Japanese Journal of Applied Physics **52** (2013) 055202 **REGULAR PAPER**
http://dx.doi.org/**10.7567/JJAP.52.055202**

Solution-Refined Method for Electrostatic Potential Distribution of Large-Scale Electron Optics

Yen-Min Lee[1], Jia-Han Li[1]*, Tony Wen-Hann Sheu[1], Kuen-Yu Tsai[2], and Jia-Yush Yen[3]

[1]*National Taiwan University, Department of Engineering Science and Ocean Engineering, Taipei 10617, Taiwan*
[2]*National Taiwan University, Department of Electrical Engineering, Taipei 10617, Taiwan*
[3]*National Taiwan University, Department of Mechanical Engineering, Taipei 10617, Taiwan*
E-mail: jiahan@ntu.edu.tw

Received March 26, 2012; revised February 6, 2013; accepted March 3, 2013; published online May 2, 2013

The solution-refined method is developed to solve electrostatic fields of the electron-beam direct-write lithography system. The prediction of accurate electron trajectories and the geometry of the developed photoresist patterns rely on high-resolution electrostatic fields in the whole system. Considering fabrication errors, such electrostatic fields cannot be solved using a cylindrical symmetry. Thus, this problem is a multiscale problem that requires a huge computer memory to solve. In our cases, the minimum number of grids of 1 nm length are applied and the total memory required approaches 75 Gbyte. Since the proposed solution-refined technique has a tradeoff with computational time, fewer central processing units (CPUs) are needed to solve this system because each CPU that solves the problem exceeds its available storage memory. The proposed technique can be used to solve the electron-beam direct-write lithography system at higher resolution and the problems exceed the available storage memory.

1. Introduction

Electron-beam direct-write lithography is a maskless writing technique of high resolution and is a potential candidate next-generation lithography for creating patterns under 22 nm and beyond.[1)] This type of lithography system uses controlled electrons to strike a photoresist layer to form a pattern directly within a vacuum chamber. As electron guns emit a large number of electrons, the pathways of these electrons within the vacuum chamber are controlled by electron lenses. Thus, it is necessary to solve Laplace's equation and calculate the electrostatic field for electron trajectory calculations. In general, Laplace's equation can be solved analytically by the superposition of harmonic functions. However, the problem in electron-beam direct-write lithography applications is too complex to find an analytical solution. Thus, we need to use numerical methods to find approximate solutions. Many numerical methods have been used to solve Laplace's equation, such as the finite-difference method (FDM),[2)] finite-element method (FEM),[3)] boundary element method (BEM),[4,5)] and charge density method (CDM).[6)] Also, some available packages based on FDM, FEM, and BEM for solving Laplace's equation have also been developed to investigate electrostatic problems, for example, SIMION,[7)] Poisson Superfish,[8)] and LORENTS 2D/RS.[9)] Comparisons of computational time as well as memory usage between the FDM, FEM, and BEM for solving electrostatic charged particle optics can be found in Ref. 10. The results show that each of these methods has particular advantages and disadvantages. It was concluded that the memory usage, computational speed, and computational accuracy of the FDM are dominated mainly by the number of mesh points. Thus, for complex and large-scale charge optic problems, it is easier to predict the required computational resources using FDM.

For charged particle optic simulators, Munro and coworkers have published a series of research results, including aberration predictions,[11,12)] the design of a photomultiplier,[13)] an overview of computational modeling techniques in charged particle optics,[14)] and the beam blur of the SCALPEL-HT/Alpha electron beam lithography tool.[15,16)] They used FDM and second-order FEM to obtain numerical results under an electrostatic field. However, more accurate results have to be obtained by fitting a Fourier-Bessel series. Consequently, an incredibly large computational memory and a long time are needed if the system becomes larger. Thus, the search for a method of solving the entire electron optic system with a high resolution still continues. Sawada and Yoneda adopted a modified axis symmetric FEM-BEM mixed technique to calculate the electrostatic potentials for the electron-beam lithography system.[17)] The electrostatic potentials are calculated using BEM at FEM interpolated grid nodes. As shown in their results, the central processing unit (CPU) time can be less than that used in pure BEM calculation and memory can be saved because only accurate electrostatic potentials are obtained in some regions far from the boundary. In addition, another reason for saving storage memory is that a modified axis symmetric assumption is used. Lencová applied fine mesh points under the FEM scheme to calculate the electrostatic field for cylindrically symmetric electron lenses.[18–20)] In this case, the problem can be treated as a two-dimensional (2D) problem. Therefore, both memory usage and computational time can be conserved in such a symmetric situation. However, it was reported that a 2D analysis of the electron beam optics can be applied when the system is assumed to be perfect with cylindrical symmetry, whereas a three-dimensional (3D) computation is unavoidable.[14)] Actually, when fabricating electron-beam direct-write lithography chambers, asymmetric matching of individual electrode pieces and misalignments between assembled elements may occur because of fabrication limits. The investigation of fabrication error budgets that affect electron trajectories was carried out by Chen et al.[21)] In their research, the software LORENTS 2D/RS[9)] is employed and the analyses are based on 2D electrostatic fields. An accurate 3D field computation for electron-beam direct-write lithography chambers is a multiscale problem that requires a huge computer memory to achieve sufficient resolution. In such circumstance, workstations or computer clusters equipped with multiple processors and multiple storage memories are often used to decompose a large problem into several subdomains for

multiprocessing computations. Generally, the open multiprocessing (OpenMP),[22] message passing interface (MPI),[23] hybrid MPI/OpenMP schemes,[24,25] and parallelized graphics processing unit (GPU)[26,27] methods are used as interfaces for multiprocessing computation. The electron optics software SIMION[7] can be used under multiple processors. However, the maximum point numbers allowed is limited at 20 billion (i.e., up to 190 Gbyte in their simulation but only 24 Gbyte in our solution-refined method) when the computer is equipped with sufficient memory. There are many examples that describe FDM combined with a multiprocessing computation scheme to have remarkable computing advantages over conventional single-processing methods.[28,29] Also, the use of parallelized GPUs, which have many processors on one graphic card, is combined with other multiprocessing computation methods to solve the problems.[30] Research employing the FDM combined with NVIDIA's GPU scheme was also reported,[26] which showed that the computational speed of the GPU (GeForce 7800 GS) can be 30 times greater than that of the two-core CPU (Opteron 890), but that the largest possible computational domain is 1615 times smaller. Antoniou et al. utilized a multicore GPU combined with MPI and OpenMP under the FDM scheme.[27] Their results show good computation acceleration, but the computational domain is limited because each processing unit has finite storage memory.

The computational ability of multiprocessing schemes is limited by the number of parallelized processors used because each parallelized processor has finite storage memory. Thus, the development of a numerical method with a greater capability to solve larger scale problems is necessary. In this paper, we propose the use of FDM-based solution-refined procedures to solve Laplace's equation for a large-scale problem, and take electrostatic field distributions of an electron-beam direct-write lithography system as our example. To show how the solution-refined method works, procedures for solving 2D problems under different mesh sizes using a single CPU are demonstrated in Sect. 2. Then, we solve 3D electrostatic field distributions of an electron-beam direct-write lithography system using the solution-refined method combined with the MPI scheme in Sect. 3. Furthermore, we also calculated the z-directional shifts and tilt situations of the 3D electrostatic fields of the lens, which cannot be solved using the cylindrical symmetry.

2. Solution Refinement under Single CPU

Taking the divergence of the electrostatic field, we can obtain Poisson's equation which is related to the charge density ρ and the electrostatic potential as[31]

$$\nabla^2\Phi = -\frac{\rho}{\varepsilon_0}, \tag{1}$$

where ε_0 stands for the vacuum permittivity. In an electron-beam lithography system, which is a high-vacuum system, we assume $\rho = 0$ for simplicity. Thus, Poisson's equation is reduced to Laplace's equation. Here, a 2D problem for Laplace's equation is taken as an example and is shown in Fig. 1. The boundary conditions of this problem are set as

$$\begin{aligned} &\Phi(0,y) = 0, \quad \Phi(x,0) = 0, \\ &\Phi(a,y) = 0, \quad \Phi(x,b) = 1, \end{aligned} \tag{2}$$

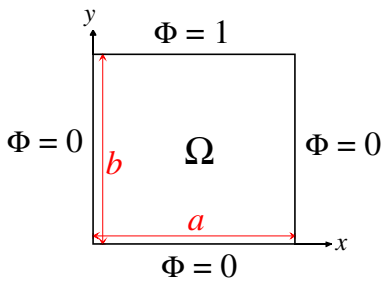

Fig. 1. (Color online) Boundaries and dimensions of 2D problem.

where a and b are the lengths along the x- and y-directions, respectively. The analytic solution of this problem is

$$\Phi(x,y) = \sum_{n-1}^{\infty}\left[B_n \sinh\left(\frac{n\pi y}{a}\right)\right]\sin\left(\frac{n\pi x}{a}\right), \tag{3}$$

where

$$B_n = \frac{2}{a}\int_{x=0}^{a}\frac{\sin\left(\frac{n\pi x}{a}\right)}{\sinh\left(\frac{n\pi b}{a}\right)}dx.$$

The memory needed in the finite-difference computation can be estimated using the equation[32]

$$N_i^{\text{total}} \times N_j^{\text{total}} \times N_k^{\text{total}} \times n_e \times 8 \text{ bytes}, \tag{4}$$

where N_i^{total}, N_j^{total}, and N_k^{total} are the number of grids along the x-, y-, and z-directions, respectively, and n_e stands for the number of storage variables.

Referring to Fig. 2, our solution-refined method entails the following steps:

(a) Creating coarse grids in the Ω domain, and solving Laplace's equation for $N_i \times N_j$ meshes in the Ω domain as shown in (a) in Fig. 2, where N_i and N_j are the coarse grid numbers along the x- and y-directions, respectively. After the coarse mesh computation, release the stored memory (i.e., $N_i \times N_j \times n_e \times 8$ bytes) except all values at $y = N_j/2$.

(b) Applying subgrids to the $y = N_j/2$ line. Solve Laplace's equation in the upper-half region of Ω that has $(\Gamma^2/2) \times N_i \times N_j$ mesh points as shown in (b) in Fig. 2, where Γ is the number ratio of subgrids to coarse grids. After finishing the fine grid computation, release the stored memory [i.e., $(\Gamma^2/2) \times N_i \times N_j \times n_e \times 8$ bytes] in the upper-half region except all fine grid values at $y = 3N_j/4$.

(c) Solving Laplace's equation in the lower-half region of Ω with a fine grid boundary at $y = N_j/2$. At this time, the total grid numbers are $(\Gamma^2/2) \times N_i \times N_j$ as shown in (c) in Fig. 2. After the fine grid computation in the lower-half region is finished, release the stored memory [i.e., $(\Gamma^2/2) \times N_i \times N_j \times n_e \times 8$ bytes] in the lower-half region except all fine grid values at $y = N_j/4$.

(d) Solving Laplace's equation in the central domain by grappling the fine grids at $y = 3N_j/4$ and $N_j/4$ from (b) and (c). At this step, the total grid numbers are $(\Gamma^2/2) \times N_i \times N_j$ mesh points as shown in (d) in Fig. 2. Then, release the stored memory [i.e.,

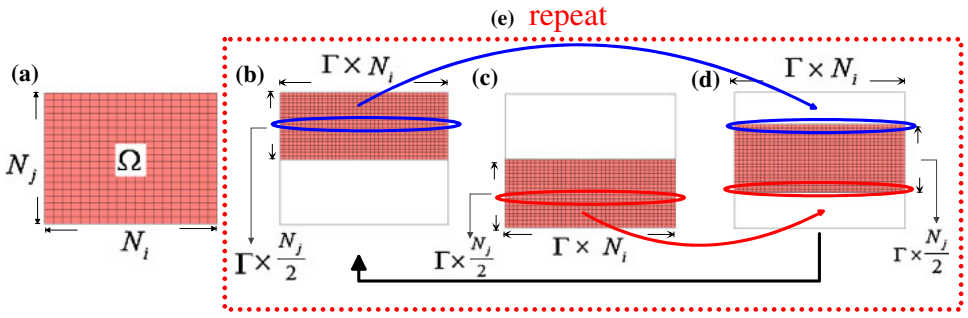

Fig. 2. (Color online) Computational sequence of solution-refined method in 2D problem. The empty regions are not computed.

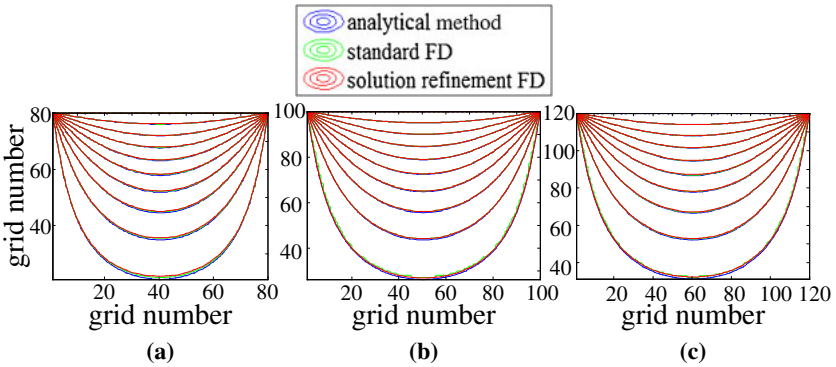

Fig. 3. (Color online) Results of 2D Laplace's equation specified in Fig. 1 for analytical method, standard finite-difference method, and solution-refined finite-difference method under (a) 80 × 80, (b) 100 × 100, and (c) 120 × 120 meshes.

$(\Gamma^2/2) \times N_i \times N_j \times n_e \times 8$ bytes] except all fine grid values at $y = N_j/2$. In addition, these values at $y = N_j/2$ will be used in step (e) for repeated computation.

(e) Repeating the steps from (b) to (d) until the computational results become stable.

Three 2D cases are used to demonstrate the accuracy of the solution-refined method described above and the steps shown in Fig. 2. The grid numbers of these three cases are $N_i \times N_j = 80 \times 80$, $N_i \times N_j = 100 \times 100$, and $N_i \times N_j = 120 \times 120$, respectively. We solve these problems numerically by the standard and solution-refined finite-difference methods. Then, the results obtained using both methods are compared with the analytical results shown in Fig. 3. The L2 norms (least-square error norms) between all the standard finite-difference results and analytical results are shown in Fig. 4. From these figs., the L2 norms between all the fine-grid results and analytical results converge from 2.375×10^{-4} to 1.088×10^{-4} as the total grid number increases from $N_i \times N_j = 80 \times 80$ to $N_i \times N_j = 120 \times 120$. This is reasonable, because more accurate results can be obtained in the fine meshes than those above in the coarse meshes. In addition, the L2 norms of the solution-refined computation approach the L2 norms of the standard finite-difference computation after five refinements.

3. Simulation of 3D Electron-Beam Direct-Write Lithography System

3.1 Introduction to electron-beam direct-write lithography system

Electron-beam direct-write lithography is a potential candidate for the next-generation lithography. This type of lithography system uses controlled electrons to strike resists to form a pattern directly inside a vacuum chamber. As electron guns emit a large number of electrons, the pathways of these electrons within the vacuum chamber are controlled by electron lenses. Generally, electron lenses are metallic structures with circular shapes that allow electrons to pass through their centers. As the desired electric voltages are applied on electron lenses, electrons can be controlled and focused on the designed focal plane. Thus, if an accurate electrostatic field within the whole chamber can be calculated, the electron trajectories and electron focal spot on top of the photoresist surface can then be predicted. The final goal is to calculate electron penetration within a photoresist (i.e., photoresist development). Many research studies related to the calculation of electron penetrations within the resist have been published.[33–35] However, most of them used an ideal focal spot size, which means that the calculated photoresist lithographic results were based on perfect electron ray focusing. Actually, when fabricating an electron-beam direct-write lithography chamber, asymmetric error may occur, which affects the resolution of the focal spot. Thus, it is necessary to compute the 3D electrostatic field to predict and control the trajectories of a moving electron within the entire system.

3.2 Parallel computation with solution refinement

The electron-optical system (EOS) comprises an electron tip, condenser lenses, silicon wafer stages, and other components.[36] These components are designed in the micrometer range.[37] The size of an electron is fairly small and can be neglected in comparison with the size of the whole system. Thus, a huge memory and a large amount of storage are

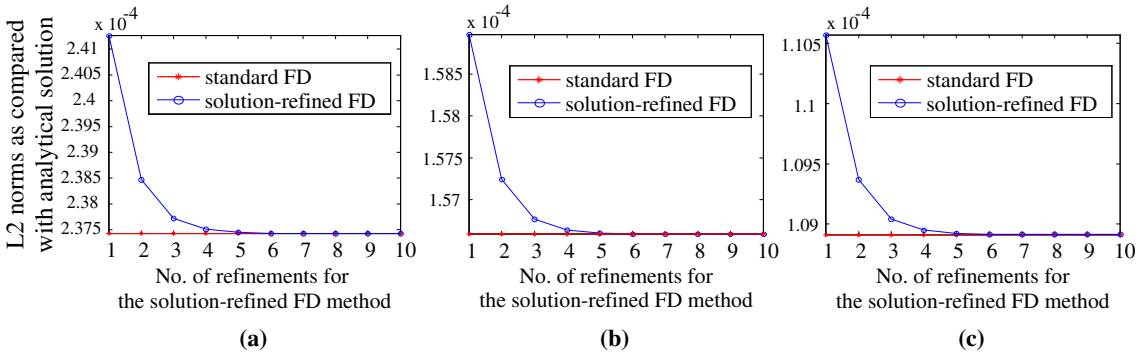

Fig. 4. (Color online) Least-square error norms (L2 norms) of 2D Laplace's equation specified in Fig. 1 between results of analytical method, standard finite-difference method, and solution-refined finite-difference method under (a) 80 × 80, (b) 100 × 100, and (c) 120 × 120 meshes.

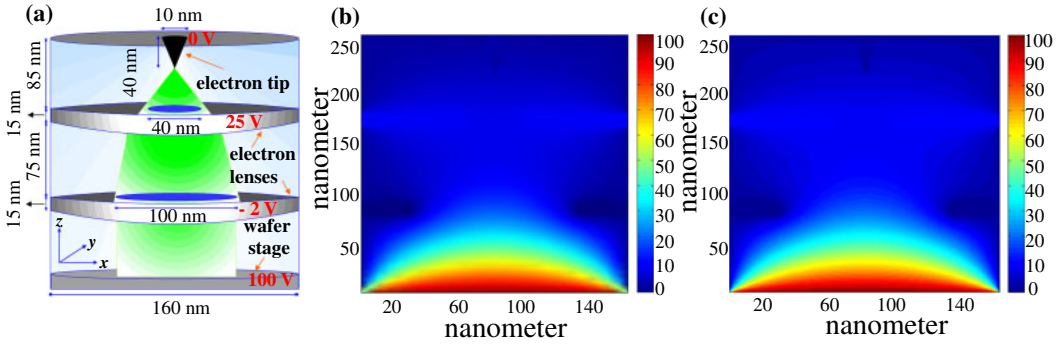

Fig. 5. (Color online) (a) Simulation dimensions of small-scale test, (b) small-scale simulation result from COMSOL Multiphysics software, and (c) small-scale simulation result from in-house FDM program.

required for an accurate simulation of this problem. To validate the FDM FORTRAN program we developed, the shrunken small-scale system shown in Fig. 5(a) is used as a test problem. The dimensions of the system shown in Fig. 5(a) along the x-, y-, and z-axes are 160, 160, and 260 nm, respectively. Furthermore, the electrostatic field of this system is calculated using our FDM code and FEM-based commercial software named COMSOL Multiphysics. In our FDM calculation, 1 nm meshes are applied in the entire computational domain, except in the electron tip-shape region. Some 0.15 nm grids are applied to smooth the electron tip shape. Thus, in this test, the grid number in our FDM code is about 6,656,000, which needs an approximately 50 MB memory and can be executed on a single CPU. The computational results at the center of the y-axis, calculated by the commercial software COMSOL Multiphysics, are shown in Fig. 5(b), and the results calculated by the FDM programs we developed are shown in Fig. 5(c). These results are not easy to compare directly because nonuniform triangular meshes are used in Fig. 5(b), but nonuniform rectangular meshes are used in Fig. 5(c). Nonetheless, good agreement can be seen between Figs. 5(b) and 5(c).

The designed large-scale 3D electron-beam lithography system is shown in Fig. 6. To solve the full 3D electrostatic field, the distributed computation is combined with the solution-refined technique. Initially, we use 20 nm meshes in the entire system under the Cartesian coordinate system. However, the cylindrical geometry of the electronic lens and electron tip cannot be described smoothly in Cartesian coordinates. Hence, the conservative level set method is

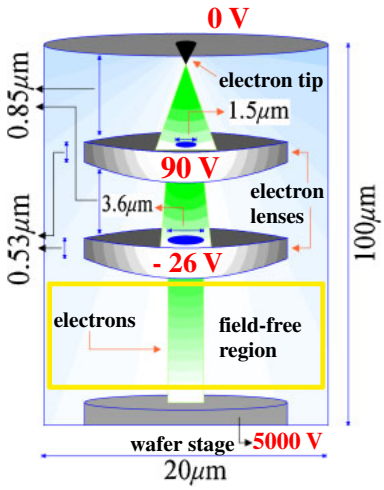

Fig. 6. (Color online) Simulated dimensions of large-scale electron-beam direct-write lithography system.

used to describe the shapes of the cylindrical lens and needlelike tip.[38] The equation listed below is used to create a smooth cylindrical shape in Cartesian coordinates.

$$\Phi = (1 + e^{(d-d_c)/\varepsilon})^{-1}, \quad (5)$$

where d is the number of grid points surrounding the circle center d_c and ε is the parameter for determining the thickness of the circle. In electron beam lithography, the size of the electron tip may differ on a case-by-case basis according to the desired specifications and the fabrication limit. Here,

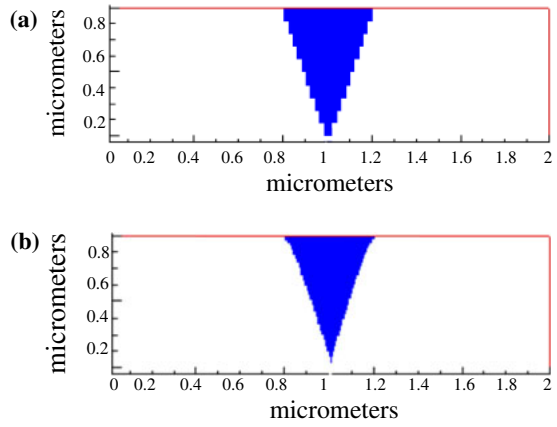

Fig. 7. (Color online) Mesh layout of (a) electron tip with 20 nm grid size (x–z plane at the center of y-axis), and (b) electron tip with 1 nm grid size (x–z plane at the center of y-axis).

we use 20 nm as the radius of the pinpoint, and the length of this tip is 800 nm. The sharp tip shape is hard to describe accurately by utilizing only 20 nm uniform grids. Thus, we apply 1 nm fine subgrids at the edge of the electron tip to smooth its shape. The mesh layout comparisons of the electron tip at the center of the y-axis are shown in Fig. 7. Figure 7(a) shows the mesh layout of the electron tip using uniform 20 nm grids in the whole tip, and Fig. 7(b) shows that in the case of 20 nm grids at the tip with 1 nm grids at its edge. It is clear that the 20 nm grid size cannot provide sufficient resolution at the edge of the electron tip. In the case of Fig. 7(b), the storage memory occupied by 1 nm grids can be neglected because there is only a small number of 1 nm grids. Therefore, the total grid number for the full 3D computation is roughly equal to the number of 20 nm meshes and can be estimated as $N_i \times N_j \times N_k = 1000 \times 1000 \times 5000$ grids. The iteration method is applied in our FDM FORTRAN program. Thus, the two variables $\Phi^{\text{fine}}(N_i, N_j, N_k)$ and $\Phi^{\text{fine}}_{\text{old}}(N_i, N_j, N_k)$ are stored during the iteration. $\Phi^{\text{fine}}(N_i, N_j, N_k)$ is the updated voltage during the finite-difference iteration in the fine grids and $\Phi^{\text{fine}}_{\text{old}}(N_i, N_j, N_k)$ is the voltage that has not been updated in the fine grids. The total memory required for a full 3D electron-beam lithography simulation is about 74.5 Gbyte. We distributed the total domain via 10 CPUs along the z-direction by one-dimensional data division. The MPI library is used for the distributed computation, and the 3D FDM parallelization programming with solution refinement processes is shown in Fig. 8. All simulations were performed in the IBM 1350 cluster at the National Center for High Performance Computing of Taiwan. The specifications of IBM 1350 are Intel X5450 3.0 GHz Quad core processors with 16 Gbyte PC2-5300 667 MHz FBD 240-pin ECC DDR2-SDRAM. The computation starts from Fig. 8(a), the coarse grids are used in the whole 3D domain as $N_i \times N_j \times N_k = 200 \times 200 \times 5000$ grids, and two variables, $\Phi^{\text{coarse}}(N_i, N_j, N_k)$ and $\Phi^{\text{coarse}}_{\text{old}}(N_i, N_j, N_k)$, are used for the finite-difference iterative method, where $\Phi^{\text{coarse}}(N_i, N_j, N_k)$ is the updated voltage during the finite-difference iteration with coarse grids, and $\Phi^{\text{coarse}}_{\text{old}}(N_i, N_j, N_k)$ is the voltage that has not been updated in the coarse grids. At this time, the total memory required is about 3 Gbyte. The required storage memories are sufficient for one CPU to calculate. In the second step, we propose to compute the whole domain using 10 CPUs. Thus, we generate nine fine-mesh interfaces that contain $N_i \times N_j = 1000 \times 1000$ grids in each interface, and divide the total number of domains into 10 subdomains along the z-axis, as shown in Fig. 8(b). Then, we distribute these subdomains into 10 CPUs for computation, and each division has $N_i \times N_j \times N_k = 1000 \times 1000 \times 500$ grids with the two variables, $\Phi^{\text{fine}}(N_i, N_j, N_k)$ and $\Phi^{\text{fine}}_{\text{old}}(N_i, N_j, N_k)$, as shown in Fig. 8(c). Therefore, each division contributes an approximately 7.45 Gbyte memory. When the computations in all CPUs are finished, as shown in Fig. 8(c), the solution refinement technique is applied. However, the boundary values in the n-th CPU should be passed onto the $(n-1)$-th CPU before the solution refinement. This step is shown in Fig. 8(e). In addition, the positions of the passed boundaries are decided by the programmer. Finally, repeat the steps from Figs. 8(c) and 8(d) until the electrostatic field does not vary very much.

However, the computational domain for each CPU in Figs. 8(c) and 8(d), which contributes approximately 7.45 Gbyte, is still overloaded for a single CPU. Thus, the computational domain is also decomposed during these two processes in each CPU. The explanations for the domain decomposition in each CPU are shown in Fig. 9. We take CPU 1 and CPU 2 as examples in Fig. 9. In the first step, the

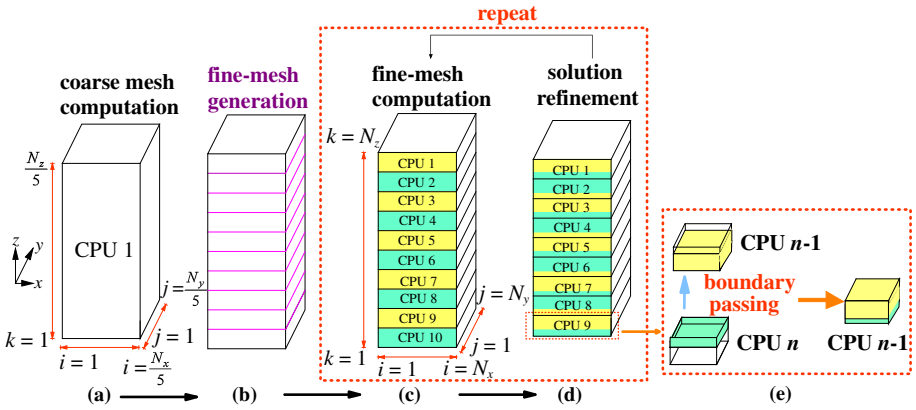

Fig. 8. (Color online) MPI-based parallel computing: (a) coarse mesh computation by 1 CPU; (b) fine-mesh generation; (c) fine-mesh parallel computation using 10 CPU; (d) solution refinement; (e) boundary passing before solution refinement.

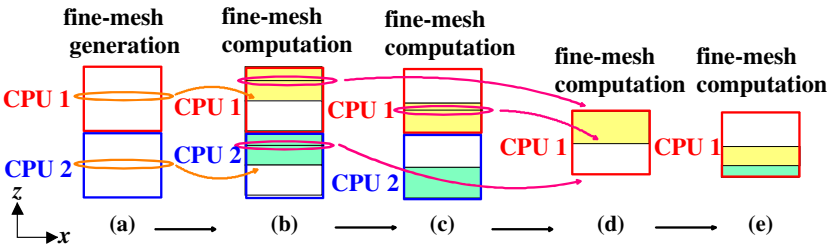

Fig. 9. (Color online) Computational procedures used for each CPU. The empty regions are not computed.

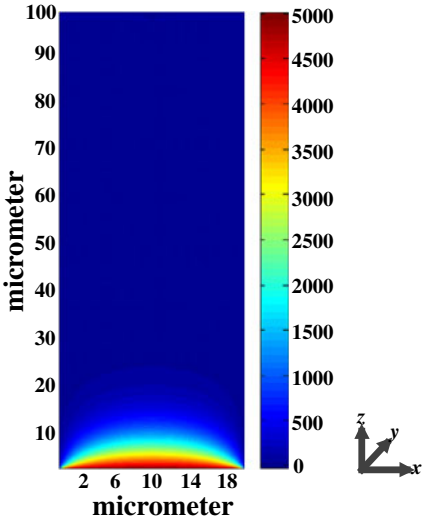

Fig. 10. (Color online) Simulated electrical potential distribution at center of y-axis.

Table I. Simulation conditions of 3D electron-beam lithography system.

Computational domain (μm)	20 × 20 × 100
Grid size (nm)	20
Number of CPUs	10
Total memory required (Gbyte)	73

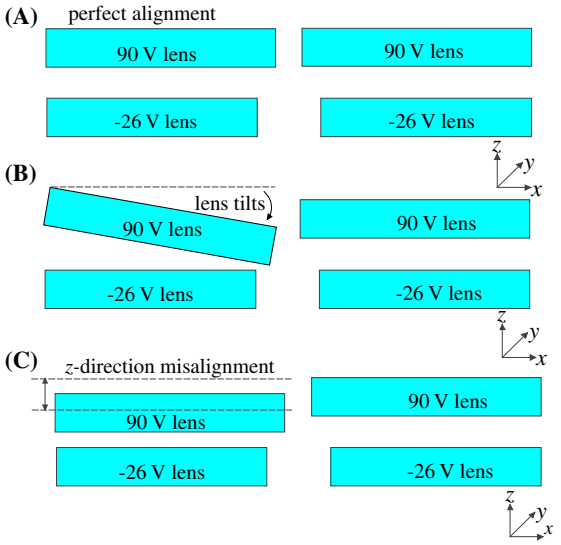

Fig. 11. (Color online) Alignment situation of electron lens: (A) perfect alignment, (B) left-half 90 V lens tilted downward, and (C) left-half 90 V lens shifted along z-direction.

solutions in the coarse meshes are computed in each domain in CPU 1 and CPU 2. Then, the fine meshes at the center of the z-axis in each domain are generated according to the solutions in the coarse meshes, as shown in Fig. 9(a). Therefore, Laplace's equation under $1000 \times 1000 \times 250$ meshes with the two unknowns, $\Phi^{\text{fine}}(N_i, N_j, N_k)$ and $\Phi^{\text{fine}}_{\text{old}}(N_i, N_j, N_k)$, for each upper-half domain can be solved in parallel using the two CPUs. This step contributes about 3.725 Gbyte in each CPU to finish the computation, as shown in Fig. 9(b). Figure 9(c) shows that Laplace's equation is solved in each lower-half domain under the same situation mentioned in Fig. 9(b). In Fig. 9(d), the top, central, and bottom fine-mesh boundaries in CPU 1 are obtained from previous procedures [i.e., Figs. 9(b) and 9(c)]. Thus, Laplace's equation can be solved using CPU 1 for the upper-half and lower-half domains with 3.725 Gbyte in Figs. 9(d) and 9(e).

The simulated voltage distribution (x–z plane) at the center of the y-axis is shown in Fig. 10. Owing to the high voltage at the wafer stage, it is difficult to observe either the voltage distribution or the electrostatic field distribution in the lens region.

The simulation conditions for Fig. 10 are listed in Table I. The available memory of each CPU is 4 Gbyte. Thus, under 20 nm resolution, at least 19 CPUs must be used to solve this problem by the standard finite-difference method with a parallel computation scheme. However, we used only 10 CPUs to solve this problem.

Imperfect fabrications occur frequently in micro-electro-mechanical systems (MEMSs)/nano-electro-mechanical systems (NEMSs), and the case for the 90 V lens misalignment, which cannot be solved using cylindrical symmetry, is simulated by our proposed method. We consider the z-direction shift and tilt situations of the electron lens in our calculation, as shown in Fig. 11. For the z-direction shift calculations, we assume that the left-half 90 V lens has 250 and 500 nm downward shifts. For the lens tilt calculations, we assume that the left-half 90 V lens has 2.5 and 5° downward tilts. The calculated voltage distributions (x–z plane, at the center of the y-axis) for the electron tip and two-well-aligned-lens region are shown in Fig. 12. Figure 12(A) shows the voltage distribution of a well-aligned system. Also, Figs. 12(B1) and 12(B2) show the voltage distribution of the lens systems in which the left-half 90 V lens has 250 and 500 nm shifts along the z-direction, respectively. Furthermore, Figs. 12(C1) and 12(C2) show the voltage distribution of the lens systems in which the left-half 90 V lens has 2.5 and 5° downward tilts, respectively. As show in Fig. 12(A), the calculated results are symmetric owing to

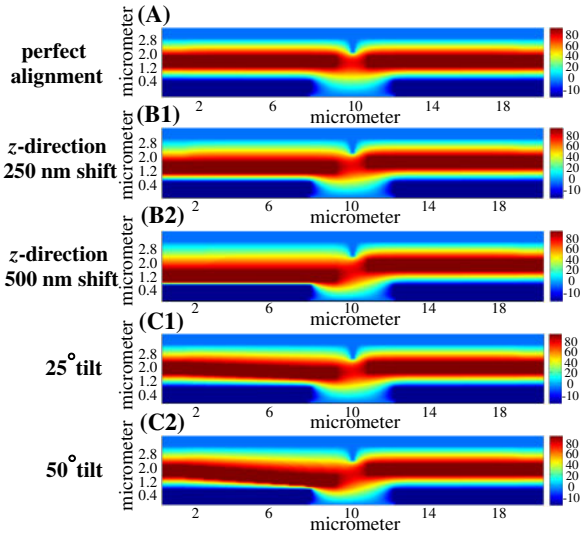

Fig. 12. (Color online) Calculated voltage distributions of lens region under different fabrication errors (x–z plane, at center of y-axis). (A) Perfect alignment. (B1) Left-half 90 V lens has a 250 nm z-direction misalignment. (B2) Left-half 90 V lens has a 500 nm z-direction misalignment. (C1) Left-half 90 V lens has a 2.5° downward tilt. (C2) Left-half 90 V lens has a 5° downward tilt.

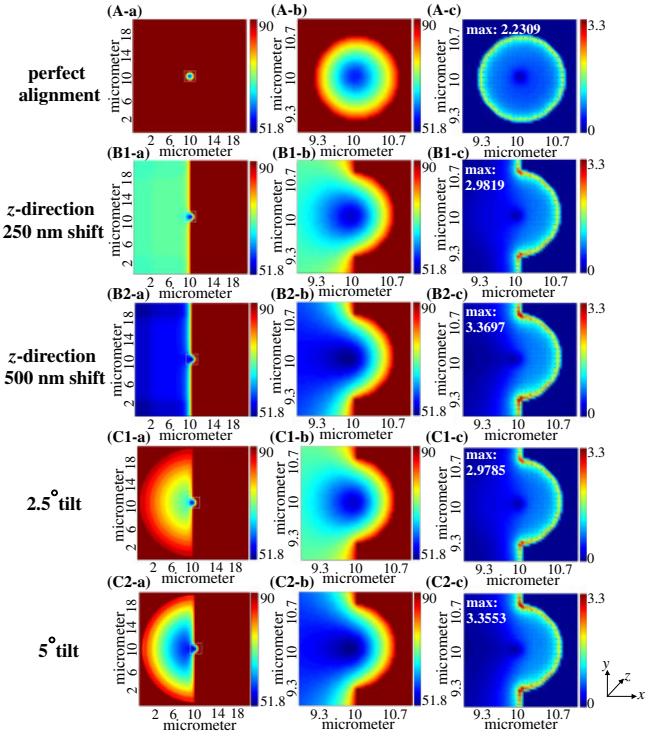

Fig. 13. (Color online) Calculated voltage distributions at center of right-half 90 V lens [subplots (A-a), (B1-a), (B2-a), (C1-a), and (C2-a)], magnified plots [subplots (A-b), (B1-b), (B2-b), (C1-b), and (C2-b)], and electrostatic fields of magnified plots [subplots (A-c), (B1-c), (B2-c), (C1-c), and (C2-c)]. (A-a), (A-b), and (A-c) show the plots for a perfectly aligned case. (B1-a), (B1-b), and (B1-c) show the plots in the case in which the 90 V lens has a 250 nm z-direction misalignment. (B2-a), (B2-b), and (B2-c) show the plots in the case in which the 90 V lens has a 500 nm z-direction misalignment. (C1-a), (C1-b), and (C1-c) show the plots in the case in which the left-half 90 V lens has a 2.5° downward tilt. (C2-a), (C2-b), and (C2-c) show the plots in the case in which the left-half 90 V lens has a 5° downward tilt.

the perfect alignment. Thus, this type of situation can be calculated by applying cylindrical symmetry. Thus, computational memory can be reduced. However, fully 3D computations are needed for the misaligned systems, as shown in Figs. 12(B1)–12(C2). As seen from these results, nonsymmetric voltage distributions can be found in Figs. 12(B1)–12(C2) owing to the lens eccentricity.

The calculated voltage distributions and electrostatic field distributions on top of the right-half 90 V lens (x–y plane) are shown in Fig. 13. Figures 13(A-a), 13(B1-a), 13(B2-a), 13(C1-a), and 13(C2-a) are the voltage distributions on top of the right-half 90 V lens. The voltages of the left-half regions drop to about 70 and 62 V in Figs. 13(B1-a) and 13(B2-a), respectively, because of the z-direction shift of the left-half electron lens. Also, the voltages of the left-half regions linearly decrease inward in Figs. 13(C1-a) and 13(C2-a) owing to the tilt of the left-half electron lens. Furthermore, Figs. 13(A-b), 13(B1-b), 13(B2-b), 13(C1-b), and 13(C2-b) show the magnified voltage plots around the hollow region where electrons are assumed to pass through. Under a perfectly aligned situation, the voltages are symmetrically distributed around the hollow region in Fig. 13(A-b). In contrast, the low-voltage regions shift to the left in Figs. 13(B1-b), 13(B2-b), 13(C1-b), and 13(C2-b) owing to lens misalignment. Moreover, the magnified electrostatic fields of the hollow regions show in Figs. 13(A-c), 13(B1-c), 13(B2-c), 13(C1-c), and 13(C2-c) provide direct information to determine if the emitted electrons can successfully pass through the lenses in the designed direction. As shown in Fig. 13(A-c), the electrostatic fields are uniformly distributed outside the hollow region. Thus, electrons will not stray when they pass through a well-aligned system. On the other hand, stronger electrostatic fields concentrate on the right in Figs. 13(B1-c), 13(B2-c), 13(C1-c), and 13(C2-c). This means that electrons will be pulled to the right when passing through the hollow region. Generally, we can see that the electrostatic field becomes stronger when the distance of the z-shift and the tilted angle increase.

The calculated voltage distributions and electrostatic field distributions at the center of the right-half 90 V lens (x–y plane) are shown in Fig. 14. Figures 14(A-a), 14(B1-a), 14(B2-a), 14(C1-a), and 14(C2-a) are the voltage distributions at the center of the right-half 90 V lens. The voltages of the left-half regions drop to about 80 V and linearly decrease inward to 72 V in Figs. 14(B2-a) and 14(C2-a) because of the z-direction shift and the tilt of the left-half electron lens. Figures 14(A-b), 14(B1-b), 14(B2-b), 14(C1-b), and 14(C2-b) are the magnified voltage plots around the hollow region where electrons are assumed to pass through. The result for Fig. 14(A-b) is symmetrically distributed around the hollow region. The result in Fig. 14(A-b) is similar to that in Fig. 13(A-b) because of perfect alignment. In contrast, the low-voltage regions shift to the left in Figs. 14(B1-b), 14(B2-b), 14(C1-b), and 14(C2-b) owing to lens misalignment. Furthermore, the calculated magnified electrostatic fields of the hollow region are shown in Figs. 14(A-c), 14(B1-c), 14(B2-c), 14(C1-c), and 14(C2-c). As seen from Fig. 14(A-c), the electrostatic fields are uniformly distributed outside the hollow region. Therefore, electrons will not stray under a perfectly aligned system. On the other hand, electrostatic fields are intensified on the

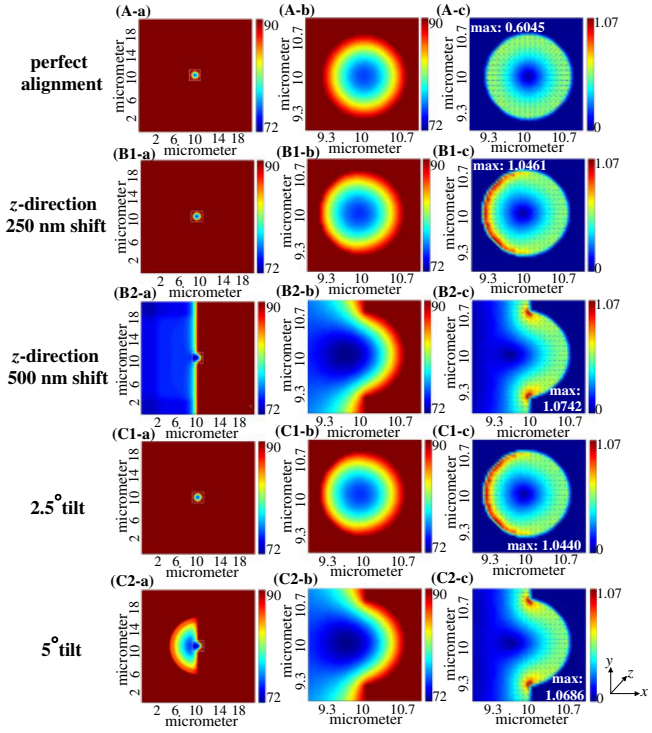

Fig. 14. (Color online) Calculated voltage distributions at center of right-half 90 V lens [subplots (A-a), (B1-a), (B2-a), (C1-a), and (C2-a)], magnified plots [subplots (A-b), (B1-b), (B2-b), (C1-b), and (C2-b)], and electrostatic fields of the magnified plots [subplots (A-c), (B1-c), (B2-c), (C1-c), and (C2-c)]. (A-a), (A-b), and (A-c) show the plots for a perfectly aligned case. (B1-a), (B1-b), and (B1-c) show the plots in the case in which the 90 V lens has a 250 nm z-direction misalignment. (B2-a), (B2-b), and (B2-c) show the plots in the case in which the 90 V lens has a 500 nm z-direction misalignment. (C1-a), (C1-b), and (C1-c) show the plots in the case in which the left-half 90 V lens has a 2.5° downward tilt. (C2-a), (C2-b), and (C2-c) show the plots in the case in which the left-half 90 V lens has a 5° downward tilt.

left in Figs. 14(B1-c) and 14(C1-c) because the stronger gradients in Figs. 14(B1-b) and 14(C1-b) are on the left. Thus, electrons will be pulled to the left when passing through these two systems. Moreover, stronger electrostatic fields concentrate on the right in Figs. 14(B2-c) and 14(C2-c). So, the electrons will be pulled to the right when passing through these two systems.

4. Conclusions

The solution-refined finite-difference method for solving Laplace's equation is proposed. The advantage of this method is that a large-scale problem can be solved even if the required memory exceeds the storage capability of the computer. We found that the numerical solutions are close to the analytical solutions and are maintained stably by solution refinement. Moreover, we calculated the 3D electrostatic fields for a well-aligned lens, a lens with a z-direction shift, and a lens in a tilted situation for the electron-beam direct-write lithography system using solution-refined FDM combined with the MPI scheme. A satisfactory simulation result can be obtained. One of the drawbacks of our proposed method is that it is time-consuming to obtain stable solutions during the refinements. However, this method can be useful for solving a large-scale problem that exceeds the memory or storage installation capability of high-performance computers. The proposed numerical method can be used to solve Laplace's equation, as shown in this paper, and it can also be used to solve other partial differential equations for solving large-scale problems.

Acknowledgments

This work was supported by the National Science Council, Taiwan (NSC-98-2622-E-002-003-CC1) and the NTU Excellent Research Project (10R80914-4). We are grateful to the National Center for High-Performance Computing, Taiwan, for providing us with the computation time and facilities.

1) International Technology Roadmap for Semiconductors (2011) [http://public.itrs.net].
2) X. Wang, Z. Yang, G. Huang, and B. Chen: Commun. Nonlinear Sci. Numer. Simulation **14** (2009) 379.
3) C. Pozrikidis: Eng. Anal. Boundary Elem. **30** (2006) 143.
4) E. Munro: Nucl. Instrum. Methods Phys. Res., Sect. A **645** (2011) 266.
5) J. R. Gibson, K. G. Evans, and S. Taylor: J. Mass Spectrom. **45** (2010) 364.
6) L. Mingyang, L. Deming, D. Lei, and G. Jie: J. Electron. **4** (1987) 305.
7) SIMION® Version 8.1 [http://simion.com/].
8) Poisson Superfish [http://laacg.lanl.gov/laacg/services/download_sf.phtml].
9) LORENTZ Version 7.0 [http://www.scientific-computing.com/].
10) D. Cubric, B. Lencova, F. H. Read, and J. Zlamal: Nucl. Instrum. Methods Phys. Res., Sect. A **427** (1999) 357.
11) T. Chisholm, B. A. Wallman, H. Liu, E. Munro, J. A. Rouse, and X. Zhu: Proc. SPIE **2858** (1996) 146.
12) E. Munro, X. Zhu, J. Rouse, and H. Liu: Proc. SPIE **2522** (1995) 54.
13) K. Tahir, J. Rouse, and E. Munro: Proc. SPIE **2014** (1993) 144.
14) E. Munro: Proc. SPIE **2014** (1993) 2.
15) X. Zhu, E. Munro, J. A. Rouse, and W. K. Waskiewicz: Proc. SPIE **3997** (2000) 170.
16) E. Munro: Proc. SPIE **3777** (1999) 215.
17) T. Sawada and K. Yoneda: IEEE Trans. Magn. **33** (1997) 1730.
18) B. Lencová: Nucl. Instrum. Methods Phys. Res., Sect. A **363** (1995) 190.
19) B. Lencová: Nucl. Instrum. Methods Phys. Res., Sect. A **519** (2004) 149.
20) B. Lencová: Ultramicroscopy **93** (2002) 263.
21) S. Y. Chen, S. C. Chen, H. H. Chen, K. Y. Tsai, and H. H. Pan: Jpn. J. Appl. Phys. **49** (2010) 06GE05.
22) V. Maris and P. E. Wannamaker: Comput. Geosci. **36** (2010) 1384.
23) D. P. Rizzetta, M. R. Visbal, and P. E. Morgan: Prog. Aerosp. Sci. **44** (2008) 397.
24) G. Tang, E. F. D'Azevedo, F. Zhang, J. C. Parker, D. B. Watson, and P. M. Jardine: Comput. Geosci. **36** (2010) 1451.
25) M. J. Chorley and D. W. Walker: J. Comput. Sci. **1** (2010) 168.
26) S. Adams, J. Payne, and R. Boppana: Proc. 2007 DoD High Performance Computing Modernization Program Users Group (HPCMP) Conf., 2007, p. 334.
27) A. S. Antoniou, K. I. Karantasis, E. D. Polychronopoulos, and J. A. Ekaterinaris: AIAA-2010-0525.
28) H. Wang, A. Trakic, F. Liu, B. K. Li, E. Weber, and S. Crozier: Int. J. Antennas Propag. **2008** (2008) 1.
29) K. Taguchi, M. Uchiya, T. Kashiwa, K. Hirayama, H. Kuribayashi, and S. Komatsu: Int. J. RF Microwave Comput. Aided Eng. **14** (2004) 253.
30) S. Mitran: Procedia Comput. Sci. **1** (2010) 745.
31) D. K. Chen: in *Field and Wave Electromagnetics*, ed. B. Rifkind (Addison-Wesley, Reading, MA, 1989) p. 427.
32) A. Taflove and S. C. Hagness: *Computational Electrodynamics: The Finite-Difference Time-Domain Method* (Artech House, Norwood, MA, 2005) 3rd ed., p. 51.
33) G. M. Mladenov, K. J. Vutova, and E. G. Koleva: Phys. Chem. Solid State **10** (2009) 707.
34) M. Stepanova, T. Fito, Zs. Szabó, K. Alti, A. P. Adeyenuwo, K. Koshelev, M. Aktary, and S. K. Dew: J. Vac. Sci. Technol. B **28** (2010) C6C48.
35) N. Tsikrikas, G. P. Patsis, I. Raptis, A. Gerardino, and E. Quesnel: Jpn. J. Appl. Phys. **47** (2008) 4909.
36) X. Yang, W. L. Gardner, L. R. Baylor, H. Cui, D. H. Lowndes, D. C. Joy, and M. L. Simpson: J. Vac. Sci. Technol. B **25** (2007) 394.
37) M. Nakasuji and H. Shimizu: Jpn. J. Appl. Phys. **39** (2000) 1934.
38) E. Olsson and G. Kreiss: J. Comput. Phys. **210** (2005) 225.