A Parameterized Mask Model for Lithography Simulation

Zhenhai Zhu
Cadence Research Labs, Berkeley, CA 94704
zhzhu@cadence.com

1. ABSTRACT

We formulate the mask modeling as a parametric model order reduction problem based on the finite element discretization of the Helmholtz equation. By using a new parametric mesh and a machine learning technique called Kernel Method, we convert the nonlinearly parameterized FEM matrices into affine forms. This allows the application of a well-understood parametric reduction technique to generate compact mask model. Since this model is based on the first principle, it naturally includes diffraction and couplings, important effects that are poorly handled by the existing heuristic mask models. Further more, the new mask model offers the capability to make a smooth trade-off between accuracy and speed.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aides—*Simulation*

General Terms

Algorithms,Performance,Design

Keywords

Lithography, Mask Model, Parameterized Model Order Reduction

2. INTRODUCTION

From lithography simulation point of view, two classes of designs pose particularly challenging problems. The first class is the memory design. With just six or seven transistors, each memory cell has small layout. The carefully designed cell is duplicated hundred millions to billions of times. Since the manufacturing defects in one cell affect the entire row of cells, the memory design is highly sensitive to such defects. Therefore, high accuracy lithography simulation is mandatory for the memory design. The second class is the custom logic design. With a finite number of unique standard cells as the building blocks, the layout is typically large, in the order of $1 \times 1mm^2$. Therefore, highly efficient lithography simulation is mandatory for the custom logic design. Since timing and power are the main concerns, the accuracy of lithography simulation has to be good enough to model the impact of manufacturing imperfection to the timing and power. Clearly, a good lithography simulation tool should inherently have the capability to make smooth trade-off between accuracy and speed for various designs. And it should be based on rigorous mathematical foundation to ensure its robustness.

DAC 2009, July 26 - 31, 2009, San Francisco, California, USA.

There are three main steps in lithography simulation [1]: photo mask modeling, aerial image simulation and photo resist simulation. This paper focuses on the photo mask modeling. The goal of this step is to compute the near field, the electric field at the bottom of the computational domain that contains the mask pattern. The existing approaches can be categorized into two groups: field solvers and heuristic models.

The field solver approach is to solve the Maxwell's equations using well-known numerical techniques such as finite difference time domain (FDTD) [2] or finite element method (FEM) [3]. This is the most accurate and robust approach. But experience indicates that even the state-of-the-art field solvers are too slow or memory-bounded to handle a medium-size mask pattern.

In practice, the heuristic models are used instead to obtain the approximate solution. A commonly used model is based on the so-called Kirchhoff approximation [4]: if there is a mask opening, the light shines through it without any change in magnitude and phase; otherwise, light is completely blocked. This approximation neglects the effects such as diffraction, polarization and coupling. Attempts have been made to obtain a modified mask model to improve the accuracy [5, 6]. The piecewise constant curve fitting approach in [5, 6] is simple and efficient but not accurate and robust. For example, it has been shown in [7] that such model can result in wrong wafer imaging prediction.

The field solvers in [2, 3] and the heuristic models in [5, 6] sit at the opposite corners in the accuracy-speed trade-off matrix and it is difficult to smoothly trade accuracy with speed. In our early work [8], the mask modeling was formulated as a parametric model order reduction problem based on the finite element discretization of the Helmholtz equation. The discretization in [8] uses a uniform and rectangular mesh. This natually leads to the affine parametric form for the stiff and the mass matrices in FEM and hence the well-understood parametric reduction technique [9, 10] can be directly applied to generate the compact mask model. However, the non-uniform and unstructure mesh is indispensable to handle complicated mask patterns. Unfortunately, as will be shown in section 6, a direct application of the technique in [9] could result in large mask models.

In this paper, we present a new approach to approximate the nonlinearly parameterized FEM matrices with affine forms. This allows the direct application of the parametric reduction in [9, 10] again. Though we demonstrate this new technique for the Helmholtz equation that governs the photo mask modeling, it should be straightforward to apply the same approach to other partial differential equations in the context of the parametric model order reduction.

3. PROBLEM FORMULATION

For the sake of simplicity, we present the problem formulation based on the 2D example shown in Fig 1. The extension to 3D cases is straightforward.

Assuming an S-polarization (TE) case, the governing 2D Helmholtz

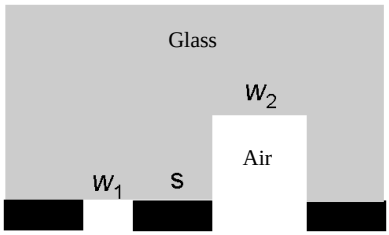

Figure 1: The structure of the 2D phase-shift mask. Variables w_1, w_2 and s are the parameters in the reduced mask model.

equation is

$$\nabla^2 u - \omega^2 \varepsilon \mu u = \nabla^2 u + k^2 u = 0 \tag{1}$$

where $u(x,y)$ is the z-component of the total electric field, ω is the frequency, ε and μ are respectively the dieletric constant and permeability. Following the standard FEM procedure [11], we obtain the weak form

$$\int_\Omega ds \nabla v \cdot \nabla u - \omega^2 \int_\Omega ds \varepsilon \mu v u - \int_{\partial\Omega} dl v \mathrm{DtN}(u) = -2 \int_{\partial\Omega} dl v \mathrm{DtN}(u_{\mathrm{in}}), \tag{2}$$

where Ω and $\partial\Omega$ are respectively the computational domain and its boundary, v is the testing function, u_{in} is the known incidence plane wave and the $\mathrm{DtN}(\dot{})$ operator defines the transparent boundary condition [12]. In this paper, we assume the periodic boundary condition at the east and the west side of the computational domain and transparent or non-reflecting boundary condition on the north and the south side of the computational domain.

Using the standard FEM piecewise polynomial basis functions [11] to discretize (2), we arrive at the parameterized system equation

$$[S(\bar{w},\bar{s}) - M(\bar{w},\bar{s}) - B]\bar{u} = \bar{r}, \tag{3}$$

where vector $\bar{w}$ and $\bar{s}$ respectively contain the widths and the spacings in the layout, the stiff matrix $S(\bar{w},\bar{s})$ corresponds to the first term in (2), the mass matrix $M(\bar{w},\bar{s})$ corresponds to the second term in (2), the matrix B corresponds to the third term in (2), and vector $\bar{r}$ corresponds to the right-hand-side term in (2).

For mask patterns with fixed topology but different width and spacing values, computing the near field involves solving equation (3) for different $\bar{w}$ and $\bar{s}$. This kind of multiple-inquery scenario is precisely what the parametric model order reduction approach is designed for.

4. PARAMETRIC MODEL ORDER REDUCTION

A parametric model order reduction technique called Reduced Basis method has been developed by the finite element research community [9]. A similar idea has also been independently proposed in the area of parameterized model order reduction for circuit simulation [10]. As will be shown later, the Reduced Basis is a very powerful idea on top of which we can build our new mask model to achieve the desirable accuracy and speed trade-off. Here we summarize its key steps. Please refer to [9] for more details.

Suppose the parameterized governing equation for the problem at hand is

$$A(\bar{\sigma})\bar{u} = \left[A_0 + \sum_i f_i(\bar{\sigma}) A_i\right] \bar{u} = \bar{r}, \tag{4}$$

where the scalar function $f_i(\bar{\sigma})$ can be arbitrary and the size of vector $\bar{u}$, $\bar{r}$ and constant matrix A_i is $N \times 1$, $N \times 1$ and $N \times N$, respectively. In practical applications, N can be as large as a few millions for a medium-sized 3D structure. The Reduced Basis method in [9] has two stages: the off-line pre-characterization and the on-line evaluation.

Off-line pre-characterization stage. We randomly generate a set $\bar{\sigma}^k = \{\sigma_1^k, \sigma_2^k, ...\}$ using the given ranges of σ_i and solve (4) for $\bar{u}_k$. After a few sampling solves, we collect all solutions into the projection matrix

$$P = [\bar{u}_1, \bar{u}_2, ..., \bar{u}_M] \tag{5}$$

and perform projection

$$\hat{A}_i = P^T A_i P; \;\; \hat{r} = P^T \bar{r}. \tag{6}$$

Now we arrive at the reduced governing equation

$$\left[\hat{A}_0 + \sum_i f_i(\bar{\sigma}) \hat{A}_i\right] \hat{u} = \hat{r}, \tag{7}$$

where the size of matrix $\hat{A}_i$ is $M \times M$ and M is the number of sampling solves.

Similar to the standard procedure in the Model Order Reduction [10], the columns in the projection matrix P are orthogonalized using techniques such as incremental QR decomposition. This makes the matrix P well conditioned. In addition, both theoretical and practical ways to estimate the error of the reduced model are readily available [9, 10]. Hence the off-line model generation can be made incremental.

On-line evaluation stage. We substitute the given set $\bar{\sigma}^*$ into (7) and solve for $\hat{u}$. The approximate solution to equation (4) is obtained from

$$u = P\hat{u}. \tag{8}$$

The key observation here is that the CPU time of the on-line stage is only related to M, not to N in (4). And M is typically many orders of magnitude smaller than N, as shown by the extensive experiments in [9, 10]. Hence equation (7) is a much more efficient but still accurate reduced model than the original model in (4). However, this dramatic efficiency gain critically depends on the fact that matrix A_i in (4) is not a function of $\bar{\sigma}$. Otherwise, the projection step in (6) involves calculating $A_i(\bar{\sigma})$ at a particular value $\bar{\sigma}^k$. This essentially means that the CPU time used by the reduced model in (7) is related to the original problem size N and hence we have gained no efficiency at all [9, 13]. This issue of representing the potentially arbitrary nonlinearity in $A(\bar{\sigma})$ in the form amenable to the projection framework in (6) is one of the main challenges in the nonlinear Model Order Reduction [14].

The main contribution in this paper is to show how to convert $S(\bar{w},\bar{s})$ and $M(\bar{w},\bar{s})$ into the appropriate form similar to that in equation (4) so that we can apply the congruence projection in (6). This is done in two steps: parameterization of mesh and parameterization of the FEM matrices.

5. PARAMETERIZE MESH

In this section, we show an effective technique to parameterize the unstructured triangular mesh in an affine form of the geometry parameters $\bar{w}$ and $\bar{s}$. This is an important first step toward parameterizing the stiff matrix $S(\bar{w},\bar{s})$ and the mass matrix $M(\bar{w},\bar{s})$ in (3).

When the size of a geometric feature changes, say w_2 in Fig 1, the mesh points surrounding the feature will move as well. However, to capture the changes in the resulting electric field, not all mesh points in the computational domain have to be moved. Only mesh points within a certain distance from the changing geometric feature need to be moved. To measure such a distance, we borrow a well-established concept called distance function from the celebrated Level Set Method [15].

5.1 The Distance Function

We use a simple example to explain the basic ideas in the distance function. Let (x_1,y_1) and (x_2,y_2) be the lower-left and upper-right corner of a rectangle, respectively. The distance from a point (x,y) to such a rectangle is defined as

$$d(x,y) = -min(min(y-y_1,y_2-y),min(x-x_1,x_2-x)), \quad (9)$$

where function $min(a,b)$ returns the smaller value of the two variables. Fig. 2 shows the contour plot of the distance function for a square where $x_1 = y_1 = 5$ and $x_2 = y_2 = 10$. It should be noted that the zero level set corresponds to the boundary of the square. All points outside of the square have positive distance and all points inside the square have negative distance.

5.2 The Blending Function

The movement of mesh points due to geometry changes depend on the distance of the mesh points to the changing geometry. Intuitively, the function that characterizes such movement should satisfy the following constrains

$$b(\eta) = 1, \quad if \ \eta = 0 \quad (10)$$

$$\frac{db}{d\eta} < 0, \quad if \ 0 < \eta < 1 \quad (11)$$

$$b(\eta) = 0, \quad if \ \eta \geq 1 \quad (12)$$

$$\frac{db}{d\eta} = 0, \quad if \ \eta = 1 \quad (13)$$

where η is the normalized distance defined as

$$\eta = \frac{d(x,y)}{B}, \quad (14)$$

B is a user-defined radius of influence, and $d(x,y)$ is the distance function like the one defined in (9). Equation (10) means that the mesh points on the geometry move by the same amount as that of the moving geometry features. Equation (11) means that the movement magnitude decreases monotonically as the mesh points are away from the geometry. Equation (12) means that the movement magnitude is zero if the mesh point is not inside the influence region. Equation (13) ensures a smooth transition of the movement at the boundary of the influence region. In spirit, function $b(\eta)$ is similar to the so-called blending function in [16] used to generate the mesh for the computational domain with moving boundaries. In this paper we have chosen the following function as the blending function

$$b(\eta) = 2(1-p(\frac{\eta}{2}+0.5)), \quad \eta \in [0,1] \quad (15)$$

where $p(\eta)$ is a third-order polynomial

$$p_3(\eta) = 3\eta^2 - 2\eta^3. \quad (16)$$

5.3 The Parametric Form of Mesh

Armed with the blending function in (15) based on the distance function like the one defined in (9), we are ready to parameterize the movement of the mesh points due to the change of geometric features. Without loss of generality, we use the parameter w_2 in Fig. 1 as an example. Suppose w_2 is changed by δw_2. Further more, again without loss of generality, suppose that the center of the opening does not move, only the left and the right wall move by $-\frac{\delta w_2}{2}$ and $\frac{\delta w_2}{2}$, respectively. Within the influence region around the left and the right side wall, the location (x,y) of a mesh point can be expressed as

$$x = x_0 + \frac{b_R - b_L}{2}\delta w_2 = x_0 + b_2 \delta w_2, \quad (17)$$

$$y = y_0 \quad (18)$$

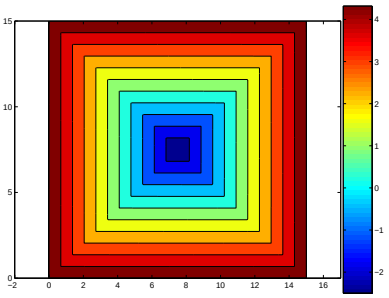

Figure 2: Contour plot of the distance to a square.

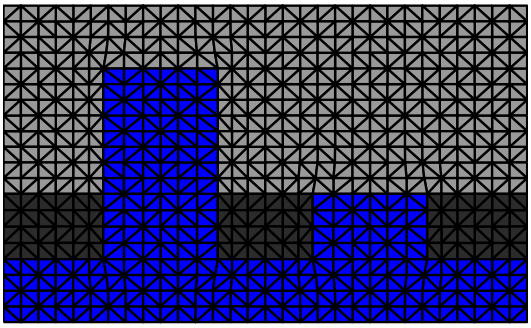

Figure 3: Triangular mesh for the 2D mask in Fig. 1. Notice the deformation of the 90-degree triangles due to the change in w_1 and w_2.

where

$$b_L = b(\frac{d_L(x_0,y_0)}{B}), \quad b_R = b(\frac{d_R(x_0,y_0)}{B}) \quad (19)$$

(x_0,y_0) is the initial location, and $d_L(x_0,y_0)$ and $d_R(x_0,y_0)$ are its distance to the left and the right side wall, respectively.

Using linear superposition, one can easily generalize (17) to the multiple parameter cases

$$x = x_0 + <\bar{b},\bar{\sigma}> \quad (20)$$

where $<\cdot;\cdot>$ is the inner product, and b_i is the blending term for parameter σ_i which represents the change of w_1, w_2 or s in Figure 1. Clearly the parametric form in (20) is an affine function of the perturbation in each geometry parameter. It should be pointed out that the simple parametric form in (17) and (18) can be easily extended to handle parameter changes along arbitrary directions. But for the purpose of modeling masks, it is probably sufficient to consider just the cases where the parameters are changing only along the horizontal direction (x-direction for 2D). This will be the case in the remaining part of this paper. Figure 3 shows the deformation in the triangular mesh for the mask shown in Fig. 1 due to the change in w_1 and w_2.

6. PARAMETERIZE THE FEM MATRICES

For the isoparametric linear triangular elements defined on the triangle element with vertexes $(x_i,y_i), i = 1,2,3$, the element stiff

matrix and mass matrix are respectively [11]

$$S^e = W\frac{\bar{\delta}_x\bar{\delta}_x^T + \bar{\delta}_y\bar{\delta}_y^T}{4A}W^T \tag{21}$$

and

$$M^e = 4AM_c \tag{22}$$

where A is the area of the triangle and can be written as

$$A = \frac{1}{2}|det([\bar{\delta}_x, \bar{\delta}_y])|, \tag{23}$$

$$\bar{\delta}_x = [x_1 - x_3, x_2 - x_1]^T, \tag{24}$$

$$\bar{\delta}_y = [y_3 - y_1, y_1 - y_2]^T, \tag{25}$$

$$W = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{26}$$

$$M_c = \frac{1}{24}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}. \tag{27}$$

6.1 The Parametric Element FEM Matrices

Suppose a triangle element is within the region of the influence defined in section 5, in view of (20) and (18), (24) and (25) become

$$\bar{\delta}_x = \begin{bmatrix} x_1 - x_3 \\ x_2 - x_1 \end{bmatrix} = \begin{bmatrix} x_{1,0} - x_{3,0} + < \bar{d}_{13}, \bar{\sigma} > \\ x_{2,0} - x_{1,0} + < \bar{d}_{21}, \bar{\sigma} > \end{bmatrix} \tag{28}$$

$$\bar{\delta}_y = \begin{bmatrix} y_{3,0} - y_{1,0} \\ y_{1,0} - y_{2,0} \end{bmatrix} \tag{29}$$

where $(x_{i,0}, y_{i,0})$ is the nominal position for the node-i, $d_{ij}^{(k)} = b_i^{(k)} - b_j^{(k)}$ and $b_i^{(k)}$ is the k-th component of the vector $\bar{b}$ in (20) for node-i. For the sake of clarity, we assume that the triangle is a 90-degree triangle that satisfies

$$\begin{bmatrix} x_{1,0} - x_{3,0} \\ x_{2,0} - x_{1,0} \end{bmatrix} = \begin{bmatrix} 0 \\ h_x \end{bmatrix}, \quad \begin{bmatrix} y_{3,0} - y_{1,0} \\ y_{1,0} - y_{2,0} \end{bmatrix} = \begin{bmatrix} h_y \\ 0 \end{bmatrix}, \tag{30}$$

then we have

$$\bar{\delta}_x = h_x \begin{bmatrix} < \frac{\bar{d}_{13}}{h_x}, \bar{\sigma} > \\ 1 + < \frac{\bar{d}_{21}}{h_x}, \bar{\sigma} > \end{bmatrix} = h_x \begin{bmatrix} Y \\ 1 + Z \end{bmatrix} \tag{31}$$

$$\bar{\delta}_y = h_y \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{32}$$

$$A = 0.5h_xh_y(1+Z) \tag{33}$$

where $Y = < \frac{\bar{d}_{13}}{h_x}, \bar{\sigma} >$ and $Z = < \frac{\bar{d}_{21}}{h_x}, \bar{\sigma} >$. Substituting (31), (32) and (33) into (21), we obtain the parametric element stiff matrix

$$\begin{aligned} S^e &= \frac{h_x}{2h_y}W\begin{bmatrix} \frac{Y^2}{1+Z} & Y \\ Y & 1+Z \end{bmatrix}W^T + \frac{h_y}{2h_x}\frac{1}{1+Z}W\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}W^T \\ &= \frac{h_x}{2h_y}\left(W\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}W^T + W\begin{bmatrix} 0 & Y \\ Y & Z \end{bmatrix}W^T\right) \\ &+ \frac{Y^2}{1+Z}\frac{h_x}{2h_y}W\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}W^T + \frac{1}{1+Z}\frac{h_y}{2h_x}W\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}W^T \\ &= S_0^e + \sum_{k=1}^{N_p}\sigma_k S_k^e + \frac{Y^2}{1+Z}S_{N_p+1}^e + \frac{1}{1+Z}S_{N_p+2}^e \end{aligned} \tag{34}$$

where

$$S_0^e = \frac{h_x}{2h_y}W\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}W^T, \quad S_k^e = \frac{1}{2h_y}W\begin{bmatrix} 0 & d_{13}^{(k)} \\ d_{13}^{(k)} & d_{21}^{(k)} \end{bmatrix}W^T \tag{35}$$

$$S_{N_p+1}^e = \frac{h_x}{2h_y}W\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}W^T, \quad S_{N_p+2}^e = \frac{h_y}{2h_x}W\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}W^T. \tag{36}$$

In view of (33) and (22), the parametric element mass matrix is

$$M^e(\bar{\sigma}) = 2h_xh_y(1+Z)M_c = M_0^e + \sum_{k=1}^{N_p}\sigma_k M_k^e \tag{37}$$

where $M_0^e = 2h_xh_yM_c$ and $M_k^e = 2h_yM_cd_{21}^{(k)}$.

6.2 The Assembled FEM Matrices

The standard way of generating matrices $S(\bar{w}, \bar{s})$ and $M(\bar{w}, \bar{s})$ is by a procedure called stamping. The 3×3 element stiff matrix $S^e(\bar{w}, \bar{s})$ and the 3×3 element mass matrix $M^e(\bar{w}, \bar{s})$ are first generated for each element. Using the local-to-global vertex index map, the entries of S^e and M^e are directly added (stamped) to the large matrices $S(\bar{w}, \bar{s})$ and $M(\bar{w}, \bar{s})$, respectively. The parametric stiff matrix is

$$S(\bar{\sigma}) = S_0 + \sum_{k=1}^{N_p}\sigma_k S_1^{(k)} + \sum_{i=1}^{N_1} f_i(\bar{\sigma})S_2^{(i)} + \sum_{j=1}^{N_2} g_j(\bar{\sigma})S_3^{(j)} \tag{38}$$

where matrices S_0, $S_1^{(k)}$, $S_2^{(i)}$ and $S_3^{(j)}$ are respectively the reults of the stamping of the element matrices S_0^e, S_k^e, $S_{N_p+1}^e$ and $S_{N_p+2}^e$ in (34),

$$f_i(\bar{\sigma}) = \frac{Y_i^2}{1+Z_i}, \quad g_j(\bar{\sigma}) = \frac{1}{1+Z_j}, \tag{39}$$

$$Y_i = < \frac{\bar{d}_{13}^{(i)}}{h_x}, \bar{\sigma} >, \quad Z_i = < \frac{\bar{d}_{21}^{(i)}}{h_x}, \bar{\sigma} >, \tag{40}$$

N_1 and N_2 are respectively the number of unique $f_i(\bar{\sigma})$ and $g_j(\bar{\sigma})$ for all the triangle elements in the computation domain, N_p is the number of parameters in the vector $\bar{\sigma}$. Similarly, the parametric mass matrix is

$$M(\bar{\sigma}) = M_0 + \sum_{k=1}^{N_p}\sigma_k M_1^{(k)}, \tag{41}$$

where matrices M_0 and $M_1^{(k)}$ are respectively the reults of the stamping of the element matrices M_0^e and M_k^e in (37).

6.3 Regression Using the Kernel Method

The parametric forms in (38) and (41) are very similar to that in (4). But since N_1 and N_2 are related to the number of elements in the computational domain, so is the size of the reduced model. Hence we have not obtained a mask model that is independent of the original problem size yet. In fact, this is the main reason why we can not directly apply the technique in [9] to generate the reduced mask model. The next critical task is to approximately represent the large number of $f_i(\bar{\sigma})$ and $g_j(\bar{\sigma})$ by the linear combination of a small number of the so-called Kernel functions that are independent of the original problem size. Essentially we seek to approximate a high-dimension space where the nonlinear function f_i and g_j reside with a much lower dimension space. This kind of approximation has been well-studied in the Machine Learning literatures and the so-called Kernel Method has been shown to be effective [17].

The gist of the Kernel Method is the following. Consider representing a scalar function $f(\bar{x}) : R^n \to R$ of multi-variable argument. An interesting class of approximations are of the form

$$f(\bar{x}) = \sum_{k=1}^{N_k}\alpha_k K(\bar{x}, \bar{x}_k) \tag{42}$$

where $\bar{x} \in R^n$ is the evaluation point, $K(\bar{x},\bar{y}) : R^n \times R^n \to R$ is the Kernel, and the N_k vectors $\bar{x_k} \in R^n$ are denoted as the "support vectors". The basic kernel methods [17] use simple function forms for the kernels and pick the coefficient α_k based on certain loss functions.

In this paper, we choose the kernel form to be the same as the nonlinear functions to be approximated. So the kernel regression form is

$$f_i(\bar{\sigma}) \approx \sum_{k=1}^{N_k} \alpha_k^{(i)} K_1(\xi_k), \quad g_j(\bar{\sigma}) \approx \sum_{k=1}^{N_k} \beta_k^{(j)} K_2(\xi_k), \qquad (43)$$

where

$$K_1(\xi_k) = \frac{\xi_k^2}{1+\xi_k}, \quad K_2(\xi_k) = \frac{1}{1+\xi_k}, \qquad (44)$$

and $\xi_k = < \bar{d_k}, \bar{\sigma} >$, N_k is the number of kernels necessary to achieve the desired accuracy. As shown in section 8, 10 to 15 kernels can achieve 3 to 4 digits of accuracy. The coefficients $\alpha_k^{(i)}$ and $\beta_k^{(j)}$ are obtained by the simple Least Square Fitting, the simplest loss function form [17]. The support vectors d_k are selected from all possible values of $\bar{d}_{13}^{(i)}$ and $\bar{d}_{21}^{(i)}$ in (40) by the K-mean Clustering Methods [17].

It should be noted that due to the locality of the parametric mesh, each mesh node is influenced by a small number of parameters, typically 2 in the 2D cases and 4 in the 3D cases. This means that the length of the vector $\bar{d}_{13}^{(i)}$ and $\bar{d}_{21}^{(i)}$ is 2 for 2D cases and 4 for 3D cases, respectively. Essentially the number of clusters is independent of the number of the parameters in the mask structures. This is a key benefit from using the parametric mesh in section 5. Covering a relatively low-dimensional space with the K-mean clustering usually means small cluster number N_k in (43). Hence the mask model size is largely determined by the number of sampling solves in (5).

Substituting (43) into (38), we obtain the final parametric form of the stiff matrix

$$S(\bar{\sigma}) \approx S_0 + \sum_{k=1}^{N_p} \sigma_k S_1^{(k)} + \sum_{k=1}^{N_k} K_1(\xi_k) S_2^{(k)} + \sum_{k=1}^{N_k} K_2(\xi_k) S_3^{(k)} \qquad (45)$$

where $S_2^{(k)}$ and $S_3^{(k)}$ are the results of the stamping $\alpha_k^{(i)} S_{N_p+1}^e$ and $\beta_k^{(j)} S_{N_p+2}^e$, respectively.

7. PARAMETRIC MASK MODEL

Now we are ready to put everything together to present the final mask model. Substituting (45) and (41) into (3), we obtain

$$\left[A_0 + \sum_{k=1}^{N_p} \sigma_k A_1^{(k)} + \sum_{k=1}^{N_k} K_1(\xi_k) A_2^{(k)} + \sum_{k=1}^{N_k} K_2(\xi_k) A_3^{(k)} \right] \bar{u} = \bar{r}, \quad (46)$$

where $A_0 = S_0 - M_0 - B$, $A_1^{(k)} = S_1^{(k)} - M_1^{(k)}$, $A_2^{(k)} = S_2^{(k)}$, and $A_3^{(k)} = S_3^{(k)}$. Using the congruence projection in (6), we obtain the reduced model

$$\left[\hat{A}_0 + \sum_{k=1}^{N_p} \sigma_k \hat{A}_1^{(k)} + \sum_{k=1}^{N_k} K_1(\xi_k) \hat{A}_2^{(k)} + \sum_{k=1}^{N_k} K_2(\xi_k) \hat{A}_3^{(k)} \right] \hat{u} = \hat{r}. \quad (47)$$

The usage of this mask model is similar to that described in section 4. But there is a key difference. In lithography simulation flow, we only care about the near field at the bottom of the mask. In view of (8), this can be accomplished by using

$$\bar{u_n} = \Lambda \bar{u} = \Lambda P \hat{u} = P_n \hat{u}, \qquad (48)$$

where $\bar{u_n}$ is the portion of $\bar{u}$ that corresponds to the near field at the bottom of the mask, and the sparse matrix Λ selects those entries from $\bar{u}$. An important benefit from (48) is that we only need to store $P_n = \Lambda P$ in the mask model. Matrix P_n is much smaller than matrix P. The algorithmic details for the offline and the online stage are shown in Algorithm 1 and Algorithm 2, respectively. We want to emphasize again that the cost of online stage is $O(N_s^3 + (N_p + N_k)N_s^2)$. This is clearly independent of the orginal problem size N.

Algorithm 1: Offline Stage to generate mask model
Input: Range of parameters in $\bar{\sigma}$; mesh; N_s: number of samplings; tol: truncation tolerance for rank revealing QR
Output: $\hat{A}_0, \hat{A}_1^{(k)}, \hat{A}_2^{(k)}, \hat{A}_3^{(k)}; \hat{r}; P_n$

(1) Form matrices $A_0, A_1^{(k)}, A_2^{(i)}, A_3^{(j)}$ and $\bar{r}$ in (46)
(2) **foreach** $k = 1 : N_s$
(3) Randomly sample $\bar{\sigma}^{(k)}$ using the given ranges
(4) Solve (46) for $\bar{u}^{(k)}$
(5) $P = [P \;\; \bar{u}^{(k)}]$
(6) Run rank-revealing incremental QR to obtain rank r and the full-rank columns $P_1, P_2, ..., P_r$ using the given truncation tolerance tol
(7) **if** $r < k$, i.e., P is rank deficient
(8) $P = [P_1, P_2, ..., P_r]$
(9) Exit the sampling loop
(10) Use P as projector and compute $\hat{A}_0, \hat{A}_1^{(k)}, \hat{A}_2^{(i)}, \hat{A}_3^{(j)}, \hat{r}$ as shown in (6)
(11) $P_n = \Lambda P$ as shown in (48)

Algorithm 2: Online Stage to evaluate mask model
Input: $\bar{\sigma}*; \hat{A}_0, \hat{A}_1^{(k)}, \hat{A}_2^{(k)}, \hat{A}_3^{(k)}; \hat{r}; P_n$
Output: $\bar{u_n}$: the near field at the bottom of the mask

(1) Instantiate the compact mask model in (47) using $\bar{\sigma}*$
(2) Solve (47) for $\hat{u}$
(3) $\bar{u_n} = P_n \hat{u}$ as shown in (48)

8. EXPERIMENTAL EVIDENCE

We use the mask in Figure 1 to validate our ideas. For the $32nm$ node, the nominal values of w_1, w_2 and s are assumed to be $128nm$. They have identically independent distribution with the variance being 10%. This 20% variation range is probably more than sufficient for practical consideration. It should be noted that if we add correlation among these parameters, it will only change the sampling but not the main steps in algorithm 1 and 2. With a randomly generated parameter set $\bar{\sigma}*$, we substitute (38) and (41) into (3) and solve (3) for $\bar{u_1}$. This is treated as the accurate solution. This way, we can clearly see the error introduced separately by the Kernel-based fitting in (43) and the model order reduction in Algorithm 1.

A) Kernel-based Regression Accuracy With the same parameter set $\bar{\sigma}*$ mentioned above, we solve (46) for $\bar{u_2}$ and then compute the relative L_2 norm error $\frac{\|\bar{u_1} - \bar{u_2}\|_2}{\|u_1\|_2}$. The relative error vs. the number of kernels is shown in Fig. 4.

B) Model Order Reduction Accuracy In this experiment, $N_k = 20$ kernels are used in (47). The corresponding small error in kernel fitting ensures that the MoR error in (47) dominates the overall error, as is clearly seen in Figure 5. With the increase of the number

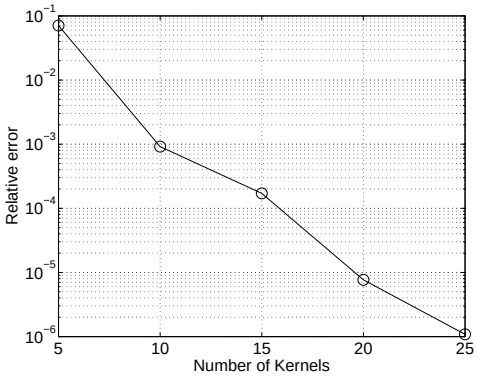

Figure 4: Relative error in regression.

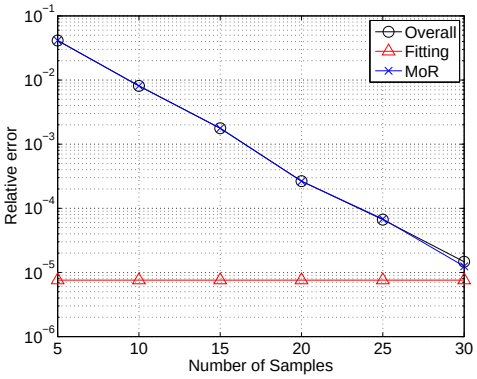

Figure 5: Convergence of the L2 norm relative error in field distribution for the mask shown in Figure 1. Each parameter varies between -10% and $+10\%$ from its nominal value of $128nm$. "MoR" refers to the error caused by the model reduction stage. "Fit" refers to the error caused by the regression fit in (43). "Overall" refers to the overall error due to both.

of samples N_s in Algorithm 1, the relative error due to model order reduction decreases rapidly in Figure 5.

C) Mask Model Accuracy Figure 5 also shows the overall error in the approximate solution provided by the new mask model. With about 8 kernels and 10 samples, the mask model can achieve a 1% overall relative error or 2-digits of accuracy.

D) Mask Model Speed The main cost of using the reduced mask model is to instantiate the small dense matrices in (47) and then invert one small dense matrix. Instantiating a few 10×10 dense matrices and then inverting a 10×10 dense matrix takes about $1e-4$ second. For the simple two-dimensional phase-shift mask shown in Fig. 1, direct use of FEM approach would result in a sparse matrix $A(\overline{\sigma})$ in (46) with N being around 5000. The CPU time used by a direct sparse solver for such a system would be in the order of a few seconds on a desktop PC. So we see a 4-order of magnitude improvement in speed.

In addition, since the CPU time at the online stage directly relates to the number of samples in Algorithm 1, the smooth trade-off between the accuracy and the CPU time is clearly demonstrated in Figure 5. Further more, our empirical studies indicate that number of samples is a weak function of the original problem size or the number of parameters. This is certainly a very attractive feature of the new mask model proposed in this paper.

9. CONCLUSIONS

In this paper, we propose two key new ideas to improve our work on the parametric mask model in [8]: 1) Parametric unstructured mesh using the distance function and the blending function; 2) Kernel-based regression to significantly reduce the number of parametric terms in the final system. Numerical experiments demonstrate that the new mask model offers smooth accuracy-speed trade-off and hence can be tuned to different design applications. The model generation in Algorithm 1 is inherently incremental and both theoretical and practical ways to estimate the model error are readily available. In addition, the parametric mesh allows the decoupling between the mesh generation and the parametrization of the FEM matrices, a considerable simplification from implementation point of view.

10. ACKNOWLEDGMENTS

We want to thank Dr. Joel Phillips from Cadence Research Labs for bringing the Kernel Methods in [14, 17] to our attention. We want to thank Prof. Per-olof Persson from University of California at Berkeley for bringing to our attention the use of blending function in [16]. And finally, we want to thank Dr. Apo Sezginer from Cadence Design Systems for the valuable discussions.

11. REFERENCES

[1] A.K.T. Wong, *Resolution Enhancement Techniques in Optical Lithography*, SPIE, Bellingham, WA, 2001.

[2] A. K. Wong and A.R. Neureuther, "Rigorous three-dimensional time-domain finite-difference electromagnetic simulation for photolithograph applications," in *IEEE Trans. on Semiconductor Manufacturing*, Nov. 1995, vol. 8, p. 419.

[3] S. Burger, R. Kohle, L. Zschiedrich, H. Nguyen, F. Schmidt, R. Marz, and C. Nolscher, "Rigorous simulation of 3D masks," in *Proc. SPIE*, 2006, vol. 6349.

[4] J.W. Goodman, *Introduction to Fourier Optics*, Roberts and Companies, Greenwood Village, CO, 2005, third edition.

[5] Jaione Tirapu-Azpiroz; Paul Burchard; Eli Yablonovitch, "Boundary layer model to account for thick mask effects in photolithography," in *Advanced Lithography, Proc. SPIE*, Feb. 2003, p. 1611.

[6] K. Adam and A.R. Neureuther, "Domain decomposition methods for the rapid electromagnetic simulation of photomask scattering," in *J. Microlitho., Microfab., and Microsys.*, Oct. 2002, vol. 14, p. 253.

[7] V. Singh, B. Hu, K. Toh, S. Bollepalli, S. Wagner, and Y. Borodovsky, "Making a tillion pixels dance," *Advanced Lithography, Proc. SPIE*, vol. 6924, Feb. 2008.

[8] Zhenhai Zhu and Frank Schmidt, "An efficient and robust mask model for lithography simulation," in *Advanced Lithography, Proc. SPIE*, Feb. 2008, vol. 6925, p. 126.

[9] G. Rozza, D.B.P. Huynh, and A. Patera, "Reduced basis approximation and a posteriori error estimation for affinely parameterized elliptic coercive partial differential equations," *Arch. Comput. Methods Eng.*, vol. 15, pp. 229–275, 2008.

[10] J.R. Phillips, "Variational interconnect analysis via PMTBR," in *International Conference on CAD*, San Jose, CA, Nov. 2004, p. 872.

[11] J.M. Jin, *The Finite Element Method in Electromagnetics*, John Willeys and Sons Inc, New York, 2002, second edition.

[12] F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*, Springer, 1998.

[13] J. Phillips, "Projection-based approaches for model reduction of weakly nonlinear, time-varying systems," *IEEE Trans. on CAD*, vol. 22, pp. 171, Feb. 2003.

[14] J.R. Phillips, J. Afonso, A. Oliveira, and L.M. Silveira, "Analog macromodeling using kernel methods," in *International Conference on CAD*, San Jose, CA, Nov. 2003, p. 446.

[15] J.A. Sethian, *Level Set methods and Fast Marching Methods*, Cambridge University Press, 1999.

[16] P. Persson, J. Peraire, and J. Bonet, "Discontinuous galerkin solution of the navier-stokes equations on deformable domains," in *Proc. 45th AIAA Aerospace Sciences Meeting and Exhibit*, Jan. 2007.

[17] T. Hastie, R. Tibshirani, and J.H. Friedman, *Elements of Statistical Learning*, Springer, 2003.