JOURNAL OF COMPUTATIONAL PHYSICS 110, 201–211 (1994)

Efficient Computational Schemes of the Conjugate Gradient Method for Solving Linear Systems

STEPAN G. MULYARCHIK, STANISLAV S. BIELAWSKI, AND ANDREW V. POPOV

Belarusian State University, prosp. F. Scoriny, 4, Minsk, 220050, Belarus, Russia

Received February 19, 1992; revised December 29, 1992

The technique of the initial guess calculation for the conjugate gradient method is proposed. Computational schemes of the linear system solution with symmetrical positive definite matrices are constructed on its basis. Their efficient modifications for systems with five-diagonal matrices are proposed. The investigation of the developed methods using the problem of two-dimensional numerical simulation of bipolar transistors has been carried out. Experimental evidence of the proposed method's efficiency has been obtained.

1. INTRODUCTION

The well-known conjugate gradient (CG) method was first applied to linear system solution apparently in [1]. From the beginning the various nonstationary iterative methods were used as preconditioners. Meanwhile, [2] proposed the idea of applying incomplete decomposition to the first-order iterative methods. Later, [3] presented a complete description of the ICCG. Today preconditioned by the incomplete Cholesky (IC) factorization the CG-method is quite efficient for symmetrical positive definite (SPD) problems. At first one of the main lines of developing the ICCG-method was searching and investigating various kinds of incomplete decomposition [4–6]. A number of references (see, for example, [7–9]) report attempts to extend the ICCG application field (it can be pointed that some generalizations of the CG-method have been obtained earlier, for example, [10]). The authors, for instance, of [11–13] made efforts to develop parallel and vector versions of the ICCG-method. These and other relevant aspects dealing with the preconditioned CG-type methods can be found, for example, in the survey [14].

The present paper proposed another kind of ICCG development. We assume that some of the conjugate directions are defined. Then an orthogonal projector can be constructed on their basis. Linear systems to be solved can be changed using the projector. Moreover, an inexpensive and more correct initial guess for the CG-method can be obtained. This approach allows us to develop a number of efficient modifications of the ICCG-method for five-diagonal linear systems. Experimental evidence of the proposed method's efficiency has been obtained.

Recently, deflation of the conjugate gradients has been proposed [15]. This approach consists of the orthogonalization of the current residual, not only to all of the previous residuals, but also to the columns of some $n \times k$ matrix E, where n is a linear system order. These columns are a set of linearly independent vectors. However, the choice of E has some difficulties. In [15] special decomposition of the domain to be simulated was carried out. Then the columns of E were defined using information about these subdomains. At last an orthogonal projector was constructed based on E. The present paper can be regarded in terms of deflated conjugate gradients. However, we obtain the projector from the very beginning. It is important to point out that we can define the matrix E according to the structure of the linear system matrix which does not depend on the domain to be simulated. Note also that such a description of E can be easily formalized.

2. THEORETICAL GROUNDS

We wish to solve

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

where A is $n \times n$ symmetrical M-matrix. The CG-method for (1) consists of computing the sequence of n vectors $p_0, p_1, \ldots, p_{n-1}$ which are the basis in $\mathbb{R}^n$ and satisfy the condition

$$p_i^T A p_j = 0, \qquad i \neq j. \tag{2}$$

This method can be written [3]

I. Construction of preconditioner H_0.

II. Initialization $r_0 = b - Ax_0$, $p_0 = H_0^{-1} r_0$.

0021-9991/94 $6.00

III. Iteration for $i=0, 1, 2, \ldots$ until convergence *DO*

$$\alpha_i = \frac{r_i^{\mathrm{T}} H_0^{-1} r_i}{p_i^{\mathrm{T}} A p_i},$$

$$x_{i+1} = x_i + \alpha_i p_i,$$

$$r_{i+1} = r_i - \alpha_i A p_i,$$

$$\beta_i = \frac{r_{i+1}^{\mathrm{T}} H_0^{-1} r_{i+1}}{r_i^{\mathrm{T}} H_0^{-1} r_i},$$

$$p_{i+1} = H_0^{-1} r_{i+1} + \beta_i p_i.$$

Here x_0 is an initial approximation, H_0 is some $n \times n$ SPD matrix. If $H_0 = I$, where I is the identity matrix, then we obtain the CG-method.

We assume that $k < n$ of A-conjugate vectors $p_0, p_1, \ldots, p_{k-1}$ are known. Denote $E_1 = \mathrm{span}(p_0, p_1, \ldots, p_{k-1})$, $E_1 \subset \mathbb{R}^n$, $\dim E_1 = k$. Then

$$Q = \sum_{i=0}^{k-1} \frac{p_i (A p_i)^{\mathrm{T}}}{p_i^{\mathrm{T}} A p_i} \tag{3}$$

is the projector onto E_1 and $R = I - Q$ is the projector onto some subspace E_2. As it follows from [16, p. 643], $\mathbb{R}^n = E_1 \oplus E_2$.

Note one property of Q and R:

$$\begin{aligned} AQ &= Q^{\mathrm{T}} A = Q^{\mathrm{T}} A Q, \\ AR &= R^{\mathrm{T}} A = R^{\mathrm{T}} A R. \end{aligned} \tag{4}$$

THEOREM 2.1. *Let $k < n$ of A-conjugate vectors $p_0, p_1, \ldots, p_{k-1}$ are known. Then the system* (1) *solution can be written in the form*

$$x = y + z = \sum_{i=0}^{k-1} \alpha_i p_i + z, \tag{5}$$

where $\alpha_i = p_i^{\mathrm{T}} b / p_i^{\mathrm{T}} A p_i$ and $z \in E_2$ is the solution of

$$Az = R^{\mathrm{T}} b. \tag{6}$$

Proof. If we substitute (5) into (1) then system (6) will be obtained. Now we must show that $z \in E_2$. According to (4) we can write $A^{-1} R^{\mathrm{T}} = R A^{-1}$. Then the result immediately follows from $z = A^{-1} R^{\mathrm{T}} b = R A^{-1} b$. ∎

As $z = Rx$ then the following system can be considered instead of (6):

$$ARx = R^{\mathrm{T}} b. \tag{7}$$

As $\mathrm{rank}\, AR = \mathrm{rank}[AR, R^{\mathrm{T}} b] = \mathrm{rank}\, R = n - k$, then the system (7) is compatible. AR is a symmetrical (it follows from (4)) and singular matrix. However, according to [17, p. 119] the CG-method can be applied to (7). Then the solution of (7) will be defined using A-conjugate vectors $\tilde{p}_0, \ldots, \tilde{p}_{l-1}$. These vectors are orthogonal to the hyperplane $ARq = 0$. Therefore $l = n - k$. Denote $p_k = \tilde{p}_0, \ldots, p_{n-1} = \tilde{p}_{l-1}$.

The solution of (7) is not unique. It can be given in the form

$$x = y + \sum_{i=k}^{n-1} \alpha_i p_i,$$

where α_i are the coefficients of vector z decomposition by the set $p_k, \ldots, p_{n-1}$ and y is an arbitrary vector, belonging to E_1. We assume

$$y = x_0 = \sum_{i=0}^{k-1} \alpha_i p_i. \tag{8}$$

Then we obtain the solution of system (1).

Thus, if we know k of vectors $p_0, \ldots, p_{k-1}$, satisfying (2), then the solution of (1) decomposes into two independent steps. At the first step the initial guess (IG) x_0 due to (8) is determined. At the second one the system (7) is solved by the conjugate gradient method. The computational scheme of the IGCG-method follows from the scheme of the CG-method if the solution of (7) is searched over E_2:

I. Calculation of x_0 according to (8).

II. Initialization of $r_0 = b - Ax_0$, $\tilde{p}_0 = Rr_0$.

III. Iterate for $i = 0, 1, 2, \ldots$ until convergence *DO*:

$$\alpha_i = \frac{r_i^{\mathrm{T}} R r_i}{\tilde{p}_i^{\mathrm{T}} A \tilde{p}_i},$$

$$x_{i+1} = x_i + \alpha_i \tilde{p}_i,$$

$$r_{i+1} = r_i - \alpha_i A \tilde{p}_i,$$

$$\beta_i = \frac{r_{i+1}^{\mathrm{T}} R r_{i+1}}{r_i^{\mathrm{T}} R r_i},$$

$$\tilde{p}_{i+1} = R r_{i+1} + \beta_i \tilde{p}_i.$$

Note 2.1. The IGCG-scheme has an advantage because it, in the absence of round-off error, ensures the exact solution in at most $n - k$ iterations, whereas the CG-scheme gives the exact solution in at most n iterations.

The rate of convergence of the conjugate gradient method depends on the matrix A condition number $\kappa = \lambda_{\max}(A)/\lambda_{\min}(A)$ [3]. Here $\lambda_{\max}(A)$, $\lambda_{\min}(A)$ are the largest and the smallest eigenvalues of A, respectively. It is known, that a successful choice of the SPD matrix H_0 allows us to decrease the condition number of the matrix $T = H_0^{-1} A$ [7]. Therefore we shall solve the system $H_0^{-1} A x = H_0^{-1} b$ instead of (1). Let its solution be computed due to the CG-scheme. If

k of A-conjugate vectors $p_0, ..., p_{k-1}$ are known, then it is possible to construct the projector R and calculate the initial guess according to (8). Then we must find only z. It can be provided according to the IGICCG-method. Its scheme immediately follows from the ICCG-scheme:

I. Construction of preconditioner H_0.

II. Calculation of x_0 according to (8).

III. Initialization of $r_0 = b - Ax_0$, $\tilde{p}_0 = RH_0^{-1} r_0$.

IV. Iteration for $i = 0, 1, 2, ...$ until convergence DO:

$$\alpha_i = \frac{r_i^T RH_0^{-1} r_i}{\tilde{p}_i^T A \tilde{p}_i},$$

$$x_{i+1} = x_i + \alpha_i \tilde{p}_i,$$

$$r_{i+1} = r_i - \alpha_i A \tilde{p}_i,$$

$$\beta_i = \frac{r_{i+1}^T RH_0^{-1} r_{i+1}}{r_i^T RH_0^{-1} r_i},$$

$$\tilde{p}_{i+1} = RH_0^{-1} r_{i+1} + \beta_i \tilde{p}_i.$$

Note, that the following system is practically solved,

$$H_0^{-1} ARx = H_0^{-1} R^T b, \tag{9}$$

beginning from initial guess x_0. Nonzero eigenvalues of the matrix $H_0^{-1}AR$ are placed within $[\lambda_{\min}(H_0^{-1}A), \lambda_{\max}(H_0^{-1}A)]$. It can be proved using extremal properties of a quadratic forms cluster [18, p. 269]. As the search of the system (9) solution is over E_2 then the rate of convergence will be affected by eigenvalues, corresponding to eigenvectors being in E_2. Therefore the matrix $H_0^{-1}AR$ condition number over E_2 does not exceed the matrix $H_0^{-1}A$ condition number. It means that convergence of the IGICCG-method is not worse than the convergence of the ICCG-method.

In order to construct A-conjugate vectors $p_0, ..., p_{k-1}$ we use the variant of a well-known orthogonalization procedure [19, p. 149]. Let us consider some set of linearly independent vectors $q_0, q_1, ..., q_{k-1}$, $k \leqslant n$, and then we perform on its basis the required vectors p_i, $i = 0, ..., k-1$, in the form

$$\begin{aligned} p_0 &= q_0, \\ p_1 &= q_1 + \mu_{1,0} p_0, \\ p_2 &= q_2 + \mu_{2,0} p_0 + \mu_{2,1} p_1, \\ &\cdots \\ p_{k-1} &= q_{k-1} + \mu_{k-1,0} p_0 + \mu_{k-1,1} p_1 + \cdots + \mu_{k-1,k-2} p_{k-2}, \end{aligned} \tag{10}$$

where the coefficients $\mu_{i,j}$ are defined by the ratio

$$\mu_{i,j} = -\frac{q_i^T A p_j}{p_j^T A p_j}, \qquad \begin{aligned} &i = 1, 2, ..., k-1; \\ &j = 0, 1, ..., i-1, \end{aligned} \tag{11}$$

which provides the property (2).

Before construction of the specific IGICCG-schemes we first consider some general properties of Q and R in the case of the following special choice of q_i.

We define k natural numbers $i_1, i_2, ..., i_k$ such that $1 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n$. Then we introduce the subsets $N_n = \{1, 2, ..., n\}$, $N_n' = \{i_1, i_2, ..., i_k\}$, $N_n'' = N_n \backslash N_n'$. Let e_i be the ith column of an $n \times n$ identity matrix. Now we can take $q_0 = e_{i_1}, q_1 = e_{i_2}, ..., q_{k-1} = e_{i_k}$.

Then A-conjugate vectors $p_0, ..., p_{k-1}$ can be obtained according to (10), (11). The projectors Q and R also can be constructed and subspaces E_1 and E_2 can be defined.

THEOREM 2.2. *All the columns of R with numbers from N_n' and all the rows of Q with the numbers from N_n'' are zero ones.*

Proof. At first we show that $z^T Ay = y^T Az = 0$ is true for arbitrary $y \in E_1$ and $z \in E_2$. Indeed, let $x = y + z$. As $y = Qx$, $z = Rx$ then $z^T Ay = x^T R^T AQx = x^T (I - Q^T) AQx = x^T (AQ - Q^T AQ) x = 0$, according to (4).

It is clear $e_{i_j} \in E_1$, where $i_j \in N_n'$. Then $e_{i_j}^T Az = 0$ for arbitrary $z \in E_2$. It means that all the elements of vector Az, $z \in E_2$ with numbers from N_n' are zeros. As the system (7) is compatible then $R^T b$ contains corresponding zero elements for arbitrary b. It can be true only when the columns of R with numbers from N_n' are zero columns. As follows from the choice of $q_0, ..., q_{k-1}$ and (10), all the elements of vectors $p_0, ..., p_{k-1}$ with the numbers from N_n'' are zeros. This means that all the rows of Q with the numbers from N_n'' are zero rows. ∎

Note 2.2. The cost of the proposed computational schemes includes both the cost of the preliminary steps which are performed only once and the cost of the iterations. The first cost, as well as in traditional CG- or ICCG-schemes, is significantly lower than the latter, if only the problem of the initial guess construction does not require substantial computational effort. In this turn, the IGICCG-scheme will be more efficient than the ICCG-scheme if the cost of its iteration is lower than the cost of the ICCG iteration. The same reasons are valid when comparing the CG- and IGCG-schemes. In some important cases the matrix A structure allows us to construct some (rather great) number of simple conjugate vectors. Then we can implement the method of the initial guess choice on the basis of these vectors, which has negligible cost. Moreover, the matrix R structure in these cases provides the efficient iteration process.

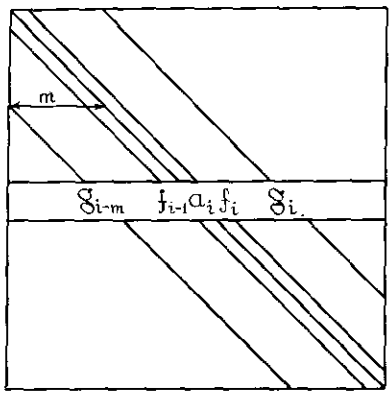

FIG. 1. Linear system matrix structure.

3. LINEAR SYSTEMS WITH THE FIVE-DIAGONAL MATRIX

Consider the linear systems with the symmetrical five-diagonal M-matrix A (Fig. 1). Assume that the IGICCG-algorithms are started from the initial approximation $x_0 = 0$. If other values of x_0 are used, then the considerations below are true for the transformed equation $A\hat{x} = \hat{b}$ where $\hat{x} = x - x_0$, $\hat{b} = b - Ax_0$.

3.1. ALGORITHM IGICCG1. We assume that m is an odd number. Consider $k = \text{int}[(n+1)/2]$ of vectors $p_i = (0, ..., 0, 1, 0, ..., 0)^T$, each of them, including only one unit component, in the odd position. These vectors satisfy condition (2). Denote $E_1 = \text{span}(p_0, ..., p_{k-1})$. As it follows from (8), the initial guess $x_0 = y$ can be written in form

$$x_0 = \left(\frac{b_1}{a_1}, 0, \frac{b_3}{a_3}, 0, ..., \frac{b_{2i-1}}{a_{2i-1}}, 0, ...\right)^T.$$

It is clear that all the odd components of the vector $r_0 = b - Ax_0$ equal zero. The matrix R structure is presented in Fig. 2. Nonzero elements of the matrix R odd row

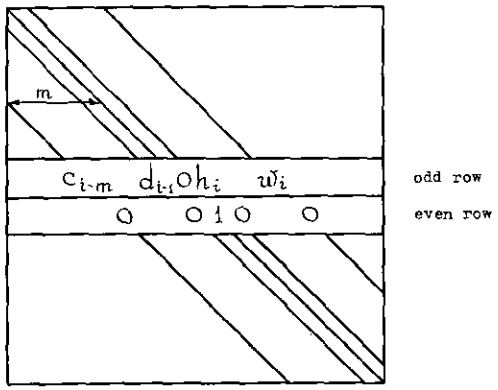

FIG. 2. Structure of projector for the IGICCG1-method.

are defined by $c_{i-m} = -g_{i-m}/a_i$; $d_{i-1} = -f_{i-1}/a_i$; $h_i = -f_i/a_i$; $w_i = -g_i/a_i$. As AR is a symmetrical matrix then the structure of R ensures that for arbitrary $q \in \mathbb{R}^n$, odd components of the vector $ARq = R^T Aq$ are also zeros. Hence, the odd elements of the vector $A\tilde{p}$ equal zeros, where $\tilde{p} \in E_2$. So, all the odd elements of the vector $r_{i+1} = r_i - \alpha_i A\tilde{p}_i$ are zeros by induction.

Hence it is not necessary to compute all the odd components of the vectors $A\tilde{p}$, $\alpha A\tilde{p}$, r. Moreover, when computing the inner products $r^T Rq$, $\tilde{p}^T A\tilde{p}$, where $q = H_0^{-1} r$, then only even components are multiplied because the odd components, at least in one factor, will be zeros.

3.2. In a more general case we shall assume that n is divisible by m and m is divisible by s. Then we introduce $k_1 = n/s$ and

$$N_n'' = \{s, 2s, ..., k_1 s\}, \qquad N_n' = N_n \backslash N_n''.$$

In this case $k = n - k_1$. Now we define vectors q_i, $i = 0, 1, ..., k-1$, as $q_i = e_j$, $j \in N_n'$, $i = j - 1 - \text{int}[j/s]$. Applying (10) and (11), we obtain k of A-conjugate vectors $p_0, ..., p_{k-1}$ and construct the projector R. In such IGICCG-algorithm vectors $p_0, ..., p_{k-1}$ have zero elements with numbers from N_n''. Then from (8) it follows that x_0 has all zero elements whose indices are divisible by s. Only entries of x_0 with numbers from N_n' need to be calculated according to Theorem 2.1.

Matrix R contains nonzero columns only with numbers si, $i = 1, 2, ..., k_1$, from Theorem 2.2. All the rows of R with numbers si, $i = 1, 2, ..., k_1$, are zero ones except for the main diagonal entries which equal unity. This fact must be taken into account when the matrix–vector product Rq will be computed. Vectors q and Rq have coinciding elements with numbers that are divisible by s. We can obtain elements of Rq with indices from N_n'' in two steps. First we calculate the values

$$\gamma_i = -\frac{(Ap_i)^T q}{p_i^T Ap_i}, \qquad i = 0, 1, ..., k-1.$$

Then we find the sum of products $\gamma_i p_i$ using backward substitution:

$$\begin{aligned} &\xi_{k-1} = \gamma_{k-1}, \\ &\xi_{k-2} = \gamma_{k-2} + \mu_{k-1,k-2}\xi_{k-1}, \\ &\qquad \cdots \\ &\xi_0 = \gamma_0 + \mu_{1,0}\xi_1 + \cdots + \mu_{k-1,0}\xi_{k-1}. \end{aligned} \tag{12}$$

As $x_0 \in E_1$ then $r_0 = b - Ax_0 = R^T b$. Hence r_0 has that all those indivisible by s elements equal zero. As $p \in E_2$ then $Ap = R^T Ap$, according to (4). Therefore all the elements of

Ap with the numbers from N_n' equal zero. As can be easily obtained from the induction vectors $r_{i+1}=r_i-\alpha_i Ap_i$, $p_i\in E_2$, also have zero entries from N_n'. Consequently only those divisible by s elements need to be multiplied when the products $p^{\mathrm{T}}Ap$, αAp, $r^{\mathrm{T}}RH^{-1}r$ will be calculated; i.e., only k_1 multiplications can be carried out.

It can be pointed that ratio for $\tilde{p}_{i+1}$ can be transformed to

$$\tilde{p}_{i+1}=R(H^{-1}r_{i+1}-\beta_i\tilde{p}_i).$$

It can be seen that only elements of $\tilde{p}_i$ with numbers from N_n'' need to be multiplied by β_i because all the elements of q with the numbers from N_n' are multiplied by zero entries of R in the product Rq.

3.2.1. ALGORITHM IGICCG2. Let $s=2$. Denote $m_1=m/2$. In this case $k_1=k$. Then application of (10), (11) leads to simple ratios for the conjugate directions,

$$\begin{aligned} &p_i=e_{2i+1}, && i=0,1,\ldots,m_1-1;\\ &p_i=e_{2i+1}+v_{2i+1-m}p_{i-m_1}, && i=m_1,\ldots,k-1, \end{aligned}$$

where the coefficients v_{2i+1} are computed recurrently

$$\begin{aligned} &v_{2i+1}=-g_{2i+1}/a_{2i+1}, && i=0,1,\ldots,m_1-1;\\ &u_{2i+1}=a_{2i+1}+v_{2i+1-m}g_{2i+1-m}, \\ &v_{2i+1}=-g_{2i+1}/u_{2i+1} && i=m_1,\ldots,k-1-m_1. \end{aligned} \tag{13}$$

Now we shall find the initial guess $x_0=y=\sum_{i=0}^{k-1}\alpha_i p_i$ using these vectors p_i. In order to compute α_i we first define

$$\begin{aligned} &\tilde{\alpha}_i=b_{2i+1}, && i=0,1,\ldots,m_1-1,\\ &\tilde{\alpha}_i=b_{2i+1}+v_{2i+1-m}\tilde{\alpha}_{i-m_1}, && i=m_1,\ldots,k-1. \end{aligned}$$

Then we obtain

$$\begin{aligned} &\alpha_i=\tilde{\alpha}_i/a_{2i+1}, && i=0,1,\ldots,m_1-1,\\ &\alpha_i=\tilde{\alpha}_i/u_{2i+1}, && i=m_1,\ldots,k-1. \end{aligned}$$

As the values u_{2i+1}, $i=m_1,\ldots,k-1$, are used in ratios for α, then we should provide an additional calculation of u_{2i+1} for $i=k-m_1,\ldots,k-1$, according to (13).

Odd components of the initial guess vector are determined by the backward substitution

$$\begin{aligned} &x_{0[2i+1]}=\alpha_i, && i=k-1,\ldots,k-m_1;\\ &x_{0[2i+1]}=\alpha_i+v_{2i+1}x_{0[2i+1+m]}, && i=k-m_1-1,\ldots,0. \end{aligned} \tag{14}$$

We shall compute the product Rq in two steps, as pointed above. At the first step we define

$$\begin{aligned} &\gamma_0=-f_1q_2/a_1,\\ &\gamma_i=-[f_{2i}q_{2i}+f_{2i+1}q_{2(i+1)}]/a_{2i+1},\\ &\qquad i=1,\ldots,m_1-1,\\ &\gamma_i=-[f_{2i}q_{2i}+f_{2i+1}q_{2(i+1)}+g_{2i+1-m}\gamma_{i-m_1}]/u_{2i+1},\\ &\qquad i=m_1,\ldots,k-1. \end{aligned}$$

At the second step the odd elements of $\tilde{q}=Rq$ are calculated according to (14) with γ against α and $\tilde{q}$ against x.

The structure of R for this case is presented in Fig. 3. Nonzero elements of R are defined by the ratios

$$\begin{aligned} \rho_{i,i-vm-1}&=\frac{f_{i-vm-1}}{a_i}\vartheta_i^{1,l}\prod_{j=1}^{v}\frac{g_{i-jm}}{a_{i-jm}};\\ \rho_{i,i-vm+1}&=\frac{f_{i-vm}}{a_i}\vartheta_i^{1,l}\prod_{j=1}^{v}\frac{g_{i-jm}}{a_{i-jm}};\\ \rho_{i,i+vm-1}&=\frac{f_{i+vm-1}}{a_{i+vm}}\vartheta_i^{v+1,l}\prod_{j=0}^{v}\frac{g_{i+jm}}{a_{i+jm}};\\ \rho_{i,i+vm+1}&=\frac{f_{i+vm}}{a_{i+vm}}\vartheta_i^{v+1,l}\prod_{j=0}^{v}\frac{g_{i+jm}}{a_{i+jm}}, \end{aligned}$$

where

$$\vartheta_i^{v,l}=1+\sigma_i^v(1+\sigma_i^{v+1}(1+\cdots(1+\sigma_i^l)\cdots)),$$

$$\sigma_i^j=\frac{g_{i+(j-1)m}^2}{a_{i+(j-1)m}a_{i+jm}},$$

$$i=1,3,\ldots,n-1;\qquad v=0,1,2,\ldots,\tilde{k};$$

$$\tilde{k}=\mathrm{int}[i/m];\qquad l=n/m-\tilde{k},\qquad \prod_{j=1}^{v}\frac{g_{i-jm}}{a_{i-jm}}=1$$

when $v=0$.

3.2.2. ALGORITHM IGICCG3. Let $s=3$. Denote $m_1=m/3$. Then

$$p_{2i}=e_{3i+1};$$

$$p_{2i+1}=e_{3i+2}+\frac{v_{3i+1}}{u_{3i+1}}p_{2i};$$

for $i=0,1,\ldots,m_1-1$ and for $i=m_1,\ldots,k_1-1$:

$$p_{2i}=e_{3i+1}-\frac{g_{3i+1-m}}{u_{3i+1-m}}p_{2(i-m_1)}+\frac{v_{3i+2-m}}{u_{3i+2-m}}p_{2(i-m_1)+1};$$

$$p_{2i+1}=e_{3i+2}-\frac{g_{3i+2-m}}{u_{3i+2-m}}p_{2(i-m_1)+1}+\frac{v_{3i+1}}{u_{3i+1}}p_{2i}.$$

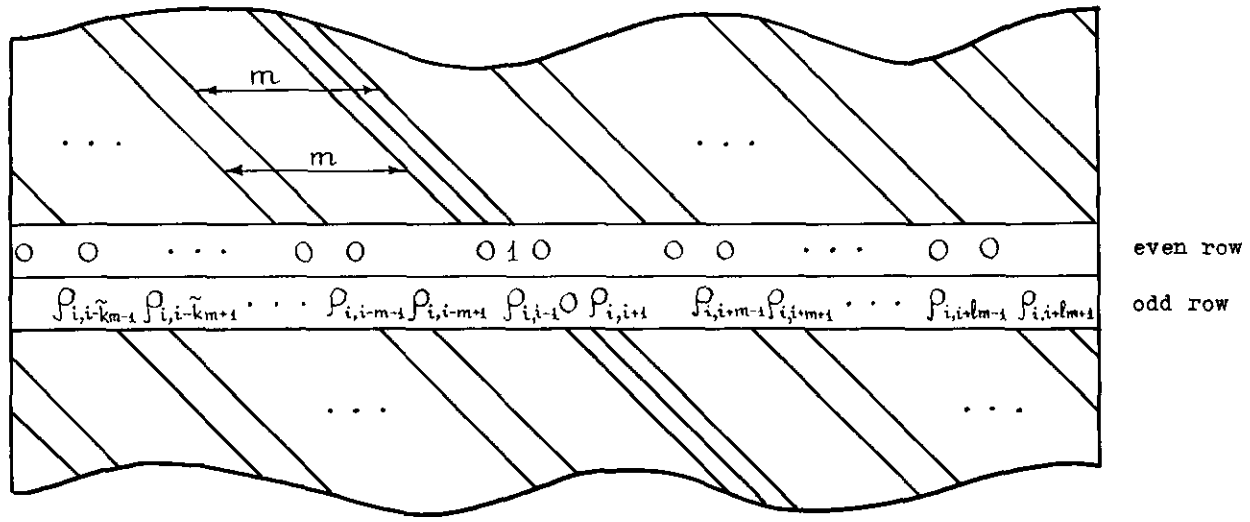

FIG. 3. Structure of projector for the IGICCG2-method.

Here U and V are given by

$$u_{3i+1} = a_{3i+1}; \qquad v_{3i+1} = -f_{3i+1};$$
$$u_{3i+2} = a_{3i+2} - f_{3i+1}^2/u_{3i+1}; \qquad v_{3i+2} = -v_{3i+1}\, g_{3i+1}/u_{3i+1}$$

for $i = 0, 1, ..., m_1 - 1$ and for $i = m_1, ..., k_1 - 1$:

$$u_{3i+1} = a_{3i+1} - g_{3i+1-m}^2/u_{3i+1-m} - v_{3i+2-m}^2/u_{3i+2-m};$$
$$v_{3i+1} = -(f_{3i+1} + g_{3i+2-m} v_{3i+2-m}/u_{3i+2-m});$$
$$u_{3i+2} = a_{3i+2} - g_{3i+2-m}^2/u_{3i+2-m} - v_{3i+1}^2/u_{3i+1};$$
$$v_{3i+2} = -g_{3i+1} v_{3i+1}/u_{3i+1}.$$

From (8) we obtain

$$x_0 = \sum_{i=0}^{k_1-1} (\alpha_{3i+1}\, p_{2i} + \alpha_{3i+2}\, p_{2i+1}),$$

where

$$\alpha_{3i+1} = b_{3i+1}/u_{3i+1};$$
$$\alpha_{3i+2} = (b_{3i+2} + v_{3i+1}\alpha_{3i+1})/u_{3i+2}$$

for $i = 0, 1, ..., m_1 - 1$ and for $i = m_1, ..., k_1 - 1$:

$$\alpha_{3i+1} = (b_{3i+1} - g_{3i+1-m}\alpha_{3i+1-m} + v_{3i+2-m}\alpha_{3i+2-m})/u_{3i+1};$$
$$\alpha_{3i+2} = (b_{3i+2} - g_{3i+2-m}\alpha_{3i+2-m} + v_{3i+1}\alpha_{3i+1})/u_{3i+2}.$$

We find the components of vector x_0 by the backward substitution again,

$$x_{0[3i+2]} = \alpha_{3i+2};$$
$$x_{0[3i+1]} = \alpha_{3i+1} + v_{3i+1} x_{0[3i+2]}/u_{3i+1} \tag{15}$$

for $i = k_1 - 1, ..., k_1 - m_1$ and for $i = k_1 - m_1 - 1, ..., 1$,

$$x_{0[3i+2]} = \alpha_{3i+2} + (v_{3i+2} x_{0[3i+1+m]} - g_{3i+2} x_{0[3i+2+m]})/u_{3i+2};$$
$$x_{0[3i+1]} = \alpha_{3i+1} + (v_{3i+1} x_{0[3i+2]} - g_{3i+1} x_{0[3i+1+m]})/u_{3i+1}. \tag{16}$$

The components of vector x_0 with the indices $3i$ are zeros.

The matrix–vector product $\tilde{q} = Rq$, where q is an arbitrary vector, is carried out in the following way. Those that are indivisible by three components of $\tilde{q}$ we find in two steps. At the first step we define the values

$$\gamma_1 = 0;$$
$$\gamma_2 = -f_2 q_3/u_2;$$
$$\gamma_{3i+1} = -f_{3i} q_{3i}/u_{3i+1};$$
$$\gamma_{3i+2} = -(f_{3i+2} q_{3(i+1)} - v_{3i+1}\gamma_{3i+1})/u_{3i+2}$$

for $i = 1, ..., m_1 - 1$ and for $i = m_1, ..., k_1 - 1$,

$$\gamma_{3i+1} = -(g_{3i+1-m}\gamma_{3i+1-m} + f_{3i} q_{3i} - v_{3i+2-m}\gamma_{3i+2-m})/u_{3i+1};$$
$$\gamma_{3i+2} = -(g_{3i+2-m}\gamma_{3i+2-m} + f_{3i+2} q_{3(i+1)} - v_{3i+1}\gamma_{3i+1})/u_{3i+2}.$$

At the second step we define $\alpha_{3i+1} = \gamma_{3i+1}$, $\alpha_{3i+2} = \gamma_{3i+2}$. Then we find the sum of the products $\gamma_i p_i$ according to (15) and (16). The values $x_{0[3i+1]}$, $x_{0[3i+2]}$ are equal in this case to the corresponding components of vector $\tilde{q} = Rq$.

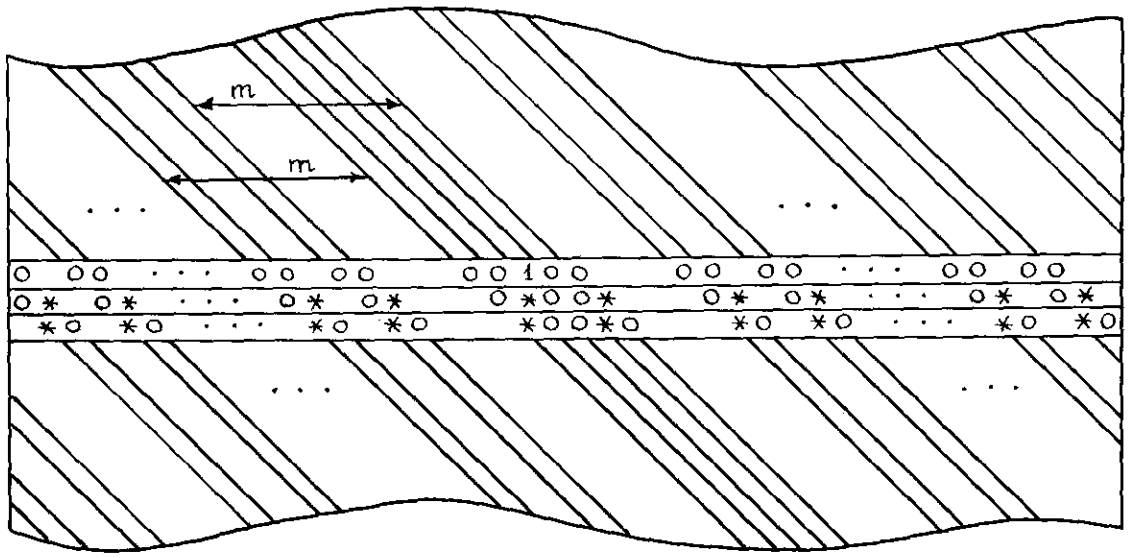

FIG. 4. Structure of projector for the IGICCG3-method.

The structure of projector R for the method under consideration is presented in Fig. 4, where nonzero entries of those that are indivisible by three rows are denoted by a "star" symbol. The ratios for nonzero elements of R will be omitted because of their complicity.

Note 3.1. The IGICCG2- and IGICCG3-algorithms do not require any additional memory for arrays U and V which can be placed, instead of elements of the main diagonal of the matrix A and residual r, with the numbers from N_n' that are unused in the iteration cycle.

3.2.3. Assuming $s=4$ the IGICCG4-algorithm can be obtained in a similar way. Here $k=3n/4$. However, it requires almost n additional memory cells. This algorithm has been constructed during the present work but we omit the corresponding ratios because of their complicity. IGICCG4 will not be used in numerical investigations but we shall present some estimates for it (see Section 4) with the aim of showing some limitations in the line of IGICCG-methods development.

Note 3.2. If we define $s=5, 6, \ldots$ then the corresponding IGICCG-methods can be obtained. We note that memory requirements increase significantly. These methods need not be developed because the cost of the matrix–vector product Rq also increases rapidly as can be seen from the next section.

4. SOME PREVIOUS COMMENTS

We present the efficiency estimate of the developed IGICCG-algorithms. Table I describes the preliminary steps cost and the iteration cost for each of the methods proposed. It also includes these costs for the traditional ICCG-method [3]. CPU times of double precision floating point multiplication, addition, and division are denoted as t_m, t_a, and t_d, respectively. Here we used an ordinary LU-decomposition of A, where $U=DL^T$ and L consists of three nonzero diagonals corresponding to nonzero diagonals of A. We implemented an iteration free of division for all the investigated methods. Therefore the preliminary steps cost increased slightly. Analysis of Table I shows that all of the developed methods have an iteration cost that is lower than the ICCG, whereas the preliminary steps costs may slightly exceed the costs of the corresponding iterations. Thus IGICCG-methods have some advantage.

TABLE I

Efficiency Estimate of the Investigated Methods

Method	Preliminary steps cost	Iteration cost	Cost of Rq
ICCG [3]	$8t_m+8t_a+3t_d$	$16t_m+13t_a$	0
IGICCG1	$10.5t_m+11t_a+4t_d$	$14t_m+10t_a$	$2.5t_m+1.5t_a$
IGICCG2	$12.5t_m+12.5t_a+4t_d$	$14t_m+10t_a$	$2.5t_m+1.5t_a$
IGICCG3	$15\frac{2}{3}t_m+14\frac{2}{3}t_a+5t_d$	$14\frac{2}{3}t_m+9\frac{2}{3}t_a$	$4\frac{2}{3}t_m+2\frac{2}{3}t_a$
IGICCG4	$16.75t_m+17.5t_a+8.25t_d$	$15.75t_m+10.5t_a$	$6.75t_m+4.25t_a$

Figure 5 presents the cost of iteration without preconditioning and computation of Rq (curve 1) and the cost of Rq (curve 2) versus k_1/n. Here the ICCG-method corresponds to $k_1/n=1$. The resulting curve (curve 3) shows that the methods IGICCG2 and IGICCG3 are approximately optimal ones and subsequent decreasing of k_1 yields to increasing the total iteration cost because the cost of Rq increases rapidly. We note that the IGICCG4-method proved to be less efficient than IGICCG3. Furthermore, it

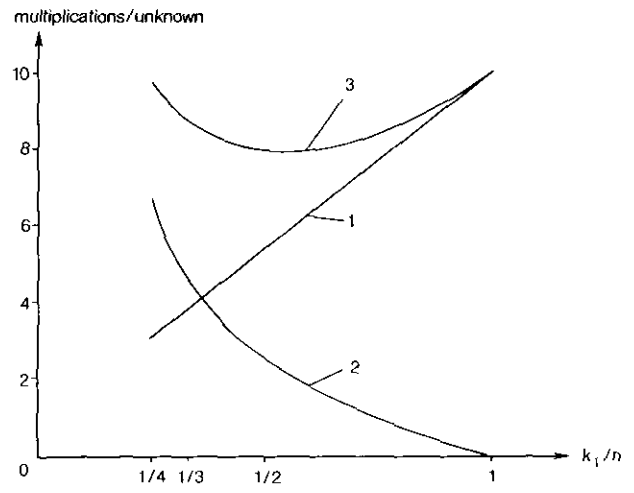

FIG. 5. Cost of iteration of investigated methods without precondition and cost of Rq products versus k_1/n.

requires additional memory as pointed out above. Hence to provide the correct comparison with IGICCG4 we can use an additional diagonal in incomplete decomposition to improve the ICCG-, IGICCG1-, IGICCG2-, and IGICCG3-methods performances. We shall not do that and IGICCG4 will not be used here.

From the previous description of the IGICCG, it is not clear whether or not these algorithms degrade because of their loss of orthogonality. Next we consider this problem in some more detail. Difference between the ICCG- and IGICCG-schemes is in the presence of an additional matrix–vector product $\tilde{q} = Rq$ in the latter. According to [20, p. 93] we can write

$$\|\Delta\tilde{q}\|_\infty \leqslant 2^{-t}(1+\delta)\,\|R\|_{l_\infty}\,\|q\|_\infty,$$

where t is number of binary digits in the mantissa of the floating-point word, δ is the number of nonzero entries in the matrix R row ($\delta = 4$ for IGICCG1 and $\delta = 2n/m$ for IGICCG2 and IGICCG3). It can be easily shown that all the three developed IGICCG-methods have $\|R\|_{l_\infty} \leqslant 1$ if only the linear system (1) matrix A is the diagonally dominant one. Then

$$\|\Delta\tilde{q}\|_\infty \leqslant 2^{-t}(1+\delta)\,\|q\|_\infty.$$

So the matrix–vector product under consideration leads to a slight (normally $n/m \leqslant 100$) increase of round-off error in the IGCG-methods. However, it is more important to estimate preconditioned versions of the methods proposed. In this case $q = (LU)^{-1} r$. Thus [21, p. 36]

$$\|\Delta q\|_\infty \leqslant 2^{-t} k_* n^{3/2} \kappa^* \,\|q\|_\infty,$$

where $k_* \sim 1$, $\kappa^* = \lambda_{\max}(LU)/\lambda_{\min}(LU)$. Hence if $(1+\delta) \leqslant k_* n^{3/2}\kappa^*$, then $\tilde{q}$ contains exactly the same number of true digits as q. Therefore the procedure of projection transformation $\tilde{q} = Rq$ will be stable. This result shows that if the preconditioned CG-method in the presence of a round-off error gives the solution of a linear system with the required accuracy, then this solution also can be obtained by using of all the three preconditioned methods proposed.

5. NUMERICAL INVESTIGATION

Consider the problem of two-dimensional steady-state numerical simulation of semiconductor devices. Drift-diffusion charge transfer equations [22] can be written as

$$\nabla^2\phi = n_{ie}(\exp(\phi)\,\Phi_n - \exp(-\phi)\,\Phi_p) - N_d + N_a, \quad (17)$$

$$\nabla\cdot(\mu_n n_{ie}\exp(\phi)\,\nabla\Phi_n) = R, \quad (18)$$

$$\nabla\cdot(\mu_p n_{ie}\exp(-\phi)\,\nabla\Phi_p) = R, \qquad (\phi, \Phi_n, \Phi_p) \in \Omega. \quad (19)$$

Boundary conditions are defined as

$$(\phi, \Phi_n, \Phi_p)|_{\partial\Omega_1} = (\phi_0, \Phi_{n0}, \Phi_{p0});$$

$$\left(\frac{\partial\phi}{\partial\eta}, \frac{\partial\Phi_n}{\partial\eta}, \frac{\partial\Phi_p}{\partial\eta}\right)\Bigg|_{\partial\Omega_2} = 0.$$

Here ϕ is the electrostatic potential, $\Phi_p = \exp(\phi_p)$, and $\Phi_n = \exp(-\phi_n)$, ϕ_p, ϕ_n are quasi-Fermi levels for holes and electrons, respectively. We have used standard models for the values μ_p, μ_n, R, n_{ie} which can be found, for example, in [23]. Note, that n_{ie}, N_d, N_a are given functions of spatial variables; R depends on the variables ϕ, Φ_n, Φ_p.

Finite-difference approximation of (17)–(19) is carried out on the continuous rectangular grid according to Scharfetter–Gummel's scheme [24]. We use one of the Gummel-like iteration procedures [25] for the nonlinear system (17)–(19) solution. Each of these equations in this case leads to the linear system with the five-diagonal symmetrical diagonally dominant M-matrix [26, p. 146].

Three structures of the bipolar transistor were simulated using the ICCG- and IGICCG-methods to solve linear systems. These structures are vertical (Fig. 6a), planar (Fig. 6b), and submicron shallow-profile. Impurity distribution of vertical and planar structures is

$$N = 5\times10^{16} + 5\times10^{20}\exp\{-7.00892 \times (x^2 + 0.1646272^2 y^2)^{1/2}\} - 5\times10^{18} \times\exp\{-|x|/2.298513\} + 5\times10^{20} t\ \mathrm{cm}^{-3},$$

where $t = 1$ for $x \geqslant 4$ and $t = 0$ for $x < 4$. The structure and doping profile of a submicron transistor is taken from [27].

Numerical experiments have been performed under the following conditions. Simulation was carried out under

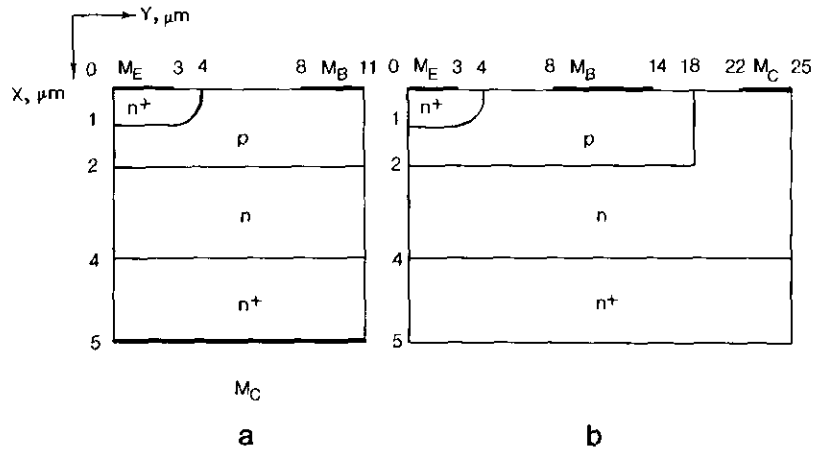

FIG. 6. The first and the second test problems ($\partial\Omega_1 = M_E + M_B + M_C$, $\partial\Omega_2 = \partial\Omega \setminus \partial\Omega_1$).

biases $V_{EB} = -0.6$ V, $V_{CB} = 1$ V (vertical and planar structures) and $V_{BE} = 0.5$ V, $V_{CE} = 1.9$ V (submicron transistor) on the grids of 24×25 mesh points (vertical and submicron devices) and 24×36 mesh points (planar structure). An initial approximation for variables ϕ, Φ_n, Φ_p was computed according to [26, p. 163].

As the matrices properties change weakly during outer iterations then the convergence characteristics are presented below only for the first outer iteration. Inner iterations were terminated as soon as $\|r_i\|_\infty / \|r_0\|_\infty < \varepsilon$, where i is number of inner iterations. All the experiments were carried out on an IBM PC AT 386/387 with double precision. The developed methods were compared to ICCG [3]. It can be pointed that one more efficient implementation of the PCG-method is known [28]. Residual vector $\hat{r}$ of the transformed system must be computed instead of system (1) residual r at each iteration of this algorithm. These residuals are connected by the ratio $r = (D + L)\hat{r}$, where L is strictly lower triangular of A and D is diagonal. Ordinary values of $\|(D+L)\|_{l_\infty}$ for systems to be solved are $\sim 10^{44}$ when the external bias $V = 1.9$ V. Therefore a small value of $\|\hat{r}\|_\infty$ can be obtained using the method [28], whereas $\|r\|_\infty$ will be large and the accuracy will be poor. The same results occurred in our numerical investigations for linear systems arising from both continuity equations (18) and (19). (It is worthwhile to point out that these systems are quite difficult to solve.) Therefore this method [28] cannot be used in our experiments.

TABLE III

Vertical Transistor Simulation ($\varepsilon = 10^{-12}$)

	Electrons			Holes		
Method	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	37	2.48	—	33	2.25	—
IGICCG1	36	1.92	1.29	32	1.76	1.28
IGICCG2	35	1.83	1.36	30	1.65	1.36
IGICCG3	32	1.65	1.50	28	1.43	1.57

Table II presents a number of inner iterations N_1 and the linear system solution time t for each of the investigated methods. In addition Table II includes the speedup factor τ for each of the developed methods. These results are obtained when $\varepsilon = 10^{-6}$. Analogous characteristics corresponding to $\varepsilon = 10^{-12}$ are presented in Table III. Note that the initial residual norms were $\|r_0\|_\infty = 0.1 \times 10^8$ and

TABLE II

Vertical Transistor Simulation ($\varepsilon = 10^{-6}$)

	Electrons			Holes		
Method	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	25	1.64	—	23	1.59	—
IGICCG1	26	1.45	1.13	22	1.24	1.28
IGICCG2	22	1.21	1.36	21	1.15	1.38
IGICCG3	17	0.94	1.74	19	1.05	1.51

TABLE IV

Planar Transistor Simulation ($\varepsilon = 10^{-6}$)

	Electrons			Holes		
Method	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	51	4.94	—	24	2.36	—
IGICCG1	50	3.95	1.25	23	1.81	1.30
IGICCG2	48	3.73	1.32	23	1.81	1.30
IGICCG3	47	3.52	1.40	22	1.60	1.48

TABLE V

Planar Transistor Simulation ($\varepsilon = 10^{-12}$)

Method	Electrons			Holes		
	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	64	6.26	—	38	3.73	—
IGICCG1	59	4.50	1.39	37	2.92	1.28
IGICCG2	59	4.50	1.39	37	2.92	1.28
IGICCG3	56	4.17	1.50	34	2.58	1.45

TABLE VII

Submicron Transistor Simulation ($\varepsilon = 10^{-12}$)

Method	Electrons			Holes			Potential		
	N_1	t, s	τ	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	65	4.28	—	52	3.46	—	16	1.10	—
IGICCG1	61	3.19	1.34	49	2.64	1.31	16	0.84	1.31
IGICCG2	48	2.88	1.66	41	2.20	1.57	11	0.66	1.67
IGICCG3	34	1.76	2.43	26	1.37	2.53	10	0.61	1.80

$\|r_0\|_\infty = 0.3 \times 10^7$ for the electron and hole continuity equations, respectively.

Results of the linear systems solution when the planar transistor was simulated are presented in Table IV ($\varepsilon = 10^{-6}$) and Table V ($\varepsilon = 10^{-12}$). The initial residual norms are the same. It can be pointed that the initial residual of Poisson's equation (17) proved to be small for both transistors and that the corresponding linear systems were solved by not more than four iterations due to the efficiency of the initial approximation choice method [26, p. 163] and because the linear system matrix in this case is the strongly diagonally dominant one. Therefore these results are not presented in Tables II, III, IV, and V.

Tables VI and VII present the above characteristics for $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-12}$, respectively, when a shallow-profile device was simulated. These tables consist of information about the solution of all three equations (17)–(19). Initial residuals are $\|r_0\|_\infty = 0.08$ for Poisson's equation and $\|r_0\|_\infty = 0.4 \times 10^7$, $\|r_0\|_\infty = 0.1 \times 10^7$ for electron and hole continuity equations, respectively.

As can be seen from Tables II–VII, efficiency of the methods proposed is not only due to lower cost of iteration but also it is due to a slight decrease in the number of iterations. Using the IGICCG4-method almost does not yield to the subsequent reduction of iterations performed, so the resulting efficiency of this method is almost similar to IGICCG3 (we must recall that IGICCG4 needs an additional memory).

Analysis of numerical investigation results indicates that all of the developed methods proved to be more efficient than the traditional one [3].

TABLE VI

Submicron Transistor Simulation ($\varepsilon = 10^{-6}$)

Method	Electrons			Holes			Potential		
	N_1	t, s	τ	N_1	t, s	τ	N_1	t, s	τ
ICCG [3]	49	3.24	—	20	1.31	—	9	0.60	—
IGICCG1	46	2.47	1.31	21	1.10	1.19	9	0.50	1.20
IGICCG2	36	1.98	1.64	21	1.10	1.19	7	0.38	1.58
IGICCG3	25	1.32	2.45	17	0.93	1.41	6	0.33	1.82

In conclusion of this section a few comments in order. Solution of the system (17)–(19) only is spatial distribution of the variables ϕ, Φ_n, Φ_p. External currents can be calculated on these bases. The use of all the investigated methods leads to exactly the same external currents. The balance of these currents is held to high accuracy ($\sim 10^{-9}$). We choose low injection regimes for all three transistors because in order to ensure fast convergence of outer iterations (in our experiments solution of (17)–(19) was obtained after five outer iterations for the first and the second problems and after four iterations for the third one) in this case we must solve linear systems with high accuracy. The stopping tolerance $\varepsilon = 10^{-6}$ is sufficient to provide this. It is worth pointing out that the second problem cannot be solved using such iterative procedures as SOR or a strongly implicit method [29] (see, for instance, [30; 26, p. 298]).

6. CONCLUSION

In this paper a technique of the initial guess choice for the CG-method has been proposed. The IGCG- and IGICCG-schemes have been developed on its basis for the solution of linear systems with the symmetrical positive definite matrices. Three preconditioned modifications of the IGICCG-method for systems with five-diagonal matrices have been proposed. Experimental evidence of their efficiency has been obtained using one of the important kinds of boundary value problems. It also can be pointed that all of the investigated methods can be improved by using another kinds of preconditioning.

ACKNOWLEDGMENT

We thank the anonymous referee for bringing the problems of finite arithmetics to our attention.

REFERENCES

1. M. R. Hestenes and E. Stiefel, *Nat. Bur. Stand. J. Res.* **49**, 409 (1952).
2. N. I. Buleev, *Mat. Sb.* **51**, 227 (1960). [Russian]
3. J. A. Meijerink and H. A. van der Vorst, *Math. Comput.* **31**, 148 (1977).
4. I. Gustafsson, *BIT* **18**, 142 (1978).
5. J. A. Meijerink and H. A. van der Vorst, *J. Comput. Phys.* **44**, 134 (1981).

6. P. Concus, G. H. Golub, and G. Meurant, *SIAM J. Sci. Stat. Comput.* **6**, 220 (1985).
7. D. S. Kershaw, *J. Comput. Phys.* **26**, 43 (1978).
8. O. Axelsson, *Linear Alg. Appl.* **29**, 1 (1980).
9. O. Axelsson, *Numer. Math.* **51**, 209 (1987).
10. P. Concus, G. H. Golub, and D. P. O'Leary, in *Sparse Matrix Computations*, edited by J. Bunch and D. J. Rose (Academic Press, New York, 1976), p. 332.
11. H. A. van der Vorst, *SIAM J. Sci. Stat. Comput.* **3**, 350 (1982).
12. E. Poole and J. Ortega, *SIAM J. Numer. Anal.* **24**, 1394 (1987).
13. H. A. van der Vorst, *Comput. Phys. Commun.* **53**, 223 (1989).
14. O. Axelsson, *BIT* **25**, 166 (1985).
15. R. A. Nicolaides, *SIAM J. Numer. Anal.* **24**, 355 (1987).
16. L. A. Zadeh and C. A. Desoer, *Linear System Theory* (McGraw–Hill, New York, 1963/Nauka, Moscow, 1970). [Russian translation]
17. B. N. Pshenichny and Yu. M. Danilin, *Numerical Methods in Extremal Problems* (Nauka, Moscow, 1975). [Russian]
18. F. R. Gantmacher, *The Theory of Matrices* (Nauka, Moscow, 1988). [Russian]
19. J. R. Rice, *Matrix Computations and Mathematical Software* (McGraw–Hill, New York, 1981).
20. G. E. Forsythe and C. B. Molet, *Computer Solution of Algebraic Systems* (Prentice–Hall, Englewood Cliffs, NJ, 1967).
21. J. H. Wilkinson and C. H. Reinsch, *Handbook for Automatic Computation. Linear Algebra, Vol. 2* (Springer-Verlag, New York, 1971/Machinostroeniye, Moscow, 1976). [Russian translation]
22. W. van Roosbroeck, *Bell Syst. Tech. J.* **29**, 560 (1950).
23. G. Baccarani, M. Rudan, R. Guerrieri, and P. Ciampolini, in *Process and Device Modeling*, edited by W. L. Engl (Elsevier Science, Amsterdam, 1986), p. 107.
24. D. L. Scharfetter and H. K. Gummel, *IEEE Trans. Electron Devices* **ED-16**, 64 (1969).
25. T. I. Seidman and S. C. Choo, *Solid-State Electron.* **15**, 1229 (1972).
26. S. G. Mulyarchik, *Numerical Simulation of Microelectron Structures* (Universitetskoe, Minsk, 1989). [Russian]
27. D. D. Tang, *IEEE Trans. Electron Devices* **ED-32**, 2224 (1985).
28. S. C. Eisenstat, *SIAM J. Sci. Stat. Comput.* **2**, 1 (1981).
29. H. L. Stone, *SIAM J. Numer. Anal.* **5**, 530 (1968).
30. S. Kumashiro and M. Sakurai, in *Proceedings, 4th Int. Conf. on the Numerical Analysis of Semiconductor Devices and Integrated Circuits (NASECODE), Dublin, Ireland, 1985*, edited by J. J. H. Miller (Boole, Dublin, 1985), p. 365.