Journal of Engineering Physics and Thermophysics, Vol. 70, No. 1, 1997

ALGORITHMS FOR SOLVING ALGEBRIZED TRANSFER EQUATIONS

S. S. Belyavskii, S. G. Mulyarchik, and A. V. Popov

UDC 539.219.1

Algorithms for solving algebrized transfer equations on a continuous rectangular grid are proposed. The efficiency of the methods proposed is illustrated by performing a numerical two-dimensional simulation of a submicron bipolar transistor in the high injection-level mode.

A number of transfer phenomena in liquid, solid, and gaseous media whose analysis is important for the solution of scientific or engineering problems are described by systems of partial differential equations (PDE). Upon the spatial discretization of a given system using the finite-difference or finite-element approach and linearization, the problem is reduced to the solution of systems of linear algebraic equations (SLAE). The matrices of the SLAE of the class of problems under consideration are usually characterized by a high order (tens of thousands or even millions), are sparse, ill-conditioned, and, as a rule, unsymmetrical. In this case, iterative procedures for solving linear systems have an advantage over direct methods [1]. The efficiency of solving the entire problem depends to a great extent on the efficiency of solving the SLAE, since the expenditures connected with solving linear systems are predominant.

Presently, the biconjugate gradient method (BCG) [2] and its improved versions CGS [3] and CGSTAB [3] are widely used in solving large sparse SLAE with an unsymmetrical matrix. A number of results attest to the efficiency of BCG-type algorithms in the analysis of charge transfer processes in VLSI elements [5, 6], oil-well simulations [7], or in solving a system of Navier–Stokes equations describing the motion of a viscous incompressible liquid [8]. The development of efficient iterative algorithms for solving SLAE is important in solving heat transfer equations [9], simulations of transfer phenomena in a dense electron-hole semiconductor plasma [10], and in a number of other important engineering applications.

In order to improve the efficiency of the methods [2-4] we propose a procedure of constructing a refined initial approximation developed primarily for the case of a symmetrical matrix of the SLAE being solved [11]. In the absence of symmetry, in contrast to [11], two orthogonal projectors should be used. Projection algorithms for the BCG, CGS, and CGSTAB methods are described. We propose efficient versions of the algorithms developed for solving systems of five-point grid equations approximating two-dimensional boundary-value problems on rectangular grids. Generalizations of projection operators described in [11, 12] are used. Results of a numerical study on the problem of two-dimensional simulation of charge transfer processes in elements of silicon VLSI with the use of the drift-diffusion approximation are presented. The results of experiments indicate the efficiency of the algorithms proposed.

Let us consider the SLAE

$$Ax = b, \tag{1}$$

where A is an $n \times n$ symmetrical matrix of coefficients. The preconditioned algorithm of the BCG method [2] can be written as follows.

BCG algorithm.

1. Construction of the preconditioner H.

Belarusian State University, Minsk, Belarus. Translated from Inzhenerno-Fizicheskii Zhurnal, Vol. 70, No. 1, pp. 156-162, January-February, 1997. Original article submitted October 3, 1994.

1062-0125/97/7001-0161$18.00

2. Initialization $r_0 = b - Ax_0$, $\hat{r}_0 = r_0$, $\rho_0 = 1$, $p_0 = \hat{p}_0 = 0$.
3. Iteration refinement ($i = 1, 2, 3, \ldots$):
   3.1) solution of the auxiliary system $Hy = r_{i-1}$;
   3.2) $\rho_i = \hat{r}_{i-1}^{\mathrm{T}} y$;
   3.3) $\beta_i = \rho_i / \rho_{i-1}$;
   3.4) $p_i = y + \beta_i p_{i-1}$;
   3.5) solution of the auxiliary system $H^{\mathrm{T}} z = \hat{r}_{i-1}$;
   3.6) $\hat{p}_i = z + \beta_i \hat{p}_{i-1}$;
   3.7) $v = Ap_i$;
   3.8) $\alpha_i = \rho_i / (\hat{p}_i^{\mathrm{T}} v)$;
   3.9) $x_i = x_{i-1} + \alpha_i p_i$;
   3.10) $r_i = r_{i-1} - \alpha_i v$;
   3.11) if $\| r_i \|$ is sufficiently small, terminate computations;
   3.12) $\hat{r}_i = \hat{r}_{i-1} - \alpha_i A^{\mathrm{T}} \hat{p}_i$.

In this case the following relationships take place [2]:

$$r_i^{\mathrm{T}} \hat{r}_j = \hat{r}_i^{\mathrm{T}} r_j = 0, \quad i \neq j,$$

$$\hat{p}_i^{\mathrm{T}} Ap_j = p_i^{\mathrm{T}} A^{\mathrm{T}} \hat{p}_j = 0, \quad i \neq j; \quad \hat{p}_i^{\mathrm{T}} Ap_i = p_i^{\mathrm{T}} A^{\mathrm{T}} \hat{p}_i \neq 0. \tag{2}$$

Let assume that the vectors

$$p_0, p_1, \ldots, p_{k-1} \quad \text{and} \quad \hat{p}_0, \hat{p}_1, \ldots, \hat{p}_{k-1},$$

where $k < n$, which satisfy condition (2), are known. We denote their linear spans as E and F, respectively. Then, using the same reasoning as in [11, 12], we can build a projector on $F^{\perp}$ along E, where $F^{\perp}$ is the A-orthogonal complement of F,

$$R = I - \sum_{i=0}^{k-1} \frac{p_i (A^{\mathrm{T}} \hat{p}_i)^{\mathrm{T}}}{\hat{p}_i^{\mathrm{T}} Ap_i}$$

and the projector on $E^{\perp}$ along F

$$\hat{R} = I - \sum_{i=0}^{k-1} \frac{\hat{p}_i (Ap_i)^{\mathrm{T}}}{\hat{p}_i^{\mathrm{T}} Ap_i},$$

where I is the $n \times n$ identity matrix. Now the solution of system (1) can be presented as follows:

$$x = y + z = \sum_{i=0}^{k-1} \alpha_i p_i + z, \quad \text{where} \quad \alpha_i = \frac{\hat{p}_i^{\mathrm{T}} b}{\hat{p}_i^{\mathrm{T}} Ap_i} \quad \text{and} \quad z \in E^{\perp}.$$

In this case it is sufficient to determine just one component of z that solves the system

$$Az = \hat{R}^{\mathrm{T}} b. \tag{3}$$

It can be shown that $z = Rx$ and therefore, instead of (3), one can solve the SLAE

$$ARx = \hat{R}^{\mathrm{T}} b, \tag{4}$$

with the singular matrix AR. Nevertheless, system (4) is compatible, since rank AR = rank $[AR, \hat{R}^T b]$ = rank R = $n-k$. The solution of (4) is not unique. In order to make x from (4) a solution of initial system (1), one must define an additional condition. The condition is obviously a refinement of the initial approximation in subspace E

$$y = \sum_{i=0}^{k-1} \alpha_i p_i . \tag{5}$$

It is known that the BCG method solves, in addition to (1), the conjugate system $A^T\hat{x} = b$ with the following initial approximation in subspace F:

$$\hat{y} = \sum_{i=0}^{k-1} \hat{\alpha}_i \hat{p}_i , \tag{6}$$

where $\hat{\alpha}_i = p_i^T \hat{b} / p_i^T A p_i$.

Relationships (5) and (6) determine the initial approximation for the BCG method. Now A-orthogonality of vectors p_i to the hyperplane $ARw = 0$ and A^T-orthogonality of vectors $\hat{p}_i$ to $A^T Rw = 0$ should be ensured in the iterative refinement process. The required orthogonalization is carried out in the framework in the projection BCG algorithm proposed in what follows.

Projection BCG algorithm.

1. Construction of the preconditioner H.
2. Computation of the refined initial approximation from (5), (6).
3. Initialization $s = b - Ax_0$, $r_0 = s - Ay$, $\hat{r}_0 = s - A^T\hat{y}$, $\rho_0 = 1$, $p_0 = \hat{p}_0 = 0$.
4. Iterative refinement ($i = 1, 2, 3, \ldots$):
   - 4.1) solution of the auxiliary system $H\tilde{y} = r_{i-1}$;
   - 4.2) projection $y = R\tilde{y}$;
   - 4.3) $\rho_i = \hat{r}_{i-1}^T y$;
   - 4.4) $\beta_i = \rho_i / \rho_{i-1}$;
   - 4.5) $p_i = y + \beta_i p_{i-1}$;
   - 4.6) solution of the auxiliary system $H^T z = \hat{r}_{i-1}$;
   - 4.7) $\hat{p}_i = \hat{R}z + \beta_i \hat{p}_{i-1}$;
   - 4.8) $v = Ap_i$;
   - 4.9) $\alpha_i = \rho_i / (\hat{p}_i^T v)$;
   - 4.10) $x_i = x_{i-1} + \alpha_i p_i$;
   - 4.11) $r_i = r_{i-1} - \alpha_i v$;
   - 4.12) if $\|r_i\|$ is sufficiently small, terminate computations;
   - 4.13) $\hat{r}_i = \hat{r}_{i-1} - \alpha_i A^T \hat{p}_i$.

If a sufficiently large number of vectors (3) satisfying condition (2) can be built, and the initial approximation (5), (6) and projection operators R and $\hat{R}$ can be computed on their basis, then the iteration expenditures for the projection BCG-algorithm can be substantially smaller than those of the traditional method [2]. The reason is in the fact that the dimension of the matrix-vector products (items 4.8 and 4.13), scalar products (items 4.3 and 4.9), and operations of the $s = ax + y$ (saxpy) type (items 4.5, 4.7, 4.11, and 4.13) is $n - k$ instead of n. The only condition is that the economy obtained should be greater than the cost of two additional projection operations (items 4.2 and 4.7). The use of generalized projectors [11, 12] leads to total reduction of the number of arithmetical operations for the projection algorithm, as will be shown in what follows. In the BCG algorithm both sequences r_i and $\hat{r}_i$ converge to zero. However, only convergence of r_i is used in the solution of system (1). In the CGS algorithm [3] the sequence $\hat{r}_i$ is not built at all, and all efforts are concentrated on converging r_i to zero.

Using the proposed method of construction of the refined initial approximation, we obtain the projection algorithm of the CGS method.

Projection CGS algorithm.

1. Construction of the preconditioner H.
2. Computation of the refined initial approximation from (5), (6).
3. Initialization $s = b - Ax_0$; $r_0 = s - Ay$, $\hat{r} = s - A^T\hat{y}$, $\rho_0 = 1$, $p_0 = q_0 = 0$.
4. Iterative refinement ($i = 1, 2, 3, \ldots$):
   4.1) $\rho_i = \hat{r}^T r_{i-1}$;
   4.2) $\beta_i = \rho_i/\rho_{i-1}$;
   4.3) $u = r_{i-1} + \beta_i q_{i-1}$;
   4.4) $p_i = u + \beta_i(q_{i-1} + \beta_i p_{i-1})$;
   4.5) solution of the auxiliary system $H\tilde{y} = p_i$;
   4.6) projection $y = R\tilde{y}$;
   4.7) $v = Ay$;
   4.8) $\alpha_i = p_i/(\hat{r}^T v)$;
   4.9) $q_i = u - \alpha_i v$;
   4.10) solution of the auxiliary system $H\tilde{z} = u + q_i$;
   4.11) projection $z = R\tilde{z}$;
   4.12) $x_i = x_{i-1} + \alpha_i z$;
   4.13) $r_i = r_{i-1} - \alpha_i Az$;
   4.14) if $\|r_i\|$ is sufficiently small, terminate computations.

In the given algorithn, computations of the scalar products (items 4.1 and 4.8), matrix-vector products (items 4.7 and 4.13) and saxpy (items 4.3, 4.4, 4.9, and 4.13) have a dimension of $n - k$. The absolute gain for the CGS algorithm as a result of application of the projective transformation appears to be one saxpy operation were than for the BCG method. At the same time, the number of projection operations remains the same (items 4.6 and 4.11).

A number of results bears witness to an advantage of the CGS over the BCG (see, e.g., [3, 6, 7]. However, both of the methods have a negative feature consisting in the absence of minimizing properties. As a result, a nonmonotonic convergence to zero of the discrepancy sequence is observed. The CGSTAB algorithm [4] is free of this shortcoming, which results from the special choice of the polynomial form of the method. CGSTAB minimizes discrepancy in the $\|r\|_2$ norm. Prior to passing to the consideration of the algorithm, we specify the preconditioning procedure. In the present work for all methods under investigation an incomplete LU-factorization of the matrix A in the form $ILU(0)$ [13] is considered as H.

By applying the procedure of construction of the refined initial approximation we obtain the following algorithm.

Projection CGSTAB algorithm.

1. Construction of the preconditioner $H = LU$.
2. Computation of the refined initial approximation from (5), (6).
3. Initialization $s = b - Ax_0$, $r_0 = s - Ay$, $\hat{r} = s - A^T\hat{y}$, $\rho_0 = \alpha = \omega_0 = 1$, $p_0 = v_0 = 0$.
4. Iterative refinement ($i = 1, 2, 3, \ldots$).
   4.1) $\rho_i = \hat{r}^T r_{i-1}$; $\beta_i = (\rho_i/\rho_{i-1})(\alpha/\omega_{i-1})$;
   4.2) $p_i = r_{i-1} + \beta_i(p_{i-1} - \omega_{i-1}v_{i-1})$;
   4.3) solution of the auxiliary system $H\tilde{y} = p_i$;
   4.4) projection $y = R\tilde{y}$;
   4.5) $v_i = Ay$;
   4.6) $\alpha = \rho_i/(\hat{r}^T v_i)$;
   4.7) $q = r_{i-1} - \alpha v_i$;
   4.8) solution of the auxiliary system $H\tilde{z} = q$;
   4.9) projection $z = R\tilde{z}$;

TABLE 1. Efficiency Estimate for Algorithms Under Investigation

Method	Projector	Iteration cost
BCG	–	$29t_m + 23t_a$
	NP2	$25t_m + 18t_a$
	NP3	$26 \cdot 1/3t_m + 17 \cdot 2/3t_a$
CGS	–	$30t_m + 25t_a$
	NP2	$25.5t_m + 19t_a$
	NP3	$26 \cdot 2/3t_m + 18 \cdot 1/3t_a$
CGSTAB	–	$35t_m + 28t_a$
	NP2	$29.5t_m + 21.5t_a$
	NP3	$30 \cdot 1/3t_m + 20 \cdot 2/3t_a$

4.10) $t = Az$;

4.11) $\omega_i = ((L^{-1}t)^T(L^{-1}q))/((L^{-1}t)^T(L^{-1}t))$;

4.12) $x_i = x_{i-1} + \alpha y + \omega_i z$;

4.13) $r_i = q - \omega_i t$;

4.14) if $\|r_i\|$ is sufficiently small, terminate computations.

It should be noted that item 4.11 assumes the solution of an auxiliary system with the lower triangular matrix, while the vector $L^{-1}q$ is an intermediate result of computations in item 4.8. It is easy to verify that application of the projective transformation to the CGSTAB algorithm is even more efficient than in the case of the two preceding algorithms, since in this case $n-k$ is the dimension of four scalar products (items 4.1, 4.6, and 4.11), two matrix-vector products (items 4.5 and 4.10), and four saxpy-type operations over vectors (items 4.2, 4.7, and 4.13).

We now come to the description of the implementation of the proposed projection algorithms. We will consider systems of five-point grid equations approximating two-dimensional boundary-value problems on a continuous rectangular grid. In this case the matrix of coefficients of system (1) has a five-diagonal structure. In numerical experiments, projection BCG, CGS, and CGSTAB algorithms are investigated along with generalizations of NP2 and NP3 projectors in terms of [12]. In this case, the estimate presented in Table 1 holds for iteration costs. For comparison, Table 1 also presents the iteration cost for the traditional implementation of the methods [2-4]. It is evident that the algorithms described in the present work have lower iteration costs than the standard versions of the methods [2-4], which explains their advantage for the same number of iterations required to reach a given accuracy.

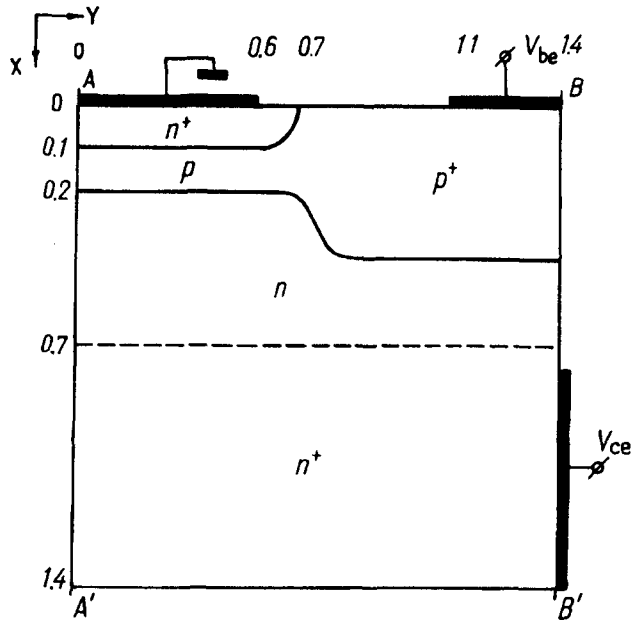
Let us consider the problem of numerical simulation of charge-transfer processes in elements of silicon VLSI. The drift-diffusion model implies solution of the following set of PDE [14]:

$$\nabla \cdot (\varepsilon \nabla \psi) = -\rho , \tag{7}$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla j_n - R , \tag{8}$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla j_p - R , \tag{9}$$

where



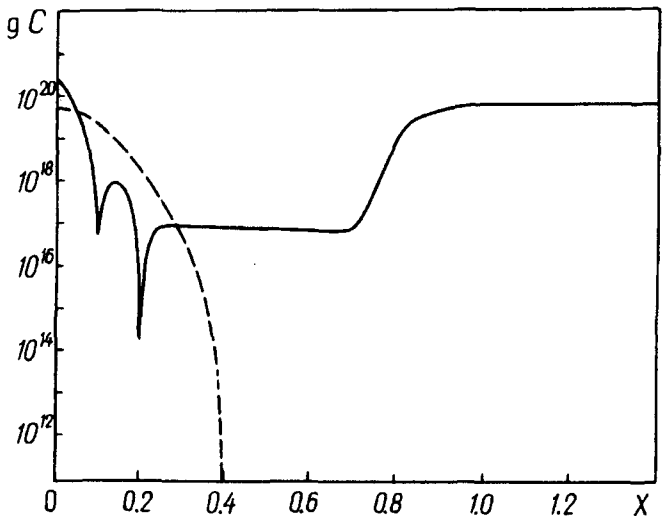


Fig. 1. Structure of a submicron bipolar transistor. X, μm, Y, μm.

Fig. 2. Impurity distribution over cross-sections. log C, cm^{-3}.

$$j_n = -q\mu_n (n\nabla\psi - kT\nabla n),$$

$$j_p = -q\mu_p (p\nabla\psi + kT\nabla p),$$

$$\rho = \begin{cases} q(p - n + C) & - \text{ in interior of semiconductor,} \\ Q_{SS} & - \text{ at the } Si/SiO_2 \text{ interface.} \end{cases}$$

The given model accounts for band gap narrowing, Shockley–Reed and Auger recombination mechanisms, generation by avalanche multiplication in a strong field, photoexcitation, and tunneling. The mobility model takes into account scattering on ionized impurities, acoustic phonons, electron-hole scattering, surface scattering, and the phenomenon of saturation of the drift velocity [14, 15].

One of most widespread approaches to solving nonlinear system (7)-(9) is successive iteration until a self-consistent solution is obtained [14, 15]. In this case, three SLAE should be successively solved in each external iteration. When using continuous rectangular grids for spatial discretization of the problem and five-point difference schemes, the matrices of coefficients of the specified SLAE have a five-diagonal structure. The matrix of the algebrized Poisson equation (7) is a symmetrical positive definite matrix. As a result of the strong diagonal prevailing, a small number of iterations of the preconditioned method of conjugate gradients is usually required for solving the corresponding linear system [13]. Efficient projection algorithms from [11, 12] were used in solving the SLAE of the Poisson equation in the experiments described below. When using charge carrier concentrations as variables, the matrices of linear systems of algebrized continuity equations (8) and (9) are unsymmetrical. SLAE of the type present serious difficulties for numerical solution [6]. To do this, we will apply here all the algorithms described in the present work.

As a test example, we consider a submicrometer-sized bipolar transistor with the depth of the emitter transition of 0.1 μm (Fig. 1). The distribution of the impurity over cross-sections $A' - A$ (solid curve) and $B' - B$ (dashed curve) is presented in Fig. 2. External displacements $V_{be} = 0.85$ V and $V_{ce} = 5$ V correspond to the high injection-level mode. The size of the spatial discretization grid is $36 \times 49 = 1764$ nodes. When using the successive quasi-Hummel algorithm KG1 [14] for solving the nonlinear system, a self-consistent solution was obtained after 24 external iterations (the condition $\delta < 10^{-4}$ was chosen as a convergence criterion, where δ is the

TABLE 2. Results of Computational Experiments

Method	Projector	N	t, sec	τ, sec
BCG	–	3194	923.82	1116.47
	NP2	2586	625.24	817.63
	NP3	2373	555.93	748.58
CGS	–	2524	716.91	909.24
	NP2	2148	518.21	710.90
	NP3	1996	469.25	661.52
CGSTAB	–	2200	796.44	990.96
	NP2	1988	581.84	773.07
	NP3	1868	521.62	712.99

maximum absolute value of the relative variation of the electrostatic potential). The accuracy of the solution of the SLAE increased with decreasing δ, according to [12].

Table 2 presents the total number of iterations N for each of the methods under investigation, the total time t for solving all SLAE and the time τ for solving the entire problem of simulation of a submicron transistor in the given mode. All the times correspond to calculations on an IBM PC/AT 386/387.

Analysis of Table 2 shows the high efficiency of application of projection operators in combination with the algorithms [2-4]. A decrease in time costs results from a decrease in iteration costs (see Table 1). In addition, as follows from Table 2, the use of orthogonal projectors and auxiliary subspaces leads to a decrease in the number of iterations, since the approach being developed involves preconditioning by means of a singular matrix. It is evident that the SLAE solution time comprises the main portion of the total solution time of the problem.

Thus, the algorithms proposed make it possible to improve the efficiency of solving the problem of numerical simulation of microelectronic structures. Inasmuch as SLAE with matrices of the considered structure emerge in various engineering and physical applications, the methods proposed have a wider area of application.

NOTATION

ε, relative dielectric constant; ψ, electrostatic potential; ρ, charge; n, p, concentrations of electrons and holes; t, time; j_n, j_p, densities of electron and hole currents; R, excess of recombination rate of charge carriers over generation rate; q, elementary charge; μ_n, μ_p, electron and hole mobilities; k, Boltzmann constant; T, absolute temperature; C, resulting dopantt concentration; Q_{ss}, bound charge at the Si/SiO_2 interface; t_m, t_a, performance times for floating-point multiplication and addition operations; Indices: ce, collector-emitter; be, base-emitter.

REFERENCES

1. L. Heigeman and D. Young, Applied Iterative Methods [Russian translation], Moscow (1986).
2. R. Fletcher, Lect. Notes Math., No. 506, 73-89 (1976).
3. P. Sonneveld, SIAM J. Sci. Stat. Comput., 10, No. 1, 36-52 (1969).
4. H. A. van der Vorst, SIAM J. Sci. Stat. Comput., 13, No. 2, 631-644 (1992).
5. C. den Heijer, in: Simulation of Semiconductor Devices and Processes, Swansea (1984), pp. 267-285.
6. G. Heiser, C. Pommerell, J. Weis, and W. Fichtner, IEEE Trans., CAD-10, No. 10, 1218-1230 (19910.
7. P. Joly and R. Eymard, J. Comput. Phys., 91, No. 2, 298-309 (1990).
8. E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang, BIT, 32, No. 3, 442-463 (1992).
9. G. E. Schneider and M. Zedan, Numer. Heat Transfer, 4, No. 1, 1-19 (1981).

10. V. A. Nikolaeva, V. I. Ryzhii, and B. N. Chetvertushkin, Inzh.-Fiz. Zh., **51**, No. 3, 492-501 (1986).
11. S. G. Mulyarchik, S. S. Bielawski, and A. V. Popov, J. Comput. Phys., **110**, No. 2, 201-211 (1994).
12. S. S. Belyavskii, S. G. Mulyarchik, and A. V. Popov, Differents. Uravn., **29**, No. 9, 1575-1584 (1993).
13. J. A. Meijerink and H. A. van der Vorst, Math. Comput., **31**, No. 137, 1480162 (1977).
14. S. G. Mulyarchik, Numerical Simulation of Microelectronic Structures [in Russian], Minsk (1989).
15. S. Selberherr, Analysis and Simulation of Semiconductor Devices, New York (1984).