An example is given in section 6.

Notation $\mathbf{S}^n$ denotes the set of symmetric real matrices of order n. I_n is the identity matrix of order n, and J_n is the matrix I_n with the order of its columns reversed. The subscripts in I_n and J_n are omitted if the dimension is clear from the context. For a symmetric matrix X, $X \succeq 0$ means X is positive semidefinite. For a vector x, $x \succeq 0$ means the components of x are nonnegative. For a vector $x \in \mathbf{R}^n$, $\mathbf{diag}(x)$ is the diagonal matrix with x on its diagonal. If $X \in \mathbf{S}^n$, $\mathbf{diag}(X)$ is the vector of the diagonal elements of X. $\mathbf{Z}$ is the set of integers, $\mathbf{R}$ is the set of real numbers, and $\mathbf{C}$ is the set of complex numbers.

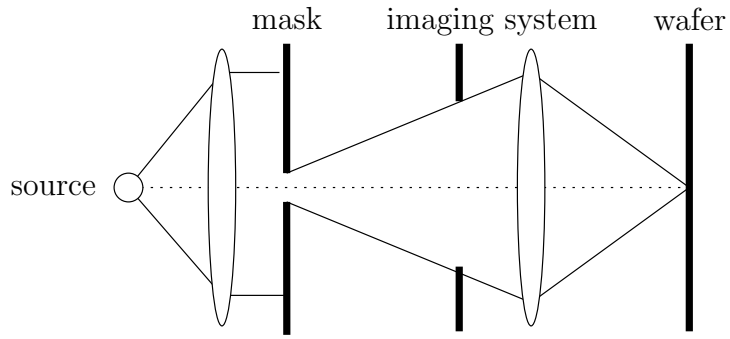

Figure 1: Optical lithography system.

2 Model calibration in optical lithography

2.1 The Hopkins model

In integrated circuit manufacturing, optical lithography is used to transfer an input pattern from the mask to the photoresistive material that covers the silicon wafer (see figure 1). A common model for the optical system is the *Hopkins model* [Hop51, Won01]

$$I(x) = \iint u(x')u^*(x'')\mathcal{K}(x-x')\mathcal{K}^*(x-x'')\mathcal{J}(x'-x'')dx'dx''. \tag{2}$$

Here $I(x)$ is the output image intensity at position x in the image plane, $u(x')$ is the value of the input image at point x', $\mathcal{K}$ is the impulse response or *point spread function* (PSF) of the system, and $\mathcal{J}$ is the *mutual intensity function* of the illumination. The superscript in u^* denotes the complex conjugate. The function $\mathcal{J}$ models the coherence properties of the illumination source. If the illumination is *coherent*, we have $\mathcal{J}(x) = 1$ and the Hopkins model reduces to

$$I(x) = \left| \int \mathcal{K}(x-x')u(x')dx' \right|^2 = |(\mathcal{K} * u)(x)|^2 .$$

If the illumination is *incoherent*, we have $\mathcal{J}(x) = \delta(x)$, where $\delta(\cdot)$ is the Dirac impulse, and

$$I(x) = \int |\mathcal{K}(x-x')u(x')|^2 \, dx.$$

Situations in between are called *partially coherent*.

2.2 Optimal coherent decomposition

If we define $\mathcal{W}$ as

$$\mathcal{W}(\xi, \xi') = \mathcal{K}(\xi)\mathcal{K}^*(\xi')\mathcal{J}(\xi' - \xi), \tag{3}$$

the Hopkins model can be written as

$$I(x) = \iint u(x')\mathcal{W}(x-x', x-x'')u^*(x'')dx'dx''.$$

For practical purposes (for example, fast simulation algorithms) it is often useful to make a low-rank approximation of $\mathcal{W}$. *Optimal coherent decompositions* [PK94, PGP97] are obtained by truncating the eigenvalue expansion of $\mathcal{W}$

$$\mathcal{W}(\xi, \xi') = \sum_{k=1}^{\infty} \lambda_k \psi_k(\xi) \psi_k^*(\xi'),$$

after the first few (largest) eigenvalues:

$$\mathcal{W}(\xi, \xi') \cong \sum_{k=1}^{r} \lambda_k \psi_k(\xi) \psi_k^*(\xi'). \tag{4}$$

This approximation has an interesting physical interpretation. If we define $\phi_k = \sqrt{\lambda_k}\psi_k$, then the output image intensity of the truncated model can be written as

$$I(x) = \sum_{k=1}^{r} \left|(\phi_k * u)(x)\right|^2, \tag{5}$$

the sum of the image intensities of r coherent sources [RDP+89].

2.3 The calibration problem

In practice the output pattern is inevitably a distorted version of the input mask. As feature sizes have shrunk, accurate models for these distortions have become increasingly important for simulation and for *optical proximity correction*, a technique by which the input mask is predistorted to bring the printed pattern closer to the desired pattern. Two effects contribute to pattern distortions in optical lithography (see figure 2): the optical diffraction in the imaging system, which is accurately modeled by the Hopkins model, and the distortions introduced during the photoresist and etching process, which are much harder to model analytically. To obtain a realistic model of these two effects, one can start from the Hopkins model of the optical system, and perturb it to fit experimental data that include distortions from resist and etching effects. We refer to this problem as *model calibration*. Model calibration is a challenging multidimensional and nonlinear system identification problem. (Note that the Hopkins model itself is already nonlinear.) Moreover in practice the available information from measurements is very limited. Often it is only possible to measure whether the output at a given point exceeds a certain threshold or not, so the estimation problem is very underdetermined.

3 Semidefinite programming formulation

In this section we formulate the calibration problem as a semidefinite programming problem. We limit the discussion to models of the form (5) with real-valued functions ϕ_k and u. The extension to complex-valued functions is straightforward.

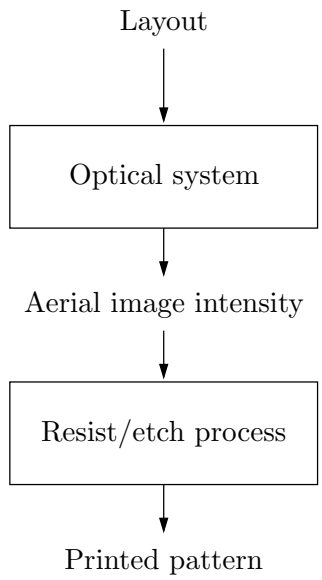

Figure 2: Model of the optical lithography process.

3.1 One-dimensional discrete systems

We first give the details for a discrete one-dimensional system of the form (5), in which x takes values in $\mathbf{Z}$. We also assume that the input u and the impulse responses ϕ_k are real, *i.e.*, $u : \mathbf{Z} \to \mathbf{R}$, $\phi_k : \mathbf{Z} \to \mathbf{R}$ and $I : \mathbf{Z} \to \mathbf{R}$, and that ϕ_k has finite length: $\phi_k(x) = 0$ for $|x| > p$. (This last assumption involves a truncation error that is small for sufficiently large p.) We can then write (5) as

$$\begin{aligned} I(x) = \sum_{k=1}^{r} |(\phi_k * u)(x)|^2 &= \sum_{k=1}^{r} \left(\sum_{\xi=-p}^{p} \phi_k(\xi) u(x-\xi) \right)^2 \\ &= \begin{bmatrix} u(x-p) \\ u(x-p+1) \\ \vdots \\ u(x+p) \end{bmatrix}^T W \begin{bmatrix} u(x-p) \\ u(x-p+1) \\ \vdots \\ u(x+p) \end{bmatrix} \end{aligned}$$

where

$$W = \sum_{k=1}^{r} \begin{bmatrix} \phi_k(p) \\ \phi_k(p-1) \\ \vdots \\ \phi_k(-p) \end{bmatrix} \begin{bmatrix} \phi_k(p) \\ \phi_k(p-1) \\ \vdots \\ \phi_k(-p) \end{bmatrix}^T . \tag{6}$$

Our goal is to estimate W from input-output measurements. We impose several conditions on W.

- W must be symmetric positive semidefinite:

$$W \succeq 0.$$

This follows from the definition (6) and guarantees that $I(x) \geq 0$ for all u.

- W must be *persymmetric* (symmetric with respect to the antidiagonal), *i.e.*,

$$JWJ = W, \tag{7}$$

where J is the identity matrix with the order of its columns reversed. This symmetry property is equivalent to requiring that the output intensity $I(x)$ at position x is invariant with respect to a reversal of the input sequence about the position x.

- W should be close to a prior model W_0, for example, the Hopkins model of the imaging system excluding etching and diffusion effects. We express this condition by imposing a limit ϵ on the deviation from W_0 in matrix (spectral) norm:

$$\|W - W_0\|_2 \leq \epsilon.$$

The choice of norm in this inequality has consequences for the numerical solution of the problem. In section 5 we will describe in detail how the bound can be handled if the spectral norm is used. A bound on the Frobenius norm, for example, would require different techniques. (Note however that if a penalty term $\|W - W_0\|_F^2$ is added to the objective, we obtain a problem that to which the techniques in [TTT07] apply.)

- W must be consistent with K input-output measurements. We assume that each measurement provides an upper and/or lower bound on the intensity $I(x)$ at some position x, in response to a given input. The measurements will be represented as vector inequalities

$$b_{\mathrm{l}} \preceq \mathbf{diag}(U^T W U) \preceq b_{\mathrm{u}},$$

where b_{l}, $b_{\mathrm{u}} \in \mathbf{R}^K$ and $U \in \mathbf{R}^{(2p+1)\times K}$. The kth inequalities on both sides specify upper and lower bounds

$$b_{\mathrm{l},k} \leq I(x) \leq b_{\mathrm{u},k},$$

where $I(x)$ is the intensity at position x resulting from the input values

$$\begin{bmatrix} u(x-p) \\ u(x-p+1) \\ \vdots \\ u(x+p) \end{bmatrix} = U_k$$

if U_k is column k of U, and x is the point at which the kth intensity measurement was made. (Note that the kth component of $\mathbf{diag}(U^T W U)$ is equal to $U_k^T W U_k$.) We allow $b_{\mathrm{l},k} = b_{\mathrm{u},k}$, in which case the constraints reduce to an equality. We also allow $b_{\mathrm{u},k} = \infty$ if there is no upper bound and $b_{\mathrm{l},k} = -\infty$ if there is no lower bound.

Several useful objectives can be represented as linear functions $\mathbf{tr}(DW)$. One example is simply the trace $\mathbf{tr}\, W$, which can be minimized to find low-rank models W (see [FHB04]). Another interesting choice is

$$D = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}, \tag{8}$$

for which we have

$$\mathbf{tr}(DW) = \sum_{k=1}^{r} \sum_{x=-p+1}^{p} (\phi_k(x) - \phi_k(x-1))^2.$$

This can serve as a measure of non-smoothness of the model.

To summarize, we are led to the following semidefinite program in the variable W:

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DW) \\ \text{subject to} & W \succeq 0 \\ & JWJ = W \\ & -\epsilon I \preceq W - W_0 \leq \epsilon I \\ & b_{\mathrm{l}} \preceq \mathbf{diag}(U^T W U) \preceq b_{\mathrm{u}}, \end{array} \tag{9}$$

where D, U, b_{l}, b_{u}, ϵ, and the prior model W_0 are given.

3.2 Two-dimensional discrete systems

The problem formulation of the previous section readily extends to two-dimensional systems. Suppose $u : \mathbf{Z}^2 \to \mathbf{R}$, $\phi_k : \mathbf{Z}^2 \to \mathbf{R}$, and $I : \mathbf{Z}^2 \to \mathbf{R}$, and that $\phi_k(x_1, x_2) = 0$ if $\max\{|x_1|, |x_2|\} > p$. The expression (5) can be written as a quadratic form

$$\begin{aligned} I(x) = \sum_{k=1}^{r} |(\phi_k * u)(x)|^2 &= \sum_{k=1}^{r} \left(\sum_{\xi_1=-p}^{p} \sum_{\xi_2=-p}^{p} \phi_k(\xi) u(x-\xi) \right)^2 \\ &= \mathbf{vec}(u,x)^T W\, \mathbf{vec}(u,x) \end{aligned} \tag{10}$$

where

$$W = \sum_{k=1}^{r} \Phi_k \Phi_k^T \tag{11}$$

and $\mathbf{vec}(u,x)$ and Φ_k are vectors of length $(2p+1)^2$ with components

$$\begin{aligned} \mathbf{vec}(u,x)_{p+1+i+(2p+1)(p+j)} &= u(x_1+i, x_2+j) \\ (\Phi_k)_{p+1+i+(2p+1)(p+j)} &= \phi_k(-i,-j) \end{aligned}$$

for $i = -p, \ldots, p$, $j = -p, \ldots, p$. With this notation, the SDP formulation (9) still applies, except for the bisymmetry constraint $JWJ = W$, which has to be replaced by a more complicated symmetry relation. We will require the intensity equation (10) to be invariant with respect to reflections of u about the two axes through x. In other words, we impose the condition that for all u,

$$\mathbf{vec}(u,x)^T W\, \mathbf{vec}(u,x) = \mathbf{vec}(\widetilde{u},x)^T W\, \mathbf{vec}(\widetilde{u},x) = \mathbf{vec}(\widehat{u},x)^T W\, \mathbf{vec}(\widehat{u},x)$$

where $\widetilde{u}$ and $\widehat{u}$ are the image u reflected about the horizontal, respectively vertical, axis through x:

$$\widetilde{u}(x_1+i, x_2+j) = u(x_1-i, x_2+j), \qquad \widehat{u}(x_1+i, x_2+j) = u(x_1+i, x_2-j),$$

for $i = 1, \ldots, p$, $j = 1, \ldots, p$. We have

$$\mathbf{vec}(\widetilde{u},x) = (I_{2p+1} \otimes J_{2p+1})\, \mathbf{vec}(u,x), \qquad \mathbf{vec}(\widehat{u},x) = (J_{2p+1} \otimes I_{2p+1})\, \mathbf{vec}(u,x),$$

where $\otimes$ denotes the Kronecker product. The symmetry constraint therefore translates into two equality constraints on W:

$$(I \otimes J)W(I \otimes J) = (J \otimes I)W(J \otimes I) = W,$$

(where $I = I_{2p+1}$, $J = J_{2p+1}$). We obtain the SDP

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DW) \\ \text{subject to} & W \succeq 0 \\ & (I \otimes J)W(I \otimes J) = (J \otimes I)W(J \otimes I) = W \\ & -\epsilon I \preceq W - W_0 \leq \epsilon I \\ & b_{\mathrm{l}} \preceq \mathbf{diag}(U^T W U) \preceq b_{\mathrm{u}}. \end{array} \tag{12}$$

4 Symmetry constraints

In this section we use the symmetry constraints in (9) and (12) to reduce the size of the matrix variable W, and therefore the complexity of solving the SDPs. Although the result can be obtained as a special case of [GP04], we give a different argument based on elementary linear algebra.

We first derive a more general result. Suppose a (possibly non-symmetric) real matrix X satisfies

$$PXP = X$$

where P is a symmetric permutation matrix. A symmetric permutation matrix satisfies $P^2 = I$ and therefore has eigenvalues ± 1, so it can be written as

$$P = P_1 P_1^T - P_2 P_2^T$$

where $P_1^T P_1 = I$, $P_2^T P_2 = I$, $P_1^T P_2 = 0$. The columns of P_1 are the eigenvectors with eigenvalue 1; the columns of P_2 are the eigenvectors with eigenvalue -1. Define

$$\tilde{X} = \begin{bmatrix} P_1 & P_2 \end{bmatrix}^T X \begin{bmatrix} P_1 & P_2 \end{bmatrix}.$$

The relation $PXP = X$ is equivalent to

$$\tilde{X} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \tilde{X} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

Therefore $\tilde{X}$ is block-diagonal with two diagonal blocks, *i.e.*, X must be of the form

$$X = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} X_1 & 0 \\ 0 & X_2 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \end{bmatrix}^T. \tag{13}$$

4.1 Bisymmetric matrices

A matrix X is *persymmetric* if XJ is symmetric. A matrix is *bisymmetric* if it is symmetric and persymmetric, *i.e.*, $X = X^T$ and $X = JXJ$.

Since J is a symmetric permutation matrix it follows from (13) that a bisymmetric matrix $X \in \mathbf{S}^N$ can be expressed as

$$X = \begin{bmatrix} P_{\mathrm{e}} & P_{\mathrm{o}} \end{bmatrix} \begin{bmatrix} X_{\mathrm{e}} & 0 \\ 0 & X_{\mathrm{o}} \end{bmatrix} \begin{bmatrix} P_{\mathrm{e}} & P_{\mathrm{o}} \end{bmatrix}^T \tag{14}$$

where $X_{\mathrm{e}} \in \mathbf{S}^{\lceil N/2 \rceil}$, $X_{\mathrm{o}} \in \mathbf{S}^{\lfloor N/2 \rfloor}$, and $P_{\mathrm{e}} \in \mathbf{R}^{N \times \lceil N/2 \rceil}$, $P_{\mathrm{o}} \in \mathbf{R}^{N \times \lfloor N/2 \rfloor}$ are defined as

$$P_{\mathrm{e}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix}, \qquad P_{\mathrm{o}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{N/2} \\ -J_{N/2} \end{bmatrix} \tag{15}$$

if N is even, and

$$P_{\mathrm{e}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{(N-1)/2} & 0 \\ 0 & \sqrt{2} \\ J_{(N-1)/2} & 0 \end{bmatrix}, \qquad P_{\mathrm{o}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{(N-1)/2} \\ 0 \\ -J_{(N-1)/2} \end{bmatrix} \tag{16}$$

if N is odd. Note that ${P_{\mathrm{e}}}^T P_{\mathrm{e}} = I$, ${P_{\mathrm{o}}}^T P_{\mathrm{o}} = I$ and ${P_{\mathrm{e}}}^T P_{\mathrm{o}} = 0$, and that $JP_{\mathrm{e}} = P_{\mathrm{e}}$, $JP_{\mathrm{o}} = -P_{\mathrm{o}}$.

Using the decomposition (14) we can relate the eigendecomposition of X to the eigendecomposition of X_{e} and X_{o}. Suppose $X_{\mathrm{e}} = V_{\mathrm{e}} \Lambda_{\mathrm{e}} V_{\mathrm{e}}^T$ and $X_{\mathrm{o}} = V_{\mathrm{o}} \Lambda_{\mathrm{o}} V_{\mathrm{o}}^T$ with V_{e} and V_{o} orthogonal and Λ_{e} and Λ_{o} diagonal. Then

$$X = \begin{bmatrix} P_{\mathrm{e}} V_{\mathrm{e}} & P_{\mathrm{o}} V_{\mathrm{o}} \end{bmatrix} \begin{bmatrix} \Lambda_{\mathrm{e}} & 0 \\ 0 & \Lambda_{\mathrm{o}} \end{bmatrix} \begin{bmatrix} P_{\mathrm{e}} V_{\mathrm{e}} & P_{\mathrm{o}} V_{\mathrm{o}} \end{bmatrix}^T.$$

The columns of $P_{\mathrm{e}} V_{\mathrm{e}}$ and $P_{\mathrm{o}} V_{\mathrm{o}}$ in this eigenvalue decomposition provide a set of eigenvectors of X with even and odd symmetry: the columns of $P_{\mathrm{e}} V_{\mathrm{e}}$ are even, while the columns of $P_{\mathrm{o}} V_{\mathrm{o}}$ are odd.

The even-odd decomposition of bisymmetric matrices can be used to reduce the SDP (9) to an equivalent problem of smaller size. Here we have $N = 2p + 1$ and the constraint $W = JWJ$ implies that

$$W = P_{\mathrm{e}} W_{\mathrm{e}} P_{\mathrm{e}}^T + P_{\mathrm{o}} W_{\mathrm{o}} P_{\mathrm{o}}^T$$

for some $W_{\mathrm{e}} \in \mathbf{S}^{p+1}$, $W_{\mathrm{o}} \in \mathbf{S}^p$, if we define P_{e}, P_{o} as in (16) with $N = 2p+1$. We will assume that W_0 has the same symmetry, and write

$$W_0 = P_{\mathrm{e}} W_{0,\mathrm{e}} P_{\mathrm{e}}^T + P_{\mathrm{o}} W_{0,\mathrm{o}} P_{\mathrm{o}}^T.$$

The SDP (9) then reduces to

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(D_{\mathrm{e}} W_{\mathrm{e}}) + \mathbf{tr}(D_{\mathrm{o}} W_{\mathrm{o}}) \\ \text{subject to} & W_{\mathrm{e}} \succeq 0, \quad W_{\mathrm{o}} \succeq 0 \\ & -\epsilon I \preceq W_{\mathrm{e}} - W_{0,\mathrm{e}} \preceq \epsilon I, \quad -\epsilon I \preceq W_{\mathrm{o}} - W_{0,\mathrm{o}} \preceq \epsilon I \\ & b_{\mathrm{l}} \preceq \mathbf{diag}(U_{\mathrm{e}}^T W_{\mathrm{e}} U_{\mathrm{e}}) + \mathbf{diag}(U_{\mathrm{o}}^T W_{\mathrm{o}} U_{\mathrm{o}}) \preceq b_{\mathrm{u}}. \end{array}$$

where $D_{\mathrm{e}} = P_{\mathrm{e}}^T D P_{\mathrm{e}}$, $D_{\mathrm{o}} = P_{\mathrm{o}}^T D P_{\mathrm{o}}$, $U_{\mathrm{e}} = P_{\mathrm{e}}^T U$, $U_{\mathrm{o}} = P_{\mathrm{o}}^T U$. This reformulation reduces the number of variables (the elements of W_{e} and W_{o}) to about one half of the number of variables in W. In §5 we will see that the cost of solving the SDP is reduced by about 1/4. Alternatively, we can interpret this technique as a change of variables

$$W := \begin{bmatrix} P_{\mathrm{e}} & P_{\mathrm{o}} \end{bmatrix}^T W \begin{bmatrix} P_{\mathrm{e}} & P_{\mathrm{o}} \end{bmatrix}$$

that transforms the SDP (9) into one of the form

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DW) \\ \text{subject to} & W \succeq 0 \\ & -\epsilon I \preceq W - W_0 \leq \epsilon I \\ & b_{\mathrm{l}} \preceq \mathbf{diag}(U^T W U) \preceq b_{\mathrm{u}}. \end{array} \tag{17}$$

where D, W, W_0 are symmetric block-diagonal matrices with two diagonal blocks, one of order $p + 1$ and one of order p.

4.2 Symmetry in two dimensions

We now extend the result of the previous section to the symmetry constraints in (12). Suppose a matrix $X \in \mathbf{S}^{NM}$ satisfies

$$X = (J_N \otimes I_M) X (J_N \otimes I_M) = (I_N \otimes J_M) X (I_N \otimes J_M). \tag{18}$$

Define P_{e}, P_{o} as in (15) and (16), and Q_{e}, Q_{o} as

$$Q_{\mathrm{e}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{M/2} \\ J_{M/2} \end{bmatrix}, \qquad Q_{\mathrm{o}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{M/2} \\ -J_{M/2} \end{bmatrix}$$

if M is even, and

$$Q_{\rm e} = \frac{1}{\sqrt{2}} \left[\begin{array}{cc} I_{(M-1)/2} & 0 \\ 0 & \sqrt{2} \\ J_{(M-1)/2} & 0 \end{array} \right], \qquad Q_{\rm o} = \frac{1}{\sqrt{2}} \left[\begin{array}{c} I_{(M-1)/2} \\ 0 \\ -J_{(M-1)/2} \end{array} \right]$$

if M is odd. The matrix

$$\left[\begin{array}{cccc} P_{\rm e} \otimes Q_{\rm e} & P_{\rm e} \otimes Q_{\rm o} & P_{\rm o} \otimes Q_{\rm e} & P_{\rm o} \otimes Q_{\rm o} \end{array} \right]$$

is orthogonal, and its columns are eigenvectors of the symmetric permutation matrices $J_N \otimes I_M$ and $I_N \otimes J_M$:

$$(J_N \otimes I_M)(P_{\rm e} \otimes Q_{\rm e}) = P_{\rm e} \otimes Q_{\rm e}, \qquad (J_N \otimes I_M)(P_{\rm e} \otimes Q_{\rm o}) = P_{\rm e} \otimes Q_{\rm o},$$

$$(J_N \otimes I_M)(P_{\rm o} \otimes Q_{\rm e}) = -P_{\rm o} \otimes Q_{\rm e}, \qquad (J_N \otimes I_M)(P_{\rm o} \otimes Q_{\rm o}) = -P_{\rm o} \otimes Q_{\rm o},$$

and

$$(I_N \otimes J_M)(P_{\rm e} \otimes Q_{\rm e}) = P_{\rm e} \otimes Q_{\rm e}, \qquad (I_N \otimes J_M)(P_{\rm o} \otimes Q_{\rm e}) = P_{\rm o} \otimes Q_{\rm e},$$

$$(I_N \otimes J_M)(P_{\rm e} \otimes Q_{\rm o}) = -P_{\rm e} \otimes Q_{\rm o}, \qquad (I_N \otimes J_M)(P_{\rm o} \otimes Q_{\rm o}) = -P_{\rm o} \otimes Q_{\rm o}.$$

It then follows from (13) that X can be decomposed as

$$\begin{aligned} X &= \left[\begin{array}{cc} P_{\rm e} \otimes Q_{\rm e} & P_{\rm e} \otimes Q_{\rm o} \end{array} \right] \left[\begin{array}{cc} X_{\rm ee} & 0 \\ 0 & X_{\rm eo} \end{array} \right] \left[\begin{array}{cc} P_{\rm e} \otimes Q_{\rm e} & P_{\rm e} \otimes Q_{\rm o} \end{array} \right]^T \\ &\quad + \left[\begin{array}{cc} P_{\rm o} \otimes Q_{\rm e} & P_{\rm o} \otimes Q_{\rm o} \end{array} \right] \left[\begin{array}{cc} X_{\rm oe} & 0 \\ 0 & X_{\rm oo} \end{array} \right] \left[\begin{array}{cc} P_{\rm o} \otimes Q_{\rm e} & P_{\rm o} \otimes Q_{\rm o} \end{array} \right]^T, \end{aligned}$$

where $X_{\rm ee} \in \mathbf{S}^{\lceil N/2 \rceil \lceil M/2 \rceil}$, $X_{\rm eo} \in \mathbf{S}^{\lceil N/2 \rceil \lfloor M/2 \rfloor}$, $X_{\rm oe} \in \mathbf{S}^{\lfloor N/2 \rfloor \lceil M/2 \rceil}$, $X_{\rm oo} \in \mathbf{S}^{\lfloor N/2 \rfloor \lfloor M/2 \rfloor}$.

As for the one-dimensional case we can use the symmetry constraints in SDP (12) (where $N = M = 2p+1$) to reduce it to a smaller equivalent SDP of the form (17) in which D, W, and W_0 are block-diagonal with four diagonal blocks of dimension $(p+1)^2$, $(p+1)p$, $(p+1)p$, and p^2. This reduces the number of optimization variables by a factor 4, and the cost per iteration by roughly 16.

5 SDPs with upper bounds

The SDP (17) includes an upper bound ($W \preceq W_0 + \epsilon I$) and two lower bounds ($W \succeq 0$, $W \succeq W_0 - \epsilon I$) on the matrix variable W. Depending on how it is handled by a solver, the inclusion of upper bounds in an SDP can increase the complexity substantially. The purpose of this section is to explain an efficient method for solving SDPs with upper and lower bounds. To simplify the notation we explain the idea for a standard form SDP. (However the ideas also apply to an SDP of the form (17) in which the linear constraints on W are inequalities, rather than equalities $\mathbf{diag}(U^T W U) = b$.)

5.1 Standard SDPs with upper bounds

Consider a standard form SDP with an added upper bound

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_i X) = b_i, \quad i = 1, \ldots, m \\ & 0 \preceq X \preceq I. \end{array} \tag{19}$$

The variable is $X \in \mathbf{S}^n$. One obvious solution for handling the upper bound, analogous to techniques common in linear programming, is to introduce a slack variable V and solve

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_i X) = b_i, \quad i = 1, \ldots, m \\ & X + V = I \\ & X \succeq 0, \quad V \succeq 0. \end{array} \tag{20}$$

Unfortunately this introduces a very large number $(n(n+1)/2)$ of new equality constraints, so this SDP is much harder to solve using general-purpose software than the SDP without upper bounds. It is therefore important to develop efficient techniques to incorporate upper bounds directly in interior-point algorithms. The method described below also appeared in [TTT07].

5.2 Interior-point method

We discuss a variation of the interior-point method of [TTT02]; see also the appendix in [VBW+05]. Skipping details that are well documented elsewhere, we note that each iteration of the algorithm applied to the standard SDP (19) involves the solution of a set of linear equations (the 'Newton equations')

$$-T^{-1}\Delta X T^{-1} + \sum_{i=1}^{m} \Delta y_i A_i = D \tag{21}$$

$$\mathbf{tr}(A_i \Delta X) = d_i, \quad i = 1, \ldots, m. \tag{22}$$

The variables are the primal and dual search directions ΔX and Δy. The matrix $T \succ 0$ and the righthand sides D, d change at each iteration. The Newton equations are usually solved by eliminating ΔX from the first equation and solving a set of m equations in m variables $H\Delta y = g$ where

$$H_{ij} = \mathbf{tr}(TA_iTA_j), \qquad g_i = d_i + \mathbf{tr}(A_iTDT), \qquad i, j = 1, \ldots, m.$$

The cost of this method is the cost of assembling the matrix H (which includes terms proportional to mn^3 and m^2n^2), plus the cost of solving the equations (proportional to m^3).

The Newton equations for the SDP with upper bounds (20) are

$$\begin{aligned} -T_1^{-1}\Delta X T_1^{-1} + \sum_{i=1}^m \Delta y_i A_i + \Delta Y &= D_1 \\ -T_2^{-1}\Delta V T_2^{-1} + \Delta Y &= D_2 \\ \mathbf{tr}(A_i \Delta X) &= d_i, \quad i = 1, \ldots, m \\ \Delta X + \Delta V &= D_3. \end{aligned}$$

ΔX, ΔV are the (primal) search directions; Δy, ΔY are dual search directions. The two matrices T_1 and T_2 are positive definite and change at each iteration. An efficient solution method that exploits the specific structure in these equations proceeds as follows. We first eliminate ΔV and ΔY:

$$-T_1^{-1}\Delta X T_1^{-1} - T_2^{-1}\Delta X T_2^{-1} + \sum_{i=1}^m \Delta y_i A_i = D \tag{23}$$

$$\mathbf{tr}(A_i \Delta X) = d_i, \quad i = 1, \ldots, m, \tag{24}$$

where $D = D_1 - D_2 - T_2^{-1} D_3 T_2^{-1}$. Comparing with the Newton equations for the standard form SDP (21), we note that the addition of the second term in (23) makes it harder to eliminate ΔX.

We therefore first determine a congruence transformation that simultaneously diagonalizes T_1 and T_2,

$$R^T T_1 R = I, \qquad R^T T_2 R = \mathbf{diag}(\gamma)^{-1},$$

where $\gamma \succ 0$ (see [GL89, p. 469]). If we define $\Delta\tilde{X} = R^T \Delta X R$, $\tilde{A}_i = R^{-1} A_i R^{-T}$, the Newton equations reduce to

$$\begin{aligned} -\Delta\tilde{X} - \mathbf{diag}(\gamma)\Delta\tilde{X}\,\mathbf{diag}(\gamma) + \sum_{i=1}^m \Delta y_i \tilde{A}_i &= R^{-1} D R^{-T} \\ \mathbf{tr}(\tilde{A}_i \Delta\tilde{X}) &= d_i, \quad i = 1, \ldots, m. \end{aligned}$$

From the first equation, we can express $\Delta\tilde{X}$ in terms of Δy:

$$\Delta\tilde{X} = \sum_{i=1}^m \Delta y_i (\tilde{A}_i \circ \Gamma) - (R^{-1} D R^{-T}) \circ \Gamma$$

where $\Gamma_{ij} = 1/(1 + \gamma_i \gamma_j)$ and $\circ$ denotes the Hadamard product. Substituting this in the second equation gives a set of equations $H\Delta y = g$ where

$$H_{ij} = \mathbf{tr}(\tilde{A}_i(\tilde{A}_j \circ \Gamma)) = \mathbf{tr}((\tilde{A}_i \circ \tilde{A}_j)\Gamma)), \quad i, j = 1, \ldots, m.$$

After solving for Δy, one easily obtains ΔX from (23).

The cost of this method is dominated by the cost of computing the matrices $\tilde{A}_i$ ($O(mn^3)$ flops), the cost of assembling H ($O(m^2n^2)$), and the cost of solving for Δy ($O(m^3)$). For

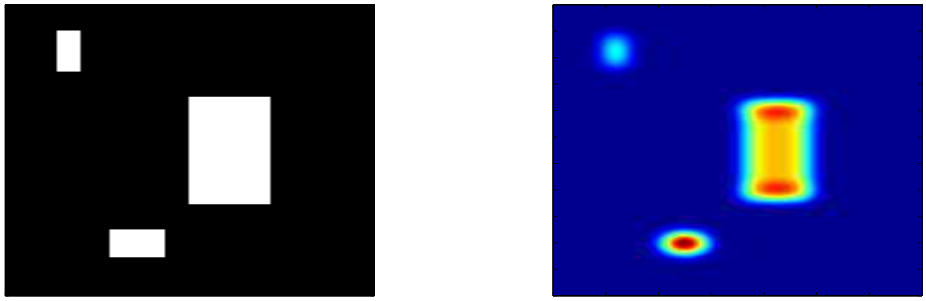

Figure 3: Mask and the output intensity image used to calibrate the optical system.

dense coefficient matrices A_i, the overall cost is therefore comparable to the cost of solving the Newton equations for the standard form SDP without upper bounds.

If the matrices A_i are sparse or low rank, as in the model calibration application, additional savings are possible, but the complexity of solving the SDP with upper bounds is higher than for an SDP problem without upper bounds (see also [TTT07]). For example, if $A_i = a_i a_i^T$, the matrix H can be expressed as

$$H_{ij} = b_i^T((b_j b_j^T) \circ \Gamma) b_i = (b_i \circ b_j)^T \Gamma (b_i \circ b_j)$$

where $b_i = R^{-1} a_i$, and assembled in $O(m^2 n^2)$ operations. For a standard form SDP with rank-one matrices and no upper bounds, the cost of assembling H would be $O(\max\{m^2 n, n^2 m\})$ operations.

6 Example

In this section we apply the SDP calibration method to a simulated optical system. We first create an 'ideal' model W_0 of dimension 1849×1849 (*i.e.*, $p = 21$ in the notation of section 3.2) by discretizing (3), with $\mathcal{J}$ and $\mathcal{K}$ given by

$$\mathcal{J}(x) = 2\frac{J_1(\sigma|x|)}{\sigma|x|}, \qquad \mathcal{K}(x) = \frac{J_1(|x|)}{|x|},$$

where J_1 is the first Bessel function of the first kind [Won01]. These values correspond to a circularly symmetric optical system, with a source in the form of a disk with radius σ.

We then add a randomly generated perturbation ΔW of the order of 5% (*i.e.*, $\|\Delta W\| = 0.05\|W_0\|$) to represent lens imperfections and resist effects. We use $\hat{W} = W_0 + \Delta W$ to compute the output image (shown in figure 3), and sample the output image at $K = 2500$ points. We then solve problem (12) with D given by (8) and equality constraints:

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DW) \\ \text{subject to} & W \succeq 0 \\ & (J \otimes I)W(J \otimes I) = (I \otimes J)W(I \otimes J) = W \\ & -\epsilon I \preceq W - W_0 \leq \epsilon I \\ & \mathbf{diag}(U^T W U) = b. \end{array} \tag{25}$$

We used the value $\epsilon = 79\|W_0\|$.

The variable W in the SDP has order 1849, and there are 2500 equality constraints. After exploiting the symmetries in the problem, the variable W is transformed to a block diagonal matrix with four blocks of dimension 484, 462, 462 and 441, respectively.

A Matlab implementation of the algorithm converged in 24 iterations and 116 minutes (on a 2.8GHz Pentium IV machine). This is not unreasonable for an SDP of this size, considering that the model calibration is required only once for a given optical lithography system.

To test the quality of the calibrated model W_{cal} obtained by solving (25), we choose a different mask and compute the output image from the model $\hat{W} = W_0 + \Delta W$ (the top images in figure figure 4). Figure 4 also shows the outputs of the uncalibrated model W_0 (bottom left) and the calibrated model W_{cal} (botrom right). We notice that the calibrated model output approaches the true output of figure 4 very closely, and much better than the uncalibrated model W_0. The error (Euclidean norm of the difference between the predicted and correct output images divided by the Euclidean norm of the correct image) is about 0.46%, compared with an error of 24.3% for the uncalibrated model.

Figure 5 shows the result of the same experiment repeated with a 15% perturbation ΔW. The error of the output predicted by the calibrated model is 8.6%. The error for the uncalibrated model is 76.3%.

7 Conclusion

We have described an application of semidefinite programming in optical lithography. The problem can be viewed as a nonlinear model estimation problem, with a model that expresses the system output as a sum of squares of linear convolutions of the input. We have also outlined several techniques for exploiting problem structure in an implementation of an interior-point method for solving the resulting large-scale SDPs. These techniques include methods for handling upper bounds on the matrix variable, symmetry constraints, and rank-one structure in the coefficient matrices.

Acknowledgment This research was supported by the UC-MICRO program under project 01-098, a grant from Numerical Technologies, and NSF project ECS-0524663.

References

[BYZ00] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10:443–461, 2000.

[FHB04] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of American Control Conference*, pages 3273–3278, 2004.

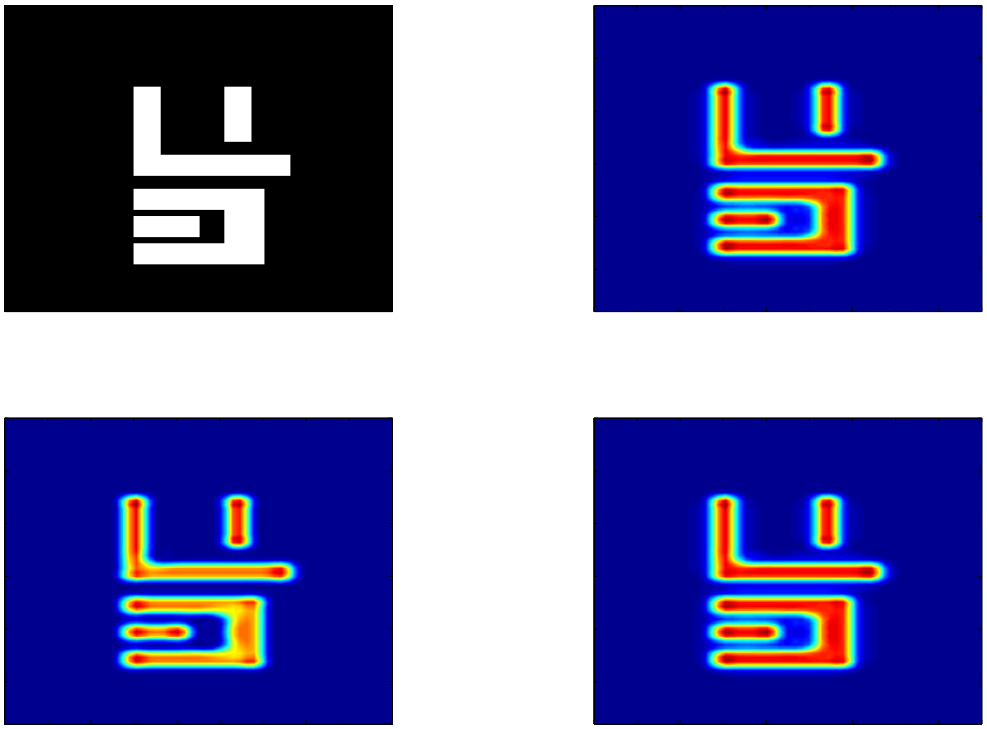

Figure 4: *Top left.* Sample mask used to evaluate the calibrated optical model. *Top right.* The output intensity of the optical system $W_0 + \Delta W$, where $\|\Delta W\| = 0.05\|W_0\|$. *Bottom left.* The output intensity of the uncalibrated optical model W_0. *Bottom right.* The output intensity of the calibrated optical model W_{cal}.

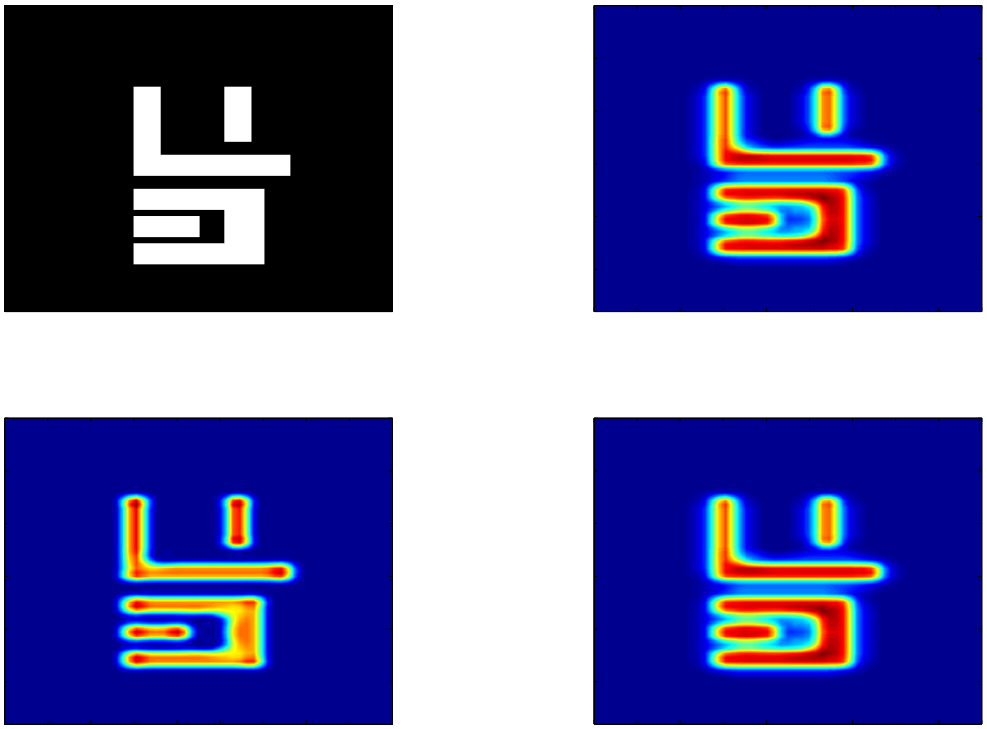

Figure 5: *Top left.* Sample mask used to evaluate the calibrated optical model. *Top right.* The output intensity of the optical system $W_0 + \Delta W$, where $\|\Delta W\| = 0.15\|W_0\|$. *Bottom left.* The output intensity of the uncalibrated optical model W_0. *Bottom right.* The output intensity of the calibrated optical model W_{cal}.

[GL89] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, Baltimore, second edition, 1989.

[GP04] K. Gatermann and P. A. Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192:95–128, 2004.

[Hop51] H. H. Hopkins. The concept of partial coherence in optics. *Proceedings of the Royal Society of London. Series A. Mathematical and Pysical Sciences*, 208(1093):263–277, 1951.

[PGP97] Y. C. Pati, A. A. Ghazanfarian, and R. F. Pease. Exploiting structure in fast aerial image computation for integrated circuit patterns. *IEEE Transactions on Semiconductor Manufacturing*, 10:62–74, 1997.

[PK94] Y. C. Pati and T. Kailath. Phase-shifting masks for microlithography: automated design and mask requirements. *Journal of the Optical Society of America, Series A*, 11:2438–2452, 1994.

[RDP+89] G. O. Raynolds, J. B. Devilis, G. B. Parrent, Jr., and B. J. Thomson. *Physical Optics Notebook: Tutorial in Fourier Optics*. SPIE Optical Engineering Press, 1989.

[RV06] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming and sum-of-squares representations of nonnegative polynomials. *SIAM Journal on Optimization*, 16:939–964, 2006.

[TTT02] K. C. Toh, R. H. Tütüncü, and M. J. Todd. *SDPT3 version 3.02. A Matlab software for semidefinite-quadratic-linear programming*, 2002. Available from `www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

[TTT07] K. C. Toh, R. H. Tütüncü, and M. J. Todd. Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems. *Pacific Journal of Optimization*, 3, 2007.

[VBW+05] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. H. Hanson, and T. Roh. Interior point algorithms for semidefinite programming problems derived from the KYP lemma. In D. Henrion and A. Garulli, editors, *Positive Polynomials in Control*, volume 312 of *Lecture Notes on Control and Information Sciences*. Springer Verlag, 2005.

[Won01] A. Kwok-Kit Wong. *Resolution Enhancement Techniques in Optical Lithography*. SPIE, 2001.